Audio Toolkit provides an easy-to-use and performance optimized framework to play and manage music and sound effects in Unity.

## Classes

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AudioCategory</td>
<td>An audio category represents a set of AudioItems. Categories allow to change the volume containing audio items.</td>
</tr>
<tr>
<td>AudioController</td>
<td>The audio managing class used to define and play audio items and categories.</td>
</tr>
<tr>
<td>AudioItem</td>
<td>The AudioItem class represents a uniquely named audio entity that can be played by scripts.</td>
</tr>
<tr>
<td>AudioLog</td>
<td></td>
</tr>
<tr>
<td>AudioLogLogData</td>
<td></td>
</tr>
<tr>
<td>AudioLogLogData_Destroy</td>
<td></td>
</tr>
<tr>
<td>AudioLogLogData_PlayClip</td>
<td></td>
</tr>
<tr>
<td>AudioLogLogData_SkippedPlay</td>
<td></td>
</tr>
<tr>
<td>AudioLogLogData_Stop</td>
<td></td>
</tr>
<tr>
<td>AudioObject</td>
<td>The object playing the audio clip.</td>
</tr>
<tr>
<td>Class Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>AudioSubItem</td>
<td>An AudioSubItem represents a specific Unity audio clip.</td>
</tr>
<tr>
<td>ObjectPoolController</td>
<td>A static class used to create and destroy poolable objects.</td>
</tr>
<tr>
<td>Playlist</td>
<td>Allows to define a playlist consisting of a list of audio IDs.</td>
</tr>
<tr>
<td>PoolableObject</td>
<td>Add this component to your prefab to make it poolable.</td>
</tr>
<tr>
<td>PoolableReferenceT</td>
<td>Auxiliary class to overcome the problem of references to pooled objects that should become invalid when objects are moved back to the pool after calling Destroy(GameObject).</td>
</tr>
<tr>
<td>RegisteredComponent</td>
<td>Derive your MonoBehaviour from RegisteredComponent to manage all references to instances of this component. All references to instances of this component will be saved in an internal array. Use GetAllOfType to retrieve this array, which is much faster than using GameObject.FindObjectsOfType function.</td>
</tr>
<tr>
<td>RegisteredComponentController</td>
<td>This controller provides fast access to all currently existing instances of RegisteredComponent in the scene.</td>
</tr>
<tr>
<td>SingletonMonoBehaviourT</td>
<td>Provides singleton-like access to a unique instance of a MonoBehaviour.</td>
</tr>
</tbody>
</table>
### Interfaces

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRegisteredComponent</td>
<td></td>
</tr>
</tbody>
</table>

### Delegates

<table>
<thead>
<tr>
<th>Delegate</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AudioObjectAudioEventDelegate</td>
<td>The audio event delegate type.</td>
</tr>
</tbody>
</table>

###Enumerations

<table>
<thead>
<tr>
<th>Enumeration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AudioItemLoopMode</td>
<td>AudioItem loop mode.</td>
</tr>
<tr>
<td>AudioPickSubItemMode</td>
<td>Used by AudioItem to determine which AudioSubItem is chosen.</td>
</tr>
<tr>
<td>AudioSubItemType</td>
<td>The type of an AudioSubItem</td>
</tr>
</tbody>
</table>
AudioCategory Class

An audio category represents a set of AudioItems. Categories allow to change the volume of all containing audio items.

Inheritance Hierarchy

- System
  - Object
    - (Default Namespace)AudioCategory

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```csharp
[SerializableAttribute]
public class AudioCategory
```

The AudioCategory type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AudioCategory</td>
<td>Instantiates an AudioCategory</td>
</tr>
</tbody>
</table>

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>audioController</td>
<td>The AudioController the category belongs to</td>
</tr>
</tbody>
</table>
isFadeOutComplete  return true if the category has completely faded out

isFadingIn  return true if the category is currently fading in

isFadingOut  return true if the category is currently fading out

parentCategory  If a parent category is specified, the category inherits the volume of its parent.

Volume  The volume factor applied to all audio items in the category (NOT including a possible parentCategory) If you change the volume by script the change will be applied to all playing audios immediately.

VolumeTotal  The volume factor applied to all audio items in the category (including a possible parentCategory and fade out/in)

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FadeIn</td>
<td>Starts a fade-in of the audio category.</td>
</tr>
<tr>
<td>FadeOut</td>
<td>Starts a fade-out of the</td>
</tr>
</tbody>
</table>
audio category.

- **GetAudioMixerGroup** Retrieves the AudioMixerGroup associated with this category. AudioMixerGroupa are inherited by the parent category.

- **GetAudioObjectPrefab** Retrieves the AudioObjectPrefab associated with this category. AudioObjectPrefabs are inherited by the parent category.

- **UnloadAllAudioClips** Unloads all AudioClips specified in the subitems from memory.

### Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AudioItems</strong></td>
<td>Define your AudioItems using Unity inspector.</td>
</tr>
<tr>
<td><strong>audioMixerGroup</strong></td>
<td>Allows to assign the category to a Unity 5 Audio Mixer Group</td>
</tr>
<tr>
<td><strong>AudioObjectPrefab</strong></td>
<td>Allows to define a specific audio object prefab for this category. If none is defined, the default prefab as set by</td>
</tr>
</tbody>
</table>
AudioObjectPrefab is taken.

<table>
<thead>
<tr>
<th>Name</th>
<th>The name of category ( = categoryID )</th>
</tr>
</thead>
</table>

See Also

Reference

(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioCategory Constructor

Instantiates an AudioCategory

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

## Syntax

```c#
public AudioCategory(
    AudioController audioController
)
```

### Parameters

*audioController*

Type: (Default Namespace)AudioController

The AudioController the category belongs to.

## See Also

**Reference**

AudioCategory Class
( Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
# AudioCategory Properties

The `AudioCategory` type exposes the following members.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>audioController</code></td>
<td>The <code>AudioController</code> the category belongs to</td>
</tr>
<tr>
<td><code>isFadeOutComplete</code></td>
<td>return <code>true</code> if the category has completely faded out</td>
</tr>
<tr>
<td><code>isFadingIn</code></td>
<td>return <code>true</code> if the category is currently fading in</td>
</tr>
<tr>
<td><code>isFadingOut</code></td>
<td>return <code>true</code> if the category is currently fading out</td>
</tr>
<tr>
<td><code>parentCategory</code></td>
<td>If a parent category is specified, the category inherits the volume of its parent.</td>
</tr>
<tr>
<td><code>Volume</code></td>
<td>The volume factor applied to all audio items in the category (NOT including a possible <code>parentCategory</code>) If you change the volume by script the change will be applied to all playing audios immediately.</td>
</tr>
<tr>
<td><code>VolumeTotal</code></td>
<td>The volume factor applied to all audio items in the</td>
</tr>
</tbody>
</table>
See Also

Reference

AudioCategory Class
(Default Namespace) Namespace
AudioCategoryaudioController Property

The AudioController the category belongs to

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```c#
public AudioController audioController { get; set; }
```

Property Value
Type: AudioController

See Also

Reference
AudioCategory Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioCategory isFadeOutComplete Property

return true if the category has completely faded out

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

C#

```csharp
public bool isFadeOutComplete { get; }
```

Property Value
Type: Boolean

See Also

Reference
AudioCategory Class
(Default Namespace) Namespace
AudioCategory

Property

return true if the category is currently fading in

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```csharp
public bool isFadingIn { get; }
```

Property Value
Type: Boolean

See Also

Reference
AudioCategory Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioCategory\isFadingOut\ Property

return \texttt{true} if the category is currently fading out

\textbf{Namespace:} (Default Namespace)
\textbf{Assembly:} AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

\section*{Syntax}

\begin{verbatim}
public bool isFadingOut { get; }
\end{verbatim}

\section*{Property Value}

Type: \texttt{Boolean}

\section*{Remarks}

If the fade-out is complete then \texttt{isFadingOut} return \texttt{false} and \texttt{isFadeOutComplete} returns \texttt{true}

\section*{See Also}

Reference
\begin{verbatim}
AudioCategory Class
(Default Namespace) Namespace
\end{verbatim}
AudioCategory

parentCategory

Property

If a parent category is specified, the category inherits the volume of its parent.

Namespace:  (Default Namespace)
Assembly:  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

C#

```c#
public AudioCategory parentCategory { get; set; }
```

Property Value

Type: AudioCategory

See Also

Reference

AudioCategory Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioCategoryVolume Property

The volume factor applied to all audio items in the category (NOT including a possible parentCategory) If you change the volume by script the change will be applied to all playing audios immediately.

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```c#
public float Volume { get; set; }
```

Property Value
Type: Single

See Also

Reference
AudioCategory Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioCategoryVolumeTotal Property

The volume factor applied to all audio items in the category (including a possible parentCategory and fade out/in)

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```c#
public float VolumeTotal { get; }
```

**Property Value**
Type: Single

### See Also

Reference
AudioCategory Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
# AudioCategory Methods

The `AudioCategory` type exposes the following members.

## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="faadin" alt="faadin" /> Fadin</td>
<td>Starts a fade-in of the audio category.</td>
</tr>
<tr>
<td><img src="faadout" alt="faadout" /> FadeOut</td>
<td>Starts a fade-out of the audio category.</td>
</tr>
<tr>
<td><img src="getmix" alt="getmix" /> GetAudioMixerGroup</td>
<td>Retrieves the AudioMixerGroup associated with this category. AudioMixerGroupa are inherited by the parent category.</td>
</tr>
<tr>
<td><img src="getobjc" alt="getobjc" /> GetAudioObjectPrefab</td>
<td>Retrieves the AudioObjectPrefab associated with this category. AudioObjectPrefabs are inherited by the parent category.</td>
</tr>
<tr>
<td><img src="unload" alt="unload" /> UnloadAllAudioClips</td>
<td>Unloads all AudioClips specified in the subitems from memory.</td>
</tr>
</tbody>
</table>
See Also

Reference
AudioCategory Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioCategoryFadeIn Method

Starts a fade-in of the audio category.

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public void FadeIn(
    float fadeInTime,
    bool stopCurrentFadeOut = true
)
```

### Parameters

- **fadeInTime**
  - Type: **SystemSingle**
  - The fade time in seconds.
- **stopCurrentFadeOut (Optional)**
  - Type: **SystemBoolean**
  - In case of an existing fade-out this parameter determines if the fade-out is stopped.

### See Also

Reference
- **AudioCategory Class**
- (Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioCategoryFadeOut Method

Starts a fade-out of the audio category.

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

**C#**

```csharp
public void FadeOut(
    float fadeOutLength,
    float startToFadeTime = 0f
)
```

### Parameters

- **fadeOutLength**
  - Type: System.Single
  - The fade time in seconds. If a negative value is specified, the fade out as specified in the corresponding FadeOut is used

- **startToFadeTime (Optional)**
  - Type: System.Single
  - Fade out starts after startToFadeTime seconds have passed

### Remarks

If the category is already fading out the requested fade-out is combined with the existing one.

### See Also

Reference
- AudioCategory Class
- (Default Namespace) Namespace
AudioCategory.GetAudioMixerGroup Method

Retrieves the AudioMixerGroup associated with this category. AudioMixerGroupa are inherited by the parent category.

Namespace:  (Default Namespace)
Assembly:  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```c#
public AudioMixerGroup GetAudioMixerGroup()
```

Return Value
Type: AudioMixerGroup
The AudioMixerGroup associated with this category.

See Also

Reference
AudioCategory Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioCategoryGetAudioObjectPrefab Method

Retrieves the AudioObjectPrefab associated with this category. AudioObjectPrefabs are inherited by the parent category.

**Namespace:** (Default Namespace)  
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

## Syntax

```c#
public GameObject GetAudioObjectPrefab()
```

## Return Value
Type: **GameObject**  
The AudioObjectPrefab associated with this category.

## See Also

**Reference**  
AudioCategory Class  
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH  
Send comments on this topic to ClockStone Support Email
AudioCategoryUnloadAllAudioClips Method

Unloads all AudioClips specified in the subitems from memory.

**Namespace:**  (Default Namespace)
**Assembly:**  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

⚠️ Syntax

```
public void UnloadAllAudioClips()
```

⚠️ Remarks

You will still be able to play the AudioClips, but you may experience performance hickups when Unity reloads the audio asset

⚠️ See Also

Reference
AudioCategory Class
(Default Namespace) Namespace
AudioCategory Fields

The `AudioCategory` type exposes the following members.

**Fields**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>AudioItems</code></td>
<td>Define your AudioItems using Unity inspector.</td>
</tr>
<tr>
<td><code>audioMixerGroup</code></td>
<td>Allows to assign the category to a Unity 5 Audio Mixer Group</td>
</tr>
<tr>
<td><code>AudioObjectPrefab</code></td>
<td>Allows to define a specific audio object prefab for this category. If none is defined, the default prefab as set by <code>AudioObjectPrefab</code> is taken.</td>
</tr>
<tr>
<td><code>Name</code></td>
<td>The name of category ( = <code>categoryID</code> )</td>
</tr>
</tbody>
</table>

See Also

Reference

`AudioCategory Class`

(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioCategory AudioItems Field

Define your AudioItems using Unity inspector.

**Namespace:** (Default Namespace)

**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```
public AudioItem[] AudioItems
```

**Field Value**

Type: AudioItem

### See Also

**Reference**

AudioCategory Class

(Default Namespace) Namespace

---

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioCategory

Field

Allows to assign the category to a Unity 5 Audio Mixer Group

**Namespace:**  (Default Namespace)

**Assembly:**  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```c#
public AudioMixerGroup audioMixerGroup
```

Field Value

Type: **AudioMixerGroup**

### See Also

**Reference**

AudioCategory Class

(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
**AudioCategory.AudioObjectPrefab**

**Field**

Allows to define a specific audio object prefab for this category. If none is defined, the default prefab as set by `AudioObjectPrefab` is taken.

**Namespace:** (Default Namespace)

**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

**Syntax**

```csharp
public GameObject AudioObjectPrefab
```

**Field Value**

Type: `GameObject`

**Remarks**

This way you can e.g. use special effects such as the reverb filter for a specific category. Just add the respective filter component to the specified prefab.

**See Also**

Reference

AudioCategory Class

(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioCategoryName Field

The name of category ( = categoryID )

Namespace:  (Default Namespace)
Assembly:  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```
public string Name
```

Field Value
Type:  String

See Also

Reference
AudioCategory Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioController Class

The audio managing class used to define and play audio items and categories.

Inheritance Hierarchy

- System
  - Object
  - Object
  - Component
  - Behaviour
  - MonoBehaviour
    - (Default Namespace)SingletonMonoBehaviour
    - (Default Namespace)AudioController

Namespace:  (Default Namespace)
Assembly:  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```c#
public class AudioController : SingletonMonoBehaviour ISerializationCallbackReceiver
```

The AudioController type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>🔄 AudioController</td>
<td>Initializes a new instance of the AudioController class</td>
</tr>
</tbody>
</table>

Top
### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ambienceSoundCrossFadeTime_In</td>
<td>Specifies a specific fade-in time for ambience sound cross fading. Only meaningful if specifyCrossFadeIn enabled.</td>
</tr>
<tr>
<td>ambienceSoundCrossFadeTime_Out</td>
<td>Specifies a specific fade-out time for ambience sound cross fading. Only meaningful if specifyCrossFadeIn enabled.</td>
</tr>
<tr>
<td>ambienceSoundEnabled</td>
<td>Gets or sets the ambienceSoundEnabled.</td>
</tr>
<tr>
<td>DisableAudio</td>
<td>Disables all audio playback.</td>
</tr>
<tr>
<td>isAdditionalAudioController</td>
<td>You may use several AudioControllers in the same scene in parallel. All but one (the main controller) must be marked as 'additional'. You can play audio items of any of those controllers with the normal Play() calls.</td>
</tr>
<tr>
<td>isSingletonObject</td>
<td>returns true if the AudioController is the main controller (not an additional controller). (Overrides SingletonMonoBehaviour)</td>
</tr>
<tr>
<td>musicCrossFadeTime_In</td>
<td>Specifies a specific fade-in time for music cross fading. Only meaningful if specifyCrossFadeIn enabled.</td>
</tr>
<tr>
<td>musicCrossFadeTime_Out</td>
<td>Specifies a specific fade-out time for music cross fading. Only meaningful if specifyCrossFadeIn enabled.</td>
</tr>
</tbody>
</table>
**musicEnabled**

Gets or sets the music enabled.

**soundMuted**

Gets or sets the sound muted.

**systemDeltaTime**

Returns the high precision audio system delta time since the last frame update.

**systemTime**

Returns the high precision audio system time size the application launched.

**Volume**

The global volume applied to all categories. You change the volume by script and the change will be applied immediately.

---

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="AddPlaylist" /> AddPlaylist</td>
<td>Adds a new playlist.</td>
</tr>
<tr>
<td><img src="image" alt="AddToCategory" /> AddToCategory(AudioCategory, AudioItem)</td>
<td>Adds a custom audio item to a category.</td>
</tr>
<tr>
<td><img src="image" alt="AddToCategory" /> AddToCategory(AudioCategory, AudioClip, String)</td>
<td>Creates an AudioItem with the name <code>audioID</code> containing a single subitem playing the specified custom AudioClip. This AudioItem is then added to the specified category.</td>
</tr>
<tr>
<td><img src="image" alt="ClearPlaylists" /> ClearPlaylists</td>
<td>Clears all music playlists.</td>
</tr>
</tbody>
</table>
Detach all audio objects possibly parented to the specified game object.

Enables the ambience sound.

Enables the music.

Enqueues an audio ID to the music playlist queue.

Starts a fade-in of an audio category.

Starts a fade-out of an audio category.

Returns the `AudioItem` with the given `audioID`.

Gets the audio item's max distance. (respects all proper default values and overwrites).

Gets a category.

Gets the category volume.

Gets the current ambience sound.
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetCurrentAudioListener</td>
<td>Gets the currently active Unity audio listener.</td>
</tr>
<tr>
<td>GetCurrentMusic</td>
<td>Gets the current music.</td>
</tr>
<tr>
<td>GetGlobalVolume</td>
<td>Gets the global volume.</td>
</tr>
<tr>
<td>GetMusicPlaylist</td>
<td>If playlistName == null, then gets a copy of the current playlist audioID array, otherwise gets a copy of the playlist with the specified name</td>
</tr>
<tr>
<td>GetPlayingAudioObjects(Boolean)</td>
<td>Returns an array of all playing audio objects.</td>
</tr>
<tr>
<td>GetPlayingAudioObjects(String, Boolean)</td>
<td>Returns an array of all playing audio objects with the specified audioID.</td>
</tr>
<tr>
<td>GetPlayingAudioObjectsCount</td>
<td>Returns the number of all playing audio objects with the specified audioID.</td>
</tr>
<tr>
<td>GetPlayingAudioObjectsInCategory</td>
<td>Returns an array of all playing audio objects in the category with name categoryName.</td>
</tr>
<tr>
<td>GetPlaylistByName</td>
<td>Retrieves a playlist by name. If playlists are</td>
</tr>
</tbody>
</table>
InitializeAudioItems
 Updates the internal `audioID` dictionary and initializes all registered `AudioItem` objects.

IsAmbienceSoundEnabled
 Determines whether ambience sound is enabled.

IsAmbienceSoundPaused
 Uses to test if ambience sound is paused.

IsMusicEnabled
 Determines whether music is enabled.

IsMusicPaused
 Uses to test if music is paused.

IsPlaying
 Determines whether the specified audio ID is playing.

IsPlaylistPlaying
 Determines whether the playlist is playing.

IsSoundMuted
 Determines whether sound is muted.

IsValidAudioID
 Tests if a given `audioID` is valid.

MuteSound
 Mutes / Unmutes the sound.
<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NewCategory</td>
<td>Creates a new audio category</td>
</tr>
<tr>
<td>OnAfterDeserialize</td>
<td></td>
</tr>
<tr>
<td>OnBeforeSerialize</td>
<td></td>
</tr>
<tr>
<td>PauseAll</td>
<td>Pauses all playing audio items (including the music).</td>
</tr>
<tr>
<td>PauseAmbienceSound</td>
<td>Pauses the currently playing ambience sound.</td>
</tr>
<tr>
<td>PauseCategory</td>
<td>Pauses all playing audio items in the specified category (including the music).</td>
</tr>
<tr>
<td>PauseMusic</td>
<td>Pauses the currently playing music.</td>
</tr>
<tr>
<td>Play(String)</td>
<td>Plays an audio item with the name <code>audioID</code></td>
</tr>
<tr>
<td>Play(String, Transform)</td>
<td>Plays an audio item with the name <code>audioID</code> parented to a specified transform.</td>
</tr>
<tr>
<td>Play(String, Vector3, Transform)</td>
<td>Plays an audio item with the name <code>audioID</code> parented to a specified transform with a world offset.</td>
</tr>
</tbody>
</table>
| Play(String, Single, Single, Single) | Plays an audio item }
with the name \texttt{audioID}.

<table>
<thead>
<tr>
<th>Method</th>
<th>Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![S]</td>
<td>\texttt{Play(String, Transform, Single, Single, Single)}</td>
<td>Plays an audio item with the name \texttt{audioID} parented to a specified transform.</td>
</tr>
<tr>
<td>![S]</td>
<td>\texttt{Play(String, Vector3, Transform, Single, Single, Single)}</td>
<td>Plays an audio item with the name \texttt{audioID} parented to a specified transform with a world offset.</td>
</tr>
<tr>
<td>![S]</td>
<td>\texttt{PlayAfter}</td>
<td>Plays an audio item with the name \texttt{audioID} right after the given \texttt{AudioObject} stops playing. (see the Unity \texttt{AudioSettings.dspTime} documentation)</td>
</tr>
<tr>
<td>![S]</td>
<td>\texttt{PlayAmbienceSound(String, Single, Single, Single)}</td>
<td>Plays an audio item with the name \texttt{audioID} as ambience sound.</td>
</tr>
<tr>
<td>![S]</td>
<td>\texttt{PlayAmbienceSound(String, Transform, Single, Single, Single)}</td>
<td>Plays an audio item with the name \texttt{audioID} as ambience sound at the specified position.</td>
</tr>
<tr>
<td>![S]</td>
<td>\texttt{PlayAmbienceSound(String, Vector3, Transform, Single, Single, Single)}</td>
<td>Plays an audio item with the name \texttt{audioID} as ambience sound at the specified position.</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>PlayAudioItem</td>
<td>Plays a specific AudioItem.</td>
<td></td>
</tr>
<tr>
<td>PlayAudioSubItem</td>
<td>Plays a specific AudioSubItem.</td>
<td></td>
</tr>
<tr>
<td>PlayMusic(String, Single, Single, Single)</td>
<td>Plays an audio item with the name <code>audioID</code> as music.</td>
<td></td>
</tr>
<tr>
<td>PlayMusic(String, Transform, Single, Single, Single)</td>
<td>Plays an audio item with the name <code>audioID</code> as music at the specified position.</td>
<td></td>
</tr>
<tr>
<td>PlayMusic(String, Vector3, Transform, Single, Single, Single)</td>
<td>Plays an audio item with the name <code>audioID</code> as music at the specified position.</td>
<td></td>
</tr>
<tr>
<td>PlayMusicPlaylist</td>
<td>Start playing the music playlist.</td>
<td></td>
</tr>
<tr>
<td>PlayNextMusicOnPlaylist</td>
<td>Jumps to the next music track on the playlist.</td>
<td></td>
</tr>
<tr>
<td>PlayPreviousMusicOnPlaylist</td>
<td>Jumps to the previous music track on the playlist.</td>
<td></td>
</tr>
<tr>
<td>PlayScheduled</td>
<td>Plays an audio item with the name <code>audioID</code> parented to a specified transform with a world offset scheduled at a specified high precision DSP time</td>
<td></td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>RemoveAudioItem</td>
<td>Removes an AudioItem from the AudioController.</td>
<td></td>
</tr>
<tr>
<td>RemoveCategory</td>
<td>Removes an audio category.</td>
<td></td>
</tr>
<tr>
<td>SetCategoryVolume</td>
<td>Changes the category volume. Also affects currently playing audio items.</td>
<td></td>
</tr>
<tr>
<td>SetCurrentMusicPlaylist</td>
<td>Sets the current playlist to the specified audioID array.</td>
<td></td>
</tr>
<tr>
<td>SetGlobalVolume</td>
<td>Changes the global volume. Effects all currently playing audio items.</td>
<td></td>
</tr>
<tr>
<td>Stop(String)</td>
<td>Stops all playing audio items with name audioID.</td>
<td></td>
</tr>
<tr>
<td>Stop(String, Single)</td>
<td>Stops all playing audio items with name audioID with a fade-out.</td>
<td></td>
</tr>
<tr>
<td>StopAll</td>
<td>Immediately stops playing audio items (including the music).</td>
<td></td>
</tr>
<tr>
<td>StopAll(Single)</td>
<td>Fades out all playing</td>
<td></td>
</tr>
</tbody>
</table>

(see the Unity AudioSettings.dspTime documentation)
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>StopAmbienceSound</td>
<td>Stops the currently playing ambience sound.</td>
</tr>
<tr>
<td>StopAmbienceSound(Single)</td>
<td>Stops the currently playing ambience sound with fade-out.</td>
</tr>
<tr>
<td>StopCategory</td>
<td>Stops all playing audio items in the specified category (including the music).</td>
</tr>
<tr>
<td>StopMusic</td>
<td>Stops the currently playing music.</td>
</tr>
<tr>
<td>StopMusic(Single)</td>
<td>Stops the currently playing music with fade-out.</td>
</tr>
<tr>
<td>UnloadAllAudioClips</td>
<td>Unloads all AudioClips specified in this AudioController from memory.</td>
</tr>
<tr>
<td>UnpauseAll</td>
<td>Un-pauses all playing audio items (including the music).</td>
</tr>
<tr>
<td>UnpauseAmbienceSound</td>
<td>Unpauses the current ambience sound.</td>
</tr>
<tr>
<td>UnpauseCategory</td>
<td>Un-pauses all playing audio items in the specified category (including the music).</td>
</tr>
</tbody>
</table>
UnpauseMusic

Unpauses the current music.

**Top**

**Fields**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>_currentInspectorSelection</td>
<td></td>
</tr>
<tr>
<td>ambienceSoundCrossFadeTime</td>
<td>If set to a value &gt; 0, music will automatically be cross-faded with this fade-in time.</td>
</tr>
<tr>
<td>AUDIO_TOOLKIT_VERSION</td>
<td>A string containing the version number of the Audio Toolkit.</td>
</tr>
<tr>
<td>AudioCategories</td>
<td>Specify your audio categories here using the Unity inspector.</td>
</tr>
<tr>
<td>AudioObjectPrefab</td>
<td>You must specify your AudioObject prefab here using the Unity inspector.</td>
</tr>
<tr>
<td>crossfadePlaylist</td>
<td>If enabled, the tracks on the playlist will get cross-faded as specified by musicCrossFadeTime.</td>
</tr>
<tr>
<td>delayBetweenPlaylistTracks</td>
<td>Mute time in between the playlist.</td>
</tr>
<tr>
<td>EqualPowerCrossfade</td>
<td>If enabled, fading is adjusted in a way that cross-fades result in the same power during fading.</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>loopPlaylist</td>
<td>specifies if the music playlist will get looped</td>
</tr>
<tr>
<td>musicCrossFadeTime</td>
<td>If set to a value &gt; 0, music will automatically be cross-faded with this fading time.</td>
</tr>
<tr>
<td>musicPlaylist</td>
<td><strong>Obsolete.</strong> For backwards compatibility we still keep the old playlist system; items will be copied over to the Default Playlist and then cleared.</td>
</tr>
<tr>
<td>musicPlaylists</td>
<td>allows to specify a list of named playlists that can be played as music</td>
</tr>
<tr>
<td>Persistent</td>
<td>If enabled, the audio controller will survive scene changes</td>
</tr>
<tr>
<td>PlayWithZeroVolume</td>
<td>If disabled, audio clips are not played if they have a resulting volume of zero.</td>
</tr>
<tr>
<td>shufflePlaylist</td>
<td>enables / disables shuffling for the music playlist</td>
</tr>
<tr>
<td>specifyCrossFadeInAndOutSeperately</td>
<td>If enabled specific cross-fade in and out times can be specified using musicCrossFadeTime_In and musicCrossFadeTime_Out and ambienceSoundCrossFadeTime_In and ambienceSoundCrossFadeTime_Out respectively</td>
</tr>
<tr>
<td>UnloadAudioClipsOnDestroy</td>
<td>If enabled all audio resources (AudioClips) specified in this AudioController are unloaded from memory when the AudioController is destroyed</td>
</tr>
</tbody>
</table>
gets destroyed (e.g. when loading a new scene and Persistent enabled)

- **UsePooledAudioObjects**
  Enables / Disables AudioObject pooling

### Remarks

At least one instance of an AudioController must exist in each scene using the Audio Toolkit. Usually there is exactly one controller, but you can have additional controllers if they are marked as such (in the Unity inspector, see `isAdditionalAudioController`) There are two options when setting up an AudioController. Either you can specify all audio files that are used in your entire project in one single AudioController. Then add this AudioController to your initial scene and set it persistent from within the inspector, so it will survive when a new scene is loaded. This way all audios are accessible from within your entire application. If you have a lot of audio files though, this may lead to a lengthy loading time and will have a rather large memory footprint. To avoid this, you can alternatively set up a specific AudioController for each scene which only contains those audio files needed in the particular scene.

### Examples

Once you have defined your audio categories and items in the Unity inspector you can play music and sound effects very easily:

```csharp
AudioController.Play( "MySoundEffect1" );
AudioController.Play( "MySoundEffect2", new Vector3(posX, posY, posZ) );
AudioController.PlayMusic( "MusicTrack1" );
AudioController.SetCategoryVolume( "Music", 0.5f );
AudioController.PauseMusic();
```
See Also

Reference
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioController Constructor

Initializes a new instance of the AudioController class

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public AudioController()
```

### See Also

Reference
AudioController Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioController Properties

The **AudioController** type exposes the following members.

## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ambienceSoundCrossFadeTime_In</td>
<td>Specifies a specific fade-in time for ambience sound cross fading. Only meaningful if specifyCrossFadeIn enabled.</td>
</tr>
<tr>
<td>ambienceSoundCrossFadeTime_Out</td>
<td>Specifies a specific fade-out time for ambience sound cross fading. Only meaningful if specifyCrossFadeIn enabled.</td>
</tr>
<tr>
<td>ambienceSoundEnabled</td>
<td>Gets or sets the ambienceSoundEnabled.</td>
</tr>
<tr>
<td>DisableAudio</td>
<td>Disables all audio playback.</td>
</tr>
<tr>
<td>isAdditionalAudioController</td>
<td>You may use several AudioControllers in the same scene in parallel (i.e., one main controller) must be marked as 'additional'. You can play audio items of any of those controllers with the normal Play() calls.</td>
</tr>
<tr>
<td>isSingletonObject</td>
<td>returns true if the AudioController is the main controller (not an additional controller). (Overrides SingletonMonoBehaviour.isSingletonObject)</td>
</tr>
<tr>
<td>musicCrossFadeTime_In</td>
<td>Specifies a specific fade-in time for music cross fading. Only meaningful if musicCrossFadeOut enabled.</td>
</tr>
<tr>
<td>Music CrossFade Time Out</td>
<td>Specifies a specific fade-out time for music cross fading. Only meaningful if <code>specifyCrossFadeInAndOutSeperately</code> enabled.</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>MusicEnabled</td>
<td>Gets or sets the music enabled status.</td>
</tr>
<tr>
<td>SoundMuted</td>
<td>Gets or sets the sound muted status.</td>
</tr>
<tr>
<td>System Delta Time</td>
<td>Returns the high precision time since the last frame update.</td>
</tr>
<tr>
<td>System Time</td>
<td>Returns the high precision time since the application launch.</td>
</tr>
<tr>
<td>Volume</td>
<td>The global volume applied to all categories. You change the volume by script and the change will be applied immediately.</td>
</tr>
</tbody>
</table>

**See Also**

Reference

AudioController Class
(Default Namespace) Namespace
AudioControllerambiencSoundCrossFadeTime_In Property

Specifies a specific fade-in time for ambience sound cross fading. Only meaningful if specifyCrossFadeInAndOutSeperately is enabled.

Namespace: (Default Namespace)  
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```csharp
public float ambienceSoundCrossFadeTime_In { get; }
```

Property Value
Type: Single

See Also

Reference  
AudioController Class  
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioControllerambiencSoundCrossFadeTime_Out Property

Specifies a specific fade-out time for ambience sound cross fading. Only meaningful if specifyCrossFadeInAndOutSeparately is enabled.

**Namespace:** (Default Namespace)

**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public float ambienceSoundCrossFadeTime_Out { get }
```

**Property Value**

Type: Single

### See Also

**Reference**
- AudioController Class
- (Default Namespace) Namespace
AudioControllerAmbienceSoundEnabled Property

Gets or sets the ambienceSoundEnabled.

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

**Syntax**

```csharp
public bool ambienceSoundEnabled { get; set; }
```

**Property Value**
Type: Boolean
true enables ambience sounds; false disables ambience sounds

**See Also**

Reference
AudioController Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioControllerDisableAudio

Property

Disables all audio playback.

**Namespace:** (Default Namespace)

**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public bool DisableAudio { get; set; }
```

### Property Value

Type: **Boolean**

### Remarks

Does not stop currently playing audios. Call `StopAll` to stop all currently playing.

### See Also

Reference

- AudioController Class
- (Default Namespace) Namespace

---

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to [ClockStone Support Email](mailto:ClockStoneSupportEmail)
AudioController

isAdditionalAudioController Property

You may use several AudioControllers in the same scene in parallel. All but one (the main controller) must be marked as 'additional'. You can play audio items of any of those controllers with the normal Play() calls.

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```c#
public bool isAdditionalAudioController { get; set; }
```

Property Value
Type: Boolean

Remarks

This can be used for games with a large amount of audio where you don't want all audio to be in memory at all time. In this case use a persistent main AudioController for audios shared between all scenes of your game, and additional AudioControllers for each scene containing specific audio for this level.

See Also

Reference
AudioController Class
(Default Namespace) Namespace
AudioControllerisSingletonObject Property

returns true if the AudioController is the main controller (not an additional controller)

Namespace:  (Default Namespace)
Assembly:  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```csharp
public override bool isSingletonObject { get; }
```

Property Value
Type: Boolean

See Also

Reference
AudioController Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioController\musicCrossFadeTime_In Property

Specifies a specific fade-in time for music cross fading. Only meaningful if `specifyCrossFadeInAndOutSeperately` is enabled.

**Namespace:** (Default Namespace)  
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

C# Copy

```csharp
public float musicCrossFadeTime_In { get; set; }
```

**Property Value**

Type: Single

### See Also

- Reference
  - [AudioController Class](#)
  - (Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to [ClockStone Support Email](#)
AudioController.musicCrossFadeTime_Out Property

Specifies a specific fade-out time for music cross fading. Only meaningful if `specifyCrossFadeInAndOutSeperately` is enabled.

**Namespace:** (Default Namespace)  
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public float musicCrossFadeTime_Out { get; set; }
```

**Property Value**  
Type: Single

### See Also

**Reference**  
AudioController Class  
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH  
Send comments on this topic to ClockStone Support Email
AudioController.musicEnabled Property

Gets or sets the musicEnabled.

**Namespace:** (Default Namespace)  
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public bool musicEnabled { get; set; }
```

### Property Value

**Type:** Boolean  
true enables music; false disables music

### See Also

Reference  
AudioController Class  
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH  
Send comments on this topic to ClockStone Support Email
AudioControllersoundMuted Property

Gets or sets the soundMuted.

**Namespace:** (Default Namespace)  
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

⚠️ Syntax

```csharp
public bool soundMuted { get; set; }
```

Property Value

Type: Boolean  
true enables sound mute; false disables sound mute

⚠️ Remarks

'Sound' means all audio except music nd ambience sound

⚠️ See Also

Reference  
AudioController Class  
(Default Namespace) Namespace

---

Copyright (c) 2012 by ClockStone Software GmbH  
Send comments on this topic to ClockStone Support Email
AudioControllersystemDeltaTime Property

Returns the high precision audio system delta time since the last frame update.

**Namespace:** (Default Namespace)

**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public static double systemDeltaTime { get; }
```

**Property Value**

Type: **Double**

### See Also

Reference

AudioController Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioControllersystemTime Property

Returns the high precision audio system time size the application launch.

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```csharp
public static double systemTime { get; }
```

Property Value
Type: Double

Remarks

The audio system time does not increase if the application is paused. For performance reasons the time only gets updated with the frame rate. However, the time value does not lose precision even if the application is running for a long time (unlike Unity's 32bit float Time.systemTime

See Also

Reference
AudioController Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioController Volume Property

The global volume applied to all categories. You change the volume by script and the change will be apply to all playing audios immediately.

**Namespace:** (Default Namespace)  
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public float Volume {
    get;
    set;
}
```

### Property Value

Type: **Single**

### See Also

Reference
- AudioController Class  
- (Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
# AudioController Methods

The `AudioController` type exposes the following members.

## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![ ] AddPlaylist</td>
<td>Adds a new playlist.</td>
</tr>
<tr>
<td>![ ] AddToCategory(AudioCategory, AudioItem)</td>
<td>Adds a custom audio item to a category.</td>
</tr>
<tr>
<td>![ ] AddToCategory(AudioCategory, AudioClip, String)</td>
<td>Creates an <code>AudioItem</code> with the name <code>audioID</code> containing a single subitem playing the specified custom <code>AudioClip</code>. This <code>AudioItem</code> is then added to the specified category.</td>
</tr>
<tr>
<td>![ ] ClearPlaylists</td>
<td>Clears all music playlist.</td>
</tr>
<tr>
<td>![ ] DetachAllAudios</td>
<td>Detaches all audio objects possibly parented to the specified game object.</td>
</tr>
<tr>
<td>![ ] EnableAmbienceSound</td>
<td>Enables the ambience sound.</td>
</tr>
<tr>
<td>![ ] EnableMusic</td>
<td>Enables the music.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>EnqueueMusic</td>
<td>Enqueues an audio ID to the music playlist queue.</td>
</tr>
<tr>
<td>FadeInCategory</td>
<td>Starts a fade-in of an audio category.</td>
</tr>
<tr>
<td>FadeOutCategory</td>
<td>Starts a fade-out of an audio category.</td>
</tr>
<tr>
<td>GetAudioItem</td>
<td>Returns the AudioItem with the given audioID.</td>
</tr>
<tr>
<td>GetAudioItemMaxDistance</td>
<td>Gets the audio item's max distance. (respects all proper default values and overwrites).</td>
</tr>
<tr>
<td>GetCategory</td>
<td>Gets a category.</td>
</tr>
<tr>
<td>GetCategoryVolume</td>
<td>Gets the category volume.</td>
</tr>
<tr>
<td>GetCurrentAmbienceSound</td>
<td>Gets the current ambience sound.</td>
</tr>
<tr>
<td>GetCurrentAudioListener</td>
<td>Gets the currently active Unity audio listener.</td>
</tr>
<tr>
<td>GetCurrentMusic</td>
<td>Gets the current music.</td>
</tr>
<tr>
<td>GetGlobalVolume</td>
<td>Gets the global volume.</td>
</tr>
<tr>
<td>GetMusicPlaylist</td>
<td>If playlistName == null then gets a copy of the</td>
</tr>
<tr>
<td>Method Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>GetPlayingAudioObjects(Boolean)</td>
<td>Returns an array of all playing audio objects.</td>
</tr>
<tr>
<td>GetPlayingAudioObjects(String, Boolean)</td>
<td>Returns an array of all playing audio objects with the specified audioID.</td>
</tr>
<tr>
<td>GetPlayingAudioObjectsCount</td>
<td>Returns the number of all playing audio objects with the specified audioID.</td>
</tr>
<tr>
<td>GetPlayingAudioObjectsInCategory</td>
<td>Returns an array of all playing audio objects in the category with name.</td>
</tr>
<tr>
<td>GetPlaylistByName</td>
<td>Retrieves a playlist by name. If playlists are named identically it will return the first one it finds.</td>
</tr>
<tr>
<td>InitializeAudioItems</td>
<td>Updates the internal audioID dictionary and initializes all registered AudioItem objects.</td>
</tr>
<tr>
<td>IsAmbienceSoundEnabled</td>
<td>Determines whether ambience sound is</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>IsAmbienceSoundPaused</td>
<td>Uses to test if ambience sound is paused</td>
</tr>
<tr>
<td>IsMusicEnabled</td>
<td>Determines whether music is enabled.</td>
</tr>
<tr>
<td>IsMusicPaused</td>
<td>Uses to test if music is paused</td>
</tr>
<tr>
<td>IsPlaying</td>
<td>Determines whether the specified audio ID is playing.</td>
</tr>
<tr>
<td>IsPlaylistPlaying</td>
<td>Determines whether the playlist is playing.</td>
</tr>
<tr>
<td>IsSoundMuted</td>
<td>Determines whether sound is muted</td>
</tr>
<tr>
<td>IsValidAudioID</td>
<td>Tests if a given audioID is valid.</td>
</tr>
<tr>
<td>MuteSound</td>
<td>Mutes / Unmutes the sound.</td>
</tr>
<tr>
<td>NewCategory</td>
<td>Creates a new audio category</td>
</tr>
<tr>
<td>OnAfterDeserialize</td>
<td></td>
</tr>
<tr>
<td>OnBeforeSerialize</td>
<td></td>
</tr>
<tr>
<td>PauseAll</td>
<td>Pauses all playing audio items (including the music).</td>
</tr>
</tbody>
</table>
| PauseAmbienceSound             | Pauses the currently
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PauseCategory</strong></td>
<td>Pauses all playing audio items in the specified category (including the music).</td>
</tr>
<tr>
<td><strong>PauseMusic</strong></td>
<td>Pauses the currently playing music.</td>
</tr>
<tr>
<td><strong>Play(String)</strong></td>
<td>Plays an audio item with the name <code>audioID</code>.</td>
</tr>
<tr>
<td><strong>Play(String, Transform)</strong></td>
<td>Plays an audio item with the name <code>audioID</code> parented to a specified transform.</td>
</tr>
<tr>
<td><strong>Play(String, Vector3, Transform)</strong></td>
<td>Plays an audio item with the name <code>audioID</code> parented to a specified transform with a world offset.</td>
</tr>
<tr>
<td><strong>Play(String, Single, Single, Single)</strong></td>
<td>Plays an audio item with the name <code>audioID</code>.</td>
</tr>
<tr>
<td><strong>Play(String, Transform, Single, Single, Single)</strong></td>
<td>Plays an audio item with the name <code>audioID</code> parented to a specified transform.</td>
</tr>
<tr>
<td><strong>Play(String, Vector3, Transform, Single, Single, Single)</strong></td>
<td>Plays an audio item with the name <code>audioID</code> parented to a specified transform.</td>
</tr>
<tr>
<td>Symbol</td>
<td>Function Name</td>
</tr>
<tr>
<td>--------</td>
<td>---------------</td>
</tr>
<tr>
<td>🎧</td>
<td>PlayAfter</td>
</tr>
<tr>
<td>🎧</td>
<td>PlayAmbienceSound(String, Single, Single, Single)</td>
</tr>
<tr>
<td>🎧</td>
<td>PlayAmbienceSound(String, Transform, Single, Single, Single)</td>
</tr>
<tr>
<td>🎧</td>
<td>PlayAmbienceSound(String, Vector3, Transform, Single, Single, Single)</td>
</tr>
<tr>
<td>🎧</td>
<td>PlayAudioItem</td>
</tr>
<tr>
<td>🎧</td>
<td>PlayAudioSubItem</td>
</tr>
<tr>
<td>🎧</td>
<td>PlayMusic(String, Single, Single)</td>
</tr>
<tr>
<td>🎧</td>
<td>PlayMusic(String, Transform,</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>PlayMusic(String, Vector3, Transform, Single, Single, Single)</td>
<td>Plays an audio item with the name <code>audioID</code> as music at the specified position.</td>
</tr>
<tr>
<td>PlayMusicPlaylist</td>
<td>Start playing the music playlist.</td>
</tr>
<tr>
<td>PlayNextMusicOnPlaylist</td>
<td>Jumps to the next music track on the playlist.</td>
</tr>
<tr>
<td>PlayPreviousMusicOnPlaylist</td>
<td>Jumps to the previous music track on the playlist.</td>
</tr>
<tr>
<td>PlayScheduled</td>
<td>Plays an audio item with the name <code>audioID</code> parented to a specified transform with a world offset scheduled at a specified high precision DSP time (see the Unity AudioSettings.dspTime documentation).</td>
</tr>
<tr>
<td>RemoveAudioItem</td>
<td>Removes an AudioItem from the AudioController.</td>
</tr>
<tr>
<td>RemoveCategory</td>
<td>Removes an audio category.</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>SetCategoryVolume</strong></td>
<td>Changes the category volume. Also effects currently playing audio items.</td>
</tr>
<tr>
<td><strong>SetCurrentMusicPlaylist</strong></td>
<td>Sets the current playlist to the specified audioID array</td>
</tr>
<tr>
<td><strong>SetGlobalVolume</strong></td>
<td>Changes the global volume. Effects all currently playing audio items.</td>
</tr>
<tr>
<td><strong>Stop(String)</strong></td>
<td>Stops all playing audio items with name audioID.</td>
</tr>
<tr>
<td><strong>Stop(String, Single)</strong></td>
<td>Stops all playing audio items with name audioID with a fade-out.</td>
</tr>
<tr>
<td><strong>StopAll</strong></td>
<td>Immediately stops playing audio items (including the music).</td>
</tr>
<tr>
<td><strong>StopAll(Single)</strong></td>
<td>Fades out all playing audio items (including the music).</td>
</tr>
<tr>
<td><strong>StopAmbienceSound</strong></td>
<td>Stops the currently playing ambience sound.</td>
</tr>
<tr>
<td><strong>StopAmbienceSound(Single)</strong></td>
<td>Stops the currently playing ambience sound with fade-out.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>StopCategory</td>
<td>Stops all playing audio items in the specified category (including the music).</td>
</tr>
<tr>
<td>StopMusic</td>
<td>Stops the currently playing music.</td>
</tr>
<tr>
<td>StopMusic(Single)</td>
<td>Stops the currently playing music with fade-out.</td>
</tr>
<tr>
<td>UnloadAllAudioClips</td>
<td>Unloads all AudioClips specified in this AudioController from memory.</td>
</tr>
<tr>
<td>UnpauseAll</td>
<td>Un-pauses all playing audio items (including the music).</td>
</tr>
<tr>
<td>UnpauseAmbienceSound</td>
<td>Unpauses the current ambience sound.</td>
</tr>
<tr>
<td>UnpauseCategory</td>
<td>Un-pauses all playing audio items in the specified category (including the music).</td>
</tr>
<tr>
<td>UnpauseMusic</td>
<td>Unpauses the current music.</td>
</tr>
</tbody>
</table>

See Also

Reference
AudioController Class
(Default Namespace) Namespace
AudioControllerAddPlaylist Method

Adds a new playlist.

**Namespace:** (Default Namespace)

**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public static void AddPlaylist(
    string playlistName,
    string[] audioItemIDs
)
```

### Parameters

**playlistName**
- Type: `System.String`
- The name of the playlist to be added

**audioItemIDs**
- Type: `System.String`
- A list of audio item IDs that will represent the playlist

### See Also

**Reference**

AudioController Class

(Default Namespace) Namespace

---

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioController AddToCategory Method

### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddToCategory(AudioCategory, AudioItem)</td>
<td>Adds a custom audio item to a category.</td>
</tr>
<tr>
<td>AddToCategory(AudioCategory, AudioClip, String)</td>
<td>Creates an AudioItem with the name <code>audioID</code> containing a single subitem playing the specified custom AudioClip. This AudioItem is then added to the specified category.</td>
</tr>
</tbody>
</table>

**See Also**

Reference

AudioController Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to *ClockStone Support Email*
AudioController

AddToCategory Method (AudioCategory, AudioItem)

Adds a custom audio item to a category.

**Namespace:** (Default Namespace)

**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public static void AddToCategory(
    AudioCategory category,
    AudioItem audioItem
)
```

### Parameters

- **category**
  - Type: (Default Namespace)AudioCategory
  - The category.

- **audioItem**
  - Type: (Default Namespace)AudioItem
  - The audio item.

### Examples

```csharp
var audioItem = new AudioItem();
audioItem.SubItemPickMode = AudioPickSubItemMode.Sequence;

audioItem.subItems = new AudioSubItem[2];
```
audioItem.subItems[ 0 ] = new AudioSubItem();
audioItem.subItems[ 0 ].Clip = audioClip0;
audioItem.subItems[ 0 ].Volume = 0.7f;

audioItem.subItems[ 1 ] = new AudioSubItem();
audioItem.subItems[ 1 ].Clip = audioClip1;
audioItem.subItems[ 1 ].Volume = 0.8f;

AddToCategory( GetCategory( "CustomSFX" ), audioItem );

See Also

Reference
AudioController Class
AddToCategory Overload
(Default Namespace) Namespace
AudioControllerNewCategory(String)
AudioControllerGetCategory(String)
AudioController

AddToCategory Method (AudioCategory, AudioClip, String)

Creates an AudioItem with the name audioID containing a single subitem playing the specified custom AudioClip. This AudioItem is then added to the specified category.

**Namespace:** (Default Namespace)

**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```c#
public static AudioItem AddToCategory(
    AudioCategory category,
    AudioClip audioClip,
    string audioID
)
```

### Parameters

- **category**
  - Type: (Default Namespace)AudioCategory
  - The category.

- **audioClip**
  - Type: AudioClip
  - The custom audio clip.

- **audioID**
  - Type: System.String
  - The audioID for the AudioItem to create.

### Return Value
Type: **AudioItem**
The **AudioItem** created with the specified **audioID**

See Also

Reference

AudioController Class
AddToCategory Overload
(Default Namespace) Namespace
AudioControllerNewCategory(String)
AudioControllerGetCategory(String)

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioController.ClearPlaylists Method

Clears all music playlist.

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public static void ClearPlaylists()
```

### See Also

Reference

AudioController Class
((Default Namespace) Namespace)

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioControllerDetachAllAudios Method

Detaches all audio objects possibly parented to the specified game object.

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public static void DetachAllAudios(
    GameObject gameObjectWithAudios
)
```

### Parameters

- **gameObjectWithAudios**
  - Type: **GameObject**
  - The GameObject with possibly playing AudioObjects.

### Remarks

Use this method on a game object BEFORE destroying it if you want to keep any audios playing parented to this object.

### See Also

**Reference**

AudioController Class
(Default Namespace) Namespace

---

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioControllerEnableAmbienceSound Method

Enables the ambience sound.

**Namespace:**  (Default Namespace)
**Assembly:**  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

**Syntax**

```csharp
public static void EnableAmbienceSound(
    bool b
)
```

**Parameters**

\(b\)

Type: **SystemBoolean**
if set to **true** [b].

**See Also**

Reference
AudioController Class
(Default Namespace) Namespace
AudioControllerEnableMusic Method

Enables the music.

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

▲ Syntax

```
public static void EnableMusic(
    bool b
)
```

Parameters

*b*
- Type: SystemBoolean
- if set to true [b].

▲ See Also

Reference
- AudioController Class
- (Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioControllerEnqueueMusic Method

Enqueues an audio ID to the music playlist queue.

**Namespace:**  (Default Namespace)  
**Assembly:**  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

**Syntax**

```csharp
public static int EnqueueMusic(
    string audioID
)
```

**Parameters**

- **audioID**  
  Type: SystemString  
  The audio ID.

**Return Value**

- Type: Int32  
  The number of music tracks on the playlist.

**See Also**

- Reference  
  AudioController Class  
  (Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioControllerFadeInCategory Method

Starts a fade-in of an audio category.

**Namespace:** (Default Namespace)  
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public static void FadeInCategory(
    string name,
    float fadeInTime,
    bool stopCurrentFadeOut = true
)
```

#### Parameters

- **name**  
  Type: `System.String`  
  The category name.
- **fadeInTime**  
  Type: `System.Single`  
  The fade time in seconds.
- **stopCurrentFadeOut** *(Optional)*  
  Type: `System.Boolean`  
  In case of an existing fade-out this parameter determines if the fade-out is stopped.

### See Also

- Reference  
  AudioController Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioControllerFadeOutCategory Method

Starts a fade-out of an audio category.

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

## Syntax

```csharp
public static void FadeOutCategory(
    string name,
    float fadeOutLength,
    float startToFadeTime = 0f
)
```

### Parameters

- **name**
  - Type: `SystemString`
  - The category name.

- **fadeOutLength**
  - Type: `SystemSingle`
  - The fade time in seconds. If a negative value is specified, the fade out as specified in the corresponding `FadeOut` is used.

- **startToFadeTime (Optional)**
  - Type: `SystemSingle`
  - Fade out starts after `startToFadeTime` seconds have passed.

### Remarks

If the category is already fading out the requested fade-out is combined with the existing one.
AudioController.GetAudioItem Method

Returns the AudioItem with the given audioID.

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```csharp
public static AudioItem GetAudioItem(
    string audioID
)
```

Parameters

audioID
Type: System.String
The audioID

Return Value
Type: AudioItem
The AudioItem if audioID is valid, else null

See Also

Reference
AudioController Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioController.GetAudioItemMaxDistance

Method

Gets the audio item's max distance. (respects all proper default values and overwrites).

**Namespace:** (Default Namespace)

**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public static float GetAudioItemMaxDistance(
    string audioID
)
```

### Parameters

**audioID**
- Type: System.String
- The audioID

### Return Value

- Type: Single
- The max distance applied to the AudioSource

### See Also

Reference
AudioController Class
(Default Namespace) Namespace
Send comments on this topic to ClockStone Support Email
AudioController

GetCategory Method

Gets a category.

**Namespace:**  (Default Namespace)

**Assembly:**  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

⚠️ Syntax

```c#
public static AudioCategory GetCategory(
    string name
)
```

Parameters

*name*

Type: System.String

The category's name.

Return Value

Type: AudioCategory

The category or null if no category with the specified name exists

⚠️ See Also

Reference

AudioController Class

(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioController

GetCategoryVolume Method

Gets the category volume.

**Namespace:** (Default Namespace)

**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public static float GetCategoryVolume(
    string name
)
```

### Parameters

**name**

Type: System.String

The category name.

### Return Value

Type: Single

The volume of the specified category

### See Also

Reference

AudioController Class

(Default Namespace) Namespace
AudioControllerGetCurrentAmbienceSound Method

Gets the current ambience sound.

Namespace:  (Default Namespace)
Assembly:  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```csharp
public static AudioObject GetCurrentAmbienceSound()
```

Return Value
Type: AudioObject
Returns a reference to the AudioObject that is currently playing the ambience sound.

See Also

Reference
AudioController Class
(Default Namespace) Namespace
AudioControllerGetCurrentAudioListener Method

Gets the currently active Unity audio listener.

**Namespace:** *(Default Namespace)*  
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public static AudioListener GetCurrentAudioListener
```

**Return Value**  
Type: **AudioListener**  
Reference of the currently active AudioListener object.

### See Also

Reference  
**AudioController Class**  
*(Default Namespace)* Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioControllerGetCurrentMusic Method

Gets the current music.

Namespace:  (Default Namespace)
Assembly:  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```csharp
public static AudioObject GetCurrentMusic()
```

Return Value

Type:  AudioObject
Returns a reference to the AudioObject that is currently playing the music.

See Also

Reference
AudioController Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioControllerGetGlobalVolume Method

Gets the global volume.

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public static float GetGlobalVolume()
```

**Return Value**

Type: Single
The global volume (between 0 and 1).

### See Also

Reference
AudioController Class
(Default Namespace) Namespace
AudioController.GetMusicPlaylist Method

If playlistName == null then gets a copy of the current playlist audioID array, otherwise gets a copy of the playlist with the specified name.

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```
C#

    public static string[] GetMusicPlaylist(
        string playlistName = null
    )
```

### Parameters

**playlistName (Optional)**
Type: System.String

[Missing <param name="playlistName"/> documentation for "M:AudioController.GetMusicPlaylist(System.String)"

### Return Value
Type: String
The playlist audio item ID array

### See Also

Reference
AudioController Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioControllerGetPlayingAudioObject Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetPlayingAudioObjects(Boolean)</td>
<td>Returns an array of all playing audio objects.</td>
</tr>
<tr>
<td>GetPlayingAudioObjects(String, Boolean)</td>
<td>Returns an array of all playing audio objects with the specified audioID.</td>
</tr>
</tbody>
</table>

See Also

Reference
AudioController Class
(Default Namespace) Namespace
AudioController.GetPlayingAudioObjects Method (Boolean)

Returns an array of all playing audio objects.

**Namespace:**  (Default Namespace)
**Assembly:**  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public static List<AudioObject> GetPlayingAudioObjects(bool includePausedAudio = false)
```

#### Parameters

*includePausedAudio* *(Optional)*

Type: *System.Boolean*

If enabled the returned array will also contain paused audios.

#### Return Value

Type: *List<AudioObject>*

Array of all playing audio objects.

### See Also

Reference

AudioController Class
GetPlayingAudioObjects Overload
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioController.GetPlayingAudioObjects Method (String, Boolean)

Returns an array of all playing audio objects with the specified audioID.

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public static List<AudioObject> GetPlayingAudioObjects(string audioID,
bool includePausedAudio = false
)
```

### Parameters

- **audioID**
  - Type: System.String
  - The audio ID.
- **includePausedAudio** *(Optional)*
  - Type: System.Boolean
  - If enabled the returned array will also contain paused audios.

### Return Value

- Type: List<AudioObject>
  - Array of all playing audio objects with the specified audioID.

### See Also

Reference
AudioController Class
GetPlayingAudioObjects Overload
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioController.GetPlayingAudioObjectsCount Method

Returns the number of all playing audio objects with the specified audioID.

**Namespace:** (Default Namespace)  
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public static int GetPlayingAudioObjectsCount(
    string audioID,
    bool includePausedAudio = false
)
```

### Parameters

- **audioID**
  - Type: `System.String`  
  - The audio ID.

- **includePausedAudio (Optional)**
  - Type: `System.Boolean`  
  - If enabled the returned array will also contain paused audios.

### Return Value

- Type: `Int32`  
  - Number of all playing audio objects with the specified audioID.

### See Also

- Reference
  - AudioController Class
AudioController.GetPlayingAudioObjectsInCategory Method

Returns an array of all playing audio objects in the category with name categoryName.

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

**Syntax**

```csharp
public static List<AudioObject> GetPlayingAudioObjectsInCategory(string categoryName, bool includePausedAudio = false)
```

**Parameters**

- **categoryName**
  - Type: System.String
  - The category name.

- **includePausedAudio** *(Optional)*
  - Type: System.Boolean
  - If enabled the returned array will also contain paused audios.

**Return Value**

Type: List<AudioObject>
Array of all playing audio objects belonging to the specified category or one of its child categories.

**See Also**
Reference
AudioController Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioControllerGetPlaylistByName Method

Retrieves a playlist by name. If playlists are named identically it will return the first one it finds

Namespace:  (Default Namespace)
Assembly:  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```csharp
public Playlist GetPlaylistByName(
    string playlistName
)
```

Parameters

`playlistName`
Type: SystemString
The playlist's name

Return Value
Type: Playlist
A playlist with the specified name, otherwise null

See Also

Reference
AudioController Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioControllerInitializeAudioItems Method

Updates the internal audioID dictionary and initializes all registered AudioItem objects.

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```c#
public void InitializeAudioItems()
```

Remarks

There is no need to call this function manually, unless AudioItem objects or categories are changed at runtime.

See Also

Reference

AudioController Class
(Default Namespace) Namespace
AudioController.IsAmbienceSoundEnabled Method

Determines whether ambience sound is enabled.

**Namespace:**  (Default Namespace)  
**Assembly:**  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public static bool IsAmbienceSoundEnabled()
```

### Return Value

Type: **Boolean**  
true if ambience sound is enabled; otherwise, **false**.

### See Also

**Reference**  
AudioController Class  
(Default Namespace) Namespace
AudioController.IsAmbienceSoundPaused Method

Uses to test if ambience sound is paused

**Namespace:**  (Default Namespace)
**Assembly:**  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

▶ Syntax

```
public static bool IsAmbienceSoundPaused()
```

Return Value
Type: **Boolean**
true if ambience sound is paused, otherwise false

▶ See Also

Reference
AudioController Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioController.IsMusicEnabled Method

Determines whether music is enabled.

**Namespace:**  (Default Namespace)  
**Assembly:**  AudioToolkit (in AudioToolkit.dll)  Version: 8.2.0.0 (8.2.0.0)

⚠️ Syntax

```c#
public static bool IsMusicEnabled()
```

**Return Value**

Type: **Boolean**  
true if music is enabled; otherwise, false.

⚠️ See Also

Reference  
AudioController Class  
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH  
Send comments on this topic to ClockStone Support Email
AudioController.IsMusicPaused Method

Uses to test if music is paused

**Namespace:**  (Default Namespace)

**Assembly:**  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```c#
public static bool IsMusicPaused()
```

### Return Value

- **Type:** Boolean
- **true** if music is paused, otherwise **false**

### See Also

- **Reference**
  - AudioController Class
  - (Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioController.IsPlaying Method

Determines whether the specified audio ID is playing.

**Namespace:** (Default Namespace)

**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public static bool IsPlaying(
    string audioID
)
```

### Parameters

**audioID**
- Type: System.String
- The audio ID.

### Return Value

Type: Boolean
- `true` if the specified audio ID is playing; otherwise, `false`.

### See Also

**Reference**
- AudioController Class
- (Default Namespace) Namespace

---

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioController.IsPlaylistPlaying Method

Determines whether the playlist is playing.

**Namespace:**  (Default Namespace)  
**Assembly:**  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public static bool IsPlaylistPlaying()
```

### Return Value

Type: **Boolean**

- **true** if the current music track is from the playlist; otherwise, **false**.

### See Also

**Reference**

- AudioController Class  
  (Default Namespace) Namespace

---

Copyright (c) 2012 by ClockStone Software GmbH  
Send comments on this topic to [ClockStone Support Email](mailto:support@clockstone.com)
AudioController.IsSoundMuted Method

Determines whether sound is muted

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public static bool IsSoundMuted()
```

### Return Value

Type: **Boolean**
- `true` if sound is muted; otherwise, `false`.

### Remarks

'Sound' means all audio except music and ambience sound.

### See Also

Reference
AudioController Class
( Default Namespace ) Namespace
AudioControllerIsValidAudioID Method

Tests if a given audioID is valid.

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```csharp
public static bool IsValidAudioID(
    string audioID
)
```

Parameters

audioID
Type: System.String
The audioID

Return Value
Type: Boolean
true if the audioID is valid

See Also

Reference
AudioController Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioControllerMuteSound Method

Mutes / Unmutes the sound.

**Namespace:** (Default Namespace)  
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

## Syntax

```csharp
public static void MuteSound(
    bool b
)
```

### Parameters

- **b**  
  Type: **SystemBoolean**  
  if set to **true** [b].

## Remarks

'Sound' means all audio except music and ambience sound.

## See Also

Reference  
AudioController Class  
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH  
Send comments on this topic to ClockStone Support Email
AudioControllerNewCategory Method

Creates a new audio category

**Namespace:** (Default Namespace)  
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public static AudioCategory NewCategory(
    string categoryName
)
```

### Parameters

- `categoryName`  
  - Type: `System.String`  
  - Name of the category.

### Return Value

- Type: `AudioCategory`  
  - Reference to the new category.

### See Also

- Reference  
  - AudioController Class  
  - (Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioControllerOnAfterDeserialize Method

[Missing <summary> documentation for "M:AudioController.OnAfterDeserialize"]

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```csharp
public void OnAfterDeserialize()
```

Implements

`ISerializationCallbackReceiver.OnAfterDeserialize`

See Also

Reference

AudioController Class
(Default Namespace) Namespace
AudioControllerOnBeforeSerialize Method

[Missing <summary> documentation for "M:AudioController.OnBeforeSerialize"]

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```csharp
public void OnBeforeSerialize()
```

Implements
ISerializationCallbackReceiver.OnBeforeSerialize

See Also

Reference
AudioController Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioControllerPauseAll Method

Pauses all playing audio items (including the music).

**Namespace:** (Default Namespace)

**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```c#
public static void PauseAll(
    float fadeOutLength = 0f
)
```

### Parameters

- **fadeOutLength (Optional)**
  - Type: System.Single
  - The fade-out time [Default=0]

### See Also

- Reference
  - AudioController Class
  - (Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioControllerPauseAmbienceSound Method

Pauses the currently playing ambience sound.

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
c# Copy

public static bool PauseAmbienceSound(
    float fadeOut = 0f
)
```

### Parameters

`fadeOut (Optional)`
Type: System.Single
The fade-out time in seconds.

### Return Value
Type: Boolean
`true` if any ambience sound was paused, otherwise `false`

### See Also

Reference
AudioController Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioControllerPauseCategory Method

Pauses all playing audio items in the specified category (including the music).

**Namespace:** (Default Namespace)  
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public static void PauseCategory(
    string categoryName,
    float fadeOutLength = 0f
)
```

### Parameters

- `categoryName`  
  Type: SystemString  
  Name of category.
- `fadeOutLength` *(Optional)*  
  Type: SystemSingle  
  The fade-out time [Default=0]

### See Also

Reference  
AudioController Class  
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioControllerPauseMusic Method

Pauses the currently playing music.

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```c#
public static bool PauseMusic(
    float fadeOut = 0f
)
```

### Parameters

**fadeOut** *(Optional)*
- Type: `System.Single`
- The fade-out time in seconds.

### Return Value
- Type: `Boolean`
  - `true` if any music was paused, otherwise `false`

### See Also

Reference
- AudioController Class
- (Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
## AudioControllerPlay Method

### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Play(String)</td>
<td>Plays an audio item with the name \texttt{audioID}.</td>
</tr>
<tr>
<td>Play(String, Transform)</td>
<td>Plays an audio item with the name \texttt{audioID} parented to a specified transform.</td>
</tr>
<tr>
<td>Play(String, Vector3, Transform)</td>
<td>Plays an audio item with the name \texttt{audioID} parented to a specified transform with a world offset.</td>
</tr>
<tr>
<td>Play(String, Single, Single, Single)</td>
<td>Plays an audio item with the name \texttt{audioID}.</td>
</tr>
<tr>
<td>Play(String, Transform, Single, Single, Single)</td>
<td>Plays an audio item with the name \texttt{audioID} parented to a specified transform.</td>
</tr>
<tr>
<td>Play(String, Vector3, Transform, Single, Single, Single)</td>
<td>Plays an audio item with the name \texttt{audioID} parented to a specified transform with a world offset.</td>
</tr>
</tbody>
</table>
See Also

Reference

AudioController Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioController Play Method (String)

Plays an audio item with the name `audioID`.

**Namespace:** (Default Namespace)  
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public static AudioObject Play(string audioID)
```

### Parameters

`audioID`  
Type: `System.String`  
The audio ID.

### Return Value

Type: `AudioObject`  
Returns the reference of the `AudioObject` that is used to play the audio item, or `null` if the `audioID` does not exist. Warning: Use `PoolableReferenceT` to store an `AudioObject` reference if you have pooling enabled.

### Remarks

If "3D sound" is enabled in the audio import settings of the audio clip the object will be placed right in front of the current audio listener which is usually on the main camera. Note that the audio object will not be parented - so you will hear when the audio listener moves.
AudioController Play Method (String, Transform)

Plays an audio item with the name `audioID` parented to a specified transform.

**Namespace:** (Default Namespace)  
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public static AudioObject Play(
    string audioID,
    Transform parentObj
)
```

### Parameters

**audioID**
- Type: `System.String`  
The audio ID.

**parentObj**
- Type: `Transform`  
The parent transform.

### Return Value

Type: `AudioObject`  
Returns the reference of the `AudioObject` that is used to play the audio item, or `null` if the audioID does not exist.

### Remarks

If the audio clip is marked as 3D the audio clip will be played at the
position of the parent transform. As the audio object will get attached to the transform, it is important to destroy the parent object using the Destroy(GameObject) function, even if the parent object is not poolable itself.

See Also

Reference
AudioController Class
Play Overload
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioController Play Method (String, Vector3, Transform)

Plays an audio item with the name audioID parented to a specified transform with a world offset.

**Namespace:** (Default Namespace)  
**Assembly:** AudioToolkit (in AudioToolkit.dll)  
Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public static AudioObject Play(  
    string audioID,  
    Vector3 worldPosition,  
    Transform parentObj = null  
)
```

### Parameters

- **audioID**  
  Type: System.String  
  The audio ID.

- **worldPosition**  
  Type: Vector3  
  The position in world coordinates.

- **parentObj (Optional)**  
  Type: Transform  
  The parent transform [default=null].

### Return Value

Type: AudioObject  
Returns the reference of the AudioObject that is used to play the audio item, or null if the audioID does not exist.
Remarks

If the audio clip is marked as 3D the audio clip will be played at the position of the parent transform. As the audio object will get attached to the transform, it is important to destroy the parent object using the `Destroy(GameObject)` function, even if the parent object is not poolable itself.

See Also

Reference

AudioController Class
Play Overload
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioController Play Method
(String, Single, Single, Single)

Plays an audio item with the name audioID.

**Namespace:**  (Default Namespace)
**Assembly:**  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```c#
public static AudioObject Play(
    string audioID,
    float volume,
    float delay = 0f,
    float startTime = 0f
)
```

### Parameters

- **audioID**
  - Type: System.String
  - The audio ID.

- **volume**
  - Type: System.Single
  - The volume between 0 and 1 [default=1].

- **delay (Optional)**
  - Type: System.Single
  - The delay [default=0].

- **startTime (Optional)**
  - Type: System.Single
  - The start time [default=0]

### Return Value
Type: AudioObject
Returns the reference of the AudioObject that is used to play the audio item, or null if the audioID does not exist. Warning: Use PoolableReferenceT to store an AudioObject reference if you have pooling enabled.

Remarks

If "3D sound" is enabled in the audio import settings of the audio clip the object will be placed right in front of the current audio listener which is usually on the main camera. Note that the audio object will not be parented - so you will hear when the audio listener moves.

See Also

Reference
AudioController Class
Play Overload
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioController Play Method
(String, Transform, Single, Single, Single)

Plays an audio item with the name **audioID** parented to a specified transform.

**Namespace:**  (Default Namespace)
**Assembly:**  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

**Syntax**

```csharp
public static AudioObject Play(
    string audioID,
    Transform parentObj,
    float volume,
    float delay = 0f,
    float startTime = 0f
)
```

**Parameters**

- **audioID**
  - Type: `SystemString`
  - The audio ID.
- **parentObj**
  - Type: `Transform`
  - The parent transform.
- **volume**
  - Type: `SystemSingle`
  - The volume between 0 and 1 [default=1].
- **delay** *(Optional)*
Type: **SystemSingle**
The delay [default=0].

**startTime (Optional)**
Type: **SystemSingle**
The start time [default=0]

Return Value
Type: **AudioObject**
Returns the reference of the AudioObject that is used to play the audio item, or **null** if the audioID does not exist.

Remarks
If the audio clip is marked as 3D the audio clip will be played at the position of the parent transform. As the audio object will get attached to the transform, it is important to destroy the parent object using the `Destroy(GameObject)` function, even if the parent object is not poolable itself.

See Also

Reference
- **AudioController Class**
- **Play Overload**
  - (Default Namespace) **Namespace**

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioControllerPlay Method (String, Vector3, Transform, Single, Single, Single)

Plays an audio item with the name audioID parented to a specified transform with a world offset.

**Namespace:** (Default Namespace)  
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

**Syntax**

```csharp
public static AudioObject Play(
    string audioID,
    Vector3 worldPosition,
    Transform parentObj,
    float volume,
    float delay = 0f,
    float startTime = 0f
)
```

**Parameters**

- **audioID**  
  Type: System.String  
  The audio ID.

- **worldPosition**  
  Type: Vector3  
  The position in world coordinates.

- **parentObj**  
  Type: Transform  
  The parent transform.
volume
  Type: SystemSingle
  The volume between 0 and 1 [default=1].

delay (Optional)
  Type: SystemSingle
  The delay [default=0].

startTime (Optional)
  Type: SystemSingle
  The start time [default=0]

Return Value
  Type: AudioObject
  Returns the reference of the AudioObject that is used to play the audio item, or null if the audioID does not exist.

Remarks
  If the audio clip is marked as 3D the audio clip will be played at the position of the parent transform. As the audio object will get attached to the transform, it is important to destroy the parent object using the Destroy(GameObject) function, even if the parent object is not poolable itself.

See Also

Reference
  AudioController Class
  Play Overload
  (Default Namespace) Namespace
AudioController PlayAfter Method

Plays an audio item with the name audioID right after the given AudioObject stops playing. (see the Unity AudioSettings.dspTime documentation)

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```c#
public static AudioObject PlayAfter(
    string audioID,
    AudioObject playingAudio,
    double deltaDspTime = 0,
    float volume = 1f,
    float startTime = 0f)
```

Parameters

audioID
- Type: System.String
- The audio ID.

playingAudio
- Type: (Default Namespace)AudioObject
- Playback will start after this AudioObject finished playing

deltaDspTime (Optional)
- Type: System.Double
- A time delta (high precision DSP time) at which to start playing. Negative values will cause audios to overlap.

volume (Optional)
- Type: System.Single
- The volume between 0 and 1 [default=1].
**startTime (Optional)**

Type: **SystemSingle**
The start time seconds [default=0]

**Return Value**

Type: **AudioObject**
Returns the reference of the AudioObject that is used to play the audio item, or **null** if the audioID does not exist.

**Remarks**

Uses the PlayScheduled function only available in Unity v4.1 or newer that allows to stitch two audios together at DSP level precision without a gap. Can not be used to chain more then one audio.

**See Also**

Reference

AudioController Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioController PlayAmbienceSound Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PlayAmbienceSound(String, Single, Single, Single)</td>
<td>Plays an audio item with the name <code>audioID</code> as ambience sound.</td>
</tr>
<tr>
<td>PlayAmbienceSound(String, Transform, Single, Single, Single)</td>
<td>Plays an audio item with the name <code>audioID</code> as ambience sound at the specified position.</td>
</tr>
<tr>
<td>PlayAmbienceSound(String, Vector3, Transform, Single, Single, Single)</td>
<td>Plays an audio item with the name <code>audioID</code> as ambience sound at the specified position.</td>
</tr>
</tbody>
</table>

See Also

Reference
AudioController Class (Default Namespace) Namespace
AudioControllerPlayAmbienceSound
Method (String, Single, Single, Single)

Plays an audio item with the name `audioID` as ambience sound.

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

## Syntax

```csharp
public static AudioObject PlayAmbienceSound(
    string audioID,
    float volume = 1f,
    float delay = 0f,
    float startTime = 0f
)
```

## Parameters

- **audioID**
  - Type: `System.String`
  - The audio ID.
- **volume** *(Optional)*
  - Type: `System.Single`
  - The volume between 0 and 1 [default=1].
- **delay** *(Optional)*
  - Type: `System.Single`
  - The delay [default=0].
- **startTime** *(Optional)*
  - Type: `System.Single`
  - The start time [default=0]
Return Value
Type: AudioObject
Returns the reference of the AudioObject that is used to play the audio item, or null if the audioID does not exist. Warning: Use PoolableReferenceT to store an AudioObject reference if you have pooling enabled.

Remarks
PlayAmbienceSound makes sure that only one ambience track is played at a time. If cross fading is enabled in the AudioController fading is performed automatically. The audio clip the object will be placed right in front of the current audio listener which is usually on the main camera. Note that the audio object will not be parented - so you will hear when the audio listener moves.

See Also
Reference
AudioController Class
PlayAmbienceSound Overload
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioController PlayAmbienceSound Method (String, Transform, Single, Single, Single)

Plays an audio item with the name audioID as ambience sound at the specified position.

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

**Syntax**

```csharp
public static AudioObject PlayAmbienceSound(
    string audioID,
    Transform parentObj,
    float volume = 1f,
    float delay = 0f,
    float startTime = 0f
)
```

**Parameters**

- **audioID**
  - Type: SystemString
  - The audio ID.

- **parentObj**
  - Type: Transform
  - The parent transform or null.

- **volume (Optional)**
  - Type: SystemSingle
  - The volume between 0 and 1 [default=1].

- **delay (Optional)**
Type: **SystemSingle**  
The delay [default=0].

*startTime (Optional)*  
Type: **SystemSingle**  
The start time [default=0]

Return Value  
Type: **AudioObject**  
Returns the reference of the AudioObject that is used to play the audio item, or `null` if the audioID does not exist. Warning: Use `PoolableReferenceT` to store an AudioObject reference if you have pooling enabled.

Remarks

PlayAmbienceSound makes sure that only one ambience track is played at a time. If cross fading is enabled in the AudioController fading is performed automatically.

See Also

Reference
- **AudioController Class**
- **PlayAmbienceSound Overload**
- (Default Namespace) **Namespace**

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to [ClockStone Support Email](mailto:SupportEmail@ClockStone.com)
AudioController PlayAmbienceSound

Method (String, Vector3, Transform, Single, Single, Single)

Plays an audio item with the name `audioID` as ambience sound at the specified position.

**Namespace:** (Default Namespace)

**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

**Syntax**

```c#
public static AudioObject PlayAmbienceSound(
    string audioID,
    Vector3 worldPosition,
    Transform parentObj = null,
    float volume = 1f,
    float delay = 0f,
    float startTime = 0f
)
```

**Parameters**

- **audioID**
  - Type: `System.String`
  - The audio ID.

- **worldPosition**
  - Type: `Vector3`
  - The position in world coordinates.

- **parentObj (Optional)**
  - Type: `Transform`
  - The parent transform or `null`. 
volume (Optional)
Type: SystemSingle
The volume between 0 and 1 [default=1].

delay (Optional)
Type: SystemSingle
The delay [default=0].

startTime (Optional)
Type: SystemSingle
The start time [default=0]

Return Value
Type: AudioObject
Returns the reference of the AudioObject that is used to play the audio item, or null if the audioID does not exist. Warning: Use PoolableReferenceT to store an AudioObject reference if you have pooling enabled.

Remarks
PlayAmbienceSound makes sure that only one ambience track is played at a time. If cross fading is enabled in the AudioController fading is performed automatically.

See Also

Reference
AudioController Class
PlayAmbienceSound Overload
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioController.PlayAudioItem Method

Plays a specific AudioItem.

**Namespace:** (Default Namespace)

**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

#### Syntax

```csharp
public AudioObject PlayAudioItem(
    AudioItem sndItem,
    float volume,
    Vector3 worldPosition,
    Transform parentObj = null,
    float delay = 0f,
    float startTime = 0f,
    bool playWithoutAudioObject = false,
    AudioObject useExistingAudioObj = null,
    double dspTime = 0,
    bool playAsMusicOrAmbienceSound = false,
    float startVolumeMultiplier = 1f
)
```

#### Parameters

- **sndItem**
  - Type: (Default Namespace)AudioItem
  - the AudioItem

- **volume**
  - Type: System.Single
  - the volume

- **worldPosition**
Type: **Vector3**
the world position

**parentObj (Optional)**
Type: **Transform**
the parent object, or null

**delay (Optional)**
Type: **System.Single**
the delay in seconds

**startTime (Optional)**
Type: **System.Single**
the start time seconds

**playWithoutAudioObject (Optional)**
Type: **System.Boolean**
if true plays the audio by using the Unity function **PlayOneShot** without creating an audio game object. Allows playing audios from within the Unity inspector.

**useExistingAudioObj (Optional)**
Type: **(Default Namespace)AudioObject**
if specified this existing audio object is used instead of creating a new **AudioObject**

**dspTime (Optional)**
Type: **System.Double**
The high precision DSP time at which to schedule playing the audio [default=0]

**playAsMusicOrAmbienceSound (Optional)**
Type: **System.Boolean**
Determines if it is effected by sound muting [default=false]

**startVolumeMultiplier (Optional)**
Type: **System.Single**
allows to adjust the start volume if e.g. a FadeOut will follow immediately after

Return Value
Type: **AudioObject**
The created **AudioObject** or null

Remarks
This function is used by the editor extension and is normally not required for application developers. Use `Play(String)` instead.

**See Also**

**Reference**

- `AudioController Class`
- `(Default Namespace) Namespace`

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to [ClockStone Support Email](mailto:support@clockstone.com)
AudioControllerPlayAudioSubItem

Method

Plays a specific AudioSubItem.

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```c#
public AudioObject PlayAudioSubItem(
    AudioSubItem subItem,
    float volume,
    Vector3 worldPosition,
    Transform parentObj,
    float delay,
    float startTime,
    bool playWithoutAudioObject,
    AudioObject useExistingAudioObj,
    double dspTime = 0,
    bool playAsMusicOrAmbienceSound = false,
    float startVolumeMultiplier = 1f
)
```

### Parameters

- **subItem**
  - Type: (Default Namespace)AudioSubItem
  - the AudioSubItem

- **volume**
  - Type: System.Single
  - the volume

- **worldPosition**
Type: **Vector3**
the world position

*parentObj*
Type: **Transform**
the parent object, or **null**

delay
Type: **SystemSingle**
the delay in seconds

*startTime*
Type: **SystemSingle**
the start time seconds

*playWithoutAudioObject*
Type: **SystemBoolean**
if **true** plays the audio by using the Unity function **PlayOneShot** without creating an audio game object. Allows playing audios from within the Unity inspector.

*useExistingAudioObj*
Type: (Default Namespace)**AudioObject**
if specified this existing audio object is used instead of creating a new **AudioObject**

dspTime **(Optional)**
Type: **SystemDouble**
The high precision DSP time at which to schedule playing the audio [default=0]

*playAsMusicOrAmbienceSound **(Optional)***
Type: **SystemBoolean**
if **true** plays the audio as music or ambience track [default=**false**]

*startVolumeMultiplier **(Optional)***
Type: **SystemSingle**
allows to adjust the start volume if e.g. a FadeOut will follow immediately after

Return Value
Type: **AudioObject**
The created **AudioObject** or **null**

Remarks
This function is used by the editor extension and is normally not required for application developers. Use `Play(String)` instead.

See Also

Reference

`AudioController Class` [Namespace]

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to [ClockStone Support Email](mailto:ClockStoneSupportEmail)
AudioController::PlayMusic Method

## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PlayMusic(String, Single, Single, Single)</td>
<td>Plays an audio item with the name <code>audioID</code> as music.</td>
</tr>
<tr>
<td>PlayMusic(String, Transform, Single, Single, Single)</td>
<td>Plays an audio item with the name <code>audioID</code> as music at the specified position.</td>
</tr>
<tr>
<td>PlayMusic(String, Vector3, Transform, Single, Single, Single)</td>
<td>Plays an audio item with the name <code>audioID</code> as music at the specified position.</td>
</tr>
</tbody>
</table>

See Also

Reference

AudioController Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioController.PlayMusic Method (String, Single, Single, Single)

Plays an audio item with the name audioID as music.

Namespace:  (Default Namespace)
Assembly:  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```csharp
public static AudioObject PlayMusic(
    string audioID,
    float volume = 1f,
    float delay = 0f,
    float startTime = 0f
)
```

Parameters

- **audioID**
  - Type: System.String
  - The audio ID.

- **volume (Optional)**
  - Type: System.Single
  - The volume between 0 and 1 [default=1].

- **delay (Optional)**
  - Type: System.Single
  - The delay [default=0].

- **startTime (Optional)**
  - Type: System.Single
  - The start time [default=0]
Return Value
Type: AudioObject
Returns the reference of the AudioObject that is used to play the audio item, or null if the audioID does not exist. Warning: Use PoolableReferenceT to store an AudioObject reference if you have pooling enabled.

Remarks
PlayMusic makes sure that only one music track is played at a time. If music cross fading is enabled in the AudioController fading is performed automatically. The audio clip the object will be placed right in front of the current audio listener which is usually on the main camera. Note that the audio object will not be parented - so you will hear when the audio listener moves.

See Also
Reference
AudioController Class
PlayMusic Overload
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioController

PlayMusic Method (String, Transform, Single, Single, Single)

Plays an audio item with the name audioID as music at the specified position.

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```csharp
public static AudioObject PlayMusic(
    string audioID,
    Transform parentObj,
    float volume = 1f,
    float delay = 0f,
    float startTime = 0f
)
```

Parameters

audioID
  Type: SystemString
  The audio ID.

parentObj
  Type: Transform
  The parent transform or null.

volume (Optional)
  Type: SystemSingle
  The volume between 0 and 1 [default=1].

delay (Optional)
**Type:** SystemSingle  
The delay [default=0].

**startTime (Optional)**  
Type: SystemSingle  
The start time [default=0]

---

**Return Value**  
Type: AudioObject  
Returns the reference of the AudioObject that is used to play the audio item, or null if the audioID does not exist. **Warning:** Use PoolableReference to store an AudioObject reference if you have pooling enabled.

---

**Remarks**

PlayMusic makes sure that only one music track is played at a time. If music cross fading is enabled in the AudioController fading is performed automatically.

---

**See Also**

Reference  
AudioController Class  
PlayMusic Overload  
(Default Namespace) Namespace

---

Copyright (c) 2012 by ClockStone Software GmbH  
Send comments on this topic to ClockStone Support Email
AudioControllerPlayMusic Method (String, Vector3, Transform, Single, Single, Single)

Plays an audio item with the name **audioID** as music at the specified position.

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public static AudioObject PlayMusic(
    string audioID,
    Vector3 worldPosition,
    Transform parentObj = null,
    float volume = 1f,
    float delay = 0f,
    float startTime = 0f
)
```

### Parameters

- **audioID**
  - Type: **System.String**
  - The audio ID.

- **worldPosition**
  - Type: **Vector3**
  - The position in world coordinates.

- **parentObj** (Optional)
Type: **Transform**
The parent transform or **null**.

*volume (Optional)*
Type: **SystemSingle**
The volume between 0 and 1 [default=1].

*delay (Optional)*
Type: **SystemSingle**
The delay [default=0].

*startTime (Optional)*
Type: **SystemSingle**
The start time [default=0]

Return Value
Type: **AudioObject**
Returns the reference of the AudioObject that is used to play the audio item, or **null** if the audioID does not exist. Warning: Use **PoolableReference** to store an AudioObject reference if you have pooling enabled.

Remarks
PlayMusic makes sure that only one music track is played at a time. If music cross fading is enabled in the AudioController fading is performed automatically.

See Also
*Reference*
**AudioController Class**
**PlayMusic Overload**
*(Default Namespace) Namespace*
AudioController.PlayMusicPlaylist Method

Start playing the music playlist.

**Namespace:** (Default Namespace)  
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public static AudioObject PlayMusicPlaylist(
    string playlistName = null
)
```

### Parameters

**playlistName (Optional)**
- Type: System.String
- [Missing `<param name="playlistName"/>` documentation for "M:AudioController.PlayMusicPlaylist(System.String)"]

### Return Value

- **Type:** AudioObject
- The AudioObject of the current music, or null if no music track could be played.

### See Also

**Reference**  
AudioController Class  
(Default Namespace) Namespace
Send comments on this topic to ClockStone Support Email
AudioController::PlayNextMusicOnPlaylist Method

Jumps to the next the music track on the playlist.

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public static AudioObject PlayNextMusicOnPlaylist()
```

#### Return Value

Type: **AudioObject**

The **AudioObject** of the current music, or **null** if no music track could be played.

### Remarks

If shuffling is enabled it will jump to the next randomly chosen track.

### See Also

Reference

AudioController Class
(Default Namespace) Namespace

---

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioController.PlayPreviousMusicOnPlaylist Method

Jumps to the previous music track on the playlist.

**Namespace:**  (Default Namespace)

**Assembly:**  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public static AudioObject PlayPreviousMusicOnPlaylist
```

### Return Value

**Type:**  AudioObject

The AudioObject of the current music, or null if no music track could be played.

### Remarks

If shuffling is enabled it will jump to the previously played track.

### See Also

Reference

AudioController Class

((Default Namespace) Namespace)

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioControllerPlayScheduled Method

Plays an audio item with the name `audioID` parented to a specified transform with a world offset scheduled at a specified high precision DSP time (see the Unity AudioSettings.dspTime documentation)

**Namespace:** (Default Namespace)  
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public static AudioObject PlayScheduled(
    string audioID,
    double dspTime,
    Vector3 worldPosition,
    Transform parentObj = null,
    float volume = 1f,
    float startTime = 0f
)
```

### Parameters

- **audioID**  
  Type: `System.String`  
  The audio ID.

- **dspTime**  
  Type: `System.Double`  
  The high precision DSP time at which to start playing.

- **worldPosition**  
  Type: `Vector3`  
  The position in world coordinates.

- **parentObj** *(Optional)*
Type: **Transform**
The parent transform.

**volume (Optional)**
Type: **SystemSingle**
The volume between 0 and 1 [default=1].

**startTime (Optional)**
Type: **SystemSingle**
The start time seconds [default=0]

Return Value
Type: **AudioObject**
Returns the reference of the AudioObject that is used to play the audio item, or **null** if the audioID does not exist.

---

**See Also**

Reference
**AudioController Class**
**(Default Namespace) Namespace**

---

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioControllerRemoveAudioItem Method

Removes an AudioItem from the AudioController.

**Namespace:**  (Default Namespace)
**Assembly:**  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```c#
public static bool RemoveAudioItem(
    string audioID
)
```

### Parameters

**audioID**
- Type: System.String
- Name of the audio item to remove.

### Return Value
- Type: Boolean
- true if the audio item was found and successfully removed, otherwise false

### See Also

Reference
AudioController Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioControllerRemoveCategory
Method

Removes an audio category.

**Namespace:**  (Default Namespace)
**Assembly:**  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```c#
public static void RemoveCategory(
    string categoryName
)
```

### Parameters

- **categoryName**
  - Type: System.String
  - Name of the category to remove.

### See Also

- Reference
  - AudioController Class
  - (Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioController.SetCategoryVolume Method

Changes the category volume. Also effects currently playing audio items.

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```csharp
public static void SetCategoryVolume(
    string name,
    float volume
)
```

Parameters

- **name**
  - Type: `System.String`
  - The category name.

- **volume**
  - Type: `System.Single`
  - The volume (between 0 and 1).

See Also

Reference
- AudioController Class
- (Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioController SetCurrentMusicPlaylist Method

Sets the current playlist to the specified audioID array

**Namespace:**  (Default Namespace)
**Assembly:**  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

**Syntax**

```csharp
public static bool SetCurrentMusicPlaylist(
    string playlistName
)
```

**Parameters**

*playlistName*
  Type: System.String
  The new playlist array

**Return Value**

Type: Boolean

[Missing <returns> documentation for "M:AudioController.SetCurrentMusicPlaylist(System.String)""]

**See Also**

Reference
AudioController Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
AudioControllerSetGlobalVolume Method

Changes the global volume. Effects all currently playing audio items.

**Namespace:**  (Default Namespace)
**Assembly:**  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public static void SetGlobalVolume(
    float volume
)
```

### Parameters

`volume`
- Type: `System.Single`
- The volume (between 0 and 1).

### Remarks

Volume change is also applied to all additional AudioControllers.

### See Also

Reference
- [AudioController Class (Default Namespace) Namespace](#)

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to [ClockStone Support Email](mailto:ClockStoneSupportEmail@ClockStone.com)
# AudioController Stop Method

## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stop(String)</td>
<td>Stops all playing audio items with name <code>audioID</code>.</td>
</tr>
<tr>
<td>Stop(String, Single)</td>
<td>Stops all playing audio items with name <code>audioID</code> with a fade-out.</td>
</tr>
</tbody>
</table>

## See Also

Reference

AudioController Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioControllerStop Method (String)

Stops all playing audio items with name audioID.

**Namespace:** (Default Namespace)  
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```c#
public static bool Stop(
    string audioID
)
```

### Parameters

- **audioID**  
  Type: System.String  
  The audio ID.

### Return Value

Type: Boolean  
Return true if any audio was stopped.

### See Also

- Reference
  - AudioController Class
  - Stop Overload
    (Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioController Stop Method
(String, Single)

Stops all playing audio items with name audioID with a fade-out.

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public static bool Stop(
    string audioID,
    float fadeOutLength
)
```

### Parameters

- **audioID**
  - Type: System.String
  - The audio ID.

- **fadeOutLength**
  - Type: System.Single
  - The fade out time. If a negative value is specified, the subitem's FadeOut value is taken.

### Return Value

- Type: Boolean
  - Return true if any audio was stopped.

### See Also

- Reference
  - AudioController Class
Stop Overload
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
## AudioControllerStopAll Method

### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>StopAll</td>
<td>Immediately stops playing audio items (including the music).</td>
</tr>
<tr>
<td>StopAll(Single)</td>
<td>Fades out all playing audio items (including the music).</td>
</tr>
</tbody>
</table>

### See Also

Reference

AudioController Class
(Default Namespace) Namespace
AudioControllerStopAll Method

Immediately stops playing audio items (including the music).

**Namespace:**  (Default Namespace)
**Assembly:**  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public static void StopAll()
```

### See Also

**Reference**
- AudioController Class
- StopAll Overload
- (Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioControllerStopAll Method (Single)

Fades out all playing audio items (including the music).

**Namespace:** (Default Namespace)  
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

**Syntax**

```csharp
public static void StopAll(
    float fadeOutLength
)
```

**Parameters**

*fadeOutLength*

  Type: System.Single  
  The fade out time. If a negative value is specified, the subitem's **FadeOut** value is taken.

**See Also**

Reference  
AudioController Class  
StopAll Overload  
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH  
Send comments on this topic to ClockStone Support Email
# AudioControllerStopAmbienceSound Method

## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="StopAmbienceSound" /></td>
<td>Stops the currently playing ambience sound.</td>
</tr>
<tr>
<td><img src="image" alt="StopAmbienceSound(Single)" /></td>
<td>Stops the currently playing ambience sound with fade-out.</td>
</tr>
</tbody>
</table>

Top

## See Also

Reference

AudioController Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to [ClockStone Support Email](mailto:support@clockstone.com)
**AudioController.StopAmbienceSound Method**

Stops the currently playing ambience sound.

- **Namespace:** (Default Namespace)
- **Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

**Syntax**

```csharp
public static bool StopAmbienceSound()
```

**Return Value**

Type: **Boolean**

- `true` if any ambience sound was stopped, otherwise `false`

**See Also**

- Reference
- AudioController Class
- StopAmbienceSound Overload
- (Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to [ClockStone Support Email](mailto:support@clockstone.com)
AudioController.StopAmbienceSound Method (Single)

Stops the currently playing ambience sound with fade-out.

**Namespace:**  (Default Namespace)  
**Assembly:**  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public static bool StopAmbienceSound(
    float fadeOut
)
```

### Parameters

* fadeOut  
  Type: System.Single  
  The fade-out time in seconds.

### Return Value

Type: Boolean  
true if any ambience sound was stopped, otherwise false

### See Also

Reference  
AudioController Class  
StopAmbienceSound Overload  
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioControllerStopCategory Method

Stops all playing audio items in the specified category (including the music).

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```csharp
public static void StopCategory(
    string categoryName,
    float fadeOutLength = 0f
)
```

Parameters

categoryName
  Type: System.String
  Name of category.

fadeOutLength (Optional)
  Type: System.Single
  The fade-out time [Default=0]

See Also

Reference
AudioController Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioControllerStopMusic Method

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>StopMusic</td>
<td>Stops the currently playing music.</td>
</tr>
<tr>
<td>StopMusic(Single)</td>
<td>Stops the currently playing music with fade-out.</td>
</tr>
</tbody>
</table>

See Also

Reference

AudioController Class
(Default Namespace) Namespace
AudioControllerStopMusic Method

Stops the currently playing music.

**Namespace:**  (Default Namespace)
**Assembly:**  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public static bool StopMusic()
```

**Return Value**

Type: **Boolean**

- `true` if any music was stopped, otherwise `false`

### See Also

**Reference**

- AudioController Class
- StopMusic Overload
- (Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioControllerStopMusic Method (Single)

Stops the currently playing music with fade-out.

**Namespace:** (Default Namespace)  
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public static bool StopMusic(float fadeOut)
```

### Parameters

*fadeOut*

Type: `System.Single`  
The fade-out time in seconds.

### Return Value

Type: `Boolean`  
`true` if any music was stopped, otherwise `false`

### See Also

**Reference**  
AudioController Class  
StopMusic Overload  
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioControllerUnloadAllAudioClips Method

Unloads all AudioClips specified in this AudioController from memory.

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```c#
public void UnloadAllAudioClips()
```

### Remarks

You will still be able to play the AudioClips, but you may experience performance hickups when Unity reloads the audio asset

### See Also

Reference
AudioController Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioController.UnpauseAll

Method

Un-pauses all playing audio items (including the music).

**Namespace:** (Default Namespace)

**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```c#
public static void UnpauseAll(
    float fadeInLength = 0f
)
```

Parameters

*fadeInLength (Optional)*

Type: System.Single

The fade-in time [Default=0]

See Also

Reference

AudioController Class

(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioControllerUnpauseAmbienceSound Method

Unpauses the current ambience sound.

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```
public static bool UnpauseAmbienceSound(
    float fadeIn = 0f
)
```

**Parameters**

*fadeIn* (Optional)
Type: System.Single
The fade-in time in seconds.

**Return Value**
Type: Boolean
true if any ambience sound was unpaused, otherwise false

### See Also

Reference
AudioController Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioControllerUnpauseCategory Method

Un-pauses all playing audio items in the specified category (including the music).

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

**Syntax**

C#

```csharp
public static void UnpauseCategory(
    string categoryName,
    float fadeInLength = 0f
)
```

**Parameters**

*categoryName*

Type: System.String
Name of category.

*fadeInLength (Optional)*

Type: System.Single
The fade-in time [Default=0]

**See Also**

Reference
AudioController Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioControllerUnpauseMusic Method

Unpauses the current music.

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```c#
public static bool UnpauseMusic(
    float fadeIn = 0f
)
```

### Parameters

- **fadeIn** *(Optional)*
  - Type: System.Single
  - The fade-in time in seconds.

### Return Value

- Type: Boolean
  - true if any music was unpaused, otherwise false

### See Also

- Reference
  - AudioController Class
  - (Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioController Fields

The AudioController type exposes the following members.

Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>_currentInspectorSelection</td>
<td></td>
</tr>
<tr>
<td>ambienceSoundCrossFadeTime</td>
<td>If set to a value &gt; 0, music will automatically be cross-faded with this fading time.</td>
</tr>
<tr>
<td>AUDIO_TOOLKIT_VERSION</td>
<td>A string containing the version number of the Audio Toolkit.</td>
</tr>
<tr>
<td>AudioCategories</td>
<td>Specify your audio categories here using the Unity inspector.</td>
</tr>
<tr>
<td>AudioObjectPrefab</td>
<td>You must specify your AudioObject prefab here using the Unity inspector.</td>
</tr>
<tr>
<td>crossfadePlaylist</td>
<td>If enabled, the tracks on the playlist will get cross-faded.</td>
</tr>
<tr>
<td>delayBetweenPlaylistTracks</td>
<td>Mute time in between two tracks on the playlist.</td>
</tr>
<tr>
<td>EqualPowerCrossfade</td>
<td>If enabled, fading is adjusted so that cross-fades result in equal power.</td>
</tr>
</tbody>
</table>
the same power during fading

- **loopPlaylist** specifies if the music playlist will be looped.

- **musicCrossFadeTime** specifies if the music will be cross-faded with this fading time.

- **musicPlaylist** is **Obsolete**. For backwards compatibility, we still keep the old playlist system. Its items will be copied over to the Default Playlist and then cleared.

- **musicPlaylists** allows to specify a list of named playlist that can be played as music.

- **Persistent** enables/disables the audio controller to survive scene changes.

- **PlayWithZeroVolume** If disabled, audios are not played if they have a resulting volume of zero.

- **shufflePlaylist** enables/disables shuffling for the music playlist.

- **specifyCrossFadeInAndOutSeparately** If enabled, specific music and ambience sound cross-fading in and out times can be specified.

- **UnloadAudioClipsOnDestroy** If enabled all audio resources are unloaded when destroying the object.
| UsePooledAudioObjects | Enables / Disables pooling |

---

**See Also**

Reference

AudioController Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioController_currentInspectorSelection

Field

[Missing <summary> documentation for "F:AudioController._currentInspectorSelection"]

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```csharp
public AudioController_CurrentInspectorSelection _currentInspectorSelection
```

Field Value

Type: AudioController_CurrentInspectorSelection

See Also

Reference

AudioController Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
ClockStone Audio Toolkit for Unity - Documentation
AudioController::AmbienceSoundCrossFadeTime

If set to a value > 0 (in seconds) music will automatically be cross-faded with this fading time.

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public float ambienceSoundCrossFadeTime
```

### Field Value

**Type:** Single

### Remarks

if ` specifyCrossFadeInAndOutSeperately ` is enabled, 
`ambienceSoundCrossFadeTime_In` and 
`ambienceSoundCrossFadeTime_Out` are used instead.

### See Also

Reference
AudioController Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioControllerAUDIToolkit_V
Field

A string containing the version number of the Audio Toolkit

**Namespace:**  (Default Namespace)
**Assembly:**  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

**Syntax**

C#

```c#
public const string AUDIO_TOOLKIT_VERSION = "8.2"
```

**Field Value**

Type: **String**

**See Also**

Reference

AudioController Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioController
AudioCategories Field

Specify your audio categories here using the Unity inspector.

Namespace:  (Default Namespace)
Assembly:  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```c#
public AudioCategory[] AudioCategories
```

Field Value
Type: AudioCategory

See Also

Reference
AudioController Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioControllerAudioObjectPrefab
Field

You must specify your AudioObject prefab here using the Unity inspector.
- **AudioObjectAudioSource (Unity built-in) PoolableObject** - only required if pooling is uses

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

⚠️ Syntax

```csharp
public GameObject AudioObjectPrefab
```

Field Value
Type: **GameObject**

⚠️ See Also

Reference
**AudioController Class**
( Default Namespace ) Namespace
AudioController.crossfadePlaylist Field

if enabled, the tracks on the playlist will get cross-faded as specified by musicCrossFadeTime

**Namespace:** (Default Namespace)  
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public bool crossfadePlaylist
```

### Field Value

Type: Boolean

### See Also

Reference  
AudioController Class  
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH  
Send comments on this topic to ClockStone Support Email
AudioController.delayBetweenPlaylistTracks Field

Mute time in between two tracks on the playlist.

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```c#
public float delayBetweenPlaylistTracks
```

### Field Value

Type: **Single**

### See Also

**Reference**
AudioController Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioControllerEqualPowerCrossfade Field

If enabled fading is adjusted in a way so that cross-fades should result in the same power during the time of fading

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

## Syntax

```
public bool EqualPowerCrossfade
```

### Field Value

Type: **Boolean**

## Remarks

Unfortunately not 100% correct as Unity uses unknown internal formulas for computing the volume.

## See Also

Reference

AudioController Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioController

Loop Playlist Field

specifies if the music playlist will get looped

**Namespace:** (Default Namespace)

**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```c#
public bool loopPlaylist
```

### Field Value

Type: **Boolean**

### See Also

**Reference**

AudioController Class

(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioController\texttt{musicCrossFadeTime} Field

If set to a value \(> 0\) (in seconds) music will automatically be cross-faded with this fading time.

\textbf{Namespace:} (Default Namespace)
\textbf{Assembly:} AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

\section*{Syntax}

\begin{verbatim}
public float musicCrossFadeTime
\end{verbatim}

\section*{Field Value}

Type: \texttt{Single}

\section*{Remarks}

if \texttt{specifyCrossFadeInAndOutSeperately} is enabled, \texttt{musicCrossFadeTime\_In} and \texttt{musicCrossFadeTime\_Out} are used instead.

\section*{See Also}

Reference
\texttt{AudioController Class}
((Default Namespace) Namespace)

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to \texttt{ClockStone Support Email}
AudioControllermusicPlaylist Field

Note: This API is now obsolete.
For backwards compatibility we still keep the old playlist system, its items will be copied over to the Default Playlist and then cleared

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```csharp
[ObsoleteAttribute]
public string[] musicPlaylist
```

Field Value
Type: String

See Also

Reference
AudioController Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioController musicPlaylists Field

allows to specify a list of named playlist that can be played as music

**Namespace:**  *(Default Namespace)*
**Assembly:**  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```c#
public Playlist[] musicPlaylists
```

**Field Value**
- **Type:** Playlist

### See Also

**Reference**
- AudioController Class
  *(Default Namespace) Namespace*

---

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to [ClockStone Support Email](mailto:info@clockstone.de)
AudioControllerPersistent Field

If enabled, the audio controller will survive scene changes

**Namespace:**  (Default Namespace)  
**Assembly:**  Audio Toolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```c#
public bool Persistent
```

### Field Value

**Type:** Boolean

### Remarks

For projects with a large number of audio files you may consider having separate AudioController version for each scene and only specify those audio items that are really required in this scene. This can reduce memory consumption and speed up loading time for the initial scene.

### See Also

**Reference**  
AudioController Class  
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioController PlayWithZeroVolume Field

If disabled, audios are not played if they have a resulting volume of zero.

**Namespace:** (Default Namespace)

**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public bool PlayWithZeroVolume
```

### Field Value

Type: Boolean

### See Also

Reference

AudioController Class

(Default Namespace) Namespace

---

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to [ClockStone Support Email](mailto:info@clockstone.com)
AudioControllers\shufflePlaylist\Field

enables / disables shuffling for the music playlist

**Namespace:**  \(\text{(Default Namespace)}\)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```c#
public bool shufflePlaylist
```

### Field Value

Type: Boolean

### See Also

Reference

AudioController Class  
\(\text{(Default Namespace)}\) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioControllersSpecifyCrossFadeInAndOutSeparately

Field

If enabled specific music and ambience sound cross-fading in and out times can be specified with `musicCrossFadeTime_In` and `musicCrossFadeTime_Out` respectively, `ambienceSoundCrossFadeTime_In` and `ambienceSoundCrossFadeTime_Out` respectively.

**Namespace:** (Default Namespace)

**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public bool specifyCrossFadeInAndOutSeparately
```

Field Value

Type: Boolean

### See Also

Reference

AudioController Class

(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioControllerUnloadAudioClipsOnDestroy

Field

If enabled all audio resources (AudioClips) specified in this AudioController are unloaded from memory when the AudioController gets destroyed (e.g. when loading a new scene and Persistent is not enabled)

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```
public bool UnloadAudioClipsOnDestroy
```

Field Value
Type: Boolean

Remarks

Uses Unity's `Resources.UnloadAsset(...)` method. Can be used to save memory if many audio resources are in use. It is recommended to use additional AudioControllers for audios that are used only within a specific scene, and a primary persistent AudioController for audio used throughout the entire application.

See Also

Reference
AudioController Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioControllerUsePooledAudioObjects Field

Enables / Disables AudioObject pooling

**Namespace:**  (Default Namespace)
**Assembly:**  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

⚠️ **Syntax**

```csharp
public bool UsePooledAudioObjects
```

Field Value
Type: **Boolean**

⚠️ **Remarks**

Warning: Use `PoolableReferenceT` to store an AudioObject reference if you have pooling enabled.

⚠️ **See Also**

Reference
AudioController Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudiolItem Class

The AudiolItem class represents a uniquely named audio entity that can be played by scripts.

Inheritance Hierarchy

- `SystemObject` (Default Namespace)AudiolItem

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

C#  

```csharp
[SerializableAttribute]
public class AudioItem
```

The AudiolItem type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AudioItem</td>
<td>Initializes a new instance of the AudioItem class</td>
</tr>
<tr>
<td>AudioItem(AudiolItem)</td>
<td>Copy constructor</td>
</tr>
</tbody>
</table>

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
category  the AudioCategory the audio item belongs to.

Top

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResetSequence</td>
<td>Resets the sub-item sequence. (So if you are using a sequence mode the first sub-item will be played next)</td>
</tr>
<tr>
<td>UnloadAudioClip</td>
<td>Unloads the AudioClip from memory.</td>
</tr>
</tbody>
</table>

Top

Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>audioSource_MaxDistance</td>
<td>Overrides the AudioSource MaxDistance value if overrideAudioSourceSettings is enabled.</td>
</tr>
<tr>
<td>audioSource_MinDistance</td>
<td>Overrides the AudioSource MinDistance value if overrideAudioSourceSettings is enabled.</td>
</tr>
<tr>
<td>Delay</td>
<td>Defers the playback of the audio item for Delay seconds.</td>
</tr>
<tr>
<td>DestroyOnLoad</td>
<td>If disabled, the audio will keep on playing if a new</td>
</tr>
</tbody>
</table>
scene is loaded.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loop</td>
<td>If enabled the audio item will get looped when played.</td>
</tr>
<tr>
<td>loopSequenceCount</td>
<td>The number of sub-items to be played in the loop mode LoopSequence.</td>
</tr>
<tr>
<td>loopSequenceOverlap</td>
<td>Specifies a time overlap for the LoopSequence.</td>
</tr>
<tr>
<td>loopSequenceRandomDelay</td>
<td>Specifies a random delay for the LoopSequence.</td>
</tr>
<tr>
<td>loopSequenceRandomPitch</td>
<td>Specifies a random pitch for the LoopSequence.</td>
</tr>
<tr>
<td>loopSequenceRandomVolume</td>
<td>Specifies a random volume for the LoopSequence.</td>
</tr>
<tr>
<td>MaxInstanceCount</td>
<td>Assures that the same audio item will not be played more than MaxInstanceCount times simultaneously.</td>
</tr>
<tr>
<td>MinTimeBetweenPlayCalls</td>
<td>Assures that the same audio item will not be played multiple times within this time frame. This is useful if several events triggered at almost the same time want to play the same audio item which can cause unwanted noise artifacts.</td>
</tr>
<tr>
<td>Name</td>
<td>The unique name of the audio item ( = audioID )</td>
</tr>
<tr>
<td><strong>overrideAudioSourceSettings</strong></td>
<td>If enabled you can specify specific AudioSource settings</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>------------------------------------------------------</td>
</tr>
<tr>
<td><strong>RandomDelay</strong></td>
<td>This is the general random delay variation for the sub items in this audio item</td>
</tr>
<tr>
<td><strong>RandomPitch</strong></td>
<td>This is the general random pitch variation for the sub items in this audio item</td>
</tr>
<tr>
<td><strong>RandomVolume</strong></td>
<td>This is the general random volume variation for the sub items in this audio item</td>
</tr>
<tr>
<td><strong>spatialBlend</strong></td>
<td>Overrides the AudioSource spatialBlend value (0=2D 1=3D)</td>
</tr>
<tr>
<td><strong>SubItemPickMode</strong></td>
<td>Determines which AudioSubItem is chosen when playing an AudioItem</td>
</tr>
<tr>
<td><strong>subItems</strong></td>
<td>Define your audio sub-item using the Unity inspector.</td>
</tr>
<tr>
<td><strong>Volume</strong></td>
<td>The volume applied to all audio sub-items of this audio item.</td>
</tr>
</tbody>
</table>

**Top**

**Remarks**

Audioltem objects are defined in an AudioCategory using the Unity inspector.

**See Also**
Audioltem Constructor

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Audioltem</code></td>
<td>Initializes a new instance of the <code>Audioltem</code> class</td>
</tr>
<tr>
<td><code>Audioltem(Audioltem)</code></td>
<td>Copy constructor</td>
</tr>
</tbody>
</table>

See Also

Reference

Audioltem Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioItem Constructor

Initializes a new instance of the AudioItem class

**Namespace:** (Default Namespace)

**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public AudioItem()
```

### See Also

- Reference
  - AudioItem Class
  - AudioItem Overload
  - (Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudiolItem Constructor (AudiolItem)

Copy constructor

**Namespace:**  (Default Namespace)  
**Assembly:**  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

> Syntax

```
public AudiolItem(
    AudiolItem orig
)
```

**Parameters**

*orig*

Type:  (Default Namespace)AudiolItem

[Missing <param name="orig"/> documentation for "M:AudiolItem.#ctor(AudiolItem)""]

> See Also

Reference

AudiolItem Class
AudiolItem Overload
((Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
The **AudioItem** type exposes the following members.

## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>category</td>
<td>the <strong>AudioCategory</strong> the audio item belongs to.</td>
</tr>
</tbody>
</table>

## See Also

**Reference**

**AudioItem Class**

**Namespace**

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioItem Category Property

the **AudioCategory** the audio item belongs to.

**Namespace:** (Default Namespace)  
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public AudioCategory category { get; }
```

Property Value  
Type: **AudioCategory**

### See Also

Reference  
**AudioItem Class**  
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH  
Send comments on this topic to ClockStone Support Email
AudioItem Methods

The `AudioItem` type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ResetSequence</code></td>
<td>Resets the sub-item sequence. (So if you are using a sequence mode the first sub-item will be played next)</td>
</tr>
<tr>
<td><code>UnloadAudioClip</code></td>
<td>Unloads the AudioClip from memory.</td>
</tr>
</tbody>
</table>

See Also

Reference

`AudioItem Class`

`(Default Namespace) Namespace`

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioItemResetSequence Method

Resets the sub-item sequence. (So if you are using a sequence mode the first sub-item will be played next)

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```c#
public void ResetSequence()
```

See Also

Reference
AudioItem Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioItemUnloadAudioClip Method

Unloads the AudioClip from memory.

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public void UnloadAudioClip()
```

### Remarks

You will still be able to play the AudioClip, but you may experience performance hiccups when Unity reloads the audio asset.

### See Also

Reference
AudioItem Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
# AudiolItem Fields

The **AudiolItem** type exposes the following members.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>audioSource_MaxDistance</td>
<td>Overrides the AudioSource MaxDistance value if <code>overrideAudioSourceSettin</code> is enabled.</td>
</tr>
<tr>
<td>audioSource_MinDistance</td>
<td>Overrides the AudioSource MinDistance value if <code>overrideAudioSourceSettin</code> is enabled.</td>
</tr>
<tr>
<td>Delay</td>
<td>Defers the playback of the audio item for <code>Delay</code> seconds.</td>
</tr>
<tr>
<td>DestroyOnLoad</td>
<td>If disabled, the audio will keep on playing if a new scene is loaded.</td>
</tr>
<tr>
<td>Loop</td>
<td>If enabled the audio item will get looped when played.</td>
</tr>
<tr>
<td>loopSequenceCount</td>
<td>The number of sub-items to be played in the loop mode <code>LoopSequence</code>.</td>
</tr>
<tr>
<td>loopSequenceOverlap</td>
<td>Specifies a time overlap for the <code>LoopSequence</code>.</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>loopSequenceRandomDelay</td>
<td>Specifies a random delay for the LoopSequence</td>
</tr>
<tr>
<td>loopSequenceRandomPitch</td>
<td>Specifies a random pitch for the LoopSequence</td>
</tr>
<tr>
<td>loopSequenceRandomVolume</td>
<td>Specifies a random volume for the LoopSequence</td>
</tr>
<tr>
<td>MaxInstanceCount</td>
<td>Assures that the same audio item will not be played more than MaxInstanceCount times simultaneously.</td>
</tr>
<tr>
<td>MinTimeBetweenPlayCalls</td>
<td>Assures that the same audio item will not be played multiple times within this time frame. This is useful if several events triggered at almost the same time want to play the same audio item which can cause unwanted noise artifacts.</td>
</tr>
<tr>
<td>Name</td>
<td>The unique name of the audio item ( = audioID )</td>
</tr>
<tr>
<td>overrideAudioSourceSettings</td>
<td>If enabled you can specify specific AudioSource settings</td>
</tr>
<tr>
<td>RandomDelay</td>
<td>This is the general random delay variation for the sub items in this audio item</td>
</tr>
<tr>
<td>RandomPitch</td>
<td>This is the general random pitch variation for the sub items in this audio item</td>
</tr>
<tr>
<td><strong>RandomVolume</strong></td>
<td>This is the general random volume variation for the sub items in this audio item</td>
</tr>
<tr>
<td>-----------------------</td>
<td>--------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>spatialBlend</strong></td>
<td>Overrides the AudioSource spatialBlend value (0=2D 1=3D)</td>
</tr>
<tr>
<td><strong>SubItemPickMode</strong></td>
<td>Determines which AudioSubItem is chosen when playing an AudioItem</td>
</tr>
<tr>
<td><strong>subItems</strong></td>
<td>Define your audio sub-item using the Unity inspector.</td>
</tr>
<tr>
<td><strong>Volume</strong></td>
<td>The volume applied to all audio sub-items of this audio item.</td>
</tr>
</tbody>
</table>

**See Also**

Reference

*AudioItem Class*

*(Default Namespace) Namespace*
AudioItemaudioSource_MaxDistance

Field

Overrides the AudioSource MaxDistance value if overrideAudioSourceSettings is enabled.

Namespace:  (Default Namespace)
Assembly:  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```c#
public float audioSource_MaxDistance
```

Field Value

Type:  Single

See Also

Reference

AudioItem Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioltemaudioSource_MinDistance

Field

Overrides the AudioSource MinDistance value if overrideAudioSourceSettings is enabled.

Namespace:  (Default Namespace)
Assembly:  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```c#
public float audioSource_MinDistance
```

Field Value
Type:  Single

See Also

Reference
Audioltem Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioItemDelay Field

Defers the playback of the audio item for Delay seconds.

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

## Syntax

```csharp
public float Delay
```

Field Value

Type: Single

## See Also

Reference

AudioItem Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioItemDestroyOnLoad Field

If disabled, the audio will keep on playing if a new scene is loaded.

**Namespace:** (Default Namespace)

**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```c#
public bool DestroyOnLoad
```

### Field Value

**Type:** Boolean

### See Also

Reference

AudioItem Class

(Default Namespace) Namespace

---

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioItemLoop Field

If enabled the audio item will get looped when played.

**Namespace:**  (Default Namespace)
**Assembly:**  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```
public AudioItemLoopMode Loop
```

**Field Value**
**Type:** AudioItemLoopMode

### See Also

Reference

**AudioItem Class**  
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioItemloopSequenceCount Field

The number of sub-items to be played in the loop modes LoopSequence.

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

**Syntax**

```c#
public int loopSequenceCount
```

**Field Value**
Type: Int32

**Remarks**

Specify 0 to loop infinitely (This is also the default value). In PlaySequenceAndLoopLast mode as many sub-item will be picked as there are sub-items specified for this audio item.

**See Also**

Reference
AudioItem Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioItem.loopSequenceOverlap Field

Specifies a time overlap for the LoopSequence

Namespace:  (Default Namespace)
Assembly:  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```c#
public float loopSequenceOverlap
```

Field Value
Type:  Single

Remarks
Positive values mean an overlap, negative values mean a gap between two consequent sub-items in the loop sequence.

See Also
Reference
AudioItem Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioItemloopSequenceRandomDelay Field

Specifies a random delay for the LoopSequence

**Namespace:** (Default Namespace)  
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public float loopSequenceRandomDelay
```

### Field Value

Type: Single

### Remarks

A random delay between 0 and this value will be added between two subsequent subitems in the LoopSequence. Can be combined with loopSequenceOverlap.

### See Also

Reference

AudioItem Class  
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH  
Send comments on this topic to ClockStone Support Email
**AudioltemloopSequenceRandomPitch Field**

Specifies a random pitch for the `LoopSequence`.

**Namespace:**  (Default Namespace)
**Assembly:**  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public float loopSequenceRandomPitch
```

**Field Value**
Type: `Single`

### Remarks

A random pitch between 0 and this value will be added to each subitem played in the `LoopSequence`.

### See Also

**Reference**
- `Audioltem Class`
- (Default Namespace) Namespace

---

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioItemloopSequenceRandomVol Field

Specifies a random volume for the LoopSequence

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```
public float loopSequenceRandomVolume
```

### Field Value

Type: Single

### Remarks

A random volume value % will be added to each subitem played in the 'LoopSequence'. Will be combined with subitem random volume value.

### See Also

Reference
- AudioItem Class
  (Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioItemMaxInstanceCount Field

Assures that the same audio item will not be played more than MaxInstanceCount times simultaneously.

**Namespace:** (Default Namespace)

**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public int MaxInstanceCount
```

### Field Value

**Type:** Int32

### Remarks

Set to 0 to disable.

### See Also

Reference

AudioItem Class (Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioItemMinTimeBetweenPlayCalls Field

Assures that the same audio item will not be played multiple times within this time frame. This is useful if several events triggered at almost the same time want to play the same audio item which can cause unwanted noise artifacts.

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```csharp
public float MinTimeBetweenPlayCalls
```

Field Value
Type: Single

See Also
Reference
AudioItem Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioItemName Field

The unique name of the audio item (= audioID)

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```
public string Name
```

**Field Value**

Type: String

### See Also

Reference

AudioItem Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioItemoverrideAudioSourceSettings Field

If enabled you can specify specific AudioSource settings

Namespace:  (Default Namespace)
Assembly:  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```c#
public bool overrideAudioSourceSettings
```

Field Value
Type: Boolean

See Also

Reference
AudioItem Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
**AudioItemRandomDelay Field**

This is the general random delay variation for the sub items in this audio item

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public float RandomDelay
```

Field Value
Type: **Single**

### See Also

**Reference**
- AudioItem Class
- (Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioItemRandomPitch Field

This is the general random pitch variation for the sub items in this audio item

**Namespace:** (Default Namespace)  
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

▲ Syntax

```csharp
public float RandomPitch
```

Field Value  
Type: **Single**

▲ See Also

Reference  
**AudioItem Class**  
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH  
Send comments on this topic to ClockStone Support Email
AudioItem RandomVolume Field

This is the general random volume variation for the sub items in this audio item

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```csharp
public float RandomVolume
```

Field Value
Type: Single

See Also

Reference
AudioItem Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioItem.spatialBlend Field

Overridden the AudioSource spatialBlend value (0=2D 1=3D)

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```csharp
public float spatialBlend
```

Field Value
Type: Single

See Also

Reference
AudioItem Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioltemSubItemPickMode
Field

Determines which AudioSubItem is chosen when playing an Audioltem

Namespace:  (Default Namespace)
Assembly:  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```c#
public AudioPickSubItemMode SubItemPickMode
```

Field Value
Type: AudioPickSubItemMode

See Also

Reference
Audioltem Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioItemsubItems Field

Define your audio sub-items using the Unity inspector.

**Namespace:** (Default Namespace)  
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```
public AudioSubItem[] subItems
```

Field Value  
Type: AudioSubItem

### See Also

Reference  
AudioItem Class  
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioItemVolume Field

The volume applied to all audio sub-items of this audio item.

**Namespace:**  (Default Namespace)

**Assembly:**  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public float Volume
```

### Field Value

**Type:**  Single

### See Also

**Reference**

AudioItem Class

(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioItemLoopMode

Enumeration

AudioItem loop mode.

**Namespace:**  (Default Namespace)
**Assembly:**  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```csharp
[SerializableAttribute]
public enum LoopMode
```

Members

<table>
<thead>
<tr>
<th>Member name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DoNotLoop</td>
<td>0</td>
<td>No looping.</td>
</tr>
<tr>
<td>LoopSubItem</td>
<td>1</td>
<td>The chosen subitem (in dependence of the SubItemPickMode will be looped.</td>
</tr>
<tr>
<td>LoopSequence</td>
<td>2</td>
<td>After the subitem chosen in dependence of the SubItemPickMode has stopped playing, a new subitem will be chosen and played.</td>
</tr>
</tbody>
</table>
PlaySequenceAndLoopLast 4
Play as many sub-items as specified by loopSequenceCount and loop the last one picked. Specify zero to play as many sub-items as specified in this audio item.

IntroLoopOutroSequence 5
Play as many sub-items as specified by loopSequenceCount (as intro) and loop the second last one picked. If see AudioItem.Stop() is called the very last list played as an outro.

See Also
Reference
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioLog Class

[Missing <summary> documentation for "T:AudioLog"]

▲ Inheritance Hierarchy

System
Object  (Default Namespace)AudioLog

Namespace:  (Default Namespace)
Assembly:  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

▲ Syntax

C#  Copy

```
public static class AudioLog
```

The AudioLog type exposes the following members.

▲ Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![ disclosed ]  Clear</td>
<td></td>
</tr>
<tr>
<td>![ disclosed ]  Log</td>
<td></td>
</tr>
</tbody>
</table>

▲ Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![ disclosed ]  logData</td>
<td></td>
</tr>
<tr>
<td>![ disclosed ]  onLogUpdated</td>
<td></td>
</tr>
</tbody>
</table>
AudioLog Methods

The AudioLog type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>•S</td>
<td>Clear</td>
</tr>
<tr>
<td>•S</td>
<td>Log</td>
</tr>
</tbody>
</table>

See Also

Reference

AudioLog Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioLogClear Method


Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

C#

```csharp
public static void Clear()
```

See Also

Reference
AudioLog Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioLogLog Method


Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```
public static void Log(  
    AudioLogLogData playClipData
)
```

Parameters

PlayClipData
Type: (Default Namespace)AudioLogLogData

[Missing <param name="playClipData"/> documentation for "M:AudioLog.Log(AudioLog.LogData)"

See Also

Reference
AudioLog Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioLog Fields

The AudioLog type exposes the following members.

Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>logData</td>
<td></td>
</tr>
<tr>
<td>onLogUpdated</td>
<td></td>
</tr>
</tbody>
</table>

See Also

Reference

AudioLog Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioLog.logData Field

[Missing <summary> documentation for "F:AudioLog.logData"]

Namespace:  (Default Namespace)
Assembly:  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```
public static LinkedList<AudioLog.LogData> logData
```

Field Value
Type: LinkedList<AudioLog.LogData>

See Also

Reference
AudioLog Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioLogonLogUpdated Field

[Missing <summary> documentation for "F:AudioLog.onLogUpdated"]

Namespace:  (Default Namespace)
Assembly:  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```csharp
public static Action onLogUpdated
```

Field Value
Type:  Action

See Also

Reference
AudioLog Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioLogLogData Class

[Missing <summary> documentation for "T:AudioLog.LogData"]

Inheritance Hierarchy

System

  Object

  (Default Namespace) AudioLogLogData

  (Default Namespace) AudioLogLogData_Destroy

  (Default Namespace) AudioLogLogData_PlayClip

  (Default Namespace) AudioLogLogData_SkippedPlay

  (Default Namespace) AudioLogLogData_Stop

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

C#

```csharp
public abstract class LogData
```

The AudioLogLogData type exposes the following members.

Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>time</td>
<td></td>
</tr>
</tbody>
</table>

See Also

Reference

(Default Namespace) Namespace
Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
LogData Fields

The AudioLogLogData type exposes the following members.

Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>time</td>
<td></td>
</tr>
</tbody>
</table>

See Also

Reference

AudioLogLogData Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioLogLogData datetime Field

[Missing <summary> documentation for "F:AudioLog.LogData.time"]

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

C#

```csharp
public float time
```

Field Value
Type: Single

See Also

Reference
AudioLogLogData Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioLogLogData_Destroy
Class

[Missing <summary> documentation for "T:AudioLog.LogData_Destroy"]

Inheritance Hierarchy

System
  Object
  (Default Namespace)
  AudioLog
    (Default Namespace)
    AudioLogLogData
      (Default Namespace)
      AudioLogLogData_Destroy

Namespace:  (Default Namespace)
Assembly:  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

C#

```csharp
public class LogData_Destroy : AudioLogLogData
```

The AudioLogLogData_Destroy type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AudioLogLogData_Destroy</td>
<td>Initializes a new instance of the AudioLogLogData_Destroy class</td>
</tr>
</tbody>
</table>

Fields
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>audioID</td>
<td></td>
</tr>
<tr>
<td>category</td>
<td></td>
</tr>
<tr>
<td>clipName</td>
<td></td>
</tr>
<tr>
<td>parentObject</td>
<td></td>
</tr>
<tr>
<td>position</td>
<td></td>
</tr>
</tbody>
</table>

**See Also**

Reference

*(Default Namespace) Namespace*

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to [ClockStone Support Email](mailto:support@clockstone.com)
**AudioLogLogData_Destroy Constructor**

Initializes a new instance of the `AudioLogLogData_Destroy` class

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public LogData_Destroy();
```

### See Also

Reference

AudioLogLogData_Destroy Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
LogData_Destroy Fields

The AudioLogLogData_Destroy type exposes the following members.

Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>audioID</td>
<td></td>
</tr>
<tr>
<td>category</td>
<td></td>
</tr>
<tr>
<td>clipName</td>
<td></td>
</tr>
<tr>
<td>parentObject</td>
<td></td>
</tr>
<tr>
<td>position</td>
<td></td>
</tr>
</tbody>
</table>

See Also

Reference
AudioLogLogData_Destroy Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioLogLogData_Destroy.audioID Field

[Missing <summary> documentation for "F:AudioLog.LogData_Destroy.audioID"]

Namespace:  (Default Namespace)
Assembly:  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```csharp
public string audioID
```

Field Value
Type:  String

See Also

Reference
AudioLogLogData_Destroy Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioLogLogData_Destroy.category Field

[Missing <summary> documentation for "F:AudioLog.LogData_Destroy.category"]

**Namespace:** (Default Namespace)  
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public string category
```

### Field Value

**Type:** String

### See Also

**Reference**

AudioLogLogData_Destroy Class  
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioLogLogData_Destroy.clipName Field

[Missing <summary> documentation for "F:AudioLog.LogData_Destroy.clipName"]

Namespace:  (Default Namespace)
Assembly:  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```c
public string clipName
```

Field Value
Type:  String

See Also

Reference
AudioLogLogData_Destroy Class  
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioLogLogData_Destroy::parentObject

[Missing <summary> documentation for "F:AudioLog.LogData_Destroy.parentObject"]

**Namespace:** (Default Namespace)  
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

## Syntax

```csharp
public string parentObject
```

### Field Value

- **Type:** String

## See Also

- **Reference**  
  AudioLogLogData_Destroy Class  
  (Default Namespace) Namespace

---

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioLogLogData_Destroy.position Field

[Missing <summary> documentation for "F:AudioLog.LogData_Destroy.position"]

**Namespace:**  (Default Namespace)
**Assembly:**  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public Vector3 position
```

**Field Value**

**Type:**  **Vector3**

### See Also

**Reference**

AudioLogLogData_Destroy Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioLogLogData_PlayClip Class

[Missing <summary> documentation for "T:AudioLog.LogData_PlayClip"]

Inheritance Hierarchy

System Object (Default Namespace) AudioLogLogData (Default Namespace) AudioLogLogData_PlayClip

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

C#

```csharp
public class LogData_PlayClip : AudioLogLogData
```

The AudioLogLogData_PlayClip type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AudioLogLogData_PlayClip</td>
<td>Initializes a new instance of the AudioLogLogData_PlayClip class</td>
</tr>
</tbody>
</table>

Fields
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>audioID</td>
<td></td>
</tr>
<tr>
<td>category</td>
<td></td>
</tr>
<tr>
<td>clipName</td>
<td></td>
</tr>
<tr>
<td>delay</td>
<td></td>
</tr>
<tr>
<td>parentObject</td>
<td></td>
</tr>
<tr>
<td>pitch</td>
<td></td>
</tr>
<tr>
<td>position</td>
<td></td>
</tr>
<tr>
<td>scheduledDspTime</td>
<td></td>
</tr>
<tr>
<td>startTime</td>
<td></td>
</tr>
<tr>
<td>volume</td>
<td></td>
</tr>
</tbody>
</table>

See Also

Reference

( Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioLogLogData_PlayClip Constructor

Initializes a new instance of the AudioLogLogData_PlayClip class

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public LogData_PlayClip()
```

### See Also

Reference
AudioLogLogData_PlayClip Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
LogData_PlayClip Fields

The AudioLogLogData_PlayClip type exposes the following members.

Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>audioID</td>
<td></td>
</tr>
<tr>
<td>category</td>
<td></td>
</tr>
<tr>
<td>clipName</td>
<td></td>
</tr>
<tr>
<td>delay</td>
<td></td>
</tr>
<tr>
<td>parentObject</td>
<td></td>
</tr>
<tr>
<td>pitch</td>
<td></td>
</tr>
<tr>
<td>position</td>
<td></td>
</tr>
<tr>
<td>scheduledDspTime</td>
<td></td>
</tr>
<tr>
<td>startTime</td>
<td></td>
</tr>
<tr>
<td>volume</td>
<td></td>
</tr>
</tbody>
</table>

See Also

Reference

AudioLogLogData_PlayClip Class
(Default Namespace) Namespace
AudioLogLogData_PlayClip audioID Field

[Missing <summary> documentation for "F:AudioLog.LogData_PlayClip.audioID"]

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```c#
public string audioID
```

Field Value
Type: String

See Also

Reference
AudioLogLogData_PlayClip Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioLogLogData_PlayClip.category Field

[Missing <summary> documentation for "F:AudioLog.LogData_PlayClip.category"]

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public string category
```

Field Value
Type: String

### See Also

Reference
AudioLogLogData_PlayClip Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioLogLogData_PlayClip.clipName Field

[Missing <summary> documentation for "F:AudioLog.LogData_PlayClip.clipName"]

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```csharp
public string clipName
```

Field Value
Type: String

See Also

Reference
AudioLogLogData_PlayClip Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioLogLogData_PlayClipDelay Field

[Missing <summary> documentation for "F:AudioLog.LogData_PlayClip.delay"]

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.d11) Version: 8.2.0.0 (8.2.0.0)

Syntax

```
public float delay
```

Field Value
Type: Single

See Also

Reference
AudioLogLogData_PlayClip Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioLogLogData_PlayClip.parentObject

Field

[Missing <summary> documentation for "F:AudioLog.LogData_PlayClip.parentObject"]

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```
public string parentObject
```

Field Value
Type: String

See Also

Reference
AudioLogLogData_PlayClip Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioLogLogData_PlayClipField

[Missing <summary> documentation for "F:AudioLog.LogData_PlayClip.pitch"]

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```
public float pitch
```

Field Value
Type: Single

See Also

Reference
AudioLogLogData_PlayClip Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioLogLogData_PlayClip.position Field

[Missing <summary> documentation for "F:AudioLog.LogData_PlayClip.position"]

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

C#

```csharp
public Vector3 position
```

Field Value
Type: Vector3

See Also

Reference
AudioLogLogData_PlayClip Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioLogLogData_PlayClip scheduledDspTime Field

[Missing <summary> documentation for "F:AudioLog.LogData_PlayClip.scheduledDspTime"]

Namespace: (Default Namespace)  
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```csharp
public float scheduledDspTime
```

Field Value  
Type: Single

See Also

Reference  
AudioLogLogData_PlayClip Class  
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH  
Send comments on this topic to ClockStone Support Email
AudioLogLogData_PlayClipstartTime Field

[Missing <summary> documentation for "F:AudioLog.LogData_PlayClip.startTime"]

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public float startTime
```

### Field Value

- **Type:** Single

### See Also

**Reference**
- AudioLogLogData_PlayClip Class
- (Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioLogLogData_PlayClip volume Field

[Missing <summary> documentation for "F:AudioLog.LogData_PlayClip.volume"]

**Namespace:**  (Default Namespace)  
**Assembly:**  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public float volume
```

**Field Value**

**Type:**  Single

### See Also

**Reference**

AudioLogLogData_PlayClip Class  
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioLogLogData_SkippedPlay Class

[Missing <summary> documentation for "T:AudioLog.LogData_SkippedPlay"]

Inheritance Hierarchy

System Object  (Default Namespace) AudioLogLogData  
                (Default Namespace) AudioLogLogData_SkippedPlay

Namespace:  (Default Namespace)  
Assembly:  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```csharp
public class LogData_SkippedPlay : AudioLogLogData
```

The AudioLogLogData_SkippedPlay type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AudioLogLogData_SkippedPlay</td>
<td>Initializes a new instance of the AudioLogLogData_SkippedPlay class</td>
</tr>
</tbody>
</table>

Fields
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>audioID</td>
<td></td>
</tr>
<tr>
<td>category</td>
<td></td>
</tr>
<tr>
<td>delay</td>
<td></td>
</tr>
<tr>
<td>parentObject</td>
<td></td>
</tr>
<tr>
<td>position</td>
<td></td>
</tr>
<tr>
<td>reasonForSkip</td>
<td></td>
</tr>
<tr>
<td>scheduledDspTime</td>
<td></td>
</tr>
<tr>
<td>startTime</td>
<td></td>
</tr>
<tr>
<td>volume</td>
<td></td>
</tr>
</tbody>
</table>

See Also

Reference

(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioLogLogData_SkippedPlay Constructor

Initializes a new instance of the AudioLogLogData_SkippedPlay class

Namespace:  (Default Namespace)  
Assembly:  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```csharp
public LogData_SkippedPlay()
```

See Also

Reference

AudioLogLogData_SkippedPlay Class  
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
LogData_SkippedPlay Fields

The AudioLogLogData_SkippedPlay type exposes the following members.

### Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>audioID</td>
<td></td>
</tr>
<tr>
<td>category</td>
<td></td>
</tr>
<tr>
<td>delay</td>
<td></td>
</tr>
<tr>
<td>parentObject</td>
<td></td>
</tr>
<tr>
<td>position</td>
<td></td>
</tr>
<tr>
<td>reasonForSkip</td>
<td></td>
</tr>
<tr>
<td>scheduledDspTime</td>
<td></td>
</tr>
<tr>
<td>startTime</td>
<td></td>
</tr>
<tr>
<td>volume</td>
<td></td>
</tr>
</tbody>
</table>

See Also

Reference
AudioLogLogData_SkippedPlay Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioLogLogData_SkippedPlay audioID

[Missing <summary> documentation for "F:AudioLog.LogData_SkippedPlay.audioID"]

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

▶ Syntax

C#

```csharp
public string audioID
```

Field Value
Type: String

▶ See Also

Reference
AudioLogLogData_SkippedPlay Class
(Default Namespace) Namespace
AudioLogLogData_SkippedPlay category Field

[Missing <summary> documentation for "F:AudioLog.LogData_SkippedPlay.category"]

Namespace:  (Default Namespace)
Assembly:  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```csharp
public string category
```

Field Value
Type:  String

See Also

Reference
AudioLogLogData_SkippedPlay Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioLogLogData_SkippedPlayDelay Field

[Missing <summary> documentation for "F:AudioLog.LogData_SkippedPlay.delay"]

**Namespace:** (Default Namespace)  
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```
c#  Copy

public float delay
```

**Field Value**  
**Type:** Single

### See Also

**Reference**  
AudioLogLogData_SkippedPlay Class  
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioLogLogData_SkippedPlay parent Object Field

[Missing <summary> documentation for "F:AudioLog.LogData_SkippedPlay.parentObject"]

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public string parentObject
```

**Field Value**
**Type:** String

### See Also

Reference
AudioLogLogData_SkippedPlay Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioLogLogData_SkippedPlayField

[Missing <summary> documentation for "F:AudioLog.LogData_SkippedPlay.position"]

Namespace:  (Default Namespace)
Assembly:  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```
public Vector3 position
```

Field Value
Type:  Vector3

See Also

Reference
AudioLogLogData_SkippedPlay Class
(Default Namespace) Namespace
AudioLogLogData_SkippedPlayField

[Missing <summary> documentation for "F:AudioLog.LogData_SkippedPlay.reasonForSkip"]

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```csharp
public string reasonForSkip
```

Field Value
Type: String

See Also

Reference
AudioLogLogData_SkippedPlay Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioLogLogData_SkippedPlay Field

[Missing <summary> documentation for "F:AudioLog.LogData_SkippedPlay.scheduledDspTime"]

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

C#

```csharp
public float scheduledDspTime
```

Field Value
Type: Single

See Also

Reference
AudioLogLogData_SkippedPlay Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioLogLogData_SkippedPlay Field

[Missing <summary> documentation for "F:AudioLog.LogData_SkippedPlay.startTime"]

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

⚠ Syntax

```csharp
public float startTime
```

Field Value
Type: Single

⚠ See Also

Reference
AudioLogLogData_SkippedPlay Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioLogLogData_SkippedPlayField

[Missing <summary> documentation for "F:AudioLog.LogData_SkippedPlay.volume"]

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```c#
public float volume
```

Field Value
Type: Single

See Also

Reference
AudioLogLogData_SkippedPlay Class
(Default Namespace) Namespace
AudioLogLogData_Stop Class

[Missing <summary> documentation for "T:AudioLog.LogData_Stop"]

▶ Inheritance Hierarchy

System
  └ Object
    (Default Namespace)AudioLog
      (Default Namespace)AudioLogLogData
        (Default Namespace)AudioLogLogData_Stop

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

▶ Syntax

```
[C#]
public class LogData_Stop : AudioLogLogData
```

The AudioLogLogData_Stop type exposes the following members.

▶ Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AudioLogLogData_Stop</td>
<td>Initializes a new instance of the AudioLogLogData_Stop class</td>
</tr>
</tbody>
</table>

▶ Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>audioID</td>
<td>category</td>
</tr>
<tr>
<td>-------------------------</td>
<td>----------</td>
</tr>
</tbody>
</table>

**See Also**

Reference

(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioLogLogData_Stop Constructor

Initializes a new instance of the AudioLogLogData_Stop class

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public LogData_Stop()
```

### See Also

Reference
AudioLogLogData_Stop Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
LogData_Stop Fields

The AudioLogLogData_Stop type exposes the following members.

Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>audioID</td>
<td></td>
</tr>
<tr>
<td>category</td>
<td></td>
</tr>
<tr>
<td>clipName</td>
<td></td>
</tr>
<tr>
<td>parentObject</td>
<td></td>
</tr>
<tr>
<td>position</td>
<td></td>
</tr>
</tbody>
</table>

See Also

Reference
AudioLogLogData_Stop Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioLogLogData_Stop.audioID Field

[Missing <summary> documentation for "F:AudioLog.LogData_Stop.audioID"]

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```csharp
public string audioID
```

Field Value
Type: String

See Also

Reference
AudioLogLogData_Stop Class
(Default Namespace)Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
Field

[Missing <summary> documentation for "F:AudioLog.LogData_Stop.category"]

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```
public string category
```

### Field Value

Type: String

### See Also

**Reference**

AudioLogLogData_Stop Class
(Default Namespace) Namespace

---

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioLogLogData_Stop.clipName Field

[Missing <summary> documentation for "F:AudioLog.LogData_Stop.clipName"]

**Namespace:** (Default Namespace)  
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```
public string clipName
```

**Field Value**  
**Type:** String

### See Also

**Reference**  
AudioLogLogData_Stop Class  
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH  
Send comments on this topic to ClockStone Support Email
AudioLogLogData_StopparentObject Field

[Missing <summary> documentation for "F:AudioLog.LogData_Stop.parentObject"]

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

⚠ Syntax

```csharp
public string parentObject
```

Field Value
Type: String

⚠ See Also

Reference
AudioLogLogData_Stop Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioLogLogData_Stop position Field

[Missing <summary> documentation for "F:AudioLog.LogData_Stop.position"]

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```csharp
public Vector3 position
```

Field Value
Type: Vector3

See Also

Reference
AudioLogLogData_Stop Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioObject Class

The object playing the audio clip associated with a AudioSubItem

Azure Inheritance Hierarchy

- SystemObject
- Object
- Component
- Behaviour
- MonoBehaviour
  (Default Namespace)RegisteredComponent
  (Default Namespace)AudioObject

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```csharp
public class AudioObject : RegisteredComponent
```

The AudioObject type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AudioObject</td>
<td>Initializes a new instance of the AudioObject class</td>
</tr>
</tbody>
</table>

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top</td>
<td></td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>audioID</td>
<td>Gets the audio ID.</td>
</tr>
<tr>
<td>audioItem</td>
<td>Gets the corresponding AudioSubItem.</td>
</tr>
<tr>
<td>audioObjectTime</td>
<td>Returns the high precision local time of this audio object.</td>
</tr>
<tr>
<td>audioTime</td>
<td>Sets or gets the current audio time relative to ClipStartTime.</td>
</tr>
<tr>
<td>category</td>
<td>Gets the category.</td>
</tr>
<tr>
<td>clipLength</td>
<td>Gets the length of the clip.</td>
</tr>
<tr>
<td>completelyPlayedDelegate</td>
<td>Gets or sets the delegate to be called once an audio clip was completely played.</td>
</tr>
<tr>
<td>isFadeOutComplete</td>
<td>return true if the audio has completely faded out.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>isFadingIn</code></td>
<td>Return <code>true</code> if the audio is currently fading in</td>
</tr>
<tr>
<td><code>isFadingOut</code></td>
<td>Return <code>true</code> if the audio is currently fading out</td>
</tr>
<tr>
<td><code>isFadingOutOrScheduled</code></td>
<td>Return <code>true</code> if the audio is currently fading out or is scheduled to fade out</td>
</tr>
<tr>
<td><code>isPlayedAsMusicOrAmbienceSound</code></td>
<td>Returns <code>true</code> if the audio object is treated as music</td>
</tr>
<tr>
<td><code>pan</code></td>
<td>Sets or gets the audio pan.</td>
</tr>
<tr>
<td><code>pitch</code></td>
<td>Sets or gets the audio pitch.</td>
</tr>
<tr>
<td><code>playCalledAtTime</code></td>
<td>Gets the <code>systemTime</code> at which the audio Play() function was called.</td>
</tr>
<tr>
<td><code>primaryAudioSource</code></td>
<td>Returns the primary</td>
</tr>
<tr>
<td>Method/Property</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>scheduledPlayingAtDspTime</td>
<td>Gets or sets the DSP time at which the audio is scheduled to play.</td>
</tr>
<tr>
<td>secondaryAudioSource</td>
<td>returns the secondary AudioSource</td>
</tr>
<tr>
<td>startedPlayingAtTime</td>
<td>Gets the systemTime at which the audio started playing.</td>
</tr>
<tr>
<td>stopAfterFadeOut</td>
<td>If enabled, the audio will stop playing if a fadeout is finished.</td>
</tr>
<tr>
<td>subItem</td>
<td>Gets the corresponding AudioSubItem</td>
</tr>
<tr>
<td>timeUntilEnd</td>
<td>Gets the time until the clip will stop.</td>
</tr>
<tr>
<td>volume</td>
<td>Gets or sets the volume.</td>
</tr>
<tr>
<td>volumeItem</td>
<td>Gets or sets the volume of the audio.</td>
</tr>
</tbody>
</table>
### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>volumeTotal</td>
<td>Gets the total volume.</td>
</tr>
<tr>
<td>volumeTotalWithoutFade</td>
<td>Gets the total volume.</td>
</tr>
<tr>
<td>DestroyAudioObject</td>
<td>Destroys the audio object (using ObjectPoolController if pooling is enabled)</td>
</tr>
<tr>
<td>DoesBelongToCategory</td>
<td>Checks if this AudioObject belongs to a specific category</td>
</tr>
<tr>
<td>FadeIn</td>
<td>Fades-in a playing audio.</td>
</tr>
<tr>
<td>FadeOut(Single)</td>
<td>Starts a fade-out. If the AudiItem mode is is a sequence, the next sub-item will continue to play after the this sub-item is completely faded out</td>
</tr>
<tr>
<td>FadeOut(Single, Single)</td>
<td>Starts a fade-out at a specified time. If the AudiItem mode is is a sequence, the next sub-item will continue to play after the this sub-item is completely faded out</td>
</tr>
<tr>
<td>FinishSequence</td>
<td>Finishes a playing sequence, depending on the AudiItem’s loop mode:</td>
</tr>
<tr>
<td></td>
<td>• LoopSequence: The</td>
</tr>
</tbody>
</table>
sequence will stop after the current item has finished playing.

- **PlaySequenceAndLoopLast**
  The sequence will stop after the current item has finished playing. If the sequence is during the looping part the looping will stop after the current loop reached its end.

- **IntroLoopOutroSequence**: The sequence will stop after the current item has finished playing. If the sequence is during the looping part the outro will be played and the sequence will stop afterwards.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>InverseTransformPitch</strong></td>
<td>Inverse pitch transformation: <strong>TransformPitch(Single)</strong></td>
</tr>
<tr>
<td><strong>InverseTransformVolume</strong></td>
<td>Inverse volume transformation <strong>TransformVolume(Single)</strong></td>
</tr>
<tr>
<td><strong>IsPaused</strong></td>
<td>Determines whether the audio clip is paused.</td>
</tr>
<tr>
<td><strong>IsPlaying</strong></td>
<td>Determines if either the primary or the secondary audio clip is playing.</td>
</tr>
<tr>
<td><strong>IsPrimaryPlaying</strong></td>
<td>Determines if the primary audio clip is playing.</td>
</tr>
<tr>
<td><strong>IsSecondaryPlaying</strong></td>
<td>Determines if the secondary audio clip is playing.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Pause</td>
<td>Pauses the audio clip.</td>
</tr>
<tr>
<td>Pause(Single)</td>
<td>Pauses the audio clip with a fade-out.</td>
</tr>
<tr>
<td>Play</td>
<td>Plays the audio clip with the specified delay.</td>
</tr>
<tr>
<td>PlayAfter</td>
<td>Plays the specified audio after the current has finished playing</td>
</tr>
<tr>
<td>PlayNow</td>
<td>Plays the specified audio.</td>
</tr>
<tr>
<td>PlayScheduled</td>
<td>Plays the audio clip at the specified high precision DSP time (see the Unity AudioSettings.dspTime documentation)</td>
</tr>
<tr>
<td>Stop</td>
<td>Stops playing this instance.</td>
</tr>
<tr>
<td>Stop(Single)</td>
<td>Stops a playing audio with fade-out.</td>
</tr>
<tr>
<td>Stop(Single, Single)</td>
<td>Stops a playing audio with fade-out at a specified time.</td>
</tr>
<tr>
<td>SwitchAudioSources</td>
<td>Switches the primary and secondary audio source</td>
</tr>
<tr>
<td>TransformPitch</td>
<td>Transforms the pitch from semitones to a multiplicative factor</td>
</tr>
<tr>
<td>TransformVolume</td>
<td>Transforms the volume to make it perceptually more intuitive to scale and cross-fade.</td>
</tr>
<tr>
<td>Unpause</td>
<td>Unpauses the audio clip.</td>
</tr>
</tbody>
</table>
Unpause(Single)  

Unpauses the audio clip with a fade-in.

Top

Remarks

If audio object pooling is enabled make sure you store references to an AudioObject by using PoolableReferenceT

Examples

```javascript
var soundFX = new PoolableReference<AudioObject>(
  AudioController.Play(
    // some other part of the code executed later when the sound may have stopped playing
    AudioObject audioObject = soundFX.Get();
    if( audioObject != null )
    {    
      // it is safe to access audioObject here
      audioObject.Stop();
    }
  )
);  
```

See Also

Reference

(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioObject Constructor

Initializes a new instance of the AudioObject class

**Namespace:** (Default Namespace)

**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public AudioObject()
```

### See Also

**Reference**

AudioObject Class

(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
### AudioObject Properties

The **AudioObject** type exposes the following members.

#### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>audioID</td>
<td>Gets the audio ID.</td>
</tr>
<tr>
<td>audioItem</td>
<td>Gets the corresponding AudioSubItem</td>
</tr>
<tr>
<td>audioObjectTime</td>
<td>Returns the high precision local time of this audio object</td>
</tr>
<tr>
<td>audioTime</td>
<td>Sets or gets the current audio time relative to <strong>ClipStartTime</strong></td>
</tr>
<tr>
<td>category</td>
<td>Gets the category.</td>
</tr>
<tr>
<td>clipLength</td>
<td>Gets the length of the clip.</td>
</tr>
<tr>
<td>completelyPlayedDelegate</td>
<td>Gets or sets the delegate</td>
</tr>
</tbody>
</table>
to be called once an audio clip was completely played.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>isFadeOutComplete</td>
<td>return true if the audio has completely faded out</td>
</tr>
<tr>
<td>isFadingIn</td>
<td>return true if the audio is currently fading in</td>
</tr>
<tr>
<td>isFadingOut</td>
<td>return true if the audio is currently fading out</td>
</tr>
<tr>
<td>isFadingOutOrScheduled</td>
<td>return true if the audio is currently fading out or is scheduled to fade out</td>
</tr>
<tr>
<td>isPlayedAsMusicOrAmbienceSound</td>
<td>Returns true if the audio object is treated as music</td>
</tr>
<tr>
<td>pan</td>
<td>Sets or gets the audio pan.</td>
</tr>
<tr>
<td>pitch</td>
<td>Sets or gets the audio pitch.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>playCalledAtTime</td>
<td>Gets the <code>systemTime</code> at which the audio Play() function was called.</td>
</tr>
<tr>
<td>primaryAudioSource</td>
<td>returns the primary <code>AudioSource</code></td>
</tr>
<tr>
<td>scheduledPlayingAtDspTime</td>
<td>Gets or sets the DSP time at which the audio is scheduled to play.</td>
</tr>
<tr>
<td>secondaryAudioSource</td>
<td>returns the secondary <code>AudioSource</code></td>
</tr>
<tr>
<td>startedPlayingAtTime</td>
<td>Gets the <code>systemTime</code> at which the audio started playing.</td>
</tr>
<tr>
<td>stopAfterFadeOut</td>
<td>If enabled, the audio will stop playing if a fadeout is finished.</td>
</tr>
<tr>
<td>subItem</td>
<td>Gets the corresponding <code>AudioSubItem</code></td>
</tr>
<tr>
<td></td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>timeUntilEnd</td>
<td>Gets the time until the clip will stop.</td>
</tr>
<tr>
<td>volume</td>
<td>Gets or sets the volume.</td>
</tr>
<tr>
<td>volumeItem</td>
<td>Gets or sets the volume of the audio item.</td>
</tr>
<tr>
<td>volumeTotal</td>
<td>Gets the total volume.</td>
</tr>
<tr>
<td>volumeTotalWithoutFade</td>
<td>Gets the total volume.</td>
</tr>
</tbody>
</table>

**See Also**

Reference

AudioObject Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioObject.audioID Property

Gets the audio ID.

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```
public string audioID { get; }
```

Property Value

Type: `String`

### See Also

**Reference**

AudioObject Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioObjectauldioItem Property

Get the corresponding AudioSubItem

**Namespace:**  (Default Namespace)

**Assembly:**  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

## Syntax

C#  Copy

```csharp
public AudioItem audioItem { get; }
```

**Property Value**

Type: AudioItem

## See Also

Reference

AudioObject Class

(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioObjectaudioObjectTime Property

Returns the high precision local time of this audio object

**Namespace:**  (Default Namespace)

**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```c#
public double audioObjectTime { get; }
```

### Property Value

Type: **Double**

### Remarks

The local time is paused when the audio object is paused.

### See Also

Reference

AudioObject Class

(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioObject.audioTime Property

Sets or gets the current audio time relative to ClipStartTime

**Namespace:**  (Default Namespace)

**Assembly:**  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

**Syntax**

```csharp
public float audioTime { get; set; }
```

**Property Value**

Type: Single

**See Also**

Reference

AudioObject Class

(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioObject.category Property

Gets the category.

**Namespace:**  (Default Namespace)
**Assembly:**  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```c#
public AudioCategory category { get; }
```

Property Value
Type: `AudioCategory`

### See Also

Reference
**AudioObject Class**
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioObjectclipLength Property

Gets the length of the clip.

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

---

**Syntax**

```csharp
public float clipLength { get; }
```

---

**Property Value**

Type: Single

---

**Remarks**

Is effected by ClipStopTime and ClipStartTime

---

**See Also**

Reference

AudioObject Class
(Default Namespace) Namespace

---

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioObjectcompletelyPlayedDelegate Property

Gets or sets the delegate to be called once an audio clip was completely played.

Namespace:  (Default Namespace)
Assembly:  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```csharp
public AudioObjectAudioEventDelegate completelyPlayedDelegate
```

Property Value
Type:  AudioObjectAudioEventDelegate

See Also

Reference
AudioObject Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioObject.isFadeOutComplete Property

return true if the audio has completely faded out

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```csharp
public bool isFadeOutComplete { get; }
```

Property Value
Type: Boolean

See Also
Reference
AudioObject Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioObject.isFadingIn Property

return true if the audio is currently fading in

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```csharp
public bool isFadingIn { get; }
```

Property Value
Type: Boolean

See Also

Reference
AudioObject Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioObject.isFadingOut Property

return true if the audio is currently fading out

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

C#

```csharp
public bool isFadingOut { get; }
```

Property Value

Type: Boolean

Remarks

If the fade-out is complete then isFadingOut return false and isFadeOutComplete returns true

See Also

Reference

AudioObject Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
## AudioObject isFadingOutOrScheduled Property

return `true` if the audio is currently fading out or is scheduled to fade out.

**Namespace:** (Default Namespace)  
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public bool isFadingOutOrScheduled { get; }
```

**Property Value**  
Type: `Boolean`

### Remarks

`isFadingOutOrScheduled` returns `true` even if the fade out is complete.

### See Also

**Reference**  
AudioObject Class  
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH  
Send comments on this topic to ClockStone Support Email
AudioObject.isPlayedAsMusicOrAmbienceSound Property

Returns true if the audio object is treated as music

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```
public bool isPlayedAsMusicOrAmbienceSound { get; }
```

**Property Value**
Type: Boolean

### See Also

**Reference**
AudioObject Class
(Default Namespace) Namespace
AudioObjectpan Property

Sets or gets the audio pan.

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public float pan { get; set; }
```

Property Value
Type: Single

### See Also

Reference
AudioObject Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioObject.pitch Property

Sets or gets the audio pitch.

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

## Syntax

```csharp
public float pitch { get; set; }
```

Property Value
Type: Single

## See Also

Reference
AudioObject Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioObjectplayCalledAtTime Property

Gets the `systemTime` at which the audio `Play()` function was called.

**Namespace:** (Default Namespace)  
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

⚠️ **Syntax**

```c#
public double playCalledAtTime { get; }
```

**Property Value**
Type: **Double**

⚠️ **Remarks**

If a play was scheduled or delayed, the actual time at which the audio started playing is different.

⚠️ **See Also**

Reference  
AudioObject Class  
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH  
Send comments on this topic to ClockStone Support Email
AudioObject primaryAudioSource Property

returns the primary AudioSource

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```c#
public AudioSource primaryAudioSource { get; }
```

Property Value
Type: AudioSource

Remarks

some features like "loop sequence" require an additional AudioSource. Functions like Stop(), FadeIn(), etc. always act on the primary audio source.

See Also

Reference
AudioObject Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioObjects.scheduledPlayingAtDspTime Property

Gets or sets the DSP time at which the audio is scheduled to play.

**Namespace:** (Default Namespace)  
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public double scheduledPlayingAtDspTime { get; set; }
```

### Return Value
Type: Double  
Returns -1 if no audio is scheduled.

### See Also

**Reference**  
AudioObject Class  
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH  
Send comments on this topic to ClockStone Support Email
**AudioObject secondary AudioSource Property**

returns the secondary AudioSource

**Namespace:** (Default Namespace)  
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public AudioSource secondaryAudioSource { get; }
```

### Property Value

Type: **AudioSource**

### Remarks

some features like "loop sequence" require an additional AudioSource. Functions like Stop(), Fadeln(), etc. always act on the primary audio source.

### See Also

Reference

AudioObject Class  
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioObjectstartedPlayingAtTime Property

Gets the systemTime at which the audio started playing.

Namespace:  (Default Namespace)
Assembly:  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```c#
public double startedPlayingAtTime { get; }
```

Property Value
Type:  Double

Remarks
If a play was scheduled or delayed, this value is different than playCalledAtTime

See Also

Reference
AudioObject Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioObject.stopAfterFadeOut Property

If enabled, the audio will stop playing if a fadeout is finished.

**Namespace:**  (Default Namespace)

**Assembly:**  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public bool stopAfterFadeOut { get; set; }
```

### Property Value

Type: **Boolean**

### Remarks

Enabled by default.

### See Also

Reference
AudioObject Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioObjectsubItem Property

Gets the corresponding AudioSubItem

**Namespace:** (Default Namespace)  
**Assembly:** AudioToolkit (in AudioToolkit.dll)  
Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public AudioSubItem subItem { get; }
```

**Property Value**  
Type: AudioSubItem

### See Also

Reference  
AudioObject Class  
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH  
Send comments on this topic to ClockStone Support Email
AudioObject::timeUntilEnd

Property

Gets the time until the clip will stop.

**Namespace:** (Default Namespace)

**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

**Syntax**

```c#
public float timeUntilEnd { get; }
```

**Property Value**

Type: Single

**Remarks**

Is effected by ClipStopTime

**See Also**

Reference

AudioObject Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioObject volume Property

Gets or sets the volume.

**Namespace:** (Default Namespace)

**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public float volume { get; set; }
```

### Property Value

Type: Single

### Remarks

This is the adjusted volume value with which the Audio clip is currently playing. It is the value resulting from multiplying the volume of the subitem, item, the category, and the script parameter. It does not contain the global volume or the fading value. "Adjusted" means that the value does not equal Unity's internal audio clip volume value, because Unity's volume range is not distributed is a perceptually even manner.

```csharp
unityVolume = Mathf.Pow( adjustedVolume, 1.6 )
```

### See Also

**Reference**

AudioObject Class

(Default Namespace) Namespace
ClockStone Audio Toolkit for Unity - Documentation
AudioObject volumeItem Property

Gets or sets the volume of the audio item.

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

C#

```csharp
public float volumeItem { get; set; }
```

Property Value

Type: Single

Remarks

This is the adjusted volume value with which the Audio clip is currently playing. It is the value resulting from multiplying the volume of the subitem and the item. It does not contain the global volume, the category, the script parameter, or the fading value. "Adjusted" means that the value does not equal Unity's internal audio clip volume value, because Unity's volume range is not distributed in a perceptually even manner.

```csharp
unityVolume = Mathf.Pow(adjustedVolume, 1.6)
```

See Also

Reference

AudioObject Class
(Default Namespace) Namespace
AudioObject

volumeTotal Property

Gets the total volume.

**Namespace:**  (Default Namespace)

**Assembly:**  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

**Syntax**

```csharp
public float volumeTotal {
    get;
}
```

**Property Value**

Type:  Single

**Remarks**

This is the adjusted volume value with which the Audio clip is currently playing. It is the value resulting from multiplying the volume of the subitem, item, the category, the script parameter, the global volume, and the fading value. "Adjusted" means that the value does not equal Unity's internal audio clip volume value, because Unity's volume range is not distributed in a perceptually even manner.

```
unityVolume = Mathf.Pow( adjustedVolume, 1.6 )
```

**See Also**

Reference

AudioObject Class

(Default Namespace) Namespace
**AudioObject volumeTotalWithoutFade Property**

Gets the total volume.

**Namespace:** (Default Namespace)  
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public float volumeTotalWithoutFade { get; }
```

### Remarks

This is the adjusted volume value with which the Audio clip is currently playing without fade in/out. It is the value resulting from multiplying the volume of the subitem, item, the category, the script parameter, the global volume and sound muting. "Adjusted" means that the value does not equal Unity's internal audio clip volume value, because Unity's volume range is not distributed in a perceptually even manner.

`unityVolume = Mathf.Pow( adjustedVolume, 1.6 )`

### See Also

Reference  
AudioObject Class  
( Default Namespace ) Namespace
## AudioObject Methods

The **AudioObject** type exposes the following members.

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DestroyAudioObject</td>
<td>Destroys the audio object (using ObjectPoolController if pooling is enabled)</td>
</tr>
<tr>
<td>DoesBelongToCategory</td>
<td>Checks if this AudioObject belongs to a specific category</td>
</tr>
<tr>
<td>FadeIn</td>
<td>Fades-in a playing audio.</td>
</tr>
<tr>
<td>FadeOut(Single)</td>
<td>Starts a fade-out. If the AudioItem mode is is a sequence, the next sub-item will continue to play after the this sub-item is completely faded out</td>
</tr>
<tr>
<td>FadeOut(Single, Single)</td>
<td>Starts a fade-out at a specified time. If the AudioItem mode is is a sequence, the next sub-item will continue to play after the this sub-item is completely faded out</td>
</tr>
<tr>
<td>FinishSequence</td>
<td>Finishes a playing sequence, depending on the AudioItem's loop mode:</td>
</tr>
<tr>
<td></td>
<td>- <strong>LoopSequence</strong>: The sequence will stop after the current item has finished playing</td>
</tr>
</tbody>
</table>
- **PlaySequenceAndLoopLast**: The sequence will stop after the current item has finished playing. If the sequence is during the looping part the looping will stop after the current loop reached its end.
- **IntroLoopOutroSequence**: The sequence will stop after the current item has finished playing. If the sequence is during the looping part the outro will be played and the sequence will stop afterwards.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InverseTransformPitch</td>
<td>Inverse pitch transformation: <em>TransformPitch(Single)</em></td>
</tr>
<tr>
<td>InverseTransformVolume</td>
<td>Inverse volume transformation <em>TransformVolume(Single)</em></td>
</tr>
<tr>
<td>IsPaused</td>
<td>Determines whether the audio clip is paused.</td>
</tr>
<tr>
<td>IsPlaying</td>
<td>Determines if either the primary or the secondary audio clip is playing.</td>
</tr>
<tr>
<td>IsPrimaryPlaying</td>
<td>Determines if the primary audio clip is playing.</td>
</tr>
<tr>
<td>IsSecondaryPlaying</td>
<td>Determines if the secondary audio clip is playing.</td>
</tr>
<tr>
<td>Pause</td>
<td>Pauses the audio clip.</td>
</tr>
<tr>
<td>Pause(Single)</td>
<td>Pauses the audio clip with a</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Play</td>
<td>Plays the audio clip with the specified delay.</td>
</tr>
<tr>
<td>PlayAfter</td>
<td>Plays the specified audio after the current has finished playing.</td>
</tr>
<tr>
<td>PlayNow</td>
<td>Plays the specified audio.</td>
</tr>
<tr>
<td>PlayScheduled</td>
<td>Plays the audio clip at the specified high precision DSP time (see the Unity AudioSettings.dspTime documentation)</td>
</tr>
<tr>
<td>Stop</td>
<td>Stops playing this instance.</td>
</tr>
<tr>
<td>Stop(Single)</td>
<td>Stops a playing audio with fade-out.</td>
</tr>
<tr>
<td>Stop(Single, Single)</td>
<td>Stops a playing audio with fade-out at a specified time.</td>
</tr>
<tr>
<td>SwitchAudioSources</td>
<td>Switches the primary and secondary audio source</td>
</tr>
<tr>
<td>TransformPitch</td>
<td>Transforms the pitch from semitones to a multiplicative factor.</td>
</tr>
<tr>
<td>TransformVolume</td>
<td>Transforms the volume to make it perceptually more intuitive to scale and cross-fade.</td>
</tr>
<tr>
<td>Unpause</td>
<td>Unpauses the audio clip.</td>
</tr>
<tr>
<td>Unpause(Single)</td>
<td>Unpauses the audio clip with a fade-in.</td>
</tr>
</tbody>
</table>
See Also

Reference

AudioObject Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioObjectDestroyAudioObject Method

Destroys the audio object (using ObjectPoolController if pooling is enabled)

Namespace: (Default Namespace)  
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```csharp
public void DestroyAudioObject()
```

See Also

Reference

AudioObject Class  
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioObjectDoesBelongToCategory Method

Checks if this AudioObject belongs to a specific category

**Namespace:** (Default Namespace)  
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```
public bool DoesBelongToCategory(
    string categoryName
)
```

### Parameters

*categoryName*

  Type: System.String  
  The name of the category

### Return Value

Type: Boolean  
true if the category with the specified name or one of its child categories contains the AudioItem the AudioObject belongs to.

### See Also

Reference  
AudioObject Class  
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioObjectFadeIn Method

Fades-in a playing audio.

**Namespace:** (Default Namespace)  
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public void FadeIn(
    float fadeInTime
)
```

### Parameters

- **fadeInTime**  
  Type: `System.Single`  
  The fade time in seconds.

### See Also

**Reference**  
AudioObject Class  
(Default Namespace) Namespace

---

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioObjectFadeOut Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FadeOut(Single)</td>
<td>Starts a fade-out. If the AudioItem mode is a sequence, the next sub-item will continue to play after the this sub-item is completely faded out.</td>
</tr>
<tr>
<td>FadeOut(Single, Single)</td>
<td>Starts a fade-out at a specified time. If the AudioItem mode is a sequence, the next sub-item will continue to play after the this sub-item is completely faded out.</td>
</tr>
</tbody>
</table>

See Also

Reference

AudioObject Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioObjectFadeOut Method (Single)

Starts a fade-out. If the AudioItem mode is is a sequence, the next sub-item will continue to play after the this sub-item is completely faded out.

Namespace:  (Default Namespace)
Assembly:  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```c#
public void FadeOut(float fadeOutLength)
```

Parameters

`fadeOutLength`
Type: **System.Single**
The fade time in seconds. If a negative value is specified, the fade out as specified in the corresponding `FadeOut` is used

Remarks

If the audio is already fading out the requested fade-out is combined with the existing one. This function only fades-out the primary audio source.

See Also

Reference
AudioObject Class
FadeOut Overload
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioObjectFadeOut Method (Single, Single)

Starts a fade-out at a specified time. If the AudioItem mode is a sequence, the next sub-item will continue to play after the this sub-item is completely faded out.

**Namespace:** (Default Namespace)  
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```
public void FadeOut(
    float fadeOutLength,
    float startToFadeTime
)
```

### Parameters

- **fadeOutLength**
  - Type: System.Single
  - The fade time in seconds. If a negative value is specified, the fade out as specified in the corresponding **FadeOut** is used.

- **startToFadeTime**
  - Type: System.Single
  - Fade out starts after **startToFadeTime** seconds have passed.

### Remarks

If the audio is already fading out the requested fade-out is combined with the existing one. This function only fades-out the primary audio source.
See Also

Reference

AudioObject Class
FadeOut Overload
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioObjectFinishSequence Method

Finishes a playing sequence, depending on the AudioItem's loop mode:

- **LoopSequence**: The sequence will stop after the current item has finished playing.
- **PlaySequenceAndLoopLast**: The sequence will stop after the current item has finished playing. If the sequence is during the looping part the looping will stop after the current loop reached its end.
- **IntroLoopOutroSequence**: The sequence will stop after the current item has finished playing. If the sequence is during the looping part the outro will be played and the sequence will stop afterwards.

**Namespace**: (Default Namespace)
**Assembly**: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```csharp
public void FinishSequence()
```

Remarks

Has no effect if the audio is not in a sequence loop mode.

See Also

Reference
AudioObject Class
(Default Namespace) Namespace
AudioObjectInverseTransformPitch Method

Inverse pitch transformation: TransformPitch(Single)

**Namespace:**  (Default Namespace)
**Assembly:**  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```c#
public static float InverseTransformPitch(
    float pitch
)
```

### Parameters

**pitch**
- Type: System.Single
- The transformed pitch

### Return Value

- Type: Single
- The pitch shift in semitones

### See Also

Reference
- AudioObject Class
- (Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioObjectInverseTransformVolume Method

Inverse volume transformation TransformVolume(Single)

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```
public static float InverseTransformVolume(
    float volume
)
```

Parameters

volume
  Type: System.Single
  The volume to inverse-transform.

Return Value
  Type: Single
  The inverse-transformed volume

See Also

Reference
  AudioObject Class
  (Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioObjectIsPaused Method

Determines whether the audio clip is paused.

**Namespace:**  (Default Namespace)

**Assembly:**  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public bool IsPaused(
    bool returnTrueIfStillFadingOut = true
)
```

### Parameters

- **returnTrueIfStillFadingOut (Optional)**
  - Type: `System.Boolean`
  - If `true` the function will return `true` even if the item is still fading out due to a Pause request with a fade-out.

### Return Value

- Type: `Boolean`
  - `true` if paused; otherwise, `false`.

### See Also

- Reference:
  - `AudioObject Class`
  - (Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioObjectIsPlaying Method

Determines if either the primary or the secondary audio clip is playing.

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```c#
public bool IsPlaying()
```

### Return Value

Type: **Boolean**
- `true` if the audio clip is playing; otherwise, `false`.

### See Also

Reference
- **AudioObject Class**
- (Default Namespace) **Namespace**

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to [ClockStone Support Email](mailto:ClockStoneSupportEmail)
AudioObjectIsPrimaryPlaying Method

Determines if the primary audio clip is playing.

**Namespace:** (Default Namespace)

**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```c#
public bool IsPrimaryPlaying()
```

### Return Value

Type: Boolean

true if the audio clip is playing; otherwise, false.

### See Also

Reference

AudioObject Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioObject.IsSecondaryPlaying Method

Determines if the secondary audio clip is playing.

**Namespace:** (Default Namespace)

**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```c#
public bool IsSecondaryPlaying()
```

### Return Value

**Type:** Boolean

- **true** if the audio clip is playing; otherwise, **false**.

### See Also

**Reference**

AudioObject Class

(Default Namespace) Namespace
AudioObjectPause Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pause</td>
<td>Pauses the audio clip.</td>
</tr>
<tr>
<td>Pause(Single)</td>
<td>Pauses the audio clip with a fade-out.</td>
</tr>
</tbody>
</table>

See Also

Reference
AudioObject Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
ClockStone Audio Toolkit for Unity - Documentation
AudioObjectPause Method

Pauses the audio clip.

**Namespace:**  (Default Namespace)
**Assembly:**  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public void Pause()
```

### See Also

Reference
- AudioObject Class
- Pause Overload
  (Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioObjectPause Method (Single)

Pauses the audio clip with a fade-out.

**Namespace:**  (Default Namespace)  
**Assembly:**  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public void Pause(
    float fadeOutTime
)
```

**Parameters**

- **fadeOutTime**  
  Type: System.Single  
  The fade-out time in seconds.

### See Also

**Reference**  
AudioObject Class  
Pause Overload  
(Default Namespace) Namespace
AudioObject Play Method

Plays the audio clip with the specified delay.

**Namespace:**  (Default Namespace)

**Assembly:**  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

⚠️ Syntax

```csharp
public void Play(
    float delay = 0f
)
```

**Parameters**

- **delay**(Optional)
  - Type:  System.Single
  - The delay [Default=0].

⚠️ See Also

- Reference
  - AudioObject Class
  - (Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioObject PlayAfter Method

Plays the specified audio after the current has finished playing

**Namespace:** (Default Namespace)

**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public void PlayAfter(
    string audioID,
    double deltaDspTime = 0,
    float volume = 1f,
    float startTime = 0f
)
```

### Parameters

**audioID**
- Type: System.String
- The audioID to be played.

**deltaDspTime (Optional)**
- Type: System.Double
- Optional delta time (high precision DSP time), Default = 0.

**volume (Optional)**
- Type: System.Single
- The volume [Default = 0].

**startTime (Optional)**
- Type: System.Single
- The start time [Default = 0].

### See Also

Reference
AudioObject Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioObjectPlayNow Method

Plays the specified audio.

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

## Syntax

```csharp
public void PlayNow(
    string audioID,
    float delay = 0f,
    float volume = 1f,
    float startTime = 0f
)
```

### Parameters

- **audioID**
  - Type: `System.String`
  - The audioID to be played.

- **delay (Optional)**
  - Type: `System.Single`
  - Start playing after this amount of seconds [Default = 0].

- **volume (Optional)**
  - Type: `System.Single`
  - The volume [Default = 0].

- **startTime (Optional)**
  - Type: `System.Single`
  - The start time [Default = 0].

## Remarks

Does not stop the secondary audio source (if playing). See
SwitchAudioSources.

See Also

Reference
AudioObject Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioObject PlayScheduled Method

Plays the audio clip at the specified high precision DSP time (see the Unity AudioSettings.dspTime documentation)

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```csharp
public void PlayScheduled(
    double dspTime
)
```

Parameters

`dspTime`

Type: `SystemDouble`

The high precision DSP time.

See Also

Reference

AudioObject Class
( Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioObjectStop Method

⚠️ Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stop</td>
<td>Stops playing this instance.</td>
</tr>
<tr>
<td>Stop(Single)</td>
<td>Stops a playing audio with fade-out.</td>
</tr>
<tr>
<td>Stop(Single, Single)</td>
<td>Stops a playing audio with fade-out at a specified time.</td>
</tr>
</tbody>
</table>

See Also

Reference

AudioObject Class
(Default Namespace) Namespace
AudioObjectStop Method

Stops playing this instance.

**Namespace:** (Default Namespace)  
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public void Stop()
```

### Remarks

Uses fade out as specified in the corresponding **FadeOut**.

### See Also

Reference  
- AudioObject Class
- Stop Overload  
  (Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioObjectStop Method (Single)

Stops a playing audio with fade-out.

**Namespace:**  (Default Namespace)

**Assembly:**  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public void Stop(
    float fadeOutLength
)
```

### Parameters

**fadeOutLength**

Type: System.Single

The fade time in seconds. If a negative value is specified, the fade out as specified in the corresponding **FadeOut** is used

### See Also

Reference

AudioObject Class
Stop Overload

( Default Namespace ) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioObjectStop Method (Single, Single)

Stops a playing audio with fade-out at a specified time.

**Namespace:**  (Default Namespace)  
**Assembly:**  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```c#
public void Stop(
    float fadeOutLength,
    float startToFadeTime
)
```

### Parameters

- **fadeOutLength**
  - Type: System.Single
  - The fade time in seconds. If a negative value is specified, the fade out as specified in the corresponding **FadeOut** is used

- **startToFadeTime**
  - Type: System.Single
  - Fade out starts after **startToFadeTime** seconds have passed

### Remarks

If the audio is already fading out the requested fade-out is combined with the existing one.

### See Also

Reference
AudioObjectSwitchAudioSources Method

Switches the primary and secondary audio source

**Namespace:** (Default Namespace)  
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```csharp
public void SwitchAudioSources()
```

Remarks

This way a single AudioObject can play two audio clips at the same time. You can use it e.g. to cross fade between two audios using the same AudioObject.

Examples

```csharp
playingAudioObject.FadeOut(3);
playingAudioObject.SwitchAudioSources();
playingAudioObject.PlayNow("otherAudioID");
playingAudioObject.FadeIn(3);
```

See Also

Reference  
AudioObject Class  
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioObjectTransformPitch Method

Transforms the pitch from semitones to a multiplicative factor

**Namespace:**  (Default Namespace)
**Assembly:**  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public static float TransformPitch(
    float pitchSemiTones
)
```

### Parameters

- **pitchSemiTones**
  - Type: `System.Single`
  - The pitch shift in semitones to transform.

### Return Value

- Type: `Single`
- The transformed pitch = \( \text{Pow}(2, \frac{\text{pitch}}{12}) \)

### See Also

- Reference
- AudioObject Class
  - (Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioObjectTransformVolume

Method

Transforms the volume to make it perceptually more intuitive to scale and cross-fade.

**Namespace:** (Default Namespace)

**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

**Syntax**

```csharp
public static float TransformVolume(
    float volume
)
```

**Parameters**

`volume`

Type: `System.Single`

The volume to transform.

**Return Value**

Type: `Single`

The transformed volume = `Pow(volume, 1.6)`

**See Also**

Reference

AudioObject Class

(Default Namespace) Namespace

---

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioObjectUnpause Method

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unpause</td>
<td>Unpauses the audio clip.</td>
</tr>
<tr>
<td>Unpause(Single)</td>
<td>Unpauses the audio clip with a fade-in.</td>
</tr>
</tbody>
</table>

See Also

Reference
AudioObject Class
(Default Namespace) Namespace
AudioObjectUnpause Method

Unpauses the audio clip.

**Namespace:**  (Default Namespace)
**Assembly:**  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public void Unpause()
```

### See Also

Reference
- AudioObject Class
- Unpause Overload
- (Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioObjectUnpause Method (Single)

Unpauses the audio clip with a fade-in.

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public void Unpause(
    float fadeInTime
)
```

### Parameters

- **fadeInTime**
  - Type: `System.Single`
  - The fade-in time in seconds.

### See Also

- Reference
  - AudioObject Class
  - Unpause Overload
  - (Default Namespace) Namespace

---

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioObjectAudioEventDelegate Delegate

The audio event delegate type.

**Namespace:** (Default Namespace)  
**Assembly:**  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public delegate void AudioEventDelegate(
    AudioObject audioObject
)
```

### Parameters

- **audioObject**  
  Type: (Default Namespace)AudioObject

### See Also

- Reference  
  (Default Namespace) Namespace

---

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioPickSubItemMode Enumeration

Used by AudioItem to determine which AudioSubItem is chosen.

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

C#

```csharp
public enum AudioPickSubItemMode
```

Members

<table>
<thead>
<tr>
<th>Member name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disabled</td>
<td>0</td>
<td>disables playback</td>
</tr>
<tr>
<td>Random</td>
<td>1</td>
<td>chooses a random subitem with probability in proportion to Probability</td>
</tr>
<tr>
<td>RandomNotSameTwice</td>
<td>2</td>
<td>chooses a random subitem with probability in proportion to Probability makes sure it is not played twice in a row (if possible)</td>
</tr>
<tr>
<td>Sequence</td>
<td>3</td>
<td>chooses the subitems in a sequence</td>
</tr>
</tbody>
</table>


<table>
<thead>
<tr>
<th>Sequence</th>
<th>Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SequenceWithRandomStart</td>
<td>4</td>
<td>chooses the subitems in a sequence starting with a random subitem</td>
</tr>
<tr>
<td>AllSimultaneously</td>
<td>5</td>
<td>chooses all subitems at the same time</td>
</tr>
<tr>
<td>TwoSimultaneously</td>
<td>6</td>
<td>chooses two different subitems at the same time (if possible)</td>
</tr>
<tr>
<td>StartLoopSequenceWithFirst</td>
<td>7</td>
<td>always chooses the first subitem. Intended to be used with AudioItemLoopMode</td>
</tr>
<tr>
<td>RandomNotSameTwiceOddsEvens</td>
<td>8</td>
<td>Same as RandomNotSameTwice but only picks from odds or evens switching every time. Useful for footsteps left/right</td>
</tr>
</tbody>
</table>

See Also

Reference
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
**AudioSubItem Class**

An AudioSubItem represents a specific Unity audio clip.

### Inheritance Hierarchy

System\Object (Default Namespace)\AudioSubItem

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
[SerializableAttribute]
public class AudioSubItem
```

The AudioSubItem type exposes the following members.

### Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![audio] AudioSubItem</td>
<td>Initializes a new instance of the AudioSubItem class</td>
</tr>
<tr>
<td>![audio] AudioSubItem(AudioSubItem, AudioItem)</td>
<td>Copy constructor</td>
</tr>
</tbody>
</table>

### Properties
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>item</td>
<td>the <strong>AudioItem</strong> the sub-item belongs to.</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ToString</strong></td>
<td>Returns the name of the audio clip for debugging. (Overrides <strong>ObjectToString</strong>.)</td>
</tr>
</tbody>
</table>

### Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Clip</strong></td>
<td>Specifies the <strong>AudioClip</strong> to be played in case of the <strong>Item</strong> mode.</td>
</tr>
<tr>
<td><strong>ClipStartTime</strong></td>
<td>Offsets the the audio clip start time (in seconds).</td>
</tr>
<tr>
<td><strong>ClipStopTime</strong></td>
<td>Ends playing the audio at this time (in seconds).</td>
</tr>
<tr>
<td><strong>Delay</strong></td>
<td>Defers the playback of the audio sub-item for <strong>Delay</strong> seconds.</td>
</tr>
<tr>
<td><strong>DisableOtherSubitems</strong></td>
<td>If enabled all other subitems which do not have this option enabled will not be played. Useful</td>
</tr>
</tbody>
</table>
for testing specific subitems within a large list of subitems.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FadeIn</strong></td>
<td>Automatic fade-in in seconds</td>
</tr>
<tr>
<td><strong>FadeOut</strong></td>
<td>Automatic fade-out in seconds</td>
</tr>
<tr>
<td><strong>individualSettings</strong></td>
<td>List of attribute names that have individual settings, i.e. that are not inherited by the parent AudioItem</td>
</tr>
<tr>
<td><strong>ItemModeAudioID</strong></td>
<td>Specifies the audioID to be played in case of the Item mode</td>
</tr>
<tr>
<td><strong>Pan2D</strong></td>
<td>Alters the pan: -1..left, +1..right</td>
</tr>
<tr>
<td><strong>PitchShift</strong></td>
<td>Alters the pitch in units of semitones (thus 12 = twice the speed)</td>
</tr>
<tr>
<td><strong>Probability</strong></td>
<td>If multiple sub-items are defined within an audio item, the specific audio clip is chosen with a probability in proportion to the Probability value.</td>
</tr>
<tr>
<td><strong>RandomDelay</strong></td>
<td>Randomly adds a delay between 0 and RandomDelay</td>
</tr>
<tr>
<td><strong>RandomPitch</strong></td>
<td>Randomly shifts the pitch</td>
</tr>
</tbody>
</table>
in units of semitones (thus 12 = twice the speed)

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RandomStartPosition</td>
<td>Starts playing at a random position.</td>
</tr>
<tr>
<td>RandomVolume</td>
<td>Randomly shifts the volume +/- this value</td>
</tr>
<tr>
<td>SubItemType</td>
<td>Specifies the type of this AudioSubItem</td>
</tr>
<tr>
<td>Volume</td>
<td>The volume applied to the audio sub-item.</td>
</tr>
</tbody>
</table>

**Remarks**

Add your AudioSubItem to an AudioItem using the Unity inspector.

**See Also**

Reference

(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioSubItem Constructor

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AudioSubItem</td>
<td>Initializes a new instance of the AudioSubItem class</td>
</tr>
<tr>
<td>AudioSubItem(AudioSubItem, AudioItem)</td>
<td>Copy constructor</td>
</tr>
</tbody>
</table>

See Also

Reference
AudioSubItem Class (Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioSubItem Constructor

Initializes a new instance of the AudioSubItem class

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```csharp
public AudioSubItem()
```

See Also

Reference
AudioSubItem Class
AudioSubItem Overload
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioSubItem Constructor (AudioSubItem, AudioItem)

Copy constructor

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public AudioSubItem(
    AudioSubItem orig,
    AudioItem item
)
```

### Parameters

**orig**

Type: (Default Namespace)AudioSubItem

[Missing <param name="orig"/> documentation for "M:AudioSubItem.ctor(AudioSubItem,AudioItem)"

**item**

Type: (Default Namespace)AudioItem

[Missing <param name="item"/> documentation for "M:AudioSubItem.ctor(AudioSubItem,AudioItem)"

### See Also

Reference
AudioSubItem Class
AudioSubItem Overload
(Default Namespace) Namespace
AudioSubItem Properties

The AudioSubItem type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>item</td>
<td>the AudioItem the sub-item belongs to.</td>
</tr>
</tbody>
</table>

See Also

Reference

AudioSubItem Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioSubItem Property

the **AudioItem** the sub-item belongs to.

**Namespace:** (Default Namespace)

**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```c#
public AudioItem item { get; }
```

### Property Value

Type: **AudioItem**

### See Also

**Reference**

AudioSubItem Class

(Default Namespace) Namespace
AudioSubItem Methods

The `AudioSubItem` type exposes the following members.

## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ToString</code></td>
<td>Returns the name of the audio clip for debugging. (Overrides <code>ObjectToString</code>.)</td>
</tr>
</tbody>
</table>

See Also

Reference

AudioSubItem Class

(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioSubItemToString Method

Returns the name of the audio clip for debugging.

Namespace:  (Default Namespace)
Assembly:  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```csharp
public override string ToString()
```

Return Value

Type:  String
The debug output string.

See Also

Reference
AudioSubItem Class
(Default Namespace) Namespace
AudioSubItem Fields

The **AudioSubItem** type exposes the following members.

### Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clip</td>
<td>Specifies the <strong>AudioClip</strong> to be played in case of the <strong>Item</strong> mode.</td>
</tr>
<tr>
<td>ClipStartTime</td>
<td>Offsets the audio clip start time (in seconds).</td>
</tr>
<tr>
<td>ClipStopTime</td>
<td>Ends playing the audio at this time (in seconds).</td>
</tr>
<tr>
<td>Delay</td>
<td>Defers the playback of the audio sub-item for <strong>Delay</strong> seconds.</td>
</tr>
<tr>
<td>DisableOtherSubitems</td>
<td>If enabled all other subitems which do not have this option enabled will not be played. Useful for testing specific subitems within a large list of subitems.</td>
</tr>
<tr>
<td>FadeIn</td>
<td>Automatic fade-in in seconds</td>
</tr>
<tr>
<td>FadeOut</td>
<td>Automatic fade-out in seconds</td>
</tr>
<tr>
<td>IndividualSettings</td>
<td>List of attribute names that have individual settings, i.e. that are not inherited by the parent AudioItem</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ItemModeAudioID</td>
<td>Specifies the <strong>audioID</strong> to be played in case of the Item mode</td>
</tr>
<tr>
<td>Pan2D</td>
<td>Alters the pan: -1..left, +1..right</td>
</tr>
<tr>
<td>PitchShift</td>
<td>Alters the pitch in units of semitones (thus 12 = twice the speed)</td>
</tr>
<tr>
<td>Probability</td>
<td>If multiple sub-items are defined within an audio item, the specific audio clip is chosen with a probability in proportion to the <strong>Probability</strong> value.</td>
</tr>
<tr>
<td>RandomDelay</td>
<td>Randomly adds a delay between 0 and RandomDelay</td>
</tr>
<tr>
<td>RandomPitch</td>
<td>Randomly shifts the pitch in units of semitones (thus 12 = twice the speed)</td>
</tr>
<tr>
<td>RandomStartPosition</td>
<td>Starts playing at a random position.</td>
</tr>
<tr>
<td>RandomVolume</td>
<td>Randomly shifts the volume +/- this value</td>
</tr>
<tr>
<td>SubItemType</td>
<td>Specifies the type of this AudioSubItem</td>
</tr>
<tr>
<td>-------------</td>
<td>----------------------------------------</td>
</tr>
</tbody>
</table>

| Volume | The volume applied to the audio sub-item. |

See Also

Reference
AudioSubItem Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioSubItemClip Field

Specifies the **AudioClip** to be played in case of the **Item** mode.

**Namespace:** (Default Namespace)  
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public AudioClip Clip
```

### Field Value

**Type:** **AudioClip**

### See Also

- Reference  
  AudioSubItem Class  
  (Default Namespace) Namespace

---

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioSubItemClipStart-Time Field

Offsets the audio clip start time (in seconds).

**Namespace:** (Default Namespace)  
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

C#

```csharp
public float ClipStartTime
```

### Field Value

Type: Single

### Remarks

Does not work with looping.

### See Also

Reference  
AudioSubItem Class  
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH  
Send comments on this topic to ClockStone Support Email
AudioSubItemClipStopTime Field

Ends playing the audio at this time (in seconds).

**Namespace:**  (Default Namespace)
**Assembly:**  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```
public float ClipStopTime
```

### Field Value

Type: **Single**

### Remarks

Can be used as a workaround for an unknown clip length (e.g. for tracker files)

### See Also

Reference

AudioSubItem Class

(Default Namespace) Namespace
AudioSubItemDelay Field

Defers the playback of the audio sub-item for Delay seconds.

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public float Delay
```

**Field Value**

Type: Single

### See Also

**Reference**
- AudioSubItem Class
- (Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioSubItemDisableOtherSubitems Field

If enabled all other subitems which do not have this option enabled will not be played. Useful for testing specific subitems within a large list of subitems.

**Namespace:** (Default Namespace)  
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

![Syntax](C#)

```c#
public bool DisableOtherSubitems
```

Field Value  
Type: Boolean

**See Also**

Reference  
AudioSubItem Class  
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH  
Send comments on this topic to ClockStone Support Email
AudioSubItemFadeIn Field

Automatic fade-in in seconds

**Namespace:**  (Default Namespace)  
**Assembly:**  AudioToolkit (in AudioToolkit.dll)  Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public float FadeIn
```

### Field Value

**Type:**  Single

### See Also

**Reference**  
AudioSubItem Class  
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioSubItemFadeOut Field

Automatic fade-out in seconds

**Namespace:**  (Default Namespace)  
**Assembly:**  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```
public float FadeOut
```

Field Value  
Type:  Single

### See Also

Reference  
AudioSubItem Class  
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioSubItem.individualSettings Field

List of attribute names that have individual settings, i.e. that are not inherited by the parent AudioItem

**Namespace:** (Default Namespace)  
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public List<string> individualSettings
```

### Field Value

Type: List<String>

### See Also

Reference
- AudioSubItem Class  
  (Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
**AudioSubItem.ItemModeAudioID Field**

Specifies the `audioID` to be played in case of the `Item` mode

**Namespace:** (Default Namespace)  
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

**Syntax**

```csharp
public string ItemModeAudioID
```

**Field Value**  
Type: `String`

**See Also**

Reference  
AudioSubItem Class  
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH  
Send comments on this topic to ClockStone Support Email
AudioSubItemPan2D Field

Alters the pan: -1..left, +1..right

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public float Pan2D
```

Field Value
Type: **Single**

### See Also

**Reference**
AudioSubItem Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioSubItemPitchShift Field

Alters the pitch in units of semitones (thus 12 = twice the speed)

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```csharp
public float PitchShift
```

Field Value

Type: Single

See Also

Reference

AudioSubItem Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioSubItemProbability Field

If multiple sub-items are defined within an audio item, the specific audio clip is chosen with a probability in proportion to the Probability value.

Namespace:  (Default Namespace)
Assembly:  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

C#

```csharp
public float Probability
```

Field Value
Type: Single

See Also

Reference
AudioSubItem Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
**AudioSubItemRandomDelay Field**

Randomly adds a delay between 0 and RandomDelay

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

## Syntax

```csharp
public float RandomDelay
```

## Field Value

Type: Single

## See Also

Reference

AudioSubItem Class
(Default Namespace) Namespace
AudioSubItemRandomPitch Field

Randomly shifts the pitch in units of semitones (thus 12 = twice the speed)

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```c#
public float RandomPitch
```

### Field Value

Type: Single

### See Also

**Reference**
- AudioSubItem Class
- (Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioSubItemRandomStartPosition

Field

Starts playing at a random position.

**Namespace:**  (Default Namespace)

**Assembly:**  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```c#
public bool RandomStartPosition
```

### Field Value

Type:  **Boolean**

### Remarks

Useful for audio loops.

### See Also

**Reference**

AudioSubItem Class

( Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioSubItemRandomVolume

Field

Randomly shifts the volume +/- this value

**Namespace:**  (Default Namespace)

**Assembly:**  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```c#
public float RandomVolume
```

Field Value

Type:  **Single**

### See Also

**Reference**

AudioSubItem Class

(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
AudioSubItemSubItemTypetype Field

Specifies the type of this AudioSubItem

**Namespace:**  (Default Namespace)
**Assembly:**  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```
public AudioSubItemSubItemTypetype
```

Field Value
Type: AudioSubItemSubItemTypetype

### See Also

Reference
AudioSubItem Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioSubItemVolume Field

The volume applied to the audio sub-item.

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public float Volume
```

### Field Value

Type: **Single**

### See Also

**Reference**
AudioSubItem Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
AudioSubItemType Enumeration

The type of an AudioSubItem

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

C# public enum AudioSubItemType

Members

<table>
<thead>
<tr>
<th>Member name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clip</td>
<td>0</td>
<td>The AudioSubItem plays an AudioClip</td>
</tr>
<tr>
<td>Item</td>
<td>1</td>
<td>The AudioSubItem plays an AudioItem</td>
</tr>
</tbody>
</table>

See Also

Reference (Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
IRegisteredComponent Interface

[Missing <summary> documentation for "T:IRegisteredComponent"]

Namespace:  (Default Namespace)
Assembly:  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```csharp
public interface IRegisteredComponent
```

The IRegisteredComponent type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>🔄 GetRegisteredComponentBaseClassType</td>
<td></td>
</tr>
</tbody>
</table>

See Also

Reference

(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
IRegisteredComponent Methods

The IRegisteredComponent type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetRegisteredComponentBaseClassType</td>
<td></td>
</tr>
</tbody>
</table>

See Also

Reference

IRegisteredComponent Interface
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
IRegisteredComponentGetMethod

[Missing <summary> documentation for "M:IRegisteredComponent.GetRegisteredComponentBaseClassType"]

**Namespace:** (Default Namespace)  
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
Type GetRegisteredComponentBaseClassType()
```

### Return Value

Type: **Type**

[Missing <returns> documentation for "M:IRegisteredComponent.GetRegisteredComponentBaseClassType"]

### See Also

Reference  
IRegisteredComponent Interface  
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH  
Send comments on this topic to ClockStone Support Email
ObjectPoolController Class

A static class used to create and destroy poolable objects.

Inheritance Hierarchy

- System
  - Object
    - (Default Namespace)
      - ObjectPoolController

Namespace:  (Default Namespace)
Assembly:  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```csharp
public static class ObjectPoolController
```

The ObjectPoolController type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![isDuringPreload]</td>
<td>isDuringPreload</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Destroy]</td>
<td>Destroys the specified game object, respectively sets the</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instantiate(GameObject)</td>
<td>Retrieves an instance of the specified prefab. Either returns a new instance or it claims an instance from the pool.</td>
</tr>
<tr>
<td>Instantiate(GameObject, Vector3, Quaternion)</td>
<td>Retrieves an instance of the specified prefab. Either returns a new instance or it claims an instance from the pool.</td>
</tr>
<tr>
<td>InstantiateWithoutPool(GameObject)</td>
<td>Instantiates the specified prefab without using pooling.</td>
</tr>
<tr>
<td>InstantiateWithoutPool(GameObject, Vector3, Quaternion)</td>
<td>Instantiates the specified prefab without using pooling.</td>
</tr>
</tbody>
</table>
| Preload | Preloads as many
instances to the pool so that there are at least as many as specified in `preloadCount`.

Top

Remarks

What is pooling?
GameObject.Instantiate(...) calls are relatively time expensive. If objects of the same type are frequently created and destroyed it is good practice to use object pools, particularly on mobile devices. This can greatly reduce the performance impact for object creation and garbage collection.

How does pooling work?

Instead of actually destroying object instances, they are just set inactive and moved to an object "pool". If a new object is requested it can then simply be pulled from the pool, instead of creating a new instance.

Awake(), Start() and OnDestroy() are called if objects are retrieved from or moved to the pool like they were instantiated and destroyed normally.

Examples

How to set up a prefab for pooling:
1. Add the PoolableObject script component to the prefab to be pooled. You can set the maximum number of objects to be stored in the pool from within the inspector.
2. Replace all `Instantiate( myPrefab )` calls with `ObjectPoolController.Instantiate( myPrefab )`
3. Replace all `Destroy( myObjectInstance )` calls with `ObjectPoolController.Destroy( myObjectInstance )`
Attention: Be aware that:
- All data must get initialized in the Awake() or Start() function
- OnDestroy() will get called a second time once the object really gets destroyed by Unity
- If a poolable objects gets parented to none-poolable object, the
  parent must be destroyed using
  ObjectPoolController.Destroy( ... ) even if it is
  none-poolable itself.
- If you store a reference to a poolable object then this reference
  does not evaluate to null after
  ObjectPoolController.Destroy( ... ) was called like
  other references to Unity objects normally would. This is
  because the object still exists - it is just in the pool. To make
  sure that a stored reference to a poolable object is still valid you
  must use PoolableReferenceT.

See Also
- Reference
  (Default Namespace) Namespace
  (Default Namespace)PoolableObject

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
ObjectPoolController Properties

The `ObjectPoolController` type exposes the following members.

## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>isDuringPreload</td>
<td></td>
</tr>
</tbody>
</table>

## See Also

Reference
- `ObjectPoolController Class (Default Namespace) Namespace`

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to [ClockStone Support Email](mailto:ClockStone.Support@ClockStoneSoftware.com)
ObjectPoolController.isDuringPreload Property

[Missing <summary> documentation for "P:ObjectPoolController.isDuringPreload"]

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```csharp
public static bool isDuringPreload { get; }
```

Property Value
Type: Boolean

See Also

Reference
ObjectPoolController Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
## ObjectPoolController Methods

The `ObjectPoolController` type exposes the following members.

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="Destroy" alt="�行" /> Destroy</td>
<td>Destroys the specified game object, respectively sets the object inactive and adds it to the pool.</td>
</tr>
<tr>
<td><img src="Instantiate(GameObject)" alt="行走" /> Instantiate(GameObject)</td>
<td>Retrieves an instance of the specified prefab. Either returns a new instance or it claims an instance from the pool.</td>
</tr>
<tr>
<td>![行走](Instantiate(GameObject, Vector3, Quaternion)) Instantiate(GameObject, Vector3, Quaternion)</td>
<td>Retrieves an instance of the specified prefab. Either returns a new instance or it claims an instance from the pool.</td>
</tr>
</tbody>
</table>
### InstantiateWithoutPool

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InstantiateWithoutPool(GameObject)</td>
<td>Instantiates the specified prefab without using pooling from the pool.</td>
</tr>
<tr>
<td>InstantiateWithoutPool(GameObject, Vector3, Quaternion)</td>
<td>Instantiates the specified prefab without using pooling from the pool.</td>
</tr>
</tbody>
</table>

### Preload

Preloads as many instances to the pool so that there are at least as many as specified in `preloadCount`.

---

**Top**

**See Also**

**Reference**

ObjectPoolController Class
(Default Namespace) Namespace

---

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
ObjectPoolControllerDestroy Method

Destroys the specified game object, respectively sets the object inactive and adds it to the pool.

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public static void Destroy(
    GameObject obj
)
```

### Parameters

- **obj**
  
  Type: **GameObject**
  The game object.

### Remarks

Can be used on none-poolable objects as well. It is good practice to use `ObjectPoolController.Destroy` whenever you may possibly make your prefab poolable in the future. Must also be used on none-poolable objects with poolable child objects so the poolable child objects are correctly moved to the pool.

### See Also

Reference
**ObjectPoolController Class**
ObjectPoolController Instantiate Method

- **Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![S] Instantiate(GameObject)</td>
<td>Retrieves an instance of the specified prefab. Either returns a new instance or it claims an instance from the pool.</td>
</tr>
<tr>
<td>![S] Instantiate(GameObject, Vector3, Quaternion)</td>
<td>Retrieves an instance of the specified prefab. Either returns a new instance or it claims an instance from the pool.</td>
</tr>
</tbody>
</table>

- **See Also**

Reference  
ObjectPoolController Class  
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
ObjectPoolControllerInstantiate Method (GameObject)

Retrieves an instance of the specified prefab. Either returns a new instance or it claims an instance from the pool.

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public static GameObject Instantiate(
    GameObject prefab
)
```

### Parameters

**prefab**

Type: `GameObject`

The prefab to be instantiated.

### Return Value

Type: `GameObject`

An instance of the prefab.

### Remarks

Can be used on none-poolable objects as well. It is good practice to use `ObjectPoolController.Instantiate` whenever you may possibly make your prefab poolable in the future.

### See Also
Reference

ObjectPoolController Class
Instantiate Overload
(Default Namespace) Namespace
ObjectPoolControllerDestroy(GameObject)

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
ObjectPoolControllerInstantiate Method (GameObject, Vector3, Quaternion)

Retrieves an instance of the specified prefab. Either returns a new instance or it claims an instance from the pool.

Namespace:  (Default Namespace)
Assembly:  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```csharp
public static GameObject Instantiate(
    GameObject prefab,
    Vector3 position,
    Quaternion quaternion
)
```

Parameters

- `prefab`  
  Type: `GameObject`  
  The prefab to be instantiated.

- `position`  
  Type: `Vector3`  
  The position in world coordinates.

- `quaternion`  
  Type: `Quaternion`  
  The rotation quaternion.

Return Value  
Type: `GameObject`
An instance of the prefab.

Remarks

Can be used on none-poolable objects as well. It is good practice to use `ObjectPoolController.Instantiate` whenever you may possibly make your prefab poolable in the future.

See Also

Reference

ObjectPoolController Class
Instantiate Overload
(Default Namespace) Namespace
ObjectPoolControllerDestroy(GameObject)

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
## ObjectPoolController InstantiateWithoutPool Method

### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![S] InstantiateWithoutPool(GameObject)</td>
<td>Instantiates the specified prefab without using pooling from the pool.</td>
</tr>
<tr>
<td>![S] InstantiateWithoutPool(GameObject, Vector3, Quaternion)</td>
<td>Instantiates the specified prefab without using pooling from the pool.</td>
</tr>
</tbody>
</table>

### See Also

Reference
- ObjectPoolController Class (Default Namespace) Namespace
Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
ObjectPoolControllerInstantiateWithoutPool Method (GameObject)

Instantiates the specified prefab without using pooling from the pool.

**Namespace**: (Default Namespace)
**Assembly**: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

**Syntax**

```csharp
public static GameObject InstantiateWithoutPool(
    GameObject prefab
)
```

**Parameters**

`prefab`
- Type: `GameObject`
- The prefab to be instantiated.

**Return Value**
- Type: `GameObject`
- An instance of the prefab.

**Remarks**

If the prefab is poolable, the `PoolableObject` component will be removed. This way no warning is generated that a poolable object was created without pooling.

**See Also**

Reference
ObjectPoolController Class
InstantiateWithoutPool Overload
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
ObjectPoolControllerInstantiateWithoutPool

Method (GameObject, Vector3, Quaternion)

Instantiates the specified prefab without using pooling from the pool.

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

> Syntax

```csharp
public static GameObject InstantiateWithoutPool(
    GameObject prefab,
    Vector3 position,
    Quaternion quaternion
)
```

**Parameters**

- **prefab**
  - Type: **GameObject**
  - The prefab to be instantiated.

- **position**
  - Type: **Vector3**
  - The position in world coordinates.

- **quaternion**
  - Type: **Quaternion**
  - The rotation quaternion.

**Return Value**

- Type: **GameObject**
  - An instance of the prefab.
Remarks

If the prefab is poolable, the `PoolableObject` component will be removed. This way no warning is generated that a poolable object was created without pooling.

See Also

Reference

- `ObjectPoolController Class`
- `InstantiateWithoutPool Overload`  
  `(Default Namespace) Namespace`

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
ObjectPoolController Preload Method

Preloads as many instances to the pool so that there are at least as many as specified in preloadCount.

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public static void Preload(
    GameObject prefab
)
```

### Parameters

`prefab`
- Type: `GameObject`
- The prefab.

### Remarks

Use ObjectPoolController.isDuringPreload to check if an object is preloaded in the `Awake()` function. If the pool already contains at least preloadCount objects, the function does nothing.

### See Also

Reference
- ObjectPoolController Class
- (Default Namespace) Namespace
- PoolableObject preloadCount
Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
Playlist Class

Allows to define a playlist consisting of a list of audio IDs

Inheritance Hierarchy

- SystemObject  (Default Namespace)
- Playlist

Namespace:  (Default Namespace)
Assembly:  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```c#
[SerializableAttribute]
public class Playlist
```

The Playlist type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![ ] Playlist</td>
<td>Initializes a new instance of the Playlist class</td>
</tr>
<tr>
<td>![ ] Playlist(String,String)</td>
<td>Initializes a new instance of the Playlist class</td>
</tr>
</tbody>
</table>

Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![ ]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>name</td>
</tr>
<tr>
<td>----------</td>
<td>-----------------------</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>playlistItems</td>
</tr>
</tbody>
</table>

**See Also**

Reference
(DefaultValue) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
Playlist Constructor

# Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Playlist</td>
<td>Initializes a new instance of the Playlist class</td>
</tr>
<tr>
<td>Playlist(String, String)</td>
<td>Initializes a new instance of the Playlist class</td>
</tr>
</tbody>
</table>

See Also

Reference

Playlist Class
( Default Namespace ) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
Playlist Constructor

Initializes a new instance of the Playlist class

**Namespace:** (Default Namespace)  
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public Playlist()
```

### See Also

**Reference**
- Playlist Class
- Playlist Overload
- (Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
Playlist Constructor (String, String)

Initializes a new instance of the Playlist class

**Namespace:** (Default Namespace)  
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public Playlist(
    string name,
    string[] playlistItems
)
```

### Parameters

- **name**
  - Type: SystemString  
  - [Missing `<param name="name"/>` documentation for "M:Playlist.#ctor(System.String,System.String[])""]

- **playlistItems**
  - Type: SystemString  
  - [Missing `<param name="playlistItems"/>` documentation for "M:Playlist.#ctor(System.String,System.String[])""]

### See Also

- Reference  
  - Playlist Class  
  - Playlist Overload  
  - (Default Namespace) Namespace
Playlist Fields

The Playlist type exposes the following members.

Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Name of the playlist as used for e.g. by PlayMusicPlaylist(String)</td>
</tr>
<tr>
<td>playlistItems</td>
<td>An array of audio IDs defining the playlist</td>
</tr>
</tbody>
</table>

See Also

Reference

Playlist Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
Playlistname Field

Name of the playlist as used for e.g. by PlayMusicPlaylist(String)

**Namespace:** (Default Namespace)

**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```c#
public string name
```

**Field Value**

Type: String

### See Also

Reference

Playlist Class

(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
PlaylistplaylistItems Field

An array of audio IDs defining the playlist

**Namespace:**  (Default Namespace)

**Assembly:**  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public string[] playlistItems
```

Field Value

Type:  **String**

### See Also

**Reference**

Playlist Class  
(Defaut Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
PoolableObject Class

Add this component to your prefab to make it poolable.

▶ Inheritance Hierarchy

```
System
    Object
    Component
    Behaviour
    MonoBehaviour
        (Default Namespace)PoolableObject
```

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

▶ Syntax

```c#
public class PoolableObject : MonoBehaviour
```

The `PoolableObject` type exposes the following members.

▶ Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>✩ PoolableObject</td>
<td>Initializes a new instance of the <code>PoolableObject</code> class</td>
</tr>
</tbody>
</table>

▶ Methods
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DeactivateAllPoolableObjectsOfMyKind</strong></td>
<td>Moves all poolable objects of this kind (instantiated from the same prefab as this instance) back to the pool.</td>
</tr>
<tr>
<td><strong>GetAllPoolableObjectsOfMyKind</strong></td>
<td>Retrieves an array of all poolable objects of this kind (instantiated from the same prefab as this instance).</td>
</tr>
<tr>
<td><strong>GetSerialNumber</strong></td>
<td>Gets the object's pool serial number. Each object has a unique serial number. Can be useful for debugging purposes.</td>
</tr>
</tbody>
</table>
- **GetUsageCount**: Gets the usage counter which gets increased each time an object is re-used from the pool.

- **IsDeactivated**: Checks if the object is deactivated and in the pool.

### Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>doNotDestroyOnLoad</td>
<td>If enabled the object will not get destroyed if a new scene is loaded.</td>
</tr>
<tr>
<td>maxPoolSize</td>
<td>The maximum number of instances of this prefab to get stored in the pool.</td>
</tr>
<tr>
<td>preloadCount</td>
<td>This number of instances will be preloaded to the pool.</td>
</tr>
<tr>
<td>sendAwakeStartOnDestroyMessage</td>
<td>If enabled Awake(), Start(), and OnDestroy() messages are sent to the poolable object whenever Destroy(GameObject) is called.</td>
</tr>
</tbody>
</table>
This way it is simulated that the object really gets instantiated respectively destroyed.

If enabled a **OnPoolable** and **OnPoolable** message is sent instance if the object is activated respectively deactivated by the ObjectPoolController.

The following messages are sent to a poolable object:
- **Awake** and **OnDestroy** whenever a poolable object is activated or deactivated from the pool. This way the same behaviour is simulated as if the object was instantiated respectively destroyed. These messages are only sent when **sendAwakeStartOnDestroyMessage** is enabled.
- **OnPoolableInstanceAwake** and **OnPoolableInstanceDestroy** when the object was actually instantiated respectively destroyed. Because of current Unity limitations **OnPoolableInstanceDestroy** does not work on Flash!
- **OnPoolableObjectActivated** and **OnPoolableObjectDeactivated** whenever a poolable object is activated or deactivated from the pool. These messages are only sent when **sendPoolableActivateDeactivateMessages** is enabled.

See Also

Reference
PoolableObject Constructor

Initializes a new instance of the PoolableObject class

Namespace:  (Default Namespace)
Assembly:  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```csharp
public PoolableObject()
```

See Also

Reference
PoolableObject Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
PoolableObject Methods

The **PoolableObject** type exposes the following members.

## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt=" " /> DeactivateAllPoolableObjectsOfMyKind</td>
<td>Moves all poolable objects of this kind (instantiated from the same prefab as this instance) back to the pool.</td>
</tr>
<tr>
<td><img src="image" alt=" " /> GetAllPoolableObjectsOfMyKind</td>
<td>Retrieves an array of all poolable objects of this kind (instantiated from the same prefab as this instance).</td>
</tr>
<tr>
<td><img src="image" alt=" " /> GetSerialNumber</td>
<td>Gets the object's pool serial</td>
</tr>
</tbody>
</table>
Each object has a unique serial number. Can be useful for debugging purposes.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetUsageCount</td>
<td>Gets the usage counter which gets increased each time an object is re-used from the pool.</td>
</tr>
<tr>
<td>IsDeactivated</td>
<td>Checks if the object is deactivated and in the pool.</td>
</tr>
</tbody>
</table>

**See Also**

Reference

PoolableObject Class (Default Namespace) Namespace
PoolableObjectDeactivateAllPoolableObjectsOfMyKind Method

Moves all poolable objects of this kind (instantiated from the same prefab as this instance) back to the pool.

Namespace:  (Default Namespace)  
Assembly:  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```
public int DeactivateAllPoolableObjectsOfMyKind()
```

Return Value

Type:  Int32  
The number of instances deactivated and moved back to its pool.

See Also

Reference  
PoolableObject Class  
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
PoolableObject.GetAllPoolableObjectsOfMyKind Method

Retrieves an array of all poolable objects of this kind (instantiated from the same prefab as this instance).

**Namespace:** (Default Namespace)  
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public PoolableObject[] GetAllPoolableObjectsOfMyKind(bool includeInactiveObjects)
```

### Parameters

- `includeInactiveObjects`  
  Type: System.Boolean  
  If enabled, the returned array will also include the inactive objects in the pool.

### Return Value

Type: PoolableObject  
The array of poolable objects.

### See Also

- **Reference**  
  PoolableObject Class  
  (Default Namespace) Namespace
PoolableObjectGetSerialNumber Method

Gets the object's pool serial number. Each object has a unique serial number. Can be useful for debugging purposes.

**Namespace:**  (Default Namespace)
**Assembly:**  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

**Syntax**

```c#
public int GetSerialNumber()
```

**Return Value**

Type: **Int32**
The serial number (starting with 1 for each pool).

**See Also**

Reference

PoolableObject Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
PoolableObjectGetUsageCount Method

Gets the usage counter which gets increased each time an object is re-used from the pool.

**Namespace:** (Default Namespace)  
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public int GetUsageCount()
```

### Return Value

Type: **Int32**  
The usage counter

### See Also

Reference  
**PoolableObject Class**  
( Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
PoolableObjectIsDeactivated Method

Checks if the object is deactivated and in the pool.

**Namespace:** (Default Namespace)  
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public bool IsDeactivated()
```

### Return Value

Type: **Boolean**  
**true** if the object is in the pool of deactivated objects, otherwise **false**.

### See Also

Reference  
PoolableObject Class  
( Default Namespace) Namespace
PoolableObject Fields

The PoolableObject type exposes the following members.

### Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>doNotDestroyOnLoad</td>
<td>If enabled the object will not get destroyed if a new scene is loaded.</td>
</tr>
<tr>
<td>maxPoolSize</td>
<td>The maximum number of instances of this prefab to get stored in the pool.</td>
</tr>
<tr>
<td>preloadCount</td>
<td>This number of instances will be preloaded to the pool.</td>
</tr>
<tr>
<td>sendAwakeStartOnDestroyMessage</td>
<td>If enabled Awake(), Start(), and OnDestroy() messages are sent to the poolable object if the object is set active respectively inactive whenever Destroy(GameObject). This way it is simulated that the object really gets instantiated respectively destroyed.</td>
</tr>
<tr>
<td>sendPoolableActivateDeactivateMessages</td>
<td>If enabled a OnPoolableObjectActivated and OnPoolableObjectDeactivated message is sent to the instance if the object is activated respectively deactivated.</td>
</tr>
</tbody>
</table>
See Also

Reference
PoolableObject Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
PoolableObject\ndoNotDestroyOnLoad\n
Field

If enabled the object will not get destroyed if a new scene is loaded

**Namespace:** (Default Namespace)  
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```c#
public bool doNotDestroyOnLoad
```

Field Value  
Type: Boolean

### See Also

Reference  
PoolableObject Class  
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
PoolableObject

maxPoolSize

Field

The maximum number of instances of this prefab to get stored in the pool.

**Namespace:** (Default Namespace)

**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```
public int maxPoolSize
```

### Field Value

Type: Int32

### See Also

Reference

PoolableObject Class

(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
PoolableObject

Field

This number of instances will be preloaded to the pool if `Preload(GameObject)` is called.

Namespace:  (Default Namespace)
Assembly:  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```c#
public int preloadCount
```

Field Value

Type: `Int32`

See Also

Reference

PoolableObject Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
Field

If enabled Awake(), Start(), and OnDestroy() messages are sent to the poolable object if the object is set active respectively inactive whenever Destroy(GameObject) or Instantiate(GameObject) is called. This way it is simulated that the object really gets instantiated respectively destroyed.

**Namespace:**  (Default Namespace)
**Assembly:**  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```c#
public bool sendAwakeStartOnDestroyMessage
```

### Field Value

**Type:**  Boolean

### Remarks

The Start() function is called immediately after Awake() by Instantiate(GameObject) and not next frame. So do not set data after Instantiate<GameObject>() that Start() relies on. In some cases you may not want the Awake(), Start(), and OnDestroy() messages to be sent for performance reasons because it may not be necessary to fully reinitialize a game object each time it is activated from the pool. You can still use the OnPoolableObjectActivated and OnPoolableObjectDeactivated messages to initialize specific data.

### See Also
Reference

PoolableObject Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
**PoolableObject sendPoolableActivateDeactivateMessages Field**

If enabled a `OnPoolableObjectActivated` and `OnPoolableObjectDeactivated` message is sent to the poolable instance if the object is activated respectively deactivated by the `ObjectPoolController`

**Namespace:** (Default Namespace)  
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public bool sendPoolableActivateDeactivateMessage
```

**Field Value**  
Type: Boolean

### See Also

**Reference**  
PoolableObject Class  
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH  
Send comments on this topic to ClockStone Support Email
PoolableReference\textit{T} Class

Auxiliary class to overcome the problem of references to pooled objects that should become \texttt{null} when objects are moved back to the pool after calling \texttt{Destroy(GameObject)}.

\section*{Inheritance Hierarchy}

\begin{itemize}
  \item \texttt{SystemObject} (Default Namespace) \texttt{PoolableReference\textit{T}}
\end{itemize}

\textbf{Namespace:} (Default Namespace)  
\textbf{Assembly:} AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

\section*{Syntax}

\begin{verbatim}
public class PoolableReference\textless T\textgreater 
where T : Component

C#
\end{verbatim}

Type Parameters

\begin{itemize}
  \item \textit{T}  
    \begin{itemize}
      \item A \texttt{UnityEngine\hspace{1pt}Component}
    \end{itemize}
\end{itemize}

The \texttt{PoolableReference\textit{T}} type exposes the following members.

\section*{Constructors}

\begin{table}[h]
\centering
\begin{tabular}{|c|c|}
\hline
\textbf{Name} & \textbf{Description} \\
\hline
\textbullet\hspace{1pt} \texttt{PoolableReference\textit{T}} & Initializes a new instance of the \texttt{PoolableReference\textit{T}} class with a \texttt{null} reference. \\
\hline
\end{tabular}
\end{table}
PoolableReference<T>(T) Initializes a new instance of the PoolableReference class with the specified reference.

PoolableReference<T>(PoolableReference<T>) Initializes a new instance of the PoolableReference class from a given PoolableReference.

## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get</td>
<td>Gets the reference to the script component, or null if the object was already destroyed or moved to the pool.</td>
</tr>
<tr>
<td>Reset</td>
<td>Resets the reference to null.</td>
</tr>
<tr>
<td>Set(T)</td>
<td>Sets the reference to a poolable object with the specified component.</td>
</tr>
<tr>
<td>Set(T, Boolean)</td>
<td>Sets the reference to a poolable object with the specified component.</td>
</tr>
</tbody>
</table>

## Examples

Instead of a normal reference to a script component on a poolable object use

```csharp
MyScriptComponent scriptComponent = PoolableObjectController.Instantiate(prefab).GetComponent<MyScriptComponent>();
```
```javascript
var myReference = new PoolableReference<MyScriptComponent>(
  scriptComponent
);
if (myReference.Get() != null) // will check if
{
  myReference.Get().MyComponentFunction();
}
```

**See Also**

Reference
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
## PoolableReference\text{T} Constructor

### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PoolableReference\text{T}</td>
<td>Initializes a new instance of the \text{PoolableReference} class with a null reference.</td>
</tr>
<tr>
<td>PoolableReference\text{T}(\text{T})</td>
<td>Initializes a new instance of the \text{PoolableReference} class with the specified reference.</td>
</tr>
<tr>
<td>PoolableReference\text{T}(PoolableReference\text{T})</td>
<td>Initializes a new instance of the \text{PoolableReference} class from a given \text{PoolableReference} instance.</td>
</tr>
</tbody>
</table>

### See Also

- Reference
- \text{PoolableReference\text{T} Class (Default Namespace) Namespace}

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
PoolableReferenceT Constructor

Initializes a new instance of the PoolableReferenceT class with a null reference.

**Namespace:** (Default Namespace)

**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

C#

```csharp
public PoolableReference()
```

### See Also

Reference

- PoolableReferenceT Class
- PoolableReferenceT Overload
- (Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
PoolableReference\textit{T} Constructor\textit{(T)}

Initializes a new instance of the \textit{PoolableReference\textit{T}} class with the specified reference.

**Namespace:** (Default Namespace)

**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```c#
public PoolableReference<T>
    (T componentOfPoolableObject)
```

### Parameters

- \textit{componentOfPoolableObject}
  - Type: \textit{T}
  - The referenced component of the poolable object.

### See Also

- Reference
- \textit{PoolableReference\textit{T} Class}
- \textit{PoolableReference\textit{T} Overload}
- (Default Namespace) Namespace
PoolableReference<T> Constructor (PoolableReference<T>)

Initializes a new instance of the PoolableReference<T> class from a given PoolableReference<T>.

**Namespace:** (Default Namespace)

**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

⚠️ Syntax

```csharp
public PoolableReference(
    PoolableReference<T> poolableReference
)
```

**Parameters**

*poolableReference*

Type: (Default Namespace)PoolableReference<T>
The poolable reference.

⚠️ See Also

Reference
PoolableReference<T> Class
PoolableReference<T> Overload
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
PoolableReferenceT Methods

The PoolableReferenceT generic type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get</td>
<td>Gets the reference to the script component, or null if the object was already destroyed or moved to the pool.</td>
</tr>
<tr>
<td>Reset</td>
<td>Resets the reference to null.</td>
</tr>
<tr>
<td>Set(T)</td>
<td>Sets the reference to a poolable object with the specified component.</td>
</tr>
<tr>
<td>Set(T, Boolean)</td>
<td>Sets the reference to a poolable object with the specified component.</td>
</tr>
</tbody>
</table>

Top

See Also

Reference
PoolableReferenceT Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
PoolableReference<T> Get Method

Gets the reference to the script component, or null if the object was already destroyed or moved to the pool.

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```c#
public T Get()
```

Return Value
Type: T
The reference to T or null

See Also

Reference
PoolableReference<T> Class
(Default Namespace) Namespace
PoolableReference<caret>TReset Method

Resets the reference to null.

**Namespace:** (Default Namespace)  
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public void Reset()
```

### See Also

Reference  
PoolableReference<caret>T Class  
(Default Namespace) Namespace
PoolableReference<T> Set Method

### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set(T)</td>
<td>Sets the reference to a poolable object with the specified component.</td>
</tr>
<tr>
<td>Set(T, Boolean)</td>
<td></td>
</tr>
</tbody>
</table>

See Also

Reference

PoolableReference<T> Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
PoolableReference\(T\) Set Method (\(T\))

[Missing <summary> documentation for "M:PoolableReference`1.Set(\(0\))"]

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```
public void Set(
    T componentOfPoolableObject
)
```

### Parameters

- **componentOfPoolableObject**
  - Type: \(T\)
  - [Missing <param name="componentOfPoolableObject"/> documentation for "M:PoolableReference`1.Set(\(0\))"]

### See Also

- Reference
  - PoolableReference\(T\) Class
  - Set Overload
  - (Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
PoolableReference<T> Set Method (T, Boolean)

Sets the reference to a poolable object with the specified component.

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

## Syntax

```csharp
public void Set(
    T componentOfPoolableObject,
    bool allowNonePoolable
)
```

### Parameters

- **componentOfPoolableObject**
  - Type: `T`
  - The component of the poolable object.
- **allowNonePoolable**
  - Type: `System.Boolean`
  - If set to false an error is output if the object does not have the `PoolableObject` component.

## See Also

- Reference
  - `PoolableReference<T> Class`
  - `Set Overload`
    - (Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
RegisteredComponent Class

Derive your MonoBehaviour class from RegisteredComponent and all references to instances of this component will be saved in an internal array. Use `GetAllOfTypeT` to retrieve this array, which is much faster than using Unity's GameObject.FindObjectsOfType() function.

▲ Inheritance Hierarchy

```
System
  Object
  Component
    Behaviour
      MonoBehaviour
        (Default Namespace)RegisteredComponent
        (Default Namespace)AudioObject
```

Namespace:  (Default Namespace)
Assembly:  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

▲ Syntax

```csharp
public abstract class RegisteredComponent : MonoBehaviour, IRegisteredComponent
```

The RegisteredComponent type exposes the following members.

▲ Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>🔄 GetRegisteredComponentBaseClassType</td>
<td></td>
</tr>
</tbody>
</table>

Top
See Also

Reference
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
RegisteredComponent Methods

The RegisteredComponent type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>✪ GetRegisteredComponentBaseClassType</td>
<td></td>
</tr>
</tbody>
</table>

See Also

Reference
RegisteredComponent Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
RegisteredComponentGetRegisteredComponentBaseClassType Method

[Missing <summary> documentation for "M:RegisteredComponent.GetRegisteredComponentBaseClassType"]

Namespace: (Default Namespace)  
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```
public Type GetRegisteredComponentBaseClassType()
```

Return Value

Type: Type  
[Missing <returns> documentation for "M:RegisteredComponent.GetRegisteredComponentBaseClassType"]

Implements

IRegisteredComponentGetRegisteredComponentBaseClassType

See Also

Reference

RegisteredComponent Class  
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
RegisteredComponentController Class

This controller provides fast access to all currently existing RegisteredComponent instances.

Inheritance Hierarchy

- System
  - Object
    - (Default Namespace)RegisteredComponentController

Namespace: (Default Namespace)
Assembly: AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```csharp
public static class RegisteredComponentController
```

The RegisteredComponentController type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetAllOfType&lt;Type&gt;</td>
<td>Retrieves an array of all currently existing instances of the class with type <code>type</code>, (type must be a RegisteredComponent)</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>GetAllOfType&lt;T&gt;</td>
<td>Retrieves an array of all currently existing instances of the class T, where T must be a RegisteredComponent</td>
</tr>
<tr>
<td>InstanceCountOfType&lt;T&gt;</td>
<td>Return the number of all currently existing instances of the class T, where T must be a RegisteredComponent</td>
</tr>
</tbody>
</table>

### Remarks

The function `GetAllOfType<T>` is understood as a replacement for Unity's slow `GameObject.FindObjectsOfType()` function.

### See Also

- Reference
  - (Default Namespace) Namespace
# RegisteredComponentController Methods

The `RegisteredComponentController` type exposes the following members.

## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>GetAllOfType(Type)</code></td>
<td>Retrieves an array of all currently existing instances of the class with type <code>type</code>, (type must be a <code>RegisteredComponent</code>)</td>
</tr>
<tr>
<td><code>GetAllOfTypeT</code></td>
<td>Retrieves an array of all currently existing instances of the class <code>T</code>, where <code>T</code> must be a <code>RegisteredComponent</code></td>
</tr>
<tr>
<td><code>InstanceCountOfTypeT</code></td>
<td>Return the number of all currently existing instances of the class <code>T</code>, where <code>T</code> must be a <code>RegisteredComponent</code></td>
</tr>
</tbody>
</table>

## See Also

- Reference
RegisteredComponentController Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
RegisteredComponentController

GetAllOfType

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetAllOfType&lt;T&gt;</td>
<td>Retrieves an array of all currently existing instances of the class T, where T must be a RegisteredComponent</td>
</tr>
<tr>
<td>GetAllOfType(Type)</td>
<td>Retrieves an array of all currently existing instances of the class with type type, (type must be a RegisteredComponent )</td>
</tr>
</tbody>
</table>

See Also

Reference
RegisteredComponentController Class (Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
RegisteredComponentController GetAllOfType Method

Retrieves an array of all currently existing instances of the class \( T \), where \( T \) must be a RegisteredComponent.

**Namespace:** (Default Namespace)

**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

**Syntax**

```csharp
public static T[] GetAllOfType<T>()
where T : IRegisteredComponent
```

**Type Parameters**

\( T \)

- a class derived from RegisteredComponent

**Return Value**

- Type: \( T \)
- The array of instances.

**See Also**

**Reference**

- RegisteredComponentController Class
- GetAllOfType Overload
- (Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
RegisteredComponentController.GetAllOfType Method (Type)

Retrieves an array of all currently existing instances of the class with type `type`, (type must be a RegisteredComponent)

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public static Object[] GetAllOfType(
    Type type
)
```

### Parameters

- **type**
  - Type: `System.Type`
  - The type of the instances to be retrieved

### Return Value

- **Type:** `Object`
  - The array of instances.

### Remarks

Use this function instead of GetAllOfType<T> if you need Flash compatibility.

### See Also

Reference
RegisteredComponentController

Method

Return the number of all currently existing instances of the class \( T \), where \( T \) must be a \texttt{RegisteredComponent}

\textbf{Namespace:} (Default Namespace)
\textbf{Assembly:} AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

\section{Syntax}

\begin{verbatim}
public static int InstanceCountOfType\langle T\rangle()
where T : IRegisteredComponent
\end{verbatim}

Type Parameters

\( T \)

a class derived from \texttt{RegisteredComponent}

Return Value

Type: \texttt{Int32}

The number of instances.

\section{See Also}

Reference

\texttt{RegisteredComponentController} Class
(\texttt{Default Namespace}) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to \texttt{ClockStone Support Email}
SingletonMonoBehaviour<T> Class

Provides singleton-like access to a unique instance of a MonoBehaviour.

Inheritance Hierarchy

System
  -- Object
    -- Component
      -- Behaviour
        -- MonoBehaviour
          (Default Namespace)SingletonMonoBehaviour<T
          (Default Namespace)AudioController

Namespace:  (Default Namespace)
Assembly:  AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

Syntax

```csharp
public abstract class SingletonMonoBehaviour<T> : MonoBehaviour
    where T : MonoBehaviour
```

Type Parameters

T

Your singleton MonoBehaviour

The SingletonMonoBehaviour<T> type exposes the following members.

Properties
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance</td>
<td>Gets the singleton instance.</td>
</tr>
<tr>
<td>isSingletonObject</td>
<td>must return true if this instance of the object is the singleton. Can be used to allow multiple objects of this type that are &quot;add-ons&quot; to the singleton.</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ActivateSingletonInstance</td>
<td>Activates the singleton instance.</td>
</tr>
<tr>
<td>DoesInstanceExist</td>
<td>Checks if an instance of this MonoBehaviour exists.</td>
</tr>
<tr>
<td>SetSingletonAutoCreate</td>
<td>Sets the object to be instantiated automatically if no instance of the singleton is found.</td>
</tr>
<tr>
<td>SetSingletonType</td>
<td>Only required for Flash builds. If this function is not called by the class deriving from SingletonMonoBehaviour in the constructor the singleton can not be found by GetSingleton(...)</td>
</tr>
</tbody>
</table>
Remarks

Makes sure that an instance is available from other Awake() calls even before the singleton's Awake() was called. (Requires AwakeSingleton() !)

Examples

Derive your own class from SingletonMonoBehaviour.

```csharp
public class MyScriptClass : SingletonMonoBehaviour
{
    public MyScriptClass()
    {
        MyScriptClass.SetSingletonType( typeof( )
    }
    public void MyFunction() { }
    protected override void Awake()
    {
        base.Awake();
    }
    void AwakeSingleton()
    {
        // all initialisation code here. Will get
        // Can get called before Awake() if an ir
        // was called earlier
    }
}
```

access the instance by writing

```csharp
MyScriptClass.Instance.MyFunction();
```
See Also

Reference

(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH

Send comments on this topic to ClockStone Support Email
Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance</td>
<td>Gets the singleton instance.</td>
</tr>
<tr>
<td>isSingletonObject</td>
<td>must return true if this instance of the object is the singleton. Can be used to allow multiple objects of this type that are &quot;add-ons&quot; to the singleton.</td>
</tr>
</tbody>
</table>

See Also

Reference
SingletonMonoBehaviourT Class
(Default Namespace) Namespace
SingletonMonoBehaviour<T> Instance Property

Gets the singleton instance.

**Namespace:** (Default Namespace)  
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

▲ Syntax

```csharp
public static T Instance { get; }
```

Return Value
Type: \(T\)
A reference to the instance if it exists, otherwise \(null\)

▲ Remarks

Outputs an error to the debug log if no instance was found.

▲ See Also

Reference
SingletonMonoBehaviour<T> Class  
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH  
Send comments on this topic to ClockStone Support Email
SingletonMonoBehaviourT\texttt{isSingletonObject} Property

must return true if this instance of the object is the singleton. Can be used to allow multiple objects of this type that are "add-ons" to the singleton.

\textbf{Namespace:} (Default Namespace)  
\textbf{Assembly:} AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

\textbf{Syntax}

```c#
public virtual bool isSingletonObject { get; }
```

Property Value  
Type: Boolean

\textbf{See Also}

Reference  
SingletonMonoBehaviourT Class  
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH  
Send comments on this topic to ClockStone Support Email
The `SingletonMonoBehaviourT` generic type exposes the following members.

## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![s] ActivateSingletonInstance</td>
<td>Activates the singleton instance.</td>
</tr>
<tr>
<td>![s] DoesInstanceExist</td>
<td>Checks if an instance of this MonoBehaviour exists.</td>
</tr>
<tr>
<td>![s] SetSingletonAutoCreate</td>
<td>Sets the object to be instantiated automatically if no instance of the singleton is found.</td>
</tr>
<tr>
<td>![s] SetSingletonType</td>
<td>Only required for Flash builds. If this function is not called by the class deriving from SingletonMonoBehaviour in the constructor the singleton can not be found by GetSingleton(...)</td>
</tr>
</tbody>
</table>
SingletonMonoBehaviourT ActivateSingleton Method

Activates the singleton instance.

**Namespace:** (Default Namespace)  
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```c#  
public static void ActivateSingletonInstance()
```

### Remarks

Call this function if you set an singleton object inactive before ever accessing the *Instance*. This is required because Unity does not (yet) offer a way to find inactive game objects.

### See Also

Reference  
SingletonMonoBehaviourT Class  
(Default Namespace) Namespace
SingletonMonoBehaviour<T>DoesInstanceExist Method

Checks if an instance of this MonoBehaviour exists.

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

**Syntax**

```csharp
public static T DoesInstanceExist()
```

**Return Value**

Type: `T`
A reference to the instance if it exists, otherwise `null`

**See Also**

Reference
SingletonMonoBehaviour<T> Class
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
**SingletonMonoBehaviourTSetSingletonAutoCreate Method**

Sets the object to be instantiated automatically if no instance of the singleton is found.

**Namespace:** (Default Namespace)
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

C# public static void SetSingletonAutoCreate(
    GameObject autoCreatePrefab
)  

### Parameters

**autoCreatePrefab**
Type: **GameObject**  
The prefab to be instantiated automatically.

### Remarks

Either the game object itself or one of its child objects must contain the singleton component

### See Also

Reference
**SingletonMonoBehaviourT Class**
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH
Send comments on this topic to ClockStone Support Email
SingletonMonoBehaviour<T> SetSingletonMethod

Only required for Flash builds. If this function is not called by the class deriving from SingletonMonoBehaviour in the constructor the singleton can not be found by GetSingleton(...)

**Namespace:** (Default Namespace)  
**Assembly:** AudioToolkit (in AudioToolkit.dll) Version: 8.2.0.0 (8.2.0.0)

### Syntax

```csharp
public static void SetSingletonType(
    Type type
)
```

### Parameters

**type**  
Type: `SystemType`  
[Missing <param name="type"/> documentation for "M:SingletonMonoBehaviour`1.SetSingletonType(System.Type)""]

### See Also

**Reference**  
SingletonMonoBehaviour<T> Class  
(Default Namespace) Namespace

Copyright (c) 2012 by ClockStone Software GmbH  
Send comments on this topic to ClockStone Support Email