

# Manual for VisiSonics' RealSpace3D Audio Unity Plugin API Reference

Beta Version

This simple manual outlines how the RealSpace3D Audio Unity PlugIn API is used via C# scripting.

Both the engine and the manual are under continuous development.

Please feel free to email VisiSonics for support ([support@visisonics.com](mailto:support@visisonics.com)). Suggestions for improvement are always welcome.

VisiSonics Corporation

06/12/2015

# VisiSonics - RealSpace3D Audio API Guide

API Documentation	3
1.1 RealSpace3D AudioSource Component	3
1.2 RealSpace3D AudioListener Component	22
1.3 RealSpace3D Virtual Room Component	22
2. RealSpace3D Audio Presets	28
Conclusion	28

# 1. API Documentation

## 1.1 RealSpace3D AudioSource Component

- **AdjustDopplerLevel**

```
void rs3d_AdjustDopplerLevel(float fDopplerLevel);
```

**Namespace:** RealSpace3D

### Parameters

**fDopplerLevel** (float) Is the level.

### Description

If set to 0, no effect is applied to the audio source.

```
// C# example
void foo()
{
    RealSpace3D_AudioSource theAudioSource =
        gameObject.GetComponent<RealSpace3D.RealSpace3D_AudioSource>();

    theAudioSource.rs3d_AdjustDopplerLevel(1.0f);
}
```

- **AdjustPitch**

```
void rs3d_AdjustPitch(float fPitch, int nIndex = 0);
void rs3d_AdjustPitch(float fPitch, string sClipName);
```

**Namespace:** RealSpace3D

### Parameters

**fPitch** (float) Is the pitch.  
**nIndex** (int) If more than one audio clip, the clip index whose pitch to adjust.  
**sClipName** (string) If more than one audio clip, the clip name whose pitch to adjust.

### Description

The pitch of the audio source.

```
// C# example
void foo()
{
    RealSpace3D_AudioSource theAudioSource =
        gameObject.GetComponent<RealSpace3D.RealSpace3D_AudioSource>();

    theAudioSource.rs3d_AdjustPitch(1.0f);
}
```

- **AdjustVolume**

```
void rs3d_AdjustVolume(float fVolume, int nIndex = 0);
void rs3d_AdjustVolume(float fVolume, string sClipName);
```

**Namespace: RealSpace3D**

### Parameters

**fVolume** (float) Is the volume.

**nIndex** (int) If more than one audio clip, the clip index whose volume to adjust.

**sClipName** (string) If more than one audio clip, the clip name whose volume to adjust.

### Description

Adjust the volume of the RS3D AudioSource.

```
// C# example
void foo()
{
    RealSpace3D_AudioSource theAudioSource =
        gameObject.GetComponent<RealSpace3D.RealSpace3D_AudioSource>();

    theAudioSource.rs3d_AdjustVolume(2.0f);
}
```

- **GetClipLength**

```
float rs3d_GetClipLength(int nIndex = 0);
float rs3d_GetClipLength(string sClipName);
```

**Namespace: RealSpace3D**

### Parameters

**nIndex** (int) If more than one audio clip, the clip index whose length to return.  
**sClipName** (string) If more than one audio clip, the clip name whose length to return.

### Description

Returns the length of the audio clip in seconds.

```
// C# example
void foo()
{
    RealSpace3D_AudioSource theAudioSource =
        gameObject.GetComponent<RealSpace3D.RealSpace3D_AudioSource>();

    float fClipLength = theAudioSource.rs3d_GetClipLength();
}
```

- **GetClipsCount**

**int rs3d\_GetClipsCount();**

**Namespace: RealSpace3D**

### Parameters

None

### Description

Returns the number of audio clips the RS3D AudioSource has.

```
// C# example
void foo()
{
    RealSpace3D_AudioSource theAudioSource =
        gameObject.GetComponent<RealSpace3D.RealSpace3D_AudioSource>();

    int nClipCount = theAudioSource.rs3d_GetClipsCount();
}
```

- **GetFastSpatialization**

**bool rs3d\_GetFastSpatialization();**

**Namespace: RealSpace3D**

**Parameters**

None

**Description**

Returns whether or not the audio source has fast spatialization on or off.

```
// C# example
void foo()
{
    RealSpace3D_AudioSource theAudioSource =
    gameObject.GetComponent<RealSpace3D.RealSpace3D_AudioSource>();

    bool bFastSpatOn = theAudioSource.rs3d_GetFastSpatialization();
}
```

- **GetOptimization**

**bool rs3d\_GetOptimization();**

**Namespace: RealSpace3D**

**Parameters**

None

**Description**

Returns whether or not the audio source has optimization on or off.

```
// C# example
void foo()
{
    RealSpace3D_AudioSource theAudioSource =
    gameObject.GetComponent<RealSpace3D.RealSpace3D_AudioSource>();

    bool bOptimizeOn = theAudioSource.rs3d_GetOptimization();
}
```

- **GetSourceName**

**string rs3d\_GetSourceName();**

**Namespace: RealSpace3D**

**Parameters**

None

**Description**

Returns the name of the object that the RealSpace3D AudioSource is associated with.

```
// C# example
void foo()
{
    RealSpace3D_AudioSource theAudioSource =
    gameObject.GetComponent<RealSpace3D.RealSpace3D_AudioSource>();

    string sSourceName = theAudioSource.rs3d_GetSourceName();
}

```

- **GetTime**

```
float rs3d_GetTime(int nIndex = 0);
float rs3d_GetTime(string sClipName);
```

**Namespace: RealSpace3D**

**Parameters**

**nIndex** (int) If more than one audio clip, the clip index whose time to return.  
**sClipName** (string) If more than one audio clip, the clip name whose time to return.

**Description**

Returns the current playback time of the audio clip in seconds.

```
// C# example
void foo()
{
    RealSpace3D_AudioSource theAudioSource =
    gameObject.GetComponent<RealSpace3D.RealSpace3D_AudioSource>();

    float fClipTime = theAudioSource.rs3d_GetTime();
}

```

- **GetTimeSamples**

```
int rs3d_GetTimeSamples(int nIndex = 0);  
int rs3d_GetTimeSamples(string sClipName);
```

Namespace: **RealSpace3D**

**Parameters**

**nIndex** (int) If more than one audio clip, the clip index whose time samples to return.

**sClipName** (string) If more than one audio clip, the clip name whose time samples to return.

**Description**

Returns the playback position in PCM samples.

```
// C# example  
void foo()  
{  
    RealSpace3D_AudioSource theAudioSource =  
        gameObject.GetComponent<RealSpace3D.RealSpace3D_AudioSource>();  
  
    int nTimeSamples = theAudioSource.rs3d_GetTimeSamples();  
}
```

- **GetVolume**

```
float rs3d_GetVolume(int nIndex = 0);  
float rs3d_GetVolume(string sClipName);
```

Namespace: **RealSpace3D**

**Parameters**

**nIndex** (int) If more than one audio clip, the clip index whose volume to return.

**sClipName** (string) If more than one audio clip, the clip name whose volume to return.

**Description**

Returns the current volume of the audio clip.

```
// C# example  
void foo()  
{
```



## VisiSonics - RealSpace3D Audio API Guide

```
RealSpace3D_AudioSource theAudioSource =
gameObject.GetComponent<RealSpace3D.RealSpace3D_AudioSource>();

float fVolume = theAudioSource.rs3d_GetVolume();
}
```

- **IsMuted**

```
bool rs3d_IsMuted(int nIndex = 0);
bool rs3d_IsMuted(string sClipName);
```

**Namespace: RealSpace3D**

### Parameters

**nIndex**        **(int)**    If more than one audio clip, the clip index whose mute status to return.

**sClipName**    **(string)** If more than one audio clip, the clip name whose mute status to return.

### Description

Returns the current mute status of the audio clip.

```
// C# example
void foo()
{
    RealSpace3D_AudioSource theAudioSource =
    gameObject.GetComponent<RealSpace3D.RealSpace3D_AudioSource>();

    bool bIsMuted = theAudioSource.rs3d_IsMuted();
}
```

- **IsPlaying**

```
bool rs3d_IsPlaying(int nIndex = 0);
bool rs3d_IsPlaying(string sClipName);
```

**Namespace: RealSpace3D**

### Parameters

**nIndex**        **(int)**    If more than one audio clip, the clip index whose playing.

**sClipName**    **(string)** If more than one audio clip, the clip name whose playing.

### Description

Returns true/false on the audio clip is playing.

```
// C# example
void foo()
{
    RealSpace3D_AudioSource theAudioSource =
        gameObject.GetComponent<RealSpace3D.RealSpace3D_AudioSource>();

    bool bClipIsPlaying = theAudioSource.rs3d_IsPlaying();
}
```

- **IsSoundSourceRunning**

```
bool rs3d_IsSoundSourceRunning(int nIndex = 0);
bool rs3d_IsSoundSourceRunning(string sClipName);
```

Namespace: **RealSpace3D**

### Parameters

**nIndex** (int) If more than one audio clip, the clip index whose running.  
**sClipName** (string) If more than one audio clip, the clip name whose running.

### Description

Returns true/false if the RS3D AudioSource is running.

```
// C# example
void foo()
{
    RealSpace3D_AudioSource theAudioSource =
        gameObject.GetComponent<RealSpace3D.RealSpace3D_AudioSource>();

    bool bIsRunning = theAudioSource.rs3d_IsSoundSourceRunning();
}
```

- **LoadAudioClip**

```
void rs3d_LoadAudioClip(AudioClip theAudioClip, int nIndex = 0);
void rs3d_LoadAudioClip(ref string sTheAudioClip, int nIndex = 0);
void rs3d_LoadAudioClip(ref AudioClip theAudioClip, int nIndex = 0);
```

**Namespace: RealSpace3D**

**Parameters**

<b>theAudioClip</b>	<b>(AudioClip)</b>	The Unity audio clip to load.
<b>sTheAudioClips</b>	<b>(ref string)</b>	Reference to string name of the audio clip to load.
<b>theAudioClip</b>	<b>(ref AudioClip)</b>	Reference to the audio clip to load.
<b>nIndex</b>	<b>(int)</b>	If more than one audio clip, the clip index to load.

**Description**

Loads an audio clip to the AudioSource. This will replace the current audio clip. Audio clips must reside in a folder or sub-folder labeled "Resources."

```
// C# example
void foo()
{
    RealSpace3D_AudioSource theAudioSource =
    gameObject.GetComponent<RealSpace3D.RealSpace3D_AudioSource>();

    string theSound = "RealSpace3D_Sounds/bond0";

    theAudioSource.rs3d_LoadAudioClip(ref theSound);

    // or

    theAudioSource.rs3d_LoadAudioClip(Resources.Load(theSound) as AudioClip);
}
```

• **LoadAudioClips**

```
void rs3d_LoadAudioClips(ref string [] sTheAudioClips);
void rs3d_LoadAudioClips(ref List<AudioClips> theAudioClips);
void rs3d_LoadAudioClips(ref AudioClip[] theAudioClips);
```

**Namespace: RealSpace3D**

**Parameters**

<b>sTheAudioClips</b>	<b>(ref string [])</b>	Reference to array of audio clip string names to load.
<b>theAudioClips</b>	<b>(ref list [])</b>	Reference to list of audio clips to load.

**theAudioClips**      (**ref AudioClip []**)      Reference to array of audio clips to load.

### Description

Loads audio clips to RealSpace3D AudioSource. Currently loaded audio clips will be removed and replaced. More or less audio clips than originally loaded can be loaded. Audio clips must reside in a folder or sub-folder labeled "Resources."

```
// C# example
void foo()
{
    RealSpace3D_AudioSource theAudioSource =
    gameObject.GetComponent<RealSpace3D.RealSpace3D_AudioSource>();

    string [] theSounds = {"RealSpace3D_Sounds/bond0",
                           "RealSpace3D_Sounds/cowboytheme 1",
                           "RealSpace3D_Sounds/tank2",
                           "RealSpace3D_Sounds/rodshornet"} ;

    theAudioSource.rs3d_LoadAudioClips(ref theSounds);

    // or

    AudioClip [] theSounds = new AudioClip[] {
        (AudioClip)Resources.Load("RealSpace3D_Sounds/bond0"),
        (AudioClip)Resources.Load("RealSpace3D_Sounds/cowboytheme1"),
        (AudioClip)Resources.Load("RealSpace3D_Sounds/tank2"),
        (AudioClip)Resources.Load("RealSpace3D_Sounds/rodsHornet")};

    theAudioSource.rs3d_LoadAudioClips(ref theSounds);
}
```

- **LoopSound**

```
void rs3d_LoopSound(bool bFlag, int nIndex = 0);
void rs3d_LoopSound(bool bFlag, string sClipName);
```

**Namespace: RealSpace3D**

### Parameters

**bFlag**            (**bool**)      Flag indicating true/false to loop audio clip.  
**nIndex**           (**int**)        If more than one audio clip, the clip index to loop.

**sClipName** (**string**) If more than one audio clip, the clip name to loop.

### Description

Sets true/false whether the audio clip should loop.

```
// C# example
void foo()
{
    RealSpace3D_AudioSource theAudioSource =
        gameObject.GetComponent<RealSpace3D.RealSpace3D_AudioSource>();

    theAudioSource.rs3d_LoopSound(true);
}
```

- **MuteSound**

```
void rs3d_MuteSound(bool bFlag, int nIndex = 0);
void rs3d_MuteSound(bool bFlag, string sClipName);
```

Namespace: **RealSpace3D**

### Parameters

**bFlag** (**bool**) Flag indication true/false to silence the audio clip.  
**nIndex** (**int**) If more than one audio clip, the clip index to mute.  
**sClipName** (**string**) If more than one audio clip, the clip name to mute.

### Description

Sets true/false to silence or not silence the audio clip.

```
// C# example
void foo()
{
    RealSpace3D_AudioSource theAudioSource =
        gameObject.GetComponent<RealSpace3D.RealSpace3D_AudioSource>();

    theAudioSource.rs3d_MuteSound(false);
}
```

- **PauseSound**

```
void rs3d_PauseSound(int nIndex = 0);
```

```
void rs3d_PauseSound(string sClipName);
```

**Namespace: RealSpace3D**

**Parameters**

**nIndex** (int) If more than one audio clip, the clip index to pause.  
**sClipName** (string) If more than one audio clip, the clip name to pause.

**Description**

Pauses the audio clip.

```
// C# example  
void foo()  
{  
    RealSpace3D_AudioSource theAudioSource =  
        gameObject.GetComponent<RealSpace3D.RealSpace3D_AudioSource>();  
  
    theAudioSource.rs3d_PauseSound(true);  
}
```

- **PlayIn3D**

```
void rs3d_PlayIn3D(bool bFlag, int nIndex = 0);  
void rs3d_PlayIn3D(bool bFlag, string sClipName);
```

**Namespace: RealSpace3D**

**Parameters**

**bFlag** (bool) Flag indicating true/false to play audio clip in RS3D sound.  
**nIndex** (int) If more than one audio clip, the clip index to play in RS3D sound.  
**sClipName** (string) If more than one audio clip, the clip name to play in RS3D sound.

**Description**

Plays audio clip in RS3D Audio spatialized sound or not.

```
// C# example  
void foo()  
{  
    RealSpace3D_AudioSource theAudioSource =  
        gameObject.GetComponent<RealSpace3D.RealSpace3D_AudioSource>();  
  
    theAudioSource.rs3d_PlayIn3D(true);  
}
```

```
}
```

- **PlayOnStart**

```
void rs3d_PlayOnStart(bool bFlag, int nIndex = 0);  
void rs3d_PlayOnStart(bool bFlag, string sClipName);
```

Namespace: **RealSpace3D**

**Parameters**

**bFlag** (bool) Flag indicating true/false to play on start.  
**nIndex** (int) If more than one audio clip, the clip index to play on start.  
**sClipName** (string) If more than one audio clip, the clip name to play on start.

**Description**

Plays audio clip on awake.

*Yes, the API says start but it actually will play on awake. We will update the engine and API on the next release but will keep this API so as not to pull the rug from anyone that is currently using it.*

```
// C# example  
void foo()  
{  
    RealSpace3D_AudioSource theAudioSource =  
        gameObject.GetComponent<RealSpace3D.RealSpace3D_AudioSource>();  
  
    theAudioSource.rs3d_PlayOnStart(true);  
}
```

- **PlaySound**

```
void rs3d_PlaySound(int nIndex = 0);  
void rs3d_PlaySound(string sClipName);
```

Namespace: **RealSpace3D**

**Parameters**

**nIndex** (int) If more than one audio clip, the clip index to play sound.  
**sClipName** (string) If more than one audio clip, the clip name to play sound.

### Description

Plays audio clip.

```
// C# example
void foo()
{
    RealSpace3D_AudioSource theAudioSource =
        gameObject.GetComponent<RealSpace3D.RealSpace3D_AudioSource>();

    theAudioSource.rs3d_PlaySound();
}
```

- **ResumeSound**

```
void rs3d_ResumeSound(int nIndex = 0);
void rs3d_ResumeSound(string sClipName);
```

Namespace: **RealSpace3D**

### Parameters

**nIndex** (int) If more than one audio clip, the clip index to resume playing.  
**sClipName** (string) If more than one audio clip, the clip name to resume playing.

### Description

Resumes playing paused audio clip.

```
// C# example
void foo()
{
    RealSpace3D_AudioSource theAudioSource =
        gameObject.GetComponent<RealSpace3D.RealSpace3D_AudioSource>();

    theAudioSource.rs3d_ResumeSound();
}
```

- **SetFastSpatialization**

```
void rs3d_SetFastSpatialization(bool bFlag);
```

Namespace: **RealSpace3D**



### Parameters

**bFlag (bool)** Flag to set.

### Description

Turns fast spatialization on/off.

```
// C# example
void foo()
{
    RealSpace3D_AudioSource theAudioSource =
        gameObject.GetComponent<RealSpace3D.RealSpace3D_AudioSource>();

    theAudioSource.rs3d_SetFastSpatialization(true);
}
```

- **SetFXReverb**

```
void rs3d_SetFXReverb(bool bFlag, float fFXReverbLength, int nIndex = 0);
void rs3d_SetFXReverb(bool bFlag, float fFXReverbLength, string sClipName);
```

Namespace: **RealSpace3D**

### Parameters

**bFlag (bool)** Flag indicator, turns FX reverb on/off.  
**fFXReverbLength (float)** Length of the reverb in milliseconds (0.0 - 2.0).  
**nIndex (int)** If more than one audio clip, the clip index to set FX reverb on.  
**sClipName (string)** If more than one audio clip, the clip name to set FX reverb on.

### Description

Sets FX Reverb on/off. If on plays for the indicated reverb length.

```
// C# example
void foo()
{
    RealSpace3D_AudioSource theAudioSource =
        gameObject.GetComponent<RealSpace3D.RealSpace3D_AudioSource>();

    theAudioSource.rs3d_SetFXReverb(true, 0.5f);
}
```

- **SetOptimization**

**void rs3d\_SetOptimization(bool bFlag);**

**Namespace: RealSpace3D**

**Parameters**

**bFlag**           **(bool)** Flag indicating true/false to turn on/off optimization.

**Description**

Turns RS3D AudioSource optimization on/off.

```
// C# example
void foo()
{
    RealSpace3D_AudioSource theAudioSource =
        gameObject.GetComponent<RealSpace3D.RealSpace3D_AudioSource>();

    theAudioSource.rs3d_SetOptimization(true);
}
```

- **SetPlay3DSound**

**void rs3d\_SetPlay3DSound(bool bFlag, int nIndex = 0);**

**void rs3d\_SetPlay3DSound(bool bFlag, string sClipName);**

**Namespace: RealSpace3D**

**Parameters**

**bFlag**           **(bool)** Flag indicating true/false to set audio clip in RS3D.

**nIndex**       **(int)** If more than one audio clip, the clip index to set audio clip in RS3D.

**sClipName**   **(string)** If more than one audio clip, the clip name to set audio clip in RS3D.

**Description**

Sets the audio clip play in 3D flag.

```
// C# example
void foo()
{
```

```
RealSpace3D_AudioSource theAudioSource =  
gameObject.GetComponent<RealSpace3D.RealSpace3D_AudioSource>();  
  
theAudioSource.rs3d_SetPlay3DSound(true);  
}
```

- **SetRolloffMode**

**void rs3d\_SetRolloffMode(ATTENUATION\_Type attenuationType);**

**Namespace: RealSpace3D**

**Parameters**

**attenuationType (ATTENUATION\_TYPE)** Enumerator indicating type of rolloff.

**ATTENUATION\_Type { Logarithmic, Linear, Custom };**

**Description**

Sets the sound source attenuation rolloff mode.

```
// C# example  
void foo()  
{  
    RealSpace3D_AudioSource theAudioSource =  
    gameObject.GetComponent<RealSpace3D.RealSpace3D_AudioSource>();  
  
    theAudioSource.rs3d_SetRolloffMode(ATTENUATION_Type.Logarithmic);  
}
```

- **SetSoundOptions**

**void rs3d\_SetSoundOptions(SoundOptions sOpts, int nIndex = 0);**  
**void rs3d\_SetSoundOptions(SoundOptions sOpts, string sClipName);**

**Namespace: RealSpace3D**

**Parameters**

**sOpts (SoundOptions)** RS3D sound options struct.

**nIndex (int)** If more than one audio clip, the clip index to set sound options.

**sClipName (string)** If more than one audio clip, the clip name to set sound options.

### Description

Sets the sound options for audio clip.

```
// C# example
void foo()
{
    RealSpace3D_AudioSource.SoundOptions sOpts;

    RealSpace3D_AudioSource theAudioSource =
    gameObject.GetComponent<RealSpace3D.RealSpace3D_AudioSource>();

    sOpts.fVolume =          2.5f;
    sOpts.fPitch =           1.0f;
    sOpts.bLoopSound =       false;
    sOpts.bMuteSound =       true;
    sOpts.bPlayOnStart =     true;
    sOpts.bPlay3DSound =     true;
    sOpts.bFXReverb =        false;
    sOpts.fFXReverbLength =  0.0f;

    theAudioSource.rs3d_SetSoundOptions(sOpts);
}
```

- **SetTime**

```
void rs3d_SetTime(float fTime, int nIndex = 0);
void rs3d_SetTime(float fTime, string sClipName);
```

Namespace: **RealSpace3D**

### Parameters

**fTime** (float) Time in audio clip to seek to.  
**nIndex** (int) If more than one audio clip, the clip index to seek the time to.  
**sClipName** (string) If more than one audio clip, the clip name to seek the time to.

### Description

Seeks the play time in the audio clip.

```
// C# example
void foo()
{
    RealSpace3D_AudioSource theAudioSource =
    gameObject.GetComponent<RealSpace3D.RealSpace3D_AudioSource>();
```

```
        theAudioSource.rs3d_SetTime(9.99f);  
    }
```

- **StopAllSounds**

```
void rs3d_StopAllSounds();
```

**Namespace: RealSpace3D**

**Parameters**

**None**

**Description**

Stops all audio clips on the sound source from playing.

```
// C# example  
void foo()  
{  
    RealSpace3D_AudioSource theAudioSource =  
        gameObject.GetComponent<RealSpace3D.RealSpace3D_AudioSource>();  
  
    theAudioSource.rs3d_StopAllSounds();  
}
```

- **StopSound**

```
void rs3d_StopSound(int nIndex = 0);  
void rs3d_StopSound(string sClipName);
```

**Namespace: RealSpace3D**

**Parameters**

**nIndex** (int) If more than one audio clip, the clip index to stop sound.  
**sClipName** (string) If more than one audio clip, the clip name to stop sound.

**Description**

Stops the audio clip from playing.

```
// C# example  
void foo()
```

```
{  
    RealSpace3D_AudioSource theAudioSource =  
        gameObject.GetComponent<RealSpace3D.RealSpace3D_AudioSource>();  
  
    theAudioSource.rs3d_StopSound();  
}
```

## 1.2 RealSpace3D AudioListener Component

The RealSpace3D AudioListener has no parameters that can be set in realtime.

## 1.3 RealSpace3D Virtual Room Component

The scripting examples assume you have either tagged the virtual rooms to identify them or you have instantiated the virtual room prefab and therefore know the virtual room you are accessing. The examples demonstrate how to instantiate the RS3D Audio Virtual Room into your scene.

- **GetPresetName**

```
string rs3d_GetPresetName();
```

**Namespace:** RealSpace3D

**Parameters**

None

**Description**

Returns the preset name currently in use by the virtual room.

*Note: The presets can be one of the VisiSonics' defaults (see section 2.0 **RealSpace3D Audio Presets** for choices) or one(s) you created in the Unity Editor.*

```
// C# example
```

```
void foo()
```

```
{
```

```
    GameObject theVirtualRoomPrefab = (GameObject)
```

```
        Instantiate(Resources.Load("RS3D_VirtualRoom"), new Vector3(10.0f,  
            5.0f, 10.0f), Quaternion.Euler(0, 180, 0));
```

## VisiSonics - RealSpace3D Audio API Guide

```
RealSpace3D_VirtualRoom theVirtualRoom =  
theVirtualRoomPrefab.GetComponent<RealSpace3D_VirtualRoom>();  
  
string sPresetName = theVirtualRoomPrefab.rs3d_GetPresetName();  
}
```

- **GetRoomName**

```
string rs3d_GetPresetName();
```

**Namespace: RealSpace3D**

**Parameters**

**None**

**Description**

Returns the virtual room name.

```
// C# example  
void foo()  
{  
    GameObject theVirtualRoomPrefab = (GameObject)  
        Instantiate(Resources.Load("RS3D_VirtualRoom"), new Vector3(10.0f,  
            5.0f, 10.0f), Quaternion.Euler(0, 180, 0));  
  
    RealSpace3D_VirtualRoom theVirtualRoom =  
        theVirtualRoomPrefab.GetComponent<RealSpace3D_VirtualRoom>();  
  
    string sRoomName = theVirtualRoomPrefab.rs3d_GetRoomName();  
}
```

- **GetRoomSize**

```
Vector3 rs3d_GetRoomSize();
```

**Namespace: RealSpace3D**

**Parameters**

**None**

**Description**

## VisiSonics - RealSpace3D Audio API Guide

Returns the size of the virtual room.

```
// C# example
void foo()
{
    GameObject theVirtualRoomPrefab = (GameObject)
        Instantiate(Resources.Load("RS3D_VirtualRoom"), new Vector3(10.0f,
            5.0f, 10.0f), Quaternion.Euler(0, 180, 0));

    RealSpace3D_VirtualRoom theVirtualRoom =
        theVirtualRoomPrefab.GetComponent<RealSpace3D_VirtualRoom>();

    Vector3 roomSize = theVirtualRoomPrefab.rs3d_GetRoomSize();
}
```

- **SetRoomPreset**

```
void rs3d_SetRoomPreset(int nIndex)
void rs3d_SetRoomPreset(string sPresetName)
```

**Namespace: RealSpace3D**

### Parameters

**nIndex** (int) The index of the preset.  
**sPresetName** (string) Preset name.

### Description

Sets the virtual room to the named preset. The virtual room walls, ceiling, and floor take on the reflective radiant energy of the materials used to define the preset.

*Note: The presets can be one of the VisiSonics' defaults (see section 2.0 **RealSpace3D Audio Presets** for choices) or one(s) you created in the Unity Editor.*

```
// C# example
void foo()
{
    GameObject theVirtualRoomPrefab = (GameObject)
        Instantiate(Resources.Load("RS3D_VirtualRoom"), new Vector3(10.0f,
            5.0f, 10.0f), Quaternion.Euler(0, 180, 0));

    RealSpace3D_VirtualRoom theVirtualRoom =
        theVirtualRoomPrefab.GetComponent<RealSpace3D_VirtualRoom>();
}
```



```
        theVirtualRoomPrefab.rs3d_SetRoomPreset("Cathedral");  
    }
```

- **SetRoomColor**

```
void rs3d_SetRoomColor(eRS3DCOLOR eColor);
```

**Namespace: RealSpace3D**

**Parameters**

**eColor (eRS3DCOLOR)** color of the virtual room's wireframe.

```
eRS3DCOLOR  
{  
    eBLACK,  
    eBLUE,  
    eCLEAR,  
    eCYAN,  
    eGRAY,  
    eGREEN,  
    eMAGENTA,  
    eRED,  
    eWHITE,  
    eYELLOW,  
};
```

**Description**

Sets the virtual room wireframe color.

*Note: This method only has effect in the editor. The virtual room is for editing purpose only, it is not visible in the final build, when called in the final build it is ignored.*

```
// C# example  
void foo()  
{  
    GameObject theVirtualRoomPrefab = (GameObject)  
        Instantiate(Resources.Load("RS3D_VirtualRoom"), new Vector3(10.0f,  
            5.0f, 10.0f), Quaternion.Euler(0, 180, 0));  
  
    RealSpace3D_VirtualRoom theVirtualRoom =  
        theVirtualRoomPrefab.GetComponent<RealSpace3D_VirtualRoom>();
```

```
    theVirtualRoomPrefab.rs3d_SetRoomColor(eRS3DCOLOR.eGREEN);  
}
```

- **SetRoomSize**

```
void rs3d_SetRoomSize(Vector3 theSize);
```

**Namespace: RealSpace3D**

**Parameters**

**theSize (Vector3)** Size of the virtual room.

**Description**

Changes the size of the virtual room.

```
// C# example  
void foo()  
{  
    GameObject theVirtualRoomPrefab = (GameObject)  
        Instantiate(Resources.Load("RS3D_VirtualRoom"), new Vector3(10.0f,  
            5.0f, 10.0f), Quaternion.Euler(0, 180, 0));  
  
    RealSpace3D_VirtualRoom theVirtualRoom =  
        theVirtualRoomPrefab.GetComponent<RealSpace3D_VirtualRoom>();  
  
    theVirtualRoomPrefab.rs3d_SetRoomSize(new Vector3(20.0f, 10.0f, 20.0f));  
}
```

- **ShowRoom**

```
void rs3d_ShowRoom(bool bFlag);
```

**Namespace: RealSpace3D**

**Parameters**

**bFlag (bool)** Indicator to show/hide room

**Description**

Shows or hides the virtual room.

## VisiSonics - RealSpace3D Audio API Guide

*Note: This method only has effect in the editor. The virtual room is for editing purpose only, it is not visible in the final build, when called in the final build it is ignored.*

```
// C# example
void foo()
{
    GameObject theVirtualRoomPrefab = (GameObject)
        Instantiate(Resources.Load("RS3D_VirtualRoom"), new Vector3(10.0f,
            5.0f, 10.0f), Quaternion.Euler(0, 180, 0));

    RealSpace3D_VirtualRoom theVirtualRoom =
        theVirtualRoomPrefab.GetComponent<RealSpace3D_VirtualRoom>();

    theVirtualRoomPrefab.rs3d_ShowRoom(false);
}
```

- **ShowRoomGuides**

```
void rs3d_ShowRoomGuides(bool bFlag);
```

**Namespace: RealSpace3D**

### **Parameters**

**bFlag (bool)** Indicator to show/hide room guides.

### **Description**

Shows or hides the virtual room guides.

*Note: This method only has effect in the editor. The virtual room is for editing purpose only, it is not visible in the final build, when called in the final build it is ignored.*

```
// C# example
void foo()
{
    GameObject theVirtualRoomPrefab = (GameObject)
        Instantiate(Resources.Load("RS3D_VirtualRoom"), new Vector3(10.0f,
            5.0f, 10.0f), Quaternion.Euler(0, 180, 0));

    RealSpace3D_VirtualRoom theVirtualRoom =
        theVirtualRoomPrefab.GetComponent<RealSpace3D_VirtualRoom>();

    theVirtualRoomPrefab.rs3d_ShowRoomGuides(false);
}
```

}

## 2. RealSpace3D Audio Presets

The following are the RealSpace3D Audio Virtual Room Presets.

1. Default
2. Cathedral
3. Office
4. Outside

Index 0 is reserved for “Custom”. User created presets start at index 5.

## Conclusion

Thank you for using RealSpace3D Audio. If you have any questions or issues send email to [support@visisonics.com](mailto:support@visisonics.com). Also, check the [realspace3daudio.com](http://realspace3daudio.com) developers forums page.