

BioWare Aurora Engine

Sound Object Format

1. Introduction

A **Sound** object is a source of sounds that play in an Area. They may be positional, playing from a specific location, or they may be area-wide, sounding the same regardless of where in the area the listener is.

Sounds are stored in the game and toolset using BioWare's Generic File Format (GFF), and it is assumed that the reader of this document is familiar with GFF.

Sound objects can be blueprints or instances. Sound blueprints are saved as GFF files having a UTS extension and "UTS " as the FileType string in their header. Sound instances are stored as Sound Structs within a module's GIT files.

2. Sound Struct

The tables in this section describe the GFF Struct for a Sound object. Some Fields are only present on Instances and others only on Blueprints.

For List Fields, the tables indicate the StructID used by the List elements.

2.1 Common Sound Fields

The Table below lists the Fields that are present in all Sound Structs, regardless of where they are found.

Table 2.1: Fields in all Sound Structs

Label	Type	Description
Active	BYTE	1 if the Sound is active and plays. 0 if the Sound is inactive and is not currently playing any wave files. Inactive Sounds can be manually activated via scripting.
Continuous	BYTE	1 if the Sound is continuous, or seamlessly looping. A continuous sound must have exactly one wave file to play, and it loops that wave continuously, over and over with no pauses. 0 if the sound is not continuous. That is, it consists of one or more wave files played individually. These waves may play in sequence, in random order, or with random or non-random pauses inbetween them. If a Sound is Continuous, then the values of the following Fields are ignored and always treated as 0: Interval, IntervalVrtn, PitchVariation, Random, RandomPositional, RandomRangeX, RandomRangeY, VolumeVrtn.
Elevation	FLOAT	Elevation of the Sound above or below the XYZ

		position at which it is placed in the toolset. When it plays, the audio emanates from the elevated position. Elevation can be negative.
Hours	DWORD	Set of bit flags specifying which hours of the day the sound will play in. Bit 0 is hour 00h00, bit 6 is 06h00, bit 14 is 14h00, etc.
Interval	DWORD	Interval in milliseconds between playing sounds in the Sound's list of waves.
IntervalVrtn	DWORD	Interval Variation measured in milliseconds. Each time a wave file finishes playing, determine how long to wait before playing the next one. Generate a random number ranging from (-IntervalVrtn) to (+IntervalVrtn), and add that to the Interval. If the resulting value is negative, treat it as 0 and play the next wave immediately.
LocName	CExoLocString	Name of the Sound as it appears on the toolset's Sound palette and in the Name field of the toolset's Sound Properties dialog. Does not appear in game.
Looping	BYTE	1 if the Sound repeatedly plays its waves. 0 if the Sound plays its waves at most once then becomes inactive.
MaxDistance	FLOAT	Radius in meters outside which a listener cannot hear the Sound at all. Must be greater than or equal to the MinDistance.
MinDistance	FLOAT	Radius in meters inside which a listener hears the Sound at maximum volume. Must be less than or equal to the MaxDistance.
PitchVariation	FLOAT	Pitch variation when playing waves in the Sound's list of waves, measured in octaves. Values from 0 to 1.0. A 0 pitch variation means the Sound always plays at normal pitch. A variation of 1 means that each time the a wave plays, its pitch is randomly anywhere from 0 to 1 octave higher or lower than normal.
Positional	BYTE	1 if the Sound plays from a specific position. The volume changes depending on the distance of the listener, and the relative volume from each speaker/headphone changes depending on the direction of the listener from the Sound. 0 if the Sound is area-wide, and has the same volume regardless of where the listener is in relation to the Sound. An area-wide Sound has no directional variation by speaker.
Priority	BYTE	Index into prioritygroups.2da.
Random	BYTE	1 if the waves in the Sound's wave list are chosen randomly each time one finishes playing. 0 if the waves are played in sequential order.
RandomPosition	BYTE	1 if the XYZ position of the Sound source varies randomly between the RandomRangeX and RandomRangeY. 0 if the position of the sound does not vary. This Field is ignored for area-wide (Positional=0) sounds.
RandomRangeX RandomRangeY	FLOAT	Random distance in meters from the Sound's XYZ position from which the Sound plays each time it plays

		a wave. These Fields are ignored if Positional=0 or RandomPosition=0.
Sounds	List	The Sound's wave list. A list of wave files to play. Each Struct in the List has Struct ID 0, and contains a single CResRef Field called Sound, which is the ResRef of a WAV file.
Tag	CExoString	Tag of the Sound. Up to 32 characters.
TemplateResRef	CResRef	For blueprints (UTS files), this should be the same as the filename. For instances, this is the ResRef of the blueprint that the instance was created from.
Times	BYTE	Times of day in which to play the Sound. 0 = time-specific. Use the Hours Field to determine when to play. 1 = Day 2 = Night 3 = Always If the Sound plays during the Day or Night, then day and night are as defined by the Mod_DawnHour and Mod_DuskHour Fields in the module's module.ifo file. See Table 2.1 of the IFO document .
Volume	BYTE	Volume to play each wave file at. Ranges from 0 (min) to 127 (full)
VolumeVrtn	BYTE	Volume Variation from 0 to 127. Each time a wave is to be played, randomly select a number from (-VolumeVrtn) to (+VolumVrtn) and add it to the Volume, then clamp the result to the range of 0 to 127 and use that as the actual volume.

The various combinations of the Looping and Random properties merit additional explanation as to how they interact with each other. There are four options for playing multiple sounds.

Continuously choose a new random sound to play (Random 1, Looping 1) - A sound is randomly chosen from the sound list and played. Once it has played and the Interval period has passed, another sound is randomly picked from the list and played. The process repeats forever or until the SoundObjectStop() scripting function is called.

Play a randomly selected sound once (Random 1, Looping 0) - One sound is randomly chosen from the list and played. After that, this sound object becomes inactive and no more sounds are played. The sound object can be reactivated via the SoundObjectPlay() scripting function. This option is most useful if Active is initially false, and the sound object is manually triggered during the game by using a script.

Continuously play sounds in order (Random0, Looping 1) - The first sound in the sound list is played, then there is a pause corresponding to the Interval, then the next sound is played, then there is a pause corresponding to the interval, and so on until all the sounds have played. This process repeats forever or until scripted to stop.

Play list in order once (Random 0, Looping 0) - The sounds are played in order with an Interval delay between them, and once all the sounds have played, the current sound object deactivates and does not play again. This option is most useful if Active is false, and the Sound is manually triggered during the game by using a script.

2.2. Sound Blueprint Fields

The Top-Level Struct in a UTS file contains all the Fields in Table 2.1 above, plus those in Table 2.2 below.

Table 2.2: Fields in Sound Blueprint Structs

Label	Type	Description
Comment	CExoString	Module designer comment.
PaletteID	BYTE	ID of the node that the Sound Blueprint appears under in the Sound palette.
TemplateResRef	CResRef	The filename of the UTS file itself. It is an error if this is different. Certain applications check the value of this Field instead of the ResRef of the actual file. If you manually rename a UTS file outside of the toolset, then you must also update the TemplateResRef Field inside it.

2.3. Sound Instance Fields

A Sound Instance Struct in a GIT file contains all the Fields in Table 2.1, plus those in Table 2.3 below.

Table 2.3: Fields in Sound Instance Structs

Label	Type	Description
GeneratedType	BYTE	0 if manually placed by the module builder in the toolset. 1 if autogenerated by the Area Properties dialog as part of the current Area's Ambient Day or Ambient Night Sound. See the AmbientSndDay and AmbientSndNight properties in Table 3.2 and 3.3 of the Area format document. If GeneratedType is 1, then the Sound instance is subject to automatic deletion when the user picks a different ambient sound for the area.
TemplateResRef	CResRef	For instances, this is the ResRef of the blueprint that the instance was created from.
XPosition YPosition ZPosition	FLOAT	(x,y,z) coordinates of the Sound within the Area that it is located in.

2.4. Sound Game Instance Fields

After a GIT file has been saved by the game, the Sound Instance Struct not only contains the Fields in Table 2.1 and Table 2.3, it also contains the Fields in Table 2.4.

INVALID_OBJECT_ID is a special constant equal to 0x7f000000 in hex.

Table 2.4: Fields in Sound Instance Structs in SaveGames

Label	Type	Description
ActionList	List	List of Actions stored on this object StructID 0. See Section 6 of the Common GFF Structs document.
ObjectId	DWORD	Object ID used by game for this object.
VarTable	List	List of scripting variables stored on this object.

		StructID 0. See Section 3 of the Common GFF Structs document.
--	--	---

3. The 2DA Files Referenced by Sound Fields

3.1. Priority Groups

In a Sound Struct, the Priority Field is an index into prioritygroups.2da. Table 3.1.1 describes the columns in the 2da.

The game and toolset are hardcoded to reference particular rows in prioritygroups, so only the programmers may change the order of rows, or add or remove rows.

Table 3.1.1: prioritygroups.2da columns

Column	Type	Description
Label	String	Programmer label
Priority	Integer	Matches up to hardcoded integer constants in sound engine source code. This means that you may not add, remove, or modify the order of rows in prioritygroups.2da.
Volume	Integer	Volume from 0 to 127
MaxPlaying	Integer	Maximum number of sounds of this priority that may play simultaneously.
Interrupt	Integer	0 if sound may be interrupted 1 if sound may not be interrupted
FadeTime	Integer	When stopping the sound, number of milliseconds of fadeout.
MaxVolumeDist	Integer	For placed Sound objects instances, the MaxDistance overrides this 2da value.
MinVolumeDist	Integer	For placed Sound objects instances, the MinDistance overrides this 2da value.
PlaybackVariance	Float	Pitch variance in octaves when playing sounds of this priority. Range is 0 to 1.0. For placed Sound objects instances, the PitchVariance overrides this 2da value.

Sound objects edited in the toolset always have the priority groups listed in Table 3.1.2, in accordance with the values of their Looping and Positional Fields. The other rows in prioritygroups.2da are used for sounds generated by the game.

Table 3.1.2: Toolset Sound Priorities

Row	Description	Looping	Positional
2	Looping area-wide ambients	1	0
3	looping positional ambients	1	1
19	single-shot global	0	0
20	single-shot positional	0	1

3.2. Default Values

There are two 2da files that determine the default values of various Sound object Fields when the user creates a new sound using the Sound Wizard in the toolset. The Sound Wizard is hardcoded to look up certain rows in the sounddefaults 2da files depending on the options that the user selected within the Wizard's GUI.

Table 3.2.1: sounddefaultspos.2da columns

Column	Type	Description
Label	String	Programmer label
RadiusInner	Float	default MinDistance
RadiusOuter	Float	default MaxDistance
RandomRngX	Float	default RandomRangeX
RandomRngY	Float	default RandomRangeY
Height	Float	default Elevation

Table 3.2.2: sounddefaultstim.2da columns

Column	Type	Description
Label	String	Programmer label
Looping	Integer	default Looping
Continuous	Integer	default Continuous
Random	Integer	default Random
Interval	Integer	default Interval
IntervalVar	Integer	default IntervalVrtn
PitchVar	Float	default PitchVrtn
VolumeVar	Float	default VolumeVrtn