Interactive 3D Audio Rendering Guidelines

Level 2.0

**Prepared by the
3D Working Group of the
Interactive Audio Special Interest Group**

September 20, 1999

revision 1.0a

Acknowledgments:

Disclaimer:

Table of Contents

Introduction

This document represents the work of a group of 3D hardware and software vendors and application developers meeting as the 3D Audio Working Group (3DWG) of the Interactive Audio Special Interest Group (IASIG) of the MIDI Manufacturers Association. The goal of the group is to influence and improve the development of platforms for interactive multimedia in the area of 3D audio playback.

This document defines the expectations of the group with regards to current acceptable Interactive 3D Audio rendering functionality, and is an extension to the work done to define minimal rendering capabilities, published as the IASIG 3D Audio Rendering Guidelines — version One ('Level 1.0').

The new 'Level 2.0' concepts are illustrated through recommended DirectSound3D Property Sets for use with Windows, but this is for convenience only and should not be interpreted to mean that this specification is only applicable to that platform. Ultimately, these guidelines should help move the entire PC industry in a common direction, towards greater quality, superior performance, and more realistic audio in multimedia applications.

The IASIG Interactive 3D Audio Rendering Guidelines — Level 2.0 specification may also be referred to by the acronym 'I3DL2'.

Abstract

The first generation of 3D audio localization technology provided startling positional accuracy, allowing listeners to perceive sounds emanating from above, below and from behind them, when rendered using two speakers or more. However, this technology was applied with great variation in features and performance in consumer PC products — all under the same banner of "3D audio." As a result, it has been difficult for consumers, let alone developers, to know for sure what to expect from PC audio systems and PC entertainment titles claiming to support "3D Audio".

The 'Level 1.0' guidelines helped correct this situation by defining minimal rendering requirements, a lexicon of terms, and evaluation guidelines regarding 3D positional audio. The Level 2.0 guidelines build upon that work by addressing more advanced environmental modeling functionality.

The Level 2.0 enhancements include:

- Environment reverberation model (conveying a sense of the space where the listener is located).
- Enhanced distance model, taking advantage of the reverberation cues in the current environment.
- Occlusion and obstruction models for rendering the muffling effects of obstacles inside environments or partitions between environments.
- A DirectSound3D property set which provides support for Level 2.0 functionality in hardware on PCs with Microsoft's Windows.

This document is a companion to the 'Level 1.0' document. Together they are intended to influence how 3D audio technology is applied to PC systems (and interactive entertainment products in general) and to assist developers and reviewers in designing and evaluating 3D audio renderers.

Overview of Contents

The Glossary section includes additions to the 'Level 1.0' glossary of terms, covering the precise definition of terms for the use of common language for users of the guideline.

- The *Model* section describes the Level 2.0 3D Audio model, and how it is used.

- The *Requirements* section states all minimum system features and functions needed for a renderer to qualify as IASIG Interactive 3D Audio — Level 2.0 (I3DL2) compliant.

- *Appendix 1* describes an example application programming interface (API) embodying the I3DL2 guideline. This is presented as an extension of an existing 'Level 1.0' implementation: Microsoft DirectSound3D (part of Direct X 5.0). The Level 2.0 extensions are implemented in the form of two Microsoft DirectX (DirectSound3D) property sets: a Listener property set and a Source property set. The different properties are listed and defined, including their minimum, maximum, and default values. Although this API specification is presented as a Level 2.0 extension of DirectSound3D for clarity and practical reasons, this Appendix provides all required elements for specifying a Level 2.0 extension to any existing 'Level 1.0'-compatible 3D audio rendering API.

- *Appendix 2* contains print-outs of header files for C/C++ implementations of drivers and applications (games) supporting the DirectSound3D property set described in Appendix 1. These files specify the data structures and the names of the properties and values defining the API, including the GUIDs (Globally Unique Identifiers) of the two property sets. In addition, this appendix provides example environment reverberation presets and material transmission presets intended to help developers of I3DL2 implementations.

Glossary — Level 2.0 Additions

- **Direct-path Sound** from a sound source: the sound that reaches the listener directly, via free propagation in the air or medium, possibly through or around an obstacle, but not via reflections on walls or obstacles.

- **Reflected Sound**: the sound that reaches the listener indirectly, via reflections on walls or obstacles.

- (Early) **Reflections**: the part of the reflected sound that reaches the listener first, and can be represented as a set of successive discrete "echoes".

- (Late) **Reverberation**: the part of the reflected sound that follows the early reflections. In a typical room, this consists of a dense succession of echoes whose energy decays exponentially (see Diffusion).

- **Obstruction:** the muffling (attenuation and filtering) of sound by an object between the listener and the source, contained within a common environment (room). The Direct path can reach the listener via *diffraction around the obstacle* and/or via *transmission through the obstacle*. In both cases, the direct path is muffled (low-pass filtered) but the reflected sound from that source is unaffected (because the source radiates in the listener's environment). Most often, the transmitted sound is negligible, and the low-pass effect only depends on the position of the source and listener relative to the obstacle (not on the transmission coefficient of the material). However, in the case of a highly transmissive obstacle (such as a curtain), the sound that goes through the obstacle may not be negligible compared to the sound that goes around it.

- **Occlusion:** the muffling of sound by a partition or wall separating two environments. Sounds which are in a different room or environment can reach the listener's environment by transmission through walls, or by traveling through any openings between the sound source's and the listener's environments. When these sounds reach the listener's environment they have been affected by the transmission or diffraction effects, and therefore both the direct-path sound from a source and the contribution by that source to the reflected sound in the listener's environment are muffled.

- **Diffusion** (echo density) is proportional to the number of echoes per second in the late reverberation. Depending on the reverberator implementation, it may or may not apply to the early reflections, and the echo density may or may not increase along the late reverberation tail (these choices are left to the implementer). Any attempt to provide an objectively measurable definition of Diffusion in an artificial reverberator would be arguable. However, the relative variation (increase or reduction) of diffusion can be defined objectively.

- **Density** (modal density) is proportional to the number of resonances per Hertz in the late reverberation. It is related to the perceived spectral coloration of the late reverberation (less coloration for higher modal density —although, depending on the implementation, that coloration may be affected by other parameters of the reverberator). The modal density can be objectively defined for certain classes of reverberation algorithms, but not for others (in the class of time-invariant feedback delay networks, the modal density is proportional to the total duration of the delays in the reverberator's feedback loop).

Models and Concepts

Environment reverberation model

Reverberation response model

The reverberation response model is described in Figure 1. The impulse response is divided into three temporal sections respectively labeled Direct (direct path), Reflections (early reflections) and Reverb (exponentially decaying late reverberation). The temporal division between these components is defined by the parameters Reflections_delay and Reverb_delay. The Decay_time is derived from the slope of the late reverberation (Reverb) decay, and defined as the time that Reverb takes to decay by 60 dB from its onset.

This reverberation model describes the response at the listener's position for a particular position of the source, assuming that the two are in the same environment (the 'current' environment). Therefore, in the present proposal, all reflection and reverberation parameters refer to the listener's room —not to the source's room if it is different (occlusion). The reverberation from the source's environment, if any, is assumed to be "pre-rendered" (added to the source signal independently from the modalities described in these guidelines).

The reverberation response is characterized by the following parameters:

- the energy in each of the three sections Direct, Reflections and Reverb at low frequencies,

- the Reflections_delay and the Reverb_delay,

- the 'Direct filter': a low-pass filter which affects the Direct component by reducing its energy at high frequencies,

- the 'Room filter': a low-pass filter which affects the Reflections and Reverb components identically by reducing their energy at high frequencies,

- the Decay_time at low and high frequencies,

- the Diffusion and the Density of the late reverberation (defined below).

The definition of 'low' and 'high' frequencies is given below in the section 'Spectral Effects'.

The late reverberation is assumed to be diffuse, i. e. coming from all directions around the listener. No particular assumption is made regarding the spatial distribution or the temporal distribution of the early reflections within the Reflections section, other than the following:

- the delay of the first reflection relative to the direct path is at least equal to Reflections_delay,

- the delay of the last early reflection relative to the direct path is less than
  Reflections_delay + Reverb_delay,

- the onset of the late reverberation decay is no earlier than Reflections_delay + Reverb_delay, relative to the direct path.
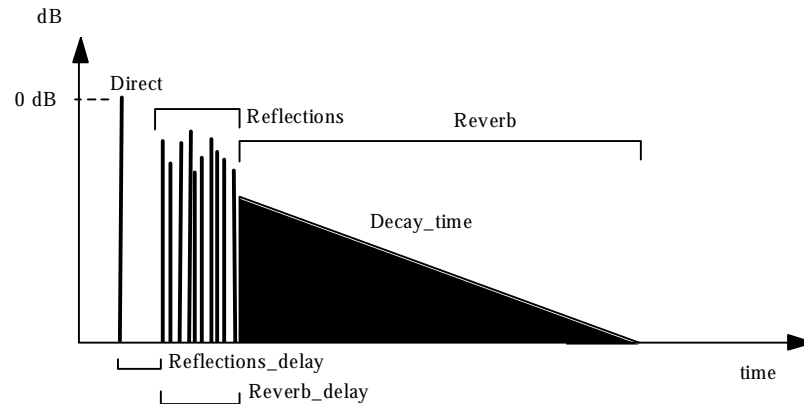
Figure 1 - Reverberation response model

Minimum and Maximum Distance, Rolloff Factor

The 'Level 1.0' guideline defines the notions of Minimum_distance, Maximum_distance and Rolloff_factor. The 'Level 1.0' renderer applies an attenuation to the source signal according to source-listener distance as follows:

- If distance ≤ Minimum_distance, the attenuation is 0 dB (no attenuation).

- If Minimum_distance ≤ distance ≤ Maximum_distance, the attenuation is –6 dB for each doubling of distance if the Rolloff Factor is 1.0. For values different from 1.0, the Rolloff Factor acts as a multiplier applied to the source-listener distance (diminished by the minimum distance). The variation of signal amplitude vs. distance is given by:

  A = min_distance / (min_distance + Rollof_factor * (distance – min_distance)).

- If distance ≥ Maximum_distance, the attenuation no longer varies with distance.

In natural environments, the total intensity of the reflected sound received by the listener varies little vs. source-listener distance, and therefore, the direct-to-reflected energy ratio decreases with distance. This provides an essential cue to the listener for assessing the distance of the source.

In general, the reflected intensity decays somewhat with increasing distance, although not as fast as the direct intensity. This Level 2.0 guideline therefore provides means for automatically attenuating the reflected sound (Reflections and Reverb) according to source-listener distance, in addition to attenuating the direct component as in 'Level 1.0'. The same values of Minimum_distance and Maximum_distance are used, with only a different value for the Rolloff_factor, denoted Room_rolloff_factor.

The default value of Room_rolloff_factor is 0.0, which implies that the reflected intensity does not vary with distance, while the default value for the (Direct) Rolloff_factor is 1.0. Setting Room_rolloff_factor to 0.5, instead, would imply that the reflected sound is also attenuated with increasing distance, although not as fast as the direct path.

Environment definition and presets

The Environment reverberation response is defined in a manner that applies to all sound sources in the Environment, and must therefore be independent of the source-listener distance:

- An Environment is characterized by the values of all reverberation response parameters when distance = Minimum_distance. This defines an 'Environment preset'.

The only reverberation response parameters which vary with distance, in the proposed model, are the energies in the three temporal sections Direct, Reflections and Reverb (energy is defined as the sum of the squared signal samples within the temporal section considered). At Minimum_distance, the intensity level of the Direct component is 0 dB (no attenuation with respect to the source signal.

The intensity level of 0 dB, for any of the three components, is obtained when the total energy in that section of the impulse response is equal to the energy of the exciting pulse (the source signal in this case). This characterization assumes that the Direct filter and the Room filter are ineffective (constant level across the whole frequency range) and that Decay_time is the same at all frequencies. Note that the intensity level of Reverb or Reflections is defined relative to the *source signal* (not relative to the Direct component attenuated by the 'Level 1.0' rolloff effect).

The Level Two guideline proposes a list of example environment presets in order to facilitate the work of the application developer or content author.

Diffusion, Density, and Room Size

Density, Diffusion or Room Size are parameters commonly found in artificial reverberation units. A problem with these parameters is that their definition is highly vendor or algorithm dependent.

Despite the difficulty of characterizing these two notions, leaving them out might significantly reduce the functionality / expressivity offered to the content author, compared to common reverberation units. therefore these guidelines include Diffusion and (modal) Density as normative properties. They are both expressed in percents (relative to the range of variation offered by a particular implementation).

The guideline does not give an absolute specification of how much echo density or modal density a given percent value means (it depends on the renderer). A Diffusion of 100% produces a smooth reverberation decay, while reducing Diffusion gives the reverberation a more "grainy" character, especially noticeable with percussive sound sources. A Density of 100% produces a natural-sounding reverberation timbre, while reducing Density produces a more "colored" or "hollow" sound (as found in small reverberant spaces such as bathrooms). The renderer should, as much as possible, ensure that the scale 0-100% maps linearly to the corresponding perceived variation in the reverberation quality.

A Room Size parameter would typically affect at least the modal density, but might also affect Reflections_delay, Reverb_delay, Decay_time, Diffusion. It is therefore regarded as a high-level parameter which does not need to be normalized in these guidelines. Its definition remains vendor dependent.

Spectral effects

The specification of spectral effects in these guidelines is based on the following assumptions and considerations:

1. All spectral effects (including decay time, occlusion, obstruction...) are specified as simple low-pass effects.

2. The API specification should not imply a particular topology or design for these different low-pass filters (the implementation of a digital low-pass filter could be one pole, two pole, 1rst or 2nd order pole-zero shelving filter...).

3. Content authors should be allowed to assume that a given parameter setting will produce essentially the same effect on all end-user platforms (renderers).

A common method for controlling a low-pass effect is to use the -3dB cut frequency (or possibly, for a shelving filter, the half-gain crossover frequency). However, for a given value of the cut frequency, the audio effect differs substantially depending on the design (6 dB/octave for a 1-pole filter, 12 dB/octave for a 2-pole filter...).

A more reliable choice is to specify the attenuation at a reference high frequency (5 kHz, for instance). All low-pass effects are specified as high-frequency attenuations in dB relative to low frequencies, taking the same reference high frequency for all effects. The interface allows the programmer to modify the value of this reference frequency, so that content authors can adjust the frequency where control of low-pass effects is guaranteed on any platform. This reference frequency could possibly change at different places in a game (but applies to all spectral effects).

This way of controlling low-pass effects is similar to a using graphic equalizer (where one controls levels in fixed frequency bands). Similarly, the high-frequency decay time is defined as the decay time at the reference high frequency. These guidelines do not offer separate controls of low- and mid-frequency settings for the decay time or other frequency-dependent effects. The reference low frequency is assumed to be 0 Hz (although filters are expected to generally have a substantially flat response up to 1 kHz).

This approach lets each hardware or software vendor decide what low-pass filter topology to use. In this way, the Level 2.0 specification can be met by implementations that are more or less limited in resources. Another advantage of this approach is the following: if two low-pass effects are cascaded, their combined effects can be described in the same terms, by adding the dB attenuations at the reference frequency. This property can be exploited by the implementer in order to lump several lowpass effects in a single digital filter unit. Furthermore, it allows the sound designer to predict the overall effect of combined low-pass filtering effects.

This method is also used in the specification of the Occlusion and Obstruction properties, and an example implementation is shown in the following chapter.

Interaction between sound sources and environment

Source processing model

As shown in Figure 2, the processing model for each sound source comprises an attenuation and a low-pass filter applied both to the direct path and the reflected sound, independently.
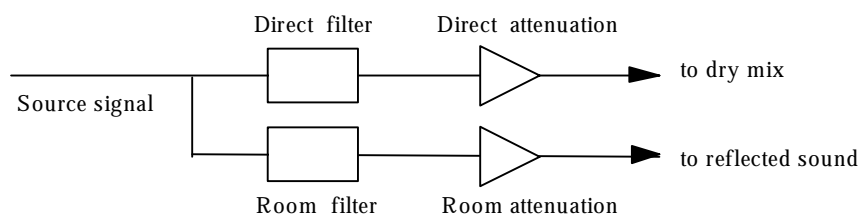

Figure 2 - Source processing model (not including 'Level 1.0' 3D positional functions)

Occlusion and Obstruction

The concepts of Occlusion and Obstruction enable the reproduction of the effects of obstacles standing between a source and the listener. Sounds can be heard through walls from other rooms, from around corners, through open and closed doors and windows, and from behind other objects. These sounds are different from the same sounds unobstructed, and this difference helps give a listener detailed information about the environment in which he or she is immersed and the sound-emitting objects within it. The Occlusion and Obstruction properties allow sources to be made to sound as if they are in other rooms, behind doors, around corners, or in any other way muffled by an obstacle.

Sounds which are transmitted through material structures undergo a frequency dependent attenuation. This attenuation usually has a low-pass character with the amount of attenuation and the slope of the high frequency roll-off being dependent on the material, thickness, and construction of the partition (the high frequency roll-off is typically about 5 to 10 dB per octave).

Sounds which travel around corners and through openings also undergo a low-pass attenuation due to a phenomenon known as diffraction. The effect of an obstacle will depend on the size of that obstacle with respect to the wavelength of the sound. If small compared to the wavelength, it will have very little effect. Thus low frequencies tend to travel around corners or swallow obstacles, and are not as attenuated by obstructions as high frequencies are. As rule of thumb, the greater the angle a sound path must make to travel around an obstacle, the greater the amount of high-frequency attenuation.

The perceptual effects of the two situations, Obstruction and Occlusion, are of similar nature: the sound undergoes an amount of low-pass attenuation. The difference between the two situations is that the reflected sound remains unaffected in the case of Obstruction, while it is affected in the case of Occlusion. Both effects are reproduced by combining a low-pass filter and a frequency-independent attenuation. In order to enable this functionality, it would in principle have been sufficient to expose the Direct and Room attenuations/filters defined in the previous section. However, programming these effects in an interactive application would have been inconvenient because it would systematically require updating four parameters simultaneously (for occlusion) or at least two parameters (obstruction).

For this reason, these guidelines provide a more hierarchical interface to control the effects of obstacles:

- The proposed interface distinguishes Obstruction (where only the dry path is muffled) and Occlusion (where both the dry path and the room effect are muffled).

- Both effects are controlled by specifying the high-frequency attenuation (the main perceived effect) optionally accompanied by an attenuation at low frequencies. In order to allow controlling both effects via a single knob, the second is relative to the first, as illustrated in Figure 3.
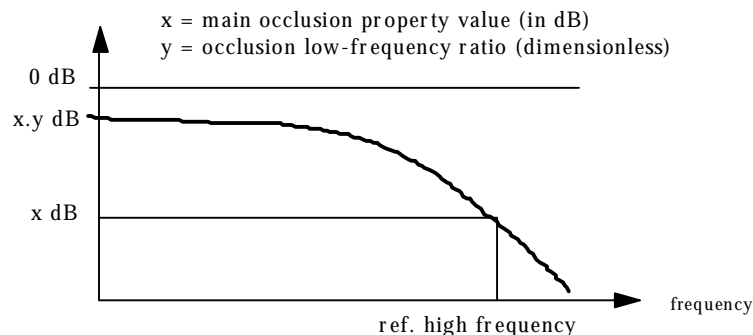
Figure 3 - Occlusion or Obstruction filter control model

Example implementation

Figures 4 and 5 show the frequency response of the Occlusion filter when the Occlusion low-frequency ratio is set to 0.25 and the main Occlusion property varies from 0 to -80 dB in steps or 4 dB.   The figures show two different implementations (1-pole or 2-pole filter) designed to maintain the same attenuation at a reference frequency of 5 kHz. The same behavior applies to the Obstruction effect.

Up to about -20 or -30 dB for the Occlusion value (attenuation at 5 kHz), the two implementations behave (and sound) very similarly. For stronger attenuations, the difference between the two frequency responses curves increases (but by then the sound is already quite attenuated). If we had used the -3dB cut frequency to specify the low-pass effect, the difference in high-frequency slope between the two types of filters would imply that the 1rst-order filter essentially lets an additional octave pass through, while the 2nd-order implementation filters it out.

In the examples, the 1-pole occlusion filter has the transfer function:

$$H(z) = k \frac{1-a}{1-a \cdot z^{-1}} \quad \text{where} \quad k = 10^{x \cdot y / 20}.$$

The filter coefficient is: $a = \dfrac{1 - g\cos \boldsymbol{w} - \sqrt{2g(1-\cos \boldsymbol{w}) - g^2(1-\cos^2 \boldsymbol{w})}}{1-g}$,

where $g = 10^{x(1-y)/10}$, $\boldsymbol{w} = 2\boldsymbol{p} \cdot F_H / F_S$, F_H is the reference high frequency and F_S is the sample rate.

The 2-pole occlusion filter was designed by cascading two identical 1-pole occlusion filters in which k and g were replaced by their square root.

Figure 4 - Occlusion filter frequency response (one pole implementation)

two pole - occl = 0 to -80 dB

Figure 5 - Occlusion filter frequency response (two pole implementation)
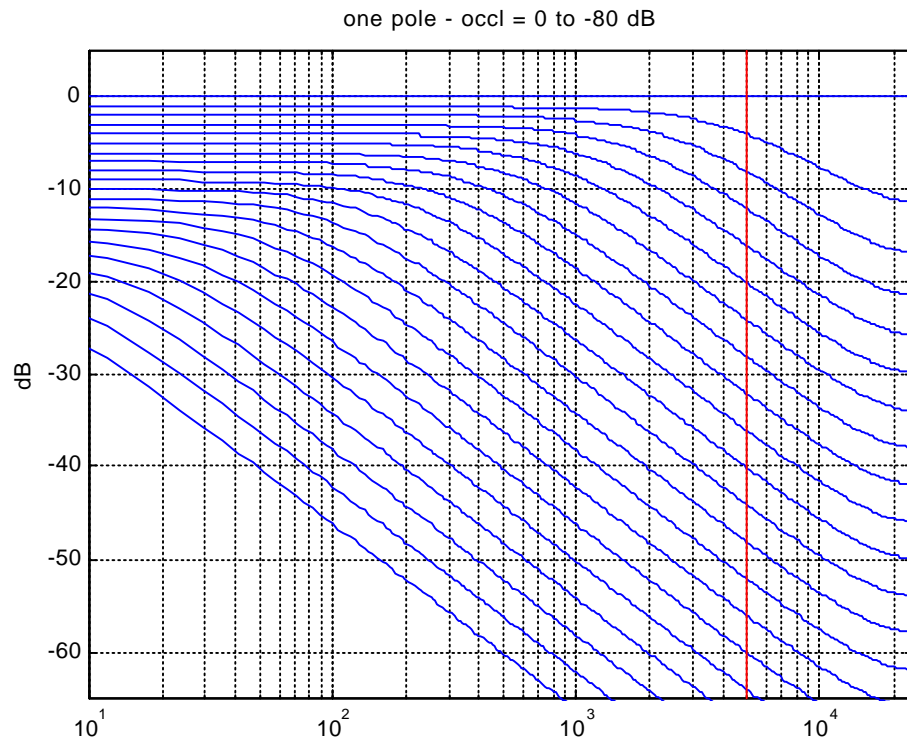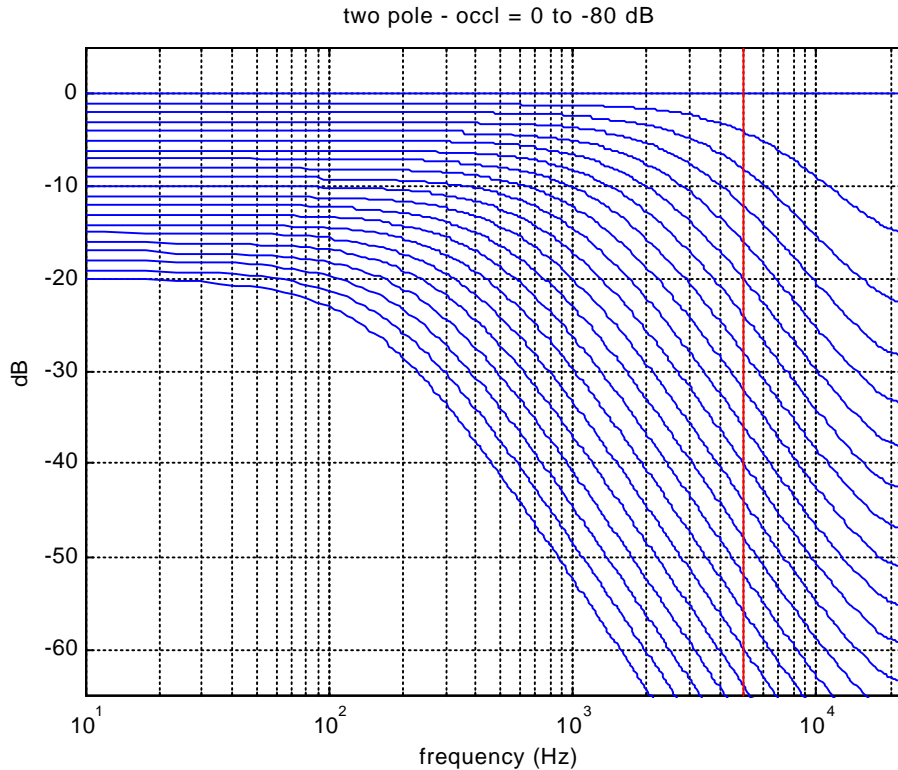
Using the Occlusion and Obstruction properties

The Occlusion property, which is used when a sound source is in a different environment than the listener, might be used in the following scenario. A loud moving sound source is in a room which has a closed door leading to a hallway. If the listener is in the hallway outside the room the sound source could have its Occlusion property set to -40 dB, which makes it sound as if it is behind a thick wall. When the source passes in front of the doorway to the room the Occlusion property value can be changed to -30 dB, which will make the sound slightly more clear since the door is thinner than the wall. While the door opens the value can be gradually changed to -8 dB, and if the source enters the room the value can again be changed to 0 dB, and the sound will be completely clear.

The Obstruction property might be used in the following way. If the listener and the sound source are in the same room and there is nothing between them then the Obstruction property for the sound source can be set to 0 dB (its default value). If the sound source moves behind a large object (a large pillar for example) the Obstruction property could be set to a value of -25 dB. As the sound source moves further behind the obstacle the Obstruction property value can be progressively lowered. The value of the Obstruction property can be related to the angle which the sound must travel around to reach the listener (the more acute the angle, the more the sound is muffled.)

Material transmission presets

A material transmission preset can be defined as an array of two values {Occlusion, Occlusion_LF_ratio}, defining the transmissivity for a given material and a given thickness. As mentioned earlier, such a preset can also be used for Obstruction in the case of highly transmissive obstacles. A list of example material presets is proposed in Appendix two of these guidelines (header file 3dl2help.h).

Minimum Requirements – Level 2.0

- **16 simultaneous sources (32 streams are recommended) with**

- sampling rate of 22050Hz (or higher) at 16-bit resolution

- object and listener 3D position (x, y, z)

- object and listener velocity

- listener orientation

- object orientation and radiation pattern

- object and listener dynamic reverberation effects

- individual low-pass filter and attenuation applied to direct-path sound for each sound source

- individual low-pass filter and attenuation applied to reflected sound for each sound source

- **rendered effects**

- distance (attenuation and reverberation)

- Doppler

- Cartesian 3D positions (x, y, z)

- radiation model

- room effect (early reflections and subsequent surrounding reverberation matching the reverberation response model as described in section Model and Concepts of these guidelines)

- occlusion and obstruction

- **all rendering is real-time and interactive**

- no audible artifacts (no popping, clicking, or other noticeable audio distortion)

- low latency (apparent synchronization of user input, visual activity, and audio activity)

Conclusion and Future Directions

This Level 2.0 guideline, combined with the previous 'Level 1.0' guideline (such as embodied in DirectSound3D for instance), provides a baseline interactive reverberation functionality which can serve a wide range of multimedia and music applications involving immersive 3D audio. Such applications include interactive 3D games, as well as Internet browsers and interactive or recorded music.

An important aspect of these guidelines is an attempt to standardize an artificial reverberator interface that vendors can agree upon for a wide range of multimedia or music applications, without constraining the design of the reverberation algorithm itself.

A key new functionality enabled by these guidelines and relevant to all above applications, is the possibility for sound sources and the listener to move between distinct audio environments and within an environment, with the principal 3D positional and reverberation cues being dynamically updated in accordance with these movements.

Furthermore, the occlusion and obstruction functionalities make it possible to render the muffling of sounds as they travel through or around walls and obstacles, whether they emanate from a different room of from the same environment as the listener's. As a result, sounds can change dramatically as they disappear behind obstacles, appear through an open window, or move from an outside space into a room.

The functionality exposed in these guidelines is limited in complexity to a level that can be enabled on a wide range of platforms offering different amounts of computational resources, without a need to address scalability issues at the API level. This simplifies the use and support of the API because scalability is exclusively handled in the implementation (driver) by the software or hardware vendor (via more or less sophisticated filter topologies or reverberation algorithms). As a result, it is hoped that application developers will be encouraged to take advantage of the proposed environmental enhancements if 3D audio software and hardware vendors widely support these guidelines.

In the future, enhancements of the functionality and user experience could go in two main directions, reviewed below:

Dynamic early reflections

Within a given environment, more refined interactive rendering can be obtained by allowing control of time, amplitude and direction of early reflections for each individual sound source. This enhanced functionality can be enabled by exposing additional low-level parameters (source properties) describing early reflections for each sound source.

An alternative or complementary approach is to provide a higher-level geometrical and physical representation of the environment, from which early reflection parameters can be automatically and dynamically derived according to the movements of the sources or the listener. This approach is particularly intended for environmental rendering of audio/visual sound scenes in a first-person perspective, when a strong connection between the audio and visual environment presentations is desired.

Multiple environments / Effects

Transitions from one environment to another, e. g. as the listener passes from one room to another are handled implicitly by the fact that all reflection and reverberation parameters refer to the listener's room.

In the future, more elaborate control over the transition between environments could be exposed to the application developer or content author ("morphing" instead of "switching").

If the source is located in a different environment than the listener's (occlusion), the reverberation from the source's environment would typically have to be either ignored or "pre-rendered" (added on the source sound). This limits either the realism or the interactivity. Future generations of interactive 3D audio systems are expected to be able to run multiple reverberation engines to render complex sound scenes (where a sound source can be assigned its own environment, different from the listener's). More generally, this possibility extends to the application of effects such as delay, modulation or distortion to individual sounds.

Appendix 1: API Example

General API concepts

The example is presented using Microsoft DirectX (DS3D) Property Sets.

The Two Property Sets

The proposed API comprises two property sets:

1. Listener property set: describes the environment ("preset") by providing settings which apply to any sound source.

2. Source property set: provides adjustments of parameters for each individual source, relative to the setting of the listener properties (i. e. relative to the environment preset).

Specific features:

- Automatic dynamic adjustment of reverberation intensity according to source-listener distance, using a 'Rolloff' parameter as in the 'Level 1.0' guideline.

- Extensive control of reverberation by exposing intensity and delay of early reflections and reverberation as listener properties.

- Special source properties allowing to reproduce the effects of obstacles standing between a source and the listener: Occlusion and Obstruction properties.

API Methods

In this example, implemented under the form of two DirectSound3D property sets (source properties and listener properties), the API comprises the following methods:

- **QuerySupport** can query to see if either property set is supported by the driver (renderer) or to see if an individual property is supported. **QuerySupport** requires the application to pass it the GUID of the property set being queried, and will fail if that property set is not supported by the driver.

- **Set** allows setting a particular property (or group of properties) to a given value (or array of values).

- **Get** allows reading the current value (or values) of a particular property (or group of properties).

Multi-channel Source Buffers

It is desirable that the reverberation engine process multi-channel source signals without mixing them to mono at its input. In DirectSound3D, this is in principle not allowed for a source buffer declared as a 3D buffer. However, this restriction only applies to the direct-path sound. If the 3D mode is disabled on a stereo source buffer, then DS3D plays the left and right channels separately and an I3DL2 compliant driver should preferably not mix the left and right signals to mono when feeding the reverberation engine.

Disabling the Reverberation Engine

The reverberation engine can be disabled (i. e. not use host CPU resources) when the Room listener property (defined in the following) is set to its minimum value (its default value).

Conventions

In the following description of the two DS3D property sets, all intensity levels are defined relative to the intensity level of the source signal, possibly attenuated by DirectSound's Volume parameter, but not by DirectSound's rolloff with distance. DirectSound's Rolloff parameter applies only to the direct path, not to the reflections and reverberation.

All intensity levels or relative attenuations are expressed in hundredth of decibels (millibels, mB). All times are expressed in seconds.

Some properties appear with the same name both as source and listener properties. Whenever this happens, the two values simply combine additively (i. e. the listener property acts as an offset applying to all sources).

Table of DS3D Property Sets

Property Name	Type	Range	Default
Listener property set			
All	array of 12 values		
Room	LONG	[-10000, 0]	-10000 mB
Room_HF	LONG	[-10000, 0]	0 mB
Room_rolloff_factor	FLOAT	[0.0, 10.0]	0.0
Decay_time	FLOAT	[0.1, 20.0]	1.0 s
Decay_HF_ratio	FLOAT	[0.1, 2.0]	0.5
Reflections	LONG	[-10000, 1000]	-10000 mB
Reflections_delay	FLOAT	[0.0, 0.3]	0.02 s
Reverb	LONG	[-10000, 2000]	-10000 mB
Reverb_delay	FLOAT	[0.0, 0.1]	0.04 s
Diffusion	FLOAT	[0.0, 100.0]	100.0 %
Density	FLOAT	[0.0, 100.0]	100.0 %
HF_reference	FLOAT	[20.0, 20000.0]	5000.0 Hz
Source property set			
All	array of 9 values		
Obstruction_all	array of 2 values		
Occlusion_all	array of 2 values		
Direct	LONG	[-10000, 1000]	0 mB
Direct_HF	LONG	[-10000, 0]	0 mB
Room	LONG	[-10000, 1000]	0 mB
Room_HF	LONG	[-10000, 0]	0 mB
Room_rolloff_factor	FLOAT	[0.0, 10.0]	0.0
Obstruction	LONG	[-10000, 0]	0 mB
Obstruction_LF_ratio	FLOAT	[0.0, 1.0]	0.0
Occlusion	LONG	[-10000, 0]	0 mB
Occlusion_LF_ratio	FLOAT	[0.0, 1.0]	0.25

DS3D Property Definitions

<u>Listener Property Set</u>

All

Array of twelve values allowing to set or get the values of all listener properties at once.

Room; Room_HF

Adjust intensity level and low-pass filter for the room effect (i.e. both the early reflections and the late reverberation). Room_HF is defined as the attenuation at high frequencies relative to the intensity at low frequencies.

Room_rolloff_factor

Controls the rolloff of room effect intensity vs. distance, in the same way that DS3D's Rolloff Factor operates on the direct path intensity. Default is 0.0, implying that the reverberation intensity does not depend on source-listener distance. A value of 1.0 means that the reverberation decays by 6 dB per doubling of distance.

Decay_time

Reverberation decay time at low frequencies.

Decay_HF_ratio

Ratio of high-frequency decay time relative to low-frequency decay time. Decay_HF_Ratio would typically control one or several filters located in the feedback loop of the reverberator, whereas Room_HF should control a filter located at the input or output of the reverberator (outside of the feedback loop).

Reflections

Adjusts intensity level of early reflections (relative to Room value).

Reflections_delay

Delay time of the first reflection (relative to the direct path).

Reverb

Adjusts intensity of late reverberation (relative to Room value).

Note: Reverb and Decay_time are independent. If the application adjusts Decay_time without changing Reverb, the driver must keep the intensity of the late reverberation constant.

Reverb_delay

Defines the time limit between the early reflections and the late reverberation (relative to the time of the first reflection).

Diffusion

Controls the echo density in the late reverberation decay. 0% = minimum. 100% = maximum.

Density

Controls the modal density in the late reverberation decay. 0% = minimum. 100% = maximum.

HF_reference

Reference high frequency.

<u>Source property set</u>

All

Array of nine values allowing to set or get the values of all source properties at once (except "all" properties).

Obstruction_all

Array of two values allowing to set or get the values of the two obstruction properties at once.

Occlusion_all

Array of two values allowing to set or get the values of the two occlusion properties at once.

Direct; Direct_HF

Relative adjustment to the direct path's intensity and low-pass filter. This is in addition to the effect of DS3D parameters and of the Occlusion and Obstruction properties.

Note: the value of Direct is allowed to be higher than 0 dB. However, if the combined settings of the different parameters / properties were to result in a gain larger than 0 dB for the direct path, the direct path intensity level would be limited to 0 dB, as DirectSound does not support amplification.

Room; Room_HF

Relative adjustment to the room effect's intensity and low-pass filter. Applies to both reflections and reverberation. This is in addition to the effect of DS3D parameters, Room_rolloff_factor and Occlusion properties.

Note: the value of Room is allowed to be higher than 0 dB. However, if the combined settings of the different parameters/properties were to result in a gain larger than 0 dB for the room effect applied to any source, it would be limited to 0 dB, as DirectSound does not support amplification.

Room_rolloff_factor

Allows control of the rolloff of reverberation intensity vs. distance specifically for an individual source (the settings of Room_rolloff_factor as listener and source properties combine additively).

Obstruction

Main control for Obstruction: allows applying an attenuation and low-pass filtering effect to the direct path. The value of the Obstruction property is equal to the attenuation applied at high frequencies.

Obstruction_LF_ratio

Defines how much Obstruction affects low frequencies, by setting the gain at low frequencies relative to the gain at high frequencies.

Occlusion

Main control for Occlusion: allows applying an attenuation and low-pass filtering effect to the direct path and the room effect. The value of the Occlusion property is equal to the attenuation applied at high frequencies.

Occlusion_LF_ratio

Defines how much Occlusion affects low frequencies, by setting the gain at low frequencies relative to the gain at high frequencies.

Appendix 2: Header Files

The first header file below, '3dl2.h', describes an implementation of the ***required*** part of the Level 2.0 guidelines in DirectSound3D. This implementation uses two GUIDs to identify the functional support of the two property sets (listener property set and source property set). This file defines all the structures that anyone supporting the guideline (vendor or developer) must support to be compliant. The actual file is also available on the IASIG web site (http://www.iasig.org).

3DL2.H - file

```
// 3DL2.H
//

#ifndef _3DL2_H_INCLUDED
#define _3DL2_H_INCLUDED


#ifdef __cplusplus
extern "C" {
#endif // __cplusplus

#pragma pack(push, 4)

// I3DL2 listener property set {DA0F0520-300A-11D3-8A2B-0060970DB011}
DEFINE_GUID(DSPROPSETID_I3DL2_ListenerProperties,
    0xDA0F0520,
    0x300A,
    0x11D3,
    0x8A2B,
    0x0060970DB011);

typedef enum
{
// sets all I3DL2 listener properties
    DSPROPERTY_I3DL2LISTENER_ALL,
// room effect level at low frequencies
    DSPROPERTY_I3DL2LISTENER_ROOM,
// room effect high-frequency level re. low frequency level
    DSPROPERTY_I3DL2LISTENER_ROOMHF,
// like DS3D flRolloffFactor but for room effect
    DSPROPERTY_I3DL2LISTENER_ROOMROLLOFFFACTOR,
// reverberation decay time at low-frequencies
    DSPROPERTY_I3DL2LISTENER_DECAYTIME,
// high-frequency to low-frequency decay time ratio
    DSPROPERTY_I3DL2LISTENER_DECAYHFRATIO,
// early reflections level relative to room effect
    DSPROPERTY_I3DL2LISTENER_REFLECTIONS,
// delay time of first reflection
    DSPROPERTY_I3DL2LISTENER_REFLECTIONSDELAY,
// late reverberation level relative to room effect
    DSPROPERTY_I3DL2LISTENER_REVERB,
// late reverberation delay time relative to first reflection
    DSPROPERTY_I3DL2LISTENER_REVERBDELAY,
// reverberation diffusion (echo density)
    DSPROPERTY_I3DL2LISTENER_DIFFUSION,
// reverberation density (modal density)
    DSPROPERTY_I3DL2LISTENER_DENSITY,
// reference high frequency
    DSPROPERTY_I3DL2LISTENER_HFREFERENCE
} DSPROPERTY_I3DL2_LISTENERPROPERTY;
```

```
// use this structure for DSPROPERTY_I3DL2LISTENER_ALL
// - all levels are hundredths of decibels (mB)
// - all times are in seconds (s)
typedef struct _I3DL2_LISTENERPROPERTIES
{
    LONG lRoom;                      // [-10000, 0]       default: -10000 mB
    LONG lRoomHF;                    // [-10000, 0]       default: 0 mB
    FLOAT flRoomRolloffFactor;       // [0.0, 10.0]       default: 0.0
    FLOAT flDecayTime;               // [0.1, 20.0]       default: 1.0 s
    FLOAT flDecayHFRatio;            // [0.1, 2.0]        default: 0.5
    LONG lReflections;               // [-10000, 1000]    default: -10000 mB
    FLOAT flReflectionsDelay;        // [0.0, 0.3]        default: 0.02 s
    LONG lReverb;                    // [-10000, 2000]    default: -10000 mB
    FLOAT flReverbDelay;             // [0.0, 0.1]        default: 0.04 s
    FLOAT flDiffusion ;              // [0.0, 100.0]      default: 100.0 %
    FLOAT flDensity;                 // [0.0, 100.0]      default: 100.0 %
    FLOAT flHFReference;             // [20.0, 20000.0]   default: 5000.0 Hz
} I3DL2_LISTENERPROPERTIES, *LPI3DL2_LISTENERPROPERTIES;

// property ranges and defaults:

#define I3DL2LISTENER_MINROOM                    (-10000)
#define I3DL2LISTENER_MAXROOM                    0
#define I3DL2LISTENER_DEFAULTROOM                (-10000)

#define I3DL2LISTENER_MINROOMHF                  (-10000)
#define I3DL2LISTENER_MAXROOMHF                  0
#define I3DL2LISTENER_DEFAULTROOMHF              0

#define I3DL2LISTENER_MINROOMROLLOFFFACTOR       0.0f
#define I3DL2LISTENER_MAXROOMROLLOFFFACTOR       10.0f
#define I3DL2LISTENER_DEFAULTROOMROLLOFFFACTOR   0.0f

#define I3DL2LISTENER_MINDECAYTIME               0.1f
#define I3DL2LISTENER_MAXDECAYTIME               20.0f
#define I3DL2LISTENER_DEFAULTDECAYTIME           1.0f

#define I3DL2LISTENER_MINDECAYHFRATIO            0.1f
#define I3DL2LISTENER_MAXDECAYHFRATIO            2.0f
#define I3DL2LISTENER_DEFAULTDECAYHFRATIO        0.5f

#define I3DL2LISTENER_MINREFLECTIONS             (-10000)
#define I3DL2LISTENER_MAXREFLECTIONS             1000
#define I3DL2LISTENER_DEFAULTREFLECTIONS         (-10000)

#define I3DL2LISTENER_MINREFLECTIONSDELAY        0.0f
#define I3DL2LISTENER_MAXREFLECTIONSDELAY        0.3f
#define I3DL2LISTENER_DEFAULTREFLECTIONSDELAY    0.02f

#define I3DL2LISTENER_MINREVERB                  (-10000)
#define I3DL2LISTENER_MAXREVERB                  2000
#define I3DL2LISTENER_DEFAULTREVERB              (-10000)

#define I3DL2LISTENER_MINREVERBDELAY             0.0f
#define I3DL2LISTENER_MAXREVERBDELAY             0.1f
#define I3DL2LISTENER_DEFAULTREVERBDELAY         0.04f

#define I3DL2LISTENER_MINDIFFUSION               0.0f
#define I3DL2LISTENER_MAXDIFFUSION               100.0f
#define I3DL2LISTENER_DEFAULTDIFFUSION           100.0f

#define I3DL2LISTENER_MINDENSITY                 0.0f
#define I3DL2LISTENER_MAXDENSITY                 100.0f
#define I3DL2LISTENER_DEFAULTDENSITY             100.0f

#define I3DL2LISTENER_MINHFREFERENCE             20.0f
#define I3DL2LISTENER_MAXHFREFERENCE             20000.0f
#define I3DL2LISTENER_DEFAULTHFREFERENCE         5000.0f
```

```
// I3DL2 buffer property set {DA0F0521-300A-11D3-8A2B-0060970DB011}
DEFINE_GUID(DSPROPSETID_I3DL2_BufferProperties,
    0xDA0F0521,
    0x300A,
    0x11D3,
    0x8A2B,
    0x0060970DB011);

typedef enum
{
// sets all I3DL2 buffer properties
    DSPROPERTY_I3DL2BUFFER_ALL,
// sets both obstruction properties
    DSPROPERTY_I3DL2BUFFER_OBSTRUCTIONALL,
// sets both occlusion properties
    DSPROPERTY_I3DL2BUFFER_OCCLUSIONALL,
// additional direct path level correction
    DSPROPERTY_I3DL2BUFFER_DIRECT,
// additional direct path high-frequency re. low-frequency level correction
    DSPROPERTY_I3DL2BUFFER_DIRECTHF,
// additional room effect level correction
    DSPROPERTY_I3DL2BUFFER_ROOM,
// additional room effect high-frequency re. low-frequency level correction
    DSPROPERTY_I3DL2BUFFER_ROOMHF,
// like DS3D flRolloffFactor but for room effect
    DSPROPERTY_I3DL2BUFFER_ROOMROLLOFFFACTOR,
// main obstruction control (attenuation at high frequencies)
    DSPROPERTY_I3DL2BUFFER_OBSTRUCTION,
// obstruction low-frequency re. high-frequency ratio
    DSPROPERTY_I3DL2BUFFER_OBSTRUCTIONLFRATIO,
// main occlusion control (attenuation at high frequencies)
    DSPROPERTY_I3DL2BUFFER_OCCLUSION,
// occlusion low-frequency re. high-frequency ratio
    DSPROPERTY_I3DL2BUFFER_OCCLUSIONLFRATIO
} DSPROPERTY_I3DL2_BUFFERPROPERTY;

// use this structure for DSPROPERTY_I3DL2BUFFER_OBSTRUCTIONALL
// - all levels are hundredths of decibels (mB)
typedef struct _I3DL2_OBSTRUCTIONPROPERTIES
{
    LONG lHFLevel;                  // [-10000, 0] default: 0 mB
    FLOAT flLFRatio;                // [0.0, 1.0]  default: 0.0
} I3DL2_OBSTRUCTIONPROPERTIES, *LPI3DL2_OBSTRUCTIONPROPERTIES;

// use this structure for DSPROPERTY_I3DL2BUFFER_OCCLUSIONALL
// - all levels are hundredths of decibels (mB)
typedef struct _I3DL2_OCCLUSIONPROPERTIES
{
    LONG lHFLevel;                  // [-10000, 0] default: 0 mB
    FLOAT flLFRatio;                // [0.0, 1.0]  default: 0.25
} I3DL2_OCCLUSIONPROPERTIES, *LPI3DL2_OCCLUSIONPROPERTIES;

// use this structure for DSPROPERTY_I3DL2BUFFER_ALL
// - all levels are hundredths of decibels (mB)
typedef struct _I3DL2_BUFFERPROPERTIES
{
    LONG lDirect;                   // [-10000, 1000] default: 0 mB
    LONG lDirectHF;                 // [-10000, 0]    default: 0 mB
    LONG lRoom;                     // [-10000, 1000] default: 0 mB
    LONG lRoomHF;                   // [-10000, 0]    default: 0 mB
    FLOAT flRoomRolloffFactor;      // [0.0, 10.0]    default: 0.0
    I3DL2_OBSTRUCTIONPROPERTIES Obstruction;
    I3DL2_OCCLUSIONPROPERTIES Occlusion;
} I3DL2_BUFFERPROPERTIES, *LPI3DL2_BUFFERPROPERTIES;
```

```
// property ranges and defaults:

#define I3DL2BUFFER_MINDIRECT                    (-10000)
#define I3DL2BUFFER_MAXDIRECT                    1000
#define I3DL2BUFFER_DEFAULTDIRECT                0

#define I3DL2BUFFER_MINDIRECTHF                  (-10000)
#define I3DL2BUFFER_MAXDIRECTHF                  0
#define I3DL2BUFFER_DEFAULTDIRECTHF              0

#define I3DL2BUFFER_MINROOM                      (-10000)
#define I3DL2BUFFER_MAXROOM                      1000
#define I3DL2BUFFER_DEFAULTROOM                  0

#define I3DL2BUFFER_MINROOMHF                    (-10000)
#define I3DL2BUFFER_MAXROOMHF                    0
#define I3DL2BUFFER_DEFAULTROOMHF                0

#define I3DL2BUFFER_MINROOMROLLOFFFACTOR         0.0f
#define I3DL2BUFFER_MAXROOMROLLOFFFACTOR         10.f
#define I3DL2BUFFER_DEFAULTROOMROLLOFFFACTOR     0.0f

#define I3DL2BUFFER_MINOBSTRUCTION               (-10000)
#define I3DL2BUFFER_MAXOBSTRUCTION               0
#define I3DL2BUFFER_DEFAULTOBSTRUCTION           0

#define I3DL2BUFFER_MINOBSTRUCTIONLFRATIO        0.0f
#define I3DL2BUFFER_MAXOBSTRUCTIONLFRATIO        1.0f
#define I3DL2BUFFER_DEFAULTOBSTRUCTIONLFRATIO    0.0f

#define I3DL2BUFFER_MINOCCLUSION                 (-10000)
#define I3DL2BUFFER_MAXOCCLUSION                 0
#define I3DL2BUFFER_DEFAULTOCCLUSION             0

#define I3DL2BUFFER_MINOCCLUSIONLFRATIO          0.0f
#define I3DL2BUFFER_MAXOCCLUSIONLFRATIO          1.0f
#define I3DL2BUFFER_DEFAULTOCCLUSIONLFRATIO      0.25f

#pragma pack(pop)

#ifdef __cplusplus
}
#endif // __cplusplus


#endif
```

The second header file below, '3dl2help.h', describes an implementation of the *"programmer's tips"* of the Level 2.0 guidelines in DirectSound3D. A developer may choose to use or ignore this information, because a Level 2.0 compliant driver (renderer) does not need to support the elements contained in this file. This file contains 30 example listener environment presets and 8 example material transmission presets for occlusion/obstruction. The material transmission presets and the most of the environment presets are taken from Creative Labs' EAX. The remaining presets have been created for I3DL2 to provide 'musical' as opposed to 'environmental' reverberation effects. The actual file is also available on the IASIG web site (http://www.iasig.org).

3DL2HELP.H - file

```
// 3DL2HELP.H
//

#ifndef _3DL2HELP_H_INCLUDED
#define _3DL2HELP_H_INCLUDED

// Example listener environment presets
//    for use with DSPROPERTY_I3DL2LISTENER_ALL.
// Each array contains 12 values whose order and types are defined
//    by structure I3DL2_LISTENERPROPERTIES (defined in 3DL2.H).

#define I3DL2_ENVIRONMENT_PRESET_DEFAULT \
-10000,    0,0.0f, 1.00f,0.50f,-10000,0.020f,-10000,0.040f,100.0f,100.0f,5000.0f

#define I3DL2_ENVIRONMENT_PRESET_GENERIC \
 -1000, -100,0.0f, 1.49f,0.83f, -2602,0.007f,    200,0.011f,100.0f,100.0f,5000.0f
#define I3DL2_ENVIRONMENT_PRESET_PADDEDCELL \
 -1000,-6000,0.0f, 0.17f,0.10f, -1204,0.001f,    207,0.002f,100.0f,100.0f,5000.0f
#define I3DL2_ENVIRONMENT_PRESET_ROOM \
 -1000, -454,0.0f, 0.40f,0.83f, -1646,0.002f,     53,0.003f,100.0f,100.0f,5000.0f
#define I3DL2_ENVIRONMENT_PRESET_BATHROOM \
 -1000,-1200,0.0f, 1.49f,0.54f,  -370,0.007f,   1030,0.011f,100.0f, 60.0f,5000.0f
#define I3DL2_ENVIRONMENT_PRESET_LIVINGROOM \
 -1000,-6000,0.0f, 0.50f,0.10f, -1376,0.003f, -1104,0.004f,100.0f,100.0f,5000.0f
#define I3DL2_ENVIRONMENT_PRESET_STONEROOM \
 -1000, -300,0.0f, 2.31f,0.64f,  -711,0.012f,     83,0.017f,100.0f,100.0f,5000.0f
#define I3DL2_ENVIRONMENT_PRESET_AUDITORIUM \
 -1000, -476,0.0f, 4.32f,0.59f,  -789,0.020f,  -289,0.030f,100.0f,100.0f,5000.0f
#define I3DL2_ENVIRONMENT_PRESET_CONCERTHALL \
 -1000, -500,0.0f, 3.92f,0.70f, -1230,0.020f,    -2,0.029f,100.0f,100.0f,5000.0f
#define I3DL2_ENVIRONMENT_PRESET_CAVE \
 -1000,    0,0.0f, 2.91f,1.30f,  -602,0.015f,  -302,0.022f,100.0f,100.0f,5000.0f
#define I3DL2_ENVIRONMENT_PRESET_ARENA \
 -1000, -698,0.0f, 7.24f,0.33f, -1166,0.020f,     16,0.030f,100.0f,100.0f,5000.0f
#define I3DL2_ENVIRONMENT_PRESET_HANGAR \
 -1000,-1000,0.0f,10.05f,0.23f,  -602,0.020f,    198,0.030f,100.0f,100.0f,5000.0f
#define I3DL2_ENVIRONMENT_PRESET_CARPETEDHALLWAY \
 -1000,-4000,0.0f, 0.30f,0.10f, -1831,0.002f, -1630,0.030f,100.0f,100.0f,5000.0f
#define I3DL2_ENVIRONMENT_PRESET_HALLWAY \
 -1000, -300,0.0f, 1.49f,0.59f, -1219,0.007f,    441,0.011f,100.0f,100.0f,5000.0f
#define I3DL2_ENVIRONMENT_PRESET_STONECORRIDOR \
 -1000, -237,0.0f, 2.70f,0.79f, -1214,0.013f,    395,0.020f,100.0f,100.0f,5000.0f
#define I3DL2_ENVIRONMENT_PRESET_ALLEY \
 -1000, -270,0.0f, 1.49f,0.86f, -1204,0.007f,     -4,0.011f,100.0f,100.0f,5000.0f
#define I3DL2_ENVIRONMENT_PRESET_FOREST \
 -1000,-3300,0.0f, 1.49f,0.54f, -2560,0.162f,  -613,0.088f, 79.0f,100.0f,5000.0f
#define I3DL2_ENVIRONMENT_PRESET_CITY \
 -1000, -800,0.0f, 1.49f,0.67f, -2273,0.007f, -2217,0.011f, 50.0f,100.0f,5000.0f
```

```
#define I3DL2_ENVIRONMENT_PRESET_MOUNTAINS \
 -1000,-2500,0.0f, 1.49f,0.21f, -2780,0.300f, -2014,0.100f, 27.0f,100.0f,5000.0f
#define I3DL2_ENVIRONMENT_PRESET_QUARRY \
 -1000,-1000,0.0f, 1.49f,0.83f,-10000,0.061f,   500,0.025f,100.0f,100.0f,5000.0f
#define I3DL2_ENVIRONMENT_PRESET_PLAIN \
 -1000,-2000,0.0f, 1.49f,0.50f, -2466,0.179f, -2514,0.100f, 21.0f,100.0f,5000.0f
#define I3DL2_ENVIRONMENT_PRESET_PARKINGLOT \
 -1000,    0,0.0f, 1.65f,1.50f, -1363,0.008f, -1153,0.012f,100.0f,100.0f,5000.0f
#define I3DL2_ENVIRONMENT_PRESET_SEWERPIPE \
 -1000,-1000,0.0f, 2.81f,0.14f,   429,0.014f,   648,0.021f, 80.0f, 60.0f,5000.0f
#define I3DL2_ENVIRONMENT_PRESET_UNDERWATER \
 -1000,-4000,0.0f, 1.49f,0.10f,  -449,0.007f,  1700,0.011f,100.0f,100.0f,5000.0f


// Examples simulating 'musical' reverb presets
//
// Name  (Decay time)  Description
//
// Small Room  (1.1s)  A small size room with a length of 5m or so.
// Medium Room (1.3s)  A medium size room with a length of 10m or so.
// Large Room  (1.5s)  A large size room suitable for live performances.
// Medium Hall (1.8s)  A medium size concert hall.
// Large Hall  (1.8s)  A large size concert hall suitable for a full orchestra.
// Plate       (1.3s)  A plate reverb simulation.


#define I3DL2_ENVIRONMENT_PRESET_SMALLROOM \
 -1000, -600,0.0f, 1.10f,0.83f,  -400,0.005f,   500,0.010f,100.0f,100.0f,5000.0f
#define I3DL2_ENVIRONMENT_PRESET_MEDIUMROOM \
 -1000, -600,0.0f, 1.30f,0.83f, -1000,0.010f,  -200,0.020f,100.0f,100.0f,5000.0f
#define I3DL2_ENVIRONMENT_PRESET_LARGEROOM \
 -1000, -600,0.0f, 1.50f,0.83f, -1600,0.020f, -1000,0.040f,100.0f,100.0f,5000.0f
#define I3DL2_ENVIRONMENT_PRESET_MEDIUMHALL \
 -1000, -600,0.0f, 1.80f,0.70f, -1300,0.015f,  -800,0.030f,100.0f,100.0f,5000.0f
#define I3DL2_ENVIRONMENT_PRESET_LARGEHALL \
 -1000, -600,0.0f, 1.80f,0.70f, -2000,0.030f, -1400,0.060f,100.0f,100.0f,5000.0f
#define I3DL2_ENVIRONMENT_PRESET_PLATE \
 -1000, -200,0.0f, 1.30f,0.90f,     0,0.002f,     0,0.010f,100.0f, 75.0f,5000.0f


// Example material transmission presets
//   for use with DSPROPERTY_I3DL2BUFFER_OCCLUSIONALL
//   or DSPROPERTY_I3DL2BUFFER_OBSTRUCTIONALL.
// Each array contains 2 values whose order and types are defined
//   by structure I3DL2_OCCLUSIONPROPERTIES
//   or I3DL2_OBSTRUCTIONPROPERTIES (defined in 3DL2.H).

#define I3DL2_MATERIAL_PRESET_SINGLEWINDOW   -2800,0.71f
#define I3DL2_MATERIAL_PRESET_DOUBLEWINDOW   -5000,0.40f
#define I3DL2_MATERIAL_PRESET_THINDOOR       -1800,0.66f
#define I3DL2_MATERIAL_PRESET_THICKDOOR      -4400,0.64f
#define I3DL2_MATERIAL_PRESET_WOODWALL       -4000,0.50f
#define I3DL2_MATERIAL_PRESET_BRICKWALL      -5000,0.60f
#define I3DL2_MATERIAL_PRESET_STONEWALL      -6000,0.68f
#define I3DL2_MATERIAL_PRESET_CURTAIN        -1200,0.15f


#endif
```

References

IASIG 3D Audio Rendering and Evaluation Guidelines, version 1.0 (1998, MMA/IASIG)

Microsoft Direct X Documentation (1995-1998 Microsoft HWDEV)