



Sensaura SDK Release 1.0

12 June 2000

© Copyright 2000 Sensaura Ltd

Sensaura
Dawley Road
Hayes
Middlesex UB3 1HH
UNITED KINGDOM

Telephone: +44 20 8848 9779
Facsimile: +44 20 8848 6565
<http://www.sensaura.com/>

Copyright

© Copyright 2000 Sensaura Ltd

All rights reserved. This product and related documentation are protected by copyright. No part may be reproduced in any form by any means without prior written authorisation of Sensaura Ltd.

Trademarks

Sensaura, the Sensaura logo, Digital Ear, Virtual Ear, MacroFX and ZoomFX are all trademarks or registered trademarks of Sensaura Ltd or Central Research Laboratories Ltd.

Microsoft, DirectX, DirectSound and DirectSound3D are all trademarks or registered trademarks of Microsoft.

Ownership of all other product names and trademarks mentioned herein is acknowledged.

Disclaimer

We reserve the right to alter or amend the information contained in this document and/or the products described herein.

This publication is provided "as is" without warranty of any kind, either express or implied, including, but not limited to, the implied warranties of merchantability, fitness for a particular purpose or non-infringement.

Acknowledgements

The Sensaura SDK documentation was written by Peter Clare, Mike Percy and Adam Philp with suggestions, contributions and proof-reading by other members of the Sensaura team.

Table of contents

1. Introduction	1
1.1 Why a Sensaura SDK?	1
1.2 SDK contents.....	2
1.3 Technical white papers.....	4
1.4 Driver support for ZoomFX	5
1.5 Developer support.....	5
2. ZoomFX property set	6
2.1 Using ZoomFX	6
2.2 API reference	8
3. Other property sets	11
3.1 The Voice Manager	11
3.2 I3DL2 reverb	11
3.3 EAX reverb.....	12
4. Source code samples	13
4.1 DS3DDemo	13
4.2 ZFXCons.....	13
4.3 ZoomFX.....	14

1. Introduction

Welcome to the first release of the Sensaura SDK!

Here, we guide you through the contents of the SDK and then cover the property set APIs supported by Sensaura sound card drivers. We concentrate on the Sensaura ZoomFX property set because that is the main focus for this SDK release. Although ZoomFX provided the impetus for this SDK release there's lots of other good stuff (e.g. technical white papers on 3D audio) in here too!

1.1 Why a Sensaura SDK?

Since Sensaura 3D positional audio technology first appeared on sound cards back in 1997, we have been firm proponents of the open API approach. Proprietary APIs, where there is an open alternative, are simply not in the interests of developers. It is worth noting the important distinction between API and rendering technology. The APIs may be open, but Sensaura is firmly proud and protective of its proprietary rendering technology. In addition to supporting the open APIs, Sensaura has developed unique enhancements to the way DirectSound3D models sounds in 3D. In the case of MacroFX (improved near field model) we were able to do this without introducing a new API. ZoomFX (large area and volumetric sound sources) does require a simple API extension to DirectSound3D.

One of the first questions we have been asked by developers new to Sensaura is: where is your SDK? Until now we have not felt the need to provide a specific Sensaura SDK since the open APIs that we support are all covered elsewhere: DirectSound and DirectSound3D in the DirectX SDK; Voice Manager property set in the DirectX SDK (albeit version 6.0/6.1 only); EAX 1.0 and 2.0 by SDKs available from Creative Labs; I3DL2 reverb from files available from the IA-SIG.

The introduction of the Sensaura ZoomFX API does now provide a reason for needing a Sensaura SDK. This could have simply been a header file plus documentation on ZoomFX and the property set. However, we have taken the opportunity to bundle together technical papers, some example source code and information on other property sets supported by Sensaura drivers. We hope that you will find all this useful. Enjoy!

As a footnote, we should add that we are still firmly behind the open API philosophy. There is nothing to stop other 3D audio technology providers implementing the ZoomFX API (though obviously using a different rendering technique from the proprietary Sensaura ZoomFX algorithm). The ZoomFX API

is basically about defining size and orientation for sound sources – treating them as something other than point sources. We feel that such information is basic and necessary enough that it should be part of the main DirectSound3D API in some future version of DirectX (rather than as a property set add-on). To that end, we are working with other members of the IA-SIG 3D Working Group to get our ideas into some future version of the 3D audio rendering guidelines.

1.2 SDK contents

Sensaura SDK 1.0 contents	
\bin	Utilities and built versions of the samples
\doc	Documentation and technical papers
\identity	Sensaura logos
\include	Property set include files
\lib	Useful libraries
\misc	Miscellaneous items
\sounds	Sample wave files for testing purposes
\src	Source code samples

1.2.1 \bin

This directory contains assorted 3D audio utilities and test programs, along with compiled versions of the source code samples.

Player3D is a simple but very effective application that lets you position a single sound source (in the form of a WAV file) in 3D space whilst allowing full horizontal and vertical positioning. This utility also allows you to add reverb (EAX 1.0, EAX 2.0 or I3DL2), turn MacroFX on or off (via the registry) and set ZoomFX properties. For software buffers, selection of the different 3D rendering algorithms (introduced with Windows 98 SE) is possible.

Athene is a comprehensive DirectSound3D test program. Position multiple sound sources and control all their 3D properties. Includes full support for sound cones (check out the funky graphics!).

Donuts is a "moving objects with sound sources attached" type demo. It includes support for turning MacroFX on or off and applying reverb.

DS3DDemo borrows from both Donuts and the well known arcade game Asteroids. However, in this case full source code is included in the SDK.

ZoomFX is the second of the source code samples. A capable 3D audio test program in its own right this demonstrates use of the ZoomFX property set.

ZFXPlay is a built version of the ZFXCons sample. Just about the simplest possible demo of ZoomFX use. A console application (i.e. no GUI) that can be run from the command line to play a wave file with ZoomFX applied. Note that since this is a console application the SDK installer does not place a link to the executable in the Sensaura SDK folder that is placed in the Windows Start menu.

1.2.2 \doc

This directory contains all the SDK documentation files (including this one) all in Adobe Portable Document Format. You will need the Adobe Acrobat reader (or browser plug-in) to view these files.

1.2.3 \identity

Sensaura logos and icons in various formats. If you have used Sensaura in your game title then you are welcome to use these logos in your product or on the packaging.

1.2.4 \include

C/C++ include files for the property sets supported by the SDK. You will probably want to add this directory to your compiler INCLUDE path.

1.2.5 \lib

SDK library files. Actually, only one file in this SDK release - the Sensaura **ds3dguid** library. If you link with this you can avoid specifying INITGUID in your sources (and hence related problems of unresolved or multiply defined externals). You will probably want to add this directory to your compiler or linker LIB path.

1.2.6 \misc

This directory contains the Microsoft HTML help update. Probably only need be used if you are running Windows 95 and an ancient version of Internet Explorer. Player3D is the only SDK utility that requires HTML help support.

1.2.7 \sounds

This directory contains a collection of sound samples useful for testing 3D positional audio. A "dry" spoken voice sample (**SensauraDescription**) is good

for demonstrating reverb effects. A couple of music samples along with various white and pink noise samples are all useful for showing off the positional capabilities of sound cards (or the Microsoft DirectSound software algorithms).

1.2.8 \src

This directory contains the SDK source code examples. A master Visual C++ version 5.0 workspace file is included. Opening this provides a convenient means to build all of the samples.

1.3 Technical white papers

Sensaura technical papers	
Digital Ear™ technology	devpc003.pdf
MacroFX™ algorithms	devpc004.pdf
An introduction to sound and hearing	devpc005.pdf
MultiDrive™ technology	devpc006.pdf
Virtual audio for headphones	devpc007.pdf
XTC™ crosstalk cancellation	devpc008.pdf
Transaural acoustic crosstalk cancellation	devpc009.pdf
EnvironmentFX™ Audio Environment Modelling	devpc010.pdf
Virtual Ear™ Technology	devpc011.pdf
ZoomFX™ for 3D-sound	devpc012.pdf
Implementing ZoomFX™ 3D objects in 3D space using DS3D	devpc013.pdf
DirectSound tips and traps	devpc014.pdf

In addition to this document which provides overall coverage of the Sensaura SDK and also reference information on the ZoomFX API, the SDK contains additional sources of information. Over the past few years, Sensaura has produced a collection of technical papers. Some provide general background information such as how humans hear in 3D. This information is relevant whatever 3D audio rendering technology you use. Other papers concentrate on specific aspects of the Sensaura technology such as MultiDrive (HRTF rendering over four or more speakers), Virtual Ear, MacroFX and ZoomFX.

1.4 Driver support for ZoomFX

Sensaura drivers have only recently included support for ZoomFX. Due to the time taken for new drivers from Sensaura to make their way through various companies QA procedures and into drivers released to the public the existing Sensaura drivers that you have for your sound card may not support ZoomFX.

Sensaura Ltd does not normally supply drivers direct to end-users. However we are happy to provide driver updates to bona fide developers. Please contact dev@sensaura.com if you require driver support.

1.5 Developer support

For general information and news on Sensaura, please visit our web site: <http://www.sensaura.com/>. Visit the developer section for updates to the utilities, technical documentation and this SDK.

If you have any questions about this SDK or developing for Sensaura then please contact the Developer Relations team on dev@sensaura.com.

Apart the contents of this SDK we are happy to provide the following technical support to developers:

- ❑ Personal visits to on at least an annual basis (failing this we would hope to see developers at events such as GDC, Meltdown, ECTS, etc.).
- ❑ We won't write your sound code for you (although we're happy to provide example code). However, if you are having problems (either general or Sensaura specific) then we're happy to do a walkthrough of your sound code and act as a technical support resource.
- ❑ If you wish, we will test games whilst in their beta or development phase to look for sound related problems. We can offer developers two distinct benefits here. First, we can test on the full range of sound hardware that uses Sensaura. Secondly, we can test with instrumented debug drivers to see what effect the game is having at audio driver level.
- ❑ Working with our chip customers and their customers (i.e. the sound card makers) we can supply developers with the latest sound card hardware and drivers – often before they are available for public release.

2. ZoomFX property set

2.1 Using ZoomFX

Sensaura ZoomFX is a patented technique for rendering sound sources so that they appear to have size (rather than the standard DirectSound3D point source model). Two technical papers, [ZoomFX™ for 3D-sound](#) and [Implementing ZoomFX™ 3D objects in 3D space using DS3D](#), describe ZoomFX and how to apply it to 3D sound buffers. If you have already used other property sets (such as EAX reverb) then the process of acquiring a property set interface and querying support for specific properties will already be familiar to you and you can skip the next few paragraphs.

Property sets are covered in the DirectX SDK documentation, but to recap here are the simple steps needed to gain access to the ZoomFX API.

- [1] Create a hardware 3D sound buffer as usual.
- [2] **QueryInterface** for **IID_IKsPropertySet**.
- [3] On the newly acquired property set interface **QuerySupport** in turn for each of the valid property IDs (**DSPROPERTY_ZOOMFXBUFFER_ALL** etc.) for property set **DSPROPSETID_ZOOMFX_BufferProperties**.
- [4] Perform **Get** or **Set** operations on individual property IDs in the ZoomFX property set as needed.
- [5] Release property set interface when done.

In practice, if many ZoomFX buffers are being created, it is not really necessary to perform step [3] more than once on the first buffer just to verify that the Sensaura driver is indeed a version that supports ZoomFX. It is of course necessary to check return codes and ensure that operations are being performed on a hardware 3D buffer.

2.1.1 The MacroFX effect property

MacroFX adds realistic near-field (i.e. sound source very close to listener) effects to 3D positioned sound sources.

When we first introduced MacroFX, we were able to do so without requiring developers to use a new API. MacroFX (described in [MacroFX™ algorithms](#)) simply made use of the positional information already being supplied via DirectSound3D. This worked out really well. It meant that developers had no new API to test for and learn. Existing games would benefit from MacroFX. No

compatibility problems between Sensaura and other sound card hardware. Just great near field effects on sound cards with Sensaura. All rather neat!

We did provide a global MacroFX control via the registry. Initially simply an on/off toggle and later an effect level. Some sound card vendors chose to expose this in their proprietary control panels. Others left it hidden.

A question frequently asked by developers has been whether application control of MacroFX could be provided on a per buffer basis. We've taken the opportunity to provide this as a property in the ZoomFX property set. Here's how it works...

When a 3D hardware buffer is created it picks up the global MacroFX setting as currently defined in the registry (actually, as it was defined when the DirectSound device was first opened). Then, the MacroFX level can be **Get** or **Set** using the property set interface.

Most applications will never need to use the MacroFX property and MacroFX can normally be left permanently enabled via the sound card control panel. However, the control is now there for applications that wish to use it.

2.1.2 Compatibility with non-Sensaura hardware

If you are using the ZoomFX property set then you will obviously be concerned with compatibility when your game is run on hardware that doesn't support ZoomFX (i.e. older Sensaura drivers and non-Sensaura hardware). As with other property sets (e.g. EAX reverb) an application must acquire the property set interface on a 3D hardware buffer and then **QuerySupport** for specific property sets and their properties. A good tactic would be to do this once, at initialization time, on a "dummy" buffer to test support for ZoomFX.

Later, during game execution, set the ZoomFX object size and orientation properties after a sound buffer has been created. If ZoomFX is not supported, simply miss out this step. It's as simple as that! The sound will still play and be positioned correctly without ZoomFX. It just won't give the feeling of size that ZoomFX adds.

2.2 API reference

2.2.1 All properties

All ZoomFX buffer properties	
<i>Property ID</i>	DSPROPERTY_ZOOMFXBUFFER_ALL
<i>Data type</i>	ZOOMFX_BUFFERPROPERTIES
<i>Range</i>	Any valid collection of individual properties
<i>Default value</i>	ZOOMFXBUFFER_ALL_DEFAULT
<i>Units</i>	(See individual properties)
<i>Operations</i>	Get, Set

This property may be used to get or set all of the ZoomFX properties.

2.2.2 Buffer bounding box

Sound source bounding box	
<i>Property ID</i>	DSPROPERTY_ZOOMFXBUFFER_BOX
<i>Data type</i>	ZOOMFX_BOX
<i>Range</i>	Any valid ZOOMFX_BOX
<i>Default value</i>	ZOOMFXBUFFER_BOX_DEFAULT
<i>Units</i>	Same as other position co-ordinates
<i>Operations</i>	Get, Set

This property is used to describe the size of a sound source. It is a bounding box that encloses the spatial volume from where sound is to be emitted. Once its size has been set it need not be set again, unless the size of the sound emitter should change. The bounding box can be oriented using the DSPROPERTY_ZOOMFXBUFFER_ORIENTATION property of the buffer.

2.2.3 Buffer orientation

Sound source orientation	
<i>Property ID</i>	DSPROPERTY_ZOOMFXBUFFER_ORIENTATION
<i>Data type</i>	ZOOMFX_ORIENTATION
<i>Range</i>	Any valid ZOOMFX_ORIENTATION
<i>Default value</i>	ZOOMFXBUFFER_ORIENTATION_DEFAULT
<i>Units</i>	Same as other position co-ordinates
<i>Operations</i>	Get, Set

This property is used to orient the bounding box. A pair of vectors is used in "Front/Top" configuration, in exactly the same way as the **vOrientFront** and **vOrientTop** vectors in the **DS3DLISTENER** structure (which are set using **IDirectSound3DListener::SetOrientation**). When these vectors are set, they will automatically be normalised to be of unit length and orthogonal to each other. Once set, the normalised versions may be retrieved by getting the property.

Note: the ZoomFX orientation should not be confused with directional sounds as specified by the sound cones of DirectSound3D. The ZoomFX orientation is used to describe the direction that the bounding box is pointing in, not the directionality of the sound.

2.2.4 MacroFX effect

Sound source MacroFX processing effect	
<i>Property ID</i>	DSPROPERTY_ZOOMFXBUFFER_MACROFX_EFFECT
<i>Data type</i>	LONG
<i>Range</i>	0 to 200
<i>Default value</i>	100 (also depends on system setting)
<i>Units</i>	Percent (where 100% is normal)
<i>Operations</i>	Get, Set

This property is used to control the amount of influence Sensaura MacroFX has on an individual sound buffer. Using this property, it is possible to set certain buffers to be unaffected by MacroFX, or alternatively, to accentuate the Near-Field effect provided by MacroFX on certain buffers. The values of **IMacroFx** represent percentage values for the effect with 0 as disabled, 100 representing

"real-world" MacroFX and values above 100 representing an "over-stated" MacroFX effect.

When a 3D buffer is first created, the initial MacroFX effect will be set according to a system-wide setting (taken from the registry). How this registry setting is controlled will depend on the particular sound card. Many sound cards that use Sensaura ship with a control panel application that allows the MacroFX effect to either be turned on or off, or a specific percentage effect level to be set. Even if the global setting is off, this property may be used to force MacroFX to be enabled for a specific buffer.

2.2.5 Data types

2.2.5.1 ZOOMFX_BOX

```
typedef struct
{
    D3DVECTOR vMin;
    D3DVECTOR vMax;
} ZOOMFX_BOX, *LPZOOMFX_BOX;
```

2.2.5.2 ZOOMFX_ORIENTATION

```
typedef struct
{
    D3DVECTOR vFront;
    D3DVECTOR vTop;
} ZOOMFX_ORIENTATION, *LPZOOMFX_ORIENTATION;
```

2.2.5.3 ZOOMFX_BUFFERPROPERTIES

```
typedef struct
{
    ZOOMFX_BOX box;
    ZOOMFX_ORIENTATION orientation;
    LONG lMacroFx;
} ZOOMFX_BUFFERPROPERTIES, *LPZOOMFX_BUFFERPROPERTIES;
```

3. Other property sets

3.1 The Voice Manager

Although never an official part of DirectX, the DirectSound Voice Manager property set has received widespread support from the major 3D audio technology providers. Voice Manager support has been present in Sensaura drivers for over two years now. The Voice Manager has also been received enthusiastically by game developers and has been used in many major titles. However, as a developer of a new game or audio application you do need to think carefully about whether you wish to use or continue to use the Voice Manager.

With DirectX 7, Microsoft implemented a new form of buffer resource management (i.e. the LOCDEFER mechanism). The strong advice from Microsoft is to use this rather than the Voice Manager property set. There are reasons why the Voice Manager might still be the preferred choice (too complicated to go into here – we cover this in one of our technical papers).

The Voice Manager property set will continue to be supported by all Sensaura drivers (WDM as well as VxD) for the foreseeable future. We expect other hardware vendors to do likewise. One simple reason why the Voice Manager isn't going to disappear anytime soon is that, if it were to be removed, this would break many existing game titles.

Since the DirectX 7 SDK no longer includes the Voice Manager header file we have included a version here for completeness. Do think carefully though about whether you wish to use it. For documentation on the Voice Manager you will have to go back to the DirectX 6.0 or 6.1 SDK and look in the **extras** directory.

3.2 I3DL2 reverb

Sensaura drivers provide full support for the I3DL2 reverb API rendered using the EnvironmentFX reverb engine. The first port of call for information should be the IA-SIG web site: <http://www.iasig.org/>. Here, you will find the latest API documentation together with the header files needed to use the API. The site also contains other useful information that should be of interest to the audio developer.

For your convenience, we include a copy of the [I3DL2 specification](#) (revision 1.0a) in this SDK. We also include the header files, **3dl2.h** and **3dl2help.h**, that are needed in order to use the property set.

WARNING: as of this writing, the header files available from the IA-SIG web site contain some minor bugs that will prevent compilation. One of the header files also does a nested include of **dsound.h**. This can be useful, but more often than not leads to problems with INITGUID and the need to instantiate GUIDs in one place only. Consequently, we advise you to use the I3DL2 header files included in this SDK.

3.3 EAX reverb

The Sensaura EnvironmentFX reverb engine, included as part of Sensaura audio drivers, supports the EAX 1.0 and 2.0 reverb property set APIs. Documentation, samples and header files for these property sets are not included in this SDK. If you wish to use these property sets then SDKs for each of EAX 1.0 and EAX 2.0 are available for public download at Creative Labs' developer web site: <http://developer.creative.com/>.

4. Source code samples

Full source code is provided for each of the samples covered in this section. Each sample directory includes a Visual C++ project file that can be used to build it. A master workspace file, **examples.dsw**, is provided in the main source directory. This can be used to build all of the source code samples.

To provide maximum compatibility for those developers still using Visual C++ 5.0, all the project and workspace files have been created using that version. No problem though for users of VC++ 6.0. The files are upward compatible – simply let Visual C++ convert where necessary.

4.1 DS3DDemo

The purpose of this demo is to show how easily you can add 3D positional audio to your games and other programs, simply by using the Microsoft DirectSound3D API.

All Sensaura sound cards support the DirectSound3D API, providing hardware acceleration of 3D positional audio to give you sounds in front, behind, above and below you. If you write your games using DirectSound3D, not only does the alien fighter fly off the top of the screen, the sound of its engine screams over the player's head. Hear the roar of the other cars and the crowd all around you as you cross the finish line in the Grand Prix. Get all this without hitting your CPU, because your Sensaura sound card is doing all the work. Even if you don't have a Sensaura sound card, DirectSound3D can take advantage of any other 3D audio hardware that may be available, so you know that your game will run on the broadest possible base of 3D-accelerated sound hardware.

For more information, see the **readme.txt** file included with the sources.

NOTE: this demo program requires runtime access to a number of X-Files (e.g. **bullet.x**). These are provided in the main **\bin** directory along with the supplied built version of DS3DDemo. Either copy your newly built version of the demo into **\bin** or copy the X-Files into the directory containing the newly built executable.

4.2 ZFXCons

A simple console application that will play a wave file with ZoomFX enabled. A starting point for the first time ZoomFX developer to show how easy it is to use.

To successfully run this driver you will need to be using a recent Sensaura driver that supports ZoomFX.

4.3 ZoomFX

A utility program that allows creation of multiple 3D buffers and the setting of their 3D properties in addition to the ZoomFX bounding box and orientation.

To successfully run this driver you will need to be using a recent Sensaura driver that supports ZoomFX.