



# OpenAL Deployment Guide (PC Windows)

Version 0.16



### **Revision History**

0.14	April 2007	Peter Harrison, Andrea D'Orta and Daniel Peacock
0.15	May 2007	Peter Harrison, Andrea D'Orta and Daniel Peacock
0.16	June 2007	Peter Harrison, Daniel Peacock

### **Copyright and Trademarks**

©2007 Creative Technology Limited. All rights reserved. Creative, Sound Blaster, and the Creative logo are registered trademarks, and Environmental Audio, EAX, and the Environmental Audio Extensions logo are trademarks of Creative Technology Ltd. in the United States and/or other countries.

All other brands and product names listed are trademarks or registered trademarks of their respective holders.

## Contents

Contents.....	3
Introduction .....	4
OpenAL on PC .....	4
Document scope .....	4
Basics of OpenAL PC deployment .....	5
Use the Enumeration Extension.....	5
Use the OpenAL re-distributable.....	5
Advanced Topics .....	6
PC OpenAL Devices .....	6
OpenAL Router .....	6
Native OpenAL .....	6
Generic Hardware .....	7
Generic Software.....	7
OS Differences.....	8
OpenAL on Windows XP.....	8
OpenAL on Windows Vista.....	8
OpenAL Device Enumeration .....	10
The default device .....	10
Enumerating Capture devices .....	10
Note about new "Enumerate All" extension.....	10
Creative OpenAL re-distributable.....	11
OpenAL re-distributable command-line parameters .....	11
OpenAL re-distributable license .....	11
Contacts .....	13

## ***Introduction***

OpenAL is the most popular and effective programming interface for the development of interactive 3D audio. Today, many hardware manufacturers, platform holders and middleware providers are creating audio rendering technologies which conform to the OpenAL specs. This allows applications developers to write their audio systems using OpenAL, safe in the knowledge that their work will be re-usable across most major platforms.

### **OpenAL on PC**

In contrast to platforms with fixed hardware configurations such as games consoles, the Windows platform is particularly varied in terms of audio capabilities. Audio solutions on a user's PC could range from a high-end gaming soundcard with full hardware accelerated OpenAL support to a basic 2 channel sound chip on the motherboard with no 3D audio driver support.

This variation in capabilities is both a blessing and a curse for application developers. OpenAL's scalability means that it is simple to make the best use of any audio DSP features present without resorting to writing custom code for particular configurations. However, different implementations of OpenAL must exist to cater for different audio solutions – versions that talk directly to high-end audio hardware and versions that emulate OpenAL using host processing. And two or more of these implementations might well co-exist on a given system.

How can the developer be sure that the OpenAL implementations on a user's PC are up to date? And how can they enable the user to switch between implementations? This document holds the answers to these and many other questions.

### **Document scope**

The [first section](#) of the document is a basic set of instructions on implementing a robust and effective OpenAL deployment / configuration strategy for your application.

If you want to know more about the nuts and bolts, then read on to the [second section](#). We'll look closely at different [PC OpenAL devices](#), plus the [Enumeration Extensions](#) that allow the developer to retrieve a list of OpenAL implementations present on the system.

If you are planning to release a PC title taking advantage of OpenAL 3D sound, then the deployment guide will help you ensure that the installation and configuration process is as painless as possible.

## ***Basics of OpenAL PC deployment***

This section presents basic advice on how to deploy OpenAL with your application.

### **Use the Enumeration Extension**

The OpenAL Enumeration Extension is part of the OpenAL 1.1 specification. This API extension is very simple to use, and allows the application developer to retrieve strings containing the name of each OpenAL device that exists on a user's PC.

We recommend that you allow the end-user to choose which OpenAL device will render the application audio. Ideally, a list of available devices retrieved using the Enumeration Extensions should be available in the application's configuration dialog or audio options menu.

Don't try to second guess the string names of OpenAL devices that might be present, or to restrict the users' choice based on the devices that were detected at install time. This gives plenty of scope for problems – what if the user obtains and installs a new OpenAL compatible hardware device, or somehow disables an existing device?

Further details of the [Enumeration Extension](#) can be found later in this guide. Documentation and sample code are shipped with the latest OpenAL SDK.

### **Use the OpenAL re-distributable**

Creative Labs develop and maintain a set of OpenAL implementations including [Generic Software](#), [Generic Hardware](#) and a [routing mechanism](#). To help developers with deployment, we also provide a [re-distributable package](#). This is an installer for the latest OpenAL components (Generic Hardware implementation, Generic Software implementation, and router).

Download the redistributable from <http://developer.creative.com>, and integrate it into your application installer. The redistributable installs OpenAL components onto the user's system, while ensuring that any newer files present are not over-written.

## ***Advanced Topics***

### **PC OpenAL Devices**

In OpenAL, the term 'Device' is used interchangeably with 'Implementation' or 'Renderer'. It means a software component or software/hardware package capable of understanding and responding to OpenAL API calls to render an audio mix.

On PC, there are currently three main device categories: [Native](#), [Generic Hardware](#) and [Generic Software](#). The [OpenAL router](#) takes care of identifying the devices present on a user's system and passing run-time calls to the selected implementation.

In the sections below, we'll start by looking at the router, and then in detail at each device category. Later you can see diagrams showing the way the components fit together on Windows [XP](#) and [Vista](#).

### **OpenAL Router**

The router component is an interface between the OpenAL API and all the OpenAL compatible devices that are present on the user's system. When the programmer uses one of the [enumeration extensions](#) to query for devices, the router is responsible for identifying implementations and passing back their names. Once the application opens a device and starts interacting with it, the router takes care of passing calls down to the appropriate DLL.

The router resides in OpenAL32.dll. Windows path behaviour means that if you link to OpenAL.lib (or use LoadLibrary without specifying a path) the application will first look for an OpenAL32.dll in the current directory. If the library is not found, the next location will be the windows system folder (the precise folder will depend on the location of the Windows installation and whether the OS / application is 32 or 64 bit).

Developers are advised to employ the standard policy for OpenAL deployment – the OpenAL [Re-distributable package](#). This installer will ensure that the most up-to-date versions of the OpenAL libraries, including OpenAL32.dll are present in the windows system folder, no matter which Windows version is running.

It is possible to just place a copy of OpenAL32.dll and other OpenAL components in the same directory as your application's main executable. However, doing this will mean that without your own intervention, your application cannot benefit from improved versions of OpenAL components released subsequently.

### **Native OpenAL**

A native OpenAL implementation talks directly to OpenAL-compatible hardware. For instance, most SoundBlaster Audigy and X-Fi series soundcards support OpenAL natively.

Native OpenAL implementations are tightly tied to hardware drivers. On Creative soundcards, they are implemented inside ct\_oal.dll, which is placed in the windows system folder by the soundcard driver installation / update script. Similarly, NVIDIA motherboards with native OpenAL support ship the implementation inside nvoal.dll.

Because native OpenAL implementations are part of a hardware driver package, only the hardware vendor can update or change the library files. Applications and end-users should never change these files – a particular version of ct\_oal.dll may very well not work properly or at all with a different hardware driver.

When enumerating OpenAL devices using the [Enumeration extension](#), a native implementation will appear with the name of the audio device. E.g.

"SB X-Fi [E800]"

"SB Audigy2ZS [A400]"

“NVIDIA® nForce™ audio”

## Generic Hardware

Although most high-performance 3D audio hardware today supports OpenAL natively, there is still a legacy of hardware which is only capable of rendering 3D audio using the DirectSound3D API on Windows XP. Fortunately it is possible to translate OpenAL calls into DirectSound3D with only a small reduction in performance and quality. To allow the end user to benefit from such hardware, Creative maintains the “Generic Hardware” device. This implementation is also sometimes known as the “wrapper” or “DirectSound3D Wrapper” device.

The Generic Hardware implementation is found alongside the [Generic Software](#) implementation, in wrap\_oal.dll. The OpenAL [re-distributable package](#) will ensure that the most up-to-date version(s) of wrap\_oal.dll are present in the windows system folder(s), no matter which Windows version is running.

In the [device enumeration](#) string, the identifier for this device will usually contain the text “Generic Hardware”.

## Generic Software

Not all users own a 3D audio accelerator card. But Windows application developers require solutions that work on every PC. So Creative maintains a fallback implementation of OpenAL that works everywhere - the Generic Software device. This device features basic but high-quality OpenAL functionality, rendered using the host CPU. The Generic Software device is also known as the “Software Renderer” or “Software Mixer” device.

If the user’s system supports DirectSound, then the Software renderer will output into a DirectSound buffer. If not, the renderer outputs to Windows Multimedia services wave output (also known as mmsystem).

The Generic Software implementation is found alongside the [Generic Hardware](#) implementation, in wrap\_oal.dll. The OpenAL [re-distributable package](#) will ensure that the most up-to-date version(s) of wrap\_oal.dll are present in the windows system folder(s), no matter which Windows version is running.

In the [device enumeration](#) string, the identifier for this device will usually contain the text “Generic Software”.

## OS Differences

OpenAL's architecture has changed somewhat with the release of Windows Vista. This section explains how the changes effect deployment of applications with OpenAL components.

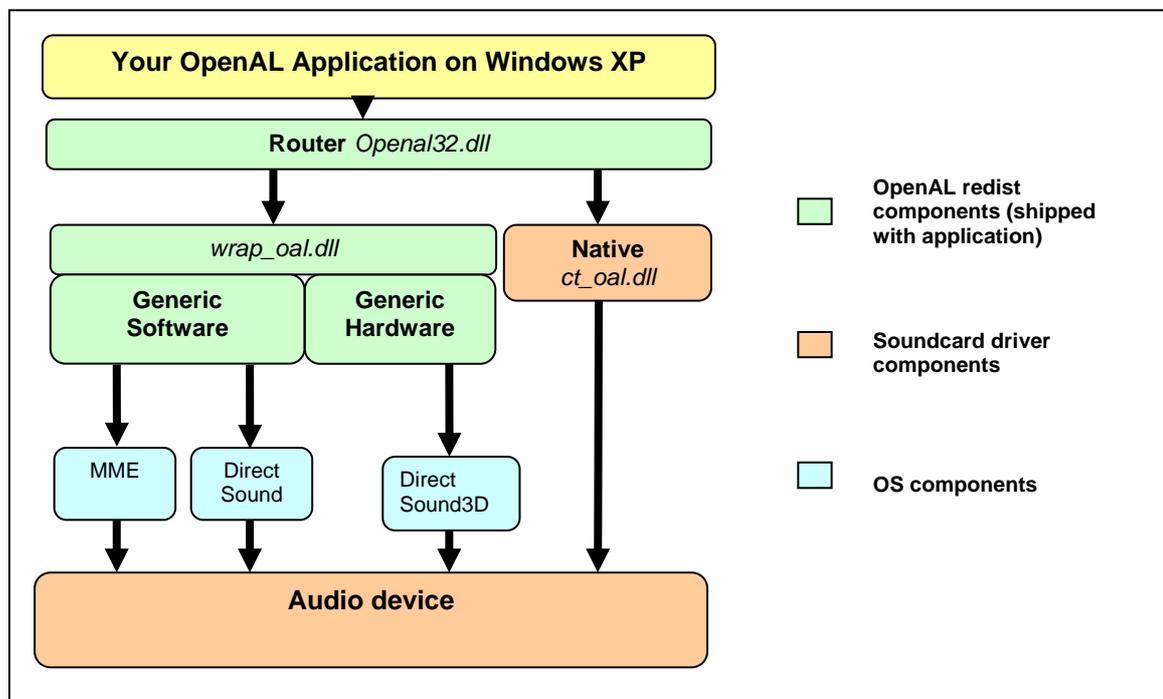
Windows Vista does not inherit the hardware abstraction layer for audio that was present under previous versions of Windows, including Windows XP. This means that there will be no more hardware acceleration of DirectSound™ and DirectSound3D although these APIs will continue to exist.

The [Native OpenAL](#) devices on Sound Blaster Audigy and Sound Blaster X-Fi™ soundcards do not require DirectSound or DirectSound 3D and so they will be completely unaffected. The [Generic Hardware](#) device will no longer be available, as it requires the use of hardware DirectSound 3D Buffers.

With Microsoft's decision to remove the audio hardware layer in Windows Vista, legacy DirectSound 3D applications will no longer be able to benefit from hardware DSP providing audio spatialization and effects. Instead they will have to rely on the new Microsoft software mixer that is built into Windows Vista. In some cases, where an application specifically looks for a hardware audio path, it may even fall back to plain stereo output. All interactive processing and rendering will be dependent on the host CPU. If you want your application to take advantage of hardware 3D audio on supported audio devices, you must use OpenAL and support Native devices through [device enumeration](#).

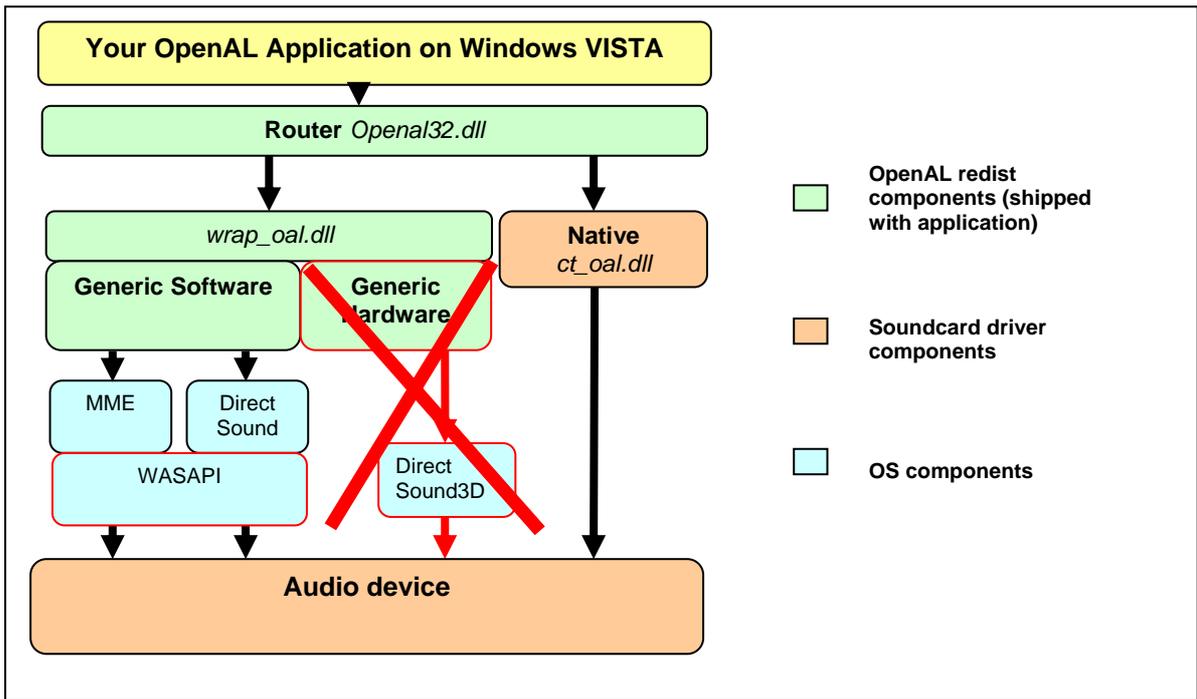
## OpenAL on Windows XP

The diagram below shows how OpenAL is structured on a Windows XP system. Note the [different renderers](#) as described in the previous section.



## OpenAL on Windows Vista

The critical change in Windows Vista is that the [Generic Hardware](#) device is no longer available.



## OpenAL Device Enumeration

OpenAL features two extensions for device enumeration. They enable developers to query the router for the names of the different devices available on a user's system.

The OpenAL Programmer's Guide includes full documentation for the Enumeration Extensions. Additionally the OpenAL SDK includes sample code describing exactly how to implement enumeration, so we won't go into code detail here.

Essentially, the extensions allow the developer to retrieve a set of null-terminated strings containing the names of the devices found on the system. To initialise OpenAL, you need to create an OpenAL device. The call to create a device – `alcOpenDevice()` - takes a string as input. To explicitly request one of the enumerated devices, just pass the corresponding string.

### The default device

To initialise OpenAL without specifying a device, pass NULL; the router will select the 'best' device according to the following rules:

- 1) Is there a [native implementation](#) available which matches the default output device as reported by Windows? If so, that will be the default device.
- 2) Can we use the [Generic Hardware](#) device? If the underlying DirectSound3D device meets a minimum threshold, it will be the default device.
- 3) Otherwise, can we use the [Generic Software](#) device? If not, then `alcOpenDevice` returns NULL.

### Enumerating Capture devices

The OpenAL 1.1 specifications describe the Capture extension, which makes it possible for an OpenAL application to capture audio, for instance from the system's microphone input. As with rendering devices, it is necessary to create a capture device before OpenAL can start capturing audio inputs. Enumeration can be carried out to identify capture devices in the same manner.

### Note about new "Enumerate All" extension

Since the release of the Creative's OpenAL SDK version 3.03, a new variant of the enumeration extension is available. This new extension, identified as "ALC\_ENUMERATE\_ALL\_EXT", caters for application developers who need to identify and use any of the audio outputs available on the user's system (something that was not possible with the original Enumeration extension).

OpenAL devices such as [Generic Software](#) and [Generic Hardware](#) are capable of rendering to one of several different audio outputs. For instance, where the user has more than one audio device installed, or in Windows Vista where the audio system identifies different "endpoints" for sound output such as Speakers, S/PDIF, etc...

In such a situation, the new enumeration extension identifies all possible outputs, allowing the developer to give the user a more complete choice of playback devices.

NOTE : This extension is in addition to the original "Enumeration" extension; if a developer is only concerned with playback on the default playback device then there is no need to use this new Enumerate All extension, the original Enumeration extension is sufficient.

## **Creative OpenAL re-distributable**

The Creative OpenAL re-distributable will ensure that the latest Creative OpenAL [router](#), [Generic Software](#) and [Generic Hardware](#) renderers and are present on the user's system.

The re-distributable ensures that newer components are never overwritten, and is designed to work with all current 32-bit and 64-bit versions of [Windows XP](#) and [Windows Vista](#). Creative's testing teams work hard to make sure that new versions of the OpenAL implementations maintain compatibility with legacy OpenAL titles. This means that if a user has multiple OpenAL games installed, they can all share one set of OpenAL libraries. And installing an OpenAL title featuring the re-distributable should never result in a previously installed game breaking.

You can download the re-distributable package from the Creative Labs Developer Relations website (<http://developer.creative.com>, Games -> Dev Kits and Tools -> OpenAL -> OpenAL Installer for Windows). As its name suggests, you can freely ship this package with your game. Read carefully the [license agreement](#) which is displayed when you invoke the installer – note that it is possible to run the installer silently with the command line parameter `-s`.

### **OpenAL re-distributable command-line parameters**

The OpenAL Installer supports the following command-line parameters:

**`/R or /r or -R or -r`**

Return error codes to report OpenAL installation issues -- 0 for no issues (default), 1 for "reboot required", and 2 for "write access denied".

**`/S or /s or -S or -s`**

Run the installation in silent mode. Nothing will be displayed.

### **OpenAL re-distributable license**

Creative Labs, Inc. is providing you with this OpenAL32.dll installer and other OpenAL files ("Software"). You may use and freely integrate with your software applications and distribute such throughout the world at no cost or further obligation to Creative.

#### **NO WARRANTY**

ANY USE BY YOU OF THE SOFTWARE IS AT YOUR OWN RISK. THE SOFTWARE IS PROVIDED FOR USE "AS IS" WITHOUT WARRANTY OF ANY KIND. TO THE MAXIMUM EXTENT PERMITTED BY LAW, CREATIVE DISCLAIMS ALL WARRANTIES OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, IMPLIED WARRANTIES OR CONDITIONS OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. CREATIVE IS NOT OBLIGATED TO PROVIDE ANY UPDATES OR UPGRADES TO THE SOFTWARE.

No other entity or person is authorized to expand or alter this warranty or any other provisions herein. Creative does not warrant that the functions contained in the Software will meet your requirements or that the operation of the Software will be uninterrupted or error-free or free from malicious code. For purposes of this paragraph, "malicious code" means any program code designed to contaminate other computer programs or computer data, consume computer resources, modify, destroy, record, or transmit data, or in some other fashion usurp the normal operation of the computer, computer system, or computer network, including viruses, Trojan horses, droppers, worms, logic bombs, and the like.

You assume full responsibility for the selection of the Software to achieve your intended results, and for the downloading, use and results obtained from the Software. You also assume the entire risk as it applies to the quality and performance of the Software.

IN NO EVENT WILL CREATIVE'S LIABILITY TO YOU OR ANY OTHER PERSON EVER EXCEED THE AMOUNT PAID BY YOU TO USE THE SOFTWARE, REGARDLESS OF THE FORM OF THE CLAIM.

## **Contacts**

### **Creative Labs Developer Relations**

<http://developer.creative.com>

<http://developer.creative.com/contactus/contactus.asp>

### **OpenAL**

<http://www.openal.org>

<http://www.openal.org/lists.html>