Creative Labs, Inc.
1901 McCarthy Blvd.
Milpitas, CA 95035

Phone (408) 428 6600
Fax (408) 428 6611
www.creative.com

Sound Blaster technologies in FMOD

A programmer's guide to accessing Sound Blaster hardware acceleration via the FMOD music and sound effects system

Sound Blaster technologies in FMOD

A programmer’s guide to accessing Sound Blaster hardware acceleration via the FMOD music and sound effects system

Revision History

0.1	September 21, 2007	Carlo Vogelsang, Peter Harrison
1.0	January 22, 2008	Peter Harrison

Trademarks and Service marks

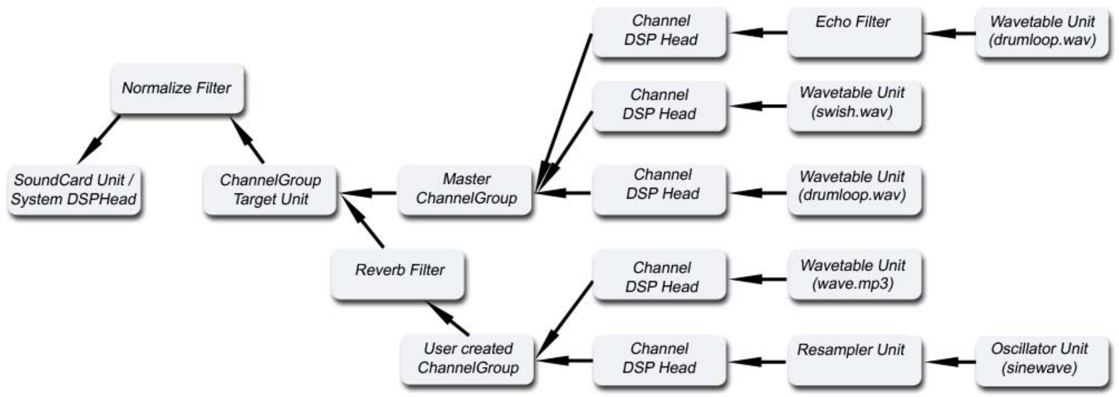
Table of Contents

Sound Blaster Technologies in FMOD

FMOD is a popular and powerful cross-platform interactive audio system. In the past, FMOD supported hardware accelerated audio on the PC platform through the DirectSound3D interface. However, under Windows Vista, DirectSound3D hardware support is no longer available. Instead, audio hardware manufacturers such as Creative Labs have adopted OpenAL as a direct interface to hardware features. FMOD has been modified to support OpenAL rendering, this document provides developers with hints and tips to ensure a first-rate experience for gamers playing FMOD titles on OpenAL hardware.

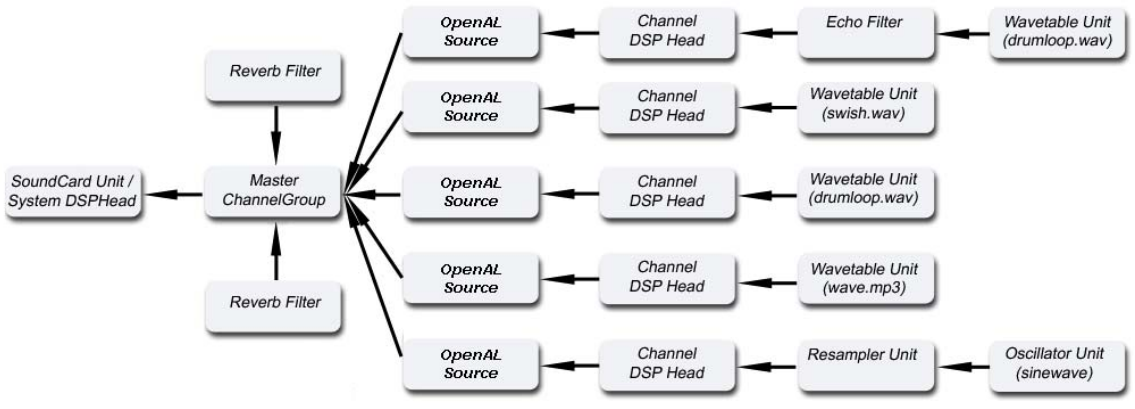
FMOD's OpenAL Output

FMOD supports a flexible, graph-based DSP architecture.


Channel DSP Head
Echo Filter
Wavetable Unit (drumloop.wav)
Normalize Filter
Channel DSP Head
Wavetable Unit (swish.wav)
SoundCard Unit / System DSPHead
ChannelGroup Target Unit
Master ChannelGroup
Channel DSP Head
Wavetable Unit (drumloop.wav)
Reverb Filter
Channel DSP Head
Wavetable Unit (wave.mp3)
User created ChannelGroup
Channel DSP Head
Resampler Unit
Oscillator Unit (sinewave)

The data flows from the right-hand side, the **Wavetable Units**, to the left-hand side the **SoundCard Unit**. In between is a network of DSP blocks including **Filters**, and **Channel Groups** which mix together multiple **Channels**.

In order to get maximum benefit from OpenAL hardware acceleration, certain parts of the DSP graph had to be slightly simplified as shown below:



The most noticeable difference is the absence of the '*User created Channel Group*'. **Channel Groups** are not available in combination with hardware acceleration. Each **Channel Group** is basically a sub-mixer that mixes together multiple **Channels** into a single output which can then have effects applied to it, the most common usage is to apply one effect to all **Channels** in a **Channel Group**, for example an "*underwater*" Channel Group that has a low-pass filter applied to it's output. Of course, the same effect can be achieved by directly applying a low-pass filter to each Channel.

The rest of this document will describe some programming tips to allow for easy portability between the FMOD OpenAL Hardware Output and the FMOD Software Output.

Initializing FMOD

Initializing the FMOD system to use OpenAL is similar to using any other backend - we just need to select the OpenAL output type:

```
FMOD::System *pFMOD = NULL;
FMOD_RESULT result = FMOD_OK;

// Make FMOD system
result = FMOD::System_Create(&pFMOD);
if (result != FMOD_OK)
  printf("FMOD::System_Create succeeded\n");
else
  // ERROR

// Select OpenAL output
result = pFMOD->setOutput(FMOD_OUTPUTTYPE_OPENAL);
if (result != FMOD_OK)
  printf("pFMOD->setOutput(FMOD_OUTPUTTYPE_OPENAL) succeeded\n");
else
  // ERROR
```

Now that the OpenAL output type is selected, the application needs to find out what OpenAL output devices are available on the system. Ideally, this list of OpenAL output devices can be exposed to the user in the audio options menu:

```
int numdrivers = 0;
char drivername[256];

// Enumerate OpenAL drivers
result = pFMOD->getNumDrivers(&numdrivers);
for (int i = 0; i < numdrivers; i++)
{
  result = pFMOD->getDriverName(i,drivername,sizeof(drivername));
  printf("pFMOD->getDriverName(%d) returns: %s\n",i,drivername);
}

// Select OpenAL driver
// -1 = primary or main sound device as selected by the OS settings
result = pFMOD->setDriver(-1);

// Initialize FMOD
// 200 = virtual channel count, can be higher than actual channels
result = pFMOD->init(200, FMOD_INIT_NORMAL, 0);

// Now check hardware channels ACTUALLY available
int iNum2D = 0, iNum3D = 0, iNumTotal = 0;
result = pFMOD->getHardwareChannels(&iNum2D, &iNum3D, &iNumTotal);
```

These are the basic steps required to get FMOD to use the OpenAL output, enabling hardware acceleration of 3D audio channels. The next section will explain how to create sounds and streams so that they will actually take advantage of hardware processing where available.

Creating hardware sounds and streams

The code below creates a basic 3D looping sound that will use hardware acceleration if available:

```
FMOD::Sound *pSound = NULL;

// Create looping 3D sound
Result = pFMOD->createSound("footsteps.wav",
FMOD_3D|FMOD_LOOP_NORMAL, 0, &pSound);
```

With the OpenAL output type it is possible to use compressed samples, such as MP2, MP3 or IMAADPCM, that are decoded in real-time and still get passed into the hardware 3D audio pipeline:

```
FMOD::Sound *pCompressedSound = NULL;

// Create looping 3D sound
Result = pFMOD->createSound("footsteps.mp3",
FMOD_3D|FMOD_CREATECOMPRESSEDSAMPLE|FMOD_LOOP_NORMAL, 0,
&pCompressedSound);
```

Additionally, **Channel** based DSP effects can also be combined with hardware acceleration. Here, an echo effect is applied to the 3D hardware sound (pSound) created in the previous example:

```
FMOD::DSP *pDSPEcho = NULL;
FMOD::Channel *pChannel = NULL;

// Create a DSP by type
result = pFMOD->createDSPByType(FMOD_DSP_TYPE_ECHO, &pDSPEcho);
if (result != FMOD_OK)
  printf("FMOD->createDSPByType(FMOD_DSP_TYPE_ECHO) succeeded\n");
else
  // ERROR

// Allocate channel for sound
result = pFMOD->playSound(FMOD_CHANNEL_FREE, pSound, true,
&pChannel);
if (result != FMOD_OK)
  printf("pFMOD->playSound() succeeded\n");
else
  // ERROR

// Add DSP to channel
result = pChannel->addDSP(pDSPEcho);
if (result != FMOD_OK)
  printf("pChannel->addDSP(Echo) succeeded\n");
else
  // ERROR
```

The only caveat for DSP effects are **ChannelGroup** based DSP effects which cannot be combined with hardware acceleration:

```
FMOD::DSP *pDSPEcho = NULL;
FMOD::Channel *pChannel = NULL;
FMOD::ChannelGroup *pChannelGroup = NULL;

// Now set up a channel group
result = pFMOD->createChannelGroup("UserGroup", &pChannelGroup);
if (result != FMOD_OK)
  printf("pFMOD->CreateChannelGroup(UserGroup) succeeded\n");
else
  // ERROR

// Create a DSP by type
result = pFMOD->createDSPByType(FMOD_DSP_TYPE_ECHO, &pDSPEcho);
if (result != FMOD_OK)
  printf("pFMOD->createDSPByType(FMOD_DSP_TYPE_ECHO) succeeded\n");
else
  // ERROR

// Allocate channel for sound
result = pFMOD->playSound(FMOD_CHANNEL_FREE, pSound, true,
&pChannel);
if (result != FMOD_OK)
  printf("pFMOD->playSound() succeeded\n");
else
  // ERROR

// Attach channel to channelgroup
result = pChannel->setChannelGroup(pChannelGroup);
if (result != FMOD_OK)
  printf("pChannel->setChannelGroup(UserGroup) succeeded\n");
else
  // ERROR

// Add DSP to channelgroup – will NOT work with openal channels
result = pChannelGroup->addDSP(pDSPEcho);
if (result != FMOD_OK)
  printf("pChannelGroup->addDSP(Echo) succeeded\n");
else
  // ERROR
```

The above example will NOT work with the OpenAL output type, the echo effect added to the **ChannelGroup** will not be heard. This is because the '*User Created ChannelGroup*' is not present in the OpenAL DSP graph. Most of the time, the same effect can be achieved by adding the DSP effect to each channel directly.