**Samples,bytes and milliseconds**

单位长度下一个左声道+右声道叫做一个sample

如果是16位立体声，1 sample = 16bits = 4bytes

如果该音频是44.1khz，那么就是每秒钟有44100个sample，那么一个sample就是1/44100秒

函数Sound::getLength会提供声音的毫秒，bytes和samples。

**Samples,Compressed samples and streams**

当声音被加载时，它要么作为静态样本作为PCM（samples）被解压缩到内存中，要么以其本机格式加载到内存中并在运行时被解压缩（compressed samples），要么从外部媒体（如硬盘或CD（streams））实时流式传输和解码（以块为单位）。

Samples:适用于需要一次播放多次的小声音，例如声音效果。它们通常只需要很少或根本不需要CPU来回放，并且可以硬件加速。

Streams:“Streams”适用于太大而无法放入内存，需要从磁盘流到FMOD管理的小环形缓冲区的大声音。根据文件格式，它们占用少量的CPU和磁盘带宽。例如，MP3比PCM解压缩的wav文件需要更多的cpu能量来实时解码。流媒体声音一次只能播放一次，不能像一个示例一样一次多次生成，因为每个流只有一个文件句柄和一个要解码的环形缓冲区。

Compressed samples:“Compressed samples”是一个选项，允许用户加载特定的压缩文件格式（例如当前的IMA ADPCM、FADPMCM、Vorbis、MP2、MP3、AT9和XMA格式）。FADPCM/Vorbis/AT9仅通过.FSB容器格式受支持，并将它们压缩在内存中而不解压缩。它们是混合在CPU上的软件，没有流的“仅一次”限制。它们比标准的PCM样本占用更多的cpu，但实际上由于没有进行任何磁盘访问和更小的内存缓冲区，因此比流占用更少的cpu。

System::createSound函数默认会去解压整个音频放入内存（就是samples）

**Channels and sounds**

当你load声音后，可以调用System::playSound来播放它，这个函数会返回一个指向Channel的指针。

FMOD将会自动选择一个channel来用作声音的播放，你不需要去管理channels。

**Sub-mixing and ChannelGroups**

如果你不想对每个单独的channel处理或控制，channel可以放置在ChannelGroup中。ChannelGroup允许你对一组channels进行同一操作，并控制音频的混合图。

ChannelGroup也可以被认为是“sub mix”，就像在信号链中一样，这将是信道信号混合到单个缓冲器中的点。然后，可以使用DSP效果（见下文）对该缓冲区进行一次处理，而不是对每个通道进行一次处理，从而节省大量的CPU时间。

2D and 3D

三维声源是一个在空间中具有位置和速度的channel。当一个3Dchannel正在播放时，它的音量、扬声器的位置和音调将根据与听者的关系自动受到影响。

听众是玩家，或者是游戏摄像机。它有一个像声源一样的位置和速度，但它也有一个方向。