Dynamic\_Runtime目录下的lib库文件是采用“动态链接VC运行库”编译而成。

Static\_Runtime目录下的lib库文件是采用“静态链接VC运行库”编译而成。

以\_Release结尾的库，链接Release的运行库。

以\_Debug结尾的库，链接Debug的运行库。

详细接口说明请阅读Interface.h

典型使用方法(Surround)：

ISurround\* pSurround = CreateSurroundInstance(); // 创建实例

pSurround-> SetSamplingRate(44100);

pSurround-> SetChannels(2);

pSurround-> SetSurround(1.0f); // 设置环绕强度

while (!feof(InputFile))

{

short saPCMBuffer[4096];

int nSampleCount = GetPCMData(saPCMBuffer);

nSampleCount = pSurround-> Process(saPCMBuffer, nSampleCount); // 处理声音

AudioOutWrite(saPCMBuffer, nSampleCount); // 写入声卡

If (bSeek) pSurround->Reset(); // 如果歌曲发生了Seek或切换，则Reset

}

ReleaseSurroundInstance(pSurround); // 释放实例

典型使用方法(DynamicVolume)：

IDynamicVolume\* pVolume = CreateDynamicVolumeInstance (); // 创建实例

pVolume-> SetSamplingRate(44100);

pVolume-> SetChannels(2);

pVolume-> SetRatio(2.5f); // 设置音量控制强度

pVolume-> SetVolume(1.0f); // 设置期望音量(1.0表示最大音量)

pVolume-> SetMaxGainFactor(8.0f); // 设置最大放大倍数

while (!feof(InputFile))

{

short saPCMBuffer[4096];

int nSampleCount = GetPCMData(saPCMBuffer);

nSampleCount = pVolume -> Process(saPCMBuffer, nSampleCount); // 处理声音

AudioOutWrite(saPCMBuffer, nSampleCount); // 写入声卡

If (bSeek) pSurround->Reset(); // 如果歌曲发生了Seek或切换，则Reset

}

ReleaseDynamicVolumeInstance (pVolume); // 释放实例

典型使用方法(Limiter)：

ILimiter\* pLimiter\_LeftChannel = CreateLimiterInstance(); // 创建实例(左声道限幅器)

ILimiter\* pLimiter\_RightChannel = CreateLimiterInstance(); // 创建实例(右声道限幅器)

float faPCMBuffer[N \* 2]; // 32位浮点PCM数据(立体声)

for (int i = 0; i < N \* 2; i += 2)

{

faPCMBuffer[i + 0] = pLimiter\_LeftChannel->Process(faPCMBuffer[i + 0]);

faPCMBuffer[i + 1] = pLimiter\_RightChannel->Process(faPCMBuffer[i + 1]);

}

ReleaseLimiterInstance(pLimiter\_LeftChannel);

ReleaseLimiterInstance(pLimiter\_RightChannel);