实现思路：

**基于webrtc branch57**

最后一个提交记录为：

commit 52b6562a10b495cf771d8388ee51990d56074059

Author: Magnus Jedvert <magjed@webrtc.org>

Date: Sat Feb 25 20:12:18 2017 +0100

Merge to 57: Clear out cached codecs when calculating new codec lists.

Without this, every time WebRtcVideoEngine2 calls supported\_codecs(),

the codec list grows.

BUG=webrtc:7020

Original Review-Url: https://codereview.webrtc.org/2639423006

Original Cr-Commit-Position: refs/heads/master@{#16178}

(cherry picked from commit be850e1b1de5bf046080ac1df7de67a3d6d12d1c)

Review-Url: https://codereview.webrtc.org/2717053002 .

Cr-Commit-Position: refs/branch-heads/57@{#7}

Cr-Branched-From: e5cbc2019003dbb40e03811d7607feb95757a4ec-refs/heads/master@{#16123}

**首先需要修改一个webrtc文件：**webrtc/audio/audio\_state.cc

diff --git a/webrtc/audio/audio\_state.cc b/webrtc/audio/audio\_state.cc

index c15ddd7..79a316f 100644

--- a/webrtc/audio/audio\_state.cc

+++ b/webrtc/audio/audio\_state.cc

@@ -35,8 +35,8 @@ AudioState::AudioState(const AudioState::Config& config)

RTC\_DCHECK(device);

// This is needed for the Chrome implementation of RegisterAudioCallback.

- device->RegisterAudioCallback(nullptr);

- device->RegisterAudioCallback(&audio\_transport\_proxy\_);

+// device->RegisterAudioCallback(nullptr);

+// device->RegisterAudioCallback(&audio\_transport\_proxy\_);

}

也就是注释最后两行内容：

device->RegisterAudioCallback(nullptr);

device->RegisterAudioCallback(&audio\_transport\_proxy\_);

对webrtc的修改只有这两行。

拿数据的思路：

使用externalMedia接口，注册回调函数，voiceEngine会在数据投递给喇叭之前回调原始数据上来。

int StartGetMixData()

{

AudioMixDataCallBack\* p = new AudioMixDataCallBack();

webrtc::VoEExternalMedia\* externalMedia = webrtc::VoEExternalMedia::GetInterface(g\_voe->engine());

//回调本地录制数据

externalMedia->RegisterExternalMediaProcessing(-1, webrtc::kRecordingAllChannelsMixed, \*p);

//回调所有远端的合成数据

externalMedia->RegisterExternalMediaProcessing(-1, webrtc::kPlaybackAllChannelsMixed, \*p);

return 0;

}

数据会回调到：

class AudioMixDataCallBack :public webrtc::VoEMediaProcess

{

public:

virtual void Process(int channel,

webrtc::ProcessingTypes type,

int16\_t audio10ms[],

size\_t length,

int samplingFreq,

bool isStereo)

{

if (type == webrtc::kPlaybackAllChannelsMixed)

{

printf("get remote mix pcm data\n");

}

if (type == webrtc::kRecordingAllChannelsMixed)

{

printf("get local record pcm data\n");

//本地声音是连续的，如果做录音mp3应该以本地回调为时间参考，在本地回调时录音

}

}

};

在这里可以将本地数据和远端数据再次进行合并，并且使用ffmpeg录音为mp3文件。

我这里只打印了一下日志。

Demo里的静态库是我修改好webrtc编译后，考过来的。