播放流程：

//只有在语音通道收发线程中UdpEndPoint::recvToIOQ对incoming赋值 -1才会在音频接收线程中进入收听

1.播放队列的建立：

ListenThread（音频收听线程） startListen(这个函数没作用)

startListen66 createListen Session::createIncomingTransaction

Initial函数中创建接收 io队列接收数据（并创建文件）

StreamPlayScheduler::delegate(StreamPlaySource \*source, vl\_bool preemption)（在创建了io队列后要对io队列登记）

source->spsStandby()

EndPointManager::getInstance()->registerRecvQueue(登记io函数)

IncomingTransaction类中的 RTPRecvIOQueue \*ioq即为音频收听线程中的io队列

this->pcmFeeder = new TransactionPCMFeeder(this->decoder, this->ioq, pktMemPool,&(this->context), this);通过这段代码把ioq地址赋值给TransactionPCMFeeder::feedPcm函数中的ioq

2.播放器的初始化：

播放部分的解码器初始化是在播放录音线程streamplay\_monitor\_looper流程中初始化并创建的：

播放录音线程中的monitor->schedualPlay();

IncomingTransaction::spsStartTrack

IncomingTransaction::startTrack CodecManager::createAudioDecoder（创建并初始化解码器）

3.结束播放：

在退出群组等操作时会调用 session\_handle为空

CTalkAppDlg::endSession stopListen（和createListen，结束收听）

stopTalking

4.创建多媒体数据收发线程：

单呼，群呼等会调用

CTalkAppDlg::startMeidaSession startHeartBeat(多媒体语音心跳包)

raiseSession1（多媒体数据收发线程）

udp\_ept\_recv\_looper (接收线程) udp\_ept\_send\_looper（发送线程）

在UdpEndPoint析构函数中取消线程 在UdpEndPoint析构函数中取消线程

录音流程：

1.录音器的初始化及队列的创建登记：

录音部分的编码器是在创建发送队列的时候初始化的，和播放部分解码器的初始化不同

CTalkAppDlg::m\_record\_insome（录音接口）

CTalkAppDlg::startTalking

createSpeakNetParam

createOutgoingTransaction

OutgoingTransaction::initialOutgoing 创建发送 io队列(并创建文件)

CodecManager::createAudioEncoder

SilkEncoder::SilkEncoder(初始化编码器)

StreamPlayScheduler::delegate(StreamPlaySource \*source, vl\_bool preemption)（创建io队列后要对io队列登记）

source->srsStandby

EndPointManager::getInstance()->registerSendQueue(登记函数)

语音通话的结束处理：

是在播放录音线程streamplay\_monitor\_looper

StreamPlayScheduler::schedualPlay

StreamPlayScheduler::processSourceEnd（在此函数中结束上一个通话）