

Mixed Stream Recording

1 Features

The On-Premises-Recording SDK supports mixing the audio and video streams of all users in the room into a single screen and then start recording. After integrating the SDK, you can start to use the Mixed-Stream-Recording function.

2 Detailed Operation

2.1 Set the Listener of Callback

Before initialization, you need to inherit the interface named 'icallback' and set the listener of the callback.

- Interface Prototype:

```
ZEGO_API bool SetCallback(ICallback* pCB)
```

- Parameter:

`pCB`: Pointer to the object of callback.

- Notes:

Before initializing the SDK, you can call the following interfaces as needed:

`SetLogDirAndSize` Set the path and size of the log file

`SetUseTestEnv` Decide whether to use a test environment.

2.2 Initialize SDK

Use the obtained AppID and AppSign of ZEGO to initialize the SDK. If the SDK is not initialized, the related interface will be abnormal. If you have no AppID and AppSign, please apply in [ZEGO Admin Console](#); You can refer to [User Manual](#) for help.

- Interface Prototype:

```
ZEGO_API bool InitSDK(unsigned int uiAppID, unsigned char* pBufAppSignature, int nSignatureSize)
```

- Parameter:

uiAppID: The digital ID obtained from ZEGO, it is unique for each developer.

pBufAppSignature: The Signature obtained from ZEGO, it is used to verify the validity of the your appid.

nSignatureSize: Size of the Signature (byte).

- Asynchronous Callback Interface:

```
class ICallback
{
public:
    virtual void OnInitSDK(int errorCode) {}
}
```

errorCode == 0 # Initialization succeeded.

2.3 Login Room

When initialize successfully, you also need to login the room before recording.

- Interface Prototype:

```
ZEGO_API bool LoginRoom(const char* pszRoomID)
```

- Parameters:

`pszRoomID`: Room ID, for the Room with a pushing-stream.

- Asynchronous Callback Interface:

```
class ICallback
{
public:
    virtual void OnLoginRoom(int errorCode, const char *pszRoomID) = 0;
}
```

2.4 Start Recording

1. Before you call the interface, the SDK must be initialized successfully.
2. It is recommended to start recording according to the stream-add information, which you can get from the stream information update callback `OnStreamUpdate` After login the room.
3. You can get the maximum number of simultaneous recording stream that the SDK allows, by calling the interface `GetMaxRecordCount`.

- Interface Prototype:

```
ZEGO_API bool StartRecordMixStream(ZegoMixStreamRecordConfig* pConfig)
```

- Parameters:

`pConfig`: Configuration of mixed-stream-recording, please refer to: `ZegoMixStreamRecordConfig`.

- Asynchronous Callback Interface:

```
class ICallback
{
public:
    virtual void OnStreamRecordBegin(const char* pszStreamID, const char *pszPathAndName) = 0;
    virtual void OnMixStreamRecordUpdate(const char** ppszStreamID, int nStreamCount) = 0;
}
```

When recording start successfully, you will receive the callback named `OnStreamRecordBegin`, When successfully pull a stream with audio and video data, you will receive the callback named `OnMixStreamRecordUpdate`.

- Notes:

If you need to record single and mixed streams simultaneously, before recording you need to call the interface: `EnableSingleAndMixRecordMode`; This mode consumes more system resources, and can only start mixed-stream recording first, and then start individual-stream recording; when you end mixed-stream recording, all individual-stream recording will also end. In this mode, the SDK has a limitation. Mixed stream recording will mix all the streams being recorded by individual-stream recording. Even if a single stream in individual-stream recording is not used as a input stream of mixed-stream recording, if this single stream has audio data, the audio data will also be recorded by mixed-stream recording.

2.5 Update Configuration of the Mixed Stream

When it is necessary to increase the input stream, change the original input stream, or change the mixed-stream output configuration, you need to reconstruct the mixed-stream configuration and call the following interface to update mixed-stream configuration .

- Interface Prototype:

```
ZEGO_API bool UpdateInputStreamConfig(ZegoStreamConfig* pInputStream, int nInputStreamCount)
```

- Asynchronous Callback Interface:

```
class ICallback
{
public:
    virtual void OnMixStreamRecordUpdate(const char** ppszStreamID, int nStreamCount) = 0;
}
```

when a mixed-stream recording starts successfully, a callback is returned, which contains a list of ID of all input single streams of the mixed-stream, only stream which is pulled successfully will be recorded. **Note: You will receive this callback when any single stream of mixed-stream recording changes, such as starting recording, abnormal recording, etc.**

2.6 Stop Recording

- Interface Prototype:

```
ZEGO_API bool StopRecordMixStream()
```

- Asynchronous Callback Interface:

```
class ICallback
{
public:
    virtual void OnStreamRecordEnd(const char* pszStreamID, const char *pszPathAndName, RecordEndReason reason) = 0;
}
```

- Notes:

When recording is no longer needed, it is recommended to do the following:

1. Call the interface `LogoutRoom` to logout the room.
2. Call the interface `LIVEROOM::SetCallback(nullptr);` to Remove the callback listener.
3. Call the interface `UnInitSDK` to deinitialization the SDK.

3 API Reference

Interface	Description
SetUseTestEnv	Set whether to use the test environment or

	not, you can use the test environment for testing during the development.
SetCallback	Set the callback listener.
InitSDK	Initialize SDK
LoginRoom	Login room
GetMaxRecordCount	Get the maximum number of simultaneous recording stream that the SDK allows
EnableSingleAndMixRecordMode	Set whether to enable individual-stream and mixed-stream simultaneous recording mode, this mode consumes more system resources.
StartRecordMixStream	Start mixed-stream recording.
UpdateInputStreamConfig	Update the input stream configuration of mixed-stream recording.
StopRecordMixStream	Stop mixed-stream recording.
SetMuxerOutType	Set recording data output type.
SetLogDirAndSize	Set the path and size of the log file, size for a single log.
SetConfig	Set configuration
SetMuxerCacheSize	Set the cache size before the file is created. (64KB~1MB)
GetSDKVersion	get the SDK version.
UploadLog	Upload log file to ZEGO Server
UnInitSDK	Deinitialization the SDK
LogoutRoom	Logout the room