# ffmpeg SDK□□□

## □□□□□release□□

https://github.com/GyanD/codexffmpeg/releases

## □□□□□dll□□□

https://github.com/donglietao/ffmpeg-dll

```
def ffmpeg():
    dll = ctypes.windll.LoadLibrary(
        "D:/cpp-sdk/FFmpeg-n7.0/ffmpeg-exe-dll/Release/release/ffmpeg-dll.dll"
    )

    ffmpeg_main = dll.ffmain

    ffmpeg_main.argtypes = [ctypes.c_int, ctypes.POINTER(ctypes.c_char_p)]
    ffmpeg_main.restype = ctypes.c_int
    argv = (ctypes.c_char_p * 10)()
    argv[0] = bytes("ffmpeg", "utf-8")
    argv[1] = bytes("-f", "utf-8")
    argv[2] = bytes("gdigrab", "utf-8")
    argv[3] = bytes("-i", "utf-8")
    argv[4] = bytes("desktop", "utf-8")
    argv[5] = bytes("-vcodec", "utf-8")
    argv[6] = bytes("libx264", "utf-8")
    argv[7] = bytes("-f", "utf-8")
    argv[8] = bytes("flv", "utf-8")
    argv[9] = bytes("rtmp://srs-push.qq829.cn:31935/live/557254322", "utf-8")
    ret = ffmpeg_main(10, argv)
    ctypes.windll.kernel32.FreeLibrary.argtypes = [ctypes.c_void_p]
    ctypes.windll.kernel32.FreeLibrary.restype = ctypes.c_int
    ctypes.windll.kernel32.FreeLibrary(dll._handle)
    return ret
```

```
ffmpeg()
```

# ffmpeg□□□c++□□□□

```
// □□□□□□□□□□
avdevice_register_all();
// □□□□□□□
avformat_network_init();
//----------------□□□□□□□----------------------//
// □□□□□□□
av_find_input_format("gdigrab");
// □□□□□□□
avformat_open_input();
// □□□□□□
avformat_find_stream_info();
// □□□
in_stream = input_fmt_ctx->streams[i];
// □□□□□□□□□
AVCodec *in_codec = avcodec_find_decoder(in_stream->codecpar->codec_id);
// □□□□□□
in_codec_ctx = avcodec_alloc_context3(in_codec);
// □□□□□□□
avcodec_parameters_to_context();
// □□□□□□
avcodec_open2();
//----------------□□□□□□□----------------------//
// □□□□□□□
av_guess_format()
// □□□□□□□□
avformat_alloc_output_context2();
// □□□□□□
AVStream *out_stream = avformat_new_stream(output_fmt_ctx, NULL);
// □□□□□□□
out_stream->codecpar->codec_id = AV_CODEC_ID_H264;
// □□□□□□□
AVCodec *out_codec = avcodec_find_encoder(out_stream->codecpar->codec_id);
// □□□□□□□□□
```

```
AVCodecContext *out_codec_ctx = avcodec_alloc_context3(out_codec);
// □□□□□□□□
avcodec_parameters_to_context(out_codec_ctx, out_stream->codecpar);
// □□□□□
avcodec_open2();
// □□SPS□PPS□□
out_stream->codecpar->extradata = out_codec_ctx->extradata;
out_stream->codecpar->extradata_size = out_codec_ctx->extradata_size;
// □□□□IO
avio_open();
// □□□□□□□□
avformat_write_header();
//----------------□□□□□----------------------//
while (1){
        // □□□□□□□□□□□
        ret = av_read_frame(input_fmt_ctx, &packet);
        // □□□□□□□□
        ret = avcodec_send_packet(in_codec_ctx, &packet);
        // □□□□□□□□□
        ret = avcodec_receive_frame(in_codec_ctx, frame);
        // □□□□
        ret = sws_scale(sws_ctx, (const uint8_t *const *)frame->data, frame->linesize,
                        0, frame->height,
                        enc_frame->data, enc_frame->linesize);
        // □□□□□□□□
        enc_frame->pts = av_rescale_q_rnd((frame->pts - start_pts), in_stream->time_base,
out_stream->time_base, AV_ROUND_NEAR_INF | AV_ROUND_PASS_MINMAX);
        enc_frame->pkt_dts = enc_frame->pts;
        //  □□□□□□□□□
        ret = avcodec_send_frame(out_codec_ctx, enc_frame);
        // □□□□□□□□□□□
        while (avcodec_receive_packet(out_codec_ctx, &enc_packet) == 0)
        {
            // □□□□□
            ret = av_interleaved_write_frame(output_fmt_ctx, &enc_packet);
            if (ret < 0)
            {
                printf("7:Error while writing output packet:%s\n", av_err2str(ret));
                break;
            }
        }
```

```
        // □□□□□□□□
        av_packet_unref(&enc_packet);
        // □□□□□□□□
        av_packet_unref(&packet);
}
//----------------□□□□-----------------------//
    // □□□□□□
    av_write_trailer(output_fmt_ctx);
    // □□□□□□□
    avio_close(output_fmt_ctx->pb);
    // □□□□□□
    av_packet_free(&packet);
    av_packet_free(&enc_packet);
    // □□□
    av_frame_free(&frame);
    av_frame_free(&enc_frame);
    // □□□□□□□□
    sws_freeContext(sws_ctx);
    // □□□□□□□□□
    avcodec_free_context(&in_codec_ctx);
    avcodec_free_context(&out_codec_ctx);
    // □□□□□□□□□□
    avformat_close_input(&input_fmt_ctx);
    avformat_free_context(input_fmt_ctx);
    // □□□□□□□
    avformat_close_input(&output_fmt_ctx);
    avformat_free_context(output_fmt_ctx);
    // □□□□□□
    avformat_network_deinit();
```

---

□□□ #4
□ □□□ □□□ 1 □□ 2024 17:42:46
□ □□□ □□□ 18 □□ 2025 03:45:38