

## MSDK Def

### 1. Error code

typedef enum \_eISIL\_ERR\_CODE

```
{  
    ISIL_ERR_SUCCESS                = 0,  
    ISIL_ERR_FAILURE                 = -1,  
    ISIL_ERR_PASSWORD_ERROR          = -1000,  
    ISIL_ERR_NOENOUGHPRI,  
    ISIL_ERR_NOINIT,  
    ISIL_ERR_INITERROR,  
    ISIL_ERR_NOLOGIN,  
    ISIL_ERR_CHANNEL_ERROR,  
    ISIL_ERR_OVER_MAXLINK,  
    ISIL_ERR_VERSIONNOMATCH,  
    ISIL_ERR_NETWORK_FAIL_CREATE_SOCKET,  
    ISIL_ERR_NETWORK_FAIL_CONNECT,  
    ISIL_ERR_NETWORK_SEND_ERROR,  
    ISIL_ERR_NETWORK_RECV_ERROR,  
    ISIL_ERR_NETWORK_RECV_TIMEOUT,  
    ISIL_ERR_NETWORK_ERROR_DATA,  
    ISIL_ERR_CREATE_BUFFER_ERROR,  
    ISIL_ERR_ORDER_ERROR,  
    ISIL_ERR_OPERNOPERMIT,  
    ISIL_ERR_COMMAND_TIMEOUT,  
    ISIL_ERR_ERROR_SERIALPORT,  
    ISIL_ERR_ERROR_ALARMPORT,  
    ISIL_ERR_PARAMETER_ERROR,  
    ISIL_ERR_PARAMETER_BUFFER_TOO_SMALL,  
    ISIL_ERR_CHAN_EXCEPTION,  
    ISIL_ERR_NODISK,  
    ISIL_ERR_ERROR_DISK_NUM,  
    ISIL_ERR_DISK_FULL,  
    ISIL_ERR_DISK_ERROR,  
    ISIL_ERR_NOT_SUPPORT,  
    ISIL_ERR_BUSY,  
    ISIL_ERR_MODIFY_FAIL,  
}
```

ISIL\_ERR\_PASSWORD\_FORMAT\_ERROR,  
ISIL\_ERR\_DISK\_FORMATING,  
ISIL\_ERR\_DVRNORESOURCE,  
ISIL\_ERR\_DVROPRATEFAILED,  
ISIL\_ERR\_OPENHOSTSOUND\_FAIL,  
ISIL\_ERR\_DVRVOICEOPENED,  
ISIL\_ERR\_TIMEINPUTERROR,  
ISIL\_ERR\_NOSPECFILE,  
ISIL\_ERR\_DIR\_ERROR,  
ISIL\_ERR\_ALLOC\_RESOUCE\_ERROR,  
ISIL\_ERR\_AUDIO\_MODE\_ERROR,  
ISIL\_ERR\_NOENOUGH\_BUF,  
ISIL\_ERR\_CREATESOCKET\_ERROR,  
ISIL\_ERR\_SETSOCKET\_ERROR,  
ISIL\_ERR\_MAX\_NUM,  
ISIL\_ERR\_USERNOTEXIST,  
ISIL\_ERR\_WRITEFLASHERROR,  
ISIL\_ERR\_UPGRADEFAIL,  
ISIL\_ERR\_CARDHAVEINIT,  
ISIL\_ERR\_PLAYERFAILED,  
ISIL\_ERR\_MAX\_USERNUM,  
ISIL\_ERR\_GETLOCALIPANDMACFAIL,  
ISIL\_ERR\_NOENCODEING,  
ISIL\_ERR\_IPMISMATCH,  
ISIL\_ERR\_MACMISMATCH,  
ISIL\_ERR\_UPGRADELANGMISMATCH,  
ISIL\_ERR\_TALKING,

ISIL\_ERR\_FILE\_CREATE = -1200,  
ISIL\_ERR\_FILE\_WRITE,  
ISIL\_ERR\_FILE\_OPEN,  
ISIL\_ERR\_FILE\_READ,  
ISIL\_ERR\_FILE\_CREATEINDEX,  
ISIL\_ERR\_FILE\_OPERATE,  
ISIL\_ERR\_FILE\_GETPLAYTIME,  
ISIL\_ERR\_FILE\_FORMAT,

```

    ISIL_ERR_FILE_PLAY,
    ISIL_ERR_FILE_READDATA_OVERSIZE,
    ISIL_ERR_FILE_BEGIN,
    ISIL_ERR_FILE_END,
    ISIL_ERR_FILE_REMAIN_FRAME,
    ISIL_ERR_FILE_RULE_END,
    ISIL_ERR_FILE_NOT_SETRULE,

    // Record
    ISIL_ERR_FrameParam_ID    = -1300,
    ISIL_ERR_CreateConnection,
    ISIL_ERR_CreateRecordset,

    ISIL_ERR_DEV_FAILURE      = -2000,

}ISILERR_CODE;

```

## 2. ISIL\_VIDEO\_STANDARD

```

typedef enum _eISIL_VIDEO_STANDARD
{
    ISILMSDK_VIDEO_STANDARD_PAL,
    ISILMSDK_VIDEO_STANDARD_NTSC,
    ISILMSDK_VIDEO_STANDARD_USER_DEFINE,

}ISIL_VIDEO_STANDARD;

```

## 3. ISIL\_AV\_PACKET

```

typedef enum ISIL_H264_NALU_TYPE_E{

    H264_NALU_PSLICE = 1,

    H264_NALU_ISLICE = 5,

    H264_NALU_IDRSLICE = 6,

```

```

    H264_NALU_SPS    = 7,

    H264_NALU_PPS    = 8,

    H264_NALU_BUTT

}H264_NALU_TYPE_E;

typedef enum ISIL_AUDIO_TYPE_E{

    ISIL_AUDIO_PCM_E  ,

    ISIL_AUDIO_ALAW_E ,

    ISIL_AUDIO_ULAW_E ,

    ISIL_AUDIO_ADPCM_E ,

    AUDIO_TYPE_BUTT

}AUDIO_TYPE_E;

typedef enum ISIL_STREAM_TYPE_E{

    ISIL_MAIN_STREAM_E,

    ISIL_SUB_STREAM_E

}STREAM_TYPE_E;

typedef struct {

    unsigned short x_width;

    unsigned short y_height;

}ISIL_IMAGE_SIZE;


typedef struct ISIL_H264_INF_T{

    H264_NALU_TYPE_E nalu_type;

    STREAM_TYPE_E stream_type;

    ISIL_IMAGE_SIZE image_size;

```

```
}H264_INF_T;
```

```
typedef struct ISIL_AUDIO_INF_T{
```

```
    AUDIO_TYPE_E audio_type_e;
```

```
    int sample;
```

```
    int bit_rate;
```

```
}AUDIO_INF_T;
```

```
typedef enum ISIL_DATA_TYPE_E{
```

```
    ISIL_H264_DATA,
```

```
    ISIL_AUDIO_DATA,
```

```
    ISIL_MJPEG_DATA,
```

```
    DATA_TYPE_BUTT
```

```
}DATA_TYPE_E;
```

```
typedef union ISIL_DATE_TYPE_U{
```

```
    H264_INF_T h264_inf;
```

```
    AUDIO_INF_T audio_inf;
```

```
}DATE_TYPE_U;
```

```
typedef struct _ISIL_AV_PACKET{
```

```
    unsigned int pts;
```

```
    unsigned short chip_id;
```

```
    unsigned short chan_id;
```

```
    unsigned int frm_seq; //frame number
```

```
    int buff_size;        //buff size
```

```

int date_len;          //valid date len
unsigned char *buff;
DATA_TYPE_E frm_type;
DATE_TYPE_U date_type_u;
volatile int ref;
void *priv;           //inner frame ptr, use for frame release.

```

```

}ISIL_AV_PACKET;

```

#### 4. ISIL\_VIDEO\_SIZE\_TYPE

```

typedef enum _eISIL_VIDEO_SIZE_TYPE

```

```

{
    ISIL_VIDEO_SIZE_QCIF = 0,

    ISIL_VIDEO_SIZE_QHALF_D1,

    ISIL_VIDEO_SIZE_CIF,

    ISIL_VIDEO_SIZE_HALF_D1,

    ISIL_VIDEO_SIZE_D1,

    ISIL_VIDEO_SIZE_HCIF,

    ISIL_VIDEO_SIZE_2CIF,

    ISIL_VIDEO_SIZE_4CIF,

    ISIL_VIDEO_SIZE_VGA,

    ISIL_VIDEO_SIZE_SVGA,

    ISIL_VIDEO_SIZE_XGA,

    ISIL_VIDEO_SIZE_720P,

    ISIL_VIDEO_SIZE_1080P,

    ISIL_VIDEO_SIZE_USER,

```

```

}ISIL_VIDEO_SIZE_TYPE;

```

#### 5. VENC\_MJPEG\_CFG

```

typedef struct ISIL_mjpeg_encode_param{

```

```

    unsigned int change_mask_flag;
    unsigned int e_image_level;
    unsigned int i_image_width_mb_size;
    unsigned int i_image_height_mb_size;
    unsigned int i_capture_frame_number;
    unsigned int i_capture_frame_stride;
    unsigned int i_capture_type;
} VENC_MJPEG_CFG;

```

## 6. ISIL\_AUDIO\_TYPE

```

typedef enum _eISIL_AUDIO_TYPE
{
    ISIL_CODEC_AUDIO_ALAW = 10, //G.711
    ISIL_CODEC_AUDIO_ULAW, //G.711
    ISIL_CODEC_AUDIO_ABADPCM, //G.722
    ISIL_CODEC_AUDIO_MPMLQ, //G.723.1
    ISIL_CODEC_AUDIO_ADPCM, //G.726
    ISIL_CODEC_AUDIO_EMBED_ADPCM, //G.727 embedded adpcm
    ISIL_CODEC_AUDIO_LDCELP, //G.728
    ISIL_CODEC_AUDIO_CSACELP, //G.729
} ISIL_MEDIASDK_AUDIO_TYPE;

```

## 7. ISIL\_STREAM\_BITRATE\_TYPE

```

typedef enum _eISIL_STREAM_BITRATE_TYPE
{
    ISIL_STREAM_BITRATE_H264_NO_RC,
    ISIL_STREAM_BITRATE_H264_CBR,
    ISIL_STREAM_BITRATE_H264_VBR,
} ISIL_STREAM_BITRATE_TYPE;

```

## 8. ISIL\_BITRATE\_VALUE

```

typedef struct _tagISIL_BITRATE_VALUE
{
    union
    {
        unsigned long lImageQuality; //when VBR , from bad to good[0,5]
        struct __tagQP           //when CQP , good to bad[0,51]
    }
}

```

```

    {
        unsigned char cIQP;    //I frame QP[0,51]
        unsigned char cPQP;    //P frame QP[0,51]
        unsigned char cBQP;    //B frame QP[0,51]
        unsigned char cReserve;

    }tQP;
};
}ISIL_BITRATE_VALUE;

```

#### 9. ISIL\_STREAM\_BITRATE\_Mode

```

typedef enum _eISIL_STREAM_BITRATE_MODE
{
    ISIL_STREAM_BITRATE_RC_IMAGE_QUALITY_FIRST = 0,
    ISIL_STREAM_BITRATE_RC_IMAGE_SMOOTH_FIRST,
}ISIL_STREAM_BITRATE_MODE;

```

#### 10. DEC\_AV\_SYNC

```

enum sync_type {
    SYNC_METHOD_NORMAL = 0,
    SYNC_METHOD_AV,
    SYNC_METHOD_MULTI,
    SYNC_METHOD_USER_SPECIFY,
    SYNC_METHOD_INVALID = 0x10, //max 16 different sync method
};

struct syn_item {
    unsigned short    type;    //video: 0; audio: 1
    unsigned short    chan_id; //channel id (logic)
};

```

```

typedef struct syn_arg {
    enum sync_type type; //sync type

    unsigned short nr; //how many syn_items (each item represent one channel) user want to sync

    unsigned short idx_ref; //which entry(represent one channel) is the reference base

    struct syn_item items[0]; //syn_items starts from here

```



```
}DEC_AV_SYNC;
```

## 11. BIND\_DEC2VO

```
typedef struct ISIL_chip_h264d_bind_info{  
    unsigned int i_h264d_logic_id;  
    unsigned int i_display_phy_id;  
}BIND_DEC2VO;
```

## 12. VDEC\_CH\_MODE

```
enum    x_mode{  
  
    ISIL_CODEC_DEC_NON_RT_FPS = 0,           // non-real-time  
  
    ISIL_CODEC_DEC_SPEED_NORMAL,             //speed x-1  
  
    ISIL_CODEC_DEC_SPEED_FAST,                //speed larger than x-1  
  
    ISIL_CODEC_DEC_SPEED_SLOW,                //speed smaller than x-1  
  
};
```

```
struct x_speed  
  
{  
  
    Enum x_mode    mode;                      //the mode specified  
  
    Unsigned long   x_speed_value;            //the real speed value  
  
};
```

```
typedef struct decoder_mode  
{  
    unsigned char loop;           //0: not loop, 1: loop  
    unsigned char direction;      //0: backward 1: forward  
    unsigned char key_frame;      //0: frame one by one 1:only for key frames  
    unsigned char is_continue;    //0: stop after display one frame  
                                   //1: continue after after display one frame  
    Struct x_speed speed;         // -1 means nonsense  
}VDEC_CH_MODE;
```

### 13. CODEC\_CHIP\_MSG

typedef struct

```
{  
    unsigned int u32ChipID;//chip handle  
    unsigned int u32Cmd;//control cmd  
    unsigned int u32Blk;//block or nonblock operator  
    unsigned int u32MsgLen;//msg length  
    char      pMsg[0];//msg content  
}CODEC_CHIP_MSG;
```

### 14. VD\_CONFIG\_REALTIME

```
#define ISIL_PHY_VD_CHAN_NUMBER (16)
```

```
enum ISIL_FRAME_MODE{  
    ISIL_FRAME_MODE_INTERLACE = 0,  
    ISIL_FRAME_MODE_PROGRESSIVE,  
    ISIL_FRAME_MODE_RESERVED,  
};
```

```
enum ISIL_VIDEO_SIZE{  
    ISIL_VIDEO_SIZE_QCIF = 0,  
    ISIL_VIDEO_SIZE_QHALF_D1,  
    ISIL_VIDEO_SIZE_CIF,  
    ISIL_VIDEO_SIZE_HALF_D1,  
    ISIL_VIDEO_SIZE_D1,  
    ISIL_VIDEO_SIZE_HCIF,  
    ISIL_VIDEO_SIZE_2CIF,  
    ISIL_VIDEO_SIZE_4CIF,  
    ISIL_VIDEO_SIZE_VGA,  
    ISIL_VIDEO_SIZE_SVGA,  
    ISIL_VIDEO_SIZE_XGA,  
    ISIL_VIDEO_SIZE_H960,  
    ISIL_VIDEO_SIZE_720P,  
    ISIL_VIDEO_SIZE_1080P,  
    ISIL_VIDEO_SIZE_USER,  
};
```

typedef struct{

```

enum ISIL_VIDEO_STANDARD video_std;
int drop_frame;//0:not drop frame,1:drop
int chan[ISIL_PHY_VD_CHAN_NUMBER];
enum ISIL_VIDEO_SIZE video_size[ISIL_PHY_VD_CHAN_NUMBER];
int frame_rate[ISIL_PHY_VD_CHAN_NUMBER];//PAL [0, 25], NTSC [0, 30];
enum ISIL_FRAME_MODE interleave[ISIL_PHY_VD_CHAN_NUMBER];
}VD_CONFIG_REALTIME;

```

#### 15. BIND\_VI2VO

```

typedef struct ISIL_chip_vi2vo_bind_info{
    unsigned int i_vi_phy_id;
    unsigned int i_display_phy_id;
}BIND_VI2VO;

```

#### 16. ISIL\_MEDIASDK\_ENCODE\_PARAM

```

typedef struct {
    unsigned char    cChannelPri;
    unsigned char    cStreamType;
    unsigned char    cAudioType;
    unsigned char    cAudioBitAlign;
    unsigned char    cAudioSampleRate;
    unsigned char    cImageSize;
    unsigned char    cRCType;
    unsigned char    cBitMode;
    unsigned char    cImageQuality;
    unsigned char    cIQP;
    unsigned char    cPQP;
    unsigned char    cBQP;
    unsigned char    cFPS;
    unsigned char    cHorizontalFlip;
    unsigned char    cVerticalFlip;
    unsigned char    cReserve[5];    //
    unsigned short   nBitRate;
    unsigned short   nGOPIntervals;
    unsigned short   nIFrameIntervals;
    unsigned short   nBPFrameInterval;

```

```

unsigned short  nWidthMBSize;
unsigned short  nHeightMBSize;
} ISIL_MEDIASDK_ENCODE_PARAM;

```

#### 17. VIDEO\_LOSS\_ALARM

参考 ISIL\_VI\_ALARM\_TYPE

```

typedef struct isil_video_lost_detection_msg{

unsigned int  msg_type;    //the value comes from
ISIL_MSG_TYPE_E(egs:ISIL_VIDEO_LOST_DETECTION_MSG)

unsigned int  msg_len;    //the length of isil_video_lost_detection_msg

unsigned short  chan_id;    //gen video_lost_msg's chan in chip

unsigned char  video_lost_valid;    //video signal lost

unsigned char  video_connect_valid; //after video lost, the video signal have been connected

}STRUCT_PACKET_ALIGN(1) isil_video_lost_detection_msg_t;

```

#### 18. BLIND\_ALARM

参考 ISIL\_VI\_ALARM\_TYPE

#### 19. NIGHT\_ALARM

参考 ISIL\_VI\_ALARM\_TYPE

#### 20. PNSWITCH\_ALARM

```

typedef enum{

    VI_ALARM_VIDEO_LOST = 0x01, //video lost

    VI_ALARM_VIDEO_CONNECT= 0x02, //video connect

    VI_ALARM_VIDEO_NIGHT_DAYTONIGHT= 0x04, //video night day to night

    VI_ALARM_VIDEO_NIGHT_NIGHTTODAY= 0x08, //video night night to day

    VI_ALARM_VIDEO_BLIND_ADD= 0x10, //video blind add

    VI_ALARM_VIDEO_BLIND_REMOVE= 0x20, //video blind remove

```

```
VI_ALARM_VIDEO_STD_CHANGE = 0x40, //video standard change

}ISIL_VI_ALARM_TYPE;
```

## 21. VENC\_OSD\_CFG

```
#define NAME_LEN (32)

#define SUB_LEN (44)

#define OSD_ATTR_DISPLAY_OFF (0)

#define OSD_ATTR_DISPLAY_ON (0x1)

#define OSD_FONT_12 (0x2)

#define OSD_FONT_24 (0x8)

#define OSD_FONT_MASK (0xe)

#define CHAR_RECTANGLE_X_LEFT (8)

#define CHAR_RECTANGLE_Y_LEFT (2)
```

```
typedef struct _ISIL_OSD_PARAM {

    unsigned int channel; //chan_id, this domain no use

    char name[NAME_LEN]; //chan's name, the maxim number is 16

    unsigned int name_attr; //name's attribute

    unsigned short name_pos_x; //display position in video window

    unsigned short name_pos_y;

    unsigned int time_attr; //display time attribute

    unsigned short time_pos_x; // display position in video window

    unsigned short time_pos_y;

    unsigned int shelter1_attr; //shelter rectangle 1
```

```
unsigned short shelter1_pos_x; //shelter rectangle 1 left top position video window

unsigned short shelter1_pos_y;

unsigned short shelter1_width;

unsigned short shelter1_height;

unsigned int shelter2_attrib; //shelter rectangle 2

unsigned short shelter2_pos_x; //shelter rectangle 2 left top position video window

unsigned short shelter2_pos_y;

unsigned short shelter2_width;

unsigned short shelter2_height;

char subtitle1[SUB_LEN]; //char rectangle 1 content

unsigned int subtitle1_attrib; // char rectangle 1 attribute

unsigned short subtitle1_pos_x; // char rectangle 1 position

unsigned short subtitle1_pos_y;

char subtitle2[SUB_LEN]; //char rectangle 2 content

unsigned int subtitle2_attrib; // char rectangle 2 attribute

unsigned short subtitle2_pos_x; // char rectangle 2 position

unsigned short subtitle2_pos_y;

char subtitle3[SUB_LEN]; //char rectangle 3 content

unsigned int subtitle3_attrib; // char rectangle 3 attribute

unsigned short subtitle3_pos_x; // char rectangle 3 position

unsigned short subtitle3_pos_y;

char subtitle4[SUB_LEN]; //char rectangle 3 content

unsigned int subtitle4_attrib; // char rectangle 3 attribute

unsigned short subtitle4_pos_x; // char rectangle 3 position
```

```
    unsigned short subtitle4_pos_y;
```

```
}ISIL_OSD_CFG;
```