Content

[1. SDK Directory Structure 9](#_Toc33711156)

[2. SDK Compile 9](#_Toc33711157)

[Windows Platform 9](#_Toc33711158)

[Linux Platform 9](#_Toc33711159)

[Other platforms 9](#_Toc33711160)

[3. SDK Interface 9](#_Toc33711161)

[4. Initialize and exit 10](#_Toc33711162)

[sdks\_dev\_init 10](#_Toc33711163)

[sdks\_dev\_quit 10](#_Toc33711164)

[sdks\_free\_result 11](#_Toc33711165)

[5. Callback function 11](#_Toc33711166)

[SDK\_DISCONN\_CB 11](#_Toc33711167)

[SDK\_CONNECT\_CB 12](#_Toc33711168)

[SDK\_STREAM\_CB 13](#_Toc33711169)

[SDK\_ALARM\_CB 15](#_Toc33711170)

[SDK\_PLAY\_TIME\_CB 17](#_Toc33711171)

[SDK\_DETECT\_CB 17](#_Toc33711172)

[SDK\_FACE\_CB 20](#_Toc33711173)

[SDK\_INTERCOM\_DB\_CB 22](#_Toc33711174)

[6. Connection 22](#_Toc33711175)

[sdks\_dev\_conn 22](#_Toc33711176)

[sdks\_dev\_conn\_ssl 23](#_Toc33711177)

[sdks\_conn\_close 24](#_Toc33711178)

[sdks\_dev\_get\_con\_sta 24](#_Toc33711179)

[7. Scene 25](#_Toc33711180)

[sdks\_dev\_addr\_req 25](#_Toc33711181)

[sdks\_dev\_live\_start 26](#_Toc33711182)

[sdks\_dev\_chg\_stream 27](#_Toc33711183)

[sdks\_dev\_live\_stop 28](#_Toc33711184)

[8. Playback 28](#_Toc33711185)

[sdks\_dev\_pb\_date\_list 28](#_Toc33711186)

[sdks\_dev\_pb\_chns\_in\_date 29](#_Toc33711187)

[sdks\_dev\_pb\_get\_rec\_list 30](#_Toc33711188)

[sdks\_dev\_pb\_start 31](#_Toc33711189)

[sdks\_dev\_pb\_seek 32](#_Toc33711190)

[sdks\_dev\_pb\_pause 32](#_Toc33711191)

[sdks\_dev\_pb\_resume 33](#_Toc33711192)

[sdks\_dev\_pb\_stop 33](#_Toc33711193)

[9. Record 34](#_Toc33711194)

[sdks\_dev\_open\_rec 34](#_Toc33711195)

[sdks\_dev\_record 35](#_Toc33711196)

[sdks\_dev\_stop\_rec 35](#_Toc33711197)

[sdks\_dev\_get\_record\_policy 36](#_Toc33711198)

[sdks\_dev\_get\_record\_state 38](#_Toc33711199)

[sdks\_dev\_get\_last\_record\_time 39](#_Toc33711200)

[10. Capture 40](#_Toc33711201)

[sdks\_get\_snap\_picture 40](#_Toc33711202)

[sdks\_get\_snap\_data 41](#_Toc33711203)

[sdks\_open\_snap 42](#_Toc33711204)

[sdks\_close\_snap 43](#_Toc33711205)

[11. Multimedia 44](#_Toc33711206)

[Android initialization 44](#_Toc33711207)

[sdks\_md\_live\_start 44](#_Toc33711208)

[sdks\_md\_live\_stop 45](#_Toc33711209)

[sdks\_md\_chg\_stream 46](#_Toc33711210)

[sdks\_md\_pb\_start 47](#_Toc33711211)

[sdks\_md\_pb\_seek 48](#_Toc33711212)

[sdks\_md\_pb\_pause 48](#_Toc33711213)

[sdks\_md\_pb\_resume 49](#_Toc33711214)

[sdks\_md\_pb\_stop 49](#_Toc33711215)

[sdks\_md\_rec\_start 50](#_Toc33711216)

[sdks\_md\_rec\_start\_width\_time 51](#_Toc33711217)

[sdks\_md\_rec\_percent 52](#_Toc33711218)

[sdks\_md\_rec\_stop 52](#_Toc33711219)

[sdks\_md\_capture 53](#_Toc33711220)

[sdks\_md\_audio\_start 54](#_Toc33711221)

[sdks\_md\_audio\_stop 54](#_Toc33711222)

[sdks\_md\_rec\_download\_start 55](#_Toc33711223)

[sdks\_md\_rec\_download\_stop 56](#_Toc33711224)

[sdks\_view\_zoomin 56](#_Toc33711225)

[sdks\_view\_zoominout\_centern 57](#_Toc33711226)

[12. Alarm 58](#_Toc33711227)

[sdks\_dev\_start\_alarm 58](#_Toc33711228)

[sdks\_dev\_stop\_alarm 59](#_Toc33711229)

[sdks\_dev\_get\_alarm\_list 59](#_Toc33711230)

[sdks\_dev\_ manual\_alarmout 61](#_Toc33711231)

[sdks\_dev\_get\_io\_alarm\_event 62](#_Toc33711232)

[sdks\_dev\_json\_set\_io\_alarm\_event 64](#_Toc33711233)

[sdks\_dev\_get\_disk\_alarm\_para 66](#_Toc33711234)

[sdks\_dev\_json\_set\_disk\_alarm\_para 68](#_Toc33711235)

[sdks\_dev\_json\_ get\_match\_alarm\_date\_list 70](#_Toc33711236)

[sdks\_dev\_json\_get\_alarm\_camera\_info\_list 72](#_Toc33711237)

[13. Equipment capabilities 73](#_Toc33711238)

[sdks\_dev\_get\_hw\_cap 73](#_Toc33711239)

[sdks\_dev\_ json\_get\_hw\_cap 74](#_Toc33711240)

[sdks\_dev\_get\_video\_cap 76](#_Toc33711241)

[sdks\_dev\_get\_sw\_cap 89](#_Toc33711242)

[sdks\_dev\_ json\_get\_sw\_cap 89](#_Toc33711243)

[sdks\_dev\_get\_audio\_cap 91](#_Toc33711244)

[sdks\_dev\_ json\_get\_audio\_cap 91](#_Toc33711245)

[sdks\_dev\_get\_language\_cap 94](#_Toc33711246)

[sdks\_dev\_get\_nw\_cap 95](#_Toc33711247)

[sdks\_dev\_get\_time\_zone\_cap 97](#_Toc33711248)

[sdks\_dev\_get\_osd\_cap 98](#_Toc33711249)

[sdks\_dev\_get\_ptz\_cap 102](#_Toc33711250)

[14. Basic equipment parameters 108](#_Toc33711251)

[sdks\_dev\_get\_general\_info 108](#_Toc33711252)

[sdks\_dev\_ json\_get\_general\_info 109](#_Toc33711253)

[sdks\_dev\_get\_dev\_name 110](#_Toc33711254)

[sdks\_dev\_ json\_get\_dev\_name 111](#_Toc33711255)

[sdks\_dev\_set\_dev\_name 112](#_Toc33711256)

[sdks\_dev\_ json\_set\_dev\_name 112](#_Toc33711257)

[sdks\_dev\_get\_dev\_time 113](#_Toc33711258)

[sdks\_dev\_ json\_get\_dev\_time 114](#_Toc33711259)

[sdks\_dev\_set\_dev\_time 115](#_Toc33711260)

[sdks\_dev\_ json\_set\_dev\_time 115](#_Toc33711261)

[sdks\_dev\_get\_dev\_ntp 116](#_Toc33711262)

[sdks\_dev\_ json\_get\_dev\_ntp 117](#_Toc33711263)

[sdks\_dev\_set\_dev\_ntp 118](#_Toc33711264)

[sdks\_dev\_ json\_set\_dev\_ntp 119](#_Toc33711265)

[sdks\_dev\_get\_dev\_id 120](#_Toc33711266)

[sdks\_dev\_set\_dev\_id 120](#_Toc33711267)

[sdks\_dev\_get\_dev\_port 121](#_Toc33711268)

[sdks\_dev\_ json\_get\_dev\_port 122](#_Toc33711269)

[sdks\_dev\_set\_dev\_port 123](#_Toc33711270)

[sdks\_dev\_ json\_set\_dev\_port 124](#_Toc33711271)

[sdks\_dev\_get\_dev\_language 125](#_Toc33711272)

[sdks\_dev\_set\_dev\_language 126](#_Toc33711273)

[sdks\_dev\_get\_security\_para 126](#_Toc33711274)

[sdks\_dev\_set\_security\_para 127](#_Toc33711275)

[sdks\_get\_dev\_time\_zone 128](#_Toc33711276)

[sdks\_set\_dev\_time\_zone 129](#_Toc33711277)

[sdks\_get\_p2p\_para 130](#_Toc33711278)

[sdks\_dev\_get\_nvr\_channel\_name（for NVR） 131](#_Toc33711279)

[sdks\_dev\_get\_nvr\_chn\_info（for NVR） 133](#_Toc33711280)

[sdks\_get\_protocol\_security\_param（for IPC） 134](#_Toc33711281)

[sdks\_set\_protocol\_security\_param（for IPC） 135](#_Toc33711282)

[sdks\_dev\_get\_video\_system 136](#_Toc33711283)

[sdks\_dev\_set\_video\_system 137](#_Toc33711284)

[15. PTZ operation 137](#_Toc33711285)

[PTZ operation type table： 137](#_Toc33711286)

[sdks\_dev\_open\_ptz 140](#_Toc33711287)

[sdks\_dev\_close\_ptz 141](#_Toc33711288)

[sdks\_dev\_ptz\_stop 141](#_Toc33711289)

[sdks\_dev\_ptz\_rotate 142](#_Toc33711290)

[sdks\_dev\_ptz\_zoom 143](#_Toc33711291)

[sdks\_dev\_ptz\_focus 143](#_Toc33711292)

[sdks\_dev\_ptz\_iris 144](#_Toc33711293)

[sdks\_dev\_ptz\_preset 144](#_Toc33711294)

[sdks\_dev\_ptz\_track 145](#_Toc33711295)

[sdks\_dev\_ptz\_scan 146](#_Toc33711296)

[sdks\_dev\_ptz\_tour 147](#_Toc33711297)

[sdks\_dev\_ptz\_keeper 148](#_Toc33711298)

[sdks\_dev\_ptz\_threeDimensionalPos 149](#_Toc33711299)

[Operate the 3D ptz function of the PTZ.去掉 149](#_Toc33711300)

[sdks\_dev\_ptz\_brush 149](#_Toc33711301)

[sdks\_dev\_ptz\_light 150](#_Toc33711302)

[sdks\_dev\_ptz\_defog 151](#_Toc33711303)

[sdks\_dev\_ptz\_postion 151](#_Toc33711304)

[sdks\_dev\_get\_ptz\_postion 152](#_Toc33711305)

[sdks\_dev\_get\_ptz\_req 153](#_Toc33711306)

[sdks\_dev\_set\_ptz\_speed 155](#_Toc33711307)

[sdks\_dev\_get\_ptz\_configue 155](#_Toc33711308)

[sdks\_dev\_get\_ptz\_timer 167](#_Toc33711309)

[sdks\_dev\_set\_ptz\_timer 169](#_Toc33711310)

[16. Video parameter 170](#_Toc33711311)

[sdks\_get\_video\_param 170](#_Toc33711312)

[sdks\_set\_video\_param 173](#_Toc33711313)

[sdks\_get\_roi\_param 175](#_Toc33711314)

[sdks\_set\_roi\_param 176](#_Toc33711315)

[sdks\_get\_svc\_stream\_para 177](#_Toc33711316)

[17. Network parameters 178](#_Toc33711317)

[sdks\_get\_net\_param 178](#_Toc33711318)

[sdks\_set\_net\_param 180](#_Toc33711319)

[sdks\_get\_ddns\_param 181](#_Toc33711320)

[sdks\_set\_ddns\_param 182](#_Toc33711321)

[sdks\_dev\_get\_ddns\_provider(for IPC) 183](#_Toc33711322)

[sdks\_dev\_get\_ftp 184](#_Toc33711323)

[sdks\_dev\_set\_ftp 185](#_Toc33711324)

[sdks\_dev\_get\_smtp 187](#_Toc33711325)

[sdks\_dev\_set\_smtp 188](#_Toc33711326)

[sdks\_dev\_get\_mtu 189](#_Toc33711327)

[sdks\_dev\_set\_mtu 189](#_Toc33711328)

[sdks\_get\_ip\_filter\_param 190](#_Toc33711329)

[sdks\_set\_ip\_filter\_param 191](#_Toc33711330)

[sdks\_dev\_get\_8021x 192](#_Toc33711331)

[sdks\_dev\_set\_8021x 193](#_Toc33711332)

[sdks\_dev\_get\_pppoe 194](#_Toc33711333)

[sdks\_dev\_set\_pppoe 195](#_Toc33711334)

[sdks\_dev\_get\_port\_mapping 196](#_Toc33711335)

[sdks\_dev\_set\_port\_mapping 197](#_Toc33711336)

[18. Front end image configuration 199](#_Toc33711337)

[sdks\_set\_sensor\_para 199](#_Toc33711338)

[sdks\_reset\_sensor\_to\_last\_param 200](#_Toc33711339)

[19. System Maintenance 200](#_Toc33711340)

[sdks\_dev\_reboot 200](#_Toc33711341)

[sdks\_dev\_reset 201](#_Toc33711342)

[20. user management 202](#_Toc33711343)

[sdks\_modify\_password\_param 202](#_Toc33711344)

[sdks\_operator\_privilege\_user 202](#_Toc33711345)

[21. Face 205](#_Toc33711346)

[sdks\_get\_face\_detect\_param(for IPC) 205](#_Toc33711347)

[sdks\_set\_face\_detect\_param(for IPC) 208](#_Toc33711348)

[sdks\_dev\_face\_detect\_start(for IPC) 210](#_Toc33711349)

[sdks\_dev\_face\_detect\_stop(for IPC) 211](#_Toc33711350)

[sdks\_dev\_face\_get\_group\_num(for NVR) 211](#_Toc33711351)

[sdks\_dev\_face\_get\_member(for NVR) 212](#_Toc33711352)

[sdks\_dev\_face\_check\_data(for NVR) 214](#_Toc33711353)

[sdks\_dev\_face\_get\_statis(for NVR) 215](#_Toc33711354)

[sdks\_dev\_face\_get\_attendance\_data(for NVR) 217](#_Toc33711355)

[sdks\_start\_database(for NVR) 218](#_Toc33711356)

[sdks\_dev\_get\_database\_index(for NVR) 219](#_Toc33711357)

[sdks\_dev\_get\_database\_info(for NVR) 222](#_Toc33711358)

[sdks\_stop\_database(for NVR) 223](#_Toc33711359)

[sdks\_start\_face(for NVR) 224](#_Toc33711360)

[sdks\_stop\_face(for NVR) 225](#_Toc33711361)

[sdks\_face\_get\_group(for NVR) 225](#_Toc33711362)

[sdks\_get\_face\_all\_node(for NVR) 226](#_Toc33711363)

[sdks\_get\_face\_by\_node(for NVR) 229](#_Toc33711364)

[sdks\_face\_add\_group(for NVR) 231](#_Toc33711365)

[sdks\_face\_rename\_group(for NVR) 232](#_Toc33711366)

[sdks\_face\_del\_group(for NVR) 233](#_Toc33711367)

[sdks\_face\_get\_group\_type(for NVR) 234](#_Toc33711368)

[sdks\_face\_add\_group\_type(for NVR) 235](#_Toc33711369)

[sdks\_face\_del\_group\_type(for NVR) 236](#_Toc33711370)

[sdks\_add\_face\_data\_to\_group（for NVR） 237](#_Toc33711371)

[sdks\_del\_face\_data（for NVR） 238](#_Toc33711372)

[sdks\_mod\_face\_data（for NVR） 239](#_Toc33711373)

[22. Other function 240](#_Toc33711374)

[sdks\_get\_device\_log 240](#_Toc33711375)

[sdks\_update\_nvr 244](#_Toc33711376)

[sdks\_get\_dev\_list 245](#_Toc33711377)

[sdks\_get\_blind\_param 246](#_Toc33711378)

[sdks\_set\_blind\_param 247](#_Toc33711379)

[sdks\_get\_mot\_param 248](#_Toc33711380)

[sdks\_set\_mot\_param 250](#_Toc33711381)

[sdks\_get\_osd\_param 250](#_Toc33711382)

[sdks\_set\_osd\_param 257](#_Toc33711383)

[23. Thermal imaging 263](#_Toc33711384)

[sdks\_dev\_get\_thermal\_cap 263](#_Toc33711385)

[sdks\_get\_thermal\_param 264](#_Toc33711386)

[sdks\_set\_thermal\_param 267](#_Toc33711387)

[sdks\_get\_thermal\_area\_temperature\_measure 269](#_Toc33711388)

[sdks\_set\_thermal\_area\_temperature\_measure 272](#_Toc33711389)

[sdks\_get\_thermal\_area\_feature\_temperature 274](#_Toc33711390)

[sdks\_get\_thermal\_one\_point\_temperature 276](#_Toc33711391)

[sdks\_get\_thermal\_any\_point\_temperature 277](#_Toc33711392)

[sdks\_get\_map\_relation 278](#_Toc33711393)

[sdks\_set\_map\_relation 280](#_Toc33711394)

[sdks\_get\_temperature\_calibration 281](#_Toc33711395)

[sdks\_set\_temperature\_calibration 283](#_Toc33711396)

[sdks\_get\_thermal\_alarm\_linkage\_param 284](#_Toc33711397)

[sdks\_set\_thermal\_alarm\_linkage\_param 294](#_Toc33711398)

[sdks\_test\_thermal\_bad\_point\_correct 298](#_Toc33711399)

[sdks\_set\_thermal\_bad\_point\_correct 298](#_Toc33711400)

[sdks\_reset\_thermal\_bad\_point\_correct 299](#_Toc33711401)

[sdks\_get\_thermal\_version 299](#_Toc33711402)

[24. License plate 301](#_Toc33711403)

[sdks\_get\_lpr\_link\_param 301](#_Toc33711404)

[sdks\_set\_lpr\_link\_param 302](#_Toc33711405)

[sdks\_lpr\_ipfilter\_list\_add 304](#_Toc33711406)

[sdks\_lpr\_ipfilter\_list\_delete 305](#_Toc33711407)

[sdks\_lpr\_ipfilter\_list\_modify 306](#_Toc33711408)

[sdks\_get\_lpr\_ipfilter\_list\_num 308](#_Toc33711409)

[sdks\_get\_lpr\_ipfilter\_list 309](#_Toc33711410)

[sdks\_lpr\_ipfilter\_list\_search\_open 311](#_Toc33711411)

[sdks\_lpr\_ipfilter\_list\_search\_get 312](#_Toc33711412)

[sdks\_lpr\_ipfilter\_list\_search\_close 314](#_Toc33711413)

[sdks\_lpr\_ipfilter\_list\_file\_download 315](#_Toc33711414)

[sdks\_get\_ai\_multi\_object\_detect\_param 316](#_Toc33711415)

[sdks\_set\_ai\_multi\_object\_detect\_param 321](#_Toc33711416)

[sdks\_get\_ai\_multi\_object\_detect\_ability 327](#_Toc33711417)

[25 . Intelligent Analysis 329](#_Toc33711418)

[sdks\_get\_ia\_version 329](#_Toc33711419)

[sdks\_get\_ia\_perimeter\_ability 331](#_Toc33711420)

[sdks\_get\_ia\_perimeter\_param 334](#_Toc33711421)

[sdks\_set\_ia\_perimeter\_param 338](#_Toc33711422)

[sdks\_get\_ia\_svf\_ability 342](#_Toc33711423)

[sdks\_get\_ia\_svf\_param 345](#_Toc33711424)

[sdks\_set\_ia\_svf\_param 350](#_Toc33711425)

[sdks\_get\_ia\_dvf\_ability 354](#_Toc33711426)

[sdks\_get\_ia\_dvf\_param 357](#_Toc33711427)

[sdks\_set\_ia\_dvf\_param 361](#_Toc33711428)

[sdks\_get\_ia\_loiter\_ability 366](#_Toc33711429)

[sdks\_get\_ia\_loiter\_param 368](#_Toc33711430)

[sdks\_set\_ia\_loiter\_param 372](#_Toc33711431)

[sdks\_get\_ia\_multi\_loiter\_ability 377](#_Toc33711432)

[sdks\_get\_ia\_ multi\_loiter\_param 380](#_Toc33711433)

[sdks\_set\_ia\_ multi\_loiter\_param 383](#_Toc33711434)

[sdks\_get\_ia\_object\_left\_ability 388](#_Toc33711435)

[sdks\_get\_ia\_ object\_left \_param 390](#_Toc33711436)

[sdks\_set\_ia\_ object\_left \_param 393](#_Toc33711437)

[sdks\_get\_ia\_object\_removed\_ability 398](#_Toc33711438)

[sdks\_get\_ia\_ object\_removed\_param 400](#_Toc33711439)

[sdks\_set\_ia\_ object\_removed\_param 403](#_Toc33711440)

[sdks\_get\_ia\_abnormal\_speed\_ability 406](#_Toc33711441)

[sdks\_get\_ia\_abnormal\_speed\_param 409](#_Toc33711442)

[sdks\_set\_ia\_abnormal\_speed\_param 413](#_Toc33711443)

[sdks\_get\_ia\_converse\_ability 417](#_Toc33711444)

[sdks\_get\_ia\_converse\_param 419](#_Toc33711445)

[sdks\_set\_ia\_converse\_param 423](#_Toc33711446)

[sdks\_get\_ia\_legal\_parking\_ability 426](#_Toc33711447)

[sdks\_get\_ia\_legal\_parking\_ param 428](#_Toc33711448)

[sdks\_set\_ia\_legal\_parking\_ param 432](#_Toc33711449)

[sdks\_get\_ia\_signal\_bad\_ability 435](#_Toc33711450)

[sdks\_get\_ia\_signal\_bad\_ param 436](#_Toc33711451)

[sdks\_set\_ia\_signal\_bad\_ param 438](#_Toc33711452)

[sdks\_get\_ia\_advanced\_ability 441](#_Toc33711453)

[sdks\_get\_ia\_advanced\_param 444](#_Toc33711454)

[sdks\_set\_ia\_advanced\_param 446](#_Toc33711455)

[26. Fisheye（For fisheye equipment） 448](#_Toc33711456)

[sdks\_get\_fisheye\_ability 448](#_Toc33711457)

[sdks\_get\_fisheye\_video\_layout 451](#_Toc33711458)

[sdks\_get\_fisheye\_param 453](#_Toc33711459)

[sdks\_set\_fisheye\_param 454](#_Toc33711460)

[27. Voice intercom 455](#_Toc33711461)

[sdks\_md\_talk\_start 455](#_Toc33711462)

[sdks\_md\_talk\_stop 456](#_Toc33711463)

[sdks\_dev\_send\_audio\_data 456](#_Toc33711464)

[28. Panoramic dynamic video 457](#_Toc33711465)

[sdks\_md\_play\_dynamic\_video 457](#_Toc33711466)

[sdks\_md\_set\_view\_port 458](#_Toc33711467)

[sdks\_md\_set\_auto\_cruise 459](#_Toc33711468)

[sdks\_md\_mouse\_button\_operate 459](#_Toc33711469)

[sdks\_md\_mouse\_drag\_operate 460](#_Toc33711470)

[sdks\_md\_mouse\_scroll\_operate 461](#_Toc33711471)

[sdks\_md\_set\_pano\_show\_mode 461](#_Toc33711472)

[sdks\_md\_set\_fisheye\_mount\_type 462](#_Toc33711473)

[sdks\_md\_set\_fisheye\_ceiling\_desktop\_show\_mode 463](#_Toc33711474)

[sdks\_md\_set\_fisheye\_wall\_show\_mode 463](#_Toc33711475)

[sdks\_md\_get\_fisheye\_rectify\_parameters 464](#_Toc33711476)

[sdks\_md\_set\_fisheye\_rectify\_parameters 466](#_Toc33711477)

[Appendix 468](#_Toc33711478)

[1.1 Alarm type 469](#_Toc33711479)

[1.2 group privilege manage type 472](#_Toc33711480)

[1.3 private area operate type 472](#_Toc33711481)

[1.4 PTZ operate type 472](#_Toc33711482)

[1.5 PTZ parameter configuration 475](#_Toc33711483)

[1.6 Front end configuration type 476](#_Toc33711484)

[1.7 Log type 480](#_Toc33711485)

[1.8 Alarm source type 482](#_Toc33711486)

[1.9 Action type 484](#_Toc33711487)

[1.10 Fisheye mode 484](#_Toc33711488)

[1.11 Dynamic video mode 485](#_Toc33711489)

[1.12 Equipment type 487](#_Toc33711490)

[1.13 Language type 488](#_Toc33711491)

[1.14 Coding type 490](#_Toc33711492)

**SDK Interface definition**

## SDK Directory Structure

Directory includes inc, lib, and example. The inc directory contains the SDK's header files, sdks.h and sdk\_def.h, which need to reference the sdks.h header file when using the SDK interface; the lib directory is the SDK library file; the example directory contains examples of SDK usage.

## SDK Compile

### Windows Platform

The library for the Windows platform includes win\_sdk.lib and win\_sdk.dll. When compiling with VS, you need to add the WIN32 and SDKS\_USE\_DLL preprocessor macros. Need to reference the sdks.h header file when using the SDK interface.

### Linux Platform

The Linux platform includes imx6\_sdk.a and -lpthread is included in the link options. You need to reference the sdks.h header file when using the SDK interface.

### Other platforms

Add sdks.h and the corresponding library file to the project. You need to reference the sdks.h header file when using the SDK interface.

## SDK Interface

The SDK contains some external interfaces. The upper layer only uses these interfaces to access the device, and does not need to care about the specific details of the communication with the device. Including initialization and exit, connection, site, playback, alarms, equipment capabilities and equipment information, etc.

## Initialize and exit

### sdks\_dev\_init

#### Define

int sdks\_dev\_init (const char\* p\_json\_setup\_in)

#### Instruction

To initialize the SDK, you must call the initialization interface before using other interfaces. The initialized items include the default settings, resource requests, and so on. P\_json\_setup\_in is a text in json format. Currently this value is not used. Set the parameter to NULL.

#### Return Value

Successful return 0, error returns other values.

### sdks\_dev\_quit

#### Define

void sdks\_dev\_quit ()

#### Instruction

Deinitialize the SDK and release the resources requested by the SDK.

#### Return value

None。

### sdks\_free\_result

#### Define

void sdks\_free\_result (void\* p\_result)

#### Instruction

Some interfaces will return some result data. After the caller finishes using these data, it needs sdks\_free\_result to release the memory.

#### Return value

None。

## Callback function

### SDK\_DISCONN\_CB

#### Define

typedef void(\*SDK\_DISCONN\_CB)(unsigned int handle, void\* p\_obj);

#### parameter

handle：sdks\_dev\_conn interface Return value

p\_obj: The context of the callback function, passed in by the user.

#### Instruction

When the session is disconnected, notify the upper layer.

#### Return value

None。

### SDK\_CONNECT\_CB

#### Define

typedef void(\*SDK\_CONNECT\_CB)(unsigned int handle, void\* p\_obj);

#### parameter

handle：Return connection handle, 0 indicates failure, greater than 0 indicates success.

p\_obj: The context of the callback function, passed in by the user.

#### Instruction

When a non-blocking connection is called, the connection callback function is used and the connection result is returned to the upper layer through the callback function.

#### Return value

None。

### SDK\_STREAM\_CB

#### Define

typedef void(\*SDK\_STREAM\_CB)(unsigned int handle, int stream\_id, void\* p\_data, void\* p\_obj);

#### parameter

handle：Return value of sdks\_dev\_conn interface

stream\_id： Stream\_id returned when live or playback streaming is turned on

p\_data：Audio and video data. For detailed information, please refer to ST\_AVFrameData definition of sdk\_def.h

p\_obj: The context of the callback function, passed in by the user.

ST\_AVFrameDatadefinition：

|  |  |  |
| --- | --- | --- |
| parameter | Instruction | comment |
| nStreamFormat | Stream format | 1 indicates the original stream, 2 indicates the TS mixed stream, 3 indicates the original encrypted stream, and 4 indicates the PS mixed stream |
| nESStreamType | Original stream type | 1 for video, 2 for audio |
| nEncoderType | Coding format | MPEG4 coding：0  H264 coding：1  MJPEG coding：2  SVC coding：3  JPEG coding：6  H265(base)：7  H265(main)：8  H265(high)：9  G7231 coding：101  G711A coding：102  G711U coding：103  G722 coding：104  G726 coding：105  G729 coding：106  AMR coding：107  PCM coding：108 |
| nCameraNo | Camera number |  |
| nSequenceId | Data frame number |  |
| nFrameType | Data frame type | 1 indicates an I frame, 2 indicates a P frame, and 0 indicates an unknown type |
| nAbsoluteTimeStamp | Absolute time stamp | Microseconds |
| nRelativeTimeStamp | Relative time stamp | Microseconds |
| pszData | Audio and video data |  |
| nDataLength | The length of pszData |  |
| nFrameRate | Frame rate | This variable is obsolete |
| nBitRate | Rate | This variable is obsolete |
| nImageFormatId | Image format | This variable is obsolete |
| nImageWidth | The width of the video |  |
| nImageHeight | The height of the video |  |
| nVideoSystem | Video format | PAL Standard：1  NTSC Standard：2 |
| nFrameBufLen | Buffer length | This variable is obsolete |
| nStreamId | Flow ID | This variable is obsolete |
| nTimezone | Time zone |  |
| nDaylightSavingTime | summer time |  |

#### Instruction

When audio and video streams arrive, stream data is returned to the upper layer via this callback function.

#### Return value

None。

### SDK\_ALARM\_CB

#### Define

typedef void(\*SDK\_ALARM\_CB)(unsigned int handle, char\* p\_data, void\* p\_obj);

#### parameter

handle：Return value of sdks\_dev\_conn interface

p\_data：The content of the alarm，The format is json

{

"data":

{

"dev\_ip":"xxx.xxx.xxx.xxx",

"dev\_type":xxx,

"main\_type": xxx,

"sub\_type": xxx,

"dev\_id":"xxx"

"src\_type":"xxx"

"src\_id":"xxx"

"alarm\_flag":"xxx"

}

"SNPointList":

[

{

"X":"xxx"

"Y:"xxx"

},

……

]

" Thermal":

{

"MaxTemperature\_X": “xxx”, //High-temperature abscissa (expressed as

a percentage)

"MaxTemperature\_Y": “xxx”, //High-temperature ordinate (expressed as a

percentage)

"MaxTemperature": “xxx”, //Maximum temperature value

"MinTemperature": “xxx”, //Minimum temperature value

"TemperatureThreshold": “xxx” //alarm threshold

}

}

例：

{

"data": {

"dev\_ip": "192.168.89.73",

"src\_type": 31,

"src\_id": 0,

"dev\_id": "180726",

"dev\_type": 0,

"main\_type": 7,

"sub\_type": 1,

"alarm\_flag": 1,

"time": "2019-10-14 15:45:35"

},

"SNPointList": [],

"AlarmAreaList": [],

" Thermal": {

"MaxTemperature\_X": 0,

"MaxTemperature\_Y": 0,

"MaxTemperature": 60.799999,

"MinTemperature": 51.599998,

"TemperatureThreshold": 30

}

}

p\_obj: The context of the callback function, which is passed in by the user.

The coordinate values are as follows：

Current code stream width \* width ratio = actual x value

Current code stream high\*high ratio=actual y value.

#### Instruction

When an alarm message is generated, the message is returned to the upper layer through this callback function。

#### Return value

None。

### SDK\_PLAY\_TIME\_CB

#### 定义

typedef void(\*SDK\_PLAY\_TIME\_CB)(unsigned int handle, int stream\_id, void\* p\_obj, const char\* p\_time);

#### 参数

handle：return value of sdks\_dev\_conn

stream\_id：stream id

p\_obj: Context of callback function. Input by user

p\_time: time stamp of current video stream

#### instruction

Return time stamp of current playing video stream when playing live view or playing back.

#### Return value

Null. No return value

### SDK\_DETECT\_CB

Definition

typedef void(\*SDK\_DETECT\_CB)(unsigned int handle, int stream\_id, void\*\* p\_result, void\* p\_data, void\* p\_obj);

handle：return value of sdks\_dev\_conn

stream\_id：stream id

p\_obj: Context of callback function. Input by user

p\_data: image data。

p\_result：target information, pls check the following graph for details

json format

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | | | Description | |
| Magic | | | Magic number，default: 0xffff ffff | |
| Vesion | | | Load protocal version number | |
| TotalLen | | | Total structure length | |
| PictureLen | | | Image data length | |
| FullImageWidth | | | Full image width，Full\_corp=0 is valid | |
| FullImageHeight | | | Full image height，Full\_corp=0 is valid | |
| SequenceID | | | Sequence ID serves as index of every image and increases | |
| Full\_crop | | | Full image and small image 0（full）1（small） | |
| TargetSize | | | Detect target number | |
| TargetDetectList | | | Target detect list | |
|  |  | TargeId | Target Id |
|  |  | Type | type：face，human body，car plate，car etc. |
|  |  | X | X coordinate of start point |
|  |  | Y | Y coordinate of start point |
|  |  | W | Square width |
|  |  | H | Square height |
|  |  | [Attr](https://cn.bing.com/dict/search?q=describe&FORM=BDVSP6&mkt=zh-cn)DataLen | Attribute data length |
|  | PersonFace | | face：type =0 | |
|  |  | Confidence | Confidence | |
|  |  | Yaw | Yaw degree | |
|  |  | Tilt | Tilt degree | |
|  |  | Pitch | Pitch degree | |
|  |  | Gender | Gender degree | |
|  |  | Age | Age | |
|  |  | Complexion | Complexion(skin color) | |
|  |  | minority | Race | |
|  |  | Temperature | Face temperatur | |
|  |  | LandMark\_size | Number of landmark. | |
|  |  | LandMark\_Data | Landmark data，one landmark is 4 byte. | |
|  | PersonBody | | Human body：type=2 | |
|  |  | Confidence | Confidence | |
|  |  | Gender | Gender | |
|  |  | Age | Age | |
|  |  | Complexion | Complexion(skin color) | |
|  |  | Backpack | Have backpack or not | |
|  |  | Human\_move | Moving or not | |
|  |  | Move\_direction | Moving direction | |
|  |  | Ride\_bike | Riding bike or not | |
|  |  | Ride\_motorbike | Riding motorbike or not | |
|  |  | Human\_face\_visible | Human face is visible or not | |
|  | Plate | | Car plate：type=3 | |
|  |  | Have\_plate | Have car plate or not ,0 no,1 yes | |
|  |  | Plate\_angleH | Horizontal angle | |
|  |  | Plate\_angleV | Horizontal angle | |
|  |  | Plate\_color | Car plate color | |
|  |  | Plate\_type | Car plate type | |
|  |  | Plate\_confidence | Car plate confidence | |
|  |  | Plate\_country | Car plate country | |
|  |  | Char\_num | Character numbers on the plate | |
|  |  | Plate\_num | Car plate number | |
|  |  | Plate\_char\_confidence | confidence of every character on the car plate | |
|  | Vehicle | | car：type=4 | |
|  |  | Have\_plate | Have car or not ,0 no,1 yes | |
|  |  | Vehicle\_ angleH | Horizontal angle | |
|  |  | Vehicle\_ angleV | Horizontal angle | |
|  |  | Vehicle\_ color | Vehicle color | |
|  |  | Vehicle\_ type | Vehicle color | |
|  |  | Vehicle\_confidence | Vehicle confidence | |
|  |  | Vehicle\_model | Vehicle type，big/small vehicle | |
|  |  | Vehicle\_speed | Vehicle speed | |
|  |  | Vehicle\_moving | Car is moving or not | |
|  |  | Char\_num | Character numbers of vehicle plate | |
|  |  | Plate\_num | Plate number | |
|  |  | Plate\_char\_confidence | confidence of every character on the car plate | |
|  |  | Move\_direction | Vehicle moving direction | |
|  |  | Move\_direction\_confidence | Confidence of Vehicle moving direction | |
|  |  | Vehicle\_facing | Vehicle facing direction | |
|  |  | Vehicle\_facing\_confidence | Confidence of vehicle facing direction | |
|  |  | trademark\_utf8 | Vehicle brand | |
|  |  | trademark\_utf8\_confidence | Confidence of vehicle brand | |

#### instruction

return time stamp of current playing video stream when playing live view or playing back

#### return value

Null.

### SDK\_FACE\_CB

#### Define

typedef void(\*SDK\_FACE\_CB)(unsigned int handle, int pic\_type,void\* p\_data, int \*data\_len, void\*\* p\_result, void\* p\_obj);//pic\_type: 1 bottom database, 2 real-time database

#### parameter

handle：Return value of sdks\_dev\_conn interface.

pic\_type: 1 represents the picture of the bottom library, 2 represents the picture of the real-time Library

p\_obj: The context of the callback function, passed in by the user (usually empty).

p\_data: Picture data.

data\_len: Picture size, that is, the length of picture data (unit: k).

p\_result：The image related information is output in JSON format. The JSON fields of the real-time library and the bottom library are different.

Bottom Library: the JSON format of p\_result is：

{

"key\_id": 832523, // Indexes

"name": "44333",// Face name

"identity": "Admin123 // Face ID

"group": "sunell", // Face database

"type": "Teacher",// Face types

"birthday": 1569340800000000, // Face date of birth, in subtle units

"s\_time": 0, // Effective start time in microseconds

"e\_time": 0, // Effective end time in microseconds

"gender": 1// Face gender

}

Real time library: the JSON format of p\_result is：

{

"key\_id": 1, // Indexes

"channel": 0, // Channel ID

"similarity": 0, // Similarity degree

"time": "2019-12-14 09-30-01 ", // Capture time in us

"name": "baby1", // Face name

"identity": "ssssww", // Face number

"group": "", // Face database

"type": "", // Face types

"birthday": 1576108800000000, // Face birthday in us

"gender": 0 // Face gender

}

**Description**

#### When the user queries the corresponding picture information of face base database or real-time database according to the index, the currently queried picture information data is returned.

#### Return value

Null.

### SDK\_INTERCOM\_DB\_CB

#### Define

typedef void(\*SDK\_INTERCOM\_DB\_CB)(unsigned int db, void\*p\_obj);

#### parameter

db：sReturn value of sdks\_dev\_conn interface.

p\_obj: The context of the callback function, passed in by the user (usually empty).

**Description**

#### Return value

Null.

## Connection

### sdks\_dev\_conn

#### Define

unsigned int sdks\_dev\_conn (const char\* p\_ip, unsigned short port, const char\* p\_user, const char\* p\_passwd, SDK\_DISCONN\_CB disconn\_cb, void\* p\_obj)

#### parameter

p\_ip：Device ip

port：Device port

p\_user：user name

p\_passwd:user password，encryption

disconn\_cb:Disconnected callback function

p\_obj: disconn\_cb Callback function context

#### Instruction

Connect the device and verify the user's legitimacy. When the connection is disconnected, the caller is notified via the disconn\_cb callback function that the connection has been disconnected。

SDK\_DISCONN\_CB is the definition of the callback function, the function definition please refer to sdk\_def.h

#### Return value

Return value as a connection handle represents a session. Return value greater than 0 indicates that the call is successful, and Return value 0 indicates that the call failed.

### sdks\_dev\_conn\_ssl

unsigned int sdks\_dev\_conn\_ssl(const char\* p\_ip, unsigned short port, const char\* p\_user, const char\* p\_passwd, SDK\_DISCONN\_CB disconn\_cb, void\* p\_obj);

#### parameter

p\_ip：device ip

port：device port

p\_user：user name

p\_passwd: user password, encrypted

disconn\_cb: disconnect the callback function

p\_obj: disconn\_cb, Context of callback function

#### instruction

SSL is used to establish the connection device and verify the legitimacy of users. When the connection is disconnected, the caller is notified through the disconn\_ cb to callback function that the connection is disconnected

SDK\_DISCONN\_CB is the definition of callback function. Pls refer to sdk\_def.h for definition of the function.

Return value

Return is connection handle and indicates one session. Return value bigger than 0 indicates function callis successful. 0 indicates failure.

### sdks\_conn\_close

#### Define

void sdks\_dev\_conn\_close (int handle);

#### parameter

handle：Return value of sdks\_dev\_conn interface

#### Instruction

Close the connection handle。

#### Return value

None

### sdks\_dev\_get\_con\_sta

#### define

int sdks\_dev\_get\_con\_sta (unsigned int handle);

#### parameter

handle：return value of sdks\_dev\_conn

#### instruction

get connection status

#### return value

-1: connection disconnect

0 ：connection connected

## Scene

### sdks\_dev\_addr\_req

#### Define

int sdks\_dev\_addr\_req(unsigned int handle, int ipprotover, char\*\* p\_result);

#### Parameter

handle: Return value of sdks\_dev\_conn interface

ipprotover: IP protocol type, 1：IPv4 2：IPv6

p\_result: Output parameters, fill in the media address parameters to the caller, return in json format, if the value returned is not NULL, need to call sdks\_free\_result function to free memory, otherwise it will lead to memory leak.

#### Instruction

The media address is used to create the video data channel.

#### Return value

Returns 0 if successful, non-zero means failure

#### Media address information parameter table

|  |  |  |
| --- | --- | --- |
| Attributes | | Description |
| Parameters | | The message begins |
| p\_result | | Search information |
|  | ip | Media server IP |
| port | Media server port |
| ip\_prover | IP protocol type：1．IPv4 2．IPv6 |

### sdks\_dev\_live\_start

#### Define

int sdks\_dev\_live\_start(unsigned int handle, int chn, int stream\_type, SDK\_STREAM\_CB stream\_cb, void\* p\_obj)

#### parameter

handle：Return value of sdks\_dev\_conn interface

chn：channel

stream\_type：Stream type, refer to the definition of video\_stream\_type\_e in sdk\_def.h，STREAM\_TYPE\_1 is HD, STREAM\_TYPE\_2 is SD, and STREAM\_TYPE\_3 is smooth. STREAM\_TYPE\_3 is not currently supported.

stream\_cb: The video stream callback function is defined in sdk\_def.h.

p\_obj:The context of stream\_cb

#### Instruction

Turn on live video streaming.

#### Return value

Return stream\_id, >=0 for success, <0 for failure. Stream\_id needs to be used when switching between stream and stop stream.

### sdks\_dev\_chg\_stream

#### Define

int sdks\_dev\_chg\_stream(unsigned int handle, int stream\_id, int new\_stream\_type)

#### parameter

handle：Return value of sdks\_dev\_conn interface

stream\_id：Return value of sdks\_dev\_live\_start or sdks\_dev\_chg\_stream

New\_stream\_type: The stream type. You can refer to the definition of video\_stream\_type\_e in sdk\_def.h. STREAM\_TYPE\_1 is HD, STREAM\_TYPE\_2 is SD, and STREAM\_TYPE\_3 is smooth. STREAM\_TYPE\_3 is not currently supported

#### Instruction

Switch video streams

#### Return value

Return stream\_id, >=0 for success, <0 for failure. Stream\_id needs to be used when switching between stream and stop stream.

### sdks\_dev\_live\_stop

#### Define

int sdks\_dev\_live\_stop(unsigned int handle, int stream\_id)

#### parameter

handle：Return value of sdks\_dev\_conn interface

stream\_id：Return value of sdks\_dev\_live\_start or sdks\_dev\_chg\_stream

#### Instruction

Turn off video streaming

#### Return value

Returns 0 success, other failures.

## Playback

### sdks\_dev\_pb\_date\_list

#### Define

int sdks\_dev\_pb\_date\_list(unsigned int handle, int chn, const char\* s\_date, const char\* e\_date, char\*\* p\_result)

#### parameter

handle：Return value of sdks\_dev\_conn interface

chn：Channel

s\_date：The starting time, the format must be yyyy-mm-dd, for example 2018-02-18

e\_date：End time, the format must be yyyy-mm-dd, for example 2018-02-20

p\_result：out parameter，Return to the caller in the format of json. After using it, you need to use sdks\_free\_result to release the memory.

#### Instruction

Get a list of dates for a channel from s\_date to video in e\_date

#### Return value

Returns 0 success, other failures.

### sdks\_dev\_pb\_chns\_in\_date

#### Define

int sdks\_dev\_pb\_chns\_in\_date(unsigned int handle, const char\* p\_date, char\*\* p\_result);

#### parameter

handle：Return value of sdks\_dev\_conn interface

p\_date：Date, the format must be yyyy-mm-dd, for example, 2018-02-18

p\_result：Out parameter, which is returned to the caller in the format of json. After using it, you need to use sdks\_free\_result to release the memory.

#### Instruction

Get a list of video channels in a certain day

#### Return value

Returns 0 success, other failures.

### sdks\_dev\_pb\_get\_rec\_list

#### Define

int sdks\_dev\_pb\_get\_rec\_list(unsigned int handle, int chn, const char\* p\_date, char\*\* p\_result);

#### Parameter

handle：Return value of sdks\_dev\_conn interface

chn：channel

p\_date：Date, the format must be yyyy-mm-dd, for example 2018-09-10 12:23:00

p\_result：The out parameter is returned to the caller in json format. After use, you need to use sdks\_free\_result to release the memory.

#### Instruction

Get a list of time periods in a channel that have videos in a day

{

"data":

[

{

"is\_alarm":false,

"s\_time":"2018-09-10 00:00:00",

"e\_time":"2018-09-10 07:23:00"

},

…

]

}

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter field | Description | Whether it is necessary | Remarks |
| is\_alarm | Channel number | Y | Whether alarm is recorded |
| s\_time | Starting time | Y | The start time of a recording period |
| e\_time | ending time | Y | End time of a recording period |

#### Return value

Returned to 0 successfully, other values failed.

### sdks\_dev\_pb\_start

#### Define

int sdks\_dev\_pb\_start(unsigned int handle, int chn, int stream\_type, const char\* s\_time, const char\* e\_time, SDK\_STREAM\_CB stream\_cb, void\* p\_obj)

#### parameter

handle：Return value of sdks\_dev\_conn interface

chn：Channel

stream\_type：For the stream type, refer to the definition of video\_stream\_type\_e in sdk\_def.h. STREAM\_TYPE\_1 is HD, STREAM\_TYPE\_2 is SD, and STREAM\_TYPE\_3 is smooth. STREAM\_TYPE\_3 is not currently supported

s\_time：Play time point

e\_time：At the end of the time, the current parameter is not used and can be passed a NULL value

stream\_cb: The video stream callback function is defined in sdk\_def.h.

p\_obj:The context of stream\_cb

#### Instruction

Request a video stream at a time

#### Return value

Return stream\_id, >=0 for success, <0 for failure. Stream\_id is needed to pause, resume, and stop playback.

### sdks\_dev\_pb\_seek

#### Define

int sdks\_dev\_pb\_seek(unsigned int handle, int stream\_id, const char\* time

#### parameter

handle：Return value of sdks\_dev\_conn interface

stream\_id：Return value of sdks\_dev\_pb\_start

time：Seek time

#### Instruction

Jump to a point in time for playback

#### Return value

Returns 0 success, other failures.

### sdks\_dev\_pb\_pause

#### Define

int sdks\_dev\_pb\_pause(unsigned int handle, int stream\_id)

#### parameter

handle：Return value of sdks\_conn interface

stream\_id：sdks\_dev\_pb\_start’s Return value

#### Instruction

Pause video streaming

#### Return value

Returns 0 success, other failures.

### sdks\_dev\_pb\_resume

#### Define

int sdks\_dev\_pb\_resume(unsigned int handle, int stream\_id)

#### parameter

handle：Return value of sdks\_conn interface

stream\_id：Return value of sdks\_dev\_pb\_start

#### Instruction

Continue playing video stream

#### Return value

Returns 0 success, other failures.

### sdks\_dev\_pb\_stop

#### Define

int sdks\_dev\_pb\_stop(unsigned int handle, int stream\_id)

#### parameter

handle：Return value of sdks\_conn interface

stream\_id：Return value of sdks\_dev\_pb\_start

#### Instruction

Stop playing video stream

#### Return value

Returns 0 success, other failures.

## Record

### sdks\_dev\_open\_rec

#### Define

int sdks\_dev\_open\_rec(const char\* p\_path);

#### parameter

p\_path：The directory of save the video, notice is only directory, example /root/video/, video module will automatic generate the MP4 file in the directory.

#### Instruction

When need to record the video of the channel, need to call this function to point the record video saved directory, this function will return record\_id which use for distinguish different record video stream.

#### Return Value

Returns >=0 success, Return <0 failures.

### sdks\_dev\_record

#### Define

int sdks\_dev\_record(int record\_id, ST\_AVFrameData\* p\_frame);

#### Parameter

record\_id：record id，return by function: sdks\_dev\_open\_rec，use for distinguish different channel record.

p\_frame：Audio or video frame. ST\_AVFrameData structure Define please refer to chapter 5 of SDK\_STREAM\_CB

#### Instruction

Record the channel audio and video frame as MP4 file

#### Return Value

Returns 0 success, other failures.

### sdks\_dev\_stop\_rec

#### Define

int sdks\_dev\_stop\_rec(int record\_id);

#### Parameter

record\_id：record id，return by function: sdks\_dev\_open\_rec, use for distinguish the different channel video.

#### Instruction

Stop record the audio and video of the channel.

#### Return Value

Returns 0 success, other failures.

### sdks\_dev\_get\_record\_policy

#### Define

int sdks\_dev\_get\_record\_policy(unsigned int handle, int chn, int record\_mode, char\*\* p\_result);

#### Parameter

handle: sdks\_conn interface Return Value；

chn: Channel

record\_mode：record id，return by function: sdks\_dev\_open\_rec, use for distinguish the different channel video.

p\_result: Output Parameter，fill record schduel Parameter for caller, return as json format, if return value is not NULL, need call sdks\_free\_result function to realse the memory， otherwise will cause a memory leak.

#### Instruction

Get record policy information.

#### Return Value

Returns 0 success, other failures.

#### Record Policy Parameter List

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Property | | | | | Instruction |
| Parameters | | | | | Message begin |
| p\_result | | | | | Process result |
| record\_policy\_list | | | | | Record policy list |
|  | record\_policy | | | |  |
|  | record\_open\_flag | | | Record policy open flag |
| record\_mode | | | Record mode：1：record，2：Snapshot |
| dev\_id | | | Device ID |
| channel\_id | | | Channel ID |
| stream\_id | | | Stream Id |
| save\_days | | | Record save days |
| audio\_open\_flag | | | Record audio or not |
| disk\_id | | | Disk ID |
| schedule\_time\_list | | | Record time list |
|  | schedule\_time | |  |
|  | weekday | Weekday |
| start\_time | Start time [0，86400] |
| end\_time | End time [0，86400] |
| snap\_shot\_para | | | Snapshot Parameter |
|  | duration | | Duration |
| quality | | Image quality |
| interval | | Snapshot interval |
| shoot\_times | | Snapshot times |
| video\_solution | | Resolution |
|  | video\_height | Video height |
| video\_width | Video width |
| resolution\_name | Resolution name |

### sdks\_dev\_get\_record\_state

#### Define

int sdks\_dev\_get\_record\_state(unsigned int handle, int chn, char\*\* p\_result);

#### Parameter

handle: sdks\_conn interface return value

chn: Channel

p\_result: Output Parameter，fill record status information Parameter for caller, return as json format, if return value is not NULL, need call sdks\_free\_result function to realse the memory， otherwise will cause a memory leak.

#### Instruction

Get record status information

#### Return Value

Returns 0 success, other failures.

#### Record status information Parameter list

|  |  |  |  |
| --- | --- | --- | --- |
| Property | | | Instruction |
| Parameters | | | Message bgin |
| p\_result | | | Process result |
| state\_list | | | Record status information list |
|  | state | |  |
|  | dev\_id | Device ID |
| channel\_id | Channel ID |
| state | Recording tatus, check Appendix 3.5 for details |

Sample information is as follows :

state\_list":[{"dev\_id":"DVRJYBAICJ52N347","channel\_id":1,"state":0},{"dev\_id":"DVRJYBAICJ52N347","channel\_id":2,"state":-7},{"dev\_id":"DVRJYBAICJ52N347","channel\_id":3,"state":-7},{"dev\_id":"DVRJYBAICJ52N347","channel\_id":4,"state":-7}]}

### sdks\_dev\_get\_last\_record\_time

#### Define

int sdks\_dev\_get\_last\_record\_time(unsigned int handle, const char\* s\_time, const char\* e\_time, char\* p\_qry\_info, char\*\* p\_result);

#### Parameter

handle: sdks\_conn interface return value

s\_time:Start time Format must be: "yyyy-mm-dd HH:mm:ss"

e\_time:End time，Format must be: "yyyy-mm-dd HH:mm:ss"

p\_qry\_info：Input Parameter，Inquiry information

p\_result：Output Parameter，fill the latest recording time Parameter for caller，return as json format, if return value is not NULL, need call sdks\_free\_result function to realse the memory， otherwise will cause a memory leak.

#### Instruction

Inquiry the latest recording time

#### Return Value

Returns 0 success, other failures.

#### Input Parameter Inquiry information Parameter list

|  |  |  |
| --- | --- | --- |
| Property | | Instruction |
| Parameters | | Message Begin |
| p\_qry\_info | | Inquiry information |
|  | record\_mode | Inquiry mode (record inquiry or Snapshot inquiry) |
| dev\_id | Device ID |
| channel\_id | Channel ID |
| major\_type | Main type |
| minor\_type | Monor type |
| precision | Precision |
| record\_interval | Inquiry interval time Length (max time length of each interval) |
| select\_mode | Inquiry mode (-1: All;1: according to time; 2: according to stye) |

#### Output Parameter latest recording time Parameter list

|  |  |  |
| --- | --- | --- |
| Property | | Instruction |
| Parameters | | Message begin |
| p\_result | | Process result |
|  | datetime | Latest recording time |

Sample information is as follows :

{"datetime":"2019- 1- 3 8:21:15"}

## Capture

### sdks\_get\_snap\_picture

#### Define

#### int sdks\_get\_snap\_picture(unsigned int handle,char\* p\_snap\_param);

#### parameter

handle： return value of sdks\_dev\_conn

p\_snap\_param： input parameter，enter in json format。

The json example format is as follows：

{"chn":1,"timeout":2,"quality":9,"path":"f:\\"}

|  |  |  |  |
| --- | --- | --- | --- |
| Key Field | instruction | selected | Remarks |
| chn | Capture channel | Y | IPC , type：1 |
| timeout | overtime time | Y | unit:sec |
| quality | Picture quality | Y | 1-9 |
| path | Image save path | Y |  |

#### instruction

Grab the image of the remote camera and save it to a directory named after the current time。

#### returned value

0:successful，non-0:failed。

### sdks\_get\_snap\_data

#### Define

#### int sdks\_get\_snap\_data(unsigned int handle,char\* p\_snap\_param,char \*\*p\_buf,int \*p\_len);

#### parameter

handle： return value of sdks\_dev\_conn。

p\_snap\_param： input parameter，enter in json format。

The json example format is as follows：：

{"chn":1,"timeout":2,"quality":9}

|  |  |  |  |
| --- | --- | --- | --- |
| Key Field | Instruction | Selected | Remarks |
| chn | Capture channel | Y | IPC , type：1 |
| timeout | overtime time | Y | unit:sec |
| quality | Picture quality | Y | 1-9 |

p\_buf ： Output parameter, the address of the picture, the external caller needs to release the memory, otherwise it will cause memory leak。

**p\_**len： Output parameter, image size

#### instruction

Grab a picture of a remote camera。

#### returned value

0:successful，non-0:failed。

### sdks\_open\_snap

#### Define

#### int sdks\_open\_snap(unsigned int handle, char\* p\_snap\_param)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

p\_snap\_param： Input parameters, expressed in json format.

the json format is：

{"chn":1,"timeout":2,"quality":9}

|  |  |  |  |
| --- | --- | --- | --- |
| Key Field | Instruction | Selected | Remarks |
| chn | Capture channel | Y | IPC , type：1 |
| timeout | overtime time | Y | unit:sec |
| quality | Picture quality | Y | 1-9 |

**Description**

#### Open the snapshot (when the IPC device software version is higher than 3.6, the snapshot interface can be called only after the snapshot is opened).

#### Return value

0 succeeded; other values failed.

### sdks\_close\_snap

#### Define

#### int sdks\_close\_snap(unsigned int handle, char\* p\_snap\_param)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

p\_snap\_param： Input parameters, expressed in json format.

the json format is：

{"chn":1,"timeout":2,"quality":9}

|  |  |  |  |
| --- | --- | --- | --- |
| Key Field | Instruction | Selected | Remarks |
| chn | Capture channel | Y | IPC , type：1 |
| timeout | overtime time | Y | unit:sec |
| quality | Picture quality | Y | 1-9 |

**Description**

#### Turn off the capture (when the IPC device software version is higher than 3.6, turn on the capture to capture the picture, and then call this interface to turn off the capture).

#### Return value

0 succeeded; other values failed.

## Multimedia

### Android initialization

#### definition

#### int sdks\_init\_android(void\* javaVm);

#### parameter

javaVm：java virtual environment

#### introduction

Initialize android environment before call the multimedia interface。

#### Return value

sucess return 0; fail return others。

### sdks\_md\_live\_start

#### definition

#### int sdks\_md\_live\_start(unsigned int handle, int chn, int stream\_type, void\* p\_wnd, bool is\_hw\_dec, SDK\_PLAY\_TIME\_CB play\_cb, void\* p\_obj)

#### parameter

handle：sdks\_dev\_conn return value .

chn：channel .

stream\_type：for stream type, pls refer to the definition of video\_stream\_type\_e in sdk\_def.h，STREAM\_TYPE\_1: HD，STREAM\_TYPE\_2: D1，STREAM\_TYPE\_ 3: Smooth. STREAM\_TYPE\_3 is not supported currently.

p\_wnd: video display window

is\_hw\_dec: hardware acceleration. True: hardware acceleration. But abnormal decode may happen in some phone model type. False: no acceleration. This will cost more performance, but there will not be phone compatibility issue.

play\_cb: return to the current time of playing video stream 。

p\_obj: context of play\_cb

#### introduction

get stream of certain channel and play in certain window

#### return value

return stream\_id，>=0 refers to success, <0 refers to fail. stream\_id: used when switching stream and close stream

### sdks\_md\_live\_stop

#### definition

#### int sdks\_md\_live\_stop(unsigned int handle, int stream\_id);

#### parameter

handle：sdks\_dev\_conn return value

stream\_id：return value of sdks\_md\_live\_start

#### introduction

close video stream

#### return value

0: success，others: fail

### sdks\_md\_chg\_stream

#### definition

#### int sdks\_md\_chg\_stream(unsigned int handle, int stream\_id, int new\_stream\_type);

#### parameter

handle：sdks\_dev\_conn return value

stream\_id：return value of sdks\_md\_live\_start

new\_stream\_type: stream type. Please refer to the definition of video\_stream\_type\_e of sdk\_def.h. STREAM\_TYPE\_1: HD，STREAM\_TYPE\_2: D1，STREAM\_TYPE\_ 3: Smooth. STREAM\_TYPE\_3 is not supported currently

#### introduction

switch live view

#### return value

0: success，others: fail

### sdks\_md\_pb\_start

#### definition

#### int sdks\_md\_pb\_start(unsigned int handle, int chn, int stream\_type, const char\* s\_time, void\* p\_wnd, bool is\_hw\_dec, SDK\_PLAY\_TIME\_CB play\_cb, void\* p\_obj)

#### parameter

handle：sdks\_dev\_conn return value .

chn：channel.

stream\_type：stream type. Please refer to the definition of video\_stream\_type\_e of sdk\_def.h. STREAM\_TYPE\_1: HD，STREAM\_TYPE\_2: D1，STREAM\_TYPE\_ 3: Smooth. STREAM\_TYPE\_3 is not supported currently

s\_time：play time.

p\_wnd：video display window

is\_hw\_dec: hardware acceleration. True: hardware acceleration. But abnormal decode may happen in some phone model type. False: no acceleration. This will cost more performance, but there will not be phone compatibility issue.

play\_cb: return to the current time of playing video stream 。

p\_obj: context of play\_cb

#### introduction

request the play back video of certain time

#### return value

return stream\_id，>=0 refers to success, <0 refers to fail. stream\_id: used when pause, continue and stop playing video

### sdks\_md\_pb\_seek

#### definition

int sdks\_md\_pb\_seek(unsigned int handle, int stream\_id, const char\* time);

#### parameter

handle：sdks\_dev\_conn return value

stream\_id：return value of sdks\_md\_pb\_start

time：time point of seek

#### introduction

jump to certain time and play

#### return value

0: success，others: fail

### sdks\_md\_pb\_pause

#### definition

int sdks\_md\_pb\_pause(unsigned int handle, int stream\_id);

#### parameter

handle：sdks\_conn return value

stream\_id：return value of sdks\_md\_pb\_start

#### introduction

pause video play

#### Return value

0: success，others: fail

### sdks\_md\_pb\_resume

#### definition

int sdks\_md\_pb\_resume(unsigned int handle, int stream\_id);

#### parameter

handle：sdks\_conn return value

stream\_id：return value of sdks\_md\_pb\_start

#### introduction

continue video play

#### Return

0: success，others: fail

### sdks\_md\_pb\_stop

#### definition

int sdks\_md\_pb\_stop(unsigned int handle, int stream\_id);

#### parameter

handle：sdks\_conn return value

stream\_id：return value of sdks\_md\_pb\_start

#### introduction

stop video play

#### return value

0: success，others: fail

### sdks\_md\_rec\_start

#### definition

int sdks\_md\_rec\_start(unsigned int handle, int stream\_id, const char\* p\_path);

#### parameter

handle：sdks\_conn return value

stream\_id：return value of sdks\_md\_pb\_start

p\_path：record storage path. Do not include file name. eg. /usr/

#### introduction

start recording of video stream

#### return value

0: success，others: fail

### sdks\_md\_rec\_start\_width\_time

#### definition

int sdks\_md\_rec\_start\_width\_time(unsigned int handle, int stream\_id, const char\* s\_time, const char\* e\_time, const char\* p\_path);

#### parameter

handle：sdks\_conn return value

stream\_id：return value of sdks\_md\_pb\_start

s\_time：recording start time

e\_time：recording stop time

p\_path：record storage path. Do not include file name. eg. /usr/

#### introduction

start recording of video stream

#### return value

0: success，others: fail

### sdks\_md\_rec\_percent

#### definition

int sdks\_md\_rec\_percent(unsigned int handle, int stream\_id);

#### parameter

handle：sdks\_conn return value

stream\_id：return value of sdks\_md\_pb\_start

#### introduction

get the percentage of current record.

#### Return value

return the percentage of current record. Not beginning returns 0; record completion returns 100.

### sdks\_md\_rec\_stop

#### definition

int sdks\_md\_rec\_stop(unsigned int handle, int stream\_id);

#### parameter

handle：sdks\_conn return value

stream\_id：return value of sdks\_md\_pb\_start

#### introduction

stop recording of video stream

#### return value

0: success，others: fail

### sdks\_md\_capture

#### definition

int sdks\_md\_capture(unsigned int handle, int stream\_id, const char\* p\_path);

#### parameter

handle：sdks\_conn return value

stream\_id：return value of sdks\_md\_pb\_start

p\_path：picture storage path including file name. eg. /usr/xxx.jpg

#### introduction

snapshot

#### return value

0: success，others: fail

### sdks\_md\_audio\_start

#### definition

int sdks\_md\_audio\_start(unsigned int handle, int stream\_id);

#### parameter

handle：sdks\_conn return value

stream\_id：return value of sdks\_md\_pb\_start or sdks\_md\_live\_start

#### introduction

open audio stream and play audio

#### return value

0: success，others: fail

。

### sdks\_md\_audio\_stop

#### definition

int sdks\_md\_audio\_stop(unsigned int handle, int stream\_id);

#### parameter

handle：sdks\_conn return value

stream\_id：return value of sdks\_md\_pb\_start or sdks\_md\_live\_start

#### introduction

close audio stream and stop playing audio

#### return value

0: success，others: fail

### sdks\_md\_rec\_download\_start

#### definition

int sdks\_md\_rec\_download\_start (unsigned int handle, int chn, int stream\_type, const char\* s\_time, const char\* e\_time, const char\* p\_path);

#### parameter

handle：sdks\_conn return value

chn：channel

stream\_type：stream type, please refer to definition of video\_stream\_type\_e in sdk\_def.h, STREAM\_TYPE\_1 is HD，STREAM\_TYPE\_2 is SD，STREAM\_TYPE\_3 is Smooth, STREAM\_TYPE\_3 is not support.

s\_time：start time of downloading record

e\_time：end time of downloading record.

p\_path：record saving path, cannot contain the file name. eg: /usr/

#### introduction

Download a record in a certain period of time.

#### return value

Return stream\_id，>=0 means succeed， <0 means failed.

stream\_id if interrupt is needed, close the downloading record. sdks\_md\_rec\_download\_startwill use stream\_id。

### sdks\_md\_rec\_download\_stop

#### definition

SDKS\_API int sdks\_md\_rec\_download\_stop(unsigned int handle, int stream\_id);

#### parameter

handle：sdks\_conn return value

stream\_id：sdks\_md\_rec\_download\_start return value

#### introduction

invoke if downloading need to be stopped, otherwise it will be stopped automatically once the downloading is finished.

#### return value

return 0 means succeed, other values mean failed.

### sdks\_view\_zoomin

#### definition

SDKS\_API int sdks\_view\_zoomin(unsigned int handle, int stream\_id, int x, int y, int w, int h);

handle：sdks\_conn return value

#### parameter

handle：sdks\_conn return value

stream\_id：sdks\_md\_pb\_start or sdks\_md\_live\_start return value

x: coordinate value of top left corner

y: coordinate value of top right corner

w: width

h: height

#### introduction

video zommin (selected area)

#### return value

return 0 means succeed, other values mean failed.

### sdks\_view\_zoominout\_centern

#### definition

SDKS\_API int sdks\_view\_zoominout\_centern(unsigned int handle, int stream\_id, int scale);

#### parameter

handle：sdks\_conn return value

stream\_id：sdks\_md\_pb\_start or sdks\_md\_live\_startreturn value

scale: the scale times of the original video, please refer to sdks\_zoomin\_graduate\_e in header file sdk\_def.h

#### introduction

video zoomout (center point)

#### return value

return 0 means succeed, other values mean failed.

## Alarm

### sdks\_dev\_start\_alarm

#### Define

int sdks\_dev\_start\_alarm(unsigned int handle, SDK\_ALARM\_CB alarm\_cb, void\* p\_obj)

#### parameter

handle：Return value of sdks\_dev\_conn interface

alarm\_cb：The callback function of the alarm message. The alarm message is returned in json format. Please refer to the definition of sdk\_def.h

  The context of p\_obj:alarm\_cb

#### Instruction

Turn on the alarm, when there is alarm in the equipment, return the alarm message to the upper layer

#### Return value

Returns 0 success, other failures.

### sdks\_dev\_stop\_alarm

#### Define

int sdks\_dev\_stop\_alarm(unsigned int handle)

#### parameter

handle：Return value of sdks\_dev\_conn interface

#### Instruction

Stop alarm

#### Return value

Returns 0 success, other failures.

### sdks\_dev\_get\_alarm\_list

#### Define

int sdks\_dev\_get\_alarm\_list(unsigned int handle, int chn, const char\* s\_time, const char\* e\_time, char\*\* p\_result)

#### parameter

handle：Return value of sdks\_dev\_conn interface

chn：Channel，If -1 is passed, all Channel's alarm data will be returned

s\_time：Start time, the format must be "yyyy-mm-dd HH:mm:ss"

e\_time：End time, the format must be "yyyy-mm-dd HH:mm:ss"

p\_result：The output parameter returns the alarm information in json format. If the returned value is not NULL, it needs to call the sdks\_free\_result function to release the memory, otherwise it will cause a memory leak. The sdks\_dev\_get\_alarm\_list function returns up to 60 alarm records at a time。

The json format is as follows：

{

"is\_more":false

"data":

[

{

"dev\_ip":"xxx.xxx.xxx.xxx",

"src\_type":xxx,

"src\_id":xxx,

"dev\_id":"xxx",

"dev\_type":xxx,

"main\_type": xxx,

"sub\_type": xxx,

"time":"yyyy-mm-dd HH:mm:ss"

},

......

]

}

|  |  |  |  |
| --- | --- | --- | --- |
| parameter | Instruction | Y/N | Note |
| is\_more | Is there more alarm | Y | If the alarm record is less than 60 (including 60), is\_more is false; if it is greater than 60, the function will only return 60 records and set is\_moreparameter to true. The upper layer can use the time of the last alarm record as s\_time to obtain subsequent records |
| dev\_ip | Model ip | Y |  |
| src\_type | Alarm source type | Y |  |
| src\_id | Alarm source id | Y | Alarm source ID, IO alarm refers to alarm input device Id, motion detection alarm refers to ChannelID, disk alarm refers to disk ID |
| dev\_id | Model Id | Y |  |
| dev\_type | Equipment type | Y |  |
| main\_type | Alarm main type | Y | See Appendix[1.1.1 Alarm main type](#_1.1.1__Alarm) |
| sub\_type | Alarm sub type | Y | See Appendix[1.1.2 Alarm sub type](#_1.1.2__Alarm) |
| time | The time when the alarm occurred | Y |  |

#### Instruction

Get the historical data of the alarm

#### Return value

Returns 0 success, other failures.

### sdks\_dev\_ manual\_alarmout

#### Define

Int sdks\_dev\_manual\_alarmout(unsigned int handle, int chn,const int alarmout\_id, int control\_flag);

#### Parameter

handle： sdks\_dev\_conn The return value of the interface

chn： channel

alarmout\_id： Alarm port id

control\_flag: Alarm status: 1 open, 0 closed

#### Instruction

Manual alarm

#### Return value

Return 0 success, other failures。

### sdks\_dev\_get\_io\_alarm\_event

#### Define

int sdks\_dev\_get\_io\_alarm\_event(unsigned int handle, int chn, int alarm\_source\_id, char\*\* p\_result);

#### parameter

handle： sdks\_dev\_conn The return value of the interface

chn:chanel ID

alarm\_source\_id：Alarm source id:io alarm refers to alarm input device id, mobile detection alarm refers to channel id, disk alarm refers to disk id

p\_result：Output parameter, populate the io alarm parameter to the caller, return in json format, if this value returned is non-null, the sks \_ free \_ result function needs to be called to release the memory, otherwise the memory leaks.

#### Instruction

Get IO alarm parameters

#### Returned value

Return 0 success, other non-0 failures

#### IO Alarm policy parameter table

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 属性 | | | | | | | | describe |
| Parameters | | | | | | | | Message Start |
| p\_result | | | | | | | | Processing result |
|  | io\_alarm\_list | | | | | | |  |
|  | io\_alarm\_event | | | | | |  |
|  | insource\_para | | | | | Query type: 1 query video; 2 check capture; 3 check alarm; 4 system log |
|  | src\_type | | | | Alarm source type,See Appendix [1.8 Alarm source type](#_1.8_Alarm_source) |
| src\_id | | | | Alarm source id |
| src\_name | | | | Alarm source name |
| enable | | | | Open a mark |
| alarm\_inval | | | | alarm interval |
| valid\_level | | | | active level |
| alarm\_time | | | | Alarm time |
|  | schedule\_time\_list | | |  |
|  | schedule\_time | | Plann time |
|  | weekday | Week |
| s\_time | start time |
| e\_time | End time |
| link\_para\_list | | | | |  |
|  | link\_para | | | | Linkage parameter |
|  | act\_type | | | Action type: see [1.9 Action type](#_1.9_Action_type) |
| act\_id | | | Action id, |
| ptz\_para\_list | | | | |  |
|  | ptz\_para | | | | Ptz action parameters |
|  | act\_type | | | Action type: see [1.9 Action type](#_1.9_Action_type) |
| act\_id | | | Action id |
| act\_name | | | Action name |
| ptz\_act\_type | | | Type of operation (preset, track, etc.) |
| ptz\_act\_id | | | Operate id (pre-set id, track id, etc., as previously set by the user, etc.) |
| ptz\_channel\_id | | | Ptz channel id |
| alarm\_out\_para\_list | | | | |  |
|  | alarm\_out\_para | | | | Alarm output action parameters |
|  | act\_type | | | Action type: see [1.9 Action type](#_1.9_Action_type) |
| act\_id | | | Action |
| act\_name | | | Action name |
| dev\_id | | | Device\_id |
| alarm\_out\_id | | | Id number of alarm output port |
| alarm\_out\_flag | | | Alarm output mark |
| event\_type\_id | | | Alarm event type |
| alarm\_time | | | Alarm output time |
| record\_para\_list | | | | |  |
|  | record\_para | | | | Linkage video parameters |
|  | act\_type | | | Linkage type: see [1.9 Action type](#_1.9_Action_type) |
| act\_id | | | Linkage id |
| act\_name | | | Linkage name |
| pre\_record\_flag | | | Pre-record mark |
| delay\_record\_time | | | Delay record time |

### sdks\_dev\_json\_set\_io\_alarm\_event

#### Define

int sdks\_dev\_json\_set\_io\_alarm\_para(unsigned int handle, const char\* p\_io\_alarm\_para);

#### parameter

handle: sdks\_dev\_conn Return value of the interface

p\_io\_alarm\_para: Input parameters, list of parameters for io alarm in json format

#### instruction

Set IO alarm policy parameters

#### return value

Return 0 success, other non-0 failures

#### IO Alarm policy parameter table

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| predicable | | | | | | | describe | |
| Parameters | | | | | | | Message Start | |
| p\_result | | | | | | | Processing result | |
|  | alarm\_event\_list | | | | | |  | |
|  | io\_alarm\_event | | | | |  | |
|  | insource\_para | | | | Query type: 1 query video; 2 check capture; 3 check alarm; 4 system log | |
|  | src\_type | | | Alarm source type,See Appendix [1.8 Alarm source type](#_1.8_Alarm_source) | |
| src\_id | | | Alarm source id | |
| src\_name | | | Alarm source name | |
| enable | | | Open mark | |
| alarm\_inval | | | alarm interval | |
| valid\_level | | | Valid level | |
| time\_list | | | |  | |
|  | time | | | Plan time | |
|  | weekday | | Weekday | |
| sTime | | start time | |
| eTime | | End time | |
| linkage\_para\_list | | | |  | |
|  | link\_para | | | Linkage parameter | |
|  | act\_type | | Action type: see [1.9 Action type](#_1.9_Action_type) | |
| act\_id | | Action id, | |
| ptz\_action\_para\_list | | | |  | |
|  | ptz\_para | | | Ptz action parameters | |
|  | act\_type | | Action type: see [1.9 Action type](#_1.9_Action_type) | |
| act\_id | | Action id | |
| act\_name | | Action name | |
| ptz\_act\_type | | Type of operation (preset, track, etc.) | |
| ptz\_act\_id | | Operate id (pre-set id, track id, etc., as previously set by the user, etc.) | |
| ptz\_channel\_id | | Ptz channel id | |
| alarm\_out\_action\_para\_list | | | |  | |
|  | alarm\_out\_para | | | Alarm output action parameters | |
|  | act\_type | | Action type: see [1.9 Action type](#_1.9_Action_type) | |
| act\_id | | action | |
| act\_name | | Action name | |
| dev\_id | | Device ID | |
| alarm\_out\_id | | Id number of alarm output port | |
| alarm\_out\_flag | | Alarm output flag | |
| event\_type\_id | | Alarm event type | |
| alarm\_time | | Alarm output time | |
| record\_action\_para\_list | | | |  | |
|  | record\_para | | | Linkage video parameters | |
|  | act\_type | | Linkage type: see [1.9 Action type](#_1.9_Action_type) | |
| act\_id | | Linkage id | |
| act\_name | | Linkage name | |
| pre\_record\_flag | | Pre-record flag | |
| delay\_time | | Delay record time | |

### sdks\_dev\_get\_disk\_alarm\_para

#### Define

SDKS\_API int sdks\_dev\_get\_disk\_alarm\_para(unsigned int handle, int chn, char\*\* p\_result);

#### Parameter

handle： sdks\_dev\_conn Return value of the interface

chn:channel id

p\_result：Output parameter, populate disk alarm parameter to caller, return in json format, if the value returned is not

NULL，need to call the sdks\_free\_result function to release memory, otherwise it will cause memory leakage。

#### Instruction

Get disk alarm parameters

#### returned value

Return 0 success, other non-0 failures

#### Disk alarm policy parameter table

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| predicable | | | | | | | | describe |
| Parameters | | | | | | | | Message Start |
| p\_result | | | | | | | | Process result |
|  | disk\_alarm\_list | | | | | | |  |
|  | disk\_alarm\_event | | | | | | Query type: 1 query video; 2 check capture; 3 check alarm; 4 system log |
|  | source\_para | | | | | Query type: 1 query video; 2 check capture; 3 check alarm; 4 system log |
|  | src\_type | | | | Alarm source type,See Appendix [1.8 Alarm source type](#_1.8_Alarm_source) |
| src\_id | | | | Alarm source id |
| src\_name | | | | Alarm source name |
| enable | | | | Enalble mark |
| alarm\_inval | | | | alarm interval |
| full\_alarm\_enable | | | | Enable disk full alarm flag. |
| alarm\_threshold | | | | Alarm threshold in percentage |
| error\_alarm\_enable | | | | Enable disk exception warning flag |
| no\_disk\_alarm\_enable | | | | Enable non-disk alarm flag |
| alarm\_time | | | | Alarm time |
|  | schedule\_time\_list | | |  |
|  | schedule\_time | | Schedule time |
|  | weekday | weekday |
| s\_time | Star time |
| e\_time | End time |
| link\_para\_list | | | | |  |
|  | link\_para | | | | Linkage parameter |
|  | act\_type | | | Action type: see [1.9 Action type](#_1.9_Action_type) |
| act\_id | | | Action id, |
| ptz\_para\_list | | | | |  |
|  | ptz\_para | | | | Ptz action parameters |
|  | act\_type | | | Action type: see the action type definition for details |
| act\_id | | | Action id |
| act\_name | | | Action name |
| ptz\_act\_type | | | Type of operation (preset, track, etc.) |
| ptz\_act\_id | | | Operate id (pre-set id, track id, etc., as previously set by the user, etc.) |
| ptz\_channel\_id | | | Ptz channel id |
| alarm\_out\_para\_list | | | | |  |
|  | alarm\_out\_para | | | | Alarm output action parameters |
|  | act\_type | | | Action type: see [1.9 Action type](#_1.9_Action_type) |
| act\_id | | | Action id |
| act\_name | | | Action Name |
| dev\_id | | | Device id |
| alarm\_out\_id | | | Id number of alarm output port |
| alarm\_out\_flag | | | Alarm output mark |
| event\_type\_id | | | Alarm event type |
| alarm\_time | | | Alarm output time |

The example information is as follows：

{"disk\_alarm\_list":[{"source\_para":{"src\_type":3,"src\_id":1,"src\_name":"","enable":0,"alarm\_inval":10,"full\_alrarm\_enable":0,"alarm\_threshold":95,"error\_alarm\_enable":0,"no\_disk\_alarm\_enable":0,"alarm\_time":{"schedule\_time\_list":[]}},"link\_para\_list":[],"ptz\_para\_list":[],"alarm\_out\_para\_list":[]}]}

### sdks\_dev\_json\_set\_disk\_alarm\_para

#### define

int sdks\_dev\_json\_set\_disk\_alarm\_para(unsigned int handle, const char\* p\_disk\_alarm\_para);

#### parameter

handle: sdks\_dev\_conn returned value of the interface

p\_disk\_alarm\_para: input parameter，Parameter list of disk Alert in json format

#### instruction

Parameter list of disk Alert in json format parameter

#### returned value

Return 0 success, other non-0 failures

#### Enter a parameter table for the parameter disk alert policy

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| nature | | | | | | | | describe |
| Parameters | | | | | | | | Message begins |
| p\_result | | | | | | | | Processing result |
|  | disk\_alarm\_list | | | | | | |  |
|  | disk\_alarm\_event | | | | | | Query type: 1 query video; 2 check capture; 3 check alarm; 4 system log |
|  | alarm\_source\_para | | | | | Query type: 1 query video; 2 check capture; 3 check alarm; 4 system log |
|  | src\_type | | | | Alarm source type,See Appendix [1.8 Alarm source type](#_1.8_Alarm_source) |
| src\_id | | | | Alarm source id |
| src\_name | | | | Alarm source name |
| enable | | | | Open Mark |
| alarm\_inval | | | | Alarm interval |
| disk\_full\_flag | | | | Enable disk full alarm flag. |
| threshold | | | | Alarm threshold in percentage |
| disk\_error\_flag | | | | Enable disk exception warning flag |
| no\_disk\_alarm\_flag | | | | Enable non-disk alarm flag |
| time\_list | | |  | | |
|  | time | | Plan time | | |
|  | weekday | week | | |
| sTime | Start time | | |
| eTime | End time | | |
| linkage\_para\_list | | | | |  |
|  | link\_para | | | | Linkage parameter |
|  | act\_type | | | Action type: see [1.9 Action type](#_1.9_Action_type) |
| act\_id | | | Action id, |
| ptz\_action\_para\_list | | | | |  |
|  | ptz\_para | | | | Ptz action parameter |
|  | act\_type | | | Action type: see [1.9 Action type](#_1.9_Action_type) |
| act\_id | | | Action id |
| act\_name | | | Action Name |
| ptz\_act\_type | | | Type of operation (preset, track, etc.) |
| ptz\_act\_id | | | Operate id (pre-set id, track id, etc., as previously set by the user, etc.) |
| ptz\_channel\_id | | | Ptz channel id |
| alarm\_out\_action\_para\_list | | | | |  |
|  | alarm\_out\_para | | | | parameterAlarm output action parameter. |
|  | act\_type | | | Action type: see [1.9 Action type](#_1.9_Action_type) |
| act\_id | | | action |
| act\_name | | | Action Name |
| dev\_id | | | device id |
| alarm\_out\_id | | | Id number of alarm output port |
| alarm\_out\_flag | | | Alarm output mark |
| event\_type\_id | | | Alarm event type |
| alarm\_time | | | Alarm output time |

### sdks\_dev\_json\_ get\_match\_alarm\_date\_list

#### Define

int sdks\_dev\_json\_get\_match\_alarm\_date\_list(unsigned int handle, const char \* p\_qry\_info, const char\* s\_time, const char\* e\_time, char \*\* p\_result);

#### parameter

handle: sdks\_dev\_conn returned value of the interface

p\_qry\_info:Enter query information in parameter,json format

s\_time：Start time, format must be "yyyy-mm-dd hh:mm:ss"

e\_time：End time, the format must be a "yyyy-mm-dd hh: mm: ss"

p\_result：Output parameter, fill the date list information to the caller that meets the requirements, and return it in json format. If the returned value is non-null, the sks \_ free \_ result function needs to be called to release the memory, otherwise it can cause the memory to leak.

#### instruction

Get the date list information that the alarm meets the requirements parameter

#### returned value

Return 0 success, other non-0 failures

#### Enter parameter query information parameter table

|  |  |  |
| --- | --- | --- |
| nature | | describe |
| Parameters | | Message begins |
| p\_qry\_info | | Processing result |
|  | record\_mode | Query Mode (Video Query or Snapshot Query) |
| dev\_id | device id |
| channel\_id | Channel id |
| major\_type | Major type |
| minor\_type | Minor type |
| precision | precision |
| record\_segment\_interval | Query period time length (maximum time span per segment) |
| select\_mode | Query mode (0: all;: query by type;: query by time) |

#### Output parameter time information parameter table

|  |  |  |  |
| --- | --- | --- | --- |
| nature | | | describe |
| Parameters | | | Message begins |
| p\_result | | | Processing result |
|  | date\_list | | Time list |
|  | date | temporal information |

The example information is as follows：

{"date\_list":[{"date":"2018-12-17"}]}

### sdks\_dev\_json\_get\_alarm\_camera\_info\_list

#### Define

int sdks\_dev\_json\_get\_alarm\_camera\_info\_list(unsigned int handle, const char\* s\_time, const char\* e\_time,const char\* p\_alarm\_info\_qry, char\*\* p\_result);

#### parameter

handle：sdks\_dev\_conn接口的returned value；

s\_time：Start time, format must be "yyyy-mm-dd hh:mm:ss"

e\_time：End time, the format must be a "yyyy-mm-dd hh: mm: ss"

p\_qry\_info:Enter query information in parameter,json format

p\_result：The output parameter, populates the alarm device information parameter to the caller and returns it in json format. If the returned value is not null, it needs to call the sdks\_free\_result function to free up memory, otherwise it will lead to memory leakage.

#### instruction

Get a list of information for the alarm device.

#### returned value

Return 0 success, other failures.

#### Enter parameter query information parameter table

|  |  |  |
| --- | --- | --- |
| nature | | describe |
| Parameters | | Message begins |
| p\_qry\_info | | Processing result |
|  | dev\_id | device id |
| dev\_ip | Equipment ip |
| select\_mode | Source id |
| src\_id | Retrieval mode |
| src\_name | Source Name |
| major\_type | Major type |
| minor\_type | Minor type |
| begin\_time | Alarm start time |
| end\_time | Alarm end time |

#### Output the parameter device information parameter table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| nature | | | | descritibe |
| Parameters | | | | Message begins |
|  | camera\_info\_list | | |  |
|  | Camera\_info | |  |
|  | dev\_id | Device id |
| camera\_id | Camera id |

## Equipment capabilities

### sdks\_dev\_get\_hw\_cap

#### Define

int sdks\_dev\_get\_hw\_cap(unsigned int handle, dev\_hw\_cap\_t\* p\_hw\_cap);

#### parameter

handle：Return value of sdks\_dev\_conn interface；

p\_hw\_cap：Fill the hardware parameter to the caller, refer to the definition of sdk\_def.h

#### Instruction

Get hardware capabilities.

#### Return value

Returns 0 success, other failures.

### sdks\_dev\_ json\_get\_hw\_cap

#### Define

int sdks\_dev\_ json\_get\_hw\_cap(unsigned int handle, char\*\* p\_result)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

p\_result：Output parameters, return the acquired hardware capabilities, expressed in json format. If the returned value is non-NULL, you need to call the sdks\_free\_result function to free the memory, otherwise it will cause a memory leak.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attributes | | | Types | Optional / required | description |
| Parameters | | | E | M |  |
|  | HardwareCapability | | E | M |  |
|  | ChannelNum | A | M | Number of channels (number of cameras) |
| AudioInNum | A | M | Number of audio inputs |
| AudioOutNum | A | M | Number of audio outputs |
| AlarmInNum | A | M | Number of alarm inputs |
| AlarmOutNum | A | M | Number of alarm outputs |
| RS485Num | A | M | RS485 serial port number |
| WireNetworkNum | A | M | Number of wired network cards |
| USBNum | A | M | USB number |
| SDNum | A | M | Number of SD cards |
| HardDiskNum | A | M | Number of hard disks |
| WiFiEnable | A | M | Whether to support wifi |
| FaceDetectionEnable | A | M | Whether to support face recognition |
| POEEnable | A | M | Whether to support poe |
| IREnable | A | M | Whether to support infrared |
| PIREnable | A | M | Whether to support PIR |
| BNCEnable | A | M | Whether to support analog channels |
| InternalPTZEnable | A | M | Whether to support built-in pan/tilt |
| AIDetectType | A | M | AI device type, AI multi-target, face, license plate; use bit representation, AI\_TYPE\_BIT |

p\_result the json format is：

{

"ChannelNum": 1,

"SoundChannelType": 0,

"AudioInNum": 0,

"AudioOutNum": 0,

"AlarmOutNum": 0,

"AlarmInNum": 0,

"RS485Num": 0,

"RS232Num": 0,

"WireNetworkNum": 1,

"USBNum": 0,

"SDNum": 0,

"HardDiskNum": 0,

"WiFiEnable": 0,

"POEEnable": 0,

"IREnable": 0,

"PIREnable": 0,

"BNCEnable": 0,

"InternalPTZEnable": 0,

"VirtualPTZType": 0,

"FaceDetectionEnable": 1,

"AIDetectType": 1

}

**Description**

#### Get hardware capabilities.

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_get\_video\_cap

#### Define

int sdks\_dev\_get\_video\_cap(unsigned int handle,int chn,char \*\*p\_result)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number, IPC fill-1;

p\_result：Output parameters, return the acquired video capabilities, expressed in json format. If the returned value is non-NULL, you need to call the sdks\_free\_result function to free the memory, otherwise it will cause a memory leak.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | | | | | | | | | Types | Optional / required | description |
| Parameters | | | | | | | | | | | | E | M |  |
|  | VideoAbility | | | | | | | | | | | E |  |  |
|  | VideoAbility ParamList | | | | | | | | | | E | M |  |
|  | VideoAbilityParam | | | | | | | | | E | M |  |
|  | ChannelId | | | | | | | | A | O | Channel number |
| ROIAbility | | | | | | | | E | M | ROI capability |
|  | Support | | | | | | | A | M | Whether to support ROI |
| MaxNum | | | | | | | A | M | Maximum number |
| MaxScale | | | | | | | A | M |  |
| MaxLevel | | | | | | | A | M | Maximum level |
| ROILevelQPList | | | | | | | E | M |  |
|  | ROILevelQP | | | | | | E | M |  |
|  | Level | | | | | A | M | ROI level |
| QP | | | | | A | M | Corresponding QP value |
| BaseStreamList | | | | | | | | E | M |  |
|  | BaseStream | | | | | | | E | M |  |
|  | StreamID | | | | | | A | M | Stream ID |
| VideoEncoderParamList | | | | | | E | M |  |
|  | | VideoEncoderParam | | | | E | M |  |
|  | VideoEncoderType | | | A | M | Video encoding type |
| EncodeTypeName | | | A | M | Video encoding name |
| EncodeLevelList | | | E | M |  |
| VideoEncoderParamItem | | | E | O |  |
|  | VideoEncoderParamItem | | E | O |  |
|  | ResolutionId | A | M | Source ID (may not be filled) |
| Name | A | M | Resolution name |
| Width | A | M | Video width |
| Hight | A | M | High video |
| MaxFrameRate | A | M | Maximum frame rate |
| MinFrameRate | A | M | Minimum frame rate |
| BitRateTypeList | A | M | Rate type  1: fixed rate 2: variable rate |
| MaxBitRate | A | M | Maximum code rate |
| MinBitRate | A | M | Latest code rate |
| MaxQuality | A | M | Maximum video quality |
| MinQuality | A | M | Minimum video quality |
| MaxIFrameInterval | A | M | Maximum frame interval |
| MinIFrameInterval | A | M | Minimum frame interval |
| SVCStreamAbilityList | | | | | | | | E | M |  |
|  | SVCStreamAbility | | | | | | | E | O |  |
|  | StreamId | | | | | | A | M | Companion flow ID |
| SourceStreamId | | | | | | A | M | Source stream ID |
| Divisor | | | | | | A | M | P frame ratio, eg: ", 5": support / 3, 1 / 5 elementary stream frame rate |

p\_result the json format is：

{

"video\_cap\_list": [{

"channel\_id": 1,

"roi\_cap": {

"support\_enable": 1,

"max\_num": 8,

"max\_scale": 50,

"max\_level": 5,

"roi\_level\_qp\_list": [{

"level": 1,

"qp": -7

}, {

"level": 2,

"qp": -9

}, {

"level": 3,

"qp": -10

}, {

"level": 4,

"qp": -12

}, {

"level": 5,

"qp": -14

}]

},

"base\_stream\_list": [{

"stream\_id": 1,

"encoder\_param": [{

"en\_type": 8,

"en\_name": "H265",

"encode\_level\_list": [{

"encoder\_level": 2

}],

"base\_cap": [{

"size": "1920x1080",

"width": 1920,

"height": 1080,

"max\_fps": 25,

"min\_fps": 1,

"max\_bps": 12000,

"min\_bps": 500,

"max\_qul": 9,

"min\_qul": 1,

"max\_inval": 75,

"min\_inval": 1

}, {

"size": "1280x720",

"width": 1280,

"height": 720,

"max\_fps": 25,

"min\_fps": 1,

"max\_bps": 8000,

"min\_bps": 200,

"max\_qul": 9,

"min\_qul": 1,

"max\_inval": 75,

"min\_inval": 1

}]

}, {

"en\_type": 1,

"en\_name": "H264",

"encode\_level\_list": [{

"encoder\_level": 1

}, {

"encoder\_level": 1

}, {

"encoder\_level": 1

}],

"base\_cap": [{

"size": "1920x1080",

"width": 1920,

"height": 1080,

"max\_fps": 25,

"min\_fps": 1,

"max\_bps": 12000,

"min\_bps": 500,

"max\_qul": 9,

"min\_qul": 1,

"max\_inval": 75,

"min\_inval": 1

}, {

"size": "1280x720",

"width": 1280,

"height": 720,

"max\_fps": 25,

"min\_fps": 1,

"max\_bps": 8000,

"min\_bps": 200,

"max\_qul": 9,

"min\_qul": 1,

"max\_inval": 75,

"min\_inval": 1

}]

}, {

"en\_type": 2,

"en\_name": "MJPEG",

"encode\_level\_list": [{

"encoder\_level": 0

}],

"base\_cap": [{

"size": "1920x1080",

"width": 1920,

"height": 1080,

"max\_fps": 12,

"min\_fps": 1,

"max\_bps": 12000,

"min\_bps": 8000,

"max\_qul": 9,

"min\_qul": 1,

"max\_inval": 0,

"min\_inval": 0

}, {

"size": "1280x720",

"width": 1280,

"height": 720,

"max\_fps": 12,

"min\_fps": 1,

"max\_bps": 9000,

"min\_bps": 5000,

"max\_qul": 9,

"min\_qul": 1,

"max\_inval": 0,

"min\_inval": 0

}]

}]

}, {

"stream\_id": 2,

"encoder\_param": [{

"en\_type": 8,

"en\_name": "H265",

"encode\_level\_list": [{

"encoder\_level": 2

}],

"base\_cap": [{

"size": "D1",

"width": 704,

"height": 576,

"max\_fps": 25,

"min\_fps": 1,

"max\_bps": 6000,

"min\_bps": 100,

"max\_qul": 9,

"min\_qul": 1,

"max\_inval": 75,

"min\_inval": 1

}, {

"size": "VGA",

"width": 640,

"height": 480,

"max\_fps": 25,

"min\_fps": 1,

"max\_bps": 3000,

"min\_bps": 100,

"max\_qul": 9,

"min\_qul": 1,

"max\_inval": 75,

"min\_inval": 1

}, {

"size": "640x360",

"width": 640,

"height": 360,

"max\_fps": 25,

"min\_fps": 1,

"max\_bps": 1500,

"min\_bps": 100,

"max\_qul": 9,

"min\_qul": 1,

"max\_inval": 75,

"min\_inval": 1

}, {

"size": "CIF",

"width": 352,

"height": 288,

"max\_fps": 25,

"min\_fps": 1,

"max\_bps": 1500,

"min\_bps": 100,

"max\_qul": 9,

"min\_qul": 1,

"max\_inval": 75,

"min\_inval": 1

}, {

"size": "QVGA",

"width": 320,

"height": 240,

"max\_fps": 25,

"min\_fps": 1,

"max\_bps": 1500,

"min\_bps": 100,

"max\_qul": 9,

"min\_qul": 1,

"max\_inval": 75,

"min\_inval": 1

}]

}, {

"en\_type": 1,

"en\_name": "H264",

"encode\_level\_list": [{

"encoder\_level": 1

}, {

"encoder\_level": 1

}, {

"encoder\_level": 1

}],

"base\_cap": [{

"size": "D1",

"width": 704,

"height": 576,

"max\_fps": 25,

"min\_fps": 1,

"max\_bps": 6000,

"min\_bps": 100,

"max\_qul": 9,

"min\_qul": 1,

"max\_inval": 75,

"min\_inval": 1

}, {

"size": "VGA",

"width": 640,

"height": 480,

"max\_fps": 25,

"min\_fps": 1,

"max\_bps": 3000,

"min\_bps": 100,

"max\_qul": 9,

"min\_qul": 1,

"max\_inval": 75,

"min\_inval": 1

}, {

"size": "640x360",

"width": 640,

"height": 360,

"max\_fps": 25,

"min\_fps": 1,

"max\_bps": 1500,

"min\_bps": 100,

"max\_qul": 9,

"min\_qul": 1,

"max\_inval": 75,

"min\_inval": 1

}, {

"size": "CIF",

"width": 352,

"height": 288,

"max\_fps": 25,

"min\_fps": 1,

"max\_bps": 1500,

"min\_bps": 100,

"max\_qul": 9,

"min\_qul": 1,

"max\_inval": 75,

"min\_inval": 1

}, {

"size": "QVGA",

"width": 320,

"height": 240,

"max\_fps": 25,

"min\_fps": 1,

"max\_bps": 1500,

"min\_bps": 100,

"max\_qul": 9,

"min\_qul": 1,

"max\_inval": 75,

"min\_inval": 1

}]

}, {

"en\_type": 2,

"en\_name": "MJPEG",

"encode\_level\_list": [{

"encoder\_level": 0

}],

"base\_cap": [{

"size": "D1",

"width": 704,

"height": 576,

"max\_fps": 12,

"min\_fps": 1,

"max\_bps": 6000,

"min\_bps": 100,

"max\_qul": 9,

"min\_qul": 1,

"max\_inval": 0,

"min\_inval": 0

}, {

"size": "VGA",

"width": 640,

"height": 480,

"max\_fps": 12,

"min\_fps": 1,

"max\_bps": 3000,

"min\_bps": 100,

"max\_qul": 9,

"min\_qul": 1,

"max\_inval": 0,

"min\_inval": 0

}, {

"size": "640x360",

"width": 640,

"height": 360,

"max\_fps": 12,

"min\_fps": 1,

"max\_bps": 1500,

"min\_bps": 100,

"max\_qul": 9,

"min\_qul": 1,

"max\_inval": 0,

"min\_inval": 0

}, {

"size": "CIF",

"width": 352,

"height": 288,

"max\_fps": 12,

"min\_fps": 1,

"max\_bps": 1500,

"min\_bps": 100,

"max\_qul": 9,

"min\_qul": 1,

"max\_inval": 0,

"min\_inval": 0

}, {

"size": "QVGA",

"width": 320,

"height": 240,

"max\_fps": 12,

"min\_fps": 1,

"max\_bps": 1500,

"min\_bps": 100,

"max\_qul": 9,

"min\_qul": 1,

"max\_inval": 0,

"min\_inval": 0

}]

}]

}, {

"stream\_id": 3,

"encoder\_param": [{

"en\_type": 8,

"en\_name": "H265",

"encode\_level\_list": [{

"encoder\_level": 2

}],

"base\_cap": [{

"size": "VGA",

"width": 640,

"height": 480,

"max\_fps": 25,

"min\_fps": 1,

"max\_bps": 3000,

"min\_bps": 100,

"max\_qul": 9,

"min\_qul": 1,

"max\_inval": 75,

"min\_inval": 1

}, {

"size": "CIF",

"width": 352,

"height": 288,

"max\_fps": 25,

"min\_fps": 1,

"max\_bps": 1500,

"min\_bps": 100,

"max\_qul": 9,

"min\_qul": 1,

"max\_inval": 75,

"min\_inval": 1

}, {

"size": "QVGA",

"width": 320,

"height": 240,

"max\_fps": 25,

"min\_fps": 1,

"max\_bps": 1500,

"min\_bps": 100,

"max\_qul": 9,

"min\_qul": 1,

"max\_inval": 75,

"min\_inval": 1

}]

}, {

"en\_type": 1,

"en\_name": "H264",

"encode\_level\_list": [{

"encoder\_level": 1

}, {

"encoder\_level": 1

}, {

"encoder\_level": 1

}],

"base\_cap": [{

"size": "VGA",

"width": 640,

"height": 480,

"max\_fps": 25,

"min\_fps": 1,

"max\_bps": 3000,

"min\_bps": 100,

"max\_qul": 9,

"min\_qul": 1,

"max\_inval": 75,

"min\_inval": 1

}, {

"size": "CIF",

"width": 352,

"height": 288,

"max\_fps": 25,

"min\_fps": 1,

"max\_bps": 1500,

"min\_bps": 100,

"max\_qul": 9,

"min\_qul": 1,

"max\_inval": 75,

"min\_inval": 1

}, {

"size": "QVGA",

"width": 320,

"height": 240,

"max\_fps": 25,

"min\_fps": 1,

"max\_bps": 1500,

"min\_bps": 100,

"max\_qul": 9,

"min\_qul": 1,

"max\_inval": 75,

"min\_inval": 1

}]

}, {

"en\_type": 2,

"en\_name": "MJPEG",

"encode\_level\_list": [{

"encoder\_level": 0

}],

"base\_cap": [{

"size": "VGA",

"width": 640,

"height": 480,

"max\_fps": 12,

"min\_fps": 1,

"max\_bps": 3000,

"min\_bps": 100,

"max\_qul": 9,

"min\_qul": 1,

"max\_inval": 0,

"min\_inval": 0

}, {

"size": "CIF",

"width": 352,

"height": 288,

"max\_fps": 12,

"min\_fps": 1,

"max\_bps": 1500,

"min\_bps": 100,

"max\_qul": 9,

"min\_qul": 1,

"max\_inval": 0,

"min\_inval": 0

}, {

"size": "QVGA",

"width": 320,

"height": 240,

"max\_fps": 12,

"min\_fps": 1,

"max\_bps": 1500,

"min\_bps": 100,

"max\_qul": 9,

"min\_qul": 1,

"max\_inval": 0,

"min\_inval": 0

}]

}]

}],

"svc\_stream\_cap\_list": [{

"stream\_id": 4,

"src\_stream\_id": "1,2",

"divisor": "1,2,3,4"

}]

}]

}

**Description**

Get video capabilities.

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_get\_sw\_cap

#### Define

int sdks\_dev\_get\_sw\_cap(unsigned int handle, dev\_sw\_cap\_t\* p\_sw\_cap);

#### parameter

handle：Return value of sdks\_dev\_conn interface；

p\_sw\_cap：Fill the software parameter to the caller, refer to the definition of sdk\_def.h

#### Instruction

Get software capabilities.

#### Return value

Returns 0 success, other failures.

### sdks\_dev\_ json\_get\_sw\_cap

#### Define

int sdks\_dev\_ json\_get\_sw\_cap(unsigned int handle, char\*\* p\_result)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

p\_result：Output parameters, return the acquired software capabilities, expressed in json format. If the returned value is non-NULL, you need to call the sdks\_free\_result function to free the memory, otherwise it will cause a memory leak.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | | Types | Optional / required | description |
| Parameters | | | | | E | M |  |
|  | SoftwareCapability | | | | E | M |  |
|  | LoginUserMaxNum | | | A | M | The maximum number of logged in users |
| PreviewMaxNum | | | A | M | The maximum number of video previews |
| PlaybackMaxNum | | | A | M | The maximum number of video playback |
| SupportFunctionList | | |  | M |  |
|  | SupportFunction | | E | M |  |
|  |  | SupportFunction | A | M | For a list of supported functions, see the function ID definition table. |

p\_result the json format is：

{

"LoginUserMaxNum": 50,

"PreviewMaxNum": 10,

"PlaybackMaxNum": 10,

"SupportFunctionList": [{

"SupportFunction": 1

}, {

"SupportFunction": 2

}, {

"SupportFunction": 3

}]

}

**Description**

Get software capabilities.

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_get\_audio\_cap

#### Define

int sdks\_dev\_get\_audio\_cap(unsigned int handle, dev\_audio\_cap\_t\* p\_audio\_cap);

#### parameter

handle：Return value of sdks\_conn interface；

p\_audio\_cap：Fill the audio parameter to the caller, refer to the definition of sdk\_def.h

#### Instruction

Get audio capabilities parameter。

#### Return value

Returns 0 success, other failures.

### sdks\_dev\_ json\_get\_audio\_cap

#### Define

int sdks\_dev\_ json\_get\_audio\_cap(unsigned int handle, char\*\* p\_result)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

p\_result：Output parameters, return the obtained audio capability parameters, expressed in json format. If the returned value is non-NULL, you need to call the sdks\_free\_result function to free the memory, otherwise it will cause a memory leak.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | | | Types | Optional / required | description |
| Parameters | | | | | | E | M |  |
|  | AudioExAbility | | | | | E | M |  |
|  | InterPhoneFlag | | | | A | M | Audio intercom effective mark |
| AudioInFlag | | | | A | M | Audio input valid flag |
| AudioOutFlag | | | | A | M | Audio output valid flag |
| AudioInTypeList | | | | E | M | Audio input type list |
|  | | AudioInType | | E | O |  |
|  | AudioInTypeId | A | M | The audio input type Id is detailed in Appendix[1.14 Coding type](#_1.14_Coding_type) |
| AudioInTypeName | A | M | Audio input type name |
| AudioInVolMax | A | M | Maximum input volume |
| AudioInVolMin | A | M | Minimum input volume |
| AudioOutTypeList | | | | E | M |  |
|  | AudioOutType | | | E | O |  |
|  | | AudioOutTypeId | A | M | The audio input type Id is detailed in Appendix [1.14 Coding type](#_1.14_Coding_type) |
| AudioOutTypeName | A | M | Audio output type name |
| AudioOutVolMax | A | M | Maximum output volume |
| AudioOutVolMin | A | M | Minimum output volume |
| AudioEncodeTypeList | | | | E | M |  |
|  | AudioEncodeType | | | E | O |  |
|  | | AudioEncodeTypeId | A | M | Audio encoding type Id |
| AudioEncodeTypeMode | A | M | Audio coding mode |
| AudioEncodeTypeName | A | M | Audio encoding type name |
| AudioDecodeTypeList | | | | E | M |  |
|  | AudioDecodeType | | | E | O |  |
|  | | AudioDecodeTypeId | A | M | Audio decoding type Id |
| AudioDecodeTypeMode | A | M | Audio decoding mode |
| AudioDecodeTypeName | A | M | Audio decoding type name |

p\_result the json format is：

{

"InterPhoneFlag": 0,

"AudioInFlag": 0,

"AudioOutFlag": 0,

"AudioInTypeList": [],

"AudioOutTypeList": [],

"AudioEncodeTypeList": [{

"AudioEncodeTypeId": 102,

"AudioEncodeTypeMode": 0,

"AudioEncodeTypeName": "G711\_ALAW"

}, {

"AudioEncodeTypeId": 103,

"AudioEncodeTypeMode": 0,

"AudioEncodeTypeName": "G711\_ULAW"

}, {

"AudioEncodeTypeId": 108,

"AudioEncodeTypeMode": 0,

"AudioEncodeTypeName": "RAW\_PCM"

}, {

"AudioEncodeTypeId": 109,

"AudioEncodeTypeMode": 0,

"AudioEncodeTypeName": "NONE"

}],

"AudioDecodeTypeList": [{

"AudioDecodeTypeId": 102,

"AudioDecodeTypeMode": 0,

"AudioDecodeTypeName": "G711\_ALAW"

}, {

"AudioDecodeTypeId": 103,

"AudioDecodeTypeMode": 0,

"AudioDecodeTypeName": "G711\_ULAW"

}, {

"AudioDecodeTypeId": 108,

"AudioDecodeTypeMode": 0,

"AudioDecodeTypeName": "RAW\_PCM"

}]

}

**Description**

Get audio capability parameters.

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_get\_language\_cap

#### Define

int sdks\_dev\_get\_language\_cap(unsigned int handle, int chn, char \*\*p\_result);

#### parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number.

p\_result: Output parameters, fill the language ability parameters to the caller, return in json format, if the returned value is not NULL, you need to call sdks\_free\_result function to release the memory, otherwise it will lead to memory leak.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | | Types | Optional / required | Attributes |
|  |  | LanguageTypeList | | | E | M |  |
|  |  | LanguageID | A | M | Language ID, see Appendix [1.13 Language type](#_1.13_Language_type) |

p\_result the json format is：

{

"language\_type\_list": [{

"language\_id": 1

}, {

"language\_id": 2

}, {

"language\_id": 3

}, …

}]

}

**Description**

#### Get device language capability request

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_get\_nw\_cap

#### Define

int sdks\_dev\_get\_nw\_cap(unsigned int handle, int chn, char\*\* p\_result);

#### parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number, IPC fill-1;

p\_result: Output parameters, fill the network capability parameters to the caller, return in json format, if the returned value is not NULL, you need to call sdks\_free\_result function to release the memory, otherwise it will lead to memory leak.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attributes | | | | Description |
| Parameters | | | |  | |
|  | net\_card\_num | | | Number of network cards |
| net\_card\_cap\_list | | |  |
|  | net\_card\_cap | |  |
|  | net\_card\_id | NIC number |
| net\_card\_type | NIC type |
| net\_card\_name | Network card name |
| description | Network card description |
| support\_alarm\_enable | Whether to support broken network alarm |
| ip\_config\_enable | Whether to support IP configurable |
| dhcp\_server\_enable | Whether to support dhcp server |
| dhcp\_client\_enable | Whether to support dhcp client |
| dns\_config\_enable | Whether to support DNS configuration |
| poe\_config\_enable | Whether to support Poe configuration |

p\_result the json format is：

{

"net\_card\_num": 1,

"net\_card\_cap\_list": [{

"net\_card\_id": 1,

"net\_card\_type": 0,

"net\_card\_name": "eth0",

"description": "NetworkCard",

"support\_alarm\_enable": 1,

"ip\_config\_enable": 1,

"dhcp\_server\_enable": 0,

"dhcp\_client\_enable": 1,

"dns\_config\_enable": 1,

"poe\_config\_enable": 0

}]

}

**Description**

#### Obtain device network capability request

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_get\_time\_zone\_cap

#### Define

int sdks\_dev\_get\_time\_zone\_cap(unsigned int handle, int chn,int language\_id, char\*\* p\_result) ;

#### parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number.

language\_id : Language type,See Appendix 1.16.

p\_result：Output parameters, fill the time zone parameters to the caller, return in json format, if the returned value is non-NULL, you need to call sdks\_free\_result function to release the memory, otherwise it will lead to memory leak。

|  |  |  |  |
| --- | --- | --- | --- |
| Key field | Explain | Is it necessary | Remarks |
| TimeZoneID | Time zone ID | Y | Only when ID is 1, the content can be displayed normally. |
| TimeZoneName | Time zone name | Y |  |
| DSTTimeZoneAbbreviation | Summer time zone abbreviation | Y |  |
| STDTimeZoneAbbreviation | Standard time zone abbreviation |  |  |

p\_result the json format is：

{

"TimeZoneAbility": [{

"TimeZoneID": 0,

"TimeZoneName": "(GMT-12:00) International Date Line West",

"DSTTimeZoneAbbreviation": "",

"STDTimeZoneAbbreviation": "GMT12"

}, {

"TimeZoneID": 1,

"TimeZoneName": "(GMT-11:00) Midway Island, Samoa",

"DSTTimeZoneAbbreviation": "",

"STDTimeZoneAbbreviation": "SST11,NUT11"

}, {

"TimeZoneID": 2,

"TimeZoneName": "(GMT-10:00) Hawaii",

"DSTTimeZoneAbbreviation": "HADT",

"STDTimeZoneAbbreviation": "HAST10,TKT10,HST10,CKT10,TAHT10"

} ,......

}

**Description**

Get time zone capability requests, different languages correspond to different time zone capability lists.

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_get\_osd\_cap

#### Define

int sdks\_dev\_get\_osd\_cap (unsigned int handle, int chn, char\*\* p\_result);

#### parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number, IPC fill-1;

p\_result：Output parameters, fill the OSD parameters to the caller, return in json format, if the returned value is not NULL, you need to call sdks\_free\_result function to release the memory, otherwise it will lead to memory leak.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attributes | | | | | Description |
| Parameters | | | | |  |
| p\_result | | | | |  |
|  | osd\_max\_num | | | | Maximum number of OSDs |
| canvas\_max\_num | | | | Maximum number of areas |
| osd\_max\_width | | | | Maximum width |
| osd\_max\_height | | | | maximum height |
| osd\_font\_cap | | | | OSD font ability |
|  | font\_type\_list | | |  |
| font\_color\_list | | | Font color list |
|  | font\_color | | Font color, -1 means support 0~255 |
|  | R | red |
| G | green |
| B | blue |
| font\_alpha\_list | | | Font transparency list |
|  | font\_alpha | | Font transparency  0: Fully transparent  16: Translucent  32: Sub-transparent  48: opaque |
| inverse\_color\_list | | | Font reverse color list |
| font\_size\_list | | | Font size list |
|  | font\_size | | Font size, the larger the value, the larger the size |
| osd\_type\_list | | | OSD type list |
|  | osd\_type | | OSD type  1: device name  2: channel ID  3: channel name  4: Time  5: Custom OSD  6: PTZ coordinates  7: PTZ operation  8: PTZ temperature |
|  | id | Type ID |
| name | name |
| osd\_format\_type\_list | | | | OSD format type list |
|  | format\_type | | | OSD time format type  1:YYYY-MM-DD hh:mm:ss ww  2:hh:mm:ss YYYY-MM-DD ww  3:MM/DD/YYYY hh:mm:ss ww  4:hh:mm:ss MM/DD/YYYY ww |
|  | format\_id | | Format ID |
| description | | description |

p\_result the json format is：

{

"osd\_max\_num": 1,

"canvas\_max\_num": 8,

"osd\_max\_width": 100,

"osd\_max\_height": 100,

"osd\_font\_cap": {

"font\_type\_list": [],

"font\_color\_list": [{

"R": -1,

"G": -1,

"B": -1

}],

"font\_alpha\_list": [{

"font\_alpha": 0

}, {

"font\_alpha": 16

}, {

"font\_alpha": 32

}, {

"font\_alpha": 48

}],

"inverse\_color\_list": [],

"font\_size\_list": [{

"font\_size": "16"

}, {

"font\_size": "24"

}, {

"font\_size": "32"

}, {

"font\_size": "48"

}, {

"font\_size": "64"

}, {

"font\_size": "96"

}],

"osd\_type\_list": [{

"id": 1,

"name": "Device Name"

}, {

"id": 2,

"name": "Channel ID"

}, {

"id": 3,

"name": "Channel Name"

}, {

"id": 4,

"name": "Time"

}, {

"id": 5,

"name": "Custom OSD"

}]

},

"osd\_format\_type\_list": [{

"format\_id": 1,

"description": "YYYY-MM-DD hh:mm:ss ww"

}, {

"format\_id": 2,

"description": "hh:mm:ss YYYY-MM-DD ww"

}, {

"format\_id": 3,

"description": "MM/DD/YYYY hh:mm:ss ww"

}, {

"format\_id": 4,

"description": "hh:mm:ss MM/DD/YYYY ww"

}]

}

**Description**

#### Obtain OSD capability parameters

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_get\_ptz\_cap

#### Define

int sdks\_dev\_get\_ptz\_cap(unsigned int handle, int chn, char\*\* p\_result);

#### parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number, IPC fill-1;

p\_result：Output parameters, fill the ptz parameter to the caller, return in json format, if the returned value is not NULL, you need to call sdks\_free\_result function to release the memory, otherwise it will lead to memory leak.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attributes | | | | Description |
| Parameters | | | |  |
| p\_result | | | |  |
|  | ver\_max\_pos\_num | | |  |
| max\_mag\_para | | | Maximum optical multiple parameter |
|  | max\_opt\_mag | | Maximum virtual optical multiple |
| max\_real\_opt\_mag | | Maximum true optical multiple |
| speed\_para | | |  |
|  | pan\_step\_period | | Pan step cycle |
| tilt\_step\_period | | Tilt step cycle |
| pan\_max\_speed | | Pan max goto speed |
| tilt\_max\_speed | | Tilt max goto speed |
| min\_speed | | Minimum speed |
| max\_speed | | Maximum speed |
| enable | | it's usable or not |
| speed\_list | | Speed parameter list |
| trace\_cap | | |  |
|  | max\_second | | Maximum number of seconds of the track |
| max\_cmd\_num | | Maximum number of tracks |
| begin\_id | | Start point ID of the track |
| end\_id | | End point ID of the track |
| enable | | it's usable or not |
| tour\_cap | | |  |
|  | max\_preset\_num | | Maximum number of preset points |
| begin\_id | | Tour point ID |
| end\_id | | End point ID of the cruise |
| enable | | it's usable or not |
| scan\_cap | | |  |
|  | max\_scan\_num | | Maximum number of scans |
| begin\_id | | Start ID of the scan |
| end\_id | | End ID of the scan |
| enable | | it's usable or not |
| ptz\_speed\_table | | |  |
|  | speed\_type | | Customized speed type, the default speed algorithm is 0, and the corresponding custom speedometer is not 0. |
| tilt\_dis\_para | | |  |
|  | moto\_min\_angle | | Motor minimum inclination |
| moto\_min\_angle | | Motor maximum inclination |
| dis\_min\_angle | | Show minimum tilt angle |
| dis\_min\_angle | | Display maximum tilt angle |
| ver\_max\_pos\_para | | |  |
|  | max\_pos\_most | | Large vertical maximum angle |
| max\_pos\_more | | a value with a small vertical maximum angle |
| preset\_pos\_para | | | Preset capability parameter |
|  | id\_range\_from | | Horizontal starting point ID |
| id\_range\_to | | Horizontal end point ID |
| max\_preset\_num | | Maximum number of preset points |
| enable | | it's usable or not |
| guard\_pos\_cap | | | Watch position capability parameter |
|  | enable | | it's usable or not |
| infrared\_lamp\_para | | | Infrared lamp capability parameter |
|  | enable | | it's usable or not |
| due\_north\_cap | | | North ability parameter |
|  | enable | | it's usable or not |
| timer\_cap\_para | | | Timer capability parameter |
|  | enable | | it's usable or not |
| three\_dimension\_pos | | | 3D positioning capability parameter |
|  | enable | | it's usable or not |
| ptz\_protocol\_list | | | PTZ protocol table parameters |
|  | ptz\_protocol | |  |
|  | protocol\_id | ID number of the agreement |
| protocol\_name | Agreement name |
| brush\_para | | | Wiper capacity parameter |
|  | enable | | it's usable or not |
| light\_para | | | General light parameters |
|  | enable | | it's usable or not |

p\_result the json format is：

{

"ver\_max\_pos\_num": 252,

"max\_mag\_para": {

"max\_opt\_mag": 30000,

"max\_real\_opt\_mag": 25090

},

"speed\_para": {

"pan\_step\_period": 614400,

"tilt\_step\_period": 512000,

"pan\_max\_speed": 36000,

"tilt\_max\_speed": 36000,

"min\_speed": 0,

"max\_speed": 64,

"enable": 1,

"speed\_list": []

},

"trace\_cap": {

"max\_second": 300,

"max\_cmd\_num": 1000,

"begin\_id": 0,

"end\_id": 5,

"enable": 1

},

"tour\_cap": {

"max\_preset\_num": 32,

"begin\_id": 0,

"end\_id": 11,

"enable": 1

},

"scan\_cap": {

"max\_scan\_num": 12,

"begin\_id": 0,

"end\_id": 11,

"enable": 1

},

"ptz\_speed\_table": {

"speed\_type": 0

},

"tilt\_dis\_para": {

"moto\_min\_angle": 0,

"moto\_min\_angle": 93,

"dis\_min\_angle": -5,

"dis\_max\_angle": 90

},

"ver\_max\_pos\_para": {

"max\_pos\_most": 132267,

"max\_pos\_more": 115200

},

"preset\_pos\_para": {

"id\_range\_from": 1,

"id\_range\_to": 400,

"max\_preset\_num": 400,

"enable": 1

},

"guard\_pos\_cap": {

"enable": 1

},

"infrared\_lamp\_para": {

"enable": 0

},

"due\_north\_cap": {

"enable": 0

},

"timer\_cap\_para": {

"enable": 1

},

"three\_dimension\_pos": {

"enable": 0

},

"ptz\_protocol\_list": [],

"brush\_para": {

"enable": 0

},

"light\_para": {

"enable": 0

}

}

**Description**

#### Get the ptz capability parameter.

#### PTZ capability parameters can be obtained regardless of whether the device supports PTZ operation. If you want to know if the device supports PTZ operations, you can call sdks\_dev\_get\_ptz\_req(unsigned int handle, int chn, char\*\* p\_result) to get the interface for the PTZ request.

#### Return value

0 succeeded; other values failed.

## Basic equipment parameters

### sdks\_dev\_get\_general\_info

#### Define

int sdks\_dev\_get\_general\_info(unsigned int handle, dev\_general\_info\_t\* p\_gene\_info);

#### parameter

handle：Return value of sdks\_conn interface；

p\_gene\_info：Fill the device's basic parameter to the caller, referring to the definition of sdk\_def.h

#### Instruction

Return device basicparameter。

#### Return value

Returns 0 success, other failures.

### sdks\_dev\_ json\_get\_general\_info

#### Define

int sdks\_dev\_ json\_get\_general\_info(unsigned int handle, char\*\* p\_result)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

p\_result：Output parameters, return the obtained device basic parameters, expressed in json format. If the returned value is non-NULL, you need to call the sdks\_free\_result function to free the memory, otherwise it will cause a memory leak.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | Types | Optional / required | | | description |
| Parameters | | | Element | | M |  | |
|  | DeviceSummaryInfo | | Element | | M |  | |
|  | DeviceId | Attribute | | M | Device ID | |
| DeviceType | Attribute | | M | Device type: See device type definition for details[1.12 Equipment type](#_1.12_Equipment_type) | |
| DeviceName | Attribute | | M | Device name | |
| DeviceIP | Attribute | | M | Device IP | |
| DevicePort | Attribute | | M | Device control port | |
| MACAddr | Attribute | | M | MAC address | |
| ManufacturerName | Attribute | | M | Manufacturer | |
| ManufacturerId | Attribute | | M | Manufacturer number | |
| ProductModel | Attribute | | M | Product number | |
| SN | Attribute | | M | serial number | |
| SoftWareInfo | Attribute | | M | Software version | |
| HardWareInfo | Attribute | | M | hardware version | |

p\_result the json format is：

{

"DeviceId": "573055",

"DeviceType": 4,

"DeviceName": "66666",

"DeviceIP": "192.168.0.222",

"DevicePort": 30001,

"MACAddr": "00:1C:27:57:30:55",

"ManufacturerName": "IPCamera",

"ManufacturerId": "003",

"ProductModel": "IPS57/30CDR/ZSD12/13",

"SN": "573055",

"SoftWareInfo": "t3.5.0804.1004.3.0.32.0.2",

"HardWareInfo": "V060201\_3"

}

**Description**

Get the basic parameters of the device.

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_get\_dev\_name

#### Define

int sdks\_dev\_get\_dev\_name(unsigned int handle, dev\_name\_t\* p\_dev\_name);

#### parameter

handle：Return value of sdks\_conn interface；

p\_dev\_name：Fill the name of the device to the caller, refer to the definition of sdk\_def.h

#### Instruction

Get the name of the device

#### Return value

0 is successful, other fails.

### sdks\_dev\_ json\_get\_dev\_name

#### Define

int sdks\_dev\_ json\_get\_dev\_name(unsigned int handle, char\*\* p\_result)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

p\_result：Output parameters, return the obtained device name, expressed in json format. If the returned value is non-NULL, you need to call the sdks\_free\_result function to free the memory, otherwise it will cause a memory leak.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attributes | | | Types | Optional / required | description |
| Parameters | | | Element | M |  |
|  | DeviceName | | Element | M |  |
|  | Name | Attribute | M | Device name |

p\_result the json format is：

{

"Name": "66666"

}

**Description**

#### Get the name of the device.

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_set\_dev\_name

#### Define

int sdks\_dev\_set\_dev\_name(unsigned int handle, dev\_name\_t\* p\_dev\_name);

#### parameter

handle：Return value of sdks\_conn interface；

p\_dev\_name：Device name, refer to the definition of sdk\_def.h

#### Instruction

Set the device name.

#### Return value

0 is successful, other fails.。

### sdks\_dev\_ json\_set\_dev\_name

#### Define

int sdks\_dev\_ json\_set\_dev\_name(unsigned int handle, char\* p\_param)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

p\_param：Input parameters, expressed in json format.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attributes | | | Types | Optional / required | description |
| Parameters | | | Element | M |  |
|  | DeviceName | | Element | M |  |
|  | Name | Attribute | M | Device name |

p\_param the json format is：

{

"Name": "66666"

}

**Description**

#### Set the device name.

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_get\_dev\_time

#### Define

int sdks\_dev\_get\_dev\_time(unsigned int handle, dev\_time\_t\* p\_dev\_time);

#### parameter

handle：Return value of sdks\_conn interface；

p\_dev\_time：Device time, refer to the definition of sdk\_def.h

#### Instruction

Get device time

#### Return value

0 is successful, other fails.。

### sdks\_dev\_ json\_get\_dev\_time

#### Define

int sdks\_dev\_ json\_get\_dev\_time(unsigned int handle, char\*\* p\_result)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

p\_result：Output parameters, return the acquired device time, expressed in json format. If the returned value is non-NULL, you need to call the sdks\_free\_result function to free the memory, otherwise it will cause a memory leak.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attributes | | | Types | Optional / required | description |
| Parameters | | | E | M |  |
|  | DeviceTime | | E | M | Equipment time |
|  | Year | A | M | Year |
| Month | A | M | Month |
| Day | A | M | Day |
| Hour | A | M | Hour |
| Minute | A | M | Minute |
| Second | A | M | Second |

p\_result the json format is：

{

"Year": 2019,

"Month": 9,

"Day": 30,

"Hour": 16,

"Minute": 40,

"Second": 33

}

**Description**

#### Get device time.

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_set\_dev\_time

#### Define

int sdks\_dev\_set\_dev\_time(unsigned int handle, dev\_time\_t\* p\_dev\_time);

#### parameter

handle：Return value of sdks\_conn interface；

p\_dev\_time：Device time, refer to the definition of sdk\_def.h。

#### Instruction

Set the device time.

#### Return value

0 is successful, other fails.

### sdks\_dev\_ json\_set\_dev\_time

#### Define

int sdks\_dev\_ json\_set\_dev\_time(unsigned int handle, char\* p\_param)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

p\_param：Input parameters, expressed in json format.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attributes | | | Types | Optional / required | description |
| Parameters | | | E | M |  |
|  | DeviceTime | | E | M | Equipment time |
|  | Year | A | M | Year |
| Month | A | M | Month |
| Day | A | M | Day |
| Hour | A | M | Hour |
| Minute | A | M | Minute |
| Second | A | M | Second |

p\_param the json format is：

{

"Year": 2019,

"Month": 9,

"Day": 30,

"Hour": 16,

"Minute": 40,

"Second": 33

}

**Description**

#### Set the device time.

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_get\_dev\_ntp

#### Define

int sdks\_dev\_get\_dev\_ntp(unsigned int handle, ntp\_param\_t\* p\_ntp\_param);

#### parameter

handle：Return value of sdks\_conn interface；

p\_ntp\_param：nptparameter，Refer to the definition of sdk\_def.h。

#### Instruction

Obtain ntpparameter。

#### Return value

0 is successful, other fails.。

### sdks\_dev\_ json\_get\_dev\_ntp

#### Define

int sdks\_dev\_ json\_get\_dev\_ntp(unsigned int handle, char\*\* p\_result);

#### parameter

handle：Return value of sdks\_dev\_conn interface.

p\_result：Output parameters, return the obtained ntp parameters, expressed in json format. If the returned value is non-NULL, you need to call the sdks\_free\_result function to free the memory, otherwise it will cause a memory leak.。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attributes | | | Types | Optional / required | description |
| Parameters | | | E | M |  |
|  | NTPParam | | E | M |  |
|  | enable | A | M | NTP service enablement flag |
| ip | A | M | NTP server IP |
| port | A | M | NTP server port |
| time | A | M | NTP calibration time |
| proto\_ver | A | M | IP protocol type |

p\_result the json format is：

{

"ip": "192.168.0.121",

"port": 0,

"enable": 0,

"proto\_ver": 0,

"time": 0

}

**Description**

#### Get the ntp parameter.

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_set\_dev\_ntp

#### Define

int sdks\_dev\_set\_dev\_ntp(unsigned int handle, ntp\_param\_t\* p\_ntp\_param);

#### parameter

handle：Return value of sdks\_conn interface；

p\_ntp\_param：Nptparameter, refer to the definition of sdk\_def.h。

#### Instruction

Set up ntpparameter

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_ json\_set\_dev\_ntp

#### Define

int sdks\_dev\_ json\_set\_dev\_ntp(unsigned int handle, char\* p\_param);

#### parameter

handle：Return value of sdks\_dev\_conn interface.

p\_param：Input parameters, expressed in json format.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attributes | | | Types | Optional / required | description |
| Parameters | | | E | M |  |
|  | NTPParam | | E | M |  |
|  | enable | A | M | NTP service enablement flag |
| ip | A | M | NTP server IP |
| port | A | M | NTP server port |
| time | A | M | NTP calibration time |
| proto\_ver | A | M | IP protocol type |

p\_param the json format is：

{

"ip": "192.168.0.121",

"port": 0,

"enable": 0,

"proto\_ver": 0,

"time": 0

}

**Description**

#### Set the ntp parameter

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_get\_dev\_id

#### Define

int sdks\_dev\_get\_dev\_id(unsigned int handle, int chn,char\*\* p\_result);

#### parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number, IPC fill-1.

p\_result：Output parameters, return device ID information. If the returned value is non-NULL, you need to call the sdks\_free\_result function to free the memory, otherwise it will cause a memory leak.

|  |  |  |
| --- | --- | --- |
| Attributes | | Description |
| Parameters | |  |
| p\_result | |  |
|  | dev\_id | dev\_id |

p\_result the json format is：

{

"dev\_id": "789asd"

}

**Description**

Get device ID parameter

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_set\_dev\_id

#### Define

int sdks\_dev\_set\_dev\_id(unsigned int handle, int chn,char\* p\_param)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number, IPC fill-1.

p\_param：Input parameters, expressed in json format.

|  |  |  |
| --- | --- | --- |
| Attributes | | Description |
| Parameters | |  |
| p\_result | |  |
|  | dev\_id | dev\_id |

p\_param the json format is：

{

"dev\_id": "789asd"

}

**Description**

#### Set device ID parameters.

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_get\_dev\_port

#### Define

int sdks\_dev\_get\_dev\_port(unsigned int handle, dev\_port\_t\* p\_dev\_port);

#### parameter

handle：Return value of sdks\_dev\_conn interface；

p\_dev\_port：Device port, refer to the definition of sdk\_def.h.

#### Instruction

Get device port information

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_ json\_get\_dev\_port

#### Define

int sdks\_dev\_ json\_get\_dev\_port(unsigned int handle, char\*\* p\_result);

#### parameter

handle：Return value of sdks\_dev\_conn interface.

p\_result：Output parameters, return the obtained device port parameters, expressed in json format. If the returned value is non-NULL, you need to call the sdks\_free\_result function to free the memory, otherwise it will cause a memory leak.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attributes | | | Types | Optional / required | description |
| Parameters | | | E | M |  |
|  | DevicePort | | E | M |  |
|  | ControlPort | A | M | Device network control port |
| TCPAVPort | A | M | TCP audio and video port |
| RTSPPort | A | M | RTSP port |
| RTPPort | A | M | RTP port |
| RTCPPort | A | M | RTCP port |
| RegisterPort | A | M | Device registration port |
| HttpPort | A | M | HTTP port |
| ReservePort1 | A | M | Flash server port 1 |
| ReservePort2 | A | M | Flash server port 2 |
| ReservePort3 | A | M | Flash server port 3 |
| ReservePort4 | A | M | Flash server port 4 |
| SslControlPort | A | O | Ssl port |

p\_result the json format is：

{

"ControlPort": 30001,

"TCPAVPort": 30001,

"RTSPPort": 554,

"RTPPort": 555,

"RTCPPort": 556,

"RegisterPort": 556,

"HttpPort": 80,

"ReservePort1": 8080,

"ReservePort2": 8081,

"ReservePort3": 8082,

"ReservePort4": 8083,

"SslControlPort": 0

}

**Description**

Get device port information.

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_set\_dev\_port

#### Define

int sdks\_dev\_set\_dev\_port(unsigned int handle, dev\_port\_t\* p\_dev\_port);

#### parameter

handle：Return value of sdks\_dev\_conn interface；

p\_dev\_port：Device port, refer to the definition of sdk\_def.h.

#### Instruction

Set the device port.

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_ json\_set\_dev\_port

#### Define

int sdks\_dev\_ json\_set\_dev\_port(unsigned int handle, char\* p\_param)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

p\_param：Input parameters, expressed in json format.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attributes | | | Types | Optional / required | description |
| Parameters | | | E | M |  |
|  | DevicePort | | E | M |  |
|  | ControlPort | A | M | Device network control port |
| TCPAVPort | A | M | TCP audio and video port |
| RTSPPort | A | M | RTSP port |
| RTPPort | A | M | RTP port |
| RTCPPort | A | M | RTCP port |
| RegisterPort | A | M | Device registration port |
| HttpPort | A | M | HTTP port |
| ReservePort1 | A | M | Flash server port 1 |
| ReservePort2 | A | M | Flash server port 2 |
| ReservePort3 | A | M | Flash server port 3 |
| ReservePort4 | A | M | Flash server port 4 |
| SslControlPort | A | O | Ssl port |

p\_param the json format is：

{

"ControlPort": 30001,

"TCPAVPort": 30001,

"RTSPPort": 554,

"RTPPort": 555,

"RTCPPort": 556,

"RegisterPort": 556,

"HttpPort": 80,

"ReservePort1": 8080,

"ReservePort2": 8081,

"ReservePort3": 8082,

"ReservePort4": 8083,

"SslControlPort": 0

}

**Description**

#### Set the device port.

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_get\_dev\_language

#### Define

int sdks\_dev\_get\_dev\_language(unsigned int handle, int chn, char\*\* p\_result);

#### parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number.

p\_result：Output parameters, return device language information. If the returned value is non-NULL, you need to call the sdks\_free\_result function to free the memory, otherwise it will cause a memory leak.

|  |  |  |
| --- | --- | --- |
| Attributes | | Description |
| Parameters | |  |
| p\_result | |  |
|  | language\_id | Language ID 1: English 2: Chinese,see [1.13 Language type](#_1.13_Language_type) |

p\_result the json format is：

{

"language\_id": 1

}

**Description**

#### Get device language information.

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_set\_dev\_language

#### Define

int sdks\_dev\_set\_dev\_language(unsigned int handle, int chn,int language\_id)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number.

language\_id: Language ID number.

**Description**

Set the device language.

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_get\_security\_para

#### Define

int sdks\_dev\_get\_security\_para(unsigned int handle, int chn,char\*\* p\_result);

#### parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number.

p\_result: Output parameters, fill the device security parameters to the caller, return in json format, if the returned value is not NULL, you need to call sdks\_free\_result function to release the memory, otherwise it will lead to memory leak.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attributes | | | Types | Optional / required | description |
| Parameters | | | E | M |  |
|  | DeviceSecurityParam | | E | M |  |
|  | WebMode | A | M | 1 ：http 2：https，http 3：https |
| CommunicationEncryptFlag | A | M |  |

p\_result the json format is：

{

"web\_mode": 1,

"communication\_encrypt\_enable": 0

}

**Description**

#### Returns the security parameters of the device.

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_set\_security\_para

#### Define

int sdks\_dev\_set\_security\_para(unsigned int handle, int web\_mode, unsigned char encrypt\_enable);

#### parameter

handle：Return value of sdks\_dev\_conn interface.

web\_mode: Web page mode,1 http；2 https和http；3 https

encrypt\_enable: Whether to encrypt.

**Description**

#### Set device security parameter information (the device will restart automatically after setting).

#### Return value

0 succeeded; other values failed.

### sdks\_get\_dev\_time\_zone

#### Define

int sdks\_get\_dev\_time\_zone(unsigned int handle, char\*\* result);

#### parameter

handle：Return value of sdks\_dev\_conn interface.

p\_result：Output parameters, return to get time zone, daylight saving time information. If the returned value is non-NULL, you need to call the sdks\_free\_result function to free the memory, otherwise it will cause a memory leak.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attributes | | | Types | Optional / required | description |
| Parameters | | | E | M |  |
|  | TimeZoneParam | | E | M |  |
|  | TimeZone | A | M | Time zone |
| DSTOpenFlag | A | M | Open flag |
| DSTOffset | A | M | Offset |
| BeginMonth | A | M | Start month |
| BeginWeekly | A | M | Start week |
| BeginWeekDays | A | M | Start day of the week |
| BeginTime | A | M | Starting time |
| EndMonth | A | M | End month |
| EndWeekly | A | M | End week |
| EndWeekDays | A | M | End of the day |
| EndTime | A | M | End Time |

p\_result the json format is：

{

"TimeZone": 210,

"DSTOpenFlag": true,

"DSTOffset": true,

"BeginMonth": 1,

"BeginWeekly": 1,

"BeginWeekDays": 1,

"BeginTime": 1,

"EndMonth": 2,

"EndWeekly": 2,

"EndWeekDays": 2,

"EndTime": 2

}

**Description**

#### Get time zone and daylight saving time information.

#### Return value

0 succeeded; other values failed.

### sdks\_set\_dev\_time\_zone

#### Define

int sdks\_set\_dev\_time\_zone(unsigned int handle, char\* p\_dev\_time);

#### parameter

handle：Return value of sdks\_dev\_conn interface.

p\_dev\_time：Input parameters, expressed in json format.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attributes | | | Types | Optional / required | description |
| Parameters | | | E | M |  |
|  | TimeZoneParam | | E | M |  |
|  | TimeZone | A | M | Time zone |
| DSTOpenFlag | A | M | Open flag |
| DSTOffset | A | M | Offset |
| BeginMonth | A | M | Start month |
| BeginWeekly | A | M | Start week |
| BeginWeekDays | A | M | Start day of the week |
| BeginTime | A | M | Starting time |
| EndMonth | A | M | End month |
| EndWeekly | A | M | End week |
| EndWeekDays | A | M | End of the day |
| EndTime | A | M | End Time |

p\_dev\_time the json format is：

{

"TimeZone": 210,

"DSTOpenFlag": true,

"DSTOffset": true,

"BeginMonth": 1,

"BeginWeekly": 1,

"BeginWeekDays": 1,

"BeginTime": 1,

"EndMonth": 2,

"EndWeekly": 2,

"EndWeekDays": 2,

"EndTime": 2

}

**Description**

#### Set time zone and daylight saving time information.

#### Return value

0 succeeded; other values failed.

### sdks\_get\_p2p\_para

#### Define

int sdks\_dev\_get\_p2p\_para(unsigned int handle, int chn, char\*\* p\_result);

#### parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number, IPC fill-1.

p\_result：Output parameters, return the obtained P2P parameter information. If the returned value is non-NULL, you need to call the sdks\_free\_result function to free the memory, otherwise it will cause a memory leak.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attributes | | | Types | Optional / required | description |
| Parameters | | | E | M |  |
|  | P2PParam | | E | M |  |
|  | IsSupportP2P | A | M | Whether to support p2p |
| UUID | A | M | UUID code of the device |

p\_result the json format is：

{

"is\_support": 0,

"uuid": ""

}

**Description**

#### Obtain P2P parameter information.

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_get\_nvr\_channel\_name（for NVR）

#### Define

#### int sdks\_dev\_get\_nvr\_channel\_name (unsigned int handle, int chn, char\*\* p\_result)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

p\_result：Output parameters to return all channel information of NVR. If the returned value is not null, you need to call the sdk'free'result function to release memory, otherwise it will cause memory leak.

|  |  |  |  |
| --- | --- | --- | --- |
| Key field | Explain | Is it necessary | Remarks |
| channel\_id | ID number | Y |  |
| camera\_name | name | Y |  |
| camera\_model | model | Y |  |

p\_result the json format is：

{

"camera\_dev\_list": [{

"channel\_id": 1,

"camera\_name": "cx",

"camera\_model": ""

}, {

"channel\_id": 2,

"camera\_name": "Channel02",

"camera\_model": ""

}, {

"channel\_id": 3,

"camera\_name": "Channel03",

"camera\_model": ""

}, {

"channel\_id": 4,

"camera\_name": "Channel04",

"camera\_model": ""

}]

}

**Description**

#### Get all channel names of NVR.

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_get\_nvr\_chn\_info（for NVR）

#### Define

#### int sdks\_dev\_get\_nvr\_chn\_info(unsigned int handle, char\*\* p\_result)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

p\_result：Output parameters, return channel status information. If the returned value is non-NULL, you need to call the sdks\_free\_result function to free the memory, otherwise it will cause a memory leak.

|  |  |  |  |
| --- | --- | --- | --- |
| parameter | Description | Whether it is necessary | Remarks |
| chn | aisle | Y |  |
| status | Channel status | Y | 0: idle, 4097: offline, 4098: online, 4099: authentication failed, 4100: unknown |
| name | Channel name | Y |  |
| id | Device ID |  |  |
| type | Device type |  | See Appendix[1.12 Equipment type](#_1.12_Equipment_type) |

p\_result the json format is：

{

"data": [{

"chn": 1,

"status": 0,

"name": "",

"id": "112f37bf79440",

"type": 1

}, {

"chn": 2,

"status": 0,

"name": "",

"id": "112f37bf79440",

"type": 1

}, {

"chn": 3,

"status": 4098,

"name": "Channel03",

"id": "112f37bf79440",

"type": 1

}, {

"chn": 4,

"status": 0,

"name": "",

"id": "112f37bf79440",

"type": 1

}]

}

**Description**

#### Returns status information for each channel.

#### Return value

0 succeeded; other values failed.

### sdks\_get\_protocol\_security\_param（for IPC）

#### Define

#### int sdks\_get\_protocol\_security\_param(unsigned int handle, char\*\* p\_result)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

p\_result：Output parameters to return the obtained security protocol information. If the returned value is not null, you need to call the sdk'free'result function to release memory, otherwise it will cause memory leak.

|  |  |  |  |
| --- | --- | --- | --- |
| Key field | Explain | Is it necessary | Remarks |
| Enable | Protocol security open sign | Y |  |
| Type | Protocol security type | Y |  |

p\_result the json format is：

{

"Enable": true,

"Type": 0

}

**Description**

#### Get security protocol information.

#### Return value

0 succeeded; other values failed.

### sdks\_set\_protocol\_security\_param（for IPC）

#### Define

#### int sdks\_set\_protocol\_security\_param(unsigned int handle, char\* p\_param)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

p\_param：Input parameters, expressed in json format.

|  |  |  |  |
| --- | --- | --- | --- |
| Key field | Explain | Is it necessary | Remarks |
| Enable | Protocol security open sign | Y |  |
| Type | Protocol security type | Y |  |

p\_ param the json format is：

{

"Enable": true,

"Type": 0

}

**Description**

#### Set the security protocol information.

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_get\_video\_system

#### Define

#### int sdks\_dev\_get\_video\_system(unsigned int handle, char\*\* p\_result)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

p\_result：Output parameters, return the obtained video format parameters. If the returned value is not null, you need to call the sdk'free'result function to release memory, otherwise it will cause memory leak.

|  |  |  |  |
| --- | --- | --- | --- |
| Key field | Explain | Is it necessary | Remarks |
| VideoSystemId | Video system ID | Y | 0：NTSC， 1：PAL |
| Frequency | Video acquisition frequency | Y |  |

p\_result the json format is：

{

"VideoSystemId": 1,

"Frequency": 50

}

**Description**

#### Get video format parameters.

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_set\_video\_system

#### Define

#### int sdks\_dev\_set\_video\_system(unsigned int handle, char\* p\_param)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

p\_param：Input parameters, expressed in json format.

|  |  |  |  |
| --- | --- | --- | --- |
| Key field | Explain | Is it necessary | Remarks |
| VideoSystemId | Video system ID | Y | 0：NTSC， 1：PAL |
| Frequency | Video acquisition frequency | Y |  |

p\_ param the json format is：

{

"VideoSystemId": 1,

"Frequency": 50

}

**Description**

#### Set the video standard parameters (the device will restart automatically after setting).

#### Return value

0 succeeded; other values failed.

## PTZ operation

### PTZ operation type table：

PTZ\_STOP = 0, // stop

PTZ\_UP = 1, // up

PTZ\_DOWN = 2, // down

PTZ\_LEFT = 3, // left

PTZ\_RIGHT = 4, // right

PTZ\_LEFT\_UP = 5, // Top left

PTZ\_LEFT\_DOWN = 6, // Bottom left

PTZ\_RIGHT\_UP = 7, // Upper right

PTZ\_RIGHT\_DOWN = 8, // Bottom right

PTZ\_ZOOM\_IN = 9, // Zoom in

PTZ\_ZOOM\_OUT = 10, // zoom out

PTZ\_FOCUS\_FAR = 11, // Far focus

PTZ\_FOCUS\_NEAR = 12, // Near focus

PTZ\_IRIS\_INC = 13, // Aperture becomes larger

PTZ\_IRIS\_DEC = 14, // Aperture reduction

PTZ\_PRESET\_SET = 15, // Preset setting

PTZ\_PRESET\_CALL = 16, // Preset bit call

PTZ\_PRESET\_DEL = 17, // Preset deletion

PTZ\_SCAN\_CALL=21, // Scan call

PTZ\_SCAN\_SET\_START = 22, // Set scan start point

PTZ\_SCAN\_SET\_STOP = 23, // Set the scan end point

PTZ\_AUTO\_FOCUS = 24, // Auto focus

PTZ\_AUTO\_IRIS = 25, // Automatic aperture

PTZ\_START\_AUTO\_STUDY = 26, // Start self-learning

PTZ\_END\_AUTO\_STUDY = 27, // End self-learning

PTZ\_RUN\_AUTO\_STUDY = 28, // Self-learning call

PTZ\_RESET = 29, // Reset

PTZ\_3D\_ORIENTATION = 30, // 3D intelligent positioning

PTZ\_TOUR\_SET\_START = 31, // Set the cruise start point

PTZ\_TOUR\_ADD\_PRESET = 32, // Add cruise presets

PTZ\_TOUR\_SET\_END = 33, // Set the cruise end point

PTZ\_TOUR\_RUN = 34, // Call parade

PTZ\_TOUR\_PAUSE = 35, // Pause cruise

PTZ\_TOUR\_DEL = 36, // Delete cruise

PTZ\_TOUR\_CONTINUE = 200, // Continue the parade (use with the suspension parade)

PTZ\_KEEPER\_SET = 37, // Watch position setting

PTZ\_KEEPER\_RUN = 38, // Running the guard position

PTZ\_RUN\_BRUSH = 39, // Wiper running

PTZ\_OPEN\_LIGHT = 40, // Turn on the light

PTZ\_CLOSE\_LIGHT = 41, // Turn off the light

PTZ\_SCAN\_REMOVE = 44, // Delete scan

PTZ\_REMOVE\_AUTO\_STUDY = 45, // Delete self-learning

PTZ\_INFRARED\_CTRL = 46, // Infrared light control

PTZ\_GET\_PTZ\_POSTION\_REQ = 47, // Request to get a PTZ location

PTZ\_GET\_PTZ\_POSTION\_RESP = 48, // PTZ position response

PTZ\_SET\_PTZ\_POSTION = 49, // Set the PTZ position

PTZ\_SET\_PTZ\_NORTH\_POSTION = 50, // Set the north position

PTZ\_GET\_PRESET\_REQ = 51, // Get a preset request

PTZ\_GET\_PRESET\_RESP = 52, // Get preset response

PTZ\_GET\_TOUR\_REQ = 53, // Get a cruise request

PTZ\_GET\_TOUR\_RESP = 54, // Get a cruise response

PTZ\_GET\_SCAN\_REQ = 55, // Get scan request

PTZ\_GET\_SCAN\_RESP = 56, // Get scan response

PTZ\_GET\_AUTO\_STUDY\_REQ = 57, // Get self-learning request

PTZ\_GET\_AUTO\_STUDY\_RESP = 58, // Get self-learning response

PTZ\_GET\_KEEPER\_REQ = 59, // Get a guard request

PTZ\_GET\_KEEPER\_RESP = 60, // Get a guard answer

PTZ\_INFRARED\_STRL\_V2 = 61, // Infrared light control extended command

PTZ\_INFRARED\_STRL\_V2\_REQ = 62, // Request infrared light control parameter command

PTZ\_STOP\_BRUSH = 63, // Wiper stop

PTZ\_360\_ROTATE\_SCAN = 64, //360°Rotary scan

PTZ\_PERPENDICVULAR\_SCAN = 65, // Vertical scan

PTZ\_HEART\_BEAT = 66, // Heartbeat

PTZ\_INFRARED\_CTRL\_V2\_RESP = 67, // Request infrared light control parameter command response

PTZ\_GET\_ALARM\_IO\_START\_REQ = 70, // Request to get the alarm IO status

PTZ\_GET\_ALARM\_IO\_START\_RESP = 71, // Alarm IO status response

PTZ\_PT\_STOP\_STATUS\_RESP = 72, // PT stop status query

PTZ\_PT\_POS\_AUTO\_RESP = 73, // Automatically report PT coordinates

PTA\_ALARM\_IO\_STATUS\_AUTO\_RESP = 74, // Automatically report IO alarm status

PTZ\_GET\_ZOOM\_VALUE = 75, // Lens magnification

PTZ\_GET\_PTZ\_VERSION = 76, // Get the PTZ version number

PTZ\_GET\_MCU\_TEMPERATURE = 77, // Get MCU temperature

PTZ\_LOAD\_DEFAULT = 78, // Clean up all operations

PTZ\_GET\_PT\_POSTION = 79,

PTZ\_SET\_VERTICAL\_MAX\_POSTION = 80,

PTZ\_LENS\_RESET = 81, /\* Autofocus lens (including ABF) reset \*//\*BOOL\*/

PTZ\_AUTO\_TRACK = 82,

PTZ\_GET\_PTZ\_ACTION\_STATUS\_REQ = 83, // Request to get PTZ motion status

PTZ\_GET\_PTZ\_ACTION\_STATUS\_RESP = 84, // PTZ motion state response

PTZ\_SET\_WIPER\_MODE = 85, // Set the wiper mode

PTZ\_GET\_WIPER\_MODE = 86, // Get the wiper mode

PTZ\_SET\_PTZ\_POWER\_SAVE = 87, // Set PTZ power saving

PTZ\_GET\_PTZ\_POWER\_SAVE = 88, // Get PTZ power saving

PTZ\_SET\_PT\_LIMIT\_POS = 89, // Set PT limit position

PTZ\_GET\_PT\_LIMIT\_POS\_REQ = 90, // Get PT limit location request

PTZ\_GET\_PT\_LIMIT\_POS\_RESP = 91, // Get PT limit location response

PTZ\_CLEAR\_PT\_LIMIT\_POS = 92, // Clear PT limit position

PTZ\_SET\_PT\_SELFCHECK = 93, // Set PT self-test

PTZ\_GET\_PT\_SELFCHECK = 94, // Get PT self-test

PTZ\_SET\_ORIENTATION = 95, // Set the installation method

PTZ\_GET\_ORIENTATION = 96, // Get the installation method

PTZ\_SET\_SHORTCUT = 97, // Set shortcuts

PTZ\_GET\_SHORTCUT = 98, // Get shortcuts

PTZ\_SET\_DN\_MODE = 99, // Set day and night mode

PTZ\_SET\_WHITE\_LIGHT = 100, // Set the white light status

PTZ\_GET\_WHITE\_LIGHT = 101, // Get white light status

PTZ\_GET\_DN\_MODE = 102, // Get day and night mode

PTZ\_SET\_ZOOM\_VALUE = 103, // Set the zoom value

PTZ\_SET\_FOCUS\_VALUE = 104, // Set the focus value

PTZ\_GET\_FOCUS\_VALUE = 105, // Get the focus value

PTZ\_BOW\_SCAN = 110, //bow scan

PTZ\_BOW\_SCAN\_SET\_STARTPOINT = 111, // Set the bow scan start point

PTZ\_BOW\_SCAN\_SET\_STOPPOINT = 112, // Set the end of the bow scan

PTZ\_BOW\_SCAN\_REMOVE = 113, // Delete bow scan

PTZ\_BOW\_SCAN\_PAUSE = 114, // Pause bow scan

PTZ\_BOW\_SCAN\_CONTINUE = 115, // Continue bow scan

PTZ\_OPEN\_DEFOG = 120, // Open fog

PTZ\_CLOSE\_DEFOG = 121 // Close fog

### sdks\_dev\_open\_ptz

#### Define

int sdks\_dev\_open\_ptz(unsigned int handle);

#### parameter

handle：Return value of sdks\_dev\_conn interface。

**Description**

#### Initialize the PTZ. This function is called only once and is called before the PTZ is operated. Otherwise, the subsequent PTZ operation will fail.

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_close\_ptz

#### Define

int sdks\_dev\_close\_ptz(unsigned int handle);

#### parameter

handle：Return value of sdks\_dev\_conn interface。

**Description**

#### Close the pan/tilt. This function is called only once. When the upper layer no longer uses the pan/tilt, this function can be used to close the pan/tilt.

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_ptz\_stop

#### Define

int sdks\_dev\_ptz\_stop(unsigned int handle, int chn);

#### parameter

handle：Return value of sdks\_dev\_conn interface。

chn：Channel number。

**Description**

#### Stop the pan/tilt operation of the specified channel

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_ptz\_rotate

#### Define

int sdks\_dev\_ptz\_rotate(unsigned int handle, int chn, int operation, int speed);

#### parameter

handle：Return value of sdks\_dev\_conn interface。

chn：Channel number。

operation：For the operation type, please refer to the PTZ operation type table (the interface operation type includes: left (3), right (4), upper (1), lower (2), upper left (5), upper right (7), lower Left (6), lower right (8), reset (29), vertical scan (65)).

speed：Speed, the value range is 0~64. The larger the value, the faster the speed. It is recommended to pass the value 32.

**Description**

#### Operate the left (3), right (4), upper (1), lower (2), upper left (5), upper right (7), lower left (6), and lower right (8) directions of the PTZ , and reset (29), vertical scan (65). Return value

0 succeeded; other values failed.

### sdks\_dev\_ptz\_zoom

#### Define

int sdks\_dev\_ptz\_zoom(unsigned int handle, int chn, int operation, int speed);

#### parameter

handle：Return value of sdks\_dev\_conn interface。

chn：Channel number。

operation：For the operation type, please refer to the PTZ operation type table (the interface operation type includes: zoom in (9), zoom out (10)).

speed：Speed, the value range is 0~64. The larger the value, the faster the speed. It is recommended to pass the value 32.

**Description**

#### Operate the pan/tilt zoom function.

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_ptz\_focus

#### Define

int sdks\_dev\_ptz\_focus(unsigned int handle, int chn, int operation);

#### parameter

handle：Return value of sdks\_dev\_conn interface。

chn：Channel number。

operation：For the operation type, please refer to the PTZ operation type table (the interface operation type includes: far focus (11), near focus (12), auto focus (24)).

**Description**

#### Operate the focus function of the gimbal.

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_ptz\_iris

#### Define

#### int sdks\_dev\_ptz\_iris(unsigned int handle, int chn, int operation)

#### parameter

handle：Return value of sdks\_dev\_conn interface。

chn：Channel number。

operation：For the operation type, please refer to the PTZ operation type table (the interface operation type includes: far focus (11), near focus (12), auto focus (24)).

**Description**

#### Operate the aperture function of the gimbal.

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_ptz\_preset

#### Define

int sdks\_dev\_ptz\_preset(unsigned int handle, int chn, int id, int operation);

#### parameter

handle：Return value of sdks\_dev\_conn interface。

chn：Channel number。

id：The id of the preset point starts from 1. When the preset point is set, if the id is repeated, the preset point set later will overwrite the previously set preset point.

operation：For the operation type, please refer to the PTZ operation type table (the interface operation type includes: preset point addition (15), preset point jump (16), preset point deletion (17)).

**Description**

#### Operate the preset function of the PTZ.

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_ptz\_track

#### Define

#### int sdks\_dev\_ptz\_track(unsigned int handle, int chn, int id, int operation)

#### parameter

handle：Return value of sdks\_dev\_conn interface。

chn：Channel number。

id：The id of the track point starts from 1. When the track point is set, if the id is repeated, the track point set later will overwrite the previously set track point.

operation：For the operation type, please refer to the PTZ operation type table (the interface operation type includes: track start point addition (26), track end point addition (27), track point jump (28), track point deletion (45)).

**Description**

#### Operate the trajectory function of the gimbal.

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_ptz\_scan

#### Define

#### int sdks\_dev\_ptz\_scan(unsigned int handle, int chn, int id, int operation)

#### parameter

handle：Return value of sdks\_dev\_conn interface。

chn：Channel number。

id：The id of the scan point starts from 1. When the scan point is set, if the id is repeated, the scan point set later will overwrite the scan point set previously.

operation：For the operation type, please refer to the PTZ operation type table (the interface operation type includes: scan start point addition (22), scan end point addition (23), scan point jump (21), scan point deletion (44)).

**Description**

#### Operate the scanning function of the PTZ.

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_ptz\_tour

#### Define

#### int sdks\_dev\_ptz\_tour(unsigned int handle, int chn, int id, int operation, int speed,int time)

#### parameter

handle：Return value of sdks\_dev\_conn interface。

chn：Channel number。

id：The id of the cruise point starts from 1. When the cruise point is set, if the id is repeated, the cruise point set later will overwrite the previously set cruise point.

operation：For the operation type, please refer to the PTZ operation type table (the interface operation type includes: cruise start point addition (31), cruise preset point addition (32), cruise end point addition (23), cruise point jump (34), Cruise point pause (35), cruise point continue (200), scan point delete (36)).

speed：Cruise preset speed (only useful when type is added to cruise preset point (32), other types can be set to 0)。

time：Cruise preset point dwell time (useful only when the type is added to the cruise preset point (32), other types can be set to 0).

**Description**

#### Cruise preset point dwell time (useful only when the type is added to the cruise preset point (32), other types can be set to 0).

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_ptz\_keeper

#### Define

#### int sdks\_dev\_ptz\_keeper(unsigned int handle, int chn, int operation, int enable, int type, int id, int time)

#### parameter

handle：Return value of sdks\_dev\_conn interface。

chn：Channel number。

operation：For the operation type, please refer to the PTZ operation type table (the interface operation type includes: watch position setting (37), watch position open (38)).

enable：Watch position open flag, start status 0x02: open; 0x00 off (only useful when the type is watched on (38), other types can be set to 0).

type：Watch position type, 1: preset position, 2: scan, 3: self-learning, 4: cruise (only useful when the type is set to watch position (37), other types can be set to 0).

id：Watch position id number (only useful if the type is set to watch position (37), other types can be set to 0).

time：Watchkeeper wait time, in seconds (s) (only useful if the type is set to watch position (37), other types can be set to 0).

**Description**

#### Operate the PTZ's watch position function.

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_ptz\_threeDimensionalPos

#### Define

int sdks\_dev\_ptz\_threeDimensionalPos(unsigned int handle, int chn, int nX, int nY, int nZoomaTate);

#### parameter

handle：Return value of sdks\_dev\_conn interface。

chn：Channel number。

nX：x coordinate range: -100~100

nY：y coordinate range: -100~100

nZoomaTate：Zoom range: -0x7F~0x7F(-127~127)

**Description**

### Operate the 3D ptz function of the PTZ.去掉

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_ptz\_brush

#### Define

#### int sdks\_dev\_ptz\_brush(unsigned int handle, int chn, int operation, int mode, int waittime) parameter

handle：Return value of sdks\_dev\_conn interface。

chn：Channel number。

operation：For the operation type, please refer to the PTZ operation type table (this interface operation type includes: open the wiper (39), close the wiper (63), set the wiper mode (85)). mode： Wiper mode, 0: OFF, 1: ON, 2: AUTO (only useful when the type is set to wiper mode (85), other types can be set to 0).

waittime：Waiting time in seconds (s) (only useful if the type is set to wiper mode (85), other types can be set to 0).

**Description**

#### Operate the wiper function of the gimbal.

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_ptz\_light

#### Define

#### int sdks\_dev\_ptz\_light(unsigned int handle, int chn, int operation)

#### parameter

handle：Return value of sdks\_dev\_conn interface。

chn：Channel number。

operation：For the operation type, please refer to the PTZ operation type table (the interface operation type includes: turning on the fluorescent lamp (40), turning off the fluorescent lamp (41)).

**Description**

#### Operate the fluorescent lamp function of the gimbal.

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_ptz\_defog

#### Define

#### int sdks\_dev\_ptz\_defog(unsigned int handle, int chn, int operation)

#### parameter

handle：Return value of sdks\_dev\_conn interface。

chn：Channel number。

operation：For the operation type, please refer to the PTZ operation type table (the interface operation type includes: open defogging (120), turn off defogging (121)).

**Description**

#### Operate the defogging function of the gimbal.

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_ptz\_postion

#### Define

#### int sdks\_dev\_ptz\_postion(unsigned int handle, int chn, int operation, int type, int p\_nPan, int p\_nTilt, int p\_nZoom)

#### parameter

handle：Return value of sdks\_dev\_conn interface。

chn：Channel number。

operation：For the operation type, please refer to the PTZ operation type table (the interface operation type includes: setting the pan/tilt position (49), setting the north position (50)).

type：Position type; 0x00 indicates the position of the step. At this time, the zoom is valid by default. Non-zero 0x00 indicates the angular position. Bit\_2 is 1 for the zoom to be valid. Otherwise, the zoom is invalid. (Only when the type is set to the position of the pan/tilt (49 is useful, other types can be set. 0).

P\_nPan：Angle/step number of horizontal position Combine p\_nPTZPosionType to determine the meaning of the parameter (only if the type is set to the position of the gimbal (49 is useful, other types can be set to 0).

P\_nTilt：**Angle/step number of vertical position Combine p\_nPTZPosionType to determine the meaning of the parameter (only if the type is set to the position of the gimbal (49 is useful, other types can be set to 0).**

P\_nZoom：Magnification (only useful if the type is set to PTZ position (49 is useful, other types can be set to 0).

**Description**

#### Set the position of the gimbal.

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_get\_ptz\_postion

#### Define

int sdks\_dev\_get\_ptz\_postion(unsigned int handle, int chn, char\*\* p\_result)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number.

p\_result: Output parameters, return the acquired PTZ location information, expressed in json format. If the returned value is non-NULL, you need to call the sdks\_free\_result function to free the memory, otherwise it will cause a memory leak.

|  |  |
| --- | --- |
| Attributes | description |
| pan | Angle/step number of horizontal position |
| tilt | Vertical position angle / number of steps |
| zoom | Zoom |

p\_result the json format is：

{

"pan": "0",

"tilt": "100",

"zoom": "0)"

}

**Description**

#### Get the PTZ location.

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_get\_ptz\_req

#### Define

int sdks\_dev\_get\_ptz\_req(unsigned int handle, int chn, char\*\* p\_result)

#### parameter

handle：Return value of sdks\_dev\_conn interface。

chn：Channel number。

p\_result: Output parameters, return the acquired PTZ request information, expressed in json format. If the returned value is non-NULL, you need to call the sdks\_free\_result function to free the memory, otherwise it will cause a memory leak.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attributes | | | Description | |
| Parameters | | | The message begins | |
| p\_result | | | process result | |
| ptz\_para\_list | | | PTZ parameter list | |
|  | ptz\_para | |  | |
|  |  | enable | Whether to start the gimbal | |
|  | channel\_id | Camera number corresponding to the cloud platform | |
|  | dev\_id | PTZ device address | |
|  | protocol | Yuntai Agreement |
|  | com\_id | Serial port id of PTZ connection |
|  | baud\_rate | Baud rate |
|  | data\_bits | Data bit |
|  | stop\_bits |  |
|  | parity | Parity |
| ptz\_id | | | Built-in pan/tilt ID | |
| ptz\_type | | | PTZ type 1 is a high speed ball, 0 is a gun | |

p\_result the json format is：

{

"ptz\_para\_list": [{

"enable": 0,

"channel\_id": 1,

"dev\_id": 1,

"protocol": 0,

"com\_id": 0,

"baud\_rate": 9600,

"data\_bits": 8,

"stop\_bits": 0,

"parity": 0

}],

"ptz\_id": 1,

"ptz\_type": 1

}

**Description**

Get the PTZ request parameters.

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_set\_ptz\_speed

#### Define

#### int sdks\_dev\_set\_ptz\_speed(unsigned int handle, int chn, int speed)

#### parameter

handle：Return value of sdks\_dev\_conn interface。

chn：Channel number。

speed: Speed (ie gear)，0~64。

**Description**

#### Set the speed of the gimbal.

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_get\_ptz\_configue

#### Define

#### int sdks\_dev\_get\_ptz\_configue(unsigned int handle, int chn, int operation, char\*\* p\_result) parameter

handle：Return value of sdks\_dev\_conn interface。

chn：Channel number。

operation：Operation type, 0: Get preset parameters, 1: Get scan parameters, 2: Get track parameters, 3: Get cruise parameters, 4: Get watch position parameters, 6: Get speed parameters.

p\_result: Output parameters, return the acquired information about the function parameters of the PTZ, expressed in json format. If the returned value is non-NULL, you need to call the sdks\_free\_result function to free the memory, otherwise it will cause a memory leak.

Get the preset position parameter information as follows:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | | | Types | Optional / required | description |
| Parameters | | | | | | E | M | The message begins |
|  | Version | | | | | A | M | Version information 1.0 |
| Result | | | | | E | M |  |
|  | Code | | | | A | M | 0: success, other errors |
| PTZPresetList | | | | | E | O |  |
|  | PTZPreset | | | | E | M |  |
|  | PresetId | | | A | M | ID number of the PTZ prefabrication point |
| PresetName | | | A | M | The name of the Yuntai prefabricated point |
| DNMode | | | A | M | DN module |
| PTZOperationInfo | | | E | M |  |
|  | CommonID | | A | M | Command ID |
| PanSpeed | | A | M | Horizontal speed |
| TiltSpeed | | A | M | Vertical speed |
| ZoomSpeedValue | | A | M | Zoom speed |
| FocusSpeedValue | | A | M | Focusing speed |
| DirectionValue | | A | M | Direction of rotation |
| StopTime | | A | M | Stop time |
| Type | | A | M | Types of |
| PTZVersion | | A | M | PTZ version number |
| MCUTemperature | | A | M | MCU temperature |
| VerticalMaxPosition | | A | M | Maximum angle in the vertical direction |
| PTZPointParamV2 | | E | M | Record operating point location information |
|  | PanPosion | A | M | Horizontal coordinate |
| TiltPosion | A | M | Vertical coordinate |
| ZoomValue | A | M | Zoom multiple |
| DZoomValue | A | M | Digital zoom value |
| DayFocusValue | A | M | Day focus value |
| NightFocusValue | A | M | Night focus value |
| AlarmIOStateList | | E | M | Alarm IO status |

p\_result the json format is：

{

"PTZPresetList": [{

"PresetId": 1,

"PresetName": "Preset1",

"DNMode": 0,

"PTZOperationInfo": {

"CommonID": 0,

"PanSpeed": 0,

"TiltSpeed": 0,

"ZoomSpeedValue": 0,

"FocusSpeedValue": 0,

"DirectionValue": 0,

"StopTime": 0,

"Type": 0,

"PTZVersion": "",

"MCUTemperature": 0,

"VerticalMaxPosition": 0,

"PTZPointParam": {

"PanPosion": 206679,

"TiltPosion": 11541,

"ZoomValue": 1494,

"DZoomValue": 0,

"DayFocusValue": 2239,

"NightFocusValue": 0,

"PTZID": 0,

"SubZoomValue": 0,

"SubDZoomValue": 0,

"SubFocusValue": 0

}

}

}, {

"PresetId": 2,

"PresetName": "Preset2",

"DNMode": 0,

"PTZOperationInfo": {

"CommonID": 0,

"PanSpeed": 0,

"TiltSpeed": 0,

"ZoomSpeedValue": 0,

"FocusSpeedValue": 0,

"DirectionValue": 0,

"StopTime": 0,

"Type": 0,

"PTZVersion": "",

"MCUTemperature": 0,

"VerticalMaxPosition": 0,

"PTZPointParam": {

"PanPosion": 206679,

"TiltPosion": 46217,

"ZoomValue": 1494,

"DZoomValue": 0,

"DayFocusValue": 2225,

"NightFocusValue": 0,

"PTZID": 0,

"SubZoomValue": 0,

"SubDZoomValue": 0,

"SubFocusValue": 0

}

}

}]

}

Get the scan parameter information as follows:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | | | | | Types | Optional / required | description |
| Parameters | | | | | | | | E | M | The message begins |
|  | Version | | | | | | | A | M | Version information 1.0 |
| Result | | | | | | | E | M |  |
|  | Code | | | | | | A | M | 0: success, other errors |
| PTZScanList | | | | | | | E | M |  |
|  | PTZScan | | | | | | E | O |  |
|  | ScanId | | | | | A | M | ID number of the pan/tilt scan |
| ScanName | | | | | A | M | Name of the pan-tilt scan |
| PTZPresetV2List | | | | | E | M |  |
|  | PTZPreset | | | | E | M |  |
|  | PresetId | | | A | M | ID number of the PTZ prefabrication point |
| PresetName | | | A | M | The name of the Yuntai prefabricated point |
| DNMode | | | A | M | DN module ID |
| PTZOperationInfo | | | E | M |  |
|  | CommonID | | A | M | Command ID |
| PanSpeed | | A | M | Horizontal speed |
| TiltSpeed | | A | M | Vertical speed |
| ZoomSpeedValue | | A | M | Zoom speed |
| FocusSpeedValue | | A | M | Focusing speed |
| DirectionValue | | A | M | Direction of rotation |
| StopTime | | A | M | Stop time |
| Type | | A | M | Types of |
| PTZVersion | | A | M | PTZ version number |
| MCUTemperature | | A | M | MCU temperature |
| VerticalMaxPosition | | A | M | Maximum angle in the vertical direction |
| PTZPointParamV2 | | E | M | Record operating point location information |
|  | PanPosion | A | M | Horizontal coordinate |
| TiltPosion | A | M | Vertical coordinate |
| ZoomValue | A | M | Zoom multiple |
| DZoomValue | A | M | Digital zoom value |
| DayFocusValue | A | M | Day focus value |
| NightFocusValue | A | M | Night focus value |
| AlarmIOStateList | A | M | Alarm IO status |

p\_result the json format is：

{

"PTZScanList": [{

"ScanId": 0,

"ScanName": "Scan1",

"PTZPresetList": [{

"PresetId": 1,

"PresetName": "StartPoint",

"DNMode": 0,

"PTZOperationInfo": {

"CommonID": 0,

"PanSpeed": 3000,

"TiltSpeed": 3000,

"ZoomSpeedValue": 0,

"FocusSpeedValue": 0,

"DirectionValue": 2,

"StopTime": 0,

"Type": 0,

"PTZVersion": "",

"MCUTemperature": 0,

"VerticalMaxPosition": 0,

"PTZPointParam": {

"PanPosion": 115324,

"TiltPosion": 24928,

"ZoomValue": 0,

"DZoomValue": 0,

"DayFocusValue": 0,

"NightFocusValue": 0,

"PTZID": 0,

"SubZoomValue": 0,

"SubDZoomValue": 0,

"SubFocusValue": 0

}

}

}, {

"PresetId": 2,

"PresetName": "EndPoint",

"DNMode": 0,

"PTZOperationInfo": {

"CommonID": 0,

"PanSpeed": 3000,

"TiltSpeed": 3000,

"ZoomSpeedValue": 0,

"FocusSpeedValue": 0,

"DirectionValue": 1,

"StopTime": 0,

"Type": 0,

"PTZVersion": "",

"MCUTemperature": 0,

"VerticalMaxPosition": 0,

"PTZPointParam": {

"PanPosion": 115324,

"TiltPosion": 45142,

"ZoomValue": 0,

"DZoomValue": 0,

"DayFocusValue": 0,

"NightFocusValue": 0,

"PTZID": 0,

"SubZoomValue": 0,

"SubDZoomValue": 0,

"SubFocusValue": 0

}

}

}]

}, {

"ScanId": 5,

"ScanName": "Scan6",

"PTZPresetList": [{

"PresetId": 1,

"PresetName": "StartPoint",

"DNMode": 0,

"PTZOperationInfo": {

"CommonID": 0,

"PanSpeed": 3000,

"TiltSpeed": 3000,

"ZoomSpeedValue": 0,

"FocusSpeedValue": 0,

"DirectionValue": 2,

"StopTime": 0,

"Type": 0,

"PTZVersion": "",

"MCUTemperature": 0,

"VerticalMaxPosition": 0,

"PTZPointParam": {

"PanPosion": 85666,

"TiltPosion": 45142,

"ZoomValue": 0,

"DZoomValue": 0,

"DayFocusValue": 0,

"NightFocusValue": 0,

"PTZID": 0,

"SubZoomValue": 0,

"SubDZoomValue": 0,

"SubFocusValue": 0

}

}

}, {

"PresetId": 2,

"PresetName": "EndPoint",

"DNMode": 0,

"PTZOperationInfo": {

"CommonID": 0,

"PanSpeed": 3000,

"TiltSpeed": 3000,

"ZoomSpeedValue": 0,

"FocusSpeedValue": 0,

"DirectionValue": 1,

"StopTime": 0,

"Type": 0,

"PTZVersion": "",

"MCUTemperature": 0,

"VerticalMaxPosition": 0,

"PTZPointParam": {

"PanPosion": 140259,

"TiltPosion": 45142,

"ZoomValue": 0,

"DZoomValue": 0,

"DayFocusValue": 0,

"NightFocusValue": 0,

"PTZID": 0,

"SubZoomValue": 0,

"SubDZoomValue": 0,

"SubFocusValue": 0

}

}

}]

}]

}

Get the track parameter information as follows:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | | | Types | Optional / required | description |
| Parameters | | | | | | E | M | The message begins |
|  | Version | | | | | A | M | Version information 1.0 |
| Result | | | | | E | M |  |
|  | Code | | | | A | M | 0: success, other errors |
| PTZTrackList | | | | | E | O |  |
|  | PTZTrack | | | | E | M |  |
|  | ID | | | A | M | Track ID |
| Name | | | A | M | Track name |
| StartPanPos | | | A | M | Starting point horizontal position |
| StartTiltPos | | | A | M | Starting point vertical position |
| StartZoomPos | | | A | M | Starting point zoom position |
| CmdItemList | | | E | M |  |
|  | PTZTraceCmdItem | | E | M |  |
|  | PTStatus | A | M | status |
| IntervalTime | A | M | waiting time |
| CmdID | A | M | Command ID |
| PanSpeed | A | M | Horizontal speed |
| TiltSpeed | A | M | Vertical speed |
| ZoomSpeed | A | M | Zoom speed |
| PanSteps | A | M | Horizontal target position |
| TiltSteps | A | M | Vertical target position |
| ZoomPos | A | M | Zoom target position |

p\_result the json format is：

{

"PTZTrackList": [{

"ID": 0,

"Name": "Track1",

"StartPanPos": 57723,

"StartTiltPos": 0,

"StartZoomPos": 0,

"CmdItemList": [{

"PTStatus": 0,

"IntervalTime": 1562,

"CmdID": 4,

"PanSpeed": 2170,

"TiltSpeed": 195,

"ZoomSpeed": 0,

"PanSteps": 57723,

"TiltSteps": 0,

"ZoomPos": 0

}, {

"PTStatus": 0,

"IntervalTime": 1409,

"CmdID": 0,

"PanSpeed": 0,

"TiltSpeed": 0,

"ZoomSpeed": 0,

"PanSteps": 80694,

"TiltSteps": 0,

"ZoomPos": 0

}, {

"PTStatus": 1,

"IntervalTime": 1214,

"CmdID": 4,

"PanSpeed": 2170,

"TiltSpeed": 195,

"ZoomSpeed": 0,

"PanSteps": 82366,

"TiltSteps": 0,

"ZoomPos": 0

}, {

"PTStatus": 0,

"IntervalTime": 1909,

"CmdID": 0,

"PanSpeed": 0,

"TiltSpeed": 0,

"ZoomSpeed": 0,

"PanSteps": 113657,

"TiltSteps": 0,

"ZoomPos": 0

}, {

"PTStatus": 1,

"IntervalTime": 918,

"CmdID": 0,

"PanSpeed": 0,

"TiltSpeed": 0,

"ZoomSpeed": 0,

"PanSteps": 115324,

"TiltSteps": 0,

"ZoomPos": 0

}]

}, {

"ID": 4,

"Name": "Track5",

"StartPanPos": 115324,

"StartTiltPos": 0,

"StartZoomPos": 0,

"CmdItemList": [{

"PTStatus": 0,

"IntervalTime": 771,

"CmdID": 1,

"PanSpeed": 550,

"TiltSpeed": 818,

"ZoomSpeed": 0,

"PanSteps": 115324,

"TiltSteps": 0,

"ZoomPos": 0

}, {

"PTStatus": 0,

"IntervalTime": 3713,

"CmdID": 0,

"PanSpeed": 0,

"TiltSpeed": 0,

"ZoomSpeed": 0,

"PanSteps": 115324,

"TiltSteps": 24444,

"ZoomPos": 0

}, {

"PTStatus": 1,

"IntervalTime": 726,

"CmdID": 0,

"PanSpeed": 0,

"TiltSpeed": 0,

"ZoomSpeed": 0,

"PanSteps": 115324,

"TiltSteps": 24928,

"ZoomPos": 0

}]

}]

}

Get the parade parameter information as follows:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | | | Types | Optional / required | description |
| Parameters | | | | | | E | M | The message begins |
|  | Version | | | | | A | M | Version information 1.0 |
| Result | | | | | E | M |  |
|  | Code | | | | A | M | 0: success, other errors |
| PTZTourList | | | | | E | O |  |
|  | PTZTour | | | | E | M |  |
|  | TourId | | | A | M | Cruise ID |
| TourName | | | A | M | Cruise name |
| PTZTourPointV2List | | | E | M |  |
|  | PTZTourPoint | | E | M |  |
|  | PresetId | A | M | Preset point ID |
| SpeedValue | A | M | speed |
| StopTime | A | M | Pause time |

p\_result the json format is：

{

"PTZTourList": [{

"TourId": 0,

"TourName": "Tour1",

"PTZTourPointV2List": [{

"PresetId": 2,

"SpeedValue": 0,

"StopTime": 5

}]

}, {

"TourId": 4,

"TourName": "Tour5",

"PTZTourPointV2List": [{

"PresetId": 3,

"SpeedValue": 0,

"StopTime": 2

}, {

"PresetId": 2,

"SpeedValue": 0,

"StopTime": 1

}]

}]

}

Obtain the guard position parameter information as follows:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attributes | | | Types | Optional / required | description |
| Parameters | | | E | M | The message begins |
|  | Version | | A | M | Version information 1.0 |
| Result | | E | M |  |
|  | Code | A | M | 0: success, other errors |
| PTZKeeper | | E | O |  |
|  | KeeperIdFromType | E | M |  |
|  |  | KeeperType |  |  |  |
|  |  | StatusId |  |  |  |
|  |  | WaitTime |  |  |  |

p\_result the json format is：

{

"PTZKeeper": {

"KeeperIdFromType": 1,

"KeeperType": 1,

"StatusId": 0,

"WaitTime": 10

}

}

Get the speed parameter information as follows:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | Types | Optional / required | description |
| Parameters | | | | E | M | The message begins |
|  | Version | | | A | M | Version information 1.0 |
| Result | | | E | M |  |
|  | Code | | A | M | 0: success, other errors |
| PTZSpeed | | | E | M |  |
|  |  | | ChannelId | A | M | Channel Id |
|  |  | | Speed | A | M | PTZ speed |

p\_result the json format is：

{

"PTZSpeed": {

"ChannelId": 0,

"Speed": 8

}

}

**Description**

#### Get information about the function parameters of the PTZ.

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_get\_ptz\_timer

#### Define

#### int sdks\_dev\_get\_ptz\_timer(unsigned int handle, int chn, char\*\* p\_result)

#### parameter

handle：Return value of sdks\_dev\_conn interface。

chn：Channel number。

p\_result: Output parameters, return the obtained PTZ timer parameter information, expressed in json format. If the returned value is non-NULL, you need to call the sdks\_free\_result function to free the memory, otherwise it will cause a memory leak.

|  |  |  |  |
| --- | --- | --- | --- |
| parameter | Explain | type | Remarks |
| ChannelId | Channel number | Int |  |
| EnableFlag | Enable PTZ timer or not | Bool |  |
| TimerMode | Time mode | Int | 1: once  2: Daily cycle |
| TimeZone | time zone | Int |  |
| DaylightSavingTime | Summer camp | Int |  |
| Year | year | Int |  |
| Month | month | Int |  |
| Day | day | Int |  |
| DayOfWeek | What day is it | Int |  |
| Hour | hour | Int |  |
| Minute | minute | Int |  |
| Second | second | Int |  |
| Milliseconds | milliseconds | Int |  |
| BeginTime | start time | Int | The number of seconds from 00:00 to the current time, for example, 5400 at 01:30 |
| EndTime | End time | Int | The number of seconds from 00:00 to the current time, for example, 5400 at 01:30 |
| PTZOperation | PTZ operation command | String |  |

p\_result the json format is：

{

"PTZTimer": [{

"ChannelId": 1,

"EnableFlag": true,

"TimerMode": 1,

"TimerDate": {

"TimeZone": 0,

"DaylightSavingTime": 0,

"Year": 2019,

"Month": 9,

"Day": 9,

"DayOfWeek": 0,

"Hour": 0,

"Minute": 0,

"Second": 0,

"Milliseconds": 0

},

"TimeSegmentList": [{

"BeginTime": 3600,

"EndTime": 9000,

"PTZOperation": "(16,1)"

}]

}]

}

**Description**

#### Get the PTZ timer parameter information.

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_set\_ptz\_timer

#### Define

#### int sdks\_dev\_set\_ptz\_timer(unsigned int handle, int chn, char\* p\_param)

#### parameter

handle：Return value of sdks\_dev\_conn interface。

chn：Channel number。

p\_param: Input parameters, expressed in json format.

|  |  |  |  |
| --- | --- | --- | --- |
| parameter | Explain | type | Remarks |
| ChannelId | Channel number | Int |  |
| EnableFlag | Enable PTZ timer or not | Bool |  |
| TimerMode | Time mode | Int | 1: once  2: Daily cycle |
| TimeZone | time zone | Int |  |
| DaylightSavingTime | Summer camp | Int |  |
| Year | year | Int |  |
| Month | month | Int |  |
| Day | day | Int |  |
| DayOfWeek | What day is it | Int |  |
| Hour | hour | Int |  |
| Minute | minute | Int |  |
| Second | second | Int |  |
| Milliseconds | milliseconds | Int |  |
| BeginTime | start time | Int | The number of seconds from 00:00 to the current time, for example, 5400 at 01:30 |
| EndTime | End time | Int | The number of seconds from 00:00 to the current time, for example, 5400 at 01:30 |
| PTZOperation | PTZ operation command | String |  |

p\_param the json format is：

{

"PTZTimer": [{

"ChannelId": 1,

"EnableFlag": true,

"TimerMode": 1,

"TimerDate": {

"TimeZone": 0,

"DaylightSavingTime": 0,

"Year": 2019,

"Month": 9,

"Day": 9,

"DayOfWeek": 0,

"Hour": 0,

"Minute": 0,

"Second": 0,

"Milliseconds": 0

},

"TimeSegmentList": [{

"BeginTime": 3600,

"EndTime": 9000,

"PTZOperation": "(16,1)"

}]

}]

}

**Description**

#### Set the PTZ timer parameter information.

#### Return value

0 succeeded; other values failed.

## Video parameter

### sdks\_get\_video\_param

#### Define

int sdks\_get\_video\_param (unsigned int handle, int chn ,char\*\* p\_result);

#### parameter

handle：sdks\_dev\_conn return value；

chn: channel ID

p\_result：output parameter, return the list of all stream video parameters in json format. If the return value is not NULL, it needs to call free function to free the memory. Otherwise it leads to memory leak.

|  |  |  |  |
| --- | --- | --- | --- |
| Key string | introduction | essential | remark |
| chn | Channel number | Y | 0或-1:device;other：NVR CH number |
| stream | Stream ID | Y | Stream ID |
| streamname | Stream name | Y | String |
| Height | Video height | Y |  |
| width | Video width | Y |  |
| fps | Frame | Y |  |
| bit\_type | Stream type | Y | 1:CBR，2:VBR |
| bps | Max bitrate | Y |  |
| quality | Video quality | Y | 1-9，1 best，9 worst |
| if\_int | I frame interval | Y |  |
| i\_unit | Iframe interval unit（sec） | Y |  |
| v\_enc | Video encode type | Y | See [1.14 Coding type](#_1.14_Coding_type) |
| v\_enc\_level | Encode level | Y | 0:low，1:mid，2:high |
| a\_enc | Audio encode type | Y | check audio encode type for detailed info [1.14 Coding type](#_1.14_Coding_type) |

p\_result the json format is：

{

"data": [{

"chn": 1,

"stream": 1,

"width": 2592,

"height": 1520,

"v\_enc": 8,

"fps": 25,

"bps": 4096,

"if\_int": 50,

"quality": 5,

"bit\_type": 1,

"v\_enc\_level": 2,

"i\_unit": 2,

"a\_enc": 102,

"streamname": "stream1"

}, {

"chn": 1,

"stream": 2,

"width": 704,

"height": 576,

"v\_enc": 8,

"fps": 25,

"bps": 1500,

"if\_int": 50,

"quality": 5,

"bit\_type": 1,

"v\_enc\_level": 0,

"i\_unit": 2,

"a\_enc": 102,

"streamname": "stream2"

}, {

"chn": 1,

"stream": 3,

"width": 352,

"height": 288,

"v\_enc": 8,

"fps": 25,

"bps": 256,

"if\_int": 50,

"quality": 5,

"bit\_type": 2,

"v\_enc\_level": 0,

"i\_unit": 2,

"a\_enc": 102,

"streamname": "stream3"

}]

}

**Description**

#### Get the video parameter information.

#### Return value

0 success, other:fail

### sdks\_set\_video\_param

#### Define

#### int sdks\_set\_video\_param(unsigned int handle, char\* p\_param);

#### parameter

handle：sdks\_dev\_conn return value；

p\_param： input parameter in json format;

|  |  |  |  |
| --- | --- | --- | --- |
| Key string | introduction | essential | remark |
| chn | Channel number | Y | 0或-1:device;other：NVR CH number |
| stream | Stream ID | Y | Stream ID |
| streamname | Stream name | Y | String |
| Height | Video height | Y |  |
| width | Video width | Y |  |
| fps | Frame | Y |  |
| bit\_type | Stream type | Y | 1:CBR，2:VBR |
| bps | Max bitrate | Y |  |
| quality | Video quality | Y | 1-9，1 best，9 worst |
| if\_int | I frame interval | Y |  |
| i\_unit | Iframe interval unit（sec） | Y |  |
| v\_enc | Video encode type | Y | See [1.14 Coding type](#_1.14_Coding_type) |
| v\_enc\_level | Encode level | Y | 0:low，1:mid，2:high |
| a\_enc | Audio encode type | Y | check audio encode type for detailed info [1.14 Coding type](#_1.14_Coding_type) |

p\_param the json format is：

{

"chn": 1,

"stream": 1,

"width": 2592,

"height": 1520,

"v\_enc": 8,

"fps": 25,

"bps": 4096,

"if\_int": 50,

"quality": 5,

"bit\_type": 1,

"v\_enc\_level": 2,

"i\_unit": 2,

"a\_enc": 102,

"streamname": "stream1"

}

**Description**

#### Set the video parameter information of a code stream.

#### Return value

#### 0 success, other:fail

**Encode type list：**

MPEG4 = 0, //MPEG4 encode

H264 = 1, //H264 encode

MJPEG = 2, //MJPEG encode

SVC = 3, //SVC encode

H264\_MAIN = 4, //H264 main profile//abort

H264\_HIGH = 5, //H264 high profile//abort

JPEG = 6, //JPEG encode

H265 = 7, //H265(base)

H265\_MAIN = 8, //H265(main)

H265\_HIGH = 9, //H265(high)

G7231 = 101, //G7231 encode

G711\_ALAW = 102, //G711A encode

G711\_ULAW = 103, //G711U encode

G722 = 104, //G722 encode

G726 = 105, //G726 encode

G729 = 106, //G729 encode

AMR = 107, //AMR encode

RAW\_PCM = 108, //PCM encode(no encode)

NONE = 109, //no audio

MOTION\_DETECTION = 201, //motion data

CAMERATAMPER\_DETECTION = 202, //mask motion data

LINECOUNTING\_DETECTION = 203, //line crossing detect data

UNKNOWN = 0xFF //unknown encode

### sdks\_get\_roi\_param

#### definition

#### int sdks\_get\_roi\_param(unsigned int handle, char\*\* p\_result);

#### parameter

handle：sdks\_dev\_conn return value

p\_result： return parmeter，return roi information in json format。If the returned value is not NULL, you need to call the free function to release the memory, otherwise it will cause a memory leak.

|  |  |  |  |
| --- | --- | --- | --- |
| Key | introduction | essential | comment |
| chn | Channel no. | Y | IPC:1 |
| streamid | Stream id | Y | 1：H265 、2.main stream、3.sub stream |
| name | Roi name | Y |  |
| indexid | Index ID | Y |  |
| enable | Enable roi | Y |  |
| level | Roi level | Y |  |
| x | Percentage of upper-left x coordinate to video width | Y |  |
| y | Percentage of upper-left y coordinate to video height | Y |  |
| width | Percentage of ROI width to video width | Y |  |
| height | Percentage of ROI height to video height | Y |  |

#### Description

Get all the ROI parameter of one channel

#### Return value

0:success; others:fail。

### sdks\_set\_roi\_param

#### definition

#### int sdks\_set\_roi\_param(unsigned int handle,int channel,int stream, char\* p\_roi\_param);

#### parameter

handle：sdks\_dev\_conn (return value of interface)。

channel: channel number ：IPC:1。

stream：stream type (：1, 2 or 3)

p\_roi\_param: json information of 8 set roi

eg:

[{"name":"ROI\_NAME\_1","indexid":1,"enable":1,"level":2,"x":5,"y":5,"width":25,"height":25},

{"name":"ROI\_NAME\_2","indexid":2,"enable":1,"level":2,"x":8,"y":9,"width":25,"height":25},

..]

#### description

set all the ROI parameter of on stream(8 ROI)

#### Return value

0:success; others:fail。

### sdks\_get\_svc\_stream\_para

#### Definition

int sdks\_get\_svc\_stream\_para(unsigned int handle, int chn, int stream\_id, char\*\* p\_result);

#### Parameter

handle: sdks\_dev\_conn (return value of interface)。

chn: The channel number, when -1 means all channels, and the rest are designated channels.

stream\_id: Stream ID: -1 indicates all streams (this parameter is set to -1).

p\_result: return parameter of svc in json format。If the return value is not NULL，you need to call the free function to release the memory, otherwise it will cause a memory leak.

|  |  |  |  |
| --- | --- | --- | --- |
| parameter | Explain | type | Remarks |
| source\_stream\_id | Original flow ID | int |  |
| svc\_stream\_id | Adjoint flow ID | int |  |
| frame\_skip\_inval | P frame skip interval | int |  |
| svc\_stream\_name | Stream name | string |  |

When chn is -1, the json format of p\_result is:

{

"svc\_para\_list": [{

"channel\_id": 1,

"svc\_stream\_list": [{

"source\_stream\_id": 1,

"svc\_stream\_id": 4,

"frame\_skip\_inval": 1,

"svc\_stream\_name": "stream4"

}]

}]

}

When chn is the remaining specified channel, the json format of p\_result is:

{

"svc\_para\_list": [{

"source\_stream\_id": 1,

"svc\_stream\_id": 4,

"frame\_skip\_inval": 1,

"svc\_stream\_name": "stream4"

}]

}

##### Description

Get svc parameter

#### Return value

0:success; others:fail。

## Network parameters

### sdks\_get\_net\_param

#### Define

#### int sdks\_get\_net \_param(unsigned int handle, char\*\* p\_result);

#### parameter

handle：sdks\_dev\_conn //Return value of interface；。

p\_result： Output parameter, returns the list of Network parameter information, represented by jsonformat. If the returned value is not NULL, you need to call the free function to release the memory, otherwise it will cause a memory leak.

|  |  |  |  |
| --- | --- | --- | --- |
| parameter | Instruction | Y/N | Note |
| netid | NIC number | Y | 1 |
| worktype | Network Type | Y | 1. Public network 2. Camera network 3. Binding |
| dhcp | Obtain IP  automatically | Y | (1: automatic acquisition, 0: automatic acquisition off) |
| ip | IP | Y |  |
| netmask | Subnet mask | Y |  |
| gw | Gateway | Y |  |
| dns1 | Preferred DNS | Y |  |
| dns2 | Alternate DNS | Y |  |
| ip\_proto | Address type | Y | “IPv4” “IPv6” |

p\_result the json format is：

{

"data": [{

"ip": "192.168.0.12",

"dhcp": 0,

"netmask": "255.255.255.0",

"gw": "192.168.0.1",

"dns1": "192.168.0.1",

"dns2": "192.168.0.2",

"netid": 1,

"worktype": 0,

"ip\_proto": "IPv4"

}]

}

**Description**

Get local network parameters.

#### Return value

0 succeeded; other values failed.

### sdks\_set\_net\_param

#### define

#### int sdks\_set\_net \_param(unsigned int handle, char\* p\_video\_param);

#### parameter

handle：sdks\_dev\_conn the return value of the interface；

p\_video\_param： Enter parameters that are passed in in json format

|  |  |  |  |
| --- | --- | --- | --- |
| Key field | instruction | Y/N | remark |
| netid | Network card number | Y |  |
| worktype | Type of network | Y | 1. Public network 2. Camera network 3. Binding |
| dhcp | auto-fetch ip | Y | (1:Automatic acquisition, 0:turn off automatic acquisition). |
| ip | IP | Y |  |
| netmask | subnet mask | Y |  |
| gw | gateway | Y |  |
| dns1 | Preferred dns | Y |  |
| dns2 | Alternate dns | Y |  |
| ip\_proto | address type | Y | “IPv4” “IPv6” |

p\_param the json format is：

{

"data": [{

"ip": "192.168.0.12",

"dhcp": 0,

"netmask": "255.255.255.0",

"gw": "192.168.0.1",

"dns1": "192.168.0.1",

"dns2": "192.168.0.2",

"netid": 1,

"worktype": 0,

"ip\_proto": "IPv4"

}]

}

**Description**

#### Set local network parameters.

#### Return value

0 succeeded; other values failed.

### sdks\_get\_ddns\_param

#### Define

#### int sdks\_get\_ddns \_param(unsigned int handle, char\*\* p\_result);

#### parameter

handle：sdks\_dev\_conn //Return value of interface；。

p\_result： Output parameter, returns DDNS service information, represented by jsonformat. If the returned value is not NULL, you need to call the free function to release the memory, otherwise it will cause a memory leak.

|  |  |  |  |
| --- | --- | --- | --- |
| parameter | Instruction | Y/N | Note |
| enable | Whether to enable ddns | Y | （1：enable，0：disable） |
| provider | Provider ID | Y |  |
| accounts | Account | Y |  |
| domainname | domain name | Y |  |
| pwd | password | Y |  |
| netcardname | Bind network card name | Y | “eth0” “wlan0” |

p\_result the json format is：

{

"enable": 0,

"provider": 1,

"accounts": "",

"domainname": "",

"password": "",

"netcardname": "eth0"

}

**Description**

#### Gets the DDNS parameter.

#### Return value

0 succeeded; other values failed.

### sdks\_set\_ddns\_param

#### Define

int sdks\_set\_ddns \_param(unsigned int handle, char\* p\_ param);

#### parameter

handle：Return value of sdks\_dev\_conn interface.

p\_param: Input the parameter in the form of JSON. The fields are as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| parameter | Instruction | Y/N | Note |
| enable | Whether to enable ddns | Y | （1：enable，0：disable） |
| provider\_id | Provider ID | Y |  |
| accounts | Account | Y |  |
| domain\_name | domain name | Y |  |
| password | password | Y |  |
| net\_card\_name | Bind network card name | Y | “eth0” “wlan0” |

p\_ param the json format is：

{

"enable": 0,

"provider": 1,

"accounts": "",

"domainname": "",

"password": "",

"netcardname": "eth0"

}

**Description**

#### Set DDNS parameters

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_get\_ddns\_provider(for IPC)

#### Define

int sdks\_dev\_get\_ddns\_provider(unsigned int handle, char\*\* p\_result);

#### parameter

handle：Return value of sdks\_dev\_conn interface.

p\_result: Output parameter, fill in the provider information parameter of DDNS and return it to the caller in JSON format,. If the returned value is not null, you need to call the free function to free memory, otherwise it will cause memory leak.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attributes | | | | Instruction |
| Parameters | | | | The message begins |
|  | Version | | |  |
| p\_result | | | | process result |
|  | provider\_list | | | DDNS provider's  information list |
|  | provider | |  |
|  | provider\_id | Provider id |
| provider\_name | Provider name |
| host\_name | Provider server domain name |

p\_result the json format is：

{

"provider\_list": [{

"provider\_id": 1,

"provider\_name": "3322\_ddns",

"host\_name": "members.3322.org"

}, {

"provider\_id": 2,

"provider\_name": "dyndns\_ddns",

"host\_name": "members.dyndns.org"

}, {

"provider\_id": 3,

"provider\_name": "no-ip\_ddns",

"host\_name": "dynupdate.no-ip.com"

}]

}

**Description**

#### Gets the DDNS provider parameters.

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_get\_ftp

#### Define

#### int sdks\_get\_ftp \_param(unsigned int handle, char\*\* p\_result);

#### parameter

handle：Return value of sdks\_dev\_conn interface；

p\_result: Output parameter, return FTP service information, represented by json format. If the returned value is not NULL, you need to call the free function to release the memory, otherwise it will cause a memory leak.

|  |  |  |  |
| --- | --- | --- | --- |
| parameter | Instruction | Y/N | Note |
| enable | Whether to enable ftp | Y | （1：enable，0：disable） |
| ser\_addr | FTP server address | Y |  |
| ser\_port | FTP server port | Y |  |
| username | FTP account | Y |  |
| password | FTP password | Y |  |
| workpath | FTP working path | Y | e.g.：eth0 |
| proto\_ver | protocol type | Y | protocol type: 1 or 2 |
| quality | Capture quality | Y | 1-9 |
| image\_num | Number of screenshots | Y |  |
| image\_int | Capture interval | Y |  |

p\_result the json format is：

{

"enable": 0,

"ser\_addr": "",

"ser\_port": 0,

"username": "",

"password": "",

"workpath": "",

"proto\_ver": 1,

"quality": 2,

"image\_num": 8,

"image\_int": 1

}

**Description**

#### Get FTP parameters.

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_set\_ftp

#### Define

int sdks\_set\_ftp \_param(unsigned int handle, char\* p\_ftp\_param);

#### parameter

handle：Return value of sdks\_dev\_conn interface；。

p\_ftp\_param：Input parameters, passed in as JSON format parameters.

|  |  |  |  |
| --- | --- | --- | --- |
| parameter | Instruction | Y/N | Note |
| enable | Whether to enable ftp | Y | （1：enable，0：disable） |
| ser\_addr | FTP server address | Y |  |
| ser\_port | FTP server port | Y |  |
| username | FTP account | Y |  |
| password | FTP password | Y |  |
| workpath | FTP working path | Y | e.g.：eth0 |
| proto\_ver | protocol type | Y | protocol type: 1 or 2 |
| quality | Capture quality | Y | 1-9 |
| image\_num | Number of screenshots | Y |  |
| image\_int | Capture interval | Y |  |

p\_ftp\_param the json format is：

{

"enable": 0,

"ser\_addr": "",

"ser\_port": 0,

"username": "",

"password": "",

"workpath": "",

"proto\_ver": 1,

"quality": 2,

"image\_num": 8,

"image\_int": 1

}

**Description**

#### Set FTP parameters.

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_get\_smtp

#### Define

int sdks\_dev\_get\_smtp(unsigned int handle,char \*\* p\_result);

#### parameter

handle：Return value of sdks\_dev\_conn interface；

p\_result: Output parameter, return smtp service information, represented by json format. If the returned value is not NULL, you need to call the free function to release the memory, otherwise it will cause a memory leak.

|  |  |  |  |
| --- | --- | --- | --- |
| parameter | Instruction | Y/N | Note |
| ser\_addr | SMTP server address | Y |  |
| ser\_port | SMTP server port | Y |  |
| username | Account | Y |  |
| password | password | Y |  |
| sender\_ddr | Sender address | Y |  |
| quality | Attachment image quality | Y |  |
| transmode | Transmission mode | Y |  |
| reci\_email | Recipient list | Y | fillin up to five recipients |

p\_result the json format is：

{

"ser\_addr": "",

"ser\_port": 25,

"username": "",

"password": "",

"sender\_ddr": "",

"quality": 2,

"transmode": 0,

"reci\_email": []

}

**Description**

#### Get SMTP parameters.

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_set\_smtp

#### Define

int sdks\_dev\_set\_smtp(unsigned int handle, char\* p\_net\_smtp);

#### parameter

handle：Return value of sdks\_dev\_conn interface；

p\_net\_smtp: Incoming parameters in json format；

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| parameter | | Instruction | Y/N | Note |
| ser\_addr | | SMTP server address | Y |  |
| ser\_port | | SMTP server port | Y |  |
| username | | Account | Y |  |
| password | | password | Y |  |
| sender\_ddr" | | Sender address | Y |  |
| quality | | Attachment image quality | Y | 1:high 2:middle 3：low |
| transmode | | Transmission mode | Y | 1 |
| reci\_email | | Recipient list | Y | fillin up to five recipients |
|  | Recipient one | Recipient one | Y |  |
| Recipient II | Recipient two | N |  |
| Recipient three | Recipient three | N |  |
|  |  | N |  |
|  |  | N |  |

p\_net\_smtp the json format is：

{

"ser\_addr": "",

"ser\_port": 25,

"username": "",

"password": "",

"sender\_ddr": "",

"quality": 2,

"transmode": 0,

"reci\_email": []

}

**Description**

#### Set the SMTP parameters.

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_get\_mtu

#### Define

sdks\_dev\_get\_mtu(unsigned int handle, int\* p\_mtu);

#### parameter

handle：Return value of sdks\_dev\_conn interface；。

p\_mtu: Obtained mtu

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_set\_mtu

#### Define

sdks\_dev\_set\_mtu(unsigned int handle, int mtu);

#### parameter

handle：Return value of sdks\_dev\_conn interface；。

mtu:set mtu

#### Return value

0 succeeded; other values failed.

### sdks\_get\_ip\_filter\_param

#### Define

int sdks\_get\_ip\_filter\_param(unsigned int handle, char\*\* p\_result)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

p\_result：Output the parameter, return the black and white list information, and return it in JSON format. If the returned value is not null, you need to call the free function to free memory, otherwise it will cause memory leak.

|  |  |  |  |
| --- | --- | --- | --- |
| parameter | Instruction | Y/N | Note |
| enable | enable | Y |  |
| mode | work mode | Y | (blacklist mode/whitelist mode)  1 black list 2 white list |
| begin | beginning Ip | Y |  |
| end | ending Ip | Y |  |
| des | Ip instruction |  |  |

p\_result the json format is：

{

"enable":1,

"mode":1,

"black\_list":[{"begin":"192.168.15.1","end":"192.168.15.2","des":"test0"},

{"begin":"192.168.15.3","end":"192.168.15.4","des":"test1"}],

"white\_list":[{"begin":"192.168.1.97","end":"192.168.1.98","des":"test3"},

{"begin":"192.168.0.97","end":"192.168.0.98","des":"test4"}

]

}

**Description**

#### Get black and white list information.

#### Return value

0 succeeded; other values failed.

### sdks\_set\_ip\_filter\_param

#### Define

int sdks\_set\_ip\_filter\_param(unsigned int handle, char\* p\_param)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

p\_param：Input parameters, expressed in json format.

|  |  |  |  |
| --- | --- | --- | --- |
| parameter | Instruction | Y/N | Note |
| enable | enable | Y |  |
| mode | work mode | Y | (blacklist mode/whitelist mode)  1 black list 2 white list |
| begin | beginning Ip | Y |  |
| end | ending Ip | Y |  |
| des | Ip instruction |  |  |

p\_ param the json format is：

{

"enable":1,

"mode":1,

"black\_list":[{"begin":"192.168.15.1","end":"192.168.15.2","des":"test0"},

{"begin":"192.168.15.3","end":"192.168.15.4","des":"test1"}],

"white\_list":[{"begin":"192.168.1.97","end":"192.168.1.98","des":"test3"},

{"begin":"192.168.0.97","end":"192.168.0.98","des":"test4"}

]

}

**Description**

Black list: up to 10 white lists can be set

White list: set up 10 blacklists at most

Set up a black and white list

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_get\_8021x

#### Define

#### int sdks\_dev\_get\_8021x(unsigned int handle, char \*\*p\_result)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

p\_result： Output parameters, return 802.1x information, and return in JSON format. If the returned value is not null, you need to call the free function to free memory, otherwise it will cause memory leak.

|  |  |  |  |
| --- | --- | --- | --- |
| Key field | Explain | Is it necessary | Remarks |
| EnableFlag | Is 802.1x authentication enabled | Y | (1: enabled, 0: not enabled) |
| Version | Version | Y |  |
| UserName | 802.1x account | Y |  |
| PassWord | 802.1x Password | Y |  |
| NetworkCardName | Bound network card name | Y |  |

p\_result the json format is：

{

"EnableFlag": 1,

"Version": 1,

"UserName": "admin",

"PassWord": "admin",

"NetworkCardName": "admin"

}

#### Define

#### Get 802.1x parameters.

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_set\_8021x

#### Define

#### int sdks\_dev\_set\_8021x(unsigned int handle, char \*p\_param)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

p\_ param：Input parameters, expressed in json format.

|  |  |  |  |
| --- | --- | --- | --- |
| Key field | Explain | Is it necessary | Remarks |
| EnableFlag | Is 802.1x authentication enabled | Y | (1: enabled, 0: not enabled) |
| Version | Version | Y |  |
| UserName | 802.1x account | Y |  |
| PassWord | 802.1x Password | Y |  |
| NetworkCardName | Bound network card name | Y |  |

p\_ param the json format is：

{

"EnableFlag": 1,

"Version": 1,

"UserName": "admin",

"PassWord": "admin",

"NetworkCardName": "admin"

}

**Description**

#### Set 802.1x parameters.

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_get\_pppoe

#### Define

#### int sdks\_dev\_get\_pppoe(unsigned int handle, char \*\*p\_result)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

p\_result： Output parameters, return PPPoE information, and return in JSON format. If the returned value is not null, you need to call the free function to free memory, otherwise it will cause memory leak.

|  |  |  |  |
| --- | --- | --- | --- |
| Key field | Explain | Is it necessary | Remarks |
| EnableFlag | Enable PPPoE identity or not | Y | (1: enabled, 0: not enabled) |
| UserName | PPPoE account | Y |  |
| PassWord | PPPoEP assword | Y |  |

p\_result the json format is：

{

"EnableFlag": 1,

" UserName ": "admin",

" PassWord ": "admin"

}

**Description**

#### Get the PPPoE parameter.

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_set\_pppoe

#### Define

#### int sdks\_dev\_set\_pppoe(unsigned int handle, char \*p\_param)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

p\_ param：Input parameters, expressed in json format.

|  |  |  |  |
| --- | --- | --- | --- |
| Key field | Explain | Is it necessary | Remarks |
| EnableFlag | Enable PPPoE identity or not | Y | (1: enabled, 0: not enabled) |
| UserName | PPPoE account | Y |  |
| PassWord | PPPoEP assword | Y |  |

p\_ param the json format is：

{

"EnableFlag": 1,

" UserName ": "admin",

" PassWord ": "admin"

}

**Description**

#### Set the PPPoE parameter.

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_get\_port\_mapping

#### Define

#### int sdks\_dev\_get\_port\_mapping(unsigned int handle, char \*\*p\_result)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

p\_result： Output parameters, return port mapping information, and return in JSON format. If the returned value is not null, you need to call the free function to free memory, otherwise it will cause memory leak.

|  |  |  |  |
| --- | --- | --- | --- |
| Key field | Explain | Is it necessary | Remarks |
| EnableFlag | Whether to enable port mapping identity | Y | (1: enabled, 0: not enabled) |
| MapMode | Mapping mode | Y | 1: automatic 2: Manual |
| PortName | Port name | Y |  |
| OutPort | External port |  |  |
| OutPortIP | External port IP |  |  |
| WorkingFlag | Status mark |  | (1: on, 0: off) |
| Enable | Enabling labelled |  | (1: enabled, 0: not enabled) |

p\_result the json format is：

{

"EnableFlag": 0,

"MapMode": 1,

"PortMappingList": [{

"PortName": "HTTP",

"OutPort": 80,

"OutPortIP": "0.0.0.0",

"WorkingFlag": 0,

"Enable": 1

}, {

"PortName": "RTSP",

"OutPort": 554,

"OutPortIP": "0.0.0.0",

"WorkingFlag": 0,

"Enable": 1

}, {

"PortName": "CONTROL",

"OutPort": 30001,

"OutPortIP": "0.0.0.0",

"WorkingFlag": 0,

"Enable": 1

}]

}

**Description**

#### Gets the port mapping parameters.

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_set\_port\_mapping

#### Define

#### int sdks\_dev\_set\_port\_mapping(unsigned int handle, char \*p\_param)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

p\_param：Input parameters, expressed in json format.

|  |  |  |  |
| --- | --- | --- | --- |
| Key field | Explain | Is it necessary | Remarks |
| EnableFlag | Whether to enable port mapping identity | Y | (1: enabled, 0: not enabled) |
| MapMode | Mapping mode | Y | 1: automatic 2: Manual |
| PortName | Port name | Y |  |
| OutPort | External port |  |  |
| OutPortIP | External port IP |  |  |
| WorkingFlag | Status mark |  | (1: on, 0: off) |
| Enable | Enabling labelled |  | (1: enabled, 0: not enabled) |

p\_ param the json format is：

{

"EnableFlag": 0,

"MapMode": 1,

"PortMappingList": [{

"PortName": "HTTP",

"OutPort": 80,

"OutPortIP": "0.0.0.0",

"WorkingFlag": 0,

"Enable": 1

}, {

"PortName": "RTSP",

"OutPort": 554,

"OutPortIP": "0.0.0.0",

"WorkingFlag": 0,

"Enable": 1

}, {

"PortName": "CONTROL",

"OutPort": 30001,

"OutPortIP": "0.0.0.0",

"WorkingFlag": 0,

"Enable": 1

}]

}

**Description**

#### Set port mapping parameters.

#### Return value

0 succeeded; other values failed.

## Front end image configuration

### sdks\_set\_sensor\_para

#### Define

int sdks\_set\_sensor\_para(unsigned int handle, char\* p\_sensor\_para);

#### parameter

handle：Return value of sdks\_dev\_conn interface.

p\_sensor\_para: Front-end image parameters in json format.

|  |  |  |
| --- | --- | --- |
| Attributes | | instruction |
| Parameters | |  |
| p\_sensor\_para | |  |
|  | channel\_id | channel ID |
| para | Front-end image parameters (ParamId, Value), you can set a single parameter or multiple parameters according to your needs. See Appendix 2.9. |

The JSON message body content example is as follows :

{"channel\_id":1,"para":"(49,8)(50,0)"}

#### Instruction

Front-end image parameter setting request, can set single parameter or multiple parameters according to requirements

#### Return value

0 succeeded; other values failed.

### sdks\_reset\_sensor\_to\_last\_param

#### Define

int sdks\_reset\_sensor\_to\_last\_param(unsigned int handle, int chn);

#### parameter

handle：Return value of sdks\_dev\_conn interface；。

chn：Channel NO.

#### Instruction

Front-end parameter configuration reset

#### Return value

0 succeeded; other values failed.

## System Maintenance

### sdks\_dev\_reboot

#### Define

int sdks\_dev\_reboot(unsigned int handle, int chn);

#### parameter

handle：Return value of sdks\_dev\_conn interface；。

chn：Channel NO.

#### Instruction

reboot device

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_reset

#### Define

int sdks\_dev\_reset(unsigned int handle, int chn);

#### parameter

handle：Return value of sdks\_dev\_conn interface；。

chn：Channel NO.

#### Instruction

reset to factory setting

#### Return value

0 succeeded; other values failed.

## user management

### sdks\_modify\_password\_param

#### Define

#### int sdks\_modify\_password\_param(unsigned int handle, char\* p\_param)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

p\_param：Input parameters, expressed in json format.

|  |  |  |  |
| --- | --- | --- | --- |
| Key field | Explain | Is it necessary | Remarks |
| NewPassword | New password | Y |  |
| OldPassword | Old password | Y |  |

p\_param the json format is：

{"NewPassword":"aaa111","OldPassword":"admin"}

**Description**

Change the device login password.

#### Return value

0 succeeded; other values failed.

### sdks\_operator\_privilege\_user

#### Define

int sdks\_operator\_privilege\_user(unsigned int handle, int chn, const char\* p\_user\_list, char\*\* p\_result);

#### parameter

handle：Return value of sdks\_dev\_conn interface；。

chn：Channel NO.

p\_user\_list: Input parameters, user information in json format, see the input parameter list for details.

p\_result：Output parameters, if and only if the operation type is to get all users, fill in the json type user parameter list, and when the operation type is increase, delete, modify the user parameters, it is empty.

#### Instruction

User permission operation

#### Return value

0 succeeded; other values failed.

#### Input parameter user information parameter table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | | | instruction |
| p\_user\_list | | | | | | User Info |
|  | type | | | | | 1Add user 2Delete user 3Modify user  4Get all users |
| camera\_id | | | | | camera ID |
| ptz\_id | | | | | PTZ ID |
| user\_list | | | | | User permission list |
|  | user | | | |  |
|  | user\_name | | | user name Rc4+Base64 encryped |
| password | | | password Rc4+Base64 encryped |
| group\_name | | | Add user group |
| lock\_flag | | | Lock flag |
| lock\_time | | | Lock time |
| multi\_login\_flag | | | Multi-login sign |
| desc | | | description |
| privilege\_list | | | Permission list |
|  | privilege | | privilege  9100000001: Video preview  9100000002: Video Control  9100000004: PTZ control  9100000008: Audio  9200000001: Video playback  9200000002: Backup  9200000004: Video Service Control  9200000008: Video strategy  9200000016: Disk Management  9300000001: Alarm retrieval  9400000001: Device Management  9400000002: Rights Management  9500000001: Self device parameter configuration  9500000002: Self system maintenance  9600000001: Log |
| description | | Permission description |
| has\_dev\_flag | | Does the group have device identification? |
| all\_dev\_flag | | All equipment |
| dev\_list | | Device List |
|  | Device ID\_channel number | 设备ID\_channel号 |

#### Output parameter user information parameter table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | | | instruction |
| p\_user\_list | | | | | | Result |
|  | type | | | | | 1Add user 2Delete user 3Modify user  4Get all users |
| camera\_id | | | | | camera ID |
| ptz\_id | | | | | PTZ ID |
| user\_list | | | | | User permission list |
|  | user | | | |  |
|  | user\_name | | | user name Rc4+Base64 encryped |
| password | | | password Rc4+Base64 encryped |
| group\_name | | | Add user group |
| lock\_flag | | | Lock flag |
| lock\_time | | | Lock time |
| multi\_login\_flag | | | Multi-login sign |
| desc | | | description |
| privilege\_list | | | Permission list |
|  | privilege | | privilege  9100000001: Video preview  9100000002: Video Control  9100000004: PTZ control  9100000008: Audio  9200000001: Video playback  9200000002: Backup  9200000004: Video Service Control  9200000008: Video strategy  9200000016: Disk Management  9300000001: Alarm retrieval  9400000001: Device Management  9400000002: Rights Management  9500000001: Self device parameter configuration  9500000002: Self system maintenance  9600000001: Log |
| description | | Permission description |
| has\_dev\_flag | | Does the group have device identification? |
| all\_dev\_flag | | All equipment |
| dev\_list | | Device List |
|  | Device ID\_channel number | 设备ID\_channel号 |

## Face

### sdks\_get\_face\_detect\_param(for IPC)

#### Define

int sdks\_get\_face\_detect\_param(unsigned int handle, int chn, char \*\*p\_result);

#### parameter

handle：Return value of sdks\_dev\_conn interface；。

chn：Channel NO.

p\_result： Output parameters, return face detection parameters, in JSON format. If the returned value is not null, you need to call the free function to free memory, otherwise it will cause memory leak.

|  |  |  |  |
| --- | --- | --- | --- |
| parameter | Explain | type | Remarks |
| FaceEnable | Whether to turn on face detection | Bool | True: on  Fasle: off |
| ShowObjectEnable | Whether to display face, human body, head and shoulder frame | Bool |  |
| UpBodyEnable | Whether upper body detection is turned on | Bool |  |
| FullBodyEnable | Whether to turn on the whole body test | Bool |  |
| FtpUploadEnable | Open FTP send matting or not | Bool |  |
| FtpUploadFullEnable | Open FTP to send panorama | Bool |  |
| Reliability | sensitivity | Int | [1,9] / / take 6 out of 9, take 3 out of 3 |
| MinPixelWidth | Face detection minimum pixel | Int | [30,300] |
| MaxPixelWidth | Face detection Max pixel | Int | [300,3000] |
| PictureQuality | Cutout quality | Int | [1,99] / / take 80 from high, take 60 from low, take 30 |
| UploadInterval | Send matting interval | Int | Unit second[1,10] |
| PitchDegree | Face elevation angle | Int | [0,90] |
| YawDegree | Left and right rotation angle | Int | [0,90] |
| TiltDegree | Left right angle | Int | [0,90] |
| SnapPictureMode | Capture mode | Int |  |
| SnapPictureNum | Capture number | Int |  |
| FirmwareVer | Version number | Int |  |
| ConsumePictureMode | Capture picture consumption mode | Int |  |
| LYFaceEnable |  | Bool |  |
| weekday | (defense time list) week | Int |  |
| sTime | (deployment time list) start time | Int | second[0，86400] |
| eTime | (deployment time list) end time | Int | second[0，86400] |
| Xval | (detection area (polygon)) x coordinate | double |  |
| Yval | (detection area (polygon)) y coordinate | double |  |

p\_result json format is：

{

"FaceEnable":true,

"ShowObjectEnable":false,

"UpBodyEnable":true,

"FullBodyEnable":true,

"FtpUploadEnable":false,

"FtpUploadFullEnable":false,

"Reliability":60,

"MinPixelWidth":80,

"MaxPixelWidth":1000,

"PictureQuality":60,

"UploadInterval":5,

"PitchDegree":0,

"YawDegree":60,

"TiltDegree":60,

"SnapPictureMode":0,

"SnapPictureNum":1,

"FirmwareVer":0,

"ConsumePictureMode":3,

"LYFaceEnable":false,

"timelist":[

{"weekday":0,"sTime":0,"eTime":86400},

{"weekday":1,"sTime":0,"eTime":86400},

{"weekday":2,"sTime":0,"eTime":86400},

{"weekday":3,"sTime":0,"eTime":86400},

{"weekday":4,"sTime":0,"eTime":86400},

{"weekday":5,"sTime":0,"eTime":86400},

{"weekday":6,"sTime":0,"eTime":86400}],

"polygonarealist":[[

{"Xval":0,"Yval":0},

{"Xval":0,"Yval":99.900002},

{"Xval":99.900002,"Yval":99.900002},

{"Xval":99.900002,"Yval":0}]]

}

#### Instruction

Get face detection parameters

#### Return value

0 succeeded; other values failed.

### sdks\_set\_face\_detect\_param(for IPC)

#### Define

int sdks\_set\_face\_detect\_param(unsigned int handle, int chn, char \*p\_param);

#### parameter

handle：Return value of sdks\_dev\_conn interface；。

chn：Channel NO.

p\_ param：Input parameters, passed in as JSON format parameters.

|  |  |  |  |
| --- | --- | --- | --- |
| parameter | Explain | type | Remarks |
| FaceEnable | Whether to turn on face detection | Bool | True: on  Fasle: off |
| ShowObjectEnable | Whether to display face, human body, head and shoulder frame | Bool |  |
| UpBodyEnable | Whether upper body detection is turned on | Bool |  |
| FullBodyEnable | Whether to turn on the whole body test | Bool |  |
| FtpUploadEnable | Open FTP send matting or not | Bool |  |
| FtpUploadFullEnable | Open FTP to send panorama | Bool |  |
| Reliability | sensitivity | Int | [1,9] / / take 6 out of 9, take 3 out of 3 |
| MinPixelWidth | Face detection minimum pixel | Int | [30,300] |
| MaxPixelWidth | Face detection Max pixel | Int | [300,3000] |
| PictureQuality | Cutout quality | Int | [1,99] / / take 80 from high, take 60 from low, take 30 |
| UploadInterval | Send matting interval | Int | Unit second[1,10] |
| PitchDegree | Face elevation angle | Int | [0,90] |
| YawDegree | Left and right rotation angle | Int | [0,90] |
| TiltDegree | Left right angle | Int | [0,90] |
| SnapPictureMode | Capture mode | Int |  |
| SnapPictureNum | Capture number | Int |  |
| FirmwareVer | Version number | Int |  |
| ConsumePictureMode | Capture picture consumption mode | Int |  |
| LYFaceEnable |  | Bool |  |
| weekday | (defense time list) week | Int |  |
| sTime | (deployment time list) start time | Int | second[0，86400] |
| eTime | (deployment time list) end time | Int | second[0，86400] |
| Xval | (detection area (polygon)) x coordinate | double |  |
| Yval | (detection area (polygon)) y coordinate | double |  |

p\_ param json format is：

{

"FaceEnable":"true",

"ShowObjectEnable":"true",

"UpBodyEnable":"false",

"FullBodyEnable":"false",

"FtpUploadEnable":"false",

"FtpUploadFullEnable":"false",

"Reliability":60,

"MinPixelWidth":80,

"MaxPixelWidth":-99999,

"PictureQuality":1,

"PitchDegree":-99999,

"YawDegree":60,

"TiltDegree":60,

"SnapPictureMode":0,

"SnapPictureNum":-99999,

"FirmwareVer":0,

"UploadInterval":1,

"ConsumePictureMode":0,

"LYFaceEnable":"false",

"timelist":[

{"sTime":0,"eTime":86400,"weekday":0},

{"sTime":0,"eTime":86400,"weekday":1},

{"sTime":0,"eTime":86400,"weekday":2},

{"sTime":0,"eTime":86400,"weekday":3},

{"sTime":0,"eTime":86400,"weekday":4},

{"sTime":0,"eTime":86400,"weekday":5},

{"sTime":0,"eTime":86400,"weekday":6}],

"polygonarealist":[[

{"Xval":10,"Yval":10},

{"Xval":60,"Yval":10},

{"Xval":60,"Yval":60},

{"Xval":10,"Yval":60}]]

}

#### Instruction

Set face detection parameters

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_face\_detect\_start(for IPC)

#### Define

int sdks\_dev\_face\_detect\_start(unsigned int handle, int chn, int stream\_type, int type, SDK\_DETECT\_CB detect\_cb, void\* p\_obj);

#### parameter

handle：Return value of sdks\_dev\_conn interface；。

chn：Channel NO.

stream\_type:code stream

Type: Picture type 4: Small picture. 5: Big picture. 7: size chart

SDK\_DETECT\_CB: Callback function, get this function will be detected when the face image is detected. See the SDK\_DETECT\_CB for the callback function description.

**Description**

#### Start face or license plate detection

#### Return value

>0 success; other values failed.

### sdks\_dev\_face\_detect\_stop(for IPC)

#### Define

int sdks\_dev\_face\_detect\_stop(unsigned int handle, int stream\_id);

#### parameter

handle：Return value of sdks\_dev\_conn interface.

stream\_type: Return value of sdks\_dev\_face\_detect\_start interface.

**Description**

#### Stop face detection.

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_face\_get\_group\_num(for NVR)

#### int sdks\_dev\_face\_get\_group\_num(unsigned int handle, int chn, char \*\*p\_result)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number, IPC fill-1.

p\_result：Output parameters, return the number of face attendance library members obtained, expressed in json format. If the returned value is non-NULL, you need to call the sdks\_free\_result function to free the memory, otherwise it will cause a memory leak.

|  |  |  |  |
| --- | --- | --- | --- |
| parameter | Description | Types | Remarks |
| Name | Attendance library name | String |  |
| MemberNum | Total number of attendance library members | Integer |  |

p\_result the json format is：

{

"AttendanceGroupList": [{

"Name": "cxy",

"MemberNum": 1

}]

}

**Description**

#### Get the number of face attendance library members.

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_face\_get\_member(for NVR)

#### int sdks\_dev\_face\_get\_member(unsigned int handle, int chn, char \*p\_param,char \*\*p\_result); parameter

handle: Return value of sdks\_dev\_conn interface；

chn： Channel NO；

p\_param：Input parameters，Expressed in json format。

p\_result：Output parameters，Returns the obtained face attendance library member information, expressed in json format. If the returned value is non-NULL, you need to call the sdks\_free\_result function to free the memory, otherwise it will cause a memory leak.

p\_param: json format is：

{

"AttendanceGroupList":[{

"Name": "cxy"

}……

],

"TotalNum": 3,

"PerPage": 10,

"Page": 1

}

p\_result: json format is：

{

"AttendanceMemberList": [{

"Key": 7,

"MemberId": "",

"Name": "3",

"Group": "cxy"

}, {

"Key": 5,

"MemberId": "",

"Name": "1",

"Group": "cxy"

}, {

"Key": 6,

"MemberId": "",

"Name": "2",

"Group": "cxy"

}……

]

}

|  |  |  |  |
| --- | --- | --- | --- |
| parameter | Description | Types | Remarks |
| Name | Attendance library name | String |  |
| TotalNum | The total number of all attendance library members queried | Integer |  |
| PerPage | Number of pages per page | Integer | Query up to 250 at a time |
| Page | Take the nth page of data | Integer | Starting from 1 |
| Key | Attendance library member unique index | Integer |  |
| MemberId | Attendance library member Id (work number) | String |  |
| Name | Attendance library member name | String |  |
| Group | Attendance library member attendance library | String |  |

**Description**

#### (by page) to get face attendance library member information.

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_face\_check\_data(for NVR)

#### int sdks\_dev\_face\_check\_data(unsigned int handle, int chn, char \*p\_param, char \*\*p\_result);

#### parameter

handle: Return value of sdks\_dev\_conn interface；

chn： Channel NO；

p\_param：Input parameters，Expressed in json format。

p\_result：Output parameters，Returns the Query face attendance data, expressed in json format. If the returned value is non-NULL, you need to call the sdks\_free\_result function to free the memory, otherwise it will cause a memory leak.

p\_param: json format is：

{

"Key": "[1]",

"StartTime": "1562238629",

"EndTime": "1562238629",

"Type": 1

}

p\_result: json format is：

{

"TotalNum": 1

}

|  |  |  |  |
| --- | --- | --- | --- |
| parameter | Description | Types | Remarks |
| Key | Query attendance member index list | Integer  Array | Up to 250 |
| StartTime | Query start time | Integer | UTC time of day 00:00:00, base  At device time, in seconds |
| EndTime | Query end time | Integer | UTC time of day 23:59:59, base  At device time, in seconds |
| Type | Query attendance data type | Integer | 0 (attendance statistics),  1 (attendance details) |
| TotalNum | Total number of queries | Integer |  |

**Description**

#### Query face attendance data.

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_face\_get\_statis(for NVR)

#### int sdks\_dev\_face\_get\_statis(unsigned int handle, int chn, char \*p\_param, char \*\*p\_result); parameter

handle: Return value of sdks\_dev\_conn interface；

chn： Channel NO；

p\_param： Input parameters，Expressed in json format。

p\_result： Output parameters，Returns the (by page) Get / Export Face Attendance Data, expressed in json format. If the returned value is non-NULL, you need to call the sdks\_free\_result function to free the memory, otherwise it will cause a memory leak.

(If the returned type type is 0, it means returning attendance statistics, if it is 1, it means returning attendance details).

p\_param json format is：{

"TotalNum": 3,

"PerPage": 10,

"Page": 1

}

p\_result json format is：

{

"AttendanceData": {

"Type": 1,

"AttendanceDetailDataList": [{

"Key": 1,

"Date": "1562198400",

"CheckinTime": "",

"CheckoutTime": "",

"State": 5

}……

]

}

}

or：

{

"AttendanceData": {

"Type": 0,

"AttendanceStatisDataList": [{

"Key": 1,

"Expect": 1,

"Checkin": 0,

"Absence": 1,

"Late": 0,

"Early": 0

}……

]

}

}

|  |  |  |  |
| --- | --- | --- | --- |
| parameter | Description | Types | Remarks |
| TotalNum | Total attendance data | Integer |  |
| PerPage | Number of pages per page | Integer | Query up to 250 at a time |
| Page | Take the nth page of data | Integer | Starting from 1 |
| Type | Query attendance data type | Integer | 0 (attendance statistics),  1 (attendance details) |
| Key | Attendance library member unique index | Integer |  |
| Expect | Should be working days | Integer |  |
| Checkin | Actual working days | Integer |  |
| Absence | Number of completions | Integer |  |
| Late | Late arrivals | Integer |  |
| Early | Early retreat | Integer |  |
| Date | date | Integer | UTC time of day 00:00:00, based on time,Unit s |
| CheckinTime | Check-in time | Integer | 30600(s)=08:30(h)  - 00:00(h), [0,86399], a number outside the range indicates that there is no punch record |
| CheckoutTime | Signing time | Integer | 64800(s)=18:00(h) - 00:00(h) |
| State | Attendance status | Integer | 1 (normal),  2 (late),  3 (early leave),  4 (late and early leave),  5 (complete),  6 (overtime) |

**Description**

#### Get/export face attendance data (by page).

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_face\_get\_attendance\_data(for NVR)

#### int sdks\_dev\_face\_check\_data(unsigned int handle, int chn, char \*p\_param, char \*path\_file) parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number, IPC fill-1.

p\_param：Input parameters, expressed in json format.

path\_file：The txt file name of the attendance data to be saved (including the path, cannot be a Chinese path).

p\_param the json format is：

{

"Group": "cxy",

"MemberNum": 18,

"StartTime": "1569772800",

"EndTime": "1570032000",

"Type": 1

}

|  |  |  |  |
| --- | --- | --- | --- |
| parameter | Description | Types | Remarks |
| MemberNum | Total number of attendance library members | Integer |  |
| Group | Attendance library name | String |  |
| StartTime | Starting time | String |  |
| EndTime | End Time | String |  |
| Type | Search type | Integer |  |

**Description**

Obtain all face attendance data, the data is saved in the txt file, and the data content can be explained by referring to the table parameters.

#### Return value

0 succeeded; other values failed.

### sdks\_start\_database(for NVR)

#### Define

#### int sdks\_start\_database(unsigned int handle, SDK\_FACEBASE\_CB facebase\_cb, void \*p\_obj) parameter

handle：Return value of sdks\_dev\_conn interface.

SDK\_FACEBASE\_CB: Callback function, Get the real-time library image information of the detected face, callback function description see SDK\_FACEBASE\_CB.

P\_obj: Callback function context.

**Description**

#### Open the face real-time library image information channel, and only after calling this interface can the real-time library image operation be started.

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_get\_database\_index(for NVR)

#### int sdks\_dev\_get\_database\_index(unsigned int handle, int chn, char \*p\_buf,int size,int size, char \*param,int \*p\_task,char \*\*p\_result);

#### parameter

handle: Return value of sdks\_dev\_conn interface；

chn： Channel NO；

face\_data: Face image data (search\_type parameter type in p\_param is 1 when the feature query is filled in)；

size： Image size (in k), the search\_type parameter type in p\_param is 1 and the feature query is filled with 0);

p\_param： Input parameters, expressed in json format;

p\_task: Input and output parameters, query task ID of the device, fill in 0 for the first query, and fill in the value of the first query when turning the page.

p\_result： Output parameters, Returns the face database of the query, expressed in json format. If the returned value is non-NULL, you need to call the sdks\_free\_result function to free the memory, otherwise it will cause a memory leak.

p\_param json format is：

{

"channel\_num": 1,

"channel\_list": [{

"chn": 1

}……

],

"s\_time": 1565860853,

"e\_time": 1566379253,

"search\_type": 1,

"similarity": 80,

"id": "1",

"name": "1",

"gender": 3

}

p\_result json format is：

{

"IndexList": [{

"HdId": 0,

"HdPos": 478875951,

"Offset": 53667,

"BlockLen": 110828,

"PictureLen": 57161,

"ID": 10,

"Similarity": -1

}, {

"HdId": 0,

"HdPos": 478875990,

"Offset": 0,

"BlockLen": 55851,

"PictureLen": 55851,

"ID": 12,

"Similarity": -1

}, {

"HdId": 0,

"HdPos": 478875990,

"Offset": 55915,

"BlockLen": 100698,

"PictureLen": 44783,

"ID": 13,

"Similarity": -1

}]

}

|  |  |  |  |
| --- | --- | --- | --- |
| parameter | Description | Types | Remarks |
| channel\_num | Number of channels | Integer |  |
| channel\_list | Channel array | Integer array | Up to 8 channels at a time |
| s\_time | Search range start time | Integer | UTC time |
| e\_time | Search range end time | Integer | UTC time |
| search\_type | Search type | Integer | 0 (search by image),  1 (search by feature) |
| similarity | Similarity | Integer | (Required when searching by image), the value is 0~100, you can enter the similarity you want to query. |
| id | Feature id | String | (Required when searching by feature) Feature id to be queried, up to 32 bytes |
| name | Feature name | String | (Required when searching by feature) Feature name to be queried, up to 32 bytes |
| gender | Characteristic gender | Integer | (Required when searching by feature) 1 (male), 2 (female), 3 (all) |
| HdId | The disk ID of the real-time library image | Integer |  |
| HdPos | The location of the disk where the live library image is located | Integer |  |
| Offset | The disk location of the live library image is offset | Integer |  |
| BlockLen | The length of the disk block where the real-time library image is located | Integer |  |
| PictureLen | Real-time library image length | Integer |  |
| ID | Real-time library image ID | Integer |  |
| Similarity | Similarity | Integer | Similarity |

**Description**

#### Index of conditional query face database (real-time database, bottom database)

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_get\_database\_info(for NVR)

#### int sdks\_dev\_get\_database\_info(unsigned int handle, char \*param)

#### parameter

handle: Return value of sdks\_dev\_conn interface；

p\_param：The input parameter is represented by the json format (this parameter is the output parameter content of the sdks\_dev\_get\_database\_index interface, indicating that the corresponding image information is obtained by using the obtained index query, and the number of lists must be less than 64 when inputting, if the number of output image index lists is greater than 64 Then group query);

p\_param json format is：

{

"IndexList": [{

"HdId": 0,

"HdPos": 478875951,

"Offset": 53667,

"BlockLen": 110828,

"PictureLen": 57161,

"ID": 10,

"Similarity": -1

}, {

"HdId": 0,

"HdPos": 478875990,

"Offset": 0,

"BlockLen": 55851,

"PictureLen": 55851,

"ID": 12,

"Similarity": -1

}, {

"HdId": 0,

"HdPos": 478875990,

"Offset": 55915,

"BlockLen": 100698,

"PictureLen": 44783,

"ID": 13,

"Similarity": -1

}]

}

|  |  |  |  |
| --- | --- | --- | --- |
| parameter | Description | Types | Remarks |
| HdId | The disk ID of the real-time library image | Integer |  |
| HdPos | The location of the disk where the live library image is located | Integer |  |
| Offset | The disk location of the live library image is offset | Integer |  |
| BlockLen | The length of the disk block where the real-time library image is located | Integer |  |
| PictureLen | Real-time library image length | Integer |  |
| ID | Real-time library image ID | Integer |  |
| Similarity | Similarity | Integer |  |

**Description**

#### According to the index, the real-time library corresponding to the face image is searched. After calling this interface, the SDK\_FACEBASE\_CB callback function will export relevant information.

#### Return value

0 succeeded; other values failed.

### sdks\_stop\_database(for NVR)

#### Define

#### int sdks\_stop\_database(unsigned int handle)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

**Description**

#### Close the face real-time library picture information channel.

#### Return value

0 succeeded; other values failed.

### sdks\_start\_face(for NVR)

#### Define

#### int sdks\_start\_face(unsigned int handle, SDK\_FACE\_CB face\_cb, void \*p\_obj)

#### parameter

handle：Return value of sdks\_dev\_conn interface。

SDK\_FACE\_CB: Callback function, get the bottom library picture information, callback function description see SDK\_FACE\_CB.

P\_obj: Callback function context.

**Description**

#### Open the face image information channel.

#### Return value

0 succeeded; other values failed.

### sdks\_stop\_face(for NVR)

#### Define

#### int sdks\_stop\_face(unsigned int handle)

#### parameter

handle：Return value of sdks\_dev\_conn interface。

**Description**

#### Close the face image information channel.

#### Return value

0 succeeded; other values failed.

### sdks\_face\_get\_group(for NVR)

#### Define

#### int sdks\_face\_get\_group(unsigned int handle, char\*\* p\_result)

#### parameter

handle：Return value of sdks\_dev\_conn interface。

p\_result：Output parameters, return the obtained face database information, expressed in json format. If the returned value is non-NULL, you need to call the sdks\_free\_result function to free the memory, otherwise it will cause a memory leak.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | Types | Optional / required | description |
| Parameters | | | | E | M | The message begins |
|  | Version | | | A | M | Version information 1.0 |
| Result | | | E | M |  |
| GroupInfoList | | | E | M |  |
|  | GroupInfo | | E | O | Face database information |
|  | Name | A | M | Face library name |
| Num | A | M | Number of entries in the face database |

p\_result the json format is：

[{

"name": "cxy",

"num": 5

}]

**Description**

#### Get the face database information.

#### Return value

0 succeeded; other values failed.

### sdks\_get\_face\_all\_node(for NVR)

#### Define

#### int sdks\_get\_face\_all\_node(unsigned int handle,char \*p\_param, char \*\*result)

#### parameter

handle：Return value of sdks\_dev\_conn interface。

p\_param：Input parameters, expressed in json format (face database information).

p\_result：Output parameters, return the obtained face database index information, expressed in json format. If the returned value is non-NULL, you need to call the sdks\_free\_result function to free the memory, otherwise it will cause a memory leak.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | | Types | Optional / required | description |
| Parameters | | | | | E | M | The message begins |
|  | Version | | | | A | M | Version information 1.0 |
| DataChannelId | | | | E | O |  |
|  | | | Id | A | M |  |
| GroupInfoList | | | | E | M |  |
|  | GroupInfo | | | E | M |  |
|  | Name | | A | M | Face library name |

p\_ param the json format is：

[{

"name": "cxy",

"num": 5

}]

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | | Types | Optional / required | description |
| Parameters | | | | | E | M | The message begins |
|  | Version | | | | A | M | Version information 1.0 |
| Result | | | | E | M |  |
|  | Code | | | A | M | Get the result (Code):  0: OK  For other errors, please refer to the front end error code table. |
| IndexList | | | | E | M |  |
|  | | Index | | E | O | Face database image index information |
|  | Key | A | M | Face library image Key |
| KeyId | A | M | Face Library Picture KeyId |
| Group | A | M | The name of the library where the face library image is located |
| HdId | A | M | Disk ID of the face library image |
| PictureLen | A | M | Face library picture length |

p\_result the json format is：

[{

"key": "1566865299\_",

"key\_id": 54,

"hd\_id": 0,

"pic\_size": 48674,

"group\_name": ""

}, {

"key": "1566866308\_",

"key\_id": 78,

"hd\_id": 0,

"pic\_size": 48674,

"group\_name": ""

}, {

"key": "1569600150\_",

"key\_id": 306,

"hd\_id": 0,

"pic\_size": 19663,

"group\_name": ""

}, {

"key": "1569600474\_",

"key\_id": 307,

"hd\_id": 0,

"pic\_size": 19663,

"group\_name": ""

}, {

"key": "1569614632\_",

"key\_id": 325,

"hd\_id": 0,

"pic\_size": 19663,

"group\_name": ""

}]

**Description**

#### Get the index of the face database.

#### Return value

0 succeeded; other values failed.

### sdks\_get\_face\_by\_node(for NVR)

#### Define

#### int sdks\_get\_face\_by\_node(unsigned int handle, char \*p\_info)

#### parameter

handle：Return value of sdks\_dev\_conn interface。

p\_info：Input parameters, expressed in json format (face database index information).

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | | Types | Optional / required | description |
| Parameters | | | | | E | M | The message begins |
|  | Version | | | | A | M | Version information 1.0 |
| Result | | | | E | M |  |
|  | Code | | | A | M | Get the result (Code):  0: OK  For other errors, please refer to the front end error code table. |
| IndexList | | | | E | M |  |
|  | | Index | | E | O | Face database image index information |
|  | Key | A | M | Face library image Key |
| KeyId | A | M | Face Library Picture KeyId |
| Group | A | M | The name of the library where the face library image is located |
| HdId | A | M | Disk ID of the face library image |
| PictureLen | A | M | Face library picture length |

p\_info the json format is：

[{

"key": "1566865299\_",

"key\_id": 54,

"hd\_id": 0,

"pic\_size": 48674,

"group\_name": ""

}, {

"key": "1566866308\_",

"key\_id": 78,

"hd\_id": 0,

"pic\_size": 48674,

"group\_name": ""

}, {

"key": "1569600150\_",

"key\_id": 306,

"hd\_id": 0,

"pic\_size": 19663,

"group\_name": ""

}, {

"key": "1569600474\_",

"key\_id": 307,

"hd\_id": 0,

"pic\_size": 19663,

"group\_name": ""

}, {

"key": "1569614632\_",

"key\_id": 325,

"hd\_id": 0,

"pic\_size": 19663,

"group\_name": ""

}]

**Description**

#### Export image information through the face database index information (export data can be viewed in the SDK\_FACE\_CB callback function).

#### Return value

0 succeeded; other values failed.

### sdks\_face\_add\_group(for NVR)

#### Define

#### int sdks\_face\_add\_group(unsigned int handle, char \*p\_db\_info)

#### parameter

handle：Return value of sdks\_dev\_conn interface。

p\_db\_info：Input parameters, expressed in json format.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | Types | Optional / required | description |
| Parameters | | | | E | M | The message begins |
|  | Version | | | A | M | Version information 1.0 |
| GroupInfoList | | | E | M |  |
|  | GroupInfo | | E | M | Face database information |
|  | Name | A | M | Face library name |

p\_db\_info the json format is：

[{

"name": "area\_0"

}]

**Description**

#### Add a face library name.

#### Return value

0 succeeded; other values failed.

### sdks\_face\_rename\_group(for NVR)

#### Define

#### int sdks\_face\_rename\_group(unsigned int handle, char \*p\_db\_info)

#### parameter

handle：Return value of sdks\_dev\_conn interface。

p\_db\_info：Input parameters, expressed in json format.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | Types | Optional / required | description |
| Parameters | | | | E | M | The message begins |
|  | Version | | | A | M | Version information 1.0 |
| GroupInfoList | | | E | M |  |
|  | GroupInfo | | E | M | Face database information |
|  | new | A | M | New face library name |
| old | A | M | Old face library name |

p\_db\_info the json format is：

[{

"new": "sunell",

"old": "area\_0"

}]

**Description**

#### Modify the face database name.

#### Return value

0 succeeded; other values failed.

### sdks\_face\_del\_group(for NVR)

#### Define

#### int sdks\_face\_del\_group(unsigned int handle, char \*p\_db\_info)

#### parameter

handle：Return value of sdks\_dev\_conn interface。

p\_db\_info：Input parameters, expressed in json format.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | Types | Optional / required | description |
| Parameters | | | | E | M | The message begins |
|  | Version | | | A | M | Version information 1.0 |
| GroupInfoList | | | E | M |  |
|  | GroupInfo | | E | M | Face database information |
|  | Name | A | M | Face library name |

p\_db\_info the json format is：

[{

"name": "sunell"

}]

**Description**

#### Delete the face library name.

#### Return value

0 succeeded; other values failed.

### sdks\_face\_get\_group\_type(for NVR)

#### Define

#### int sdks\_face\_get\_group\_type(unsigned int handle, char\*\* p\_result)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

p\_result：Output parameters, return the obtained face database type information, expressed in json format. If the returned value is non-NULL, you need to call the sdks\_free\_result function to free the memory, otherwise it will cause a memory leak.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | Types | Optional / required | description |
| Parameters | | | | E | M | The message begins |
|  | Version | | | A | M | Version information 1.0 |
| Result | | | E | M |  |
| TypeInfoList | | | E | M |  |
|  | TypeInfo | | E | O | Face library type information |
|  | Name | A | M | Face library type name |
| Num | A | O | Number of entries associated with the face database type |

p\_result the json format is：

[{

"type\_name": "Teacher",

"num": 1

}, {

"type\_name": "Student",

"num": 4

}]

**Description**

Get the face database type information.

#### Return value

0 succeeded; other values failed.

### sdks\_face\_add\_group\_type(for NVR)

#### Define

#### int sdks\_face\_add\_group\_type(unsigned int handle, char \*p\_db\_info)

#### parameter

handle：Return value of sdks\_dev\_conn interface。

p\_db\_info：Input parameters, expressed in json format.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | Types | Optional / required | description |
| Parameters | | | | E | M | The message begins |
|  | Version | | | A | M | Version information 1.0 |
| TypeInfoList | | | E | M |  |
|  | TypeInfo | | E | M | Face library type information |
|  | Name | A | M | Face library type name |

p\_db\_info the json format is：

[{

"name": "footboot"

}]

**Description**

#### Add a face database type.

#### Return value

0 succeeded; other values failed.

### sdks\_face\_del\_group\_type(for NVR)

#### Define

#### int sdks\_face\_del\_group\_type(unsigned int handle, char \*p\_db\_info)

#### parameter

handle：Return value of sdks\_dev\_conn interface。

p\_db\_info：Input parameters, expressed in json format.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | Types | Optional / required | description |
| Parameters | | | | E | M | The message begins |
|  | Version | | | A | M | Version information 1.0 |
| TypeInfoList | | | E | M |  |
|  | TypeInfo | | E | M | Face library type information |
|  | Name | A | M | Face library type name |

p\_db\_info the json format is：

[{

"name": "footboot"

}]

**Description**

#### Delete the face database type.

#### Return value

0 succeeded; other values failed.

### sdks\_add\_face\_data\_to\_group（for NVR）

#### Define

#### int sdks\_add\_face\_data\_to\_group(unsigned int handle, char \*p\_param, char \*pic\_data, int pic\_size)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

p\_param： Input parameters, expressed in json format.

pic\_data: Input parameters, picture data.

pic\_size: Input parameters, image data size, in K.

|  |  |  |  |
| --- | --- | --- | --- |
| Key field | Explain | Is it necessary | Remarks |
| name | Name | Y |  |
| id | Characteristic ID | Y | Uniquely identify each face image |
| gender | Gender | Y |  |
| birthday | Birthday |  | UTC timestamp in seconds |
| group | Affiliated group |  |  |
| type | Identity type |  |  |
| s\_time | Effective start time |  | UTC timestamp, in seconds, 0 for permanent |
| e\_time | Effective end time |  | UTC timestamp, in seconds, 0 for permanent |

p\_param the json format is：

{

"name": "pic\_three",

"id": "666",

"gender": 1,

"birthday": 1577165768,

"group": "rrrttt11",

"type": "Teacher",

"s\_time": 0,

"e\_time": 0

}

**Description**

#### Add face image to face database.

#### Return value

0 succeeded; other values failed.

### sdks\_del\_face\_data（for NVR）

#### Define

#### int sdks\_del\_face\_data(unsigned int handle,char \*p\_param)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

p\_param：Input parameters, expressed in json format.

|  |  |  |  |
| --- | --- | --- | --- |
| Key field | Explain | Is it necessary | Remarks |
| group | Affiliated group |  |  |
| key\_id | Face database picture key |  | which is returned by the sdks\_get\_face\_all\_node. |
| key | Face database picture keyID |  | which is returned by the sdks\_get\_face\_all\_node. |
| hd\_id | Disk ID of face database image |  | The disk ID where the picture is located, which is returned by the sdks\_get\_face\_all\_node. |

p\_param the json format is：

[{

"group": "rrrttt11",

"key\_id": 831796,

"key": "",

"hd\_id": 0

}]

**Description**

#### Delete face image of face database.

#### Return value

0 succeeded; other values failed.

### sdks\_mod\_face\_data（for NVR）

#### Define

#### int sdks\_mod\_face\_data(unsigned int handle, char \*p\_param, char \*pic\_data, int pic\_size) parameter

handle: Return value of sdks\_dev\_conn interface.

p\_param： Input parameters, expressed in json format.

pic\_data: Input parameters, picture data.

pic\_size: Input parameters, image data size, in K.

|  |  |  |  |
| --- | --- | --- | --- |
| Key field | Explain | Is it necessary | Remarks |
| key\_id | Face database picture key |  | which is returned by the sdks\_get\_face\_all\_node. |
| key | Face database picture keyID |  | which is returned by the sdks\_get\_face\_all\_node. |
| hd\_id | Disk ID of face database image |  | The disk ID where the picture is located, which is returned by the sdks\_get\_face\_all\_node. |
| name | Name | Y |  |
| id | Characteristic ID | Y | Uniquely identify each face image |
| gender | Gender | Y |  |
| birthday | Birthday |  | UTC timestamp in seconds |
| group | Affiliated group |  |  |
| type | Identity type |  |  |
| s\_time | Effective start time |  | UTC timestamp, in seconds, 0 for permanent |
| e\_time | Effective end time |  | UTC timestamp, in seconds, 0 for permanent |

p\_param the json format is：

{

"key\_id": 831796,

"key": "",

"hd\_id": 0,

"name": "llllfffffffff",

"id": "66666",

"gender": 2,

"birthday": 1568947901,

"group": "rrrttt11",

"type": "Teacher",

"s\_time": 0,

"e\_time": 0

}

**Description**

#### Modify the image of the face database.

#### Return value

0 succeeded; other values failed.

## Other function

### sdks\_get\_device\_log

#### Define

#### int sdks\_get\_device\_log(unsigned int handle, char \*p\_param, char \*\*p\_result)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

p\_param：Input parameters, expressed in json format.

p\_result：Output parameters, return the obtained device log information, expressed in json format. If the returned value is non-NULL, you need to call the sdks\_free\_result function to free the memory, otherwise it will cause a memory leak. (The returned log information is output as a list in 100 logs. When there are more logs, the interface may be waiting for a few seconds. Please be patient.)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | Types | Optional / required | description |
| Parameters | | | | E | M |  |
| Version | | | | A | M |  |
|  | QueryInfo | | | E | M |  |
|  | RecordMode | | A | M | Query mode, set to 4 |
| DeviceId | | A | M | Device ID |
| ChannelId | | A | M | Channel number, IPC is set to -1, NVR is set to 0 |
| SelectMode | | A | M | Query mode, set to 2 |
| MajorType | | A | M | Main type, set to -1 |
| MinorType | | A | M | Subtype, set to -1 |
| Precision | | A | M | Precision, set to 0 |
| RecordSegmentInterval | | A | M | Query segment length, set to 0 |
| BeginTime | | E | M | Start time of the log, in the format YY-MM-DD HH:MM:SS:MM |
| EndTime | | E | M | End time of the log, in the format YY-MM-DD HH:MM:SS:MM |
| QueryInfoType | | E | M |  |
|  | Type | A | M | Log Type 4: System Log 3: Alarm Log |

p\_param the json format is：

{

"DeviceId": "307807A7F03D3DF",

"ChannelId": 0,

"SelectMode": 2,

"RecordMode": 4,

"MajorType": -1,

"MinorType": -1,

"Precision": 0,

"RecordSegmentInterval": 0,

"Type": 4,

"BeginTime": "2019-08-06 00:00:00:00",

"EndTime": "2019-10-15 00:00:00:00"

}

The system log parameter information is as follows:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | | | Types | Optional / required | description |
| Parameters | | | | | | E | M |  |
| Version | | | | | | A | M |  |
|  | Result | | | | | E | M |  |
|  | Code | | | | A | M |  |
| LogQueryResult | | | | | E | M |  |
|  | | LogInfoList | | | E | M | Log list |
|  | LogInfo | | E | M | Log information |
|  | DeviceId | A | M | Device ID |
| DeviceIp | A | M | Device IP |
| CameraId | A | M | Camera ID |
| UserName | A | M | username |
| MajorType | A | M | Log main type, See Appendix [1.7.1 Main type](#_1.7.1_Main_type) |
| MinorType | A | M | Log Subtype, See Appendix [1.7.2 Sub type](#_1.7.2_Sub_type) |
| LogData | A | M | Log information |
| OldParamInfo | A | M | Old parameter information |
| NewParamInfo | A | M | New parameter information |
| Time | | E | M | Log time, format is YY-MM-DD HH:MM:SS:MM |

p\_result the json format is：

{

"LogInfoList": [

[{

"DeviceId": "307807A7F03D3DF",

"DeviceIp": "192.168.0.121",

"CameraId": 0,

"UserName": "admin",

"MajorType": 4,

"MinorType": 2,

"Time": "2019-10-07 22:38:29:00",

"LogData": "[admin] 192.168.0.190 sync time: MANUAL",

"OldParamInfo": "",

"NewParamInfo": ""

}.........

}]

]

}

The alarm log parameter information is as follows:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | | Types | Optional / required | description |
| Parameters | | | | | E | M |  |
| Version | | | | | A | M |  |
|  | Result | | | | E | M |  |
|  | Code | | | A | M |  |
| AlarmQueryResult | | | | E | M | Alarm log query result |
|  | AlarmInfoList | | | E | M | Alarm log information list |
|  | AlarmInfoRecordSet | | E | M | Alarm log information |
|  | DeviceId | A | M | Device ID |
| DeviceIP | A | M | Device IP |
| DeviceType | A | M | Equipment type |
| SourceType | A | M | Alarm source type |
| SourceId | A | M | Alarm source ID |
| MajorType | A | M | Log main type, See Appendix [1.1.1 Alarm main type](#_1.1.1__Alarm) |
| MinorType | A | M | Log Subtype, See Appendix [1.1.2 Alarm sub type](#_1.1.2__Alarm) |
| Description | A | M | Alarm description |
| BeginTime | E | M | Alarm start time, format is YY-MM-DD HH:MM:SS:MM |
| EndTime | E | M | Alarm end time, format is YY-MM-DD HH:MM:SS:MM |

p\_result the json format is：

{

"AlarmInfoList": [

[{

"DeviceId": "307807A7F03D3DF",

"DeviceIp": "",

"DeviceType": 5,

"SourceType": 0,

"SourceId": 1,

"MajorType": 1,

"MinorType": 4,

"Description": "",

"StartTime": "2019-10-04 07:31:21:00",

"EndTime": "2019-10-04 07:31:31:00"

}.........

}]

]

}

**Description**

#### Get device log information.

#### Return value

0 succeeded; other values failed.

### sdks\_update\_nvr

#### Define

#### int sdks\_update\_nvr(unsigned int handle, char \*p\_path)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

p\_path: Upgrade package path of NVR / DVR, such as e://picture/v4.3.0814.0000.003.0.0.42.0\_.

**Description**

#### Upgrade NVR / DVR devices. (wait about 1 minute after calling the interface to wait for the device background upgrade)

#### Return value

0 succeeded; other values failed.

### sdks\_get\_dev\_list

#### Define

int sdks\_get\_dev\_list(char \*\*p\_json\_out);

#### Parameter

p\_json\_out： Output parameters that return a list of devices searched, represented in json format. If the returned value is not null, it needs to call the sdks\_free\_result function to free memory, otherwise it will cause memory leakage.

json data;

for example:

：

{

"devices":

[

{

"addr":"xxx",

"port":xxx,

"type":xxx,

"fw":"xxx"

},

......

]

}

|  |  |  |  |
| --- | --- | --- | --- |
| parameter | explain | necessary to choose or not | remarks |
| addr | Equipment ip | Y |  |
| port | port | Y |  |
| type | device type | Y | See [1.12 Equipment type](#_1.12_Equipment_type) |
| fw | Device firmware version | Y |  |

#### Instruction

Get face detection parameters

#### Return value

0 succeeded; other values failed.

### sdks\_get\_blind\_param

#### Define

int sdks\_get\_blind\_param(unsigned int handle, const int chn,char\*\* result);

handle：Return value of the interface for sdks\_dev\_conn；

chn:Channel id

|  |  |  |  |
| --- | --- | --- | --- |
| parameter | instruction | necessary to choose or not | remarks |
| chnid | number of channel | Y | 1:ipc, other equipment fill in channel number |
| areaid | Regional id | Y |  |
| enable | enable | Y |  |
| name | name | Y |  |
| mirror | mirror image | Y |  |
| type | Masking type | Y | 1:Color block 2: mosaic 3: color block + mosaic |
| red | red | Y | Red value（0-255） |
| green | green | Y | Green value（0-255） |
| blue | blue | Y | blue valve（0-255） |
| alpha | pellucidity | Y | diaphaneity（0-100） |
| x |  | Y | Percentage of x coordinates in the upper left corner of the total video area width |
| y |  | Y | The percentage of the y coordinate in the upper corner of the total video area height |
| width |  | Y | Area width as a percentage of total video width |
| height |  | Y | Percentage of area height to total video height |

Get json eg.

{"chnData":

[{"chnid":1,

"areaid":1,

"enable":1,

"name":"TEST1",

"mirror":0,

"type":1,

"red":200,

"green":50,

"blue":100,

"alpha":20,

"x":30,

"y":30,

"width":20,

"height":20

}]

}

#### Instruction

#### Get all privacy masking information.

#### Return value

0 succeeded; other values failed.

### sdks\_set\_blind\_param

#### Define

int sdks\_set\_blind\_param(unsigned int handle, char\* p\_blind\_param);

handle：Return value of the interface for sdks\_dev\_conn；

p\_blind\_param：the json value set; the parameter analysis is the same as the acquisition interface

For example：

{"chnid":1,"areaid":1,"enable":1,"name":"TEST1","mirror":0,"type":1,"red":200,"green":50,"blue":100,"alpha":20,"topx":30,"topy":30,"width":20,"heigth":20}

#### Instruction

Set up privacy masking information in an area。

#### Return value

0 succeeded; other values failed。

### sdks\_get\_mot\_param

#### Define

int sdks\_get\_mot\_param(unsigned int handle, char\*\* result);

handle： Return value of the interface for sdks\_dev\_conn；

result: json return value

|  |  |  |  |
| --- | --- | --- | --- |
| parameter | instruction | necessary to choose or not | remarks |
| enable | Enable motion detection | Y | 1:ipc, other equipment fill in channel number |
| alarmval | Alarm interval | Y | 1-1800s |
| sensitivity | sensitivity | Y | 1-10 |
| blocknum | Number of mobile detection blocks | Y | 1: color block 2: mosaic 3: color block mosaic |
| areadata | detection area | Y | The whole region list is a one-dimensional array, in which one bit represents a grid, 0 indicates that there is no need to detect, 1 indicates that it needs to be detected, every 8 bytes is stored in an array, and less than eight bits are filled with 0. The resulting binary array is encoded by base64 to get this string. |
| widthnum | Wide block number | Y | This represents the division of the width of the whole picture into several pieces. |
| highnum | High block number | Y | This represents the division of the height of the whole picture into several pieces. |
| timelist | Time list | Y | A day is divided into 48 squares, a grid 1800s |
| weekday |  | Y | Sunday to Saturday:(06). |
| sTime |  | Y | The time, in seconds, of the beginning. |
| eTime |  | Y | End time in seconds |

[

{"mot\_source\_param":

{"enable":1,

"alarmval":10,

"blocknum":1,

"sensitivity":5,

"widthnum":22,

"highnum":18,

"areadata":"AAAAAAAAAAAAAAAAB/AAH8AAfwAB/AAH8AAfwAB/AAH8",

"timelist":

[{"weekday":0,"sTime":1800,"eTime":73800},

{"weekday":1,"sTime":1800,"eTime":73800},

{"weekday":2,"sTime":1800,"eTime":73800},

{"weekday":3,"sTime":1800,"eTime":73800},

{"weekday":4,"sTime":1800,"eTime":73800}

]

}

}

]

#### instruction

#### Sets the movement detection parameter information.

#### returned value

0 succeeded; other values failed.

### sdks\_set\_mot\_param

#### Define

int sdks\_set\_blind\_param(unsigned int handle, char\* p\_blind\_param);

handle：Return value of the interface for sdks\_dev\_conn；

p\_blind\_param：the json value set; the parameter analysis is the same as the acquisition interface

For example：

{"enable":1,"alarmval":8,"blocknum":1,"sensitivity":7,"widthnum":22,"highnum":18,"areadata":"//g//g//g//g//g//g//g//g//g//g//g//g","timelist":[{"weekday":0,"sTime":0,"eTime":86400},{"weekday":1,"sTime":0,"eTime":86400},{"weekday":2,"sTime":0,"eTime":86400}]}

#### Instrution

Set up privacy masking information in an area。

#### Return value

0 succeeded; other values failed.

### sdks\_get\_osd\_param

#### Define

#### int sdks\_get\_osd\_param(unsigned int handle, char\*\* p\_result)

#### parameter

handle：Return value of sdks\_dev\_conn interface。

p\_result：Output parameters, return the obtained OSD parameter information, expressed in json format. If the returned value is non-NULL, you need to call the sdks\_free\_result function to free the memory, otherwise it will cause a memory leak.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | | | | | | | | Types | Optional / required | description |
| Parameters | | | | | | | | | | | E | M | The message begins |
|  | Version | | | | | | | | | | A | M | Version information 1.0 |
| Result | | | | | | | | | | E | M |  |
|  | | Code | | | | | | | | A | M | Get the result: 0 success, other errors |
| OSDUserParam | | | | | | | | | | E | M |  |
|  | CameraId | | | | | | | | | A | O | Channel ID |
| OSDGlobalParam | | | | | | | | | E | M |  |
|  | | TimeFormatID | | | | | | | A | M | Time format type: See time format definition for details. |
| OSDFontAlpha | | | | | | | A | M | transparency |
| InverseFlag | | | | | | | A | M | Whether to enable reverse color |
| ChannelId | | | | | | | A | M | 0 or -1: NVR or IPC device itself, other: NVR channel number |
| OSDFontColor | | | | | | | A | M | font color |
|  | R | | | | | | A | M | Red 0-255 |
| G | | | | | | A | M | Green 0-255 |
| B | | | | | | A | M | Blue 0-255 |
| OSDFontInverseColor | | | | | | | E | M | Inverted color |
|  | | R | | | | | A | M | Red 0-255 |
| G | | | | | A | M | Green 0-255 |
| B | | | | | A | M | Blue 0-255 |
| CanvasProperList | | | | | | | | | E | M |  |
|  | | CanvasProper | | | | | | | E | M |  |
|  | | | FrontWidth | | | | A | M | Wide font size |
| FrontHeight | | | | A | M | High font size |
| AlignMode | | | | A | M | Align mode: 0 left aligned; 1 right aligned |
| AreaID | | | | A | M | Canvas area ID |
| AreaParam | | | | E | M | Regional parameters (only starting x, y are used) |
|  | TopX | | | A | M | The coordinates of the upper left corner of the area as a percentage of the total video area width (value range 0-100) |
| TopY | | | A | M | The coordinates of the upper left corner of the area as a percentage of the height of the total video area (value range 0-100) |
| Width | | | A | M | The width of the area as a percentage of the total width of the video area |
| Height | | | A | M | The height of the area as a percentage of the height of the total video area |
| OSDInfoList | | | | E | M | Video quality, see quality definition |
|  | | OSDInfo | | E | M |  |
|  | AreaID | A | M | Area ID (which belongs to the ID of the canvas) |
| ArrowID | A | M | Line number |
| Type | A | M | Watermark type: 1 device name, 2 camera number, 3 camera name, 4 time watermark, 5 text watermark, 6PTZ position operation watermark, 7PTZ behavior operation watermark (now displayed in the upper right corner), 8PTZ temperature |
| EnableFlag | A | M | Enable tag |
| ShowMode | A | M | Align mode: 0 left aligned; 1 right aligned |
| Info | A | M | Text message: only valid for text watermark |

p\_result the json format is：

{

"OSDUserParam": [{

"CameraId": 1,

"OSDGlobalParam": {

"TimeFormatID": 1,

"OSDFontAlpha": 16,

"InverseFlag": true,

"ChannelId": 0,

"OSDFontColor": {

"R": 255,

"G": 0,

"B": 0

},

"OSDFontInverseColor": {

"R": 0,

"G": 255,

"B": 255

}

},

"CanvasProperList": [{

"FrontWidth": 64,

"FrontHeight": 64,

"AlignMode": 0,

"AreaID": 1,

"AreaParam": {

"TopX": 5,

"TopY": 2,

"Width": 8,

"Height": 1

},

"OSDInfoList": [{

"AreaID": 1,

"ArrowID": 0,

"Type": 4,

"EnableFlag": true,

"ShowMode": 0,

"Info": "ASVFD"

}]

}, {

"FrontWidth": 64,

"FrontHeight": 64,

"AlignMode": 0,

"AreaID": 2,

"AreaParam": {

"TopX": 0,

"TopY": 14,

"Width": 0,

"Height": 1

},

"OSDInfoList": [{

"AreaID": 2,

"ArrowID": 0,

"Type": 5,

"EnableFlag": true,

"ShowMode": 0,

"Info": ""

}]

}, {

"FrontWidth": 64,

"FrontHeight": 64,

"AlignMode": 0,

"AreaID": 3,

"AreaParam": {

"TopX": 0,

"TopY": 14,

"Width": 0,

"Height": 1

},

"OSDInfoList": [{

"AreaID": 3,

"ArrowID": 0,

"Type": 5,

"EnableFlag": true,

"ShowMode": 0,

"Info": ""

}]

}, {

"FrontWidth": 64,

"FrontHeight": 64,

"AlignMode": 0,

"AreaID": 4,

"AreaParam": {

"TopX": 0,

"TopY": 14,

"Width": 0,

"Height": 1

},

"OSDInfoList": [{

"AreaID": 4,

"ArrowID": 0,

"Type": 5,

"EnableFlag": true,

"ShowMode": 0,

"Info": ""

}]

}, {

"FrontWidth": 64,

"FrontHeight": 64,

"AlignMode": 0,

"AreaID": 5,

"AreaParam": {

"TopX": 0,

"TopY": 14,

"Width": 0,

"Height": 1

},

"OSDInfoList": [{

"AreaID": 5,

"ArrowID": 0,

"Type": 5,

"EnableFlag": true,

"ShowMode": 0,

"Info": ""

}]

}, {

"FrontWidth": 64,

"FrontHeight": 64,

"AlignMode": 0,

"AreaID": 6,

"AreaParam": {

"TopX": 0,

"TopY": 14,

"Width": 0,

"Height": 1

},

"OSDInfoList": [{

"AreaID": 6,

"ArrowID": 0,

"Type": 5,

"EnableFlag": true,

"ShowMode": 0,

"Info": ""

}]

}, {

"FrontWidth": 64,

"FrontHeight": 64,

"AlignMode": 0,

"AreaID": 7,

"AreaParam": {

"TopX": 0,

"TopY": 14,

"Width": 0,

"Height": 1

},

"OSDInfoList": [{

"AreaID": 7,

"ArrowID": 0,

"Type": 5,

"EnableFlag": true,

"ShowMode": 0,

"Info": ""

}]

}, {

"FrontWidth": 64,

"FrontHeight": 64,

"AlignMode": 0,

"AreaID": 8,

"AreaParam": {

"TopX": 0,

"TopY": 14,

"Width": 0,

"Height": 1

},

"OSDInfoList": [{

"AreaID": 8,

"ArrowID": 0,

"Type": 5,

"EnableFlag": true,

"ShowMode": 0,

"Info": ""

}]

}]

}]

}

**Description**

#### Obtain the OSD parameter information of a channel.

#### Return value

0 succeeded; other values failed.

### sdks\_set\_osd\_param

#### Define

#### int sdks\_set\_osd\_param(unsigned int handle, char\* p\_osd\_param)

#### parameter

handle：Return value of sdks\_dev\_conn interface。

p\_osd\_param：Input parameters, expressed in json format.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | | | | | | | | Types | Optional / required | description |
| Parameters | | | | | | | | | | | E | M | The message begins |
|  | Version | | | | | | | | | | A | M | Version information 1.0 |
| Result | | | | | | | | | | E | M |  |
|  | | Code | | | | | | | | A | M | Get the result: 0 success, other errors |
| OSDUserParam | | | | | | | | | | E | M |  |
|  | CameraId | | | | | | | | | A | O | Channel ID |
| OSDGlobalParam | | | | | | | | | E | M |  |
|  | | TimeFormatID | | | | | | | A | M | Time format type: See time format definition for details. |
| OSDFontAlpha | | | | | | | A | M | transparency |
| InverseFlag | | | | | | | A | M | Whether to enable reverse color |
| ChannelId | | | | | | | A | M | 0 or -1: NVR or IPC device itself, other: NVR channel number |
| OSDFontColor | | | | | | | A | M | font color |
|  | R | | | | | | A | M | Red 0-255 |
| G | | | | | | A | M | Green 0-255 |
| B | | | | | | A | M | Blue 0-255 |
| OSDFontInverseColor | | | | | | | E | M | Inverted color |
|  | | R | | | | | A | M | Red 0-255 |
| G | | | | | A | M | Green 0-255 |
| B | | | | | A | M | Blue 0-255 |
| CanvasProperList | | | | | | | | | E | M |  |
|  | | CanvasProper | | | | | | | E | M |  |
|  | | | FrontWidth | | | | A | M | Wide font size |
| FrontHeight | | | | A | M | High font size |
| AlignMode | | | | A | M | Align mode: 0 left aligned; 1 right aligned |
| AreaID | | | | A | M | Canvas area ID |
| AreaParam | | | | E | M | Regional parameters (only starting x, y are used) |
|  | TopX | | | A | M | The coordinates of the upper left corner of the area as a percentage of the total video area width (value range 0-100) |
| TopY | | | A | M | The coordinates of the upper left corner of the area as a percentage of the height of the total video area (value range 0-100) |
| Width | | | A | M | The width of the area as a percentage of the total width of the video area |
| Height | | | A | M | The height of the area as a percentage of the height of the total video area |
| OSDInfoList | | | | E | M | Video quality, see quality definition |
|  | | OSDInfo | | E | M |  |
|  | AreaID | A | M | Area ID (which belongs to the ID of the canvas) |
| ArrowID | A | M | Line number |
| Type | A | M | Watermark type: 1 device name, 2 camera number, 3 camera name, 4 time watermark, 5 text watermark, 6PTZ position operation watermark, 7PTZ behavior operation watermark (now displayed in the upper right corner), 8PTZ temperature |
| EnableFlag | A | M | Enable tag |
| ShowMode | A | M | Align mode: 0 left aligned; 1 right aligned |
| Info | A | M | Text message: only valid for text watermark |

p\_osd\_param the json format is：

{

"OSDUserParam": [{

"CameraId": 1,

"OSDGlobalParam": {

"TimeFormatID": 1,

"OSDFontAlpha": 16,

"InverseFlag": true,

"ChannelId": 0,

"OSDFontColor": {

"R": 255,

"G": 0,

"B": 0

},

"OSDFontInverseColor": {

"R": 0,

"G": 255,

"B": 255

}

},

"CanvasProperList": [{

"FrontWidth": 64,

"FrontHeight": 64,

"AlignMode": 0,

"AreaID": 1,

"AreaParam": {

"TopX": 0,

"TopY": 15,

"Width": 74,

"Height": 1

},

"OSDInfoList": [{

"AreaID": 1,

"ArrowID": 0,

"Type": 5,

"EnableFlag": true,

"ShowMode": 0

}]

}, {

"FrontWidth": 64,

"FrontHeight": 64,

"AlignMode": 0,

"AreaID": 2,

"AreaParam": {

"TopX": 0,

"TopY": 15,

"Width": 74,

"Height": 1

},

"OSDInfoList": [{

"AreaID": 2,

"ArrowID": 0,

"Type": 5,

"EnableFlag": true,

"ShowMode": 0,

"Info": "ENGLISH"

}]

}, {

"FrontWidth": 64,

"FrontHeight": 64,

"AlignMode": 0,

"AreaID": 3,

"AreaParam": {

"TopX": 0,

"TopY": 15,

"Width": 74,

"Height": 1

},

"OSDInfoList": [{

"AreaID": 3,

"ArrowID": 0,

"Type": 5,

"EnableFlag": true,

"ShowMode": 0

}]

}, {

"FrontWidth": 64,

"FrontHeight": 64,

"AlignMode": 0,

"AreaID": 4,

"AreaParam": {

"TopX": 0,

"TopY": 15,

"Width": 74,

"Height": 1

},

"OSDInfoList": [{

"AreaID": 4,

"ArrowID": 0,

"Type": 5,

"EnableFlag": true,

"ShowMode": 0

}]

}, {

"FrontWidth": 64,

"FrontHeight": 64,

"AlignMode": 0,

"AreaID": 5,

"AreaParam": {

"TopX": 0,

"TopY": 15,

"Width": 74,

"Height": 1

},

"OSDInfoList": [{

"AreaID": 5,

"ArrowID": 0,

"Type": 5,

"EnableFlag": true,

"ShowMode": 0

}]

}, {

"FrontWidth": 64,

"FrontHeight": 64,

"AlignMode": 0,

"AreaID": 6,

"AreaParam": {

"TopX": 0,

"TopY": 15,

"Width": 74,

"Height": 1

},

"OSDInfoList": [{

"AreaID": 6,

"ArrowID": 0,

"Type": 5,

"EnableFlag": true,

"ShowMode": 0

}]

}, {

"FrontWidth": 64,

"FrontHeight": 64,

"AlignMode": 0,

"AreaID": 7,

"AreaParam": {

"TopX": 0,

"TopY": 15,

"Width": 74,

"Height": 1

},

"OSDInfoList": [{

"AreaID": 7,

"ArrowID": 0,

"Type": 5,

"EnableFlag": true,

"ShowMode": 0

}]

}, {

"FrontWidth": 64,

"FrontHeight": 64,

"AlignMode": 0,

"AreaID": 8,

"AreaParam": {

"TopX": 0,

"TopY": 15,

"Width": 74,

"Height": 1

},

"OSDInfoList": [{

"AreaID": 8,

"ArrowID": 0,

"Type": 5,

"EnableFlag": true,

"ShowMode": 0

}]

}]

}]

}

**Description**

#### Set the OSD parameter information of a channel.

#### Return value

0 succeeded; other values failed.

## Thermal imaging

### sdks\_dev\_get\_thermal\_cap

#### int sdks\_dev\_get\_thermal\_cap(unsigned int handle,int channel, char \*\*p\_result)

#### Parameter

handle: Return value of the interface for sdks\_dev\_conn

channel: number of channel

p\_result: Output parameter, returned in json format, if the returned value does not need to call the sdks\_free\_result function to free memory, otherwise it will lead to memory leakage.

#### Instruction

Obtain thermal imaging ability.

#### Return value of interface

Success returns 0 and non-0 indicates failure

#### Capability information parameter list

|  |  |  |
| --- | --- | --- |
| Predicable | | describe |
| Parameters | | Message Start |
| p\_result | | Query information |
|  | is\_thermal | Whether thermal imaging product 1 is 0 no |
| sensor\_res\_w | Detector resolution width |
| sensor\_res\_h | Detector resolution height |
| max\_measure\_area\_num | Maximum number of supported measurement areas |
| max\_point\_measure\_area\_num | The maximum number of measurement areas of the maximum support point is 0, which indicates that the temperature measurement of the point area is not supported |
| max\_line\_measure\_area\_num | The maximum number of support line measurement areas is 0: indicating that the temperature measurement of the line area is not supported |
| support\_polygon\_type | 0:not supported  1:only support rectangle area  2:only support polygon area  3:support rectangle and polygon area |
|  | max\_polygon\_measure\_area\_num | Max polygon measure area 0:means  Not support polygon measure |
|  | max\_shieldmeasure\_area\_num | Max shield measure area number  0:means not support area measure |

Example：

{

"is\_thermal": 1,

"sensor\_res\_w": 512,

"sensor\_res\_h": 640,

"max\_measure\_area\_num": 20,

"max\_point\_measure\_area\_num": 20,

"max\_line\_measure\_area\_num": 20,

"support\_polygon\_type": 2,

"max\_polygon\_measure\_area\_num": 20,

"max\_shieldmeasure\_area\_num": 0

}

### sdks\_get\_thermal\_param

#### int sdks\_get\_thermal\_param(unsigned int handle,int channel, char \*\*p\_result)

#### Parameter

handle:Return value of the interface for sdks\_dev\_conn

chn: number of channel

p\_result: Output parameter, returned in json format, if the returned value does not need to call the sdks\_free\_result function to free memory, otherwise it will lead to memory leakage.

#### Instruction

Get thermal camera basic parameter。

#### Returned value

Success return 0，not 0 means failed

#### Get a list of basic information parameters for thermal imaging

|  |  |  |
| --- | --- | --- |
| attribute | | describe |
| Parameters | | Message begins |
| p\_result | | Query information |
|  | chn\_id | Whether thermal imaging product 1 is 0 no |
| is\_open\_tempera\_measure | Whether to turn on temperature measurement |
| thermal\_measure\_mode | Detector resolution height |
| cur\_preset\_id | Current preset id (product support with PTZ) |
| area\_feature\_tempr\_show\_m | Thermal imaging measurement mode.  0: general temperature measurement mode  1: preset temperature measurement mode (product support with PTZ)  2: face temperature measurement mode (binocular human temperature measurement product support) |
| physics\_info | Correction factor |
| display\_mode | Regional temperature display mode  0: hidden area and temperature osd  1: lower left display mode  2: lower right display mode  3: Top Right Display Mode  4: only the temperature measurement area is displayed |
|  | envit\_temperature | ambient temperature |
|  | cavity\_temoperature | Cavity temoperature |
|  | temperature\_unit | Temperature unit  0: Celsius temperature  1: Fahrenheit |
|  | is\_display\_al\_area | Whether the alarm area is displayed |
|  | distance | Installation distance (support for human temperature measurement products) |
|  | threshold | Face alarm threshold (binocular human temperature measurement product support) |
|  | alarm\_int | Alarm interval |

For Example：

{

"chn\_id": 1,

"is\_open\_tempera\_measure": 1,

"thermal\_measure\_mode": 0,

"cur\_preset\_id": 0,

"area\_feature\_tempr\_show\_m": 0,

"physics\_info": 0,

"display\_mode": 1,

"envit\_temperature": 25,

"cavity\_temoperature": 40.419998,

"temperature\_unit": 0,

"is\_display\_al\_area": 0,

"distance": 0,

"threshold": 0,

"alarm\_int": 10

}

### sdks\_set\_thermal\_param

#### int sdks\_set\_thermal\_param(unsigned int handle, char\* p\_param);

#### parameter

handle: sdks\_dev\_conn Return value of interface

p\_param: Output parameters, input parameters, jason fromat，

#### instruction

set the thermal imaging basic parameters.

#### Return Value

Success returns 0, non-0 indicates failure

#### Set the list of basic information parameters for thermal imaging

|  |  |  |
| --- | --- | --- |
| attribute | | describe |
|  | is\_open\_tempera\_measure | Whether to turn on temperature measurement |
| thermal\_measure\_mode | Thermal imaging measurement mode.  0: general temperature measurement mode  1: preset temperature measurement mode (product support with cloud platform)  2: face temperature measurement mode (binocular human temperature measurement product support) |
| cur\_preset\_id | Current preset id (product support with PTZ) |
| area\_feature\_tempr\_show\_m | Osd temperature display mode  0: the highest temperature  2: highest temperature and lowest temperature  5:highest temperature, lowest temperature and average temperature |
| physics\_info | Correction factor |
| display\_mode | Zone temperature display mode  0: Hidden area and temperature OSD  1: Lower left display mode  2: Lower right display mode  3: upper right display mode  4: Only the temperature measurement area is displayed |
|  | envit\_temperature | Ambient temperature |
|  | cavity\_temoperature | Cavity teperature |
|  | temperature\_unit | Temperature unit |
|  | is\_display\_al\_area | Whether the alarm area is displayed |
|  | distance | Installation distance (support for human temperature measurement products) |
|  | threshold | Face alarm threshold (binocular human temperature measurement product support) |
|  | alarm\_int | Alarm interval |

例：

**{**http://tool.oschina.net/tools/json_format/Expanded.gif

**"is\_open\_tempera\_measure"**: 1**,**

**"thermal\_measure\_mode"**: 0**,**

**"cur\_preset\_id"**: 0**,**

**"area\_feature\_tempr\_show\_m"**: 0**,**

**"physics\_info"**: 0**,**

**"display\_mode"**: 1**,**

**"envit\_temperature"**: 25**,**

**"cavity\_temoperature"**: 39.9**,**

**"temperature\_unit"**: 0**,**

**"is\_display\_al\_area"**: 1**,**

**"distance"**: 0**,**

**"threshold"**: 0**,**

**"alarm\_int"**: 10

**}**

### sdks\_get\_thermal\_area\_temperature\_measure

#### int sdks\_get\_thermal\_area\_temperature\_measure(unsigned int handle,char \*p\_param, char \*\*p\_result);

#### parameter

interface returned value for handle: sdks\_dev\_conn

p\_param: input parameter with json format。

p\_result: The output parameter is returned in json format, and if this value is not NULL, you need to call the sdks\_free\_result function to free the memory, otherwise it will cause a memory leak.

#### instruction

Get thermal camera basic parameter。

#### Returned value

Success returns 0 and non-0 indicates failure

Request parameter list

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Predicable | | | | Describe |
|  | chn\_id | | | Device channel id |
| area\_oper\_type | | | Thermal imagin area operation configuration type  1:represents a single region 2:represents multiple regions 3:represents all regions |
| Value | | | Preset bit ID(valid only in the preset bit temperature measurement mode) |
|  | area\_list | | Area ID list |
|  | area\_id | Area ID value |

Eg.

{

"area\_list": [

{

"area\_id": 0

},

{

"area\_id": 1

}

],

"chn\_id": 1,

"area\_oper\_type": 2,

"value": 0

}

#### Return value parameter list

|  |  |  |  |
| --- | --- | --- | --- |
| 描nature | | | description |
|  | Measure\_mode | | Thermal imaging measurement mode  0:normal mode  1:preset temperature  measurement mode  2:human face temperature measure mode |
| preset\_id | | Preset id (valid in preset measurement mode) |
| area\_list | | Preset id (valid in preset measurement mode) |
|  | source\_id |  |
| source\_type |  |
| area\_id | area\_id |
| area\_name | Area name |
| alarm\_enable\_flag | Area temperature alarm enable mark |
| alarm\_type | Area temperature alarm Type  0: temperature difference alarm 1: threshold alarm |
| warning\_alarm\_value | Area temperature alarm early warning value |
| alarm\_value | Zone temperature alarm value. |
| Emissivity | Emissivity (0.0 ~ 1.0) |
| target\_space | object distance |
| area\_en\_flag | area-on flag |
| area\_shape\_type | Area shape type 1: point, 2: straight line, 3: rectangle, 4: polygons |
| point\_coordinate | Area boundary points (capable of describing a region boundary, percentage representation) coordinate list (arranged in a clockwise direction), range[0,99.99] |

Eg.

{

"measure\_mode": 0,

"preset\_id": 0,

"area\_list": [

{

"source\_id": 0,

"source\_type": 23,

"area\_id": 0,

"area\_name": "Area0",

"alarm\_enable\_flag": 0,

"alarm\_type": 1,

"warning\_alarm\_value": 90,

"alarm\_value": 100,

"emissivity": 0.95,

"target\_space": 15,

"area\_en\_flag": 0,

"area\_shape\_type": 3,

"point\_coordinate": "(0.000000, 0.000000);(99.989998, 0.000000);(99.989998, 99.989998);(0.000000, 99.989998)"

},

{

"source\_id": 0,

"source\_type": 23,

"area\_id": 1,

"area\_name": "Area1",

"alarm\_enable\_flag": 0,

"alarm\_type": 1,

"warning\_alarm\_value": 48,

"alarm\_value": 50,

"emissivity": 0.95,

"target\_space": 15,

"area\_en\_flag": 0,

"area\_shape\_type": 0,

"point\_coordinate": ""

}

]

}

### sdks\_set\_thermal\_area\_temperature\_measure

#### int sdks\_set\_thermal\_area\_temperature\_measure(unsigned int handle,char \*p\_param);

#### Parameter

handle: Return value of the interface for sdks\_dev\_conn

p\_param: Input parameters, input in json format.

#### Instruction

Request for setting thermal imaging measurement parameters.

#### Return value

Success returns 0, non-0 indicates failure

#### Request parameter list

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 描nature | | | description | |
|  | area\_oper\_type | | | thermal imaging area operation configuration type 1: representing a single area 2: representing a plurality of areas 3: representing all areas |
| Measure\_mode | | | Thermal imaging measurement mode  0: normal mode  1: preset bit temperature measurement mode  2: face temperature measurement mode |
| preset\_id | | | Preset id (valid in preset measurement mode) |
| area\_list | | | List of regional temperature measurement parameters and regional temperature alarm parameters |
|  | area\_id | | Area\_id |
| area\_name | | Area\_id |
| alarm\_enable\_flag | | Area temperature alarm enable mark |
| alarm\_type | | Area temperature alarm Type  0: temperature difference alarm 1: threshold alarm |
| warning\_alarm\_value | | Area temperature alarm early warning value |
| alarm\_value | | Zone temperature alarm value. |
| emissivity | | Emissivity (0.0 ~ 1.0) |
| target\_space | | object distance |
| area\_en\_flag | | area-on flag |
| area\_shape\_type | | Area shape type 1: point, 2: straight line, 3: rectangle, 4: polygons |
| point\_coordinate | | Area boundary points (capable of describing a region boundary, percentage representation) coordinate list (arranged in a clockwise direction), range[0,99.99] |

Eg.

{

"area\_oper\_type": 3,

"measure\_mode": 0,

"preset\_id": 0,

"channel": 1,

"area\_list": [{

"area\_id": 0,

"area\_name": "area\_0",

"alarm\_en": 1,

"alarm\_type": 1,

"warning\_alarm\_value": 91,

"alarm\_value": 100,

"emissivity": 0.950000,

"target\_space": 15,

"area\_en\_flag": 1,

"area\_shape\_type": 3,

"point\_coor\_list": "(0, 0);(99, 0);(99, 99);(0, 99)"

}]

}

### sdks\_get\_thermal\_area\_feature\_temperature

#### int sdks\_get\_thermal\_area\_feature\_temperature(unsigned int handle, char \*p\_param, char\*\* p\_result)

#### parameter

handle: Return value of the interface for sdks\_dev\_conn

p\_param: Input parameters, input in json format.

p\_result: Output parameter returned in json format. If this value returned is non-null, it is necessary to call the sks \_ free \_ result function to release the memory, otherwise the memory leaks.

#### Instruction

acquiring the characteristic temperature response of the thermal imaging temperature measuring region

#### Return value

Success returns 0, non-0 indicates failure

#### Request parameter list

|  |  |  |
| --- | --- | --- |
| Nature | | describe |
|  | chn\_id | Device channel id |
| area\_oper\_type | Thermal imaging area operation configuration type.  1: representing a single region  2: representing a plurality of regions  3: representing all regions |

For example：

{"chn\_id":0,"area\_oper\_type":2}

#### Return parameter list

|  |  |  |
| --- | --- | --- |
| nature | | describe |
|  | area\_id | Area\_id |
| temperature\_unit | Temperature unit |
|  | area\_max\_temperature | Area maximum temperature |
|  | area\_min\_temperature | Area minimum temperature |
|  | area\_ave\_temperature | Area average temperature |
|  | max\_point\_x | X coordinate of are maximum temp point |
|  | max\_point\_y | Y coordinate of are maximum temp point |
|  | min\_point\_x | X coordinate of are minimum temp point |
|  | min\_point\_y | Y coordinate of are minimum temp point |

Eg.

[

{

"area\_id": 0,

"temperature\_unit": 0,

"area\_max\_temperature": 29,

"area\_min\_temperature": 19,

"area\_ave\_temperature": 23,

"max\_point\_x": 56,

"max\_point\_y": 72,

"min\_point\_x": 56,

"min\_point\_y": 72

},

{

"area\_id": 1,

"temperature\_unit": 0,

"area\_max\_temperature": 27,

"area\_min\_temperature": 20,

"area\_ave\_temperature": 22,

"max\_point\_x": 25,

"max\_point\_y": 19,

"min\_point\_x": 25,

"min\_point\_y": 19

},

{

"area\_id": 2,

"temperature\_unit": 0,

"area\_max\_temperature": 27,

"area\_min\_temperature": 20,

"area\_ave\_temperature": 25,

"max\_point\_x": 43,

"max\_point\_y": 78,

"min\_point\_x": 43,

"min\_point\_y": 78

},

{

"area\_id": 3,

"temperature\_unit": 0,

"area\_max\_temperature": 28,

"area\_min\_temperature": 28,

"area\_ave\_temperature": 28,

"max\_point\_x": 64,

"max\_point\_y": 77,

"min\_point\_x": 64,

"min\_point\_y": 77

}

]

### sdks\_get\_thermal\_one\_point\_temperature

#### int sdks\_get\_thermal\_one\_point\_temperature(unsigned int handle, int chn, int x, int y, char\*\* p\_result);

#### Parameter

handle: Return value of the interface for sdks\_dev\_conn

channel:number of channel

x: Point X coordinate

y: Point y coordinate

p\_result: Output parameter returned in json format. If this value returned is non-null, it is necessary to call the sks \_ free \_ result function to release the memory, otherwise the memory leaks.

#### Instruction

Get the temperature value of any point in the full screen area of the thermal imaging image.

#### Return value

Success returns 0, non-0 indicates failure

#### Return a list of parameters

|  |  |
| --- | --- |
| Parameter | remark |
| point\_x | Point x coordinate |
| point\_y | Point y coordinate |
| temperature\_value | point temperature value |
| temperature\_unit | Temperature\_unit |

Eg.

{"point\_x":100,"point\_y":150,"temperature\_value":23.848297,"temperature\_unit":0}

### sdks\_get\_thermal\_any\_point\_temperature

#### int sdks\_get\_thermal\_any\_point\_temperature(unsigned int handle, int chn, int x, int y, char\*\* p\_result);

#### parameter

handle: Return value of the interface for sdks\_dev\_conn

channel:chanel number

x: Point x coordinate

y: Point y coordinate

p\_result: Output parameter returned in json format. If this value returned is non-null, it is necessary to call the sks \_ free \_ result function to release the memory, otherwise the memory leaks.

#### Instruction

Get the temperature value of any number of points in the full screen area of thermal imaging.

#### Return value

Success returns 0, non-0 indicates failure

#### Request parameter list

|  |  |  |  |
| --- | --- | --- | --- |
| Nature | | | describe |
|  | area\_list | | Area\_list |
|  | Point\_x | Point x coordinates, percentage representation. [0,99.99] |
| Point\_y | Point y coordinate, percentage representation. [0,99.99] |
| temperature\_value |  |
| temperature\_unit |  |

Eg.

{"point\_x":100,"point\_y":150,"temperature\_value":23.848297,"temperature\_unit":0}

### sdks\_get\_map\_relation

#### int sdks\_get\_map\_relation(unsigned int handle, int chnnel, char\*\* p\_result);

#### Parameter

handle: Return value of the interface for sdks\_dev\_conn

channel:channel number

p\_result: Output parameter, returned in json format, if the returned value does not need to call the sdks\_free\_result function to free memory, otherwise it will lead to memory leakage.

#### Instruction

Obtaining the relationship between thermal imaging and visible light mapping

#### returned value

Success returns 0 and non-0 indicates failure

#### return parameter list

|  |  |  |
| --- | --- | --- |
| predicable | | describe |
| src\_point | | point coordinate list |
|  | x\_val | Point x coordinate, percent representation. [0,99.99] |
|  | y\_val | Point y coordinates, percent representation. [0,99.99] |
| dst\_point | | List of point coordinate |
|  | x\_val | Point x coordinate, percent representation. [0,99.99] |
|  | Yval | Point y coordinates, percent representation. [0,99.99] |

For example：

{

"src\_point": [

{

"x\_val": 10,

"y\_val": 15

},

{

"x\_val": 11,

"y\_val": 16

}

],

"dst\_point": [

{

"x\_val": 25,

"y\_val": 30

},

{

"x\_val": 26,

"y\_val": 31

}

]

}

### sdks\_set\_map\_relation

#### int sdks\_set\_map\_relation(unsigned int handle, char\*\* p\_param);

#### parameter

handle:Return value of the interface for sdks\_dev\_conn

p\_param: input parameters, input in json format

#### instruction

Set the relationship between thermal imaging and visible light mapping

#### returned value

Success returns 0 and non-0 indicates failure

#### Parameter list

|  |  |  |
| --- | --- | --- |
| predicable | | describe |
| src\_point | | point coordinate list |
|  | x\_val | Point x coordinate, percent representation. [0,99.99] |
|  | y\_val | Point y coordinates, percent representation. [0,99.99] |
| dst\_point | | point coordinate list |
|  | x\_val | Point x coordinate, percent representation. [0,99.99] |
|  | Yval | Point y coordinates, percent representation. [0,99.99] |

For example：

{

"src\_point": [

{

"src\_x": 10,

"src\_y": 15

},

{

"src\_x": 11,

"src\_y": 16

}

],

"dst\_point": [

{

"dst\_x": 25,

"dst\_y": 30

},

{

"dst\_x": 26,

"dst\_y": 31

}

],

"chn\_id": 1

}

### sdks\_get\_temperature\_calibration

#### int sdks\_get\_temperature\_calibration(unsigned int handle, int channel, char\*\* p\_result);

#### parameter

handle: Return value of the interface for sdks\_dev\_conn

channel:number of channel

p\_result: Output parameter, returned in json format, if the returned value does not need to call the sdks\_free\_result function to free memory, otherwise it will lead to memory leakage.

#### instruction

Request for obtaining Thermal Imaging temperature Measurement Index parameters

#### return value

Success returns 0 and non-0 indicates failure

#### return parameter list

|  |  |  |
| --- | --- | --- |
| predicable | | describe |
|  | enable | Whether or not to turn on |
| show\_obj\_en | Whether it is in the video overlay area and temperature. |
| black\_body\_temp | black-body temperature |
| tempera\_unit | degree-day |
| emiss\_vity | emissivity |
| target\_space | Installation distance |
| top\_x | Black body region starting point x coordinate, percent representation. [0,99.99] |
| top\_y | The black body region start-point y-coordinate, percent representation. [0,99.99] |
| width | The blackbody area is wide and the percentage indicates. [0, 99.99] |
|  | height | Blackbody area is high, percentage indicates. [0, 99.99] |

Eg.

{

"enable": 1,

"show\_obj\_en": 1,

"black\_body\_temp": 40,

"tempera\_unit": 0,

"emiss\_vity": 0.098,

"target\_space": 5,

"top\_x": 64.720009,

"top\_y": 23.46999,

"width": 36.12001,

"height": 36.389999

}

### sdks\_set\_temperature\_calibration

#### int sdks\_set\_temperature\_calibration(unsigned int handle, int channel, char\*\* p\_param);

#### parameter

handle: Return value of the interface for sdks\_dev\_conn

channel:number of channel

p\_param: input parameters, input in json format

#### instruction

Set thermal imaging temperature measurement calibration parameters

#### returned value

Success returns 0 and non-0 indicates failure

#### Parameter list

|  |  |  |
| --- | --- | --- |
| predicable | | describe |
|  | enable | Whether or not to turn on |
| show\_obj\_en | Whether or not to turn on |
| black\_body\_temp | black-body temperature |
| tempera\_unit | degree-day |
| emiss\_vity | emissivity |
| target\_space | Installation distance |
| top\_x | Starting point x coordinates, percentage representation. [0, 99.99] |
| top\_y | Start y coordinate, expressed as a percentage. [0,99.99]. |
| width | The percentage is expressed in terms of width and percentage. [0,99.99] |
|  | height | High, percentage indicates. [0, 99.99] |

For example：

{

"enable": 1,

"show\_obj\_en": 1,

"black\_body\_temp": 40,

"tempera\_unit": 0,

"emiss\_vity": 0.098,

"target\_space": 5,

"top\_x": 64.72001,

"top\_y": 23.46999,

"width": 36.12001,

"height": 36.389999

}

### sdks\_get\_thermal\_alarm\_linkage\_param

int sdks\_get\_thermal\_alarm\_linkage\_param(unsigned int handle, int channel, char\*\* p\_result);

#### parameter

handle: Return value of the interface for sdks\_dev\_conn

channel:number of channel

p\_result: Output parameter, returned in json format, if the returned value does not need to call the sdks\_free\_result function to free memory, otherwise it will lead to memory leakage.

#### instruction

Get thermal imaging temperature alarm cloth anti-linkage parameter

#### returned value

Success returns 0 and non-0 indicates failure

#### Parameter list

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| predicable | | | | | | | | describe |
|  |  | source\_type | | | | | | Alarm source type  29：Face temperature measurement alarm  31：temperature threshold alarm  32：Temperature threshold warning  33：temperature difference alarm  34：Temperature difference early warning |
|  | source\_id | | | | | | Alarm source id |
|  |  | alarm\_out | | | | | Alarm source id |
|  | | act\_type | | | Action type 1:io output 2: Mail 3: cloud Action 4: video 5: message 6: log 7:ftp 8: message push |
| act\_id | | | Action id |
| act\_name | | | Action Name |
| dev\_id | | | device id |
| a\_out\_id | | | The id number of the alarm output port. |
| a\_out\_flag | | | Alarm output flag |
| e\_type\_id | | | Alarm event type |
| a\_time | | | output time |
|  | | record\_act | | | | linked video parameters |
|  | | act\_type | | linked video parameters |
| act\_id | | Linkage id |
| act\_name | | Linkage name |
| pre\_record\_flag | | Pre-recorded mark |
| delay\_record\_time | | Delay recording time |
|  | | ptz\_action | | | | Ptz action parameters |
|  | | act\_type | | Type of action |
| act\_id | | Action id |
| act\_name | | Action Name |
| ptz\_act\_type | | Operation type (preset bit, track, etc.) |
| ptz\_act\_id | | Operate id (pre-set id, track id, etc., as previously set by the user, etc.) |
| ptz\_chn\_id | | Ptz channel id |
|  | alarm\_link | | | | | Alarm linkage parameter |
|  | | act\_type | | | Type of action  1:io output 2: Mail 3: cloud Action 4: video 5: message 6: log 7:ftp 8: message push |
| act\_id | | | Action id |
|  | schedule | | | | | Time-of-deployment list |
|  | | ScheduleTime | | | Layout time |
|  | | week\_day | The number of days of the week |
| s\_time | starting time |
| e\_time | terminal time |

Eg.

[

{

"source\_type": 0,

"source\_id": 0,

"alarm\_out": [

{

"act\_type": 1,

"act\_id": 1,

"act\_name": "AlarmOut1",

"dev\_id": 1,

"a\_out\_id": 1,

"a\_out\_flag": 0,

"e\_type\_id": 0,

"a\_time": 0

},

{

"act\_type": 1,

"act\_id": 2,

"act\_name": "AlarmOut2",

"dev\_id": 2,

"a\_out\_id": 2,

"a\_out\_flag": 0,

"e\_type\_id": 0,

"a\_time": 0

}

],

"ptz\_action": [ ],

"record\_act": [ ],

"alarm\_link": [ ],

"schedule": [ ]

},

{

"source\_type": 0,

"source\_id": 0,

"alarm\_out": [

{

"act\_type": 1,

"act\_id": 1,

"act\_name": "AlarmOut1",

"dev\_id": 1,

"a\_out\_id": 1,

"a\_out\_flag": 1,

"e\_type\_id": 0,

"a\_time": 0

},

{

"act\_type": 1,

"act\_id": 2,

"act\_name": "AlarmOut2",

"dev\_id": 2,

"a\_out\_id": 2,

"a\_out\_flag": 0,

"e\_type\_id": 0,

"a\_time": 0

}

],

"ptz\_action": [ ],

"record\_act": [ ],

"alarm\_link": [

{

"act\_type": 2,

"act\_id": 1

},

{

"act\_type": 1,

"act\_id": 1

}

],

"schedule": [

{

"week\_day": 0,

"s\_time": 0,

"e\_time": 86400

},

{

"week\_day": 1,

"s\_time": 0,

"e\_time": 86400

},

{

"week\_day": 2,

"s\_time": 0,

"e\_time": 86400

},

{

"week\_day": 3,

"s\_time": 0,

"e\_time": 86400

},

{

"week\_day": 4,

"s\_time": 0,

"e\_time": 86400

},

{

"week\_day": 5,

"s\_time": 0,

"e\_time": 86400

},

{

"week\_day": 6,

"s\_time": 0,

"e\_time": 86400

}

]

},

{

"source\_type": 0,

"source\_id": 0,

"alarm\_out": [

{

"act\_type": 1,

"act\_id": 1,

"act\_name": "AlarmOut1",

"dev\_id": 1,

"a\_out\_id": 1,

"a\_out\_flag": 1,

"e\_type\_id": 0,

"a\_time": 0

},

{

"act\_type": 1,

"act\_id": 2,

"act\_name": "AlarmOut2",

"dev\_id": 2,

"a\_out\_id": 2,

"a\_out\_flag": 0,

"e\_type\_id": 0,

"a\_time": 0

}

],

"ptz\_action": [ ],

"record\_act": [ ],

"alarm\_link": [

{

"act\_type": 2,

"act\_id": 1

},

{

"act\_type": 1,

"act\_id": 1

}

],

"schedule": [

{

"week\_day": 0,

"s\_time": 0,

"e\_time": 86400

},

{

"week\_day": 1,

"s\_time": 0,

"e\_time": 86400

},

{

"week\_day": 2,

"s\_time": 0,

"e\_time": 86400

},

{

"week\_day": 3,

"s\_time": 0,

"e\_time": 86400

},

{

"week\_day": 4,

"s\_time": 0,

"e\_time": 86400

},

{

"week\_day": 5,

"s\_time": 0,

"e\_time": 86400

},

{

"week\_day": 6,

"s\_time": 0,

"e\_time": 86400

}

]

},

{

"source\_type": 0,

"source\_id": 0,

"alarm\_out": [

{

"act\_type": 1,

"act\_id": 1,

"act\_name": "AlarmOut1",

"dev\_id": 1,

"a\_out\_id": 1,

"a\_out\_flag": 1,

"e\_type\_id": 0,

"a\_time": 0

},

{

"act\_type": 1,

"act\_id": 2,

"act\_name": "AlarmOut2",

"dev\_id": 2,

"a\_out\_id": 2,

"a\_out\_flag": 0,

"e\_type\_id": 0,

"a\_time": 0

}

],

"ptz\_action": [ ],

"record\_act": [ ],

"alarm\_link": [

{

"act\_type": 4,

"act\_id": 1

},

{

"act\_type": 1,

"act\_id": 1

}

],

"schedule": [

{

"week\_day": 1,

"s\_time": 16200,

"e\_time": 50400

},

{

"week\_day": 2,

"s\_time": 16200,

"e\_time": 50400

},

{

"week\_day": 3,

"s\_time": 16200,

"e\_time": 50400

},

{

"week\_day": 4,

"s\_time": 16200,

"e\_time": 50400

},

{

"week\_day": 5,

"s\_time": 16200,

"e\_time": 50400

}

]

},

{

"source\_type": 0,

"source\_id": 0,

"alarm\_out": [

{

"act\_type": 1,

"act\_id": 1,

"act\_name": "AlarmOut1",

"dev\_id": 1,

"a\_out\_id": 1,

"a\_out\_flag": 1,

"e\_type\_id": 0,

"a\_time": 0

},

{

"act\_type": 1,

"act\_id": 2,

"act\_name": "AlarmOut2",

"dev\_id": 2,

"a\_out\_id": 2,

"a\_out\_flag": 1,

"e\_type\_id": 0,

"a\_time": 0

}

],

"ptz\_action": [ ],

"record\_act": [ ],

"alarm\_link": [

{

"act\_type": 1,

"act\_id": 1

},

{

"act\_type": 1,

"act\_id": 2

}

],

"schedule": [

{

"week\_day": 1,

"s\_time": 5400,

"e\_time": 64800

},

{

"week\_day": 2,

"s\_time": 5400,

"e\_time": 64800

},

{

"week\_day": 3,

"s\_time": 5400,

"e\_time": 64800

},

{

"week\_day": 4,

"s\_time": 5400,

"e\_time": 64800

}

]

},

{

"source\_type": 0,

"source\_id": 0,

"alarm\_out": [

{

"act\_type": 1,

"act\_id": 1,

"act\_name": "AlarmOut0",

"dev\_id": 1,

"a\_out\_id": 1,

"a\_out\_flag": 0,

"e\_type\_id": 0,

"a\_time": 0

},

{

"act\_type": 1,

"act\_id": 2,

"act\_name": "AlarmOut1",

"dev\_id": 2,

"a\_out\_id": 2,

"a\_out\_flag": 0,

"e\_type\_id": 0,

"a\_time": 0

}

],

"ptz\_action": [ ],

"record\_act": [ ],

"alarm\_link": [ ],

"schedule": [

{

"week\_day": 0,

"s\_time": 0,

"e\_time": 86400

},

{

"week\_day": 1,

"s\_time": 0,

"e\_time": 86400

},

{

"week\_day": 2,

"s\_time": 43200,

"e\_time": 86400

},

{

"week\_day": 3,

"s\_time": 43200,

"e\_time": 86400

}

]

}

]

### sdks\_set\_thermal\_alarm\_linkage\_param

int sdks\_set\_thermal\_alarm\_linkage\_param(unsigned int handle, int channel, char\* param);

#### parameter

handle: Return value of the interface for sdks\_dev\_conn

channel:number of channel

p\_param: input parameters, in json format

#### Instruction

#### Set the thermal imaging temperature alarm linkage parameter.

#### returned value

Success returns 0 and non-0 indicates failure

#### Parameter list

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Predicable | | | | | | | | describe |
| thermal\_linkage\_list | | | | | | | |  |
|  |  | source\_type | | | | | | Alarm source type  29：Face temperature measurement alarm  31：temperature threshold alarm  32：Temperature threshold warning  33：temperature difference alarm  34：Temperature difference early warning |
|  | source\_id | | | | | | Alarm source id |
|  |  | alarm\_out | | | | | Alarm output action |
|  | | act\_type | | | Action type  1:io output 2: Mail 3: cloud Action 4: video 5: message 6: log 7:ftp 8: message push |
| act\_id | | | Action id |
| act\_name | | | Action Name |
| dev\_id | | | device id |
| a\_out\_id | | | The id number of the alarm output port. |
| a\_out\_flag | | | Alarm output flag |
| e\_type\_id | | | Alarm output flag |
| a\_time | | | output time |
|  | | record\_act | | | | linked video parameters |
|  | | act\_type | | linkage type |
| act\_id | | Linkage id |
| act\_name | | Linkage name |
| pre\_record\_flag | | Linkage name |
| delay\_record\_time | | Delay recording time |
|  | | ptz\_action | | | | Ptz action parameters |
|  | | act\_type | | Type of action |
| act\_id | | Action id |
| act\_name | | Action Name |
| ptz\_act\_type | | Operation type (preset bit, track, etc.) |
| ptz\_act\_id | | Operate id (pre-set id, track id, etc., as previously set by the user, etc.) |
| ptz\_chn\_id | | Ptz channel id |
|  | alarm\_link | | | | | Alarm linkage parameter |
|  | | act\_type | | | Action type  1:io output 2: Mail 3: cloud Action 4: video 5: message 6: log 7:ftp 8: message push |
| act\_id | | | Action id |
|  | Schedule | | | | | Time-of-deployment list |
|  | | ScheduleTime | | | Layout time |
|  | | week\_day | The number of days of the week |
| s\_time | The number of days of the week |
| e\_time | terminal time |

Eg.

{

"thermal\_linkage\_list": [

{

"source\_type": 1,

"source\_id": 1,

"alarm\_out\_list": [

{

"act\_name": "AlarmOut0",

"act\_id": 1,

"act\_type": 1,

"dev\_id": 1,

"alarm\_out\_id": 1,

"event\_id": 1,

"alarm\_time": 0,

"alarm\_out\_flag": 0

},

{

"act\_name": "AlarmOut1",

"act\_id": 2,

"act\_type": 1,

"dev\_id": 2,

"alarm\_out\_id": 2,

"event\_id": 2,

"alarm\_time": 0,

"alarm\_out\_flag": 0

}

],

"schedule\_list": [

{

"weekday": 0,

"start\_time": 0,

"end\_time": 86400

},

{

"weekday": 1,

"start\_time": 0,

"end\_time": 86400

},

{

"weekday": 2,

"start\_time": 43200,

"end\_time": 86400

},

{

"weekday": 3,

"start\_time": 43200,

"end\_time": 86400

}

]

}

]

}

### sdks\_test\_thermal\_bad\_point\_correct

int sdks\_test\_thermal\_bad\_point\_correct(unsigned int handle,char \*p\_param)

#### parameter

handle: Return value of the interface for sdks\_dev\_conn

channel:number of channel

p\_param: Enter parameters that are passed in json format

#### Instruction

Test bad point correction

#### returned value

Success returns 0 and non-0 indicates failure

#### Parameter list

|  |  |  |  |
| --- | --- | --- | --- |
| Predicable | | | describe |
|  |  | point\_x | Point x coordinates, percentage representation. [099.99] |
| point\_y | Point y coordinate, percentage representation. [099.99] |
| chn\_id | Channel id |

Eg.

{"chn\_id":1,"point\_x":10,"point\_y":30}

### sdks\_set\_thermal\_bad\_point\_correct

int sdks\_set\_thermal\_bad\_point\_correct (unsigned int handle,int channel)

#### Parameter

handle: Return value of the interface for sdks\_dev\_conn

channel:number of channel

#### Instruction

Set up thermal imaging damage point correction.

#### returned value

Success returns 0 and non-0 indicates failure

### sdks\_reset\_thermal\_bad\_point\_correct

int sdks\_reset\_thermal\_bad\_point\_correct (unsigned int handle,int channel)

#### parameter

handle:Return value of the interface for sdks\_dev\_conn

channel:number of channel

#### Instruction

number of channel

#### returned value

Success returns 0 and non-0 indicates failure

### sdks\_get\_thermal\_version

int sdks\_get\_thermal\_version(unsigned int handle,int channel,char \*\*p\_result)

#### parameter

handle: Return value of the interface for sdks\_dev\_conn

channel:number of channel

p\_result: Output parameter, returned in json format, if the returned value does not need to call the sdks\_free\_result function to free memory, otherwise it will lead to memory leakage.

#### Return a list of parameters

|  |  |  |
| --- | --- | --- |
| Predicable | | describe |
|  | Ver | Movement version |
| "ver | Core serial number |

For example：

{"ver":"20180930","seq":"014881"}

#### instruction

Get thermal camera version

#### Return value

Success return 0，not 0 means failed

## License plate

### sdks\_get\_lpr\_link\_param

#### int sdks\_get\_lpr\_link\_param(unsigned int handle, int chn, char \*\*p\_result);

#### parameter

handle: Return value of the interface for sdks\_dev\_conn.

chn： Channel number.

p\_result：Output parameters, return the obtained license plate recognition linkage parameters, expressed in json format. If the returned value is non-NULL, you need to call the sdks\_free\_result function to free the memory, otherwise it will cause a memory leak.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | Types | Optional / required | description | |
| Parameters | | | E | M | The message begins | |
|  | Version | | A | M | Version information 1.0 | |
| LprLinkParam | | E | M |  | |
|  | BlackListFTPUpload | A | M | Blacklist license plate screenshot FTP upload |
| BlackListOpenBarrier | A | M | Blacklist license plate open |
| BlackListSMTP | A | M | Blacklist SMTP |
| WhilteListFTPUpload | A | M | FTP upload of screenshot of white list license plate |
| WhilteListOpenBarrier | A | M | White list license plate open |
| WhilteListSMTP | A | M | Whitelist SMTP |
| NoListFTPUpload | A | M | No FTP upload of the screenshot of the list and license plate |
| NoListOpenBarrier | A | M | Not on the list. The license plate is open |
| NoListSMTP | A | M | Not on list SMTP |
| SnapshotFTPUpload | A | M | Snapshot FTP upload |
| OpenBarrierLevel | A | M | Opening level (false: low, true: high) |
| OpenBarrierDuration | A | M | Opening duration |
| LprOSD | A | M | OSD display of license plate recognition information |
| LprOSDDuration | A | M | OSD display duration (zero is always displayed) |

p\_result json format is：

{

"LprLinkParam": {

"BlackListFTPUpload": false,

"BlackListOpenBarrier": false,

"BlackListSMTP": false,

"WhilteListFTPUpload": false,

"WhilteListOpenBarrier": true,

"WhilteListSMTP": false,

"NoListFTPUpload": false,

"NoListOpenBarrier": false,

"NoListSMTP": false,

"SnapshotFTPUpload": false,

"OpenBarrierLevel": true,

"OpenBarrierDuration": 20,

"LprOSD": false,

"LprOSDDuration": 60

}

}

Description

#### Obtain the license plate recognition linkage parameter.

#### Return value

0 succeeded; other values failed.

### sdks\_set\_lpr\_link\_param

#### int sdks\_set\_lpr\_link\_param (unsigned int handle, int chn, char \*p\_param);

#### Parameter

handle: Return value of the interface for sdks\_dev\_conn.

chn： Channel number.

p\_param：Input parameters, expressed in json format.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | Types | Optional / required | description | |
| Parameters | | | E | M | The message begins | |
|  | Version | | A | M | Version information 1.0 | |
| LprLinkParam | | E | M |  | |
|  | BlackListFTPUpload | A | M | Blacklist license plate screenshot FTP upload |
| BlackListOpenBarrier | A | M | Blacklist license plate open |
| BlackListSMTP | A | M | Blacklist SMTP |
| WhilteListFTPUpload | A | M | FTP upload of screenshot of white list license plate |
| WhilteListOpenBarrier | A | M | White list license plate open |
| WhilteListSMTP | A | M | Whitelist SMTP |
| NoListFTPUpload | A | M | No FTP upload of the screenshot of the list and license plate |
| NoListOpenBarrier | A | M | Not on the list. The license plate is open |
| NoListSMTP | A | M | Not on list SMTP |
| SnapshotFTPUpload | A | M | Snapshot FTP upload |
| OpenBarrierLevel | A | M | Opening level (false: low, true: high) |
| OpenBarrierDuration | A | M | Opening duration |
| LprOSD | A | M | OSD display of license plate recognition information |
| LprOSDDuration | A | M | OSD display duration (zero is always displayed) |

p\_param json format is：

{

"LprLinkParam": {

"BlackListFTPUpload": false,

"BlackListOpenBarrier": false,

"BlackListSMTP": false,

"WhilteListFTPUpload": false,

"WhilteListOpenBarrier": true,

"WhilteListSMTP": false,

"NoListFTPUpload": false,

"NoListOpenBarrier": false,

"NoListSMTP": false,

"SnapshotFTPUpload": false,

"OpenBarrierLevel": true,

"OpenBarrierDuration": 20,

"LprOSD": false,

"LprOSDDuration": 60

}

}

Description

#### Setting the license plate recognition linkage parameter.

#### Return value

0 succeeded; other values failed.

### sdks\_lpr\_ipfilter\_list\_add

#### sdks\_lpr\_ipfilter\_list\_add(unsigned int handle, int chn, char \*p\_param);

#### Parameter

handle: Return value of the interface for sdks\_dev\_conn.

chn： Channel number.

p\_param：Input parameters, expressed in json format.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | | Types | Optional / required | description | |
| Parameters | | | | | E | M | The message begins | |
|  | Version | | | | A | M | Version information 1.0 | |
| ChannelId | | | | E | M |  | |
|  | | Id | | A | M |  | |
| LprIpfilterList | | | | E | M |  | |
|  |  | | Text | A | M | License plate number, 5~30 characters, can only contain English characters and numbers, otherwise the setting is invalid |
| Type | A | M | List type, 0: blacklist, 1: white list |
| StartTime | A | M | Validity start time, valid before the current time, UTC timestamp, based on device time, in seconds |
|  |  |  | | EndTime | A | M | Expiration date is valid later than the current time, UTC timestamp, based on device time, in seconds |

p\_param json format is：

{

"LprIpfilterList": [{

"Text": "BCDS6",

"Type": 1,

"StartTime": 1565539200,

"EndTime": 1573710777

}]

}

Description

Add a license plate to identify the black and white list.

#### Return value

0 succeeded; other values failed.

### sdks\_lpr\_ipfilter\_list\_delete

#### int sdks\_lpr\_ipfilter\_list\_delete(unsigned int handle, int chn, char \*p\_param);

#### Parameter

handle: Return value of the interface for sdks\_dev\_conn.

chn： Channel number.

p\_param：Input parameters, expressed in json format.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | | Types | Optional / required | description | |
| Parameters | | | | | E | M | The message begins | |
|  | Version | | | | A | M | Version information 1.0 | |
| ChannelId | | | | E | M |  | |
|  | | Id | | A | M |  | |
| LprIpfilterList | | | | E | M |  | |
|  |  | | Text | A | M | License plate number, 5~30 characters, can only contain English characters and numbers, otherwise the setting is invalid |
| Type | A | M | List type, 0: blacklist, 1: white list |
| StartTime | A | M | Validity start time, valid before the current time, UTC timestamp, based on device time, in seconds |
|  |  |  | | EndTime | A | M | Expiration date is valid later than the current time, UTC timestamp, based on device time, in seconds |

p\_param json format is：

{

"LprIpfilterList": [{

"Text": "BCDS6",

"Type": 1,

"StartTime": 1565539200,

"EndTime": 1573710777

}]

}

Description

#### Remove the license plate recognition black and white list.

#### Return value

0 succeeded; other values failed.

### sdks\_lpr\_ipfilter\_list\_modify

#### sdks\_lpr\_ipfilter\_list\_modify(unsigned int handle, int chn, char \*p\_param);

#### Parameter

handle: Return value of the interface for sdks\_dev\_conn.

chn： Channel number.

p\_param：Input parameters, expressed in json format.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | | Types | Optional / required | description | |
| Parameters | | | | | E | M | The message begins | |
|  | Version | | | | A | M | Version information 1.0 | |
| ChannelId | | | | E | M |  | |
|  | | Id | | A | M |  | |
| ListOld | | | | E | M |  | |
|  | LprPlateNum | | | E | M |  |
|  | | Text | A | M | License plate number, 5~30 characters, can only contain English characters and numbers, otherwise the setting is invalid |
| Type | A | M | List type, 0: blacklist, 1: white list |
| StartTime | A | M | Validity start time, valid before the current time, UTC timestamp, based on device time, in seconds |
| EndTime | A | M | Expiration date is valid later than the current time, UTC timestamp, based on device time, in seconds |
| ListNew | | | | E | M |  |
|  | LprPlateNum | | | E | M |  |
|  | | Text | A | M | License plate number, 5~30 characters, can only contain English characters and numbers, otherwise the setting is invalid |
| Type | A | M | List type, 0: blacklist, 1: white list |
| StartTime | A | M | Validity start time, valid before the current time, UTC timestamp, based on device time, in seconds |
| EndTime | A | M | Expiration date is valid later than the current time, UTC timestamp, based on device time, in seconds |

p\_param json format is：

{

"ListOld": [{

"Text": "ABCDEFG",

"Type": 1,

"StartTime": 1566316800,

"EndTime": 1566662400

}],

"ListNew": [{

"Text": "VFTR",

"Type": 0,

"StartTime": 1566316800,

"EndTime": 1566662400

}]

}

Description

#### Modify the license plate recognition black and white list.

#### Return value

0 succeeded; other values failed.

### sdks\_get\_lpr\_ipfilter\_list\_num

#### int sdks\_get\_lpr\_ipfilter\_list\_num(unsigned int handle, int chn, char \*\*p\_result);

#### Parameter

handle: Return value of the interface for sdks\_dev\_conn.

chn： Channel number.

p\_result：Output parameters, return the number of blacklists recognized by the license plate, expressed in json format. If the returned value is non-NULL, you need to call the sdks\_free\_result function to free the memory, otherwise it will cause a memory leak.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attributes | | | Types | Optional / required | description |
| Parameters | | | E | M | The message begins |
| Version | | | A | M | Version information 1.0 |
|  | LprListSize | | E | M |  |
|  | Value | A | M | Number of black and white lists |

p\_result json format is：

{

"LprListSize": {

"Value": 3

}

}

Description

#### Get the number of black and white lists that the license plate recognizes.

#### Return value

0 succeeded; other values failed.

### sdks\_get\_lpr\_ipfilter\_list

#### int sdks\_get\_lpr\_ipfilter\_list(unsigned int handle, int chn, char \*p\_param, char \*\*p\_result);

#### Parameter

handle: Return value of the interface for sdks\_dev\_conn.

chn： Channel number.

p\_param：Input parameters, expressed in json format.

p\_result：Output parameters, return the obtained license plate recognition black and white list, expressed in json format. If the returned value is non-NULL, you need to call the sdks\_free\_result function to free the memory, otherwise it will cause a memory leak.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | Types | Optional / required | description | |
| Parameters | | | | E | M | The message begins | |
|  | Version | | | A | M | Version information 1.0 | |
| ChannelId | | | E | M |  | |
|  | | Id | A | M |  | |
| BeginIndex | | | E | M |  | |
|  | Id | | A | M | Start number |
| EndIndex | | | E | M |  |
|  | Id | | A | M | End number |

p\_param json format is：

{

"BeginIndex": 0,

"EndIndex": 10

}

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | Types | Optional / required | description | |
| Parameters | | | | E | M | The message begins | |
|  | Version | | | A | M | Version information 1.0 | |
| LprPlateNumList | | | E | M |  | |
|  | LprPlateNum | | E | M |  |
|  | Text | A | M | License plate number |
| Type | A | M | List type, 0: blacklist, 1: white list |
| StartTime | A | M | Validity start time, UTC timestamp, based on device time, in seconds |
| EndTime | A | M | Expiration date, UTC timestamp, based on device time, in seconds |

p\_result json format is：

{

"LprIpfilterList": [{

"Text": "FSD5456S",

"Type": 1,

"StartTime": 1567525878,

"EndTime": 1567612278

}, {

"Text": "FSDF",

"Type": 0,

"StartTime": 1568217064,

"EndTime": 1568735464

}, {

"Text": "VFVV",

"Type": 0,

"StartTime": 1567525849,

"EndTime": 1567612249

}]

}

Description

#### Get a blacklist of license plate recognition.

#### Return value

0 succeeded; other values failed.

### sdks\_lpr\_ipfilter\_list\_search\_open

#### int sdks\_lpr\_ipfilter\_list\_search\_open(unsigned int handle, int chn, char \*p\_param, char \*\*p\_result);

#### Parameter

handle: Return value of the interface for sdks\_dev\_conn.

chn： Channel number.

p\_param：Input parameters, expressed in json format.

p\_result：Output parameters, return the obtained license plate recognition black and white list search open response parameters, expressed in json format. If the returned value is non-NULL, you need to call the sdks\_free\_result function to free the memory, otherwise it will cause a memory leak.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | Types | Optional / required | description | |
| Parameters | | | | E | M | The message begins | |
|  | Version | | | A | M | Version information 1.0 | |
| ChannelId | | | E | M |  | |
|  | | Id | A | M |  | |
| Num | | | E | M |  | |
|  | String | | A | M | Search number |

p\_param json format is：

{

"Number": "J333"

}

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | Types | Optional / required | description | |
| Parameters | | | E | M | The message begins | |
|  | Version | | A | M | Version information 1.0 | |
| QueryHandle | | E | M |  | |
|  | Id | A | M | Query handle |
| Total | | E | M |  |
|  | Value | A | M |  |

p\_result json format is：

{

"Value": 1,

"QueryHandle ": 11

}

Description

#### Get the license plate recognition black and white list search open response parameters.

#### Return value

0 succeeded; other values failed.

### sdks\_lpr\_ipfilter\_list\_search\_get

#### int sdks\_lpr\_ipfilter\_list\_search\_get(unsigned int handle, int chn, char \*p\_param, char \*\*p\_result);

#### Parameter

handle: Return value of the interface for sdks\_dev\_conn.

chn： Channel number.

p\_param：Input parameters, expressed in json format.

p\_result：Output parameters, return the obtained license plate recognition black and white list search to obtain response parameters, expressed in json format. If the returned value is non-NULL, you need to call the sdks\_free\_result function to free the memory, otherwise it will cause a memory leak.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | Types | Optional / required | description | |
| Parameters | | | | E | M | The message begins | |
|  | Version | | | A | M | Version information 1.0 | |
| ChannelId | | | E | M |  | |
|  | | Id | A | M |  | |
| QueryHandle | | | E | M |  | |
|  | | Id | A | M | Query handle | |
| BeginIndex | | | E | M |  | |
|  | Id | | A | M | Start number |
| EndIndex | | | E | M |  |
|  | Id | | A | M | End number |

p\_param json format is：

{

"QueryHandle": 11,

"BeginIndex": 0,

"EndIndex": 100

}

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | Types | Optional / required | description | |
| Parameters | | | | E | M | The message begins | |
|  | Version | | | A | M | Version information 1.0 | |
| LprPlateNumList | | | E | M |  | |
|  | LprPlateNum | | E | M |  |
|  | Text | A | M | License plate number |
| Type | A | M | List type, 0: blacklist, 1: white list |
| StartTime | A | M | Validity start time, UTC timestamp, based on device time, in seconds |
| EndTime | A | M | Expiration date, UTC timestamp, based on device time, in seconds |

p\_result json format is：

{

"LprIpfilterList": [{

"Text": "J333",

"Type": 0,

"StartTime": 1568217064,

"EndTime": 1568735464

}]

}

Description

#### Obtain the license plate recognition black and white list search to obtain the response parameters.

#### Return value

0 succeeded; other values failed.

### sdks\_lpr\_ipfilter\_list\_search\_close

#### int sdks\_lpr\_ipfilter\_list\_search\_close(unsigned int handle, int chn, char \*p\_param); Parameter

handle: Return value of the interface for sdks\_dev\_conn.

chn： Channel number.

p\_param：Input parameters, expressed in json format.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | Types | Optional / required | description | |
| Parameters | | | | E | M | The message begins | |
|  | Version | | | A | M | Version information 1.0 | |
| ChannelId | | | E | M |  | |
|  | | Id | A | M |  | |
| QueryHandle | | | E | M |  | |
|  | Id | | A | M | Query handle |

p\_param json format is：

{

"QueryHandle": 11

}

Description

#### The license plate recognition black and white list search is off.

#### Return value

0 succeeded; other values failed.

### sdks\_lpr\_ipfilter\_list\_file\_download

#### int sdks\_lpr\_ipfilter\_list\_file\_download(unsigned int handle, int chn, char \*p\_param);

#### Parameter

handle: Return value of the interface for sdks\_dev\_conn.

chn： Channel number.

p\_param：Input parameters, expressed in json format.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | Types | Optional / required | description | |
| Parameters | | | E | M | The message begins | |
|  | Version | | A | M | Version information 1.0 | |
| OperatorType | | E | M |  | |
|  | OperatorType | A | M | Generally set to 1 |
|  | CameraID | A | M | Generally set to 1 |
|  | PTZID | A | M | Generally set to 0 |
| Type | | E | M |  |
|  | Value | A | M | This value can only be set to 0 at this time. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attributes | | | Types | Optional / required | description |
| Parameters | | | E | M | The message begins |
| Version | | | A | M | Version information 1.0 |
|  | FileName | | E | M |  |
|  | FileName | A | M | File name, the file path is the path under the device, and can be written, generally set to /tmp/LprList.tar |

p\_param json format is：

{

"Value": 0,

"OperatorTypeList": {

"OperatorType": 1,

"CameraID": 1,

"PTZID": 0

},

"FileName": "/tmp/LprList.tar"

}

Description

#### License plate recognition black and white list file download.

#### Return value

0 succeeded; other values failed.

### sdks\_get\_ai\_multi\_object\_detect\_param

#### int sdks\_get\_ai\_multi\_object\_detect\_param(unsigned int handle, int chn, char \*\*p\_result);

#### Parameter

handle: Return value of the interface for sdks\_dev\_conn.

chn： Channel number.

p\_result：Output parameters, return the acquired AI multi-target detection parameters, expressed in json format. If the returned value is non-NULL, you need to call the sdks\_free\_result function to free the memory, otherwise it will cause a memory leak.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | | | Types | Optional / required | description |
| Parameters | | | | | | E | M | The message begins |
| Version | | | | | | A | M | Version information 1.0 |
|  | Result | | | | | E | M |  |
|  | Code | | | | A | M | Get the result (Code):  0: OK  For other errors, please refer to the front end error code table. |
| FaceDetectParam | | | | | E | M |  |
|  | FaceEnable | | | | A | M | Face detection enable |
| UpBodyEnable | | | | A | M | Head and shoulder detection enable |
| FullBodyEnable | | | | A | M | Humanoid detection enable |
| PlateEnable | | | | A | M | License plate detection enable |
| VehicleEnable | | | | A | M | Vehicle detection enable |
| YawDegree | | | | A | M | Side angle [0,90] |
| TiltDegree | | | | A | M | Bevel [0,90] |
| PitchDegree | | | | A | M | Elevation angle [0,90] (temporarily reserved) |
| SnapPictureMode | | | | A | M | Capture mode. 0: Timing. 1: optimal |
| SnapPictureNum | | | | A | M | Number of shots (temporary support) |
| ConsumePictrueMode | | | | A | M | Reserved,  Consumption picture mode. 3: Big picture. 4: Small picture  It is only used by the Antelope Cloud Platform Agreement. Consumption big picture or small picture, determined by another interface |
| ShowObjectEnable | | | | A | M | Video stream overlay recognition box enabled |
| ShowAreaEnable | | | | A | M | Video stream overlay detection area enabled (thermal imaging support only) |
| FtpUploadEnable | | | | A | M | FTP upload thumbnail enable |
| FtpUploadFullViewEnable | | | | A | M | FTP upload big picture enable |
| Reliability | | | | A | M | Confidence [1,99]. Take 90 high, take 60, take 30 low |
| FaceMinPixelWidth | | | | A | M | Face detection minimum pixel [30,300] |
| FaceMaxPixelWidth | | | | A | M | Face detection maximum pixel [300, 2000] |
| UpBodyMinPixelWidth | | | | A | M | Head and shoulder detection minimum pixel [30,300] |
| UpBodyMaxPixelWidth | | | | A | M | Head and shoulder detection maximum pixel [300, 2000] |
| FullBodyMinPixelWidth | | | | A | M | Humanoid detection minimum pixel [30,300] |
| FullBodyMaxPixelWidth | | | | A | M | Humanoid detection maximum pixel [300, 2000] |
| PlateMinPixelWidth | | | | A | M | License plate detection minimum pixel [30,300] |
| PlateMaxPixelWidth | | | | A | M | License plate detection maximum pixel [300, 2000] |
| VehicleMinPixelWidth | | | | A | M | Vehicle detection minimum pixel [30,300] |
| VehicleMaxPixelWidth | | | | A | M | Vehicle detection maximum pixel [300, 2000] |
| PictureQuality | | | | A | M | Jpg picture quality [20-99], high [80], medium [60], low [30] |
| UploadInterval | | | | A | M | Capture interval (seconds) [1,10] |
| ScheduleTimeParam | | | | E | M | Arming time parameter |
|  | ScheduleTimeList | | | E | M | Arming time list |
|  | ScheduleTime | | E | M | Arming time |
|  | WeekDay | A | M | The number of days in a week |
| StartTime | A | M | Start time |
| EndTime | A | M | End Time |
|  |  | PolygonAreaList | | | | A | M | Detection area list |
|  | PolygonArea | | | A | M | Detection area (polygon) |
|  | SNPoint | | A | M | List of detection area coordinate points (percentage and sorted counterclockwise) |
|  | Xval | A | M | Abscissa |
| Yval | A | M | Y-axis |

p\_result json format is：

{

"AIMultiObjectDetectParam": {

"FaceEnable": true,

"UpBodyEnable": false,

"FullBodyEnable": false,

"PlateEnable": false,

"VehicleEnable": false,

"PitchDegree": 60,

"YawDegree": 60,

"TiltDegree": 30,

"SnapPictureMode": 0,

"SnapPictureNum": 1,

"ConsumePictrueMode": 0,

"ShowObjectEnable": true,

"ShowAreaEnable": true,

"FtpUploadEnable": true,

"FtpUploadFullViewEnable": true,

"UploadInterval": 5,

"Reliability": 60,

"FaceMinPixelWidth": 70,

"FaceMaxPixelWidth": 1000,

"UpBodyMinPixelWidth": 30,

"UpBodyMaxPixelWidth": 1000,

"FullBodyMinPixelWidth": 30,

"FullBodyMaxPixelWidth": 1000,

"PlateMinPixelWidth": 30,

"PlateMaxPixelWidth": 1000,

"VehicleMinPixelWidth": 30,

"VehicleMaxPixelWidth": 1000,

"PictureQuality": 60,

"LYFlag": false,

"AlgoVersion": "V1.4.1.1",

"ScheduleTimeList": [{

"WeekDay": 0,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 1,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 2,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 3,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 4,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 5,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 6,

"StartTime": 0,

"EndTime": 86400

}],

"PolygonAreaList": [{

"Xval": 0,

"Yval": 0

}, {

"Xval": 0,

"Yval": 99.500000

}, {

"Xval": 99.500000,

"Yval": 99.500000

}, {

"Xval": 99.500000,

"Yval": 0

}],

"TemperatureAlarmDeploymentList": [{

"SourceType": 29,

"SourceID": 1,

"AlarmRule": 2,

"AlarmThreshold": 30,

"AlarmInterval": 10,

"ScheduleTimeList": [{

"WeekDay": 0,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 1,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 2,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 3,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 4,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 5,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 6,

"StartTime": 0,

"EndTime": 86400

}],

"AlarmLinkageParamList": [{

"ActionType": 1,

"ActionID": 1

}, {

"ActionType": 1,

"ActionID": 2

}, {

"ActionType": 7,

"ActionID": 1

}],

"AlarmOutActionList": [],

"RecordActionParamList": [],

"PTZActionParamList": []

}]

}

}

**Description**

Get AI multi-target detection parameters(including license plate and face detection parameters).

#### Return value

0 succeeded; other values failed.

### sdks\_set\_ai\_multi\_object\_detect\_param

#### int sdks\_set\_ai\_multi\_object\_detect\_param(unsigned int handle, int chn, char \*p\_param); Parameter

handle: Return value of the interface for sdks\_dev\_conn.

chn： Channel number.

p\_param：Input parameters, expressed in json format.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | | | Types | Optional / required | description |
| Parameters | | | | | | E | M | The message begins |
| Version | | | | | | A | M | Version information 1.0 |
|  | Result | | | | | E | M |  |
|  | Code | | | | A | M | Get the result (Code):  0: OK  For other errors, please refer to the front end error code table. |
| FaceDetectParam | | | | | E | M |  |
|  | FaceEnable | | | | A | M | Face detection enable |
| UpBodyEnable | | | | A | M | Head and shoulder detection enable |
| FullBodyEnable | | | | A | M | Humanoid detection enable |
| PlateEnable | | | | A | M | License plate detection enable |
| VehicleEnable | | | | A | M | Vehicle detection enable |
| YawDegree | | | | A | M | Side angle [0,90] |
| TiltDegree | | | | A | M | Bevel [0,90] |
| PitchDegree | | | | A | M | Elevation angle [0,90] (temporarily reserved) |
| SnapPictureMode | | | | A | M | Capture mode. 0: Timing. 1: optimal |
| SnapPictureNum | | | | A | M | Number of shots (temporary support) |
| ConsumePictrueMode | | | | A | M | Reserved,  Consumption picture mode. 3: Big picture. 4: Small picture  It is only used by the Antelope Cloud Platform Agreement. Consumption big picture or small picture, determined by another interface |
| ShowObjectEnable | | | | A | M | Video stream overlay recognition box enabled |
| ShowAreaEnable | | | | A | M | Video stream overlay detection area enabled (thermal imaging support only) |
| FtpUploadEnable | | | | A | M | FTP upload thumbnail enable |
| FtpUploadFullViewEnable | | | | A | M | FTP upload big picture enable |
| Reliability | | | | A | M | Confidence [1,99]. Take 90 high, take 60, take 30 low |
| FaceMinPixelWidth | | | | A | M | Face detection minimum pixel [30,300] |
| FaceMaxPixelWidth | | | | A | M | Face detection maximum pixel [300, 2000] |
| UpBodyMinPixelWidth | | | | A | M | Head and shoulder detection minimum pixel [30,300] |
| UpBodyMaxPixelWidth | | | | A | M | Head and shoulder detection maximum pixel [300, 2000] |
| FullBodyMinPixelWidth | | | | A | M | Humanoid detection minimum pixel [30,300] |
| FullBodyMaxPixelWidth | | | | A | M | Humanoid detection maximum pixel [300, 2000] |
| PlateMinPixelWidth | | | | A | M | License plate detection minimum pixel [30,300] |
| PlateMaxPixelWidth | | | | A | M | License plate detection maximum pixel [300, 2000] |
| VehicleMinPixelWidth | | | | A | M | Vehicle detection minimum pixel [30,300] |
| VehicleMaxPixelWidth | | | | A | M | Vehicle detection maximum pixel [300, 2000] |
| PictureQuality | | | | A | M | Jpg picture quality [20-99], high [80], medium [60], low [30] |
| UploadInterval | | | | A | M | Capture interval (seconds) [1,10] |
| ScheduleTimeParam | | | | E | M | Arming time parameter |
|  | ScheduleTimeList | | | E | M | Arming time list |
|  | ScheduleTime | | E | M | Arming time |
|  | WeekDay | A | M | The number of days in a week |
| StartTime | A | M | Start time |
| EndTime | A | M | End Time |
|  |  | PolygonAreaList | | | | A | M | Detection area list |
|  | PolygonArea | | | A | M | Detection area (polygon) |
|  | SNPoint | | A | M | List of detection area coordinate points (percentage and sorted counterclockwise) |
|  | Xval | A | M | Abscissa |
| Yval | A | M | Y-axis |

p\_param json format is：

{

"AIMultiObjectDetectParam": {

"FaceEnable": true,

"UpBodyEnable": false,

"FullBodyEnable": false,

"PlateEnable": false,

"VehicleEnable": false,

"PitchDegree": 60,

"YawDegree": 60,

"TiltDegree": 30,

"SnapPictureMode": 0,

"SnapPictureNum": 1,

"ConsumePictrueMode": 0,

"ShowObjectEnable": true,

"ShowAreaEnable": true,

"FtpUploadEnable": true,

"FtpUploadFullViewEnable": true,

"UploadInterval": 5,

"Reliability": 60,

"FaceMinPixelWidth": 70,

"FaceMaxPixelWidth": 1000,

"UpBodyMinPixelWidth": 30,

"UpBodyMaxPixelWidth": 1000,

"FullBodyMinPixelWidth": 30,

"FullBodyMaxPixelWidth": 1000,

"PlateMinPixelWidth": 30,

"PlateMaxPixelWidth": 1000,

"VehicleMinPixelWidth": 30,

"VehicleMaxPixelWidth": 1000,

"PictureQuality": 60,

"LYFlag": false,

"AlgoVersion": "V1.4.1.1",

"ScheduleTimeList": [{

"WeekDay": 0,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 1,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 2,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 3,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 4,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 5,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 6,

"StartTime": 0,

"EndTime": 86400

}],

"PolygonAreaList": [{

"Xval": 0,

"Yval": 0

}, {

"Xval": 0,

"Yval": 99.500000

}, {

"Xval": 99.500000,

"Yval": 99.500000

}, {

"Xval": 99.500000,

"Yval": 0

}],

"TemperatureAlarmDeploymentList": [{

"SourceType": 29,

"SourceID": 1,

"AlarmRule": 2,

"AlarmThreshold": 30,

"AlarmInterval": 10,

"ScheduleTimeList": [{

"WeekDay": 0,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 1,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 2,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 3,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 4,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 5,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 6,

"StartTime": 0,

"EndTime": 86400

}],

"AlarmLinkageParamList": [{

"ActionType": 1,

"ActionID": 1

}, {

"ActionType": 1,

"ActionID": 2

}, {

"ActionType": 7,

"ActionID": 1

}],

"AlarmOutActionList": [],

"RecordActionParamList": [],

"PTZActionParamList": []

}]

}

}

**Description**

Setting AI multi-target detection parameters(including license plate and face detection parameters).

#### Return value

0 succeeded; other values failed.

### sdks\_get\_ai\_multi\_object\_detect\_ability

#### int sdks\_get\_ai\_multi\_object\_detect\_ability(unsigned int handle, int chn, char \*\*p\_result); Parameter

handle: Return value of the interface for sdks\_dev\_conn.

chn： Channel number.

p\_result：Output parameters, the ability to return the acquired AI multi-target detection parameter configuration, expressed in json format. If the returned value is non-NULL, you need to call the sdks\_free\_result function to free the memory, otherwise it will cause a memory leak.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attributes | | | Types | Optional / required | description |
| Parameters | | | E | M | The message begins |
| Version | | | A | M | Version information 1.0 |
|  | Result | | E | M |  |
|  | Code | A | M | Get the result (Code):  0: OK  For other errors, please refer to the front end error code table. |
| AIMultiObjectDetectAbility | | E | M |  |
|  | FaceEnable | A | M | Face detection configuration open flag |
| UpBodyEnable | A | M | Head and shoulder detection configuration open flag |
| FullBodyEnable | A | M | Humanoid detection configuration open flag |
| PlateEnable | A | M | License plate detection configuration open flag |
| VehicleEnable | A | M | Vehicle detection configuration open flag |
| PitchDegreeEnable | A | M | Elevation angle [0,90] (temporarily reserved) |
| YawDegreeEnable | A | M | Side angle [0,90] |
| TiltDegreeEnable | A | M | Bevel [0,90] |
| ShowObjectEnable | A | M | Whether to open the target tracking information (draw a rectangle on the target) |
| ShowAreaEnable | A | M | Whether the video prints the detection area |
| FaceMaxPixelWidthEnable | A | M | Face maximum size configuration open flag |
| UpBodyMaxPixelWidthEnable | A | M | Head and shoulder maximum size configuration open sign |
| FullBodyMaxPixelWidthEnable | A | M | Whole body (human body) maximum size configuration open flag |
| PlateMaxPixelWidthEnable | A | M | License plate maximum size configuration open sign |
| VehicleMaxPixelWidthEnable | A | M | Vehicle maximum size configuration open sign |
| SnapPictureModeList | A | M | List of supported snap modes |

p\_result json format is：

{

"AIMultiObjectDetectAbility": {

"FaceEnable": false,

"UpBodyEnable": false,

"FullBodyEnable": false,

"PlateEnable": true,

"VehicleEnable": false,

"PitchDegreeEnable": false,

"YawDegreeEnable": false,

"TiltDegreeEnable": false,

"ShowObjectEnable": false,

"ShowAreaEnable": false,

"FaceMaxPixelWidthEnable": false,

"UpBodyMaxPixelWidthEnable": false,

"FullBodyMaxPixelWidthEnable": false,

"PlateMaxPixelWidthEnable": false,

"VehicleMaxPixelWidthEnable": false,

"SnapPictureModeList": [{

"SnapPictureMode": 0

}, {

"SnapPictureMode": 1

}]

}

}

Description

#### Ability to acquire AI multi-target detection parameter configurations.

#### Return value

0 succeeded; other values failed.

## 25 . Intelligent Analysis

### sdks\_get\_ia\_version

#### Define

#### int sdks\_get\_ia\_version(unsigned int handle, int chn, char \*\*p\_result)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number.

p\_result：The output parameters, expressed in json format, return the acquired smart analysis version information.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | | Types | Optional / required | description |
| Parameters | | | | | E | M |  |
|  | IntelligenceLibInfo | | | | E | M |  |
|  | Version | | | A | M | version |
| StreamID | | | A | M | Reserved |
| LIB | | | E | M |  |
|  | ID | | A | M | Algorithm library ID, the ID value of each capability needs to be consistent with this value |
| enable | |  |  | The algorithm library uses flags |
| Resolution | | E | M |  |
|  | videoWidth | A | M | Video source analysis width |
| videoHeight | A | M | High video source analysis |
| Common | | E | M |  |
|  | totalAnalyseRegionNum | A | M | Intelligent analysis of the number of all areas and |
| MaxEnableFunsNum | A | M | Reserved |

p\_result the json format is：

{

"Version": 1,

"StreamID": 1,

"LIB": [{

"ID": 5,

"enable": false,

"videoWidth": 352,

"videoHeight": 288,

"totalAnalyseRegionNum": 32,

"MaxEnableFunsNum": 0

}, {

"ID": 6,

"enable": true,

"videoWidth": 352,

"videoHeight": 288,

"totalAnalyseRegionNum": 0,

"MaxEnableFunsNum": 0

}]

}

**Description**

#### Get intelligent analysis version information.

#### Return value

0 succeeded; other values failed.

### sdks\_get\_ia\_perimeter\_ability

#### Define

#### int sdks\_get\_ia\_perimeter\_ability(unsigned int handle, int chn, char \*\*p\_result)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number.

p\_result：The output parameters, expressed in json format, return the acquired intrusion configuration capabilities.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | Types | Optional / required | description |
| Parameters | | | | E | M |  |
|  | PerimeterAbility | | | E | M | Intrusion |
|  | LibId | | A | M | Algorithm library ID |
| enable | | A | M | Whether to configure intrusion detection capability options |
| Region | | E | M | Drawing detection area ability |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | edgeNum | A | M | Limiting a region with several edges |
|  | maxRegionNum | A | M | Number of restricted areas |
| DetailGet | | E | M | The backend gets the details of the test results. |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
| OSDDrawLine | | E | M | Print the detection area in the video |
|  | enable | A | M | Whether to enable this feature |
| TargetTypeConstrain | | E | M | Limit target type: person\car\person and car |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | min | A | M | Minimum value |
|  | max | A | M | Maximum value |
| TargetType | | E | M | Target type |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | min | A | M | Minimum value |
|  | max | A | M | Maximum value |
| TargetSizeConstrain | | E | M | Limit target size |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | min | A | M | Minimum value |
|  | max | A | M | Maximum value |
| MinTargetSize | | E | M | Minimum target size |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | min | A | M | Minimum value |
|  | max | A | M | Maximum value |
|  | unit | A | M |  |
| MaxTargetSize | | E | M | Maximum target size |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | min | A | M | Minimum value |
|  | max | A | M | Maximum value |
|  | unit | A | M |  |
| Sensitivity | | E | M | Sensitivity, not used |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | min | A | M | Minimum value |
|  | max | A | M | Maximum value |

p\_result the json format is：

{

"LibId": 6,

"Enable": true,

"RegionEnable": true,

"EdgeNum": 8,

"MaxRegionNum": 8,

"DetailGetEnable": true,

"OSDLineEnable": true,

"TargetTypeConstrainEnable": true,

"TargetTypeConstrainMax": 1,

"TargetTypeConstrainMin": 0,

"TargetTypeEnable": true,

"TargetTypeMax": 2,

"TargetTypeMin": 0,

"TargetSizeConstrainEnable": true,

"TargetSizeConstrainMax": 1,

"TargetSizeConstrainMin": 0,

"MinTargetSizeEnable": true,

"MinTargetSizeMax": 1000000,

"MinTargetSizeMin": 0,

"MinTargetSizeUnit": "cm2",

"MaxTargetSizeEnable": true,

"MaxTargetSizeMax": 1000000,

"MaxTargetSizeMin": 0,

"MaxTargetSizeUnit": "cm2",

"SensitivityEnable": false,

"SensitivityMax": 0,

"SensitivityMin": 0

}

**Description**

Get intrusion configuration capabilities。

#### Return value

0 succeeded; other values failed.

### sdks\_get\_ia\_perimeter\_param

#### Define

#### int sdks\_get\_ia\_perimeter\_param(unsigned int handle, int chn, char \*\*p\_result)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number.

p\_result：The output parameter, expressed in json format, returns the acquired intrusion configuration parameters.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | | | | | | | Types | Optional / required | description |
| Parameters | | | | | | | | | | E | M |  |
|  | PerimeterParamList | | | | | | | | | E | M | Intrusion parameter |
|  | PerimeterParam | | | | | | | | E | M |  |
|  | IsGetDetail | | | | | | | A | M | Upload detection result information switch |
| IntelligentEvent | | | | | | | E | M |  |
|  | SourceType | | | | | | A | M | Alarm source type,see[1.8 Alarm source type](#_1.8_Alarm_source) |
| SourceID | | | | | | A | M | Alarm source ID |
| SourceName | | | | | | A | M | Alarm source name |
| EnableFlag | | | | | | A | M | Open tag |
| AlarmInterval | | | | | | A | M | Alarm interval |
| DrawEnable | | | | | | A | M | Print detection area open mark in video |
| ScheduleTimeParam | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
|  | ScheduleTimeList | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
|  | | ScheduleTime | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| AlarmLinkageParamList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| PTZActionParamList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| AlarmOutActionList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| RecordActionParamList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| PerimeterRegionParamList | | | | | | | E | M |  |
|  | PerimeterRegionParam | | | | | | E | M |  |
|  | | Sensitivity | | | | A | M | Sensitivity |
| TargetTypeConstrain | | | | A | M | Limited target open sign |
| TargetType | | | | A | M | Target type |
| TargetSizeConstrain | | | | A | M | Limited target size open flag |
| MinTargetSize | | | | A | M | smallest size |
| MaxTargetSize | | | | A | M | biggest size |
| Region | | | | E | M | region |
|  | | SNPoint | | E | M | point |
|  | Xval | A | M | X value percentage floating point |
| Yval | A | M | Y value percentage floating point |

p\_result the json format is：

{

"PerimeterParamList": [{

"PerimeterParam": {

"IsGetDetail": false

},

"IntelligentEvent": {

"SourceType": 8,

"SourceID": 1,

"SourceName": "",

"EnableFlag": false,

"AlarmInterval": 0,

"DrawEnable": false,

"ScheduleTimeList": [{

"WeekDay": 0,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 1,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 2,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 3,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 4,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 5,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 6,

"StartTime": 0,

"EndTime": 86400

}],

"AlarmLinkageParamList": [],

"PTZActionParamList": [],

"AlarmOutActionList": [],

"RecordActionParamList": []

},

"PerimeterRegionParamList": [{

"Sensitivity": 5,

"TargetTypeConstrain": false,

"TargetType": 0,

"TargetSizeConstrain": false,

"MinTargetSize": 1111,

"MaxTargetSize": 9999,

"SNPoint": [{

"Xval": 0,

"Yval": 0

}, {

"Xval": 0,

"Yval": 99.500000

}, {

"Xval": 99.500000,

"Yval": 99.500000

}, {

"Xval": 99.500000,

"Yval": 0

}]

}]

}]

}

**Description**

Get the intrusion configuration parameters.

#### Return value

0 succeeded; other values failed.

### sdks\_set\_ia\_perimeter\_param

#### Define

#### int sdks\_set\_ia\_perimeter\_param(unsigned int handle, int chn, char \*p\_param)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number.

p\_param：Input parameters, expressed in json format.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | | | | | | | Types | Optional / required | description |
| Parameters | | | | | | | | | | E | M |  |
|  | PerimeterParamList | | | | | | | | | E | M | Intrusion parameters, up to 2 |
|  | PerimeterParam | | | | | | | | E | M |  |
|  | IsGetDetail | | | | | | | A | M | Upload detection result information switch |
| IntelligentEvent | | | | | | | E | M |  |
|  | SourceType | | | | | | A | M | Alarm source type,see[1.8 Alarm source type](#_1.8_Alarm_source) |
| SourceID | | | | | | A | M | Alarm source ID |
| SourceName | | | | | | A | M | Alarm source name |
| EnableFlag | | | | | | A | M | Open tag |
| AlarmInterval | | | | | | A | M | Alarm interval |
| DrawEnable | | | | | | A | M | Print detection area open mark in video |
| ScheduleTimeParam | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
|  | ScheduleTimeList | | | | | E | M |  |
|  | | ScheduleTime | | | E |  |  |
| AlarmLinkageParamList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| PTZActionParamList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| AlarmOutActionList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| RecordActionParamList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| PerimeterRegionParamList | | | | | | | E | M |  |
|  | PerimeterRegionParam | | | | | | E | M |  |
|  | | Sensitivity | | | | A | M | Sensitivity |
| TargetTypeConstrain | | | | A | M | Limited target open sign |
| TargetType | | | | A | M | Target type |
| TargetSizeConstrain | | | | A | M | Limited target size open flag |
| MinTargetSize | | | | A | M | smallest size |
| MaxTargetSize | | | | A | M | biggest size |
| Region | | | | E | M | region |
|  | | SNPoint | | E | M | point |
|  | Xval | A | M | X value percentage floating point |
| Yval | A | M | Y value percentage floating point |

p\_param the json format is：

{

"PerimeterParamList": [{

"PerimeterParam": {

"IsGetDetail": false

},

"IntelligentEvent": {

"SourceType": 8,

"SourceID": 1,

"SourceName": "",

"EnableFlag": false,

"AlarmInterval": 0,

"DrawEnable": false,

"ScheduleTimeList": [{

"WeekDay": 0,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 1,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 2,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 3,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 4,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 5,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 6,

"StartTime": 0,

"EndTime": 86400

}],

"AlarmLinkageParamList": [{

"ActionType": 1,

"ActionID": 1

}, {

"ActionType": 2,

"ActionID": 1

}, {

"ActionType": 4,

"ActionID": 1

}, {

"ActionType": 7,

"ActionID": 1

}, {

"ActionType": 3,

"ActionID": 2

}],

"PTZActionParamList": [],

"AlarmOutActionList": [],

"RecordActionParamList": []

},

"PerimeterRegionParamList": [{

"Sensitivity": 5,

"TargetTypeConstrain": false,

"TargetType": 0,

"TargetSizeConstrain": false,

"MinTargetSize": 1111,

"MaxTargetSize": 9999,

"SNPoint": [{

"Xval": 0,

"Yval": 0

}, {

"Xval": 0,

"Yval": 99.500000

}, {

"Xval": 99.500000,

"Yval": 99.500000

}, {

"Xval": 99.500000,

"Yval": 0

}]

}]

}]

}

**Description**

#### Set the intrusion configuration parameters.

#### Return value

0 succeeded; other values failed.

### sdks\_get\_ia\_svf\_ability

#### Define

#### int sdks\_get\_ia\_svf\_ability(unsigned int handle, int chn, char \*\*p\_result)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number.

p\_result：The output parameters, expressed in json format, return the acquired alert line configuration capabilities.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | Types | Optional / required | description |
| Parameters | | | | E | M |  |
|  | TripWireAbility | | | E | M | Warning line |
|  | LibId | | A | M | Algorithm library ID |
| enable | | A | M | Whether to configure the cordon detection capability option |
| Region | | E | M | Drawing detection area ability |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | edgeNum | A | M | Limiting a region with several edges |
|  | maxRegionNum | A | M | Number of restricted areas |
| DetailGet | | E | M | The backend gets the details of the test results. |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
| OSDDrawLine | | E | M | Print the detection area in the video |
|  | enable | A | M | Whether to enable this feature |
| TriggerDirection | | E | M | Warning line prohibition direction angle |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
| Bidirection | | E | M | Double direction |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
| TargetTypeConstrain | | E | M | Limit target type: person\car\person and car |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | min | A | M | Minimum value |
|  | max | A | M | Maximum |
| TargetType | | E | M | Target type |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | min | A | M | Minimum value |
|  | max | A | M | Maximum value |
| TargetSizeConstrain | | E | M | Limit target size |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | min | A | M | Minimum value |
|  | max | A | M | Maximum value |
| MinTargetSize | | E | M | Minimum target size |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | min | A | M | Minimum value |
|  | max | A | M | Maximum value |
|  | unit | A | M |  |
| MaxTargetSize | | E | M | Maximum target size |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | min | A | M | Minimum value |
|  | max | A | M | Maximum value |
|  | unit | A | M |  |

p\_result the json format is：

{

"LibId": 6,

"Enable": true,

"RegionEnable": true,

"EdgeNum": 8,

"MaxRegionNum": 8,

"DetailGetEnable": true,

"OSDLineEnable": true,

"TriggerDirectionEnable": true,

"BidirectionEnable": true,

"TargetTypeConstrainEnable": true,

"TargetTypeConstrainMax": 1,

"TargetTypeConstrainMin": 0,

"TargetTypeEnable": true,

"TargetTypeMax": 2,

"TargetTypeMin": 0,

"TargetSizeConstrainEnable": true,

"TargetSizeConstrainMax": 1,

"TargetSizeConstrainMin": 0,

"MinTargetSizeEnable": true,

"MinTargetSizeMax": 1000000,

"MinTargetSizeMin": 0,

"MinTargetSizeUnit": "cm2",

"MaxTargetSizeEnable": true,

"MaxTargetSizeMax": 1000000,

"MaxTargetSizeMin": 0,

"MaxTargetSizeUnit": "cm2"

}

**Description**

#### Get alert line configuration capabilities.

#### Return value

0 succeeded; other values failed.

### sdks\_get\_ia\_svf\_param

#### Define

#### int sdks\_get\_ia\_svf\_param(unsigned int handle, int chn, char \*\*p\_result)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number.

p\_result：Output parameters, expressed in json format, return the obtained warning line configuration parameters。

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | | | | | | Types | Optional / required | description |
| Parameters | | | | | | | | | E | M |  |
|  | TripWireParamList | | | | | | | | E | M |  |
|  | TripWireParam | | | | | | | E | M | Warning line parameter |
|  | IsGetDetail | | | | | | A | M | Upload detection result information switch |
| IntelligentEvent | | | | | | E | M |  |
|  | SourceType | | | | | A | M | Alarm source type,see[1.8 Alarm source type](#_1.8_Alarm_source) |
| SourceID | | | | | A | M | Alarm source ID |
| SourceName | | | | | A | M | Alarm source name |
| EnableFlag | | | | | A | M | Open tag |
| AlarmInterval | | | | | A | M | Alarm interval |
| DrawEnable | | | | | A | M | Print detection area open mark in video |
| ScheduleTimeParam | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
|  | ScheduleTimeList | | | | E | M |  |
|  | | ScheduleTime | | E |  |  |
| AlarmLinkageParamList | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| PTZActionParamList | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| AlarmOutActionList | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| RecordActionParamList | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| TripWireRegionParamList | | | | | | E | M |  |
|  | TripWireRegionParam | | | | | E | M |  |
|  | | TargetTypeConstrain | | | A | M | Limited target open sign |
| TargetType | | | A | M | Target type |
| TargetSizeConstrain | | | A | M | Limited target size open flag |
| MinTargetSize | | | A | M | smallest size |
| MaxTargetSize | | | A | M | biggest size |
| IsBidirection | | | A | M | Double direction sign |
| TriggerDirection | | | A | M | direction |
| TripWireLine | | | E | M | line |
|  | | LineCrossStartX | A | M | Starting point X |
| LineCrossStartY | A | M | Starting point Y |
| LineCrossEndX | A | M | End point X |
| LineCrossEndY | A | M | End point Y |

p\_result the json format is：

{

"TripWireParamList": [{

"TripWireParam": {

"IsGetDetail": false

},

"IntelligentEvent": {

"SourceType": 9,

"SourceID": 1,

"SourceName": "",

"EnableFlag": false,

"AlarmInterval": 0,

"DrawEnable": false,

"ScheduleTimeList": [{

"WeekDay": 0,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 1,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 2,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 3,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 4,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 5,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 6,

"StartTime": 0,

"EndTime": 86400

}],

"AlarmLinkageParamList": [{

"ActionType": 1,

"ActionID": 1

}, {

"ActionType": 2,

"ActionID": 1

}, {

"ActionType": 4,

"ActionID": 1

}, {

"ActionType": 7,

"ActionID": 1

}, {

"ActionType": 3,

"ActionID": 2

}],

"PTZActionParamList": [],

"AlarmOutActionList": [],

"RecordActionParamList": []

},

"TripWireRegionParamList": [{

"TargetTypeConstrain": false,

"TargetType": 0,

"TargetSizeConstrain": false,

"MinTargetSize": 1111,

"MaxTargetSize": 9999,

"IsBidirection": true,

"TriggerDirection": 1,

"TripWireLine": {

"LineCrossStartX": 24,

"LineCrossStartY": 21,

"LineCrossEndX": 71,

"LineCrossEndY": 25

}

}]

}]

}

**Description**

#### Get the alert line configuration parameters.

#### Return value

0 succeeded; other values failed.

### sdks\_set\_ia\_svf\_param

#### Define

#### int sdks\_set\_ia\_svf\_param(unsigned int handle, int chn, char \*p\_param)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number.

p\_param：Input parameters, expressed in json format.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | | | | | | Types | Optional / required | description |
| Parameters | | | | | | | | | E | M |  |
|  | TripWireParamList | | | | | | | | E | M |  |
|  | TripWireParam | | | | | | | E | M | Warning line parameter |
|  | IsGetDetail | | | | | | A | M | Upload detection result information switch |
| IntelligentEvent | | | | | | E | M |  |
|  | SourceType | | | | | A | M | Alarm source type,see[1.8 Alarm source type](#_1.8_Alarm_source) |
| SourceID | | | | | A | M | Alarm source ID |
| SourceName | | | | | A | M | Alarm source name |
| EnableFlag | | | | | A | M | Open tag |
| AlarmInterval | | | | | A | M | Alarm interval |
| DrawEnable | | | | | A | M | Print detection area open mark in video |
| ScheduleTimeParam | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
|  | ScheduleTimeList | | | | E | M |  |
|  | | ScheduleTime | | E |  |  |
| AlarmLinkageParamList | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| PTZActionParamList | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| AlarmOutActionList | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| RecordActionParamList | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| TripWireRegionParamList | | | | | | E | M |  |
|  | TripWireRegionParam | | | | | E | M |  |
|  | | TargetTypeConstrain | | | A | M | Limited target open sign |
| TargetType | | | A | M | Target type |
| TargetSizeConstrain | | | A | M | Limited target size open flag |
| MinTargetSize | | | A | M | smallest size |
| MaxTargetSize | | | A | M | biggest size |
| IsBidirection | | | A | M | Double direction sign |
| TriggerDirection | | | A | M | direction |
| TripWireLine | | | E | M | line |
|  | | LineCrossStartX | A | M | Starting point X |
| LineCrossStartY | A | M | Starting point Y |
| LineCrossEndX | A | M | End point X |
| LineCrossEndY | A | M | End point Y |

p\_param the json format is：

{

"TripWireParamList": [{

"TripWireParam": {

"IsGetDetail": false

},

"IntelligentEvent": {

"SourceType": 9,

"SourceID": 1,

"SourceName": "",

"EnableFlag": false,

"AlarmInterval": 0,

"DrawEnable": false,

"ScheduleTimeList": [{

"WeekDay": 0,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 1,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 2,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 3,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 4,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 5,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 6,

"StartTime": 0,

"EndTime": 86400

}],

"AlarmLinkageParamList": [{

"ActionType": 1,

"ActionID": 1

}, {

"ActionType": 2,

"ActionID": 1

}, {

"ActionType": 4,

"ActionID": 1

}, {

"ActionType": 7,

"ActionID": 1

}, {

"ActionType": 3,

"ActionID": 2

}],

"PTZActionParamList": [],

"AlarmOutActionList": [],

"RecordActionParamList": []

},

"TripWireRegionParamList": [{

"TargetTypeConstrain": false,

"TargetType": 0,

"TargetSizeConstrain": false,

"MinTargetSize": 1111,

"MaxTargetSize": 9999,

"IsBidirection": true,

"TriggerDirection": 1,

"TripWireLine": {

"LineCrossStartX": 24,

"LineCrossStartY": 21,

"LineCrossEndX": 71,

"LineCrossEndY": 25

}

}]

}]

}

**Description**

#### Set the guard line configuration parameters.

#### Return value

0 succeeded; other values failed.

### sdks\_get\_ia\_dvf\_ability

#### Define

#### int sdks\_get\_ia\_dvf\_ability(unsigned int handle, int chn, char \*\*p\_result)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number.

p\_result：The output parameters, expressed in json format, return the acquired dual alert line configuration capabilities.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | Types | Optional / required | description |
| Parameters | | | | E | M |  |
|  | MultiTripWireAbility | | | E | M | Double warning line |
|  | LibId | | A | M | Algorithm library ID |
| enable | | A | M | Whether to configure the cordon detection capability option |
| Region | | E | M | Drawing detection area ability |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | edgeNum | A | M | Limiting a region with several edges |
|  | maxRegionNum | A | M | Number of restricted areas |
| DetailGet | | E | M | The backend gets the details of the test results. |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
| OSDDrawLine | | E | M | Print the detection area in the video |
|  | enable | A | M | Whether to enable this feature |
| TriggerDirection | | E | M | Warning line prohibition direction angle |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
| TimeInterval | | E | M | Crossing the double warning time |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | min | A | M | Minimum value |
|  | max | A | M | Maximum value |
|  | unit | A | M | Unit: S seconds |
| TargetTypeConstrain | | E | M | Limit target type: person\car\person and car |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | min | A | M | Minimum value |
|  | max | A | M | Maximum value |
| TargetType | | E | M | Target type |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | min | A | M | Minimum value |
|  | max | A | M | Maximum value |
| TargetSizeConstrain | | E | M | Limit target size |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | min | A | M | Minimum value |
|  | max | A | M | Maximum value |
| MinTargetSize | | E | M | Minimum target size |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | min | A | M | Minimum value |
|  | max | A | M | Maximum value |
|  | unit | A | M |  |
| MaxTargetSize | | E | M | Maximum target size |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | min | A | M | Minimum value |
|  | max | A | M | Maximum value |
|  | unit | A | M |  |

p\_result the json format is：

{

"LibId": 6,

"Enable": true,

"RegionEnable": false,

"EdgeNum": 8,

"MaxRegionNum": 4,

"DetailGetEnable": true,

"OSDLineEnable": true,

"TriggerDirectionEnable": true,

"TimeIntervalEnable": true,

"TimeIntervalMax": 60,

"TimeIntervalMin": 1,

"TimeIntervalUnit": "S",

"TargetTypeConstrainEnable": true,

"TargetTypeConstrainMax": 1,

"TargetTypeConstrainMin": 0,

"TargetTypeEnable": true,

"TargetTypeMax": 2,

"TargetTypeMin": 0,

"TargetSizeConstrainEnable": true,

"TargetSizeConstrainMax": 1,

"TargetSizeConstrainMin": 0,

"MinTargetSizeEnable": true,

"MinTargetSizeMax": 1000000,

"MinTargetSizeMin": 0,

"MinTargetSizeUnit": "cm2",

"MaxTargetSizeEnable": true,

"MaxTargetSizeMax": 1000000,

"MaxTargetSizeMin": 0,

"MaxTargetSizeUnit": "cm2"

}

**Description**

#### Get dual alert line configuration capabilities.

#### Return value

0 succeeded; other values failed.

### sdks\_get\_ia\_dvf\_param

#### Define

#### int sdks\_get\_ia\_dvf\_param(unsigned int handle, int chn, char \*\*p\_result)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number.

p\_result：The output parameters, expressed in json format, return the acquired double alert line configuration parameters.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | | | | | | Types | Optional / required | description |
| Parameters | | | | | | | | | E | M |  |
|  | MultiTripWireParamList | | | | | | | | E | M |  |
|  | MultiTripWireParam | | | | | | | E | M | Double warning line parameter |
|  | IsGetDetail | | | | | | A | M | Upload detection result information switch |
| IntelligentEvent | | | | | | E | M |  |
|  | SourceType | | | | | A | M | Alarm source type,see[1.8 Alarm source type](#_1.8_Alarm_source) |
| SourceID | | | | | A | M | Alarm source ID |
| SourceName | | | | | A | M | Alarm source name |
| EnableFlag | | | | | A | M | Open tag |
| AlarmInterval | | | | | A | M | Alarm interval |
| DrawEnable | | | | | A | M | Print detection area open mark in video |
| ScheduleTimeParam | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
|  | ScheduleTimeList | | | | E | M |  |
|  | | ScheduleTime | | E |  |  |
| AlarmLinkageParamList | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| PTZActionParamList | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| AlarmOutActionList | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| RecordActionParamList | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| MultiTripWireRegionParamList | | | | | | E | M |  |
|  | MultiTripWireRegionParam | | | | | E | M |  |
|  | | TargetTypeConstrain | | | A | M | Limited target open sign |
| TargetType | | | A | M | Target type |
| TargetSizeConstrain | | | A | M | Limited target size open flag |
| MinTargetSize | | | A | M | smallest size |
| MaxTargetSize | | | A | M | biggest size |
| TimeInterval | | | A | M | Crossing the double warning time |
| TriggerDirection1 | | | A | M | First line direction |
| TriggerDirection2 | | | A | M | Second line direction |
| LineCross1 | | | E | M | First line coordinate |
|  | | LineCrossStartX | A | M | Starting point X |
| LineCrossStartY | A | M | Starting point Y |
| LineCrossEndX | A | M | End point X |
| LineCrossEndY | A | M | End point Y |
| LineCross2 | | | E | M | Second line coordinate |
|  | | LineCrossStartX | A | M | Starting point X |
| LineCrossStartY | A | M | Starting point Y |
| LineCrossEndX | A | M | End point X |
| LineCrossEndY | A | M | End point Y |

p\_result the json format is：

{

"MultiTripWireParamList": [{

"MultiTripWireParam": {

"IsGetDetail": false

},

"IntelligentEvent": {

"SourceType": 10,

"SourceID": 1,

"SourceName": "",

"EnableFlag": false,

"AlarmInterval": 0,

"DrawEnable": false,

"ScheduleTimeList": [{

"WeekDay": 0,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 1,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 2,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 3,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 4,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 5,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 6,

"StartTime": 0,

"EndTime": 86400

}],

"AlarmLinkageParamList": [{

"ActionType": 1,

"ActionID": 1

}, {

"ActionType": 2,

"ActionID": 1

}, {

"ActionType": 4,

"ActionID": 1

}, {

"ActionType": 7,

"ActionID": 1

}, {

"ActionType": 3,

"ActionID": 2

}],

"PTZActionParamList": [],

"AlarmOutActionList": [],

"RecordActionParamList": []

},

"MultiTripWireRegionParamList": [{

"TargetTypeConstrain": false,

"TargetType": 0,

"TargetSizeConstrain": false,

"MinTargetSize": 1111,

"MaxTargetSize": 9999,

"TimeInterval": 5,

"TriggerDirection1": 1,

"TriggerDirection2": 2,

"LineCross1": {

"LineCrossStartX": 24,

"LineCrossStartY": 21,

"LineCrossEndX": 71,

"LineCrossEndY": 25

},

"LineCross2": {

"LineCrossStartX": 26,

"LineCrossStartY": 38,

"LineCrossEndX": 109,

"LineCrossEndY": 46

}

}]

}]

}

**Description**

Get the double alert line configuration parameters.

#### Return value

0 succeeded; other values failed.

### sdks\_set\_ia\_dvf\_param

#### Define

#### int sdks\_set\_ia\_dvf\_param(unsigned int handle, int chn, char \*p\_param)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number.

p\_param：Input parameters, expressed in json format.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | | | | | | Types | Optional / required | description |
| Parameters | | | | | | | | | E | M |  |
|  | MultiTripWireParamList | | | | | | | | E | M |  |
|  | MultiTripWireParam | | | | | | | E | M | Double warning line parameter |
|  | IsGetDetail | | | | | | A | M | Upload detection result information switch |
| IntelligentEvent | | | | | | E | M |  |
|  | SourceType | | | | | A | M | Alarm source type,see[1.8 Alarm source type](#_1.8_Alarm_source) |
| SourceID | | | | | A | M | Alarm source ID |
| SourceName | | | | | A | M | Alarm source name |
| EnableFlag | | | | | A | M | Open tag |
| AlarmInterval | | | | | A | M | Alarm interval |
| DrawEnable | | | | | A | M | Print detection area open mark in video |
| ScheduleTimeParam | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
|  | ScheduleTimeList | | | | E | M |  |
|  | | ScheduleTime | | E |  |  |
| AlarmLinkageParamList | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| PTZActionParamList | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| AlarmOutActionList | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| RecordActionParamList | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| MultiTripWireRegionParamList | | | | | | E | M |  |
|  | MultiTripWireRegionParam | | | | | E | M |  |
|  | | TargetTypeConstrain | | | A | M | Limited target open sign |
| TargetType | | | A | M | Target type |
| TargetSizeConstrain | | | A | M | Limited target size open flag |
| MinTargetSize | | | A | M | smallest size |
| MaxTargetSize | | | A | M | biggest size |
| TimeInterval | | | A | M | Crossing the double warning time |
| TriggerDirection1 | | | A | M | First line direction |
| TriggerDirection2 | | | A | M | Second line direction |
| LineCross1 | | | E | M | First line coordinate |
|  | | LineCrossStartX | A | M | Starting point X |
| LineCrossStartY | A | M | Starting point Y |
| LineCrossEndX | A | M | End point X |
| LineCrossEndY | A | M | End point Y |
| LineCross2 | | | E | M | Second line coordinate |
|  | | LineCrossStartX | A | M | Starting point X |
| LineCrossStartY | A | M | Starting point Y |
| LineCrossEndX | A | M | End point X |
| LineCrossEndY | A | M | End point Y |

p\_param the json format is：

{

"MultiTripWireParamList": [{

"MultiTripWireParam": {

"IsGetDetail": false

},

"IntelligentEvent": {

"SourceType": 10,

"SourceID": 1,

"SourceName": "",

"EnableFlag": false,

"AlarmInterval": 0,

"DrawEnable": false,

"ScheduleTimeList": [{

"WeekDay": 0,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 1,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 2,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 3,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 4,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 5,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 6,

"StartTime": 0,

"EndTime": 86400

}],

"AlarmLinkageParamList": [{

"ActionType": 1,

"ActionID": 1

}, {

"ActionType": 2,

"ActionID": 1

}, {

"ActionType": 4,

"ActionID": 1

}, {

"ActionType": 7,

"ActionID": 1

}, {

"ActionType": 3,

"ActionID": 2

}],

"PTZActionParamList": [],

"AlarmOutActionList": [],

"RecordActionParamList": []

},

"MultiTripWireRegionParamList": [{

"TargetTypeConstrain": false,

"TargetType": 0,

"TargetSizeConstrain": false,

"MinTargetSize": 1111,

"MaxTargetSize": 9999,

"TimeInterval": 5,

"TriggerDirection1": 1,

"TriggerDirection2": 2,

"LineCross1": {

"LineCrossStartX": 24,

"LineCrossStartY": 21,

"LineCrossEndX": 71,

"LineCrossEndY": 25

},

"LineCross2": {

"LineCrossStartX": 26,

"LineCrossStartY": 38,

"LineCrossEndX": 109,

"LineCrossEndY": 46

}

}]

}]

}

**Description**

#### Set the dual alert line configuration parameters.

#### Return value

0 succeeded; other values failed.

### sdks\_get\_ia\_loiter\_ability

#### Define

#### int sdks\_get\_ia\_loiter\_ability(unsigned int handle, int chn, char \*\*p\_result)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number.

p\_result：The output parameters, expressed in json format, return the acquired configuration capabilities.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | Types | Optional / required | description |
| Parameters | | | | E | M |  |
|  | LoiterAbility | | | E | M | |  |  | | --- | --- | | wander |  | |
|  | LibId | | A | M | Algorithm library ID |
| enable | | A | M | Whether to configure intrusion detection capability options |
| Region | | E | M | Drawing detection area ability |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | edgeNum | A | M | Limiting a region with several edges |
|  | maxRegionNum | A | M | Number of restricted areas |
| DetailGet | | E | M | The backend gets the details of the test results. |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
| OSDDrawLine | | E | M | Print the detection area in the video |
|  | enable | A | M | Whether to enable this feature |
| AnalysisPath | | E | M | Path analysis |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | min | A | M |  |
|  | max | A | M |  |
| TargetTypeConstrain | | E | M | Limit target type: person\car\person and car |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | min | A | M | Minimum value |
|  | max | A | M | Maximum value |
| TargetType | | E | M | Target type |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | min | A | M | Minimum value |
|  | max | A | M | Maximum value |
| TargetSizeConstrain | | E | M | Limit target size |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | min | A | M | Minimum value |
|  | max | A | M | Maximum value |
| MinTargetSize | | E | M | Minimum target size |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | min | A | M | Minimum value |
|  | max | A | M | Maximum value |
|  | unit | A | M |  |
| MaxTargetSize | | E | M | Maximum target size |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | min | A | M | Minimum value |
|  | max | A | M | Maximum value |
|  | unit | A | M |  |
| MinTime | | E | M | the shortest time |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | min | A | M | Minimum value |
|  | max | A | M | Maximum value |
|  | unit | A | M | Unit: S seconds |

p\_result the json format is：

{

"LibId": 6,

"Enable": true,

"RegionEnable": true,

"EdgeNum": 8,

"MaxRegionNum": 8,

"DetailGetEnable": true,

"OSDLineEnable": true,

"AnalysisPathEnable": true,

"AnalysisPathMax": 1,

"AnalysisPathMin": 0,

"TargetTypeConstrainEnable": true,

"TargetTypeConstrainMax": 1,

"TargetTypeConstrainMin": 0,

"TargetTypeEnable": true,

"TargetTypeMax": 2,

"TargetTypeMin": 0,

"TargetSizeConstrainEnable": true,

"TargetSizeConstrainMax": 1,

"TargetSizeConstrainMin": 0,

"MinTargetSizeEnable": true,

"MinTargetSizeMax": 1000000,

"MinTargetSizeMin": 0,

"MinTargetSizeUnit": "cm2",

"MaxTargetSizeEnable": true,

"MaxTargetSizeMax": 1000000,

"MaxTargetSizeMin": 0,

"MaxTargetSizeUnit": "cm2",

"MinTimeEnable": true,

"MinTimeMax": 60,

"MinTimeMin": 5,

"MinTimeUnit": "S"

}

**Description**

#### Get wander configuration capabilities.

#### Return value

0 succeeded; other values failed.

### sdks\_get\_ia\_loiter\_param

#### Define

#### int sdks\_get\_ia\_loiter\_param(unsigned int handle, int chn, char \*\*p\_result)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number.

p\_result：The output parameters, expressed in json format, return the obtained configuration parameters.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | | | | | | | Types | Optional / required | description |
| Parameters | | | | | | | | | | E | M |  |
|  | LoiterParamList | | | | | | | | | E | M | Wander parameter |
|  | LoiterParam | | | | | | | | E | M |  |
|  | IsGetDetail | | | | | | | A | M | Parameter upload detection result information switch |
| IntelligentEvent | | | | | | | E | M |  |
|  | SourceType | | | | | | A | M | Alarm source type,see[1.8 Alarm source type](#_1.8_Alarm_source) |
| SourceID | | | | | | A | M | Alarm source ID |
| SourceName | | | | | | A | M | Alarm source name |
| EnableFlag | | | | | | A | M | Open tag |
| AlarmInterval | | | | | | A | M | Alarm interval |
| DrawEnable | | | | | | A | M | Print detection area open mark in video |
| ScheduleTimeParam | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
|  | ScheduleTimeList | | | | | E | M |  |
|  | | ScheduleTime | | | E |  |  |
| AlarmLinkageParamList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| PTZActionParamList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| AlarmOutActionList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| RecordActionParamList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| LoiterRegionParamList | | | | | | | E | M |  |
|  | LoiterRegionParam | | | | | | E | M |  |
|  | | TargetTypeConstrain | | | | A | M | Limited target open sign |
| TargetType | | | | A | M | Target type |
| TargetSizeConstrain | | | | A | M | Limited target size open flag |
| MinTargetSize | | | | A | M | smallest size |
| MaxTargetSize | | | | A | M | biggest size |
| MinTime | | | | A | M | the shortest time |
| IsAnalysisPath | | | | A | M | Analysis path open flag |
| Region | | | | E | M | region |
|  | | SNPoint | | E | M | point |
|  | Xval | A | M | X value |
| Yval | A | M | Y value |

p\_result the json format is：

{

"LoiterParamList": [{

"LoiterParam": {

"IsGetDetail": false

},

"IntelligentEvent": {

"SourceType": 11,

"SourceID": 1,

"SourceName": "",

"EnableFlag": false,

"AlarmInterval": 0,

"DrawEnable": false,

"ScheduleTimeList": [{

"WeekDay": 0,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 1,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 2,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 3,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 4,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 5,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 6,

"StartTime": 0,

"EndTime": 86400

}],

"AlarmLinkageParamList": [{

"ActionType": 1,

"ActionID": 1

}, {

"ActionType": 2,

"ActionID": 1

}, {

"ActionType": 4,

"ActionID": 1

}, {

"ActionType": 7,

"ActionID": 1

}, {

"ActionType": 3,

"ActionID": 2

}],

"PTZActionParamList": [],

"AlarmOutActionList": [],

"RecordActionParamList": []

},

"LoiterRegionParamList": [{

"TargetTypeConstrain": false,

"TargetType": 0,

"TargetSizeConstrain": false,

"MinTargetSize": 1111,

"MaxTargetSize": 9999,

"MinTime": 5,

"IsAnalysisPath": false,

"SNPoint": [{

"Xval": 0,

"Yval": 0

}, {

"Xval": 0,

"Yval": 99.500000

}, {

"Xval": 99.500000,

"Yval": 99.500000

}, {

"Xval": 99.500000,

"Yval": 0

}]

}]

}]

}

**Description**

Get the configuration parameters.

#### Return value

0 succeeded; other values failed.

### sdks\_set\_ia\_loiter\_param

#### Define

#### int sdks\_set\_ia\_loiter\_param(unsigned int handle, int chn, char \*p\_param)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number.

p\_param：Input parameters, expressed in json format.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | | | | | | | Types | Optional / required | description |
| Parameters | | | | | | | | | | E | M |  |
|  | LoiterParamList | | | | | | | | | E | M | Wander parameter |
|  | LoiterParam | | | | | | | | E | M |  |
|  | IsGetDetail | | | | | | | A | M | Parameter upload detection result information switch |
| IntelligentEvent | | | | | | | E | M |  |
|  | SourceType | | | | | | A | M | Alarm source type,see[1.8 Alarm source type](#_1.8_Alarm_source) |
| SourceID | | | | | | A | M | Alarm source ID |
| SourceName | | | | | | A | M | Alarm source name |
| EnableFlag | | | | | | A | M | Open tag |
| AlarmInterval | | | | | | A | M | Alarm interval |
| DrawEnable | | | | | | A | M | Print detection area open mark in video |
| ScheduleTimeParam | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
|  | ScheduleTimeList | | | | | E | M |  |
|  | | ScheduleTime | | | E |  |  |
| AlarmLinkageParamList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| PTZActionParamList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| AlarmOutActionList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| RecordActionParamList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| LoiterRegionParamList | | | | | | | E | M |  |
|  | LoiterRegionParam | | | | | | E | M |  |
|  | | TargetTypeConstrain | | | | A | M | Limited target open sign |
| TargetType | | | | A | M | Target type |
| TargetSizeConstrain | | | | A | M | Limited target size open flag |
| MinTargetSize | | | | A | M | smallest size |
| MaxTargetSize | | | | A | M | biggest size |
| MinTime | | | | A | M | the shortest time |
| IsAnalysisPath | | | | A | M | Analysis path open flag |
| Region | | | | E | M | region |
|  | | SNPoint | | E | M | point |
|  | Xval | A | M | X value |
| Yval | A | M | Y value |

p\_param the json format is：

{

"LoiterParamList": [{

"LoiterParam": {

"IsGetDetail": false

},

"IntelligentEvent": {

"SourceType": 11,

"SourceID": 1,

"SourceName": "",

"EnableFlag": false,

"AlarmInterval": 0,

"DrawEnable": false,

"ScheduleTimeList": [{

"WeekDay": 0,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 1,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 2,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 3,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 4,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 5,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 6,

"StartTime": 0,

"EndTime": 86400

}],

"AlarmLinkageParamList": [{

"ActionType": 1,

"ActionID": 1

}, {

"ActionType": 2,

"ActionID": 1

}, {

"ActionType": 4,

"ActionID": 1

}, {

"ActionType": 7,

"ActionID": 1

}, {

"ActionType": 3,

"ActionID": 2

}],

"PTZActionParamList": [],

"AlarmOutActionList": [],

"RecordActionParamList": []

},

"LoiterRegionParamList": [{

"TargetTypeConstrain": false,

"TargetType": 0,

"TargetSizeConstrain": false,

"MinTargetSize": 1111,

"MaxTargetSize": 9999,

"MinTime": 5,

"IsAnalysisPath": true,

"SNPoint": [{

"Xval": 0,

"Yval": 0

}, {

"Xval": 0,

"Yval": 99.500000

}, {

"Xval": 99.500000,

"Yval": 99.500000

}, {

"Xval": 99.500000,

"Yval": 0

}]

}]

}]

}

**Description**

#### Set the configuration parameters.

#### Return value

0 succeeded; other values failed.

### sdks\_get\_ia\_multi\_loiter\_ability

#### Define

#### int sdks\_get\_ia\_multi\_loiter\_ability(unsigned int handle, int chn, char \*\*p\_result)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number.

p\_result：The output parameters, expressed in json format, return the acquired multiplayer configuration capabilities.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | Types | Optional / required | description |
| Parameters | | | | E | M |  |
|  | MultiLoiterAbility | | | E | M | Multiplayer |
|  | LibId | | A | M | Algorithm library ID |
| enable | | A | M | Whether to configure intrusion detection capability options |
| Region | | E | M | Drawing detection area ability |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | edgeNum | A | M | Limiting a region with several edges |
|  | maxRegionNum | A | M | Number of restricted areas |
| DetailGet | | E | M | The backend gets the details of the test results. |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
| OSDDrawLine | | E | M | Print the detection area in the video |
|  | enable | A | M | Whether to enable this feature |
| AnalysisPath | | E | M | Path analysis |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | min | A | M |  |
|  | max | A | M |  |
| TargetTypeConstrain | | E | M | Limit target type: person\car\person and car |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | min | A | M | Minimum value |
|  | max | A | M | Maximum value |
| TargetSizeConstrain | | E | M | Limit target size |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | min | A | M | Minimum value |
|  | max | A | M | Maximum value |
| MinTargetSize | | E | M | Minimum target size |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | min | A | M | Minimum value |
|  | max | A | M | Maximum value |
|  | unit | A | M |  |
| MaxTargetSize | | E | M | Maximum target size |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | min | A | M | Minimum value |
|  | max | A | M | Maximum value |
|  | unit | A | M |  |
| MinTime | | E | M | the shortest time |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | min | A | M | Minimum value |
|  | max | A | M | Maximum value |
|  | unit | A | M | Unit: S seconds |
| ForbiddenType | | E | M | Limit number of people |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
| Minimum | | E | M | Limit the minimum number of people |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | min | A | M | Minimum value |
|  | max | A | M | Maximum value |
| Maximum | | E | M | Limit the maximum number of people |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | Min | A | M | Minimum value |
|  | max | A | M | Maximum value |

p\_result the json format is：

{

"LibId": 6,

"Enable": true,

"RegionEnable": true,

"EdgeNum": 8,

"MaxRegionNum": 8,

"DetailGetEnable": false,

"OSDLineEnable": true,

"AnalysisPathEnable": false,

"AnalysisPathMax": 1,

"AnalysisPathMin": 0,

"TargetSizeConstrainEnable": true,

"TargetSizeConstrainMax": 1,

"TargetSizeConstrainMin": 0,

"TargetSizeConstrainUnit": "",

"MinTargetSizeEnable": true,

"MinTargetSizeMax": 1000000,

"MinTargetSizeMin": 0,

"MinTargetSizeUnit": "cm2",

"MaxTargetSizeEnable": true,

"MaxTargetSizeMax": 1000000,

"MaxTargetSizeMin": 0,

"MaxTargetSizeUnit": "cm2",

"MinTimeEnable": true,

"MinTimeMax": 60,

"MinTimeMin": 5,

"MinTimeUnit": "S",

"ForbiddenTypeEnable": true,

"MinimumEnable": true,

"MinimumMax": 99999,

"MinimumMin": 1,

"MaximumEnable": true,

"MaximumMax": 99999,

"MaximumMin": 1

}

**Description**

#### Get multiplayer configuration capabilities.

#### Return value

0 succeeded; other values failed.

### sdks\_get\_ia\_ multi\_loiter\_param

#### Define

#### int sdks\_get\_ia\_ multi\_loiter\_param(unsigned int handle, int chn, char \*\*p\_result)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number.

p\_result：The output parameters, expressed in json format, return the acquired multiplayer configuration parameters.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | | | | | | | Types | Optional / required | description |
| Parameters | | | | | | | | | | E | M |  |
|  | MultiLoiterParamList | | | | | | | | | E | M | Multi-person parameter |
|  | MultiLoiterParam | | | | | | | | E | M |  |
|  | IsGetDetail | | | | | | | A | M | Upload detection result information switch |
| IntelligentEvent | | | | | | | E | M |  |
|  | SourceType | | | | | | A | M | Alarm source type,see[1.8 Alarm source type](#_1.8_Alarm_source) |
| SourceID | | | | | | A | M | Alarm source ID |
| SourceName | | | | | | A | M | Alarm source name |
| EnableFlag | | | | | | A | M | Open tag |
| AlarmInterval | | | | | | A | M | Alarm interval |
| DrawEnable | | | | | | A | M | Print detection area open mark in video |
| ScheduleTimeParam | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
|  | ScheduleTimeList | | | | | E | M |  |
|  | | ScheduleTime | | | E |  |  |
| AlarmLinkageParamList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| PTZActionParamList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| AlarmOutActionList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| RecordActionParamList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| MultiLoiterRegionParamList | | | | | | | E | M |  |
|  | MultiLoiterRegionParam | | | | | | E | M |  |
|  | | TargetSizeConstrain | | | | A | M | Limited target size open flag |
| MinTargetSize | | | | A | M | smallest size |
| MaxTargetSize | | | | A | M | biggest size |
| MinTime | | | | A | M | the shortest time |
| AnalysisPath | | | | A | M | Analysis path open flag |
| ForbiddenType | | | | A | M | Restricted number open sign |
| Minimum | | | | A | M | Minimum number of people |
| Maximum | | | | A | M | Maximum number of people |
| Region | | | | E | M | region |
|  | | SNPoint | | E | M | point |
|  | Xval | A | M | X value |
| Yval | A | M | Y value |

p\_result the json format is：

{

"MultiLoiterParamList": [{

"MultiLoiterParam": {

"IsGetDetail": false

},

"IntelligentEvent": {

"SourceType": 12,

"SourceID": 1,

"SourceName": "",

"EnableFlag": false,

"AlarmInterval": 0,

"DrawEnable": false,

"ScheduleTimeList": [{

"WeekDay": 0,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 1,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 2,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 3,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 4,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 5,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 6,

"StartTime": 0,

"EndTime": 86400

}],

"AlarmLinkageParamList": [{

"ActionType": 1,

"ActionID": 1

}, {

"ActionType": 2,

"ActionID": 1

}, {

"ActionType": 4,

"ActionID": 1

}, {

"ActionType": 7,

"ActionID": 1

}, {

"ActionType": 3,

"ActionID": 2

}],

"PTZActionParamList": [],

"AlarmOutActionList": [],

"RecordActionParamList": []

},

"MultiLoiterRegionParamList": [{

"TargetSizeConstrain": false,

"MinTargetSize": 1111,

"MaxTargetSize": 9999,

"MinTime": 5,

"AnalysisPath": false,

"ForbiddenType": false,

"Minimum": 1,

"Maximum": 3,

"SNPoint": [{

"Xval": 0,

"Yval": 0

}, {

"Xval": 0,

"Yval": 99.500000

}, {

"Xval": 99.500000,

"Yval": 99.500000

}, {

"Xval": 99.500000,

"Yval": 0

}]

}]

}]

}

**Description**

#### Get multiplayer configuration parameters.

#### Return value

0 succeeded; other values failed.

### sdks\_set\_ia\_ multi\_loiter\_param

#### Define

#### int sdks\_set\_ia\_ multi\_loiter\_param(unsigned int handle, int chn, char \*p\_param)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number.

p\_param：Input parameters, expressed in json format.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | | | | | | | Types | Optional / required | description |
| Parameters | | | | | | | | | | E | M |  |
|  | MultiLoiterParamList | | | | | | | | | E | M | Multi-person parameter |
|  | MultiLoiterParam | | | | | | | | E | M |  |
|  | IsGetDetail | | | | | | | A | M | Upload detection result information switch |
| IntelligentEvent | | | | | | | E | M |  |
|  | SourceType | | | | | | A | M | Alarm source type,see[1.8 Alarm source type](#_1.8_Alarm_source) |
| SourceID | | | | | | A | M | Alarm source ID |
| SourceName | | | | | | A | M | Alarm source name |
| EnableFlag | | | | | | A | M | Open tag |
| AlarmInterval | | | | | | A | M | Alarm interval |
| DrawEnable | | | | | | A | M | Print detection area open mark in video |
| ScheduleTimeParam | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
|  | ScheduleTimeList | | | | | E | M |  |
|  | | ScheduleTime | | | E |  |  |
| AlarmLinkageParamList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| PTZActionParamList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| AlarmOutActionList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| RecordActionParamList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| MultiLoiterRegionParamList | | | | | | | E | M |  |
|  | MultiLoiterRegionParam | | | | | | E | M |  |
|  | | TargetSizeConstrain | | | | A | M | Limited target size open flag |
| MinTargetSize | | | | A | M | smallest size |
| MaxTargetSize | | | | A | M | biggest size |
| MinTime | | | | A | M | the shortest time |
| AnalysisPath | | | | A | M | Analysis path open flag |
| ForbiddenType | | | | A | M | Restricted number open sign |
| Minimum | | | | A | M | Minimum number of people |
| Maximum | | | | A | M | Maximum number of people |
| Region | | | | E | M | region |
|  | | SNPoint | | E | M | point |
|  | Xval | A | M | X value |
| Yval | A | M | Y value |

p\_param the json format is：

{

"MultiLoiterParamList": [{

"MultiLoiterParam": {

"IsGetDetail": false

},

"IntelligentEvent": {

"SourceType": 12,

"SourceID": 1,

"SourceName": "",

"EnableFlag": false,

"AlarmInterval": 0,

"DrawEnable": false,

"ScheduleTimeList": [{

"WeekDay": 0,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 1,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 2,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 3,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 4,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 5,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 6,

"StartTime": 0,

"EndTime": 86400

}],

"AlarmLinkageParamList": [{

"ActionType": 1,

"ActionID": 1

}, {

"ActionType": 2,

"ActionID": 1

}, {

"ActionType": 4,

"ActionID": 1

}, {

"ActionType": 7,

"ActionID": 1

}, {

"ActionType": 3,

"ActionID": 2

}],

"PTZActionParamList": [],

"AlarmOutActionList": [],

"RecordActionParamList": []

},

"MultiLoiterRegionParamList": [{

"TargetSizeConstrain": false,

"MinTargetSize": 1111,

"MaxTargetSize": 9999,

"MinTime": 5,

"AnalysisPath": false,

"ForbiddenType": false,

"Minimum": 1,

"Maximum": 3,

"SNPoint": [{

"Xval": 0,

"Yval": 0

}, {

"Xval": 0,

"Yval": 99.500000

}, {

"Xval": 99.500000,

"Yval": 99.500000

}, {

"Xval": 99.500000,

"Yval": 0

}]

}]

}]

}

**Description**

#### Set multiplayer configuration parameters.

#### Return value

0 succeeded; other values failed.

### sdks\_get\_ia\_object\_left\_ability

#### Define

#### int sdks\_get\_ia\_object\_left\_param(unsigned int handle, int chn, char \*\*p\_result)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number.

p\_result：The output parameters, expressed in json format, return the acquired legacy configuration capabilities.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | Types | Optional / required | description |
| Parameters | | | | E | M |  |
|  | ObjLeftAbility | | | E | M | Item left |
|  | LibId | | A | M | Algorithm library ID |
| enable | | A | M | Whether to configure intrusion detection capability options |
| Region | | E | M | Drawing detection area ability |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | edgeNum | A | M | Limiting a region with several edges |
|  | maxRegionNum | A | M | Number of restricted areas |
| DetailGet | | E | M | The backend gets the details of the test results. |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
| OSDDrawLine | | E | M | Print the detection area in the video |
|  | enable | A | M | Whether to enable this feature |
| MinTargetSize | | E | M | Minimum target size |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | min | A | M | Minimum value |
|  | max | A | M | Maximum value |
|  | unit | A | M |  |
| MaxTargetSize | | E | M | Maximum target size |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | min | A | M | Minimum value |
|  | max | A | M | Maximum value |
|  | unit | A | M |  |
| MinTime | | E | M | The shortest time left |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | min | A |  | Minimum value |
|  | max | A | M | Maximum value |
|  | unit | A | M | Time unit: S seconds |

p\_result the json format is：

{

"LibId": 6,

"Enable": true,

"RegionEnable": true,

"EdgeNum": 8,

"MaxRegionNum": 8,

"DetailGetEnable": true,

"OSDLineEnable": true,

"MinTargetSizeEnable": true,

"MinTargetSizeMax": 40000,

"MinTargetSizeMin": 10,

"MinTargetSizeUnit": "cm2",

"MaxTargetSizeEnable": true,

"MaxTargetSizeMax": 40000,

"MaxTargetSizeMin": 10,

"MaxTargetSizeUnit": "cm2",

"MinTimeEnable": true,

"MinTimeMax": 60,

"MinTimeMin": 5,

"MinTimeUnit": "S"

}

**Description**

#### Get the ability to configure legacy items.

#### Return value

0 succeeded; other values failed.

### sdks\_get\_ia\_ object\_left \_param

#### Define

#### int sdks\_get\_ia\_ object\_left \_param(unsigned int handle, int chn, char \*\*p\_result)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number.

p\_result：The output parameter, expressed in json format, returns the retrieved configuration parameters of the item.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | | | | | | | Types | Optional / required | description |
| Parameters | | | | | | | | | | E | M |  |
|  | ObjLeftParamList | | | | | | | | | E | M | Item legacy parameter |
|  | ObjLeftParam | | | | | | | | E | M |  |
|  | IsGetDetail | | | | | | | A | M | Upload detection result information switch |
| IntelligentEvent | | | | | | | E | M |  |
|  | SourceType | | | | | | A | M | Alarm source type,see[1.8 Alarm source type](#_1.8_Alarm_source) |
| SourceID | | | | | | A | M | Alarm source ID |
| SourceName | | | | | | A | M | Alarm source name |
| EnableFlag | | | | | | A | M | Open tag |
| AlarmInterval | | | | | | A | M | Alarm interval |
| DrawEnable | | | | | | A | M | Print detection area open mark in video |
| ScheduleTimeParam | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
|  | ScheduleTimeList | | | | | E | M |  |
|  | | ScheduleTime | | | E |  |  |
| AlarmLinkageParamList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| PTZActionParamList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| AlarmOutActionList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| RecordActionParamList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| ObjLeftRegionParamList | | | | | | | E | M |  |
|  | ObjLeftRegionParam | | | | | | E | M |  |
|  | | MinTargetSize | | | | A | M | smallest size |
| MaxTargetSize | | | | A | M | biggest size |
| MinTime | | | | A | M | The shortest time left |
| Region | | | | E | M | region |
|  | | SNPoint | | E | M | point |
|  | Xval | A | M | X value |
| Yval | A | M | Y value |

p\_result the json format is：

{

"ObjLeftParamList": [{

"ObjLeftParam": {

"IsGetDetail": false

},

"IntelligentEvent": {

"SourceType": 13,

"SourceID": 1,

"SourceName": "",

"EnableFlag": false,

"AlarmInterval": 0,

"DrawEnable": false,

"ScheduleTimeList": [{

"WeekDay": 0,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 1,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 2,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 3,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 4,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 5,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 6,

"StartTime": 0,

"EndTime": 86400

}],

"AlarmLinkageParamList": [{

"ActionType": 1,

"ActionID": 1

}, {

"ActionType": 2,

"ActionID": 1

}, {

"ActionType": 4,

"ActionID": 1

}, {

"ActionType": 7,

"ActionID": 1

}, {

"ActionType": 3,

"ActionID": 2

}],

"PTZActionParamList": [],

"AlarmOutActionList": [],

"RecordActionParamList": []

},

"ObjLeftRegionParamList": [{

"MinTargetSize": 1111,

"MaxTargetSize": 9999,

"MinTime": 5,

"SNPoint": [{

"Xval": 0,

"Yval": 0

}, {

"Xval": 0,

"Yval": 99.500000

}, {

"Xval": 99.500000,

"Yval": 99.500000

}, {

"Xval": 99.500000,

"Yval": 0

}]

}]

}]

}

**Description**

#### Get the item legacy configuration parameters.

#### Return value

0 succeeded; other values failed.

### sdks\_set\_ia\_ object\_left \_param

#### Define

#### int sdks\_set\_ia\_ object\_left \_param(unsigned int handle, int chn, char \*p\_param)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number.

p\_param：Input parameters, expressed in json format.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | | | | | | | Types | Optional / required | description |
| Parameters | | | | | | | | | | E | M |  |
|  | ObjLeftParamList | | | | | | | | | E | M | Item legacy parameter |
|  | ObjLeftParam | | | | | | | | E | M |  |
|  | IsGetDetail | | | | | | | A | M | Upload detection result information switch |
| IntelligentEvent | | | | | | | E | M |  |
|  | SourceType | | | | | | A | M | Alarm source type,see[1.8 Alarm source type](#_1.8_Alarm_source) |
| SourceID | | | | | | A | M | Alarm source ID |
| SourceName | | | | | | A | M | Alarm source name |
| EnableFlag | | | | | | A | M | Open tag |
| AlarmInterval | | | | | | A | M | Alarm interval |
| DrawEnable | | | | | | A | M | Print detection area open mark in video |
| ScheduleTimeParam | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
|  | ScheduleTimeList | | | | | E | M |  |
|  | | ScheduleTime | | | E |  |  |
| AlarmLinkageParamList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| PTZActionParamList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| AlarmOutActionList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| RecordActionParamList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| ObjLeftRegionParamList | | | | | | | E | M |  |
|  | ObjLeftRegionParam | | | | | | E | M |  |
|  | | MinTargetSize | | | | A | M | smallest size |
| MaxTargetSize | | | | A | M | biggest size |
| MinTime | | | | A | M | The shortest time left |
| Region | | | | E | M | region |
|  | | SNPoint | | E | M | point |
|  | Xval | A | M | X value |
| Yval | A | M | Y value |

p\_param the json format is：

{

"ObjLeftParamList": [{

"ObjLeftParam": {

"IsGetDetail": false

},

"IntelligentEvent": {

"SourceType": 13,

"SourceID": 1,

"SourceName": "",

"EnableFlag": false,

"AlarmInterval": 0,

"DrawEnable": false,

"ScheduleTimeList": [{

"WeekDay": 0,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 1,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 2,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 3,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 4,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 5,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 6,

"StartTime": 0,

"EndTime": 86400

}],

"AlarmLinkageParamList": [{

"ActionType": 1,

"ActionID": 1

}, {

"ActionType": 2,

"ActionID": 1

}, {

"ActionType": 4,

"ActionID": 1

}, {

"ActionType": 7,

"ActionID": 1

}, {

"ActionType": 3,

"ActionID": 2

}],

"PTZActionParamList": [],

"AlarmOutActionList": [],

"RecordActionParamList": []

},

"ObjLeftRegionParamList": [{

"MinTargetSize": 1111,

"MaxTargetSize": 9999,

"MinTime": 5,

"SNPoint": [{

"Xval": 0,

"Yval": 0

}, {

"Xval": 0,

"Yval": 99.500000

}, {

"Xval": 99.500000,

"Yval": 99.500000

}, {

"Xval": 99.500000,

"Yval": 0

}]

}]

}]

}

**Description**

#### Set the item legacy configuration parameters.

#### Return value

0 succeeded; other values failed.

### sdks\_get\_ia\_object\_removed\_ability

#### Define

#### int sdks\_get\_ia\_object\_removed\_param(unsigned int handle, int chn, char \*\*p\_result)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number.

p\_result：The output parameter, expressed in json format, returns the acquired item removal configuration capability.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | Types | Optional / required | description |
| Parameters | | | | E | M |  |
|  | ObjMovedAbility | | | E | M | Item removal |
|  | LibId | | A | M | Algorithm library ID |
| enable | | A | M | Whether to configure intrusion detection capability options |
| Region | | E | M | Drawing detection area ability |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | edgeNum | A | M | Limiting a region with several edges |
|  | maxRegionNum | A | M | Number of restricted areas |
| DetailGet | | E | M | The backend gets the details of the test results. |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
| OSDDrawLine | | E | M | Print the detection area in the video |
|  | enable | A | M | Whether to enable this feature |
| MinTargetSize | | E | M | Minimum target size |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | min | A | M | Minimum value |
|  | max | A | M | Maximum value |
|  | unit | A | M |  |
| MaxTargetSize | | E | M | Maximum target size |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | min | A | M | Minimum value |
|  | max | A | M | Maximum value |
|  | unit | A | M |  |
| MinTime | | E | M | The shortest time left |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | min | A |  | Minimum value |
|  | max | A | M | Maximum value |
|  | unit | A | M | Time unit: S seconds |

p\_result the json format is：

{

"LibId": 6,

"Enable": true,

"RegionEnable": true,

"EdgeNum": 8,

"MaxRegionNum": 8,

"DetailGetEnable": true,

"OSDLineEnable": true,

"MinTargetSizeEnable": true,

"MinTargetSizeMax": 40000,

"MinTargetSizeMin": 10,

"MinTargetSizeUnit": "cm2",

"MaxTargetSizeEnable": true,

"MaxTargetSizeMax": 40000,

"MaxTargetSizeMin": 10,

"MaxTargetSizeUnit": "cm2",

"MinTimeEnable": true,

"MinTimeMax": 60,

"MinTimeMin": 5,

"MinTimeUnit": "S"

}

**Description**

#### Get the item removal configuration ability

#### Return value

0 succeeded; other values failed.

### sdks\_get\_ia\_ object\_removed\_param

#### Define

#### int sdks\_get\_ia\_ object\_removed\_param(unsigned int handle, int chn, char \*\*p\_result)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number.

p\_result：The output parameters are represented in json format, and the returned items are returned to remove the configuration parameters.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | | | | | | | Types | Optional / required | description |
| Parameters | | | | | | | | | | E | M |  |
|  | ObjMovedParamList | | | | | | | | | E | M | Item removal parameter |
|  | ObjMovedParam | | | | | | | | E | M |  |
|  | IsGetDetail | | | | | | | A | M | Upload detection result information switch |
| IntelligentEvent | | | | | | | E | M |  |
|  | SourceType | | | | | | A | M | Alarm source type,see[1.8 Alarm source type](#_1.8_Alarm_source) |
| SourceID | | | | | | A | M | Alarm source ID |
| SourceName | | | | | | A | M | Alarm source name |
| EnableFlag | | | | | | A | M | Open tag |
| AlarmInterval | | | | | | A | M | Alarm interval |
| DrawEnable | | | | | | A | M | Print detection area open mark in video |
| ScheduleTimeParam | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
|  | ScheduleTimeList | | | | | E | M |  |
|  | | ScheduleTime | | | E |  |  |
| AlarmLinkageParamList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| PTZActionParamList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| AlarmOutActionList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| RecordActionParamList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| ObjMovedRegionParamList | | | | | | | E | M |  |
|  | ObjMovedRegionParam | | | | | | E | M |  |
|  | | MinTargetSize | | | | A | M | smallest size |
| MaxTargetSize | | | | A | M | biggest size |
| MinTime | | | | A | M | Remove the shortest time |
| Region | | | | E | M | region |
|  | | SNPoint | | E | M | point |
|  | Xval | A | M | X value |
| Yval | A | M | Y value |

p\_result the json format is：

{

"ObjMovedParamList": [{

"ObjMovedParam": {

"IsGetDetail": false

},

"IntelligentEvent": {

"SourceType": 13,

"SourceID": 1,

"SourceName": "",

"EnableFlag": false,

"AlarmInterval": 0,

"DrawEnable": false,

"ScheduleTimeList": [{

"WeekDay": 0,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 1,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 2,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 3,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 4,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 5,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 6,

"StartTime": 0,

"EndTime": 86400

}],

"AlarmLinkageParamList": [{

"ActionType": 1,

"ActionID": 1

}, {

"ActionType": 2,

"ActionID": 1

}, {

"ActionType": 4,

"ActionID": 1

}, {

"ActionType": 7,

"ActionID": 1

}, {

"ActionType": 3,

"ActionID": 2

}],

"PTZActionParamList": [],

"AlarmOutActionList": [],

"RecordActionParamList": []

},

"ObjMovedRegionParamList": [{

"MinTargetSize": 1111,

"MaxTargetSize": 9999,

"MinTime": 5,

"SNPoint": [{

"Xval": 0,

"Yval": 0

}, {

"Xval": 0,

"Yval": 99.500000

}, {

"Xval": 99.500000,

"Yval": 99.500000

}, {

"Xval": 99.500000,

"Yval": 0

}]

}]

}]

}

**Description**

#### Get the item removal configuration parameters.

#### Return value

0 succeeded; other values failed.

### sdks\_set\_ia\_ object\_removed\_param

#### Define

#### int sdks\_set\_ia\_ object\_removed\_param(unsigned int handle, int chn, char \*p\_param)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number.

p\_param：Input parameters, expressed in json format.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | | | | | | | Types | Optional / required | description |
| Parameters | | | | | | | | | | E | M |  |
|  | ObjMovedParamList | | | | | | | | | E | M | Item removal parameter |
|  | ObjMovedParam | | | | | | | | E | M |  |
|  | IsGetDetail | | | | | | | A | M | Upload detection result information switch |
| IntelligentEvent | | | | | | | E | M |  |
|  | SourceType | | | | | | A | M | Alarm source type,see[1.8 Alarm source type](#_1.8_Alarm_source) |
| SourceID | | | | | | A | M | Alarm source ID |
| SourceName | | | | | | A | M | Alarm source name |
| EnableFlag | | | | | | A | M | Open tag |
| AlarmInterval | | | | | | A | M | Alarm interval |
| DrawEnable | | | | | | A | M | Print detection area open mark in video |
| ScheduleTimeParam | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
|  | ScheduleTimeList | | | | | E | M |  |
|  | | ScheduleTime | | | E |  |  |
| AlarmLinkageParamList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| PTZActionParamList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| AlarmOutActionList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| RecordActionParamList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| ObjMovedRegionParamList | | | | | | | E | M |  |
|  | ObjMovedRegionParam | | | | | | E | M |  |
|  | | MinTargetSize | | | | A | M | smallest size |
| MaxTargetSize | | | | A | M | biggest size |
| MinTime | | | | A | M | Remove the shortest time |
| Region | | | | E | M | region |
|  | | SNPoint | | E | M | point |
|  | Xval | A | M | X value |
| Yval | A | M | Y value |

p\_param the json format is：

{

"ObjMovedParamList": [{

"ObjMovedParam": {

"IsGetDetail": false

},

"IntelligentEvent": {

"SourceType": 13,

"SourceID": 1,

"SourceName": "",

"EnableFlag": false,

"AlarmInterval": 0,

"DrawEnable": false,

"ScheduleTimeList": [{

"WeekDay": 0,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 1,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 2,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 3,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 4,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 5,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 6,

"StartTime": 0,

"EndTime": 86400

}],

"AlarmLinkageParamList": [{

"ActionType": 1,

"ActionID": 1

}, {

"ActionType": 2,

"ActionID": 1

}, {

"ActionType": 4,

"ActionID": 1

}, {

"ActionType": 7,

"ActionID": 1

}, {

"ActionType": 3,

"ActionID": 2

}],

"PTZActionParamList": [],

"AlarmOutActionList": [],

"RecordActionParamList": []

},

"ObjMovedRegionParamList": [{

"MinTargetSize": 1111,

"MaxTargetSize": 9999,

"MinTime": 5,

"SNPoint": [{

"Xval": 0,

"Yval": 0

}, {

"Xval": 0,

"Yval": 99.500000

}, {

"Xval": 99.500000,

"Yval": 99.500000

}, {

"Xval": 99.500000,

"Yval": 0

}]

}]

}]

}

**Description**

#### Set item removal configuration parameters.

#### Return value

0 succeeded; other values failed.

### sdks\_get\_ia\_abnormal\_speed\_ability

#### Define

#### int sdks\_get\_ia\_abnormal\_speed\_ability(unsigned int handle, int chn, char \*\*p\_result) parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number.

p\_result：The output parameters, expressed in json format, return the acquired speed exception configuration capability.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | Types | Optional / required | description |
| Parameters | | | | E | M |  |
|  | AbnormalSpeedAblity | | | E | M | Abnormal speed |
|  | LibId | | A | M | Algorithm library ID |
| enable | | A | M | Whether to configure intrusion detection capability options |
| Region | | E | M | Drawing detection area ability |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | edgeNum | A | M | Limiting a region with several edges |
|  | maxRegionNum | A | M | Number of restricted areas |
| DetailGet | | E | M | The backend gets the details of the test results. |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
| TargetTypeConstrain | | E | M | Limit target type: person\car\person and car |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | min | A | M | Minimum value |
|  | max | A | M | Maximum value |
| TargetType | | E | M | Target type |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | min | A | M | Minimum value |
|  | max | A | M | Maximum value |
| TargetSizeConstrain | | E | M | Limit target size |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | min | A | M | Minimum value |
|  | max | A | M | Maximum value |
| MinTargetSize | | E | M | Limit target size |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | min | A | M | Minimum value |
|  | max | A | M | Maximum value |
|  | unit | A | M |  |
| MaxTargetSize | | E | M | Limit target size |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | min | A | M | Minimum value |
|  | max | A | M | Maximum value |
|  | unit | A | M |  |
| RateLimitingType | | E | M | Speed limit type |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | min | A | M | Minimum value |
|  | max | A | M | Maximum value |
| MinRate | | E | M | Minimum speed |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | min | A | M | Minimum value |
|  | max | A | M | Maximum value |
|  | unit | A | M | Unit: m / s |
| MaxRate | | E | M |  |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | min | A | M | Minimum value |
|  | max | A | M | Maximum value |
|  | unit | A | M | Unit: m / s |

p\_result the json format is：

{

"LibId": 6,

"Enable": true,

"RegionEnable": true,

"EdgeNum": 8,

"MaxRegionNum": 8,

"DetailGetEnable": true,

"TargetTypeConstrainEnable": true,

"TargetTypeConstrainMax": 1,

"TargetTypeConstrainMin": 0,

"TargetTypeEnable": true,

"TargetTypeMax": 2,

"TargetTypeMin": 0,

"TargetSizeConstrainEnable": true,

"TargetSizeConstrainMax": 1,

"TargetSizeConstrainMin": 0,

"MinTargetSizeEnable": true,

"MinTargetSizeMax": 1000000,

"MinTargetSizeMin": 0,

"MinTargetSizeUnit": "cm2",

"MaxTargetSizeEnable": true,

"MaxTargetSizeMax": 1000000,

"MaxTargetSizeMin": 0,

"MaxTargetSizeUnit": "cm2",

"RateLimitingTypeEnable": false,

"RateLimitingTypeMax": 1,

"RateLimitingTypeMin": 0,

"MinRateEnable": true,

"MinRateMax": 1000,

"MinRateMin": 0,

"MinRateUnit": "m/s",

"MaxRateEnable": true,

"MaxRateMax": 1000,

"MaxRateMin": 0,

"MaxRateUnit": "m/s"

}

**Description**

#### Get speed exception configuration capabilities.

#### Return value

0 succeeded; other values failed.

### sdks\_get\_ia\_abnormal\_speed\_param

#### Define

#### int sdks\_get\_ia\_abnormal\_speed\_param(unsigned int handle, int chn, char \*\*p\_result)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number.

p\_result：The output parameter, expressed in json format, returns the obtained speed exception configuration parameter.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | | | | | | | Types | Optional / required | description |
| Parameters | | | | | | | | | | E | M |  |
|  | AbnormalSpeedParamList | | | | | | | | | E | M | Speed anomaly parameter |
|  | AbnormalSpeedParam | | | | | | | | E | M |  |
|  | IsGetDetail | | | | | | | A | M | Upload detection result information switch |
| IntelligentEvent | | | | | | | E | M |  |
|  | SourceType | | | | | | A | M | Alarm source type,see[1.8 Alarm source type](#_1.8_Alarm_source) |
| SourceID | | | | | | A | M | Alarm source ID |
| SourceName | | | | | | A | M | Alarm source name |
| EnableFlag | | | | | | A | M | Open tag |
| AlarmInterval | | | | | | A | M | Alarm interval |
| DrawEnable | | | | | | A | M | Print detection area open mark in video |
| ScheduleTimeParam | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
|  | ScheduleTimeList | | | | | E | M |  |
|  | | ScheduleTime | | | E |  |  |
| AlarmLinkageParamList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| PTZActionParamList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| AlarmOutActionList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| RecordActionParamList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| AbnormalSpeedRegionParamList | | | | | | | E | M |  |
|  | AbnormalSpeedRegionParam | | | | | | E | M |  |
|  | | TargetTypeConstrain | | | | A | M | Limited target type open flag |
| TargetType | | | | A | M | Target type |
| TargetSizeConstrain | | | | A | M | Limited target size open flag |
| MinTargetSize | | | | A | M | smallest size |
| MaxTargetSize | | | | A | M | biggest size |
| RateLimitingType | | | | A | M | Speed limit type |
| MinRate | | | | A | M | Minimum speed |
| MaxRate | | | | A | M | Maximum speed |
| Region | | | | E | M | region |
|  | | SNPoint | | E | M | point |
|  | Xval | A | M | X value |
| Yval | A | M | Y value |

p\_result the json format is：

{

"AbnormalSpeedParamList": [{

"AbnormalSpeedParam": {

"IsGetDetail": false

},

"IntelligentEvent": {

"SourceType": 15,

"SourceID": 1,

"SourceName": "",

"EnableFlag": false,

"AlarmInterval": 0,

"DrawEnable": false,

"ScheduleTimeList": [{

"WeekDay": 0,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 1,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 2,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 3,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 4,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 5,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 6,

"StartTime": 0,

"EndTime": 86400

}],

"AlarmLinkageParamList": [{

"ActionType": 1,

"ActionID": 1

}, {

"ActionType": 2,

"ActionID": 1

}, {

"ActionType": 4,

"ActionID": 1

}, {

"ActionType": 7,

"ActionID": 1

}, {

"ActionType": 3,

"ActionID": 2

}],

"PTZActionParamList": [],

"AlarmOutActionList": [],

"RecordActionParamList": []

},

"AbnormalSpeedRegionParamList": [{

"TargetTypeConstrain": false,

"TargetType": 0,

"TargetSizeConstrain": false,

"MinTargetSize": 1111,

"MaxTargetSize": 9999,

"RateLimitingType": 1,

"MinRate": 1,

"MaxRate": 2,

"SNPoint": [{

"Xval": 0,

"Yval": 0

}, {

"Xval": 0,

"Yval": 99.500000

}, {

"Xval": 99.500000,

"Yval": 99.500000

}, {

"Xval": 99.500000,

"Yval": 0

}]

}]

}]

}

**Description**

#### Get the speed exception configuration parameters.

#### Return value

0 succeeded; other values failed.

### sdks\_set\_ia\_abnormal\_speed\_param

#### Define

#### int sdks\_set\_ia\_abnormal\_speed\_param(unsigned int handle, int chn, char \*p\_param)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number.

p\_param：Input parameters, expressed in json format.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | | | | | | | Types | Optional / required | description |
| Parameters | | | | | | | | | | E | M |  |
|  | AbnormalSpeedParamList | | | | | | | | | E | M | Speed anomaly parameter |
|  | AbnormalSpeedParam | | | | | | | | E | M |  |
|  | IsGetDetail | | | | | | | A | M | Upload detection result information switch |
| IntelligentEvent | | | | | | | E | M |  |
|  | SourceType | | | | | | A | M | Alarm source type,see[1.8 Alarm source type](#_1.8_Alarm_source) |
| SourceID | | | | | | A | M | Alarm source ID |
| SourceName | | | | | | A | M | Alarm source name |
| EnableFlag | | | | | | A | M | Open tag |
| AlarmInterval | | | | | | A | M | Alarm interval |
| DrawEnable | | | | | | A | M | Print detection area open mark in video |
| ScheduleTimeParam | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
|  | ScheduleTimeList | | | | | E | M |  |
|  | | ScheduleTime | | | E |  |  |
| AlarmLinkageParamList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| PTZActionParamList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| AlarmOutActionList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| RecordActionParamList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| AbnormalSpeedRegionParamList | | | | | | | E | M |  |
|  | AbnormalSpeedRegionParam | | | | | | E | M |  |
|  | | TargetTypeConstrain | | | | A | M | Limited target type open flag |
| TargetType | | | | A | M | Target type |
| TargetSizeConstrain | | | | A | M | Limited target size open flag |
| MinTargetSize | | | | A | M | smallest size |
| MaxTargetSize | | | | A | M | biggest size |
| RateLimitingType | | | | A | M | Speed limit type |
| MinRate | | | | A | M | Minimum speed |
| MaxRate | | | | A | M | Maximum speed |
| Region | | | | E | M | region |
|  | | SNPoint | | E | M | point |
|  | Xval | A | M | X value |
| Yval | A | M | Y value |

p\_param the json format is：

{

"AbnormalSpeedParamList": [{

"AbnormalSpeedParam": {

"IsGetDetail": false

},

"IntelligentEvent": {

"SourceType": 15,

"SourceID": 1,

"SourceName": "",

"EnableFlag": false,

"AlarmInterval": 0,

"DrawEnable": false,

"ScheduleTimeList": [{

"WeekDay": 0,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 1,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 2,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 3,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 4,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 5,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 6,

"StartTime": 0,

"EndTime": 86400

}],

"AlarmLinkageParamList": [{

"ActionType": 1,

"ActionID": 1

}, {

"ActionType": 2,

"ActionID": 1

}, {

"ActionType": 4,

"ActionID": 1

}, {

"ActionType": 7,

"ActionID": 1

}, {

"ActionType": 3,

"ActionID": 2

}],

"PTZActionParamList": [],

"AlarmOutActionList": [],

"RecordActionParamList": []

},

"AbnormalSpeedRegionParamList": [{

"TargetTypeConstrain": false,

"TargetType": 0,

"TargetSizeConstrain": false,

"MinTargetSize": 1111,

"MaxTargetSize": 9999,

"RateLimitingType": 1,

"MinRate": 1,

"MaxRate": 2,

"SNPoint": [{

"Xval": 0,

"Yval": 0

}, {

"Xval": 0,

"Yval": 99.500000

}, {

"Xval": 99.500000,

"Yval": 99.500000

}, {

"Xval": 99.500000,

"Yval": 0

}]

}]

}]

}

**Description**

#### Set the speed exception configuration parameters.

#### Return value

0 succeeded; other values failed.

### sdks\_get\_ia\_converse\_ability

#### Define

#### int sdks\_get\_ia\_converse\_ability(unsigned int handle, int chn, char \*\*p\_result)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number.

p\_result：The output parameters, expressed in json format, return the acquired retrograde configuration capabilities.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | Types | Optional / required | description |
| Parameters | | | | E | M |  |
|  | ConverseAbility | | | E | M | Retrograde ability |
|  | LibId | | A | M | Algorithm library ID |
| enable | | A | M | Whether to configure intrusion detection capability options |
| Region | | E | M | Drawing detection area ability |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | edgeNum | A | M | Limiting a region with several edges |
|  | maxRegionNum | A | M | Number of restricted areas |
| DetailGet | | E | M | The backend gets the details of the test results. |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
| OSDDrawLine | | E | M | Whether to enable this function, whether the user can configure |
|  | enable | A | M | Whether to enable this feature |
| TargetTypeConstrain | | E | M | Limit target type: person\car\person and car |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | min | A | M | Minimum value |
|  | max | A | M | Maximum value |
| TargetType | | E | M | Target type |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | min | A | M | Minimum value |
|  | max | A | M | Maximum value |
| TargetSizeConstrain | | E | M | Limit target size |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | min | A | M | Minimum value |
|  | max | A | M | Maximum value |
| MinTargetSize | | E | M | Minimum target size |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | min | A | M | Minimum value |
|  | max | A | M | Maximum value |
|  | unit | A | M |  |
| MaxTargetSize | | E | M | Maximum target size |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | min | A | M | Minimum value |
|  | max | A | M | Maximum value |
|  | unit | A | M |  |
| ConverseAngle | | E | M | Retrograde configuration angle |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | min | A | M | Minimum value |
|  | max | A | M | Maximum value |
|  | unit | A | M | unit: |

p\_result the json format is：

{

"LibId": 6,

"Enable": true,

"RegionEnable": true,

"EdgeNum": 8,

"MaxRegionNum": 8,

"DetailGetEnable": true,

"OSDLineEnable": true,

"TargetTypeConstrainEnable": true,

"TargetTypeConstrainMax": 1,

"TargetTypeConstrainMin": 0,

"TargetTypeEnable": true,

"TargetTypeMax": 2,

"TargetTypeMin": 0,

"TargetSizeConstrainEnable": true,

"TargetSizeConstrainMax": 1,

"TargetSizeConstrainMin": 0,

"MinTargetSizeEnable": true,

"MinTargetSizeMax": 1000000,

"MinTargetSizeMin": 0,

"MinTargetSizeUnit": "cm2",

"MaxTargetSizeEnable": true,

"MaxTargetSizeMax": 1000000,

"MaxTargetSizeMin": 0,

"MaxTargetSizeUnit": "cm2",

"ConverseAngleEnable": true,

"ConverseAngleMax": 360,

"ConverseAngleMin": 0,

"ConverseAngleUnit": "degree"

}

**Description**

#### Get retrograde configuration capabilities.

#### Return value

0 succeeded; other values failed.

### sdks\_get\_ia\_converse\_param

#### Define

#### int sdks\_get\_ia\_converse\_param(unsigned int handle, int chn, char \*\*p\_result)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number.

p\_result：The output parameters, expressed in json format, return the obtained retrograde configuration parameters.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | | | | | | | Types | Optional / required | description |
| Parameters | | | | | | | | | | E | M |  |
|  | ConverseParamList | | | | | | | | | E | M | Retrograde parameter |
|  | ConverseParam | | | | | | | | E | M |  |
|  | IsGetDetail | | | | | | | A | M | Upload detection result information switch |
| IntelligentEvent | | | | | | | E | M |  |
|  | SourceType | | | | | | A | M | Alarm source type,see[1.8 Alarm source type](#_1.8_Alarm_source) |
| SourceID | | | | | | A | M | Alarm source ID |
| SourceName | | | | | | A | M | Alarm source name |
| EnableFlag | | | | | | A | M | Open tag |
| AlarmInterval | | | | | | A | M | Alarm interval |
| DrawEnable | | | | | | A | M | Print detection area open mark in video |
| ScheduleTimeParam | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
|  | ScheduleTimeList | | | | | E | M |  |
|  | | ScheduleTime | | | E |  |  |
| AlarmLinkageParamList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| PTZActionParamList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| AlarmOutActionList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| RecordActionParamList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| ConverseRegionParamList | | | | | | | E | M |  |
|  | ConverseRegionParam | | | | | | E | M |  |
|  | | TargetTypeConstrain | | | | A | M | Limited target type open flag |
| TargetType | | | | A | M | Target type |
| TargetSizeConstrain | | | | A | M | Limited target size open flag |
| MinTargetSize | | | | A | M | smallest size |
| MaxTargetSize | | | | A | M | biggest size |
| ConverseAngle | | | | A | M | Retrograde direction angle |
| Region | | | | E | M | region |
|  | | SNPoint | | E | M | point |
|  | Xval | A | M | X value |
| Yval | A | M | Y value |

p\_result the json format is：

{

"ConverseParamList": [{

"ConverseParam": {

"IsGetDetail": false

},

"IntelligentEvent": {

"SourceType": 16,

"SourceID": 1,

"SourceName": "",

"EnableFlag": false,

"AlarmInterval": 0,

"DrawEnable": false,

"ScheduleTimeList": [{

"WeekDay": 0,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 1,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 2,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 3,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 4,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 5,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 6,

"StartTime": 0,

"EndTime": 86400

}],

"AlarmLinkageParamList": [{

"ActionType": 1,

"ActionID": 1

}, {

"ActionType": 2,

"ActionID": 1

}, {

"ActionType": 4,

"ActionID": 1

}, {

"ActionType": 7,

"ActionID": 1

}, {

"ActionType": 3,

"ActionID": 2

}],

"PTZActionParamList": [],

"AlarmOutActionList": [],

"RecordActionParamList": []

},

"ConverseRegionParamList": [{

"TargetTypeConstrain": false,

"TargetType": 0,

"TargetSizeConstrain": false,

"MinTargetSize": 1111,

"MaxTargetSize": 9999,

"ConverseAngle": 1,

"SNPoint": [{

"Xval": 0,

"Yval": 0

}, {

"Xval": 0,

"Yval": 99.500000

}, {

"Xval": 99.500000,

"Yval": 99.500000

}, {

"Xval": 99.500000,

"Yval": 0

}]

}]

}]

}

**Description**

#### Get the retrograde configuration parameters.

#### Return value

0 succeeded; other values failed.

### sdks\_set\_ia\_converse\_param

#### Define

#### int sdks\_set\_ia\_converse\_param(unsigned int handle, int chn, char \*p\_param)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number.

p\_param：Input parameters, expressed in json format.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | | | | | | | Types | Optional / required | description |
| Parameters | | | | | | | | | | E | M |  |
|  | ConverseParamList | | | | | | | | | E | M | Retrograde parameter |
|  | ConverseParam | | | | | | | | E | M |  |
|  | IsGetDetail | | | | | | | A | M | Upload detection result information switch |
| IntelligentEvent | | | | | | | E | M |  |
|  | SourceType | | | | | | A | M | Alarm source type,see[1.8 Alarm source type](#_1.8_Alarm_source) |
| SourceID | | | | | | A | M | Alarm source ID |
| SourceName | | | | | | A | M | Alarm source name |
| EnableFlag | | | | | | A | M | Open tag |
| AlarmInterval | | | | | | A | M | Alarm interval |
| DrawEnable | | | | | | A | M | Print detection area open mark in video |
| ScheduleTimeParam | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
|  | ScheduleTimeList | | | | | E | M |  |
|  | | ScheduleTime | | | E |  |  |
| AlarmLinkageParamList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| PTZActionParamList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| AlarmOutActionList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| RecordActionParamList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| ConverseRegionParamList | | | | | | | E | M |  |
|  | ConverseRegionParam | | | | | | E | M |  |
|  | | TargetTypeConstrain | | | | A | M | Limited target type open flag |
| TargetType | | | | A | M | Target type |
| TargetSizeConstrain | | | | A | M | Limited target size open flag |
| MinTargetSize | | | | A | M | smallest size |
| MaxTargetSize | | | | A | M | biggest size |
| ConverseAngle | | | | A | M | Retrograde direction angle |
| Region | | | | E | M | region |
|  | | SNPoint | | E | M | point |
|  | Xval | A | M | X value |
| Yval | A | M | Y value |

p\_param 的json格式为：

{

"ConverseParamList": [{

"ConverseParam": {

"IsGetDetail": false

},

"IntelligentEvent": {

"SourceType": 16,

"SourceID": 1,

"SourceName": "",

"EnableFlag": false,

"AlarmInterval": 0,

"DrawEnable": false,

"ScheduleTimeList": [{

"WeekDay": 0,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 1,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 2,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 3,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 4,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 5,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 6,

"StartTime": 0,

"EndTime": 86400

}],

"AlarmLinkageParamList": [{

"ActionType": 1,

"ActionID": 1

}, {

"ActionType": 2,

"ActionID": 1

}, {

"ActionType": 4,

"ActionID": 1

}, {

"ActionType": 7,

"ActionID": 1

}, {

"ActionType": 3,

"ActionID": 2

}],

"PTZActionParamList": [],

"AlarmOutActionList": [],

"RecordActionParamList": []

},

"ConverseRegionParamList": [{

"TargetTypeConstrain": false,

"TargetType": 0,

"TargetSizeConstrain": false,

"MinTargetSize": 1111,

"MaxTargetSize": 9999,

"ConverseAngle": 1,

"SNPoint": [{

"Xval": 0,

"Yval": 0

}, {

"Xval": 0,

"Yval": 99.500000

}, {

"Xval": 99.500000,

"Yval": 99.500000

}, {

"Xval": 99.500000,

"Yval": 0

}]

}]

}]

}

**Description**

#### Set the retrograde configuration parameters.

#### Return value

0 succeeded; other values failed.

### sdks\_get\_ia\_legal\_parking\_ability

#### Define

#### int sdks\_get\_ia\_legal\_parking\_ability(unsigned int handle, int chn, char \*\*p\_result)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number.

p\_result：The output parameters, expressed in json format, return the acquired illegal parking configuration capabilities

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | Types | Optional / required | description |
| Parameters | | | | E | M |  |
|  | NoParkingAbility | | | E | M | Illegal parking |
|  | LibId | | A | M | Algorithm library ID |
| enable | | A | M | Whether to configure the illegal parking detection capability option |
| Region | | E | M | Drawing detection area ability |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | edgeNum | A | M | Limiting a region with several edges |
|  | maxRegionNum | A | M | Number of restricted areas |
| DetailGet | | E | M | The backend gets the details of the test results. |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
| OSDDrawLine | | E | M | Print the detection area in the video |
|  | enable | A | M | Whether to enable this feature |
| MinTargetSize | | E | M | Minimum target size |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | min | A | M | Minimum value |
|  | max | A | M | Maximum value |
|  | unit | A | M |  |
| MaxTargetSize | | E | M | Maximum target size |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | min | A | M | Minimum value |
|  | max | A | M | Maximum value |
|  | unit | A | M |  |
| MinTime | | E | M | The shortest time left |
|  | enable | A | M | Whether to enable this function, whether the user can configure |
|  | min | A |  | Minimum value |
|  | max | A | M | Maximum value |
|  | unit | A | M | Time unit: S seconds |

p\_result the json format is：

{

"LibId": 6,

"Enable": true,

"RegionEnable": true,

"EdgeNum": 8,

"MaxRegionNum": 8,

"DetailGetEnable": true,

"OSDLineEnable": true,

"MinTargetSizeEnable": true,

"MinTargetSizeMax": 1000000,

"MinTargetSizeMin": 0,

"MinTargetSizeUnit": "cm2",

"MaxTargetSizeEnable": true,

"MaxTargetSizeMax": 1000000,

"MaxTargetSizeMin": 0,

"MaxTargetSizeUnit": "cm2",

"MinTimeEnable": true,

"MinTimeMax": 60,

"MinTimeMin": 5,

"MinTimeUnit": "S"

}

**Description**

#### Get illegal parking configuration capabilities.

#### Return value

0 succeeded; other values failed.

### sdks\_get\_ia\_legal\_parking\_ param

#### Define

#### int sdks\_get\_ia\_legal\_parking\_param(unsigned int handle, int chn, char \*\*p\_result)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number.

p\_result：The output parameters are expressed in json format and return the obtained illegal parking configuration parameters.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | | | | | | | Types | Optional / required | description |
| Parameters | | | | | | | | | | E | M |  |
|  | NoParkingParamList | | | | | | | | | E | M | Illegal parking parameter |
|  | NoParkingParam | | | | | | | | E | M |  |
|  | IsGetDetail | | | | | | | A | M | Upload detection result information switch |
| IntelligentEvent | | | | | | | E | M |  |
|  | SourceType | | | | | | A | M | Alarm source type,see[1.8 Alarm source type](#_1.8_Alarm_source) |
| SourceID | | | | | | A | M | Alarm source ID |
| SourceName | | | | | | A | M | Alarm source name |
| EnableFlag | | | | | | A | M | Open tag |
| AlarmInterval | | | | | | A | M | Alarm interval |
| DrawEnable | | | | | | A | M | Print detection area open mark in video |
| ScheduleTimeParam | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
|  | ScheduleTimeList | | | | | E | M |  |
|  | | ScheduleTime | | | E |  |  |
| AlarmLinkageParamList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| PTZActionParamList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| AlarmOutActionList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| RecordActionParamList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| NoParkingRegionParamList | | | | | | | E | M |  |
|  | NoParkingRegionParam | | | | | | E | M |  |
|  | | MinTargetSize | | | | A | M | smallest size |
| MaxTargetSize | | | | A | M | biggest size |
| MinTime | | | | A | M | Minimum parking time |
| Region | | | | E | M | region |
|  | | SNPoint | | E | M | point |
|  | Xval | A | M | X value |
| Yval | A | M | Y value |

p\_result the json format is：

{

"NoParkingParamList": [{

"NoParkingParam": {

"IsGetDetail": false

},

"IntelligentEvent": {

"SourceType": 17,

"SourceID": 1,

"SourceName": "",

"EnableFlag": false,

"AlarmInterval": 0,

"DrawEnable": false,

"ScheduleTimeList": [{

"WeekDay": 0,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 1,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 2,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 3,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 4,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 5,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 6,

"StartTime": 0,

"EndTime": 86400

}],

"AlarmLinkageParamList": [{

"ActionType": 1,

"ActionID": 1

}, {

"ActionType": 2,

"ActionID": 1

}, {

"ActionType": 4,

"ActionID": 1

}, {

"ActionType": 7,

"ActionID": 1

}, {

"ActionType": 3,

"ActionID": 2

}],

"PTZActionParamList": [],

"AlarmOutActionList": [],

"RecordActionParamList": []

},

"NoParkingRegionParamList": [{

"MinTargetSize": 1111,

"MaxTargetSize": 9999,

"MinTime": 5,

"SNPoint": [{

"Xval": 0,

"Yval": 0

}, {

"Xval": 0,

"Yval": 99.500000

}, {

"Xval": 99.500000,

"Yval": 99.500000

}, {

"Xval": 99.500000,

"Yval": 0

}]

}]

}]

}

**Description**

#### Get illegal parking configuration parameters.

#### Return value

0 succeeded; other values failed.

### sdks\_set\_ia\_legal\_parking\_ param

#### Define

#### int sdks\_set\_ia\_legal\_parking\_param(unsigned int handle, int chn, char \*p\_param)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number.

p\_param：Input parameters, expressed in json format.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | | | | | | | Types | Optional / required | description |
| Parameters | | | | | | | | | | E | M |  |
|  | NoParkingParamList | | | | | | | | | E | M | Illegal parking parameter |
|  | NoParkingParam | | | | | | | | E | M |  |
|  | IsGetDetail | | | | | | | A | M | Upload detection result information switch |
| IntelligentEvent | | | | | | | E | M |  |
|  | SourceType | | | | | | A | M | Alarm source type,see[1.8 Alarm source type](#_1.8_Alarm_source) |
| SourceID | | | | | | A | M | Alarm source ID |
| SourceName | | | | | | A | M | Alarm source name |
| EnableFlag | | | | | | A | M | Open tag |
| AlarmInterval | | | | | | A | M | Alarm interval |
| DrawEnable | | | | | | A | M | Print detection area open mark in video |
| ScheduleTimeParam | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
|  | ScheduleTimeList | | | | | E | M |  |
|  | | ScheduleTime | | | E |  |  |
| AlarmLinkageParamList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| PTZActionParamList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| AlarmOutActionList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| RecordActionParamList | | | | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| NoParkingRegionParamList | | | | | | | E | M |  |
|  | NoParkingRegionParam | | | | | | E | M |  |
|  | | MinTargetSize | | | | A | M | smallest size |
| MaxTargetSize | | | | A | M | biggest size |
| MinTime | | | | A | M | Minimum parking time |
| Region | | | | E | M | region |
|  | | SNPoint | | E | M | point |
|  | Xval | A | M | X value |
| Yval | A | M | Y value |

p\_param the json format is：

{

"NoParkingParamList": [{

"NoParkingParam": {

"IsGetDetail": false

},

"IntelligentEvent": {

"SourceType": 17,

"SourceID": 1,

"SourceName": "",

"EnableFlag": false,

"AlarmInterval": 0,

"DrawEnable": false,

"ScheduleTimeList": [{

"WeekDay": 0,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 1,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 2,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 3,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 4,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 5,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 6,

"StartTime": 0,

"EndTime": 86400

}],

"AlarmLinkageParamList": [{

"ActionType": 1,

"ActionID": 1

}, {

"ActionType": 2,

"ActionID": 1

}, {

"ActionType": 4,

"ActionID": 1

}, {

"ActionType": 7,

"ActionID": 1

}, {

"ActionType": 3,

"ActionID": 2

}],

"PTZActionParamList": [],

"AlarmOutActionList": [],

"RecordActionParamList": []

},

"NoParkingRegionParamList": [{

"MinTargetSize": 1111,

"MaxTargetSize": 9999,

"MinTime": 5,

"SNPoint": [{

"Xval": 0,

"Yval": 0

}, {

"Xval": 0,

"Yval": 99.500000

}, {

"Xval": 99.500000,

"Yval": 99.500000

}, {

"Xval": 99.500000,

"Yval": 0

}]

}]

}]

}

**Description**

#### Set illegal parking configuration parameters.

#### Return value

0 succeeded; other values failed.

### sdks\_get\_ia\_signal\_bad\_ability

#### Define

#### int sdks\_get\_ia\_signal\_bad\_ability(unsigned int handle, int chn, char \*\*p\_result)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number.

p\_result：The output parameters are expressed in json format and return the acquired video signal exception configuration capability.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | Types | Optional / required | description |
| Parameters | | | | E | M |  |
|  | SignalBadAbility | | | E | M | Abnormal signal |
|  | LibId | | A | M | Algorithm library ID |
| enable | | A | M | Whether to configure the signal anomaly detection capability option |
| DetailGet | | E | M | The backend gets the details of the test results. |
|  | enable | A | M | Whether to enable this function, whether the user can configure |

p\_result the json format is：

{

"LibId": 0,

"Enable": false,

"DetailGetEnable": false

}

**Description**

#### Get the ability to configure video signal exceptions.

#### Return value

0 succeeded; other values failed.

### sdks\_get\_ia\_signal\_bad\_ param

#### Define

#### int sdks\_get\_ia\_signal\_bad\_param(unsigned int handle, int chn, char \*\*p\_result)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number.

p\_result：The output parameters are expressed in json format and return the acquired video signal exception configuration parameters.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | | | | Types | Optional / required | description |
| Parameters | | | | | | | E | M |  |
|  | SignalBadParamList | | | | | | E | M | Video signal anomaly parameter |
|  | SignalBadParam | | | | | E | M |  |
|  | IsGetDetail | | | | A | M | Upload detection result information switch |
| IntelligentEvent | | | | E | M |  |
|  | SourceType | | | A | M | Alarm source type,see[1.8 Alarm source type](#_1.8_Alarm_source) |
| SourceID | | | A | M | Alarm source ID |
| SourceName | | | A | M | Alarm source name |
| EnableFlag | | | A | M | Open tag |
| AlarmInterval | | | A | M | Alarm interval |
| DrawEnable | | | A | M | Print detection area open mark in video |
| ScheduleTimeParam | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
|  | ScheduleTimeList | | E | M |  |
|  | ScheduleTime | E |  |  |
| AlarmLinkageParamList | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| PTZActionParamList | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| AlarmOutActionList | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| RecordActionParamList | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |

p\_result the json format is：

{

"SignalBadParamList": [{

"SignalBadParam": {

"IsGetDetail": false

},

"IntelligentEvent": {

"SourceType": 18,

"SourceID": 1,

"SourceName": "",

"EnableFlag": false,

"AlarmInterval": 0,

"DrawEnable": false,

"ScheduleTimeList": [{

"WeekDay": 0,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 1,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 2,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 3,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 4,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 5,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 6,

"StartTime": 0,

"EndTime": 86400

}],

"AlarmLinkageParamList": [],

"PTZActionParamList": [],

"AlarmOutActionList": [],

"RecordActionParamList": []

}

}]

}

**Description**

#### Get the video signal exception configuration parameters.

#### Return value

0 succeeded; other values failed.

### sdks\_set\_ia\_signal\_bad\_ param

#### Define

#### int sdks\_set\_ia\_signal\_bad\_param(unsigned int handle, int chn, char \*p\_param)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number.

p\_param：Input parameters, expressed in json format.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | | | | Types | Optional / required | description |
| Parameters | | | | | | | E | M |  |
|  | SignalBadParamList | | | | | | E | M | Video signal anomaly parameter |
|  | SignalBadParam | | | | | E | M |  |
|  | IsGetDetail | | | | A | M | Upload detection result information switch |
| IntelligentEvent | | | | E | M |  |
|  | SourceType | | | A | M | Alarm source type,see[1.8 Alarm source type](#_1.8_Alarm_source) |
| SourceID | | | A | M | Alarm source ID |
| SourceName | | | A | M | Alarm source name |
| EnableFlag | | | A | M | Open tag |
| AlarmInterval | | | A | M | Alarm interval |
| DrawEnable | | | A | M | Print detection area open mark in video |
| ScheduleTimeParam | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
|  | ScheduleTimeList | | E | M |  |
|  | ScheduleTime | E |  |  |
| AlarmLinkageParamList | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| PTZActionParamList | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| AlarmOutActionList | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |
| RecordActionParamList | | | E | M | See [IO Alarm policy parameter table](#_IO_Alarm_policy) |

p\_param the json format is：

{

"SignalBadParamList": [{

"SignalBadParam": {

"IsGetDetail": false

},

"IntelligentEvent": {

"SourceType": 18,

"SourceID": 1,

"SourceName": "",

"EnableFlag": false,

"AlarmInterval": 0,

"DrawEnable": false,

"ScheduleTimeList": [{

"WeekDay": 0,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 1,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 2,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 3,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 4,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 5,

"StartTime": 0,

"EndTime": 86400

}, {

"WeekDay": 6,

"StartTime": 0,

"EndTime": 86400

}],

"AlarmLinkageParamList": [],

"PTZActionParamList": [],

"AlarmOutActionList": [],

"RecordActionParamList": []

}

}]

}

**Description**

#### Set the video signal exception configuration parameters.

#### Return value

0 succeeded; other values failed.

### sdks\_get\_ia\_advanced\_ability

#### Define

#### int sdks\_get\_ia\_advanced\_ability(unsigned int handle, int chn, char \*\*p\_result)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number.

p\_result：The output parameters, expressed in json format, return the advanced configuration capabilities acquired.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | | | Types | Optional / required | description |
| Parameters | | | | | | E | M |  |
|  | AdvancedAbility | | | | | E | M | Advanced configuration capability |
|  | LibId | | | | A | M | Algorithm library ID |
| enable | | | | A | M | Whether to configure intrusion detection capability options |
| Scene | | | | E | M | Indoor/outdoor |
|  | | enable | | A | M | Whether to enable this function, whether the user can configure |
|  | | min | | A | M |  |
|  | | max | | A | M |  |
| Hight | | | | E | M | Installation height |
|  | | enable | | A | M | Whether to enable this function, whether the user can configure |
| Angle | | | | E | M | installation angle |
|  | | enable | | A | M | Whether to enable this feature |
| FOV | | | | E | M | Lens field of view |
|  | | enable | | A | M | Whether to enable this function, whether the user can configure |
| CameraShake | | | | E | M | Camera shake |
|  | | enable | | A | M | Whether to enable this function, whether the user can configure |
|  | | min | | A | M | Minimum value |
|  | | max | | A | M | Maximum value |
| HighNoise | | | | E | M | High noise environment |
|  | | enable | | A | M | Whether to enable this function, whether the user can configure |
|  | | min | | A | M | Minimum value |
|  | | max | | A | M | Maximum value |
| LowContrast | | | | E | M | Low contrast |
|  | | enable | | A | M | Whether to enable this function, whether the user can configure |
|  | | min | | A | M | Minimum value |
|  | | max | | A | M | Maximum value |
|  | | unit | | A | M |  |
| PeriodMotion | | | | E | M | Periodic motion background |
|  | | enable | | A | M | Whether to enable this function, whether the user can configure |
|  | | min | | A | M | Minimum value |
|  | | max | | A | M | Maximum |
| PeriodMotionTime | | | | E | M | Periodic motion background time |
|  | | enable | | A | M | Whether to enable this function, whether the user can configure |
|  | | min | | A | M | Minimum value |
|  | | max | | A | M | Maximum |
|  | | unit | |  | M | Unit: second |
| PixelToRealSize | | | | E | M | Conversion of camera pixels to physical object length |
|  | enable | | | A | M | Whether to enable this function, whether the user can configure |
|  | RealSize | | | E | M | Actual length of the object |
|  |  | | enable | A | M | Whether to enable this function, whether the user can configure |
|  |  | | unit | A | M | Unit centimeter |
|  | SegmentLayType | | | E | M | Line orientation (horizontal / vertical) |
|  |  | | enable | A | M | Whether to enable this function, whether the user can configure |
|  | Segment | | | E | M | Line segment parameter |
|  |  | | enable | A | M | Whether to enable this function, whether the user can configure |

p\_result the json format is：

{

"LibId": 6,

"Enable": true,

"SceneEnable": true,

"SceneMin": 0,

"SceneMax": 1,

"HightEnable": false,

"AngleEnable": false,

"FOVEnable": false,

"CameraShakeEnable": false,

"CameraShakeMin": 0,

"CameraShakeMax": 1,

"HighNoiseEnable": false,

"HighNoiseMin": 0,

"HighNoiseMax": 1,

"LowContrastEnable": false,

"LowContrastMin": 0,

"LowContrastMax": 1,

"PeriodMotionEnable": false,

"PeriodMotionMin": 0,

"PeriodMotionMax": 1,

"PeriodMotionTimeUnit": "S",

"PeriodMotionTimeEnable": false,

"PeriodMotionTimeMin": 1,

"PeriodMotionTimeMax": 60,

"RealSizeEnable": true,

"RealSizeUnit": "cm",

"PixelToRealSizeEnable": true,

"SegmentLayTypeEnable": true,

"SegmentEnable": true

}

**Description**

#### Get advanced configuration capabilities.

#### Return value

0 succeeded; other values failed.

### sdks\_get\_ia\_advanced\_param

#### Define

#### int sdks\_get\_ia\_advanced\_param(unsigned int handle, int chn, char \*\*p\_result)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number.

p\_result：The output parameters, expressed in json format, return the advanced configuration parameters obtained.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | Types | Optional / required | description |
| Parameters | | | | E | M |  |
|  | AdvancedParam | | | E | M |  |
| Scene | | | A | M | Scene: indoor / outdoor |
| AlarmInterval | | | A | M | Alarm interval |
| Hight | | | A | M | Installation height |
| Angle | | | A | M | Mounting angle, relative to vertical down |
| FOV | | | A | M | Lens field of view |
| CameraShake | | | A | M | Camera shake (0: off 1: on) |
| HighNoise | | | A | M | High noise environment (0: off 1: on) |
| LowContrast | | | A | M | Low contrast (0: off 1: on) |
| PeriodMotion | | | A | M | Periodic motion background (0: off 1: on) |
| PeriodMotionTime | | | A | M | Periodic motion background time (seconds) |
| PixelToRealSizeList | | | E | M | Conversion of camera pixels to physical object length |
|  | PixelToRealSize | | E | M |  |
| RealSize | | A | M | Actual length of the object in centimeters |
| SegmentLayType | | A | M | Line orientation (horizontal / vertical) |
|  | Segment | E | M | Line segment parameter |
| LineCrossStartX | A | M | Line starting point X |
| LineCrossStartY | A | M | Line starting point Y |
| LineCrossEndX | A | M | Line end point X |
| LineCrossEndY | A | M | Line end point Y |

p\_result the json format is：

{

"AdvancedParamList": [{

"AdvancedParam": {

"Scene": 0,

"Hight": 0,

"Angle": 0,

"FOV": 0,

"CameraShake": 0,

"HighNoise": 0,

"LowContrast": 0,

"PeriodMotion": 0,

"PeriodMotionTime": 15,

"AlarmInterval": 8

},

"IntelligentEvent": {

"SourceType": 0,

"SourceID": 1,

"SourceName": "",

"EnableFlag": false,

"AlarmInterval": 8,

"DrawEnable": false,

"ScheduleTimeList": [],

"AlarmLinkageParamList": [],

"PTZActionParamList": [],

"AlarmOutActionList": [],

"RecordActionParamList": []

},

"PixelToRealSizeList": [{

"RealSize": 100,

"SegmentLayType": 0,

"TripWireLine": {

"LineCrossStartX": 77,

"LineCrossStartY": 39,

"LineCrossEndX": 0,

"LineCrossEndY": 39

}

}]

}]

}

**Description**

Get advanced configuration parameters.

#### Return value

0 succeeded; other values failed.

### sdks\_set\_ia\_advanced\_param

#### Define

#### int sdks\_set\_ia\_advanced\_param(unsigned int handle, int chn, char \*p\_param)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number.

p\_param：Input parameters, expressed in json format.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Attributes | | | | Types | Optional / required | description |
| Parameters | | | | E | M |  |
|  | AdvancedParam | | | E | M |  |
| Scene | | | A | M | Scene: indoor / outdoor |
| AlarmInterval | | | A | M | Alarm interval |
| Hight | | | A | M | Installation height |
| Angle | | | A | M | Mounting angle, relative to vertical down |
| FOV | | | A | M | Lens field of view |
| CameraShake | | | A | M | Camera shake (0: off 1: on) |
| HighNoise | | | A | M | High noise environment (0: off 1: on) |
| LowContrast | | | A | M | Low contrast (0: off 1: on) |
| PeriodMotion | | | A | M | Periodic motion background (0: off 1: on) |
| PeriodMotionTime | | | A | M | Periodic motion background time (seconds) |
| PixelToRealSizeList | | | E | M | Conversion of camera pixels to physical object length |
|  | PixelToRealSize | | E | M |  |
| RealSize | | A | M | Actual length of the object in centimeters |
| SegmentLayType | | A | M | Line orientation (horizontal / vertical) |
|  | Segment | E | M | Line segment parameter |
| LineCrossStartX | A | M | Line starting point X |
| LineCrossStartY | A | M | Line starting point Y |
| LineCrossEndX | A | M | Line end point X |
| LineCrossEndY | A | M | Line end point Y |

p\_param the json format is：

{

"AdvancedParamList": [{

"AdvancedParam": {

"Scene": 0,

"Hight": 0,

"Angle": 0,

"FOV": 0,

"CameraShake": 0,

"HighNoise": 0,

"LowContrast": 0,

"PeriodMotion": 0,

"PeriodMotionTime": 15,

"AlarmInterval": 8

},

"IntelligentEvent": {

"SourceType": 0,

"SourceID": 1,

"SourceName": "",

"EnableFlag": false,

"AlarmInterval": 0,

"DrawEnable": false,

"ScheduleTimeList": [],

"AlarmLinkageParamList": [],

"PTZActionParamList": [],

"AlarmOutActionList": [],

"RecordActionParamList": []

},

"PixelToRealSizeList": [{

"RealSize": 100,

"SegmentLayType": 0,

"TripWireLine": {

"LineCrossStartX": 77,

"LineCrossStartY": 39,

"LineCrossEndX": 0,

"LineCrossEndY": 39

}

}]

}]

}

**Description**

#### Set advanced configuration parameters.

#### Return value

1. succeeded; other values failed.

## 26. Fisheye（For fisheye equipment）

### sdks\_get\_fisheye\_ability

#### Define

#### int sdks\_get\_fisheye\_ability(unsigned int handle, int chn, char \*\*p\_result)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number.

p\_result：The output parameters are expressed in json format and return the acquired fisheye capability information.

|  |  |  |
| --- | --- | --- |
| parameter | Description | Remarks |
| FisheyeSupportFlag | Whether to support fisheye |  |
| FisheyeMountType | Supported installation mode | Refer to Appendix [1.10.3 Installation mode](#_1.10.3_Installation_mode) |
| FisheyeVideoMode | Fisheye video mode | Refer to Appendix[1.10.1 Fisheye video mode](#_1.10.1_Fisheye_video) |
| DewarpMode | Correction mode | Refer to Appendix [1.10.2 Correction mode](#_1.10.2_Correction_mode) |
| Moveable | Is it movable |  |

p\_result the json format is：

{

"FisheyeSupportFlag": true,

"FishMountAbility": {

"FisheyeMountTypeList": [{

"FisheyeMountType": 1

}, {

"FisheyeMountType": 2

}, {

"FisheyeMountType": 0

}]

},

"FisheyeVideoAbilityList": [{

"FisheyeVideoMode": 0,

"DewarpModeList": [{

"DewarpMode": 6,

"Moveable": false

}, {

"DewarpMode": 5,

"Moveable": true

}, {

"DewarpMode": 4,

"Moveable": true

}, {

"DewarpMode": 0,

"Moveable": true

}, {

"DewarpMode": 2,

"Moveable": true

}, {

"DewarpMode": 8,

"Moveable": true

}, {

"DewarpMode": 10,

"Moveable": true

}]

}, {

"FisheyeVideoMode": 1,

"DewarpModeList": [{

"DewarpMode": 11,

"Moveable": true

}, {

"DewarpMode": 12,

"Moveable": true

}, {

"DewarpMode": 13,

"Moveable": true

}]

}]

}

**Description**

#### Get fish eye ability information.

#### Return value

0 succeeded; other values failed.

### sdks\_get\_fisheye\_video\_layout

#### Define

#### int sdks\_get\_fisheye\_video\_layout(unsigned int handle, int chn, char \*\*p\_result)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number.

p\_result：The output parameters, expressed in json format, return the fisheye layout information obtained.

|  |  |  |
| --- | --- | --- |
| parameter | Description | Remarks |
| DewarpMode | Correction mode | Refer to Appendix 1.13 |
| channleID | ID number | 0 is always a fisheye or panoramic video, the other is the ptzID number |
| startX | Start X coordinate point | Percentage (0~1)x100, precision is 2 digits after the decimal point |
| startY | Start Y coordinate point | Percentage (0~1)x100, precision is 2 digits after the decimal point |
| width | width | Percentage (0~1)x100, precision is 2 digits after the decimal point |
| height | height | Percentage (0~1)x100, precision is 2 digits after the decimal point |

p\_result the json format is：

{

"DewarpMode": 8,

"VideoRectList": [{

"channleID": 1,

"startX": 0,

"startY": 0,

"width": 50,

"height": 50

}, {

"channleID": 2,

"startX": 50,

"startY": 0,

"width": 50,

"height": 50

}, {

"channleID": 3,

"startX": 50,

"startY": 50,

"width": 50,

"height": 50

}, {

"channleID": 4,

"startX": 0,

"startY": 50,

"width": 50,

"height": 50

}]

}

**Description**

#### Get fisheye layout information.

#### Return value

0 succeeded; other values failed.

### sdks\_get\_fisheye\_param

#### Define

#### int sdks\_get\_fisheye\_param(unsigned int handle, int chn, char \*\*p\_result)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number.

p\_result：The output parameter, expressed in json format, returns the obtained fisheye parameter information.

|  |  |  |
| --- | --- | --- |
| parameter | Description | Remarks |
| VersionId | Channel number |  |
| FisheyeVideoMode | Fisheye video mode | Refer to Appendix [1.10.1 Fisheye video mode](#_1.10.1_Fisheye_video) |
| DewarpMode | Correction mode | Refer to Appendix [1.10.2 Correction mode](#_1.10.2_Correction_mode) |
| FisheyeMountType | Installation mode | Refer to Appendix [1.10.3 Installation mode](#_1.10.3_Installation_mode) |

p\_result the json format is：

{

"FisheyeDewarpModeParam": {

"VersionId": 1,

"FisheyeVideoMode": 0,

"DewarpMode": 8

},

"FisheyeMountParam": {

"FisheyeMountType": 1

}

}

**Description**

Get fisheye parameter information.

#### Return value

0 succeeded; other values failed.

### sdks\_set\_fisheye\_param

#### Define

#### int sdks\_set\_fisheye\_param(unsigned int handle, int chn, char \*p\_param)

#### parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number.

p\_param：Input parameters, expressed in json format.

|  |  |  |
| --- | --- | --- |
| parameter | Description | Remarks |
| VersionId | Channel number |  |
| FisheyeVideoMode | Fisheye video mode | Refer to Appendix [1.10.1 Fisheye video mode](#_1.10.1_Fisheye_video) |
| DewarpMode | Correction mode | Refer to Appendix [1.10.2 Correction mode](#_1.10.2_Correction_mode) |
| FisheyeMountType | Installation mode | Refer to Appendix [1.10.3 Installation mode](#_1.10.3_Installation_mode) |

p\_param the json format is：

{

"FisheyeDewarpModeParam": {

"VersionId": 1,

"FisheyeVideoMode": 0,

"DewarpMode": 8

},

"FisheyeMountParam": {

"FisheyeMountType": 1

}

}

**Description**

Set fisheye parameter information.

#### Return value

1. succeeded; other values failed.

## 27. Voice intercom

### sdks\_md\_talk\_start

#### Define

#### int sdks\_md\_talk\_start(unsigned int handle, int chn, SDK\_INTERCOM\_DB\_CB intercom\_db\_cb, void\* obj);

#### parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number.

intercom\_db\_cb：Callback function。

obj：Callback function context

**Description**

#### Turn on voice talk

#### Return value

0 succeeded; other values failed.

### sdks\_md\_talk\_stop

#### Define

#### int sdks\_md\_talk\_stop(unsigned int handle, int chn);

#### parameter

handle：Return value of sdks\_dev\_conn interface.

chn：Channel number.

**Description**

Turn off voice intercom

#### Return value

0 succeeded; other values failed.

### sdks\_dev\_send\_audio\_data

#### Define

#### int sdks\_dev\_send\_audio\_data(unsigned int handle, char \*p\_data, int audio\_len)

#### parameter

handle: Return value of sdks\_dev\_conn interface.

p\_data：Input parameters, raw PCM bare stream audio data.

audio\_len：Input parameters, The audio data size needs to be an integer multiple of 320.

**Description**

Send the recording to the device for playback(this interface needs to be called under windows to collect and input audio data).

#### Return value

0 succeeded; other values failed.

## 28. Panoramic dynamic video

### sdks\_md\_play\_dynamic\_video

#### definition

#### int sdks\_md\_play\_dynamic\_video(unsigned int handle, int chn, int stream\_type, int play\_type, void\* p\_wnd, bool is\_hw\_dec, SDK\_PLAY\_TIME\_CB play\_cb, void\* p\_obj)

#### parameter

handle：sdks\_dev\_conn return value .

chn：channel .

stream\_type：for stream type, pls refer to the definition of video\_stream\_type\_e in sdk\_def.h，STREAM\_TYPE\_1: HD，STREAM\_TYPE\_2: D1，STREAM\_TYPE\_ 3: Smooth. STREAM\_TYPE\_3 is not supported currently.

play\_type: Play type, 0: normal, 1: quadruple panorama, 2: fisheye.

p\_wnd: video display window

is\_hw\_dec: hardware acceleration. True: hardware acceleration. But abnormal decode may happen in some phone model type. False: no acceleration. This will cost more performance, but there will not be phone compatibility issue.

play\_cb: return to the current time of playing video stream 。

p\_obj: context of play\_cb

#### introduction

get stream of certain channel and play in certain window

#### return value

return stream\_id，>=0 refers to success, <0 refers to fail. stream\_id: used when switching stream and close stream

### sdks\_md\_set\_view\_port

#### Define

#### int sdks\_md\_set\_view\_port(unsigned int handle, int stream\_id, int left,int right,int top,int bottom)

#### parameter

handle: Return value of sdks\_dev\_conn interface.

stream\_id：Return value of sdks\_md\_live\_start interface.

left：The x coordinate of the upper left corner of the video window.

Right：The x coordinate of the upper right corner of the video window.

top：The x coordinate of the lower left corner of the video window.

bottom：The x coordinate of the lower right corner of the video window.

**Description**

#### Switch the video display window size (call this interface when changing the video window size). Return value

0 succeeded; other values failed.

### sdks\_md\_set\_auto\_cruise

#### Define

#### int sdks\_md\_set\_auto\_cruise(unsigned int handle, int stream\_id, int is\_auto\_cruise)

#### parameter

handle: Return value of sdks\_dev\_conn interface.

stream\_id：Return value of sdks\_md\_live\_start interface.

is\_auto\_cruise：Whether to enable the cruise flag, 0: off, 1: on.

**Description**

#### Turn off the cruise.

#### Return value

0 succeeded; other values failed.

### sdks\_md\_mouse\_button\_operate

#### Define

#### int sdks\_md\_mouse\_button\_operate(unsigned int handle, int stream\_id, int p\_button, int p\_action, int p\_positionx, int p\_positiony)

#### parameter

handle: Return value of sdks\_dev\_conn interface.

stream\_id：Return value of sdks\_md\_live\_start interface.

p\_button：Mouse button, 0: left button, 1: right button.

p\_action：The mouse button presses the mark, 1: presses, 0: releases.

p\_positionx: The abscissa of the video window.

p\_positiony: The ordinate of the video window.

**Description**

#### The mouse button handles the event.

#### Return value

0 succeeded; other values failed.

### sdks\_md\_mouse\_drag\_operate

#### Define

#### int sdks\_md\_mouse\_drag\_operate(unsigned int handle, int stream\_id, int p\_positionx, int p\_positiony)

#### parameter

handle: Return value of sdks\_dev\_conn interface.

stream\_id：Return value of sdks\_md\_live\_start interface.

p\_positionx: The abscissa of the video window.

p\_positiony: The ordinate of the video window.

**Description**

#### The mouse slides to handle the event (simultaneously by pressing the left mouse button and pressing the mouse to slide the interface to freely drag the video).

#### Return value

0 succeeded; other values failed.

### sdks\_md\_mouse\_scroll\_operate

#### Define

#### int sdks\_md\_mouse\_scroll\_operate(unsigned int handle, int stream\_id, float p\_delta, int p\_positionx, int p\_positiony)

#### parameter

handle: Return value of sdks\_dev\_conn interface.

stream\_id：Return value of sdks\_md\_live\_start interface.

p\_delta：The mouse wheel slides the variable.

p\_positionx: The abscissa of the video window.

p\_positiony: The ordinate of the video window.

**Description**

#### The mouse wheel scrolls to handle events (the video display can be zoomed in and out).

#### Return value

0 succeeded; other values failed.

### sdks\_md\_set\_pano\_show\_mode

#### Define

#### int sdks\_md\_set\_pano\_show\_mode(unsigned int handle, int stream\_id, int p\_show\_mode)

#### parameter

handle: Return value of sdks\_dev\_conn interface.

stream\_id：Return value of sdks\_md\_live\_start interface.

p\_show\_mode：The four-view panoramic video display mode, see Appendix[1.11.1 Playback device type](#_1.11.1_Playback_device), is called when the playback type selects four-view panorama playback.

**Description**

#### Set the panorama display mode.

#### Return value

0 succeeded; other values failed.

### sdks\_md\_set\_fisheye\_mount\_type

#### Define

#### int sdks\_md\_set\_fisheye\_mount\_type(unsigned int handle, int stream\_id, int p\_mount\_type) parameter

handle: Return value of sdks\_dev\_conn interface.

stream\_id：Return value of sdks\_md\_live\_start interface.

p\_mount\_type：For the fisheye installation method, see Appendix [1.11.5 Fisheye installation](#_1.11.5_Fisheye_installation). This interface is called when the playback type selects Fisheye Play.

**Description**

#### Set the fisheye installation mode.

#### Return value

0 succeeded; other values failed.

### sdks\_md\_set\_fisheye\_ceiling\_desktop\_show\_mode

#### Define

#### int sdks\_md\_set\_fisheye\_ceiling\_desktop\_show\_mode(unsigned int handle, int stream\_id, int p\_ceiling\_desktop\_show\_mode)

#### parameter

handle: Return value of sdks\_dev\_conn interface.

stream\_id：Return value of sdks\_md\_live\_start interface.

p\_ceiling\_desktop\_show\_mode：For the fisheye display mode, see Appendix [1.11.6 Fisheye display mode (for ceiling mount and desktop mount)](#_1.11.6__Fisheye). This interface is called when the fisheye installation type selects the ceiling or desktop.

**Description**

#### Set the fisheye display mode.

#### Return value

0 succeeded; other values failed.

### sdks\_md\_set\_fisheye\_wall\_show\_mode

#### Define

#### int sdks\_md\_set\_fisheye\_wall\_show\_mode(unsigned int handle, int stream\_id, int p\_wall\_show\_mode)

#### parameter

handle: Return value of sdks\_dev\_conn interface.

stream\_id：Return value of sdks\_md\_live\_start interface.

p\_wall\_show\_mode：For the fisheye display mode, see Appendix[1.11.7 Fisheye display mode (for wall mounting)](#_1.11.7_Fisheye_display) . This interface is called when the fisheye installation type selects the wall mount.

**Description**

#### Set the fisheye display mode.

#### Return value

0 succeeded; other values failed.

### sdks\_md\_get\_fisheye\_rectify\_parameters

#### Define

#### int sdks\_md\_get\_fisheye\_rectify\_parameters(unsigned int handle, int stream\_id, char\*\* p\_result)

#### parameter

handle: Return value of sdks\_dev\_conn interface.

stream\_id：Return value of sdks\_md\_live\_start interface.

p\_result：The fisheye correction parameters obtained are output in json format. (that is, get the location information of the current image)

|  |  |  |
| --- | --- | --- |
| Parameter | Description | Remarks |
| MountType | Fisheye installation |  |
| ShowMode | Display mode |  |
| PanDegreeList | Rotate the angle in the radial direction | A total of 8 sets of position parameters |
| TiltDegreeList | Rotate the angle along the tangential direction of the arc | A total of 8 sets of position parameters |
| ZoomDepthList | FOV field of view | A total of 8 sets of position parameters |

p\_result the json format is：

{

"MountType": 0,

"ShowMode": 2,

"PanDegreeList": [{

"PanDegree": 34.251350

}, {

"PanDegree": 0.358040

}, {

"PanDegree": 318.227844

}, {

"PanDegree": 263.849396

}, {

"PanDegree": 116.467010

}, {

"PanDegree": 216.589203

}, {

"PanDegree": 80.111465

}, {

"PanDegree": 13.899063

}],

"TiltDegreeList": [{

"TiltDegree": -68

}, {

"TiltDegree": -66.576637

}, {

"TiltDegree": -68

}, {

"TiltDegree": 68

}, {

"TiltDegree": 58.091599

}, {

"TiltDegree": 68

}, {

"TiltDegree": -50.253513

}, {

"TiltDegree": -19.050955

}],

"ZoomDepthList": [{

"ZoomDepth": 0

}, {

"ZoomDepth": 0

}, {

"ZoomDepth": 0

}, {

"ZoomDepth": 0

}, {

"ZoomDepth": 0

}, {

"ZoomDepth": 0

}, {

"ZoomDepth": 0

}, {

"ZoomDepth": 0

}]

}

**Description**

#### Obtain the fisheye correction parameters. (Available in fisheye mode, the role of this interface is to get the location information in the current video mode, so that the subsequent settings)

#### Return value

0 succeeded; other values failed.

### sdks\_md\_set\_fisheye\_rectify\_parameters

#### Define

#### int sdks\_md\_set\_fisheye\_rectify\_parameters(unsigned int handle, int stream\_id, char\* p\_param)

#### parameter

handle: Return value of sdks\_dev\_conn interface.

stream\_id：Return value of sdks\_md\_live\_start interface.

p\_param：Enter the fisheye correction parameters to be set in json format. (that is, setting the location information of the acquired image)

|  |  |  |
| --- | --- | --- |
| Parameter | Description | Remarks |
| MountType | Fisheye installation |  |
| ShowMode | Display mode |  |
| PanDegreeList | Rotate the angle in the radial direction | A total of 8 sets of position parameters |
| TiltDegreeList | Rotate the angle along the tangential direction of the arc | A total of 8 sets of position parameters |
| ZoomDepthList | FOV field of view | A total of 8 sets of position parameters |

p\_param the json format is：

{

"MountType": 0,

"ShowMode": 2,

"PanDegreeList": [{

"PanDegree": 34.251350

}, {

"PanDegree": 0.358040

}, {

"PanDegree": 318.227844

}, {

"PanDegree": 263.849396

}, {

"PanDegree": 116.467010

}, {

"PanDegree": 216.589203

}, {

"PanDegree": 80.111465

}, {

"PanDegree": 13.899063

}],

"TiltDegreeList": [{

"TiltDegree": -68

}, {

"TiltDegree": -66.576637

}, {

"TiltDegree": -68

}, {

"TiltDegree": 68

}, {

"TiltDegree": 58.091599

}, {

"TiltDegree": 68

}, {

"TiltDegree": -50.253513

}, {

"TiltDegree": -19.050955

}],

"ZoomDepthList": [{

"ZoomDepth": 0

}, {

"ZoomDepth": 0

}, {

"ZoomDepth": 0

}, {

"ZoomDepth": 0

}, {

"ZoomDepth": 0

}, {

"ZoomDepth": 0

}, {

"ZoomDepth": 0

}, {

"ZoomDepth": 0

}]

}

**Description**

#### Set the fisheye correction parameters. (Available in fisheye mode, the function of this interface is to display the mode and location information of the fisheye video display according to the saved location information when the video is subsequently opened)

#### Return value

0 succeeded; other values failed.

## Appendix

Alarm type definition：

### Alarm type

#### 1.1.1 Alarm main type

|  |  |
| --- | --- |
| **Value** | **Explain** |
| 1 | Safety alarm |
| 4 | Disk alarm |
| 5 | Video alarm |
| 6 | Intelligent analysis alarm |

#### 1.1.2 Alarm sub type

##### 1.1.2.1 Security alarm subtype

|  |  |
| --- | --- |
| **Value** | **Explain** |
| 1 | I/O alarm |
| 2 | Motion detection alarm |
| 3 | Camera blocking alarm |
| 4 | Video loss alarm |
| 5 | Networkdisconnection alarm |
| 9 | PIR analysis police |
| 10 | NVR channel I / O alarm |

##### 1.1.2.2 Analyses alarm subtype

|  |  |
| --- | --- |
| **Value** | **Explain** |
| 21 | Intelligent analysis trip wire detection alarm |
| 22 | Intelligent analysis mobile detection alarm |
| 23 | Intelligent analysis occlusion detection alarm |
| 24 | Intelligent analysis perimeter intrusion alarm |
| 25 | Intelligent analysis of double trip wire alarm |
| 26 | Intelligent analysis wandering alarm |
| 27 | Intelligent analysis multi person wandering alarm |
| 28 | Intelligent analysis of items left behind alarm |
| 29 | Intelligent analysis goods removal alarm |
| 30 | Intelligent analysis of abnormal speed alarm |
| 31 | Intelligent analysis retrograde alarm |
| 32 | Intelligent analysis of illegal parking alarm |
| 33 | Intelligent analysis camera shift alarm |
| 34 | Intelligent analysis of video signal abnormal alarm |
| 35 |  |
| 37 | License plate recognition alarm |

##### 1.1.2.3 Disk alarm subtype

|  |  |
| --- | --- |
| **Value** | **Explain** |
| 1 | Disk status OK |
| 2 | Disk read write exception |
| 3 | Network disk connection failed |
| 4 | Disk full |
| 5 | Disk does not exist |
| 6 | Disk used space reaches the specified threshold |
| 7 | Disk is not formatted |
| 8 | Insufficient storage space on device |
| 9 | Data version too low |
| 10 | Data version too high |
| 11 | Disk access mismatch |

##### 1.1.2.4 Video alarm subtype

|  |  |
| --- | --- |
| **Value** | **Explain** |
| 1 | Data source connection successful |
| 2 | Data source connection username and password error |
| 3 | Data source connection does not have permission |
| 4 | Data source connection reached the maximum number of connections |
| 5 | Data source reaches maximum limit rate |
| 6 |  |
| 7 |  |
| 8 |  |
| 9 | Storage failure |
| 10 | Startup video |
| 11 | Stop video |
| 12 |  |
| 13 |  |
| 14 |  |
| 15 |  |

##### 1.1.2.5 Temperature detection alarm subtype

|  |  |
| --- | --- |
| **Value** | **Explain** |
| 0 | Temperature threshold warning |
| 1 | Temperature threshold alarm |
| 4 | Temperature difference warning |
| 5 | Temperature difference alarm |

### group privilege manage type

|  |  |
| --- | --- |
| **Value** | **Explain** |
| 1 | add group |
| 2 | delete group |
| 3 | modify group |
| 4 | get all group |
| 5 | get self-group info |
| 6 | get group by group name |
| 7 | get group by user name |
| 9 | check if the admin group |

### private area operate type

|  |  |
| --- | --- |
| **Value** | **Explain** |
| 1 | add private area |
| 2 | delete private area |
| 3 | delete all private area |
| 4 | modify private area |
| 5 | get all private area list |
| 6 | get private area |
| 7 | goto private area |
| 8 | set all private area |

### PTZ operate type

|  |  |
| --- | --- |
| **Value** | **Explain** |
| -1 | invaild command |
| 0 | stop |
| 1 | rotate up |
| 2 | rotate down |
| 3 | rotate left |
| 4 | rotate right |
| 5 | rotate leftup |
| 6 | rotate leftdown |
| 7 | rotate rightup |
| 8 | rotate rightdown |
| 9 | zoom in |
| 10 | zoom out |
| 11 | far\_focus |
| 12 | near\_focus |
| 13 | increase iris |
| 14 | decrease iris |
| 15 | set preset |
| 16 | invoke preset |
| 17 | remove preset |
| 18 | set trace |
| 19 | invoke trace |
| 20 | remove trace |
| 21 | scan |
| 22 | set scan startpoint |
| 23 | set scan stoppoint |
| 24 | run auto focus |
| 25 | run auto iris |
| 26 | start auto study |
| 27 | end auto study |
| 28 | run auto study |
| 29 | reset |
| 30 | three-dimensional orientation |
| 31 | set tour startpoint |
| 32 | add tour presetpoint |
| 33 | set tour endpoint |
| 34 | run tour |
| 35 | stop tour |
| 36 | delete tour |
| 37 | set keeper |
| 38 | run keeper |
| 39 | run brush |
| 40 | open light |
| 41 | close light |
| 44 | /remove scan |
| 45 | remove auto study |
| 46 | IR control |
| 47 | get PTZ position require |
| 48 | get PTZ position response |
| 49 | set PTZ position |
| 50 | set ptz north position |
| 51 | get preset require |
| 52 | get preset response |
| 53 | get tour require |
| 54 | get tour response |
| 55 | get scan require |
| 56 | get scan response |
| 57 | get auto study require |
| 58 | /get auto study response |
| 59 | get keeper require |
| 60 | get keeper rsponse |
| 61 | IR control extend |
| 62 | IR control response |
| 63 | stop brush |
| 64 | 360° rotate scan |
| 65 | perpendicular scan |
| 66 | heart beat |
| 67 | IR control response |
| 70 | get alarm IO status |
| 71 | alarm IO status response |
| 72 | PT stop status query |
| 73 | auto upload PT coordinate |
| 74 | auto upload IO alarm status |
| 75 | get zoom value |
| 76 | get PTZ version |
| 77 | get MCU temperature |
| 78 | set to default |
| 79 |  |
| 80 |  |

### 1.5 PTZ parameter configuration

|  |  |
| --- | --- |
| **Value** | **Explain** |
| 0 | config preset |
| 1 | config scan |
| 2 | config track |
| 3 | config tour |
| 4 | config keeper |
| 5 | set PTZ speed |
| 6 | get PTZ speed |

### Front end configuration type

##### 1.6.1 Time slot

|  |  |
| --- | --- |
| **Value** | **Explain** |
| 8000 | Enable （0：close 1：open） |
| 8001 | Start time (hours 0-24) |
| 8002 | Start time (minutes 0 10 20 30 40 50) |
| 8003 | End time (hours 0-24) |
| 8004 | End time (minutes 0 10 20 30 40 50) |

##### 1.6.2 Image

|  |  |
| --- | --- |
| **Value** | **Explain** |
| 34 | Brightness (0 - 100) |
| 2321 | Sharpness （0 - 100） |
| 2353 | Contrast （0 - 100） |
| 2337 | saturation（0 - 100） |

##### 1.6.3 Scenes

|  |  |
| --- | --- |
| **Value** | **Explain** |
| 176 | Scene (0: indoor 1: outdoor) |
| 2384 | Mirror (0: off 1: level 2: vertical 3: horizontal + vertical) |
| 2385 | Corridor mode (0: off 1: on) |

##### 1.6.4 Exposure

|  |  |
| --- | --- |
| **Value** | **Explain** |
| 16 | Exposure mode (0: Auto 1: Manual 2: Shutter priority) |
| 96 | Metering area (4: global 0: center point 1: center area) |
| 80 | aperture（1：automatic； 0：Fully open； 1048576：close；256：F1.0； 257：F1.1； 258：F1.2；259：F1.3；260：F1.4；262：F1.6； 263：F1.7；264：F1.8；512：F2.0；514：F2.2；516：F2.4；518：F2.6；520：F2.8；770：F3.2；772：F3.4；774：F3.6；1024：F4.0；1029：F4.5；1032：F4.8； 1280： F5.0；1286：F5.6；1539：F6.3；1544：F6.8；1793：F7.1； 2048：F8.0； 2304：F9.0； 2310：F9.6；4096：F10.0；4352：F11.0；4864：F13.0；5120：F14.0；5632：F16.0； 6144：F18.0；6400：F19.0；8192：F20.0；8704：F22.0；9472：F25.0；9984：F27.0；10496：F29.0；12800：F32.0；13824：F36.0；14336：F38.0；16384：F40.0；17664：F45.0；20992：F52.0；21504：F54.0；22528：F58.0；25600：F64.0；） |
| 81 | Aperture speed (automatic mode is valid) -1: This function is not supported |
| 49 | Maximum shutter（Exposure auto mode is active 0 ：1/1， 1：1/2，2：1/5，3：1/10，4：1/12.5，5：1/15，6 ：1/20 ，7 ：1/25，8 ：1/30，9 ：1/50，10 ：1/60，11：1/100，12：1/120，13：1/125，14 ：1/150，15：1/200，16 ：1/250，17 ：1/500，18 : 1/1000，19：1/2000，20：1/5000，21 ：1/10000，22：1/20000，23：1/50000，24：1/100000，25：1/200000） |
| 50 | Fixed shutter（Exposure manual mode and shutter priority are effective 0 ：1/1， 1：1/2，2：1/5，3：1/10，4：1/12.5，5：1/15，6 ：1/20 ，7 ：1/25，8 ：1/30，9 ：1/50，10 ：1/60，11 ：1/100，12 ：1/120，13 ：1/125，14 ：1/150，15 ：1/200，16 ：1/250，17 ：1/500，18 : 1/1000，19 ：1/2000，20 ：1/5000，21 ：1/10000，22 ：1/20000，23 ：1/50000，24 ：1/100000，25 ：1/200000） |
| 65 | Maximum gain (exposure auto mode and shutter priority valid 0-100) |
| 66 | Fixed gain (exposure manual mode active 0 - 100) |

##### 1.6.5 White balance

|  |  |
| --- | --- |
| **Value** | **Explain** |
| 768 | Mode (0: Auto 1: Tungsten 2: Fluorescent 3: Fluorescent 4: Shadow 9: Manual) |
| 770 | Red gain (manual mode valid 0 - 100) |
| 771 | Blue gain (manual mode valid 0 - 100) |

##### 1.6.6 Day and night

|  |  |
| --- | --- |
| **Value** | **Explain** |
| 144 | Day and night mode (0: Auto 1: Day mode 2: Night mode 3: Timing) |
| 145 | Day to night threshold （0 - 100） |
| 146 | Night turn day threshold （0 - 100） |
| 147 | Delay (unit: second, range: 0 - 180) |
| 149 | Day to night (hours 0 - 24) |
| 150 | Day to night (0 10 20 30 40 50) |
| 151 | Turnover time (hours 0 - 24) |
| 152 | Turnover time (0 10 20 30 40 50) |
| 160 | Infrared light switch (0: off 1: open) |
| 161 | Infrared light mode (1: Auto 2: Manual) |
| 163 | Infrared light intensity (0 - 100) |

##### 1.6.7 Noise reduction

|  |  |
| --- | --- |
| **Value** | **Explain** |
| 2304 | 2D noise reduction (0: off 1: on) |
| 2305 | 2D mode (1: Auto 2: Manual) |
| 2307 | 2D manual strength (2D noise reduction manual mode is valid 0 - 100) |
| 2306 | 2D maximum intensity (2D noise reduction automatic mode valid 0 - 100) |
| 2308 | 3D noise reduction (0: off 1: on) |
| 2309 | 3D mode (1: Auto 2: Manual) |
| 2313 | 3D manual strength (3D noise reduction manual mode valid 0 - 100) |
| 2312 | 3D maximum intensity (3D noise reduction automatic mode is valid 0 - 100) |

##### 1.6.8 Image enhancement

|  |  |
| --- | --- |
| **Value** | **Explain** |
| 2368 | Wide dynamic switch (0: off 1: on) |
| 2369 | Wide dynamic value (0 - 100) |
| 112 | Strong light suppression switch (0: off 1: on) |
| 113 | Light suppression value (0 - 100) |
| 128 | Backlight compensation switch (0: off 1: on) |
| 130 | Backlight compensation value (0 - 100) |
| 2416 | Electronic image stabilization (0: off 1: on) |
| 2400 | Defogging switch (0: off 1: on) |
| 2401 | Defogging value (0 - 100) |

##### 1.6.9 Thermal Imaging

|  |  |
| --- | --- |
| **Value** | **Explain** |
| 2384 | Mirror (parameters: 0: normal; 1: around; 2: vertical; 3: horizontal + vertical) |
| 8201 | Pseudo color mode(Parameters: 0: white heat, 1: black heat, 2: rainbow, 3: iron red, 4: amber, 5: bone china, 6: cold tone, 7: brass, 8: hot, 9: white, 10: Spring Flowers, 11: Midsummer, 12: Autumn Leaves, 13: Long Winter, 14: Purple Blue, 15: Red Flame, 16: Rose) |
| 8202 | Electronic amplification (parameter: 0: off; 1:2 times; 2:4 times) |
| 8203 | Video gain (parameter: 0: automatic; 1: manual) |
| 8204 | Video gain (parameter: 0: automatic; 1: manual) |
| 8205 | Trigger correction |
| 8206 | Brightness (0 - 100) |
| 8207 | Focus step |
| 8208 | DVE type |
| 8217 | Temperature bar display (parameters: 1: on, 2: off) |

### 1.7 Log type

#### 1.7.1 Main type

|  |  |
| --- | --- |
| **Value (HEX)** | **Explain** |
| 0x2 | Abnormal log |
| 0x3 | Operation log |
| 0x4 | Operation log V2 |

#### 1.7.2 Sub type

|  |  |
| --- | --- |
| **Value (HEX)** | **Explain** |
| 0x01 | user management |
| 0x02 | system maintenance |
| 0x03 | Equipment configuration |
| 0x04 | Video operation |
| 0x05 | Audio and video control |
| 0x06 | Audio and video on demand |
| 0x07 | Web access mode and SSL encryption configuration |
| 0x11 | NVR user management |
| 0x12 | NVR configuration |
| 0x13 | NVR channel management |
| 0x14 | Video operation |
| 0x15 | Audio and video |
| 0x21 | Signal loss |
| 0x22 | Illegal access |
| 0x23 | Disk full |
| 0x24 | disk error |
| 0x25 | MODEM drop line |
| 0x26 | IP address conflict |
| 0x27 | Disk does not exist |
| 0x28 | disk write protected |
| 0x29 | Disk is not formatted |
| 0x30 | Alarm recording disk full |
| 0x31 | Scheduled recording disk full |
| 0x32 | 7 \* 24 recording disk full |
| 0x41 | Boot up |
| 0x42 | Shutdown |
| 0x43 | Illegal shutdown |
| 0x50 | Local landing |
| 0x51 | Log out locally |
| 0x52 | Local configuration parameters |
| 0x53 | Local playback by file |
| 0x54 | Local playback by time |
| 0x55 | Start recording locally |
| 0x56 | Stop recording locally |
| 0x57 | Local PTZ control |
| 0x58 | Local Preview |
| 0x59 | Local modification time |
| 0x5a | Local upgrade |
| 0x5b | Local backup files |
| 0x70 | Remote login |
| 0x71 | Remote logout login |
| 0x72 | Remote start recording |
| 0x73 | Remote stop recording |
| 0x74 | Start transparent transfer |
| 0x75 | Stop transparent transfer |
| 0x76 | Get parameters remotely |
| 0x77 | Remote configuration parameters |
| 0x78 | Get status remotely |
| 0x79 | Remote defense |
| 0x7a | Remote withdrawal |
| 0x7b | Remote Reboot |
| 0x7c | Start voice talk |
| 0x7d | Stop voice talk |
| 0x7e | Remote upgrade |
| 0x7f | Remote playback by file |
| 0x80 | Remote playback by time |
| 0x81 | Remote pan tilt control |
| 0x82 | Start live video remotely |
| 0x83 | Remote stop of real-time video |
| 0x84 | Remote start real time audio |
| 0x85 | Remote stop of real-time audio |
| 0x86 | Device storage format (SD card format) |

### 1.8 Alarm source type

|  |  |
| --- | --- |
| **Value** | **Explain** |
| 0 | Unknown alarm source |
| 1 | IO alarm |
| 2 | Motion detection alarm |
| 3 | Disk alarm |
| 4 | Video alarm |
| 5 | Network alarm |
| 6 | Video loss alarm source |
| 7 |  |
| 8 | Intelligent analysis perimeter intrusion alarm |
| 9 | Intelligent analysis Warning line alarm |
| 10 | Intelligent analysis Double alarm line alarm |
| 11 | Intelligent analysis Walk alarm |
| 12 | Intelligent analysis Multi-person alarm |
| 13 | Intelligent analysis Item legacy alarm |
| 14 | Intelligent analysis Item removal alarm |
| 15 | Intelligent analysis Abnormal speed alarm |
| 16 | Intelligent analysis Retrograde alarm |
| 17 | Intelligent analysis Illegal parking alarm |
| 18 | Intelligent analysis Video signal abnormal alarm |
| 19 |  |
| 20 |  |
| 21 | Channel IO alarm |
| 22 | PIR alarm |
| 23 | Temperature detection alarm |
| 26 | Third-party IPS alarm |
| 27 | License Plate Recognition |
| 28 | Smart home NVR alarm |
| 29 | Face temperature alarm |
| 30 | Intelligent motion detection alarm |
| 31 | Temperature threshold alarm |
| 32 | Temperature threshold warning |
| 33 | Temperature difference alarm |
| 34 | Temperature difference warning |
| 35 | Temperature interval alarm |
| 36 | Face alarm |
| 37 | Humanoid alarm |
| 38 | Vehicle alarm |

### 1.9 Action type

|  |  |
| --- | --- |
| **Value** | **Explain** |
| 0 | Unknown type |
| 1 | I/O |
| 2 | SMTP |
| 3 | PTZ |
| 4 | Recording |
| 5 | Message |
| 6 | Log |
| 7 | FTP |
| 8 | Push |

### 1.10 Fisheye mode

##### 1.10.1 Fisheye video mode

|  |  |
| --- | --- |
| **Value** | **Explain** |
| 0 | Single channel mode |
| 1 | Multi-channel mode |

##### 1.10.2 Correction mode

|  |  |
| --- | --- |
| **Value** | **Explain** |
| 0 | Fisheye mode |
| 1 | Not implemented, normal mode |
| 2 | 360 degree panorama mode |
| 3 | Not implemented, PTZ mode |
| 4 | 1 Fisheye +3PTZ mode |
| 5 | 1 Fisheye +5PTZ mode |
| 6 | 1 Fisheye + 7PTZ mode |
| 7 | Not implemented, 1 fisheye +8PTZ mode |
| 8 | 4PTZ mode |
| 9 | 1O 4PTZ mode |
| 10 | 180 degree panorama |
| 11 | 1 fisheye +1 panorama + 3PTZ |
| 12 | 1 fisheye main stream + 1 fisheye stream + 3PTZ |
| 13 | 1 fisheye + 4PTZ |

##### 1.10.3 Installation mode

|  |  |
| --- | --- |
| **Value** | **Explain** |
| 0 | Wall mounted |
| 1 | Ceiling |
| 2 | Desktop |

### 1.11 Dynamic video mode

##### 1.11.1 Playback device type

|  |  |
| --- | --- |
| **Value** | **Explain** |
| 0 | Normal play type |
| 1 | Quadruple panorama playback |
| 2 | Fisheye play |

##### 1.11.2 Button action

|  |  |
| --- | --- |
| **Value** | **Explain** |
| 0 | Release button |
| 1 | Button press |

##### 1.11.3 Button press

|  |  |
| --- | --- |
| **Value** | **Explain** |
| 0 | Left button |
| 1 | intermediate |
| 2 | Right click |

##### 1.**11.4 Panoramic display mode**

|  |  |
| --- | --- |
| **Value** | **Explain** |
| 0 | Original image (panoramic preview) mode |
| 1 | Perspective mode |
| 2 | Crystal ball mode |
| 3 | Fisheye mode |
| 4 | Asteroid mode |

##### 1.11.5 Fisheye installation

|  |  |
| --- | --- |
| **Value** | **Explain** |
| 0 | Ceiling installation |
| 1 | Desktop installation |
| 2 | Wall mounting |

##### 1.11.6 Fisheye display mode (for ceiling mount and desktop mount)

|  |  |
| --- | --- |
| **Value** | **Explain** |
| 0 | Fisheye original |
| 1 | Fisheye Original + 3PTZ |
| 2 | Fisheye Original + 8PTZ |
| 3 | 180° panorama |
| 4 | 360° panorama |
| 5 | 360° panorama + 1PTZ |
| 6 | 360° panorama + 3PTZ |
| 7 | 360° panorama + 6PTZ |
| 8 | 360° panorama + 8PTZ |
| 9 | 2PTZ |
| 10 | 4PTZ |
| 11 | Fisheye Original + 1PTZ + 360° Panorama |

##### 1.11.7 Fisheye display mode (for wall mounting)

|  |  |
| --- | --- |
| **Value** | **Explain** |
| 0 | Fisheye original |
| 3 | Fisheye panorama |

### 1.12 Equipment type

|  |  |
| --- | --- |
| **Value** | **Explain** |
| 1 | Network camera equipment |
| 2 | Digital video recorder equipment |
| 3 | Digital video server equipment |
| 4 | Network speed dome |
| 5 | NVR |
| 6 | Onvif device |
| 7 | decoder |
| 8 | License plate recognition camera |
| 9 | Fisheye equipment type |
| 10 | NVR labeled 4.0 |
| 11 | Quadocular panoramic camera |
| 13 | Thermal imaging equipment |
| 14 | Human body thermometer |
| 15 | Face Detection |
| 16 | Domestic license plate - Zhixin |
| 17 | Thermal imaging double warehouse ordinary temperature measuring equipment |
| 18 | AI MULTI OBJECT DETECT |
| 19 | Thermal imaging double warehouse ordinary temperature measuring equipment (visible light connection movement) |
| 20 | Zhongan Yuntai double ip connected pmd1030 movement |
| 21 | Face smart box |
| 100 | HK DVR |
| 101 | Allianz DVR |
| 102 | Dahua DVR |
| 103 | Used for compatible local servers used by NVR clients |
| 104 | Movement |

### 1.13 Language type

|  |  |
| --- | --- |
| **Value** | **Explain** |
| 1 | English |
| 2 | Simplified Chinese |
| 3 | Russian |
| 4 | French |
| 5 | Japanese |
| 6 | German |
| 7 | Spanish |
| 8 | Portuguese |
| 9 | Polish language |
| 10 | Arabic |
| 11 | Bosnian |
| 12 | Bulgarian |
| 13 | Catalan |
| 14 | Traditional Chinese |
| 15 | Croatian |
| 16 | Czech |
| 17 | Danish language |
| 18 | Dutch |
| 19 | Estonian |
| 20 | Farsi |
| 21 | Filipino |
| 22 | Finnish language |
| 23 | Greek language |
| 24 | Hebrew |
| 25 | Hungarian language |
| 26 | Italian |
| 27 | Korean |
| 28 | Latvian |
| 29 | Lithuania language |
| 30 | Romania |
| 31 | Serbian |
| 32 | Slovak |
| 33 | Slovenian |
| 34 | Swedish language |
| 35 | Thai |
| 36 | Turkish language |
| 37 | Ukraine language |
| 38 | Vietnamese |
| 39 | Norwegian |

### 1.14 Coding type

|  |  |
| --- | --- |
| **Value** | **Explain** |
| 0 | MPEG4 Code |
| 1 | H264 Code |
| 2 | MJPEG Code |
| 3 | SVC Code |
| 4 | H264 main profile//Discard |
| 5 | H264 high profile//Discard |
| 6 | JPEG Code |
| 7 | H265(base) |
| 8 | H265(main) |
| 9 | H265(high) |
| 101 | G7231 Code |
| 102 | G711A Law coding |
| 103 | G711U Law coding |
| 104 | G722 Code |
| 105 | G726 Code |
| 106 | G729 Code |
| 107 | AMR Code |
| 108 | PCM Code，No coding |
| 109 | No audio |
| 201 | Mobile detection data |
| 202 | Occlusion detection data |
| 203 | Cross border detection data |
| 0xFF | Unknown coding |