



Device Network SDK (Security Control)

Developer Guide

Legal Information

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, THE DOCUMENT IS PROVIDED "AS IS" AND "WITH ALL FAULTS AND ERRORS". OUR COMPANY MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. IN NO EVENT WILL OUR COMPANY BE LIABLE FOR ANY SPECIAL, CONSEQUENTIAL, INCIDENTAL, OR INDIRECT DAMAGES, INCLUDING, AMONG OTHERS, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION OR LOSS OF DATA, CORRUPTION OF SYSTEMS, OR LOSS OF DOCUMENTATION, WHETHER BASED ON BREACH OF CONTRACT, TORT (INCLUDING NEGLIGENCE), OR OTHERWISE, IN CONNECTION WITH THE USE OF THE DOCUMENT, EVEN IF OUR COMPANY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES OR LOSS.

Contents

Chapter 1 Overview	1
1.1 Introduction	1
1.2 Product Scope	1
1.3 Update History	3
Chapter 2 Typical Applications	32
2.1 System Configuration	32
2.2 User Management	36
2.3 Alarm Module Configuration	38
2.3.1 Detector Configuration	38
2.3.2 Zone Configuration	44
2.3.3 Partition Configuration	45
2.3.4 Timer Configuration	46
2.3.5 Siren Configuration	48
2.3.6 Repeater Configuration	48
2.3.7 Output Module Configuration	49
2.3.8 Relay Configuration	49
2.3.9 Keyfob Configuration	51
2.3.10 Keypad Configuration	54
2.3.11 Panic Button Configuration	55
2.3.12 Card Configuration	55
2.3.13 Card Reader Configuration	57
2.3.14 Extension Module Configuration	57
2.3.15 Module Locking and Unlocking	58
2.3.16 Alarm Lamp Configuration	59
2.3.17 Access Module Configuration	60
2.4 Alarm and Event	61

2.4.1 Alarm/Event Notification	61
2.4.2 Data Uploading	66
2.5 Control and Operation	67
2.6 Status Monitoring	69
2.7 Capture and Recording	72
2.8 Maintenance	73
Chapter 3 API Reference	79
3.1 NET_DVR_AlarmHostClearAlarm	79
3.2 NET_DVR_AlarmHostSubSystemCloseAlarmChan	79
3.3 NET_DVR_AlarmHostSubSystemSetupAlarmChan	80
3.4 NET_DVR_BypassAlarmChan	80
3.5 NET_DVR_Cleanup	81
3.6 NET_DVR_FindAlarmHostLog	82
3.7 NET_DVR_FindNextAlarmHostLog	82
3.8 NET_DVR_GetAlarmDeviceUser	83
3.9 NET_DVR_GetBatteryVoltage	85
3.10 NET_DVR_GetDeviceAbility	85
3.11 NET_DVR_GetDeviceConfig	86
3.12 NET_DVR_GetDVRConfig	87
3.13 NET_DVR_GetErrorMsg	88
3.14 NET_DVR_GetLastError	89
3.15 NET_DVR_GetNextRemoteConfig	89
3.16 NET_DVR_GetSDKLocalCfg	90
3.17 NET_DVR_GetSTDAbility	91
3.18 NET_DVR_GetSTDConfig	91
3.19 NET_DVR_Init	92
3.20 NET_DVR_Login_V40	93
3.21 NET_DVR_Logout	94

3.22 NET_DVR_RemoteControl	94
3.23 NET_DVR_SendRemoteConfig	95
3.24 NET_DVR_SetAlarmDeviceUser	96
3.25 NET_DVR_SetAlarmHostOut	97
3.26 NET_DVR_SetConnectTime	97
3.27 NET_DVR_SetDeviceConfig	98
3.28 NET_DVR_SetDVRConfig	100
3.29 NET_DVR_SetSDKInitCfg	101
3.30 NET_DVR_SetSDKLocalCfg	102
3.31 NET_DVR_SetSTDConfig	103
3.32 NET_DVR_StartRemoteConfig	103
3.32.1 fRemoteConfigCallback	105
3.33 NET_DVR_STDXMLConfig	106
3.34 NET_DVR_StopRemoteConfig	107
3.35 NET_DVR_UnBypassAlarmChan	108
Appendix A. Appendixes	109
A.1 Data Structure	109
A.1.1 CHAR_ENCODE_CONVERT	109
A.1.2 DETECTOR_TYPE	110
A.1.3 NET_DVR_ALARMHOST_CID_ALL_MINOR_TYPE	111
A.1.4 NET_DVR_ALARMHOST_LOG_RET	115
A.1.5 NET_DVR_ALARMHOST_MAIN_STATUS_V51	133
A.1.6 NET_DVR_ALARMHOST_OTHER_STATUS_V51	136
A.1.7 NET_DVR_ALARMHOST_REPORT_CENTER_CFG_V40	138
A.1.8 NET_DVR_ALARMHOST_SEARCH_LOG_PARAM	141
A.1.9 NET_DVR_ALARMHOSTDIALCFG	156
A.1.10 NET_DVR_ALARMIN_PARAM_V50	157
A.1.11 NET_DVR_ALARM_CAPTRUE_CFG	161

A.1.12 NET_DVR_ALARM_DEVICE_USER	162
A.1.13 NET_DVR_ALARM_ISAPI_INFO	165
A.1.14 NET_DVR_ALARM_ISAPI_PICDATA	166
A.1.15 NET_DVR_ALARM_LAMP_CFG	167
A.1.16 NET_DVR_ALARMIN_SETUP	167
A.1.17 NET_DVR_ALARMOUT_PARAM	168
A.1.18 NET_DVR_CETTIFICATE_INFO	170
A.1.19 NET_DVR_CID_ALARM	171
A.1.20 NET_DVR_CONTROL_PARAM	173
A.1.21 NET_DVR_DEVICEINFO_V40	175
A.1.22 NET_DVR_DEVICEINFO_V30	178
A.1.23 NET_DVR_INIT_CFG_ABILITY	182
A.1.24 NET_DVR_IPADDR	183
A.1.25 NET_DVR_IPADDR_UNION	183
A.1.26 NET_DVR_LIST_INFO	183
A.1.27 NET_DVR_LOCAL_ABILITY_PARSE_CFG	184
A.1.28 NET_DVR_LOCAL_ASYNC_CFG	184
A.1.29 NET_DVR_LOCAL_BYTE_ENCODE_CONVERT	185
A.1.30 NET_DVR_LOCAL_CERTIFICATION	185
A.1.31 NET_DVR_LOCAL_CFG_TYPE_PTZ	186
A.1.32 NET_DVR_LOCAL_CHECK_DEV	187
A.1.33 NET_DVR_LOCAL_GENERAL_CFG	187
A.1.34 NET_DVR_LOCAL_LOG_CFG	189
A.1.35 NET_DVR_LOCAL_MEM_POOL_CFG	189
A.1.36 NET_DVR_LOCAL_MODULE_RECV_TIMEOUT_CFG	190
A.1.37 NET_DVR_LOCAL_PORT_MULTI_CFG	190
A.1.38 NET_DVR_LOCAL_PROTECT_KEY_CFG	191
A.1.39 NET_DVR_LOCAL_SDK_PATH	191

A.1.40 NET_DVR_LOCAL_STREAM_CALLBACK_CFG	191
A.1.41 NET_DVR_LOCAL_TALK_MODE_CFG	192
A.1.42 NET_DVR_LOCAL_TCP_PORT_BIND_CFG	192
A.1.43 NET_DVR_LOCAL_UDP_PORT_BIND_CFG	193
A.1.44 NET_DVR_MESSAGE_CALLBACK_PARAM_V51	194
A.1.45 NET_DVR_MIME_UNIT	194
A.1.46 NET_DVR_MODULE_INFO	195
A.1.47 NET_DVR_PHONECENTERDIALCFG	196
A.1.48 NET_DVR_RECORD_PASSBACK_MANUAL_COND	197
A.1.49 NET_DVR_RECORD_PASSBACK_MANUAL_TASK_RET	198
A.1.50 NET_DVR_REMOTECONTROLLER_PERMISSION_CFG	198
A.1.51 NET_DVR_RTSP_PARAMS_CFG	199
A.1.52 NET_DVR_SCHEDTIME	200
A.1.53 NET_DVR_SIMXML_LOGIN	200
A.1.54 NET_DVR_SIP_CFG_V50	201
A.1.55 NET_DVR_STD_ABILITY	203
A.1.56 NET_DVR_STD_CONFIG	204
A.1.57 NET_DVR_STREAM_INFO	205
A.1.58 NET_DVR_TIME	205
A.1.59 NET_DVR_TIME_EX	206
A.1.60 NET_DVR_USER_LOGIN_INFO	206
A.1.61 NET_DVR_XML_CONFIG_INPUT	208
A.1.62 NET_DVR_XML_CONFIG_OUTPUT	209
A.1.63 NET_SDK_CALLBACK_STATUS_NORMAL	210
A.1.64 NET_SDK_LOCAL_CFG_TYPE	210
A.2 Request URIs	212
A.2.1 /ISAPI/SecurityCP/BasicParam/AlarmLampSchedTimeConfig	214
A.2.2 /ISAPI/SecurityCP/BasicParam/AlarmLampSchedTimeConfig/capabilities	215

A.2.3 /ISAPI/SecurityCP/BasicParam/audioFileList/capabilities	216
A.2.4 /ISAPI/SecurityCP/BasicParam/audioFileList/type=	216
A.2.5 /ISAPI/SecurityCP/BasicParam/audioInOutCfg	216
A.2.6 /ISAPI/SecurityCP/BasicParam/audioInOutCfg/capabilities	217
A.2.7 /ISAPI/SecurityCP/BasicParam/DetectorCfg	218
A.2.8 /ISAPI/SecurityCP/BasicParam/DetectorCfg/capabilities	218
A.2.9 /ISAPI/SecurityCP/BasicParam/ZoneAssociatedDetectorCfg	219
A.2.10 /ISAPI/SecurityCP/BasicParam/ZoneAssociatedDetectorCfg/capabilities	219
A.2.11 /ISAPI/SecurityCP/capabilities?format=json	220
A.2.12 /ISAPI/SecurityCP/CheckResult/capabilities?format=json	220
A.2.13 /ISAPI/SecurityCP/CheckResult?format=json	221
A.2.14 /ISAPI/SecurityCP/Configuration/accessModule/addType/capabilities?format=json	221
A.2.15 /ISAPI/SecurityCP/Configuration/accessModule/addType?format=json	221
A.2.16 /ISAPI/SecurityCP/Configuration/ARC/capabilities?format=json	222
A.2.17 /ISAPI/SecurityCP/Configuration/ARC/<ID>?format=json	222
A.2.18 /ISAPI/SecurityCP/Configuration/ARC/manualTest/capabilities?format=json	223
A.2.19 /ISAPI/SecurityCP/Configuration/ARC/manualTest/status?format=json	223
A.2.20 /ISAPI/SecurityCP/Configuration/ARC/manualTest?format=json	224
A.2.21 /ISAPI/SecurityCP/Configuration/ARC?format=json	224
A.2.22 /ISAPI/SecurityCP/Configuration/capabilities?format=json	225
A.2.23 /ISAPI/SecurityCP/Configuration/card/capabilities?format=json	225
A.2.24 /ISAPI/SecurityCP/Configuration/card/currentAdd?format=json	225
A.2.25 /ISAPI/SecurityCP/Configuration/card/currentAddAsyn?format=json	226
A.2.26 /ISAPI/SecurityCP/Configuration/card/mode/capabilities?format=json	227
A.2.27 /ISAPI/SecurityCP/Configuration/card/mode?format=json	227
A.2.28 /ISAPI/SecurityCP/Configuration/card/<ID>?format=json	227
A.2.29 /ISAPI/SecurityCP/Configuration/card?format=json	228

A.2.30 /ISAPI/SecurityCP/Configuration/cardReader/<ID>?format=json	229
A.2.31 /ISAPI/SecurityCP/Configuration/cardReader/capabilities?format=json	229
A.2.32 /ISAPI/SecurityCP/Configuration/cardReader/currentAddAsyn?format=json	230
A.2.33 /ISAPI/SecurityCP/Configuration/cardReader/mode/capabilities?format=json	230
A.2.34 /ISAPI/SecurityCP/Configuration/cardReader/mode?format=json	231
A.2.35 /ISAPI/SecurityCP/Configuration/cardReader?format=json	231
A.2.36 /ISAPI/SecurityCP/Configuration/curtainInfraredDetector/capabilities?format=json	232
A.2.37 /ISAPI/SecurityCP/Configuration/curtainInfraredDetector/zone/<ID>?format=json	232
A.2.38 /ISAPI/SecurityCP/Configuration/curtainInfraredDetector?format=json	233
A.2.39 /ISAPI/SecurityCP/Configuration/deviceTime/capabilities?format=json	233
A.2.40 /ISAPI/SecurityCP/Configuration/deviceTime?format=json	234
A.2.41 /ISAPI/SecurityCP/Configuration/eventRecord/channels/<ID>/capabilities? format=json	235
A.2.42 /ISAPI/SecurityCP/Configuration/eventRecord/channels/<ID>?format=json	235
A.2.43 /ISAPI/SecurityCP/Configuration/extensionModule/<ID>?format=json	236
A.2.44 /ISAPI/SecurityCP/Configuration/extensionModule/capabilities?format=json	236
A.2.45 /ISAPI/SecurityCP/Configuration/extensionModule?format=json	237
A.2.46 /ISAPI/SecurityCP/Configuration/faultCheckCfg/capabilities?format=json	237
A.2.47 /ISAPI/SecurityCP/Configuration/faultCheckCfg?format=json	237
A.2.48 /ISAPI/SecurityCP/Configuration/glassBreakDetector/capabilities?format=json ...	238
A.2.49 /ISAPI/SecurityCP/Configuration/glassBreakDetector?format=json	239
A.2.50 /ISAPI/SecurityCP/Configuration/glassBreakDetector/zone/<ID>?format=json	239
A.2.51 /ISAPI/SecurityCP/Configuration/indoorDualTechnologyDetector/capabilities? format=json	240
A.2.52 /ISAPI/SecurityCP/Configuration/indoorDualTechnologyDetector/zone/<ID>? format=json	240
A.2.53 /ISAPI/SecurityCP/Configuration/indoorDualTechnologyDetector?format=json ...	241
A.2.54 /ISAPI/SecurityCP/Configuration/keypad/<ID>?format=json	242

A.2.55 /ISAPI/SecurityCP/Configuration/keypad/capabilities?format=json	242
A.2.56 /ISAPI/SecurityCP/Configuration/keypad?format=json	243
A.2.57 /ISAPI/SecurityCP/Configuration/keypadAddList/capabilities?format=json	243
A.2.58 /ISAPI/SecurityCP/Configuration/keypadAddList?format=json	243
A.2.59 /ISAPI/SecurityCP/Configuration/keypadFaultProcessCfg/<ID>?format=json	244
A.2.60 /ISAPI/SecurityCP/Configuration/keypadFaultProcessCfg/capabilities?format=json	244
A.2.61 /ISAPI/SecurityCP/Configuration/keypadFaultProcessCfg?format=json	245
A.2.62 /ISAPI/SecurityCP/Configuration/magneticContact/capabilities?format=json	245
A.2.63 /ISAPI/SecurityCP/Configuration/magneticContact/zone/<ID>?format=json	246
A.2.64 /ISAPI/SecurityCP/Configuration/magneticContact?format=json	247
A.2.65 /ISAPI/SecurityCP/Configuration/messageSendARC/capabilities?format=json	247
A.2.66 /ISAPI/SecurityCP/Configuration/messageSendARCList?format=json	248
A.2.67 /ISAPI/SecurityCP/Configuration/messageSendARC?format=json	248
A.2.68 /ISAPI/SecurityCP/Configuration/messageSendCloud/capabilities?format=json ...	249
A.2.69 /ISAPI/SecurityCP/Configuration/messageSendCloud?format=json	249
A.2.70 /ISAPI/SecurityCP/Configuration/messageSendDirect/capabilities?format=json ..	250
A.2.71 /ISAPI/SecurityCP/Configuration/messageSendDirect?format=json	250
A.2.72 /ISAPI/SecurityCP/Configuration/messageSendMail/<ID>?format=json	251
A.2.73 /ISAPI/SecurityCP/Configuration/messageSendMail?format=json	251
A.2.74 /ISAPI/SecurityCP/Configuration/messageSendMail/capabilities?format=json	252
A.2.75 /ISAPI/SecurityCP/Configuration/messageSendPhone/<ID>?format=json	252
A.2.76 /ISAPI/SecurityCP/Configuration/messageSendPhone/capabilities?format=json ..	253
A.2.77 /ISAPI/SecurityCP/Configuration/messageSendPhone?format=json	253
A.2.78 /ISAPI/SecurityCP/Configuration/messageSendPhoneAnvanced/<ID>?format=json	253
A.2.79 /ISAPI/SecurityCP/Configuration/messageSendPhoneAnvanced/capabilities? format=json	254
A.2.80 /ISAPI/SecurityCP/Configuration/messageSendPhoneAnvanced?format=json	255

A.2.81 /ISAPI/SecurityCP/Configuration/muteVoicePlanCFG/capabilities?format=json ...	255
A.2.82 /ISAPI/SecurityCP/Configuration/muteVoicePlanCFG?format=json	256
A.2.83 /ISAPI/SecurityCP/Configuration/notRelateZones/capabilities?format=json	257
A.2.84 /ISAPI/SecurityCP/Configuration/notRelateZones?format=json	257
A.2.85 /ISAPI/SecurityCP/Configuration/outputModules/<ID>?format=json	257
A.2.86 /ISAPI/SecurityCP/Configuration/outputModules/capabilities?format=json	258
A.2.87 /ISAPI/SecurityCP/Configuration/outputModules?format=json	258
A.2.88 /ISAPI/SecurityCP/Configuration/outputs/<ID>?format=json	259
A.2.89 /ISAPI/SecurityCP/Configuration/outputs/capabilities?format=json	259
A.2.90 /ISAPI/SecurityCP/Configuration/outputs?format=json	260
A.2.91 /ISAPI/SecurityCP/Configuration/outputsModule/<ID>?format=json	261
A.2.92 /ISAPI/SecurityCP/Configuration/outputsModule/capabilities?format=json	261
A.2.93 /ISAPI/SecurityCP/Configuration/outputsModule?format=json	262
A.2.94 /ISAPI/SecurityCP/Configuration/panicButton/capabilities?format=json	262
A.2.95 /ISAPI/SecurityCP/Configuration/panicButton/zone/<ID>?format=json	263
A.2.96 /ISAPI/SecurityCP/Configuration/panicButton?format=json	263
A.2.97 /ISAPI/SecurityCP/Configuration/passiveInfraredDetector/capabilities?format=json	264
A.2.98 /ISAPI/SecurityCP/Configuration/passiveInfraredDetector/zone/<ID>?format=json	264
A.2.99 /ISAPI/SecurityCP/Configuration/passiveInfraredDetector?format=json	265
A.2.100 /ISAPI/SecurityCP/Configuration/pircam/capabilities?format=json	266
A.2.101 /ISAPI/SecurityCP/Configuration/pircam/zone/<ID>?format=json	266
A.2.102 /ISAPI/SecurityCP/Configuration/pircam/zone?format=json	267
A.2.103 /ISAPI/SecurityCP/Configuration/PSTNCfg/<ID>?format=json	267
A.2.104 /ISAPI/SecurityCP/Configuration/PSTNCfg/capabilities?format=json	268
A.2.105 /ISAPI/SecurityCP/Configuration/PSTNCfg?format=json	268
A.2.106 /ISAPI/SecurityCP/Configuration/publicSubSys/<ID>?format=json	269
A.2.107 /ISAPI/SecurityCP/Configuration/publicSubSys/capabilities?format=json	269

A.2.108 /ISAPI/SecurityCP/Configuration/publicSubSys?format=json	270
A.2.109 /ISAPI/SecurityCP/Configuration/registerMode/capabilities?format=json	270
A.2.110 /ISAPI/SecurityCP/Configuration/registerMode/registerStatus?format=json	271
A.2.111 /ISAPI/SecurityCP/Configuration/registerMode?format=json	271
A.2.112 /ISAPI/SecurityCP/Configuration/remoteCfgPermissonUserName/capabilities? format=json	272
A.2.113 /ISAPI/SecurityCP/Configuration/remoteCfgPermissonUserName?format=json .	272
A.2.114 /ISAPI/SecurityCP/Configuration/remoteCtrl/capabilities?format=json	273
A.2.115 /ISAPI/SecurityCP/Configuration/remoteCtrl/currentAdd?format=json	273
A.2.116 /ISAPI/SecurityCP/Configuration/remoteCtrl/currentAddAsyn?format=json	273
A.2.117 /ISAPI/SecurityCP/Configuration/remoteCtrl/mode/capabilities?format=json ...	274
A.2.118 /ISAPI/SecurityCP/Configuration/remoteCtrl/mode?format=json	275
A.2.119 /ISAPI/SecurityCP/Configuration/remoteCtrl/<ID>?format=json	275
A.2.120 /ISAPI/SecurityCP/Configuration/remoteCtrl?format=json	276
A.2.121 /ISAPI/SecurityCP/Configuration/repeaters/capabilities?format=json	276
A.2.122 /ISAPI/SecurityCP/Configuration/repeaters/<ID>?format=json	277
A.2.123 /ISAPI/SecurityCP/Configuration/repeaters?format=json	277
A.2.124 /ISAPI/SecurityCP/Configuration/signalStrengthDetection/currentAsyn?format=json	278
A.2.125 /ISAPI/SecurityCP/Configuration/signalStrengthDetection/mode/capabilities? format=json	278
A.2.126 /ISAPI/SecurityCP/Configuration/signalStrengthDetection/mode?format=json ..	279
A.2.127 /ISAPI/SecurityCP/Configuration/slimMagneticContact/capabilities?format=json	279
A.2.128 /ISAPI/SecurityCP/Configuration/slimMagneticContact/zone/<ID>?format=json	280
A.2.129 /ISAPI/SecurityCP/Configuration/slimMagneticContact?format=json	281
A.2.130 /ISAPI/SecurityCP/Configuration/subSys/<ID>?format=json	281
A.2.131 /ISAPI/SecurityCP/Configuration/subSys/capabilities?format=json	282
A.2.132 /ISAPI/SecurityCP/Configuration/subSys?format=json	282

A.2.133 /ISAPI/SecurityCP/Configuration/subSysTime/capabilities?format=json	282
A.2.134 /ISAPI/SecurityCP/Configuration/subSysTime/<ID>?format=json	283
A.2.135 /ISAPI/SecurityCP/Configuration/subSysTime?format=json	283
A.2.136 /ISAPI/SecurityCP/Configuration/systemManage/capabilities?format=json	284
A.2.137 /ISAPI/SecurityCP/Configuration/systemManage?format=json	284
A.2.138 /ISAPI/SecurityCP/Configuration/users/capabilities?format=json	285
A.2.139 /ISAPI/SecurityCP/Configuration/users/<ID>?format=json	285
A.2.140 /ISAPI/SecurityCP/Configuration/users?format=json	286
A.2.141 /ISAPI/SecurityCP/Configuration/wiredDetector/capabilities?format=json	287
A.2.142 /ISAPI/SecurityCP/Configuration/wirelessSiren/capabilities?format=json	287
A.2.143 /ISAPI/SecurityCP/Configuration/wirelessSiren/<ID>?format=json	287
A.2.144 /ISAPI/SecurityCP/Configuration/wirelessSiren?format=json	288
A.2.145 /ISAPI/SecurityCP/Configuration/wirelessSiren/currentAddAsyn?format=json ...	288
A.2.146 /ISAPI/SecurityCP/Configuration/wirelessSiren/mode/capabilities?format=json	289
A.2.147 /ISAPI/SecurityCP/Configuration/wirelessSiren/mode?format=json	290
A.2.148 /ISAPI/SecurityCP/Configuration/zoneAlarmTimeFilter/capabilities?format=json	290
A.2.149 /ISAPI/SecurityCP/Configuration/zoneAlarmTimeFilter?format=json	290
A.2.150 /ISAPI/SecurityCP/Configuration/zones/capabilities?format=json	291
A.2.151 /ISAPI/SecurityCP/Configuration/zones/currentAddAsyn?format=json	292
A.2.152 /ISAPI/SecurityCP/Configuration/zones/<ID>?format=json	292
A.2.153 /ISAPI/SecurityCP/Configuration/zones?format=json	293
A.2.154 /ISAPI/SecurityCP/control/bypassRecover/<ID>?format=json	293
A.2.155 /ISAPI/SecurityCP/control/outputs?format=json	294
A.2.156 /ISAPI/SecurityCP/control/outputs/<ID>?format=json	294
A.2.157 /ISAPI/SecurityCP/control/clearAlarm/<ID>?format=json	294
A.2.158 /ISAPI/SecurityCP/control/arm/<ID>?ways=<string>&format=json	295
A.2.159 /ISAPI/SecurityCP/Control/audioFile/name=	296

A.2.160 /ISAPI/SecurityCP/control/disarm/<ID>?format=json	296
A.2.161 /ISAPI/SecurityCP/control/bypassRecover?format=json	296
A.2.162 /ISAPI/SecurityCP/control/bypass?format=json	297
A.2.163 /ISAPI/SecurityCP/control/bypass/<ID>?format=json	297
A.2.164 /ISAPI/SecurityCP/control/capabilities?format=json	298
A.2.165 /ISAPI/SecurityCP/control/siren/<ID>?format=json	298
A.2.166 /ISAPI/SecurityCP/control/systemFault?format=json	299
A.2.167 /ISAPI/SecurityCP/FileExport/pircam/capabilities?format=json	299
A.2.168 /ISAPI/SecurityCP/FileExport/pircam?format=json	299
A.2.169 /ISAPI/SecurityCP/Log/search?format=json	300
A.2.170 /ISAPI/SecurityCP/pircam/picture/channels/<ID>/mode?format=json	300
A.2.171 /ISAPI/SecurityCP/pircam/picture/channels/<ID>?format=json	301
A.2.172 /ISAPI/SecurityCP/pircam/picture/mode/capabilities?format=json	301
A.2.173 /ISAPI/SecurityCP/pircam/picture/channels/<ID>/currentAddAsyn?format=json	302
A.2.174 /ISAPI/SecurityCP/status/acPowerStatus?format=json	303
A.2.175 /ISAPI/SecurityCP/status/armStatus?format=json	303
A.2.176 /ISAPI/SecurityCP/status/batteries?format=json	303
A.2.177 /ISAPI/SecurityCP/status/capabilities?format=json	304
A.2.178 /ISAPI/SecurityCP/status/cardReaderStatus?format=json	304
A.2.179 /ISAPI/SecurityCP/status/communication?format=json	305
A.2.180 /ISAPI/SecurityCP/status/exDevStatus?format=json	305
A.2.181 /ISAPI/SecurityCP/status/extensionModuleStatus?format=json	305
A.2.182 /ISAPI/SecurityCP/status/host?format=json	306
A.2.183 /ISAPI/SecurityCP/status/hostItself?format=json	307
A.2.184 /ISAPI/SecurityCP/status/keypadStatus?format=json	307
A.2.185 /ISAPI/SecurityCP/status/outputModStatus?format=json	307
A.2.186 /ISAPI/SecurityCP/status/outputStatus?format=json	308

A.2.187 /ISAPI/SecurityCP/status/repeaterStatus?format=json	308
A.2.188 /ISAPI/SecurityCP/status/sirenStatus?format=json	309
A.2.189 /ISAPI/SecurityCP/status/subSystems?format=json	309
A.2.190 /ISAPI/SecurityCP/status/systemFault?format=json	310
A.2.191 /ISAPI/SecurityCP/status/zones?format=json	310
A.2.192 /ISAPI/SecurityCP/surroundEnvironmentCfg/capabilities?format=json	311
A.2.193 /ISAPI/SecurityCP/surroundEnvironmentCfg?format=json	311
A.2.194 /ISAPI/SecurityCP/sysAutoCheckTimeCfg/capabilities?format=json	312
A.2.195 /ISAPI/SecurityCP/sysAutoCheckTimeCfg?format=json	312
A.2.196 /ISAPI/SecurityCP/sysCheckManually/capabilities?format=json	313
A.2.197 /ISAPI/SecurityCP/sysCheckManually?format=json	313
A.2.198 /ISAPI/SecurityCP/videoBroadcast/customizeUpload?format=json	314
A.2.199 /ISAPI/SecurityCP/voicePrompt/capabilities?format=json	315
A.2.200 /ISAPI/SecurityCP/voicePrompt?format=json	315
A.2.201 /ISAPI/System/capabilities	316
A.2.202 /ISAPI/System/moduleLock/config/capabilities?format=json	316
A.2.203 /ISAPI/System/moduleLock/config?format=json	317
A.2.204 /ISAPI/System/moduleLock/unlockModule?format=json	318
A.3 Request and Response Messages	318
A.3.1 JSON_accessModuleAddResult	318
A.3.2 JSON_accessModuleAddType	318
A.3.3 JSON_AccessModuleAddTypeCap	319
A.3.4 JSON_accessModuleType	320
A.3.5 JSON_ACPowerStatus	320
A.3.6 JSON_AlarmHostStatus	320
A.3.7 JSON_AlarmHostStatusCond	323
A.3.8 JSON_ARC	324
A.3.9 JSON_ARCCap	327

A.3.10 JSON_ARCManualTest	333
A.3.11 JSON_ARCManualTestCap	334
A.3.12 JSON_ARCManualTestID	334
A.3.13 JSON_ARCManualTestStatus	334
A.3.14 JSON_ArmFault	334
A.3.15 JSON_ArmStatusList	336
A.3.16 JSON_BatteryList	338
A.3.17 JSON_Cap_CardMode	338
A.3.18 JSON_Cap_CardReaderMode	339
A.3.19 JSON_Cap_CheckResult	340
A.3.20 JSON_Cap_MuteVoicePlanCFG	340
A.3.21 JSON_Cap_PircamMode	341
A.3.22 JSON_Cap_RemoteCfgUserName	341
A.3.23 JSON_Cap_SysAutoCheckTimeCfg	342
A.3.24 JSON_Cap_SysCheckManually	342
A.3.25 JSON_Cap_voicePromptCfg	342
A.3.26 JSON_Card	343
A.3.27 JSON_CardCap	343
A.3.28 JSON_CardMode	345
A.3.29 JSON_CardReader	345
A.3.30 JSON_CardReaderCap	346
A.3.31 JSON_CardReaderList	347
A.3.32 JSON_CardReaderMode	348
A.3.33 JSON_CheckResult	348
A.3.34 JSON_Cloud	349
A.3.35 JSON_CloudCap	350
A.3.36 JSON_CommuniStatus	351
A.3.37 JSON_CurtainInfraredDetector	352

A.3.38 JSON_CurtainInfraredDetectorCap	352
A.3.39 JSON_DeviceTime	354
A.3.40 JSON_DeviceTimeCap	355
A.3.41 JSON_Direct	355
A.3.42 JSON_DirectCap	356
A.3.43 JSON_EventNotificationAlert_SecurityCPAlarmEventMsg	357
A.3.44 JSON_EventNotificationAlert_Alarm/EventInfo	362
A.3.45 JSON_EventRecord	363
A.3.46 JSON_EventRecordCap	363
A.3.47 JSON_ExDevStatus	364
A.3.48 JSON_ExtensionList	373
A.3.49 JSON_ExtensionModule	374
A.3.50 JSON_ExtensionModuleCap	375
A.3.51 JSON_FaultCheckParameter	376
A.3.52 JSON_FaultCheckParameterCap	377
A.3.53 JSON_FileExportCap	378
A.3.54 JSON_FileExportCond	378
A.3.55 JSON_FileExportInfo	378
A.3.56 JSON_GlassBreakDetector	378
A.3.57 JSON_GlassBreakDetectorCap	379
A.3.58 JSON_HostConfigCap	380
A.3.59 JSON_HostControlCap	384
A.3.60 JSON_HostStatus	386
A.3.61 JSON_HostStatusCap	386
A.3.62 JSON_IndoorDualTechnologyDetector	388
A.3.63 JSON_IndoorDualTechnologyDetectorCap	388
A.3.64 JSON_Keypad	389
A.3.65 JSON_KeypadAddList	391

A.3.66 JSON_KeypadAddListCap	391
A.3.67 JSON_KeypadCap	392
A.3.68 JSON_KeypadFaultProcessCfg	395
A.3.69 JSON_KeypadFaultProcessCfgCap	396
A.3.70 JSON_KeypadList	397
A.3.71 JSON_List_ARC	399
A.3.72 JSON_List_Card	402
A.3.73 JSON_List_CardReader	402
A.3.74 JSON_List_CurtainInfraredDetector	403
A.3.75 JSON_List_ExtensionModule	404
A.3.76 JSON_List_GlassBreakDetector	405
A.3.77 JSON_List_ID	406
A.3.78 JSON_List_IndoorDualTechnologyDetector	406
A.3.79 JSON_List_IPAddress	406
A.3.80 JSON_List_Keypad	407
A.3.81 JSON_List_KeypadFaultProcessCfg	409
A.3.82 JSON_List_MagneticContact	410
A.3.83 JSON_List_Mail	411
A.3.84 JSON_List_ModuleInfo	411
A.3.85 JSON_List_ModuleLock	411
A.3.86 JSON_List_Output	412
A.3.87 JSON_List_OutputModule	415
A.3.88 JSON_List_OutPutsModule	416
A.3.89 JSON_List_PanicButton	418
A.3.90 JSON_List_PassiveInfraredDetector	418
A.3.91 JSON_List_Phone	419
A.3.92 JSON_List_PhoneAnvanced	420
A.3.93 JSON_List_pircam	423

A.3.94 JSON_List_PSTNCfg	425
A.3.95 JSON_List_PublicSubSys	425
A.3.96 JSON_List_RemoteCtrl	426
A.3.97 JSON_List_Repeater	427
A.3.98 JSON_List_Siren	428
A.3.99 JSON_List_SlimMagneticContact	430
A.3.100 JSON_List_SubSys	430
A.3.101 JSON_List_SubSysTime	431
A.3.102 JSON_List_UserCfg	432
A.3.103 JSON_List_Zone	433
A.3.104 JSON_MagneticContact	433
A.3.105 JSON_MagneticContactCap	434
A.3.106 JSON_Mail	437
A.3.107 JSON_MailCap	437
A.3.108 JSON_Manage	437
A.3.109 JSON_ManageCap	439
A.3.110 JSON_ModuleLockCap	443
A.3.111 JSON_MuteVoicePlanCFG	443
A.3.112 JSON_NotRelateZones	444
A.3.113 JSON_NotRelateZonesCap	444
A.3.114 JSON_Operate	445
A.3.115 JSON_Output	445
A.3.116 JSON_OutputCap	448
A.3.117 JSON_OutputCond	454
A.3.118 JSON_OutputModList	454
A.3.119 JSON_OutputModule	456
A.3.120 JSON_OutputModuleCap	456
A.3.121 JSON_OutputsCtrl	457

A.3.122 JSON_OutputSearch_Config	458
A.3.123 JSON_OutputSearch_Status	460
A.3.124 JSON_OutPutsModule	462
A.3.125 JSON_OutPutsModuleCap	463
A.3.126 JSON_OutputsModuleCond	466
A.3.127 JSON_OutputsModuleSearch	466
A.3.128 JSON_PanicButton	469
A.3.129 JSON_PanicButtonCap	469
A.3.130 JSON_PassiveInfraredDetectorCap	471
A.3.131 JSON_PassiveInfraredDetector	473
A.3.132 JSON_Phone	473
A.3.133 JSON_Picture	474
A.3.134 JSON_pircam	474
A.3.135 JSON_Pircam	476
A.3.136 JSON_PircamCap	476
A.3.137 JSON_PircamMode	479
A.3.138 JSON_PSTNCfg	479
A.3.139 JSON_PSTNCfgCap	480
A.3.140 JSON_PhoneAnvanced	481
A.3.141 JSON_PhoneAnvancedCap	484
A.3.142 JSON_PhoneCap	488
A.3.143 JSON_PublicSubSys	489
A.3.144 JSON_PublicSubSysCap	489
A.3.145 JSON_RegisterMode	490
A.3.146 JSON_RegisterModeCap	490
A.3.147 JSON_RemoteCfgUserName	491
A.3.148 JSON_RemoteCtrl	491
A.3.149 JSON_RemoteCtrlCap	493

A.3.150 JSON_RemoteCtrlMode	495
A.3.151 JSON_RemoteCtrlModeCap	495
A.3.152 JSON_Repeater	496
A.3.153 JSON_RepeaterCap	496
A.3.154 JSON_RepeaterList	497
A.3.155 JSON_ResponseStatus	498
A.3.156 JSON_Result	498
A.3.157 JSON_SearchDescription	499
A.3.158 JSON_SearchResult	499
A.3.159 JSON_SecurityCPCap	500
A.3.160 JSON_SendARC	502
A.3.161 JSON_SendARCCap	503
A.3.162 JSON_SendARCList	505
A.3.163 JSON_SignalStrengthDetection	506
A.3.164 JSON_SignalStrengthDetectionCap	506
A.3.165 JSON_SignalStrengthDetectionMode	507
A.3.166 JSON_Siren	507
A.3.167 JSON_SirenCap	509
A.3.168 JSON_SirenList	514
A.3.169 JSON_SirenMode	515
A.3.170 JSON_SirenModeCap	516
A.3.171 JSON_SlimMagneticContact	516
A.3.172 JSON_SlimMagneticContactCap	517
A.3.173 JSON_SubSys	518
A.3.174 JSON_SubSysCap	518
A.3.175 JSON_SirenCtrl	519
A.3.176 JSON_SubSysList	519
A.3.177 JSON_SubSysTime	520

A.3.178 JSON_SubSysTimeCap	521
A.3.179 JSON_SurrondParaCap	523
A.3.180 JSON_SurrondParaCfg	524
A.3.181 JSON_SysAutoCheckTimeCfg	524
A.3.182 JSON_SysCheckManually	524
A.3.183 JSON_TimeCfg	525
A.3.184 JSON_TimeCfgCap	525
A.3.185 JSON_id	525
A.3.186 JSON_UserCfg	525
A.3.187 JSON_UserCfgCap	526
A.3.188 JSON_voicePromptCfg	527
A.3.189 JSON_WiredDetectorCap	528
A.3.190 JSON_WiredDetectorType	529
A.3.191 JSON_WirelessRecv	530
A.3.192 JSON_Zone	530
A.3.193 JSON_ZoneCond	535
A.3.194 JSON_ZoneList	535
A.3.195 JSON_ZonesCap	537
A.3.196 JSON_ZoneSearch	556
A.3.197 XML_AlarmHostAbility	559
A.3.198 XML_AudioFileList	586
A.3.199 XML_AudioInOutCfg	587
A.3.200 XML_CAMERAPARA	588
A.3.201 XML_Cap_AlarmLampConfig	609
A.3.202 XML_Cap_AudioFileList	609
A.3.203 XML_Cap_AudioInOutCfg	610
A.3.204 XML_Cap_DetectorCfg	611
A.3.205 XML_Cap_LampSchedTimeList	612

A.3.206 XML_Cap_ZoneAssociatedDetectorCfg	612
A.3.207 XML_Desc_AlarmHostAbility	612
A.3.208 XML_DeviceCap	613
A.3.209 XML_DetectorCfg	625
A.3.210 XML_EmergencyAlarmProductCap	625
A.3.211 XML_EventNotificationAlert_AlarmEventInfo	626
A.3.212 XML_IpViewDevAbility	627
A.3.213 XML_LampSchedTimeList	635
A.3.214 XML_ResponseStatus	635
A.3.215 XML_ZoneAssociatedDetectorCfg	636
A.3.216 XML_ZoneCondList	637
A.4 Device Network SDK Errors	637
A.5 Response Codes of Text Protocol	680
A.6 Error Codes Categorized by Functional Modules	719
A.7 Log Types for ISAPI	736

Chapter 1 Overview

This manual provides the integration methods and processes based on HCNetsDK for the applications of security control panel.

1.1 Introduction

A security control panel uses embedded microcontroller technology for monitoring arming zones, handling alarm signal from the triggers, and uploading alarm reports to the central alarm monitoring station through multiple transmission methods such as wired network, wireless network, and so on. Here in this manual, you can implement different functions, such as event/ alarm configuration, zone or partition control and operation, status monitoring, security control system management and configuration, user management, and so on, by accessing to security control panels via protocols.

1.2 Product Scope

The product scope shows the product types and models that support integrating the applications in this manual based on HCNetsDK.

Table 1-1 Supported Protocol Type and Model

Product Type	Product Model
Wireless Security Control Panel	DS-PWA32-H
	DS-PWA32-HG
	DS-PWA32-HS
	DS-PWA32-HR
	DS-PWA32-HGR
	DS-PWA32-HSR
	DS-PWA32-K
	DS-PWA32-KG
	DS-PWA32-KS
	DS-PWA32-KT
	DS-PWA32-KGT
	DS-PWA32-KST

Product Type	Product Model
	DS-PWA32-N
	DS-PWA32-NG
	DS-PWA32-NS
	DS-PWA32-NT
	DS-PWA32-NGT
	DS-PWA32-NST
	DS-PWA32-NKT
	DS-PWA32-NK
	DS-PWA32-NKGT
	DS-PWA32-NKG
	DS-PWA32-NKST
	DS-PWA32-NKS
Network Security Control Panel	DS-19A08-BN
	DS-19A08-BNG
	DS-19A16-BN
	DS-19A16-BNG
	DS-19A08-01BNE
	DS-19A08-01BNG
	DS-19A08-01BN
	DS-19A08-F/K1
	DS-19A08-F/K1G
	DS-19A08-F/K2
	DS-19A08-F/K2G
Hybrid Security Control Panel	DS-PHA20-M
	DS-PHA64-M
	DS-PHA20-W2M
	DS-PHA64-LP
	DS-PHA64-W4M

Product Type	Product Model
	DS-PHA20-P
	DS-PHA20-W2P
	DS-PHA20-B
	DS-PHA64-B
	DS-PHA20-W2B
	DS-PHA64-W4B
	DS-PHA64-P2
	DS-PHA64-W4P2
Wired Keypad	DS-PKG-H4L
	DS-PKG-H8L
RS-485 Wireless Receiver	DS-PM-RSWR-868
	DS-PM-RSWR-433

1.3 Update History

Summary of Changes in Version 6.1.7.X_July, 2021

- Extended the configuration capability of the registration mode ***JSON_RegisterModeCap*** (related URI: [/ISAPI/SecurityCP/Configuration/registerMode/capabilities?format=json](#) ; related API: ***NET_DVR_STDXMLConfig***):
added three nodes: **exDevType** (peripheral module type), **wirelessKeypadRecvAddress** (wireless receiving module address of the wireless keypad), and **wirelessRemoteCtrlRecvAddress** (wireless receiving module address of the wireless keyfob).
- Extended the message about the parameters of the registration mode ***JSON_RegisterMode*** (related URI: [/ISAPI/SecurityCP/Configuration/registerMode?format=json](#) ; related API: ***NET_DVR_STDXMLConfig***):
added a node **wirelessRecvAddress** (wireless receiving module address).
- Extended the message about keypad configuration capability ***JSON_KeypadCap*** , the message about the parameters of a keypad ***JSON_Keypad*** , and the message about the parameters of all keypads ***JSON_List_Keypad*** (related URIs: [/ISAPI/SecurityCP/Configuration/keypad/capabilities?format=json](#) , [/ISAPI/SecurityCP/Configuration/keypad/<ID>?format=json](#) , and [/ISAPI/SecurityCP/Configuration/keypad?format=json](#) ; related API: ***NET_DVR_STDXMLConfig***):
added three nodes: **armAndDisarmAuthorityCfg** (permission for arming/disarming the partition), **alarmBuzzerEnabled** (whether to enable alarm buzzer) and **buttonBuzzerEnabled** (whether to enable button buzzer).

4. Extended the message about keypad status **JSON KeypadList** (related URI: **/ISAPI/SecurityCP/status/keypadStatus?format=json** ; related API: **NET_DVR_STDXMLConfig**): added a node **ZoneList** (list of linked zones).
5. Extended the message about siren configuration capability **JSON SirenCap** and the message about all sirens' parameters **JSON List Siren** (related URIs: **/ISAPI/SecurityCP/Configuration/wirelessSiren/capabilities?format=json** and **/ISAPI/SecurityCP/Configuration/wirelessSiren?format=json** ; related API: **NET_DVR_STDXMLConfig**): added a node **accessModuleType** (access module type).
6. Extended the message about the siren status **JSON SirenList** (related URI: **/ISAPI/SecurityCP/status/sirenStatus?format=json** ; related API: **NET_DVR_STDXMLConfig**): added 2 nodes: **accessModuleType** (access module type) and **address** (wired access module address).
7. Extended the message about the advanced configuration capability of the phone notification **JSON PhoneAnvancedCap** , message about the parameters of a specific phone number **JSON PhoneAnvanced** , and message about the parameters of all phone numbers **JSON List PhoneAnvanced** (related URIs: **/ISAPI/SecurityCP/Configuration/messageSendPhoneAnvanced/capabilities?format=json** , **/ISAPI/SecurityCP/Configuration/messageSendPhoneAnvanced/<ID>?format=json** , and **/ISAPI/SecurityCP/Configuration/messageSendPhoneAnvanced?format=json** ; related API: **NET_DVR_STDXMLConfig**): added a sub node **WeekPlanCfg** (week schedule information) to the node **Message** and **Call**.
8. Extended the message about the capability of controlling card addition mode in asynchronous mode **JSON Cap CardMode** and message about parameters of controlling card addition mode in asynchronous mode **JSON CardMode** (related URIs: **/ISAPI/SecurityCP/Configuration/card/mode/capabilities?format=json** and **/ISAPI/SecurityCP/Configuration/card/mode?format=json** ; related API: **NET_DVR_STDXMLConfig**): added a node **wirelessRecvAddress** (address of the wireless receiving module).
9. Extended the message about the configuration capability of logical relays **JSON OutputCap** and message about the parameters of the specified logical relays **JSON OutputSearch Config** (related URIs: **/ISAPI/SecurityCP/Configuration/outputs/capabilities?format=json** and **/ISAPI/SecurityCP/Configuration/outputs?format=json** ; related API: **NET_DVR_STDXMLConfig**): added 5 nodes: **notRelatedOutputNo** (No. of unlinked relay), **modifiedOutputNo** (relay No. that has been modified), **accessModuleType** (access module type), **relatedAccessModuleID** (No. of the linked access module), and **relayAttrib** (relay attribute).
10. Extended the message about the parameters of a logical relay **JSON Output** and the message about the parameters of all logical relays **JSON List Output** (related URIs: **/ISAPI/SecurityCP/Configuration/outputs/<ID>?format=json** and **/ISAPI/SecurityCP/Configuration/outputs?format=json** ; related API: **NET_DVR_STDXMLConfig**): added 2 nodes: **modifiedOutputNo** (relay No. that has been modified) and **accessModuleType** (access module type).
11. Extended the message about the configuration capability of security control system **JSON ManageCap** and message about the configuration parameters of security control system **JSON Manage** (related URIs: **/ISAPI/SecurityCP/Configuration/systemManage/capabilities?**

format=json and **/ISAPI/SecurityCP/Configuration/systemManage?format=json** ; related API: **NET_DVR_STDXMLConfig**):

added 4 nodes: **jammingSensitivity** (jamming sensitivity), **UKLocalCertificationEnabled** (whether to enable UK local certification), **ATPFaultSendDelayTime** (the delay time of reporting ATP malfunction to ARC), and **faultIndicatorEnabled** (whether to enable malfunction indicator light).

12. Extended the message about the configuration capability of the panic button **JSON_PanicButtonCap** , message about panic button parameters of a specific zone **JSON_PanicButton** , and message about the parameters of all panic buttons **JSON_List_PanicButton** (related URIs: **/ISAPI/SecurityCP/Configuration/panicButton/capabilities?format=json** , **/ISAPI/SecurityCP/Configuration/panicButton/zone/<ID>?format=json** , and **/ISAPI/SecurityCP/Configuration/panicButton?format=json** ; related API: **NET_DVR_STDXMLConfig**):
added 2 nodes: **triggerMode** (trigger mode of the panic button) and **confirmAlarmInterval** (time interval for uploading acknowledgment alarm).
13. Extended the message about configuration capability of alarm receiving center **JSON_ARCCap** , message about parameters of an alarm receiving center **JSON_ARC** , and message about parameters of all alarm receiving centers **JSON_List_ARC** (related URIs: **/ISAPI/SecurityCP/Configuration/ARC/capabilities?format=json** , **/ISAPI/SecurityCP/Configuration/ARC/<ID>?format=json** , and **/ISAPI/SecurityCP/Configuration/ARC?format=json** ; related API: **NET_DVR_STDXMLConfig**):
added 3 nodes: **transMethod** (transmission mode), **SpareARCList** (spare ARC list), and **FSKCfg** (FSK configuration).
14. Extended the message about the zone configuration capability **JSON_ZonesCap** (related URI: **/ISAPI/SecurityCP/Configuration/zones/capabilities?format=json** ; related API: **NET_DVR_STDXMLConfig**):
added 4 nodes: **detectorContactModeList** (list of detector contact modes), **alarmResistance** (alarm resistance), **tamperResistance** (tamper resistance), and **accessModuleType** (access module type).
15. Extended the message about parameters of a specific zone **JSON_Zone** and message about all zones' parameters **JSON_List_Zone** (related URIs: **/ISAPI/SecurityCP/Configuration/zones/<ID>?format=json** , **/ISAPI/SecurityCP/Configuration/zones?format=json** , and **/ISAPI/SecurityCP/Configuration/zones/currentAddAsyn?format=json** ; related API: **NET_DVR_STDXMLConfig**):
added 3 nodes: **alarmResistance** (alarm resistance), **tamperResistance** (tamper resistance), and **accessModuleType** (access module type).
16. Added a URI for getting the capability of wired detectors according to the wired detector type: POST **/ISAPI/SecurityCP/Configuration/wiredDetector/capabilities?format=json** (related API: **NET_DVR_STDXMLConfig**).
17. Added URIs for ARC manual test (related API: **NET_DVR_STDXMLConfig**):
Get the capability: GET **/ISAPI/SecurityCP/Configuration/ARC/manualTest/capabilities?format=json** ;
Configure parameters: PUT **/ISAPI/SecurityCP/Configuration/ARC/manualTest?format=json** ;

Get the manual test status of a single ARC: POST [/ISAPI/SecurityCP/Configuration/ARC/manualTest/status?format=json](#) .

18. Added URIs for adding zones, relays, or sirens to the access module (related API: [NET_DVR_STDXMLConfig](#)):
Get the capability according to the access module type: POST [/ISAPI/SecurityCP/Configuration/accessModule/addType/capabilities?format=json](#) ;
Set parameters: PUT [/ISAPI/SecurityCP/Configuration/accessModule/addType?format=json](#) .
19. Extended the message about configuration capability of security control panel [JSON_HostConfigCap](#) (related URI: [/ISAPI/SecurityCP/Configuration/capabilities?format=json](#) ; related API: [NET_DVR_STDXMLConfig](#)):
added one node **isSptARCMANualTest** (whether it supports ARC manual test).
20. Extended the message about the capability of security control panel [JSON_SecurityCPCap](#) (related URI: [/ISAPI/SecurityCP/capabilities?format=json](#) ; related API: [NET_DVR_STDXMLConfig](#)):
added one node **localAccessModuleType** (local access module type).

Summary of Changes in Version 6.1.6.10_Nov., 2020

1. Extended the siren configuration capability message [JSON_SirenCap](#) (related URI: [/ISAPI/SecurityCP/Configuration/wirelessSiren/capabilities?format=json](#) ; related API: [NET_DVR_STDXMLConfig](#)):
added 2 nodes "company" (company name) and "supportSirenCtrlIDList" (ID list of the sirens that support test).
2. Extended message about parameters for a siren [JSON_Siren](#) and message about all sirens' parameters [JSON_List_Siren](#) (related URIs: [/ISAPI/SecurityCP/Configuration/wirelessSiren/<ID>?format=json](#) and [/ISAPI/SecurityCP/Configuration/wirelessSiren?format=json](#) ; related API: [NET_DVR_STDXMLConfig](#)):
added one node "company" (company name).
3. Added URIs for relay linkage configuration (when the relay is closed/open) (related API: [NET_DVR_STDXMLConfig](#)):
Get capability: GET [/ISAPI/SecurityCP/Configuration/outputsModule/capabilities?format=json](#) ;
Get linkage configuration parameters of all relays or search for linkage configuration parameters by condition: GET or POST [/ISAPI/SecurityCP/Configuration/outputsModule?format=json](#) ;
Set relay's linkage configuration parameters: PUT [/ISAPI/SecurityCP/Configuration/outputsModule/<ID>?format=json](#) .
4. Extended the message about the parameters of the currently added keyfob in asynchronous mode [JSON_RemoteCtrl](#) (related URI: [/ISAPI/SecurityCP/Configuration/remoteCtrl/currentAddAsyn?format=json](#) ; related API: [NET_DVR_STDXMLConfig](#)):
added 2 nodes "failedReason" (reason for failure) and "relatedNetUserName" (linked network user name).
5. Extended message about the parameters of the currently added card in asynchronous mode [JSON_Card](#) (related URI: [/ISAPI/SecurityCP/Configuration/card/currentAddAsyn?format=json](#) ; related API: [NET_DVR_STDXMLConfig](#)):

- added 2 nodes **"failedReason"** (reason for failure) and **"relatedNetUserName"** (linked network user name).
6. Extended message about keypad configuration capability **JSON_KeypadCap** (related URI: **/ISAPI/SecurityCP/Configuration/keypad/capabilities?format=json** ; related API: **NET_DVR_STDXMLConfig**):
added 3 nodes **"attribute"** (keypad type), **"company"** (company name), and **"phoneNo"** (phone number).
 7. Extended message about configuration parameters of all keypads **JSON_List_Keypad** and message about configuration parameters of one keypad **JSON_Keypad** (related URIs: **/ISAPI/SecurityCP/Configuration/keypad?format=json** and **/ISAPI/SecurityCP/Configuration/keypad/<ID>?format=json** ; related API: **NET_DVR_STDXMLConfig**):
added 3 nodes **"attribute"** (keypad type), **"company"** (company name), and **"phoneNo"** (phone number).
 8. Extended the message about configuration capability of alarm receiving center **JSON_ARCCap** (related URI: **/ISAPI/SecurityCP/Configuration/ARC/capabilities?format=json** ; related API: **NET_DVR_STDXMLConfig**):
added 3 nodes **"authEnabled"** (whether to enable authentication), **"userName"** (user name), and **"password"** (password).
 9. Extended the message about parameters of all alarm receiving centers **JSON_List_ARC** and the message about parameters of an alarm receiving center **JSON_ARC** (related URIs: **/ISAPI/SecurityCP/Configuration/ARC?format=json** and **/ISAPI/SecurityCP/Configuration/ARC/<ID>?format=json** ; related API: **NET_DVR_STDXMLConfig**):
added 3 nodes **"authEnabled"** (whether to enable authentication), **"userName"** (user name), and **"password"** (password).
 10. Extended message about configuration capability of phone notification via PSTN **JSON_PSTNCfgCap** (related URI: **/ISAPI/SecurityCP/Configuration/PSTNCfg/capabilities?format=json** ; related API: **NET_DVR_STDXMLConfig**):
added one node **"reportPeriodEnabled"** (whether to enable test report uploading period).
 11. Extended the message about parameters of all phone notifications via PSTN **JSON_List_PSTNCfg** and message about parameters of a specific phone notification **JSON_PSTNCfg** (related URIs: **/ISAPI/SecurityCP/Configuration/PSTNCfg?format=json** and **/ISAPI/SecurityCP/Configuration/PSTNCfg/<ID>?format=json** ; related API: **NET_DVR_STDXMLConfig**):
added one node **"reportPeriodEnabled"** (whether to enable test report uploading period).
 12. Extended message about status of all partitions **JSON_SubSysList** (related URI: **/ISAPI/SecurityCP/status/subSystems?format=json** ; related API: **NET_DVR_STDXMLConfig**):
added one node **"name"** (partition name).

Summary of Changes in Version 6.1.5.25_Nov., 2020

1. Extended the message about the capability of security control panel **JSON_SecurityCPCap** (related URI: **/ISAPI/SecurityCP/capabilities?format=json** ; related API: **NET_DVR_STDXMLConfig**):

added one node "isSptPircamFileExport" (whether it supports exporting pictures captured by pircam).

2. Added URIs for exporting the picture captured by pircam (detector equipped with camera) (related API: **NET_DVR_STDXMLConfig**):

Get capability: GET **/ISAPI/SecurityCP/FileExport/pircam/capabilities?format=json** ;

Export the picture captured by pircam: POST **/ISAPI/SecurityCP/FileExport/pircam?format=json** .

Summary of Changes in Version 6.1.5.35_Nov., 2020

1. Extended ARC configuration capability message **JSON_ARCCap** , parameter message of an ARC **JSON_ARC** , and parameter message of all ARCs **JSON_List_ARC** (related URIs: **/ISAPI/SecurityCP/Configuration/ARC/capabilities?format=json** , **/ISAPI/SecurityCP/Configuration/ARC?format=json** , and **/ISAPI/SecurityCP/Configuration/ARC/<ID>?format=json** ; related API: **NET_DVR_STDXMLConfig**):

added a node **timeStampGMTEnabled** (whether to enable GMT time stamp).

2. Extended log types in **Log Types for ISAPI** :

added an operation log "cardNoNotRegistered"(Card No. not registered).

Summary of Changes in Version 6.1.5.5_July, 2020

1. Extended zone configuration capability message **JSON_ZonesCap** (related URI: **/ISAPI/SecurityCP/Configuration/zones/capabilities?format=json** ; related API: **NET_DVR_STDXMLConfig**):

added a detector type "pircam" (pircam detector) to the node **detectorType**;

added two sub nodes **linkagePircamCapCfg** (whether it supports configuring pircam capture linkage) and **linkageFileName** (name length of the linked file) to the node **ZonesCap** and **WiredZonesCap**, respectively.

2. Extended parameter message of all zones **JSON_List_Zone** and parameter message of a specific zone **JSON_Zone** (related URIs: **/ISAPI/SecurityCP/Configuration/zones?format=json** and **/ISAPI/SecurityCP/Configuration/zones/<ID>?format=json** ; related API: **NET_DVR_STDXMLConfig**):

added two nodes **linkagePircamCapCfg** (whether it supports configuring pircam capture linkage) and **linkageFileName** (name of the linked file).

3. Extended siren configuration capability message **JSON_SirenCap** , parameter message of a specific siren **JSON_Siren** , and parameter message of all sirens **JSON_List_Siren** (related URIs: **/ISAPI/SecurityCP/Configuration/wirelessSiren/capabilities?format=json** , **/ISAPI/SecurityCP/Configuration/wirelessSiren/<ID>?format=json** , and **/ISAPI/SecurityCP/Configuration/wirelessSiren?format=json** ; related API: **NET_DVR_STDXMLConfig**):

added 12 nodes: **LEDEnabled** (whether to enable the LED indicator), **LEDLatchTime** (delay time of the LED indicator), **findMeEnabled** (whether to enable the Find Me function), **location** (siren location), **ArmAndDisarmIndicatorCfg** (indicator settings for arming and disarming), **tamperEnabled** (whether to enable siren tampering), **tryAlarmEnabled** (whether to enable alarm attempt), **preRegisterEnabled** (whether to enable pre-registration), **buzzEnabled** (whether to enable linking the buzzer to buzz when the alarm is triggered),

- alarmStrobeFlashEnabled** (whether to enable linking the alarm lamp to flicker when the alarm is triggered), **sounderAlarmDuration** (siren's output duration when the alarm is triggered), and **heartBeatInterval** (heartbeat interval of the security control panel and the peripheral).
4. Added two URIs of controlling the asynchronous mode of adding the siren (related API: **NET_DVR_STDXMLConfig**):
Get the capability: GET [/ISAPI/SecurityCP/Configuration/wirelessSiren/mode/capabilities?format=json](#) ;
Control the mode: PUT [/ISAPI/SecurityCP/Configuration/wirelessSiren/mode?format=json](#) .
 5. Added a URI of getting parameters of the currently added siren in asynchronous mode (related API: **NET_DVR_STDXMLConfig**): GET [/ISAPI/SecurityCP/Configuration/wirelessSiren/currentAddAsyn?format=json](#) .
 6. Extended card reader configuration capability message **JSON_CardReaderCap** , parameter message of a specific card reader **JSON_CardReader** , and parameter message of all card readers **JSON_List_CardReader** (related URIs: [/ISAPI/SecurityCP/Configuration/cardReader/capabilities?format=json](#) , [/ISAPI/SecurityCP/Configuration/cardReader/<ID>?format=json](#) , and [/ISAPI/SecurityCP/Configuration/cardReader?format=json](#) ; related API: **NET_DVR_STDXMLConfig**):
added three nodes: **enabled** (whether to enable the card reader), **LEDEnabled** (whether to enable the LED indicator), and **heartBeatInterval** (heartbeat interval of the security control panel and the peripheral).
 7. Added two URIs of controlling the asynchronous mode of adding card reader parameters (related API: **NET_DVR_STDXMLConfig**):
Get the capability: GET [/ISAPI/SecurityCP/Configuration/cardReader/mode/capabilities?format=json](#) ;
Control the mode: PUT [/ISAPI/SecurityCP/Configuration/cardReader/mode?format=json](#) .
 8. Added a URI of getting the parameters of the currently added card reader in asynchronous mode (related API: **NET_DVR_STDXMLConfig**): GET [/ISAPI/SecurityCP/Configuration/cardReader/currentAddAsyn?format=json](#) .
 9. Added three URIs of configuring pircam parameters (related API: **NET_DVR_STDXMLConfig**):
Get configuration capability: GET [/ISAPI/SecurityCP/Configuration/pircam/capabilities?format=json](#) ;
Get parameters of all zones: GET [/ISAPI/SecurityCP/Configuration/pircam/zone?format=json](#) ;
Get or set parameters of a specific zone: GET or PUT [/ISAPI/SecurityCP/Configuration/pircam/zone/<ID>?format=json](#) .
 10. Added four URIs of controlling pircam (detector equipped with camera) capture (related API: **NET_DVR_STDXMLConfig**):
Get the picture captured by the pircam in synchronous mode: GET [/ISAPI/SecurityCP/pircam/picture/channels/<ID>?format=json](#) ;
Get the capability of controlling the pircam to capture pictures or record videos in asynchronous mode: GET [/ISAPI/SecurityCP/pircam/picture/mode/capabilities?format=json](#) ;
Control the pircam to capture pictures in asynchronous mode: PUT [/ISAPI/SecurityCP/pircam/picture/channels/<ID>/mode?format=json](#) ;

Get pircam capture parameters being added currently in asynchronous mode: GET [/ISAPI/SecurityCP/pircam/picture/channels/<ID>/currentAddAsyn?format=json](#) .

11. Extended message about alarm or event details of the security control panel [JSON_EventNotificationAlert_SecurityCPAlarmEventMsg](#) :
added a node **imageURL** (picture URL);
added two sub nodes **remoteCtrlNo** (keyfob No.) and **userName** (user name) to the node **CIDEvent**.
12. Extended advanced configuration capability message of phone notification [JSON_PhoneAnvancedCap](#) , advanced notification parameter message of all phone numbers [JSON_List_PhoneAnvanced](#) , and advanced notification parameter message of a specific phone number [JSON_PhoneAnvanced](#) (related URIs: [/ISAPI/SecurityCP/Configuration/messageSendPhoneAnvanced/capabilities?format=json](#) , [/ISAPI/SecurityCP/Configuration/messageSendPhoneAnvanced?format=json](#) , and [/ISAPI/SecurityCP/Configuration/messageSendPhoneAnvanced/<ID>?format=json](#) ; related API: [NET_DVR_STDXMLConfig](#)):
added a sub node **intelligentAlarmEnable** (whether to enable smart alarm notification) to the node **Message** and **Call**, respectively.
13. Extended capability message of getting security control panels' status [JSON_HostStatusCap](#) (related URI: [/ISAPI/SecurityCP/status/capabilities?format=json](#) ; related API: [NET_DVR_STDXMLConfig](#)):
added two nodes **isSptHostItself** (whether it supports getting status of the security control panel itself) and **isSptACPower** (whether it supports getting AC power supply status).
14. Added a URI of getting the status of the security control panel itself (related API: [NET_DVR_STDXMLConfig](#)): GET [/ISAPI/SecurityCP/status/hostItself?format=json](#) .
15. Extended message of zone status list [JSON_ZoneList](#) (related URI: [/ISAPI/SecurityCP/status/zones?format=json](#) ; related API: [NET_DVR_STDXMLConfig](#)):
added six sub nodes to the node **Zone**, i.e., **chargeValue** (battery power value), **temperature** (temperature), **detectorType** (type of the detector linked to the zone), **model** (model), **stayAway** (whether the zone is stay armed), and **zoneType** (zone type).
16. Extended message of siren status list [JSON_SirenList](#) (related URI: [/ISAPI/SecurityCP/status/sirenStatus?format=json](#) ; related API: [NET_DVR_STDXMLConfig](#)):
added three sub nodes to the node **Siren**, i.e., **chargeValue** (battery power value), **temperature** (temperature), and **model** (model).
17. Extended message of card reader status list [JSON_CardReaderList](#) (related URI: [/ISAPI/SecurityCP/status/cardReaderStatus?format=json](#) ; related API: [NET_DVR_STDXMLConfig](#)):
added three sub nodes to the node **Siren**, i.e., **chargeValue** (battery power value), **temperature** (temperature), and **model** (model).
18. Added a URI of getting the AC power supply status (related API: [NET_DVR_STDXMLConfig](#)): GET [/ISAPI/SecurityCP/status/acPowerStatus?format=json](#) .
19. Extended message about all status of the security control panel [JSON_AlarmHostStatus](#) (related URI: [/ISAPI/SecurityCP/status/host?format=json](#) ; related API: [NET_DVR_STDXMLConfig](#)):
added two nodes **HostStatus** (status information of the security control panel) and **ACPowerStatus** (status information of the AC power supply).

20. Added URIs of configuring voice prompt parameters (related API: **NET_DVR_STDXMLConfig**):
Get configuration capability: GET **/ISAPI/SecurityCP/voicePrompt/capabilities?format=json** ;
Get or set parameters: GET or PUT **/ISAPI/SecurityCP/voicePrompt?format=json** .
21. Added URIs of configuring device environment parameters (related API: **NET_DVR_STDXMLConfig**):
Get configuration capability: GET **/ISAPI/SecurityCP/surroundEnvironmentCfg/capabilities?format=json** ;
Get or set parameters: GET or PUT **/ISAPI/SecurityCP/surroundEnvironmentCfg?format=json** .
22. Added a URI of uploading the audio file of custom voice prompt (related API: **NET_DVR_STDXMLConfig**): POST **/ISAPI/SecurityCP/videoBroadcast/customizeUpload?format=json** .
23. Extended capability message of the security control panel **JSON_SecurityCPCap** (related URI: **/ISAPI/SecurityCP/capabilities?format=json** ; related API: **NET_DVR_STDXMLConfig**):
added seven nodes: **isSptSysCheckManually** (whether it supports enabling system detection manually), **isSptSysCheckResult** (whether it supports getting the system detection result), **isSptSysAutoCheckTimeCfg** (whether it supports audio and video system detection), **isSptVideoFileUpload** (whether it supports uploading the audio file), **isSptPircamCapture** (whether it supports pircam capture), **isSptCustomizeCfg** (whether it supports configuring the custom audio file), and **isSptCustomizeUpload** (whether it supports uploading the custom file).
24. Extended configuration capability message of security control panel **JSON_HostConfigCap** (related URI: **/ISAPI/SecurityCP/Configuration/capabilities?format=json** ; related API: **NET_DVR_STDXMLConfig**):
added four sub nodes to the node **ExDevice**, i.e., **isSptPircam** (whether it supports configuring pircam parameters), **isSptMuteVoicePlanCFG** (whether it supports configuring muting schedule), **isSptVoicePromptCfg** (whether it supports configuring voice prompt parameters), and **isSptSurroundEnvironmentCfg** (whether it supports configuring device environment parameters);
added a node **isSptSignalStrengthDetection** (whether it supports signal strength detection).
25. Added 8 sub status codes to status code 4 (Invalid Operation) in **Response Codes of Text Protocol** :
0x40008038-"overAudioFileNumLimit" (The number of audio files exceeds the limit),
0x40008039-"audioFileNamelsLong" (The audio file name is too long),
0x4000803a-"audioFormatIsWrong" (The audio file format is invalid),
0x4000803b-"audioFileIsLarge" (The size of the audio file exceeds the limit),
0x4000803c-"pircamCapTimeOut" (Capturing of pircam timed out),
0x4000803d-"pircamCapFail" (Capturing of pircam failed), 0x4000803e-"pircamIsCaping" (The pircam is capturing), and 0x4000803f-"audioFileHasExisted" (The audio file already exists).
26. Extended the log types **Log Types for ISAPI** :
added 10 alarm log types: "soundIntensityMutation" (Sudden Increase of Sound Intensity Detection), "soundIntensityMutationStop" (Sudden Increase of Sound Intensity Detection Ended), "soundIntensitySteepFall" (Sudden Decrease of Sound Intensity Detection), "soundIntensitySteepFallStop" (Sudden Decrease of Sound Intensity Detection Ended), "moveAlarm" (Motion Alarm), "moveAlarmRestored" (Motion Alarm Restored),

"lowTemperatureAlarm" (Low Temperature Alarm), "lowTemperatureAlarmRestored" (Low Temperature Alarm Restored), "highTemperatureAlarm" (High Temperature Alarm), and "highTemperatureAlarmRestored" (High Temperature Alarm Restored);
added 20 operation log types: "SSHEnabled" (SSH Enabled), "SSHDisabled" (SSH Disabled), "installationModeEntered" (Installation Mode Enabled), "installationModeExited" (Installation Mode Disabled), "diagnosisModeConfigured" (Diagnosis Mode Configured), "fileExported" (File Exported), "audioFileUploaded" (Audio File Uploaded), "audioFileDeleted" (Audio File Deleted), "PIRCAMCapture" (PIRCAM Captured), "SIPIntercomStarted" (SIP Intercom Started), "SIPIntercomEnded" (SIP Intercom Ended), "enrollmentModeEntered" (Registration Mode Enabled), "enrollmentModeExited" (Registration Mode Disabled), "videoAudioSelfCheckStarted" (Self-Test of Audio and Video Started), "videoAudioSelfCheckStopped" (Self-Test of Audio and Video Stopped), "cardReaderunlocked" (Card Reader Unlocked), "cardReaderlocked" (Card Reader Locked), "videoAudioSelfCheckEnded" (Self-Test of Audio and Video Ended), "previewStart" (Live View Started), and "previewStop" (Live View Stopped).

Summary of Changes in Version 6.1.4.45_June, 2020

1. Extended zone configuration capability message **JSON_ZonesCap** (related URI: **/ISAPI/SecurityCP/Configuration/zones/capabilities?format=json** ; related API: **NET_DVR_STDXMLConfig**):
added a zone property "newKeyZoneTriggerTypeCfg" (trigger type settings of the key zone (new version)) to the sub node **sptProp** of the node **ZoneType** of **ZoneTypeList** of **ZonesCap**;
added two sub nodes **newKeyZoneTriggerTypeCfg** (trigger type settings of the key zone (new version)) and **zoneStatusCfg** (zone status settings) to the node **ZonesCap**.
2. Extended parameter message of all zones **JSON_List_Zone** and parameter message of a specific zone **JSON_Zone** (related URIs: **/ISAPI/SecurityCP/Configuration/zones?format=json** and **/ISAPI/SecurityCP/Configuration/zones/<ID>?format=json** ; related API: **NET_DVR_STDXMLConfig**):
added two nodes **newKeyZoneTriggerTypeCfg** (trigger type settings of the key zone (new version)) and **zoneStatusCfg** (zone status settings).
3. Extended siren configuration capability message **JSON_SirenCap** (related URI: **/ISAPI/SecurityCP/Configuration/wirelessSiren/capabilities?format=json** ; related API: **NET_DVR_STDXMLConfig**):
added a node **supportSirenCtrlIDList** (ID list of sirens that support siren test alarm).
4. Extended configuration capability message of the logical relay **JSON_OutputCap** , parameter message of all logical relays **JSON_List_Output** , result message of getting logical relays' parameters by specific conditions **JSON_OutputSearch_Config** , and parameter message of a logical relay **JSON_Output** (related URIs: **/ISAPI/SecurityCP/Configuration/outputs/capabilities?format=json** , **/ISAPI/SecurityCP/Configuration/outputs?format=json** , and **/ISAPI/SecurityCP/Configuration/outputs/<ID>?format=json** ; related API: **NET_DVR_STDXMLConfig**):
added a node **LinkageList** (linked event list).
5. Extended keyfob configuration capability message **JSON_RemoteCtrlCap** , parameter message of all keyfobs **JSON_List_RemoteCtrl** , and parameter message of a keyfob **JSON_RemoteCtrl**

(related URIs: [/ISAPI/SecurityCP/Configuration/remoteCtrl/capabilities?format=json](#) , [/ISAPI/SecurityCP/Configuration/remoteCtrl?format=json](#) , [/ISAPI/SecurityCP/Configuration/remoteCtrl/<ID>?format=json](#) , [/ISAPI/SecurityCP/Configuration/remoteCtrl/currentAdd?format=json](#) , and [/ISAPI/SecurityCP/Configuration/remoteCtrl/currentAddAsyn?format=json](#) ; related API: **NET DVR STDXMLConfig**):

added a node **alarmVoicePromptEnabled** (whether to enable voice prompt for panic alarm).

6. Extended the configuration capability message of alarm receiving center **JSON_ARCCap** , parameter message of all alarm receiving centers **JSON_List_ARC** , and parameter message of a specific alarm receiving center **JSON_ARC** (related URIs: [/ISAPI/SecurityCP/Configuration/ARC/capabilities?format=json](#) , [/ISAPI/SecurityCP/Configuration/ARC?format=json](#) , and [/ISAPI/SecurityCP/Configuration/ARC/<ID>?format=json](#) ; related API: **NET DVR STDXMLConfig**):
added two nodes **periodicTestEnabled** (whether to enable periodic test) and **periodicTestTimeCfg** (periodic test interval).
7. Extended capability message **JSON_ManageCap** and parameter message **JSON_Manage** of managing and configuring the security control system (related URIs: [/ISAPI/SecurityCP/Configuration/systemManage/capabilities?format=json](#) and [/ISAPI/SecurityCP/Configuration/systemManage?format=json](#) ; related API: **NET DVR STDXMLConfig**):
added two nodes **tamperLinkageAlarmEnabled** (whether to enable tampering alarm linkage) and **oneKeyLockEnabled** (whether to enable one-push locking panel).
8. Extended capability message of the security control panel **JSON_SecurityCPCap** (related URI: [/ISAPI/SecurityCP/capabilities?format=json](#) ; related API: **NET DVR STDXMLConfig**):
added a node **isSptOneKeyAlarm** (whether it supports one-push alarm).

Summary of Changes in Version 6.1.4.15_June, 2020

1. Extended siren configuration capability message **JSON_SirenCap** (related URI: [/ISAPI/SecurityCP/Configuration/wirelessSiren/capabilities?format=json](#) ; related API: **NET DVR STDXMLConfig**):
added four values to the node **volume**: 0 (muted), 1 (low), 2 (medium), 3 (high);
added a node **LinkageList** (linked event list).
2. Extended parameter message of a siren **JSON_Siren** and parameter message of all sirens **JSON_List_Siren** (related URIs: [/ISAPI/SecurityCP/Configuration/wirelessSiren/<ID>?format=json](#) and [/ISAPI/SecurityCP/Configuration/wirelessSiren?format=json](#) ; related API: **NET DVR STDXMLConfig**):
added a node **LinkageList** (linked event list).
3. Extended keypad configuration capability message **JSON_KeypadCap** , parameter message of a keypad **JSON_Keypad** , and parameter message of all keypads **JSON_List_Keypad** (related URIs: [/ISAPI/SecurityCP/Configuration/keypad/capabilities?format=json](#) , [/ISAPI/SecurityCP/Configuration/keypad/<ID>?format=json](#) , and [/ISAPI/SecurityCP/Configuration/keypad?format=json](#) ; related API: **NET DVR STDXMLConfig**):
added a node **alarmVoicePromptEnabled** (whether to enable voice prompt for panic alarm).
4. Extended ARC configuration capability message **JSON_ARCCap** , parameter message of an ARC **JSON_ARC** , and parameter message of all ARCs **JSON_List_ARC** (related URIs: [/ISAPI/SecurityCP/Configuration/ARC/capabilities?format=json](#) , [/ISAPI/SecurityCP/](#)

[Configuration/ARC/<ID>?format=json](#) , and [/ISAPI/SecurityCP/Configuration/ARC?format=json](#) ; related API: [NET_DVR_STDXMLConfig](#)):

added three nodes **ARCchannelList** (ARC channel list), **subSystem** (linked partition), and **sysEvent** (system event).

Summary of Changes in Version 6.1.4.15_Apr., 2020

1. Extended SIP parameter structure [NET_DVR_SIP_CFG_V50](#) (related APIs: [NET_DVR_GetDVRConfig](#) with command 16044-"NET_DVR_GET_SIP_CFG_V50" and [NET_DVR_SetDVRConfig](#) with command 16045-"NET_DVR_SET_SIP_CFG_V50"): added a member **byCalledTargetName** (user name of the called person) by 32 bytes.
2. Extended capability message of video intercom devices [XML_IpViewDevAbility](#) (related API: [NET_DVR_GetDeviceAbility](#) ; capability type: 0x014-"IP_VIEW_DEV_ABILITY"): added two sub nodes to the node **<SipServerLogin>**, i.e., **<localPort>** (local port No.) and **<CalledTargetName>** (user name length of the called person).
3. Extended capability message of front-end devices [XML_CAMERAPARA](#) (related API: [NET_DVR_GetDeviceAbility](#) ; capability type: 0x009-"IPC_FRONT_PARAMETER_V20"): added a sub node **<IPStartChanNoDefault>** (start digital channel No.) to the node **<ChannelEntry>** of **<ChannelList>**; added a sub node **<isNotSupportDigitalChanCfg>** (whether setting digital channels is not supported) to the node **<PowerLineFrequencyMode>**, **<WDR>**, and **<LightInhibit>** of **ChannelEntry** of **<ChannelList>**, respectively.

Summary of Changes in Version 6.1.3.5_Dec., 2019

1. Extended zone configuration capability message [JSON_ZonesCap](#) (related URI: [/ISAPI/SecurityCP/Configuration/zones/capabilities?format=json](#) ; related API: [NET_DVR_STDXMLConfig](#)): added four sub nodes: **extendZoneNo** (extended zone No. after enabling double-detector zone), **relatedZoneNo** (linked zone No.), **SupportedDoubleZoneNos** (No. list of zones that support double-detector zone settings), and **doubleZoneCfgEnable** (whether to enable double-detector zone) to the node **WiredZonesCap** (wired zone capability).
2. Extended parameter message of all zones [JSON_List_Zone](#) and parameter message of a specific zone [JSON_Zone](#) (related URIs: [/ISAPI/SecurityCP/Configuration/zones?format=json](#) and [/ISAPI/SecurityCP/Configuration/zones/<ID>?format=json](#) ; related API: [NET_DVR_STDXMLConfig](#)): added three nodes: **extendZoneNo** (extended zone No. after enabling double-detector zone), **relatedZoneNo** (linked zone No.), and **doubleZoneCfgEnable** (whether to enable double-detector zone).
3. Extended partition configuration capability message [JSON_SubSysCap](#) , parameter message of a specific partition [JSON_SubSys](#) , and parameter message of all partitions [JSON_List_SubSys](#) (related URIs: [/ISAPI/SecurityCP/Configuration/subSys/capabilities?format=json](#) , [/ISAPI/SecurityCP/Configuration/subSys/<ID>?format=json](#) , and [/ISAPI/SecurityCP/Configuration/subSys?format=json](#) ; related API: [NET_DVR_STDXMLConfig](#)): added a node **isPublicSubSys** (whether the partition is a public partition).

4. Extended siren configuration capability message ***JSON_SirenCap*** (related URI: ***/ISAPI/SecurityCP/Configuration/wirelessSiren/capabilities?format=json*** ; related API: ***NET_DVR_STDXMLConfig***):
added two nodes: **subSystem** (number of partitions that can be linked) and **subSystemNo** (range of partition No.).
5. Extended parameter message of a specific siren ***JSON_Siren*** and parameter message of all sirens ***JSON_List_Siren*** (related URIs: ***/ISAPI/SecurityCP/Configuration/wirelessSiren/<ID>?format=json*** and ***/ISAPI/SecurityCP/Configuration/wirelessSiren?format=json*** ; related API: ***NET_DVR_STDXMLConfig***):
added a node **subSystem** (partitions linked to the siren).
6. Extended configuration capability message ***JSON_DirectCap*** and parameter message ***JSON_Direct*** of alarm receiving center notification in arming mode (related URIs: ***/ISAPI/SecurityCP/Configuration/messageSendDirect/capabilities?format=json*** and ***/ISAPI/SecurityCP/Configuration/messageSendDirect?format=json*** ; related API: ***NET_DVR_STDXMLConfig***):
added a node **intelligentAlarmEnable** (whether to enable smart alarm notification).
7. Extended configuration capability message ***JSON_SendARCCap*** and parameter message ***JSON_SendARC*** of alarm receiving center notification in listening mode (related URIs: ***/ISAPI/SecurityCP/Configuration/messageSendARC/capabilities?format=json*** and ***/ISAPI/SecurityCP/Configuration/messageSendARC?format=json*** ; related API: ***NET_DVR_STDXMLConfig***):
added a node **intelligentAlarmEnable** (whether to enable smart alarm notification).
8. Extended notification parameter message of all alarm receiving centers ***JSON_SendARCList*** (related URI: ***/ISAPI/SecurityCP/Configuration/messageSendARCList?format=json*** ; related API: ***NET_DVR_STDXMLConfig***):
added a sub node **intelligentAlarmEnable** (whether to enable smart alarm notification) to the node **SendARC**.
9. Extended configuration capability message ***JSON_CloudCap*** and parameter message ***JSON_Cloud*** of Hik-Connect notification (related URIs: ***/ISAPI/SecurityCP/Configuration/messageSendCloud/capabilities?format=json*** and ***/ISAPI/SecurityCP/Configuration/messageSendCloud?format=json*** ; related API: ***NET_DVR_STDXMLConfig***):
added a node **intelligentAlarmEnable** (whether to enable smart alarm notification).
10. Extended advanced configuration capability message of phone notification ***JSON_PhoneAnvancedCap*** , advanced notification parameter message of all phone numbers ***JSON_List_PhoneAnvanced*** , and advanced notification parameter message of a specific phone number ***JSON_PhoneAnvanced*** (related URIs: ***/ISAPI/SecurityCP/Configuration/messageSendPhoneAnvanced/capabilities?format=json*** , ***/ISAPI/SecurityCP/Configuration/messageSendPhoneAnvanced?format=json*** , and ***/ISAPI/SecurityCP/Configuration/messageSendPhoneAnvanced/<ID>?format=json*** ; related API: ***NET_DVR_STDXMLConfig***):
added a sub node **intelligentAlarmEnable** (whether to enable smart alarm notification) to the node **Message** and **Call**, respectively.
11. Extended message about alarm or event details of the security control panel ***JSON_EventNotificationAlert_SecurityCPAlarmEventMsg*** :

added an event type "intelligentAlarmEvent" (smart alarm) to the sub node **type** of the node **CIDEvent**;
added a sub node **extensionModule** (extension module No.) to the node **CIDEvent**.

Summary of Changes in Version 6.1.3.5_Nov., 2019

1. Extended configuration capability message of security control panel **JSON HostConfigCap** (related URL: [/ISAPI/SecurityCP/Configuration/capabilities?format=json](#) ; related API: **NET_DVR_STDXMLConfig**):
added a node **isSptMobCalibration** (whether it supports map calibration for the radar).
2. Extended the log types in **Log Types for ISAPI** :
added four operation log types: "scaleCfg" (Scale Settings), "radarTrailCfg" (Radar Pattern Settings), "MapImportCfg" (Map Importing Settings), and "radarCalibrationCfg" (Radar Calibration Settings).

Summary of Changes in Version 6.1.0.20_Oct., 2019

1. Added URLs of getting user names of users that can remotely configure devices (related API: **NET_DVR_STDXMLConfig**):
Get capability: GET [/ISAPI/SecurityCP/Configuration/remoteCfgPermissonUserName/capabilities?format=json](#) ;
Get user names: GET [/ISAPI/SecurityCP/Configuration/remoteCfgPermissonUserName?format=json](#) .
2. Extended zone configuration capability message **JSON ZonesCap** (related URL: [/ISAPI/SecurityCP/Configuration/zones/capabilities?format=json](#) ; related API: **NET_DVR_STDXMLConfig**):
added a zone type "Key" (key zone) to the sub node **type** of **ZonesCap** and **WiredZonesCap**.
3. Extended configuration capability message of partition timer **JSON SubSysTimeCap** (related URL: [/ISAPI/SecurityCP/Configuration/subSysTime/capabilities?format=json](#) ; related API: **NET_DVR_STDXMLConfig**):
added two sub nodes: **WeekCfg** (week schedule configuration) and **HolidayExceptionsCfg** (holiday schedule configuration) to the node **SubSysTimeCap**.
4. Extended timer parameter message of all partitions **JSON List SubSysTime** (related URL: [/ISAPI/SecurityCP/Configuration/subSysTime?format=json](#) ; related API: **NET_DVR_STDXMLConfig**):
added two sub nodes: **WeekCfg** (week schedule configuration) and **HolidayExceptionsCfg** (holiday schedule configuration) to the node **SubSysTime** of **List**.
5. Extended timer parameter message of a specific partition **JSON SubSysTime** (related URL: [/ISAPI/SecurityCP/Configuration/subSysTime/<ID>?format=json](#) ; related API: **NET_DVR_STDXMLConfig**):
added two sub nodes: **WeekCfg** (week schedule configuration) and **HolidayExceptionsCfg** (holiday schedule configuration) to the node **SubSysTime**.
6. Extended configuration capability message of logical relay **JSON OutputCap** (related URL: [/ISAPI/SecurityCP/Configuration/outputs/capabilities?format=json](#) ; related API: **NET_DVR_STDXMLConfig**):

- added four sub nodes: **minorType** (minor event type), **subSystem** (number of linked partitions), **subSystemNo** (range of partition No.), and **durationConstOutputEnable** (whether it supports configuring relay output duration) to the node **OutputCap**.
7. Extended parameter message of all logical relays ***JSON_List_Output*** (related URL: ***/ISAPI/SecurityCP/Configuration/outputs?format=json*** ; related API: ***NET_DVR_STDXMLConfig***): added three sub nodes: **minorType** (minor event type), **subSystem** (linked partitions), and **durationConstOutputEnable** (whether it supports configuring relay output duration) to the node **Output** of **List**.
 8. Extended result message of getting logical relays' parameters by specific conditions ***JSON_OutputSearch_Config*** (related URL: ***/ISAPI/SecurityCP/Configuration/outputs?format=json*** ; related API: ***NET_DVR_STDXMLConfig***): added three sub nodes: **minorType** (minor event type), **subSystem** (linked partitions), and **durationConstOutputEnable** (whether it supports configuring relay output duration) to the node **Output** of **List** of **OutputSearch**.
 9. Extended parameter message of a specific logical relay ***JSON_Output*** (related URL: ***/ISAPI/SecurityCP/Configuration/outputs/<ID>?format=json*** ; related API: ***NET_DVR_STDXMLConfig***): added three sub nodes: **minorType** (minor event type), **subSystem** (linked partitions), and **durationConstOutputEnable** (whether it supports configuring relay output duration) to the node **Output**.
 10. Extended configuration capability message of alarm receiving center notification in arming mode ***JSON_DirectCap*** (related URL: ***/ISAPI/SecurityCP/Configuration/messageSendDirect/capabilities?format=json*** ; related API: ***NET_DVR_STDXMLConfig***): added 10 sub nodes: **zoneAlarmTamperEnabled** (whether to enable alarm and tampering event notification of the supported zone), **exDevTamperEventEnabled** (whether to enable peripheral tampering alarm notification), **hostTamperEventEnabled** (whether to enable tampering alarm notification of security control panel), **emergencyEventEnabled** (whether to enable panic alarm notification), **medicalEventEnabled** (whether to enable medical alarm notification), **gasEventEnabled** (whether to enable gas alarm notification), **fireEventEnabled** (whether to enable fire alarm notification), **hostStatusEventEnabled** (whether to enable notification of security control panel status), **exDevStatusEventEnabled** (whether to enable peripheral status notification), and **detectorStatusEventEnabled** (whether to enable detector status notification) to the node **DirectCap**.
 11. Extended parameter message of alarm receiving center notification in arming mode ***JSON_Direct*** (related URL: ***/ISAPI/SecurityCP/Configuration/messageSendDirect?format=json*** ; related API: ***NET_DVR_STDXMLConfig***): added 10 sub nodes: **zoneAlarmTamperEnabled** (whether to enable alarm and tampering event notification of the supported zone), **exDevTamperEventEnabled** (whether to enable peripheral tampering alarm notification), **hostTamperEventEnabled** (whether to enable tampering alarm notification of security control panel), **emergencyEventEnabled** (whether to enable panic alarm notification), **medicalEventEnabled** (whether to enable medical alarm notification), **gasEventEnabled** (whether to enable gas alarm notification), **fireEventEnabled** (whether to enable fire alarm notification), **hostStatusEventEnabled** (whether to enable notification of security control panel status), **exDevStatusEventEnabled** (whether to enable

- peripheral status notification), and **detectorStatusEventEnabled** (whether to enable detector status notification) to the node **Direct**.
12. Extended configuration capability of alarm receiving center notification in listening mode **JSON_SendARCCap** (related URL: **/ISAPI/SecurityCP/Configuration/messageSendARC/capabilities?format=json** ; related API: **NET DVR STDXMLConfig**):
added 10 sub nodes: **zoneAlarmTamperEnabled** (whether to enable alarm and tampering event notification of the supported zone), **exDevTamperEventEnabled** (whether to enable peripheral tampering alarm notification), **hostTamperEventEnabled** (whether to enable tampering alarm notification of security control panel), **emergencyEventEnabled** (whether to enable panic alarm notification), **medicalEventEnabled** (whether to enable medical alarm notification), **gasEventEnabled** (whether to enable gas alarm notification), **fireEventEnabled** (whether to enable fire alarm notification), **hostStatusEventEnabled** (whether to enable notification of security control panel status), **exDevStatusEventEnabled** (whether to enable peripheral status notification), and **detectorStatusEventEnabled** (whether to enable detector status notification) to the node **SendARCCap**.
 13. Extended notification parameter message of all alarm receiving centers in listening mode **JSON_SendARCList** (related URL: **/ISAPI/SecurityCP/Configuration/messageSendARCList?format=json** ; related API: **NET DVR STDXMLConfig**):
added 10 sub nodes: **zoneAlarmTamperEnabled** (whether to enable alarm and tampering event notification of the supported zone), **exDevTamperEventEnabled** (whether to enable peripheral tampering alarm notification), **hostTamperEventEnabled** (whether to enable tampering alarm notification of security control panel), **emergencyEventEnabled** (whether to enable panic alarm notification), **medicalEventEnabled** (whether to enable medical alarm notification), **gasEventEnabled** (whether to enable gas alarm notification), **fireEventEnabled** (whether to enable fire alarm notification), **hostStatusEventEnabled** (whether to enable notification of security control panel status), **exDevStatusEventEnabled** (whether to enable peripheral status notification), and **detectorStatusEventEnabled** (whether to enable detector status notification) to the node **SendARC** of **SendARCList**.
 14. Extended notification parameter message of a specific alarm receiving center in listening mode **JSON_SendARC** (related URL: **/ISAPI/SecurityCP/Configuration/messageSendARC?format=json** ; related API: **NET DVR STDXMLConfig**):
added 10 sub nodes: **zoneAlarmTamperEnabled** (whether to enable alarm and tampering event notification of the supported zone), **exDevTamperEventEnabled** (whether to enable peripheral tampering alarm notification), **hostTamperEventEnabled** (whether to enable tampering alarm notification of security control panel), **emergencyEventEnabled** (whether to enable panic alarm notification), **medicalEventEnabled** (whether to enable medical alarm notification), **gasEventEnabled** (whether to enable gas alarm notification), **fireEventEnabled** (whether to enable fire alarm notification), **hostStatusEventEnabled** (whether to enable notification of security control panel status), **exDevStatusEventEnabled** (whether to enable peripheral status notification), and **detectorStatusEventEnabled** (whether to enable detector status notification) to the node **SendARC**.

15. Extended configuration capability message of Hik-Connect notification ***JSON CloudCap*** (related URL: </ISAPI/SecurityCP/Configuration/messageSendCloud/capabilities?format=json> ; related API: ***NET DVR STDXMLConfig***):
added 10 sub nodes: **zoneAlarmTamperEnabled** (whether to enable alarm and tampering event notification of the supported zone), **exDevTamperEventEnabled** (whether to enable peripheral tampering alarm notification), **hostTamperEventEnabled** (whether to enable tampering alarm notification of security control panel), **emergencyEventEnabled** (whether to enable panic alarm notification), **medicalEventEnabled** (whether to enable medical alarm notification), **gasEventEnabled** (whether to enable gas alarm notification), **fireEventEnabled** (whether to enable fire alarm notification), **hostStatusEventEnabled** (whether to enable notification of security control panel status), **exDevStatusEventEnabled** (whether to enable peripheral status notification), and **detectorStatusEventEnabled** (whether to enable detector status notification) to the node **CloudCap**.
16. Extended parameter message of Hik-Connect notification ***JSON Cloud*** (related URL: </ISAPI/SecurityCP/Configuration/messageSendCloud?format=json> ; related API: ***NET DVR STDXMLConfig***):
added 10 sub nodes: **zoneAlarmTamperEnabled** (whether to enable alarm and tampering event notification of the supported zone), **exDevTamperEventEnabled** (whether to enable peripheral tampering alarm notification), **hostTamperEventEnabled** (whether to enable tampering alarm notification of security control panel), **emergencyEventEnabled** (whether to enable panic alarm notification), **medicalEventEnabled** (whether to enable medical alarm notification), **gasEventEnabled** (whether to enable gas alarm notification), **fireEventEnabled** (whether to enable fire alarm notification), **hostStatusEventEnabled** (whether to enable notification of security control panel status), **exDevStatusEventEnabled** (whether to enable peripheral status notification), and **detectorStatusEventEnabled** (whether to enable detector status notification) to the node **Cloud**.
17. Extended advanced configuration capability message of phone notification ***JSON PhoneAnvancedCap*** (related URL: </ISAPI/SecurityCP/Configuration/messageSendPhoneAnvanced/capabilities?format=json> ; related API: ***NET DVR STDXMLConfig***):
added 10 sub nodes: **zoneAlarmTamperEnabled** (whether to enable alarm and tampering event notification of the supported zone), **exDevTamperEventEnabled** (whether to enable peripheral tampering alarm notification), **hostTamperEventEnabled** (whether to enable tampering alarm notification of security control panel), **emergencyEventEnabled** (whether to enable panic alarm notification), **medicalEventEnabled** (whether to enable medical alarm notification), **gasEventEnabled** (whether to enable gas alarm notification), **fireEventEnabled** (whether to enable fire alarm notification), **hostStatusEventEnabled** (whether to enable notification of security control panel status), **exDevStatusEventEnabled** (whether to enable peripheral status notification), and **detectorStatusEventEnabled** (whether to enable detector status notification) to the node **Message** and **Call** of **PhoneAnvancedCap**;
added a sub node **numbersOfCalls** (phone call times) to the node **Call** of **PhoneAnvancedCap**.

18. Extended advanced parameter message of phone notification **JSON List PhoneAnvanced** (related URL: **/ISAPI/SecurityCP/Configuration/messageSendPhoneAnvanced?format=json** ; related API: **NET DVR STDXMLConfig**): added 10 sub nodes: **zoneAlarmTamperEnabled** (whether to enable alarm and tampering event notification of the supported zone), **exDevTamperEventEnabled** (whether to enable peripheral tampering alarm notification), **hostTamperEventEnabled** (whether to enable tampering alarm notification of security control panel), **emergencyEventEnabled** (whether to enable panic alarm notification), **medicalEventEnabled** (whether to enable medical alarm notification), **gasEventEnabled** (whether to enable gas alarm notification), **fireEventEnabled** (whether to enable fire alarm notification), **hostStatusEventEnabled** (whether to enable notification of security control panel status), **exDevStatusEventEnabled** (whether to enable peripheral status notification), and **detectorStatusEventEnabled** (whether to enable detector status notification) to the node **Message** and **Call** of **PhoneAnvanced** of **List**; added a sub node **numbersOfCalls** (phone call times) to the node **Call** of **PhoneAnvanced** of **List**.
19. Extended advanced parameter message of phone notification of a specific phone number **JSON PhoneAnvanced** (related URL: **/ISAPI/SecurityCP/Configuration/messageSendPhoneAnvanced/<ID>?format=json** ; related API: **NET DVR STDXMLConfig**): added 10 sub nodes: **zoneAlarmTamperEnabled** (whether to enable alarm and tampering event notification of the supported zone), **exDevTamperEventEnabled** (whether to enable peripheral tampering alarm notification), **hostTamperEventEnabled** (whether to enable tampering alarm notification of security control panel), **emergencyEventEnabled** (whether to enable panic alarm notification), **medicalEventEnabled** (whether to enable medical alarm notification), **gasEventEnabled** (whether to enable gas alarm notification), **fireEventEnabled** (whether to enable fire alarm notification), **hostStatusEventEnabled** (whether to enable notification of security control panel status), **exDevStatusEventEnabled** (whether to enable peripheral status notification), and **detectorStatusEventEnabled** (whether to enable detector status notification) to the node **Message** and **Call** of **PhoneAnvanced**; added a sub node **numbersOfCalls** (phone call times) to the node **Call** of **PhoneAnvanced**.
20. Added URLs of filtering duplicate zone alarms in the configured time interval (related API: **NET DVR STDXMLConfig**): Get configuration capability: GET **/ISAPI/SecurityCP/Configuration/zoneAlarmTimeFilter/capabilities?format=json** ; Get or set parameters: GET or PUT **/ISAPI/SecurityCP/Configuration/zoneAlarmTimeFilter?format=json** .
21. Extended alarm and event example message of the security control panel **JSON EventNotificationAlert SecurityCPAlarmEventMsg** : added 10 event types: "zoneAlarmTamper" (alarm and tampering event of the supported zone), "exDevTamperEvent" (peripheral tampering alarm), "hostTamperEvent" (tampering alarm of security control panel), "emergencyEvent" (panic alarm), "medicalEvent" (medical alarm), "gasEvent" (gas alarm), "fireEvent" (fire alarm), "hostStatusEvent" (security control panel status), "exDevStatusEvent" (peripheral status), and "detectorStatusEvent" (detector status) to the sub node **type** of the node **CIDEvent**;

- added two sub nodes: **zoneCompatible** (whether to support zone compatibility) and **alarmCenterNo** (alarm receiving center No.) to the node **CIDEvent**.
22. Extended configuration capability message of security control system **JSON_ManageCap** (related URL: </ISAPI/SecurityCP/Configuration/systemManage/capabilities?format=json> ; related API: **NET_DVR_STDXMLConfig**):
added two sub nodes: **wordVoiceEnabled** (whether to enable audio prompt) and **disArmAndClearAlarmVoicePrompt** (whether the security control panel plays the audio prompt of the fault once again when disarming or clearing alarms) to the node **ManageCap**.
23. Extended parameter message of security control system **JSON_Manage** (related URL: </ISAPI/SecurityCP/Configuration/systemManage?format=json> ; related API: **NET_DVR_STDXMLConfig**):
added two sub nodes: **wordVoiceEnabled** (whether to enable audio prompt) and **disArmAndClearAlarmVoicePrompt** (whether the security control panel plays the audio prompt of the fault once again when disarming or clearing alarms) to the node **Manage**.
24. Extended configuration capability message of security control panel **JSON_HostConfigCap** (related URL: </ISAPI/SecurityCP/Configuration/capabilities?format=json> ; related API: **NET_DVR_STDXMLConfig**):
added a sub node **isSptRemoteCfgPermissonUserType** (whether it supports getting user names of users that have the permission to remotely configure devices) to the node **ExDevice** of **HostConfigCap**;
added a sub node **isSptzoneAlarmTimeFilter** (whether it supports filtering duplicate zone alarms in the configured time interval) to the node **HostConfigCap**.

Summary of Changes in Version 6.1.0.15_July, 2019

1. Extended zone configuration capability message **JSON_ZonesCap** (related URL: </ISAPI/SecurityCP/Configuration/zones/capabilities?format=json> ; related API: **NET_DVR_STDXMLConfig**):
added a sub node **isSupportZonesOverlay** (whether to support zone overlay) to the node **ZonesCap** (wireless zone capability).
2. Extended parameter message of all logical relays **JSON_List_Output** and parameter message of a specific logical relay **JSON_Output** (related URLs: </ISAPI/SecurityCP/Configuration/outputs?format=json> and </ISAPI/SecurityCP/Configuration/outputs/<ID>?format=json> ; related API: **NET_DVR_STDXMLConfig**):
added two nodes: **alarmLine** (trigger line No.) and **followModeEnabled** (whether to enable following mode).
3. Extended configuration capability message of logical relay **JSON_OutputCap** (related URL: </ISAPI/SecurityCP/Configuration/outputs/capabilities?format=json> ; related API: **NET_DVR_STDXMLConfig**):
added three nodes: **alarmLine** (number of linked trigger lines), **alarmLineNo** (range of trigger line No.), and **followModelEnabled** (whether to enable following mode).
4. Extended the alarm event example message of the security control panel **JSON_EventNotificationAlert_SecurityCPAlarmEventMsg** :

added four sub nodes: **standardCIDcode** (standard CID code), **NVRList** (NVR information), **AlarmLineRule** (alarm rule), and **alarmLineNo1** (trigger line No.) to the node **CIDEvent** (CID alarm events).

Summary of Changes in Version 6.1.0.11_June, 2019

1. Extended module locking and unlocking capability message **JSON_ModuleLockCap** (related URL: **/ISAPI/System/moduleLock/config/capabilities?format=json** ; related API: **NET_DVR_STDXMLConfig**):
added two nodes: **address** (module address) and **sirenAttrib** (siren attribute).
2. Extended module locking parameter message **JSON_List_ModuleLock** (related URL: **/ISAPI/System/moduleLock/config?format=json** ; related API: **NET_DVR_STDXMLConfig**):
added two nodes: **address** (module address) and **moduleAttrib** (module attribute).
3. Extended zone configuration capability message **JSON_ZonesCap** (related URL: **/ISAPI/SecurityCP/Configuration/zones/capabilities?format=json** ; related API: **NET_DVR_STDXMLConfig**):
added a sub node **remoteRelatedChan** (range of remotely linked channel No.) to the node **RelatedChan** of **ZonesCap** (wireless zone capability);
added a sub node **remoteRelatedChan** (range of remotely linked channel No.) to the node **RelatedChan** of **WiredZonesCap** (wired zone capability).
4. Extended partition configuration capability message **JSON_SubSysCap** , parameter message of a specific partition **JSON_SubSys** , and parameter message of all partitions **JSON_List_SubSys** (related URLs: **/ISAPI/SecurityCP/Configuration/subSys/capabilities?format=json** , **/ISAPI/SecurityCP/Configuration/subSys/<ID>?format=json** , and **/ISAPI/SecurityCP/Configuration/subSys?format=json** ; related API: **NET_DVR_STDXMLConfig**):
added a node **oneKeyArmEnabled** (whether to enable one-touch arming).
5. Extended siren configuration capability message **JSON_SirenCap** (related URL: **/ISAPI/SecurityCP/Configuration/wirelessSiren/capabilities?format=json** ; related API: **NET_DVR_STDXMLConfig**):
added a node **sirenLinkageType** (type of siren that supports being linked).
6. Extended repeater configuration capability message **JSON_RepeaterCap** , parameter message of a specific repeater **JSON_Repeater** , and parameter message of all repeaters **JSON_List_Repeater** (related URLs: **/ISAPI/SecurityCP/Configuration/repeaters/capabilities?format=json** , **/ISAPI/SecurityCP/Configuration/repeaters/<ID>?format=json** , and **/ISAPI/SecurityCP/Configuration/repeaters?format=json** ; related API: **NET_DVR_STDXMLConfig**):
added a node **linkageAddress** (linked module address).
7. Extended condition message of getting logical relays' parameters **JSON_OutputCond** (related URL: **/ISAPI/SecurityCP/Configuration/outputs?format=json** ; related API: **NET_DVR_STDXMLConfig**):
added a node **moduleType** (module type).
8. Added the function of getting the list of keypads that can be added (related API: **NET_DVR_STDXMLConfig**):
Get the capability: GET **/ISAPI/SecurityCP/Configuration/keypadAddList/capabilities?format=json** ;

Get the list: GET [/ISAPI/SecurityCP/Configuration/keypadAddList?format=json](#) .

9. Added the function of extension module configuration, refer to [Extension Module Configuration](#) for details.
10. Extended keypad configuration capability message [JSON_KeypadCap](#) , parameter message of a specific keypad [JSON_Keypad](#) , and parameter message of all keypads [JSON_List_Keypad](#) (related URLs: [/ISAPI/SecurityCP/Configuration/keypad/capabilities?format=json](#) , [/ISAPI/SecurityCP/Configuration/keypad/<ID>?format=json](#) , and [/ISAPI/SecurityCP/Configuration/keypad?format=json](#) ; related API: [NET_DVR_STDXMLConfig](#)):
added five nodes: **address** (keypad address), **type** (keypad model), **version** (keypad version), **status** (keypad status), and **tamperEvident** (tampering status).
11. Extended configuration capability message of alarm receiving center [JSON_ARCCap](#) (related URL: [/ISAPI/SecurityCP/Configuration/ARC/capabilities?format=json](#) ; related API: [NET_DVR_STDXMLConfig](#)):
added a protocol type "PSTN-CID" to the sub node **protocol** of the node **ProtoList**.
12. Added the function of phone notification via PSTN (Public Switched Telephone Network) (related API: [NET_DVR_STDXMLConfig](#)):
Get the configuration capability: GET [/ISAPI/SecurityCP/Configuration/PSTNCfg/capabilities?format=json](#) ;
Get the parameters of all phone notifications: GET [/ISAPI/SecurityCP/Configuration/PSTNCfg?format=json](#) ;
Set the parameters of a specific phone notification: PUT [/ISAPI/SecurityCP/Configuration/PSTNCfg/<ID>?format=json](#) .
13. Extended operation and control capability message [JSON_HostControlCap](#) (related URL: [/ISAPI/SecurityCP/control/capabilities?format=json](#) ; related API: [NET_DVR_STDXMLConfig](#)):
added two nodes: **isSptSirenCtrl** (whether to support controlling siren) and **sirenCtrlType** (type of siren that supports being controlled).
14. Added a URL to control a specific siren (related API: [NET_DVR_STDXMLConfig](#)): PUT [/ISAPI/SecurityCP/control/siren/<ID>?format=json](#) .
15. Extended status message of all partitions [JSON_SubSysList](#) (related URL: [/ISAPI/SecurityCP/status/subSystems?format=json](#) ; related API: [NET_DVR_STDXMLConfig](#)):
added a node **enabled** (whether to enable the partition).
16. Extended status message of all peripherals [JSON_ExDevStatus](#) (related URL: [/ISAPI/SecurityCP/status/exDevStatus?format=json](#) ; related API: [NET_DVR_STDXMLConfig](#)):
added a sub node **sirenAttrib** (siren attribute) to the node **Siren** of **SirenList**;
added a sub node **moduleAttrib** (module attribute) to the node **ExtensionModule** of **ExtensionList**;
added two sub nodes: **keypadAttrib** (keypad attribute) and **address** (keypad address) to the node **Keypad** of **KeypadList**.
17. Extended fault parameter message of systems and partitions [JSON_ArmFault](#) (related URL: [/ISAPI/SecurityCP/status/systemFault?format=json](#) ; related API: [NET_DVR_STDXMLConfig](#)):
added 9 values: "RS485ZoneModTamperEvident" (RS-485 zone module tampered),
"RS485WirelessacceptorTamperEvident" (RS-485 wireless receiver module tampered),
"RS485ZoneModOffline" (RS-485 zone module offline), "RS485OutputModOffline" (RS-485

- output module offline), "RS485WirelessacceptorOffline" (RS-485 wireless receiver module offline), "telLineBroken" (telephone line disconnected), "RS485DisConnect" (RS-485 bus exception), "keypadTamperEvident" (keypad tampered), "keypadOffline" (keypad offline) to the sub node **info** (fault information) of the node **Fault** of **FaultList** in the node **SysFault**.
18. Extended search result message of zone status **JSON ZoneSearch** (related URL: [/ISAPI/SecurityCP/status/zones?format=json](#) ; related API: [NET DVR STDXMLConfig](#)): added a sub node **zoneAttrib** (zone attribute) to the node **Zone** of **ZoneList**.
 19. Extended condition message of getting relay status **JSON OutputCond** (related URL: [/ISAPI/SecurityCP/status/outputStatus?format=json](#) ; related API: [NET DVR STDXMLConfig](#)): added a node **moduleType** (module type).
 20. Extended siren status message **JSON SirenList** (related URL: [/ISAPI/SecurityCP/status/sirenStatus?format=json](#) ; related API: [NET DVR STDXMLConfig](#)): added a sub node **sirenAttrib** (siren attribute) to the node **Siren** of **SirenList**.
 21. Extended extension module status message **JSON ExtensionList** (related URL: [/ISAPI/SecurityCP/status/extensionModuleStatus?format=json](#) ; related API: [NET DVR STDXMLConfig](#)): added a sub node **moduleAttrib** (module attribute) to the node **ExtensionModule** of **ExtensionList**.
 22. Extended keypad status message **JSON KeypadList** (related URL: [/ISAPI/SecurityCP/status/keypadStatus?format=json](#) ; related API: [NET DVR STDXMLConfig](#)): added two nodes: **keypadAttrib** (keypad attribute) and **address** (keypad address) to the node **Keypad** of **KeypadList**.
 23. Extended parameter message of all statuses of security control panel **JSON AlarmHostStatus** (related URL: [/ISAPI/SecurityCP/status/host?format=json](#) ; related API: [NET DVR STDXMLConfig](#)): added a sub node **enabled** (whether to enable the partition) to the node **SubSys** of **SubSysList**; added a sub node **sirenAttrib** (siren attribute) to the node **SirenList** of **ExDevStatus**; added a sub node **moduleAttrib** (module attribute) to the node **ExtensionModule** of **ExtensionList** in the node **ExDevStatus**; added two sub nodes: **keypadAttrib** (keypad attribute) and **address** (keypad address) to the node **Keypad** of **KeypadList** in the node **ExDevStatus**.
 24. Extended configuration capability message of fault detection **JSON FaultCheckParameterCap** and fault detection parameter message **JSON FaultCheckParameter** (related URLs: [/ISAPI/SecurityCP/Configuration/faultCheckCfg/capabilities?format=json](#) and [/ISAPI/SecurityCP/Configuration/faultCheckCfg?format=json](#) ; related API: [NET DVR STDXMLConfig](#)): added two nodes: **telLineBrokenEnabled** (whether to enable telephone line disconnection detection) and **RS485AbnormalEnabled** (whether to enable RS-485 exception detection).
 25. Added the function of configuring keypad linkage of system fault (related API: [NET DVR STDXMLConfig](#)): Get the configuration capability: GET [/ISAPI/SecurityCP/Configuration/keypadFaultProcessCfg/capabilities?format=json](#) ; Get the linkage parameters of all keypads: GET [/ISAPI/SecurityCP/Configuration/keypadFaultProcessCfg?format=json](#) ;

Set the linkage parameters of a specific keypad: PUT [/ISAPI/SecurityCP/Configuration/keypadFaultProcessCfg/<ID>?format=json](#) .

26. Added the function of configuring registration mode (related API: [NET DVR STDXMLConfig](#)):
Get the configuration capability: GET [/ISAPI/SecurityCP/Configuration/registerMode/capabilities?format=json](#) ;
Get the parameters: GET [/ISAPI/SecurityCP/Configuration/registerMode?format=json](#) ;
Set the parameters: PUT [/ISAPI/SecurityCP/Configuration/registerMode?format=json](#) .
27. Extended configuration capability message of security control panel **JSON HostConfigCap** (related URL: [/ISAPI/SecurityCP/Configuration/capabilities?format=json](#) ; related API: [NET DVR STDXMLConfig](#)):
added a node **isSptKeypadFaultProcessCfg** (whether the device supports configuring keypad linkage parameters of the system fault).
28. Extended the log types **Log Types for ISAPI** :
added four alarm log types: "RS-485AlarmInputModuleEvident" (RS-485 Zone Module Tampered), "RS-485AlarmInputModuleTamperReset" (RS-485 Zone Module Tampering Reset), "RS-485WirelessReceiverTamperEvident" (RS-485 Wireless Receiver Module Tampered), "RS-485WirelessReceiverTamperEvidentReset" (RS-485 Wireless Receiver Module Tampering Reset);
added six exception log types: "RS-485AlarmInputModuleDisconnected" (RS-485 Zone Module Offline), "RS-485AlarmInputModuleConnected" (RS-485 Zone Module Online), "RS-485WirelessReceiverDisconnected" (RS-485 Wireless Receiver Module Offline), "RS-485WirelessReceiverConnected" (RS-485 Wireless Receiver Module Online), "keypadDisconnected" (Keypad Offline), and "keypadConnected" (Keypad Online);
added 9 operation log types: "delRS-485InputModule" (RS-485 Zone Module Deleted), "delRS-485OutputModule" (RS-485 Output Module Deleted), "delRS-485WirelessReceiver" (RS-485 Wireless Receiver Module Deleted), "enrollRS-485InputModule" (RS-485 Zone Module Registered), "enrollRS-485OutputModule" (RS-485 Output Module Registered), "delRS-485OutputModule" (RS-485 Output Module Deleted), "enrollRS-485WirelessReceiver" (RS-485 Wireless Receiver Module Registered), "enrollKeypad" (Keypad Registered), and "delKeypad" (Keypad Deleted).

Summary of Changes in Version 6.1.0.11_June, 2019

1. Added the function of locking and unlocking module, refer to [Module Locking and Unlocking](#) for details.
2. Extended the zone configuration capability message **JSON ZonesCap** (related URL: [/ISAPI/SecurityCP/Configuration/zones/capabilities?format=json](#) ; related API: [NET DVR STDXMLConfig](#)):
added a node **WiredZonesCap** (wired zone capability);
added 10 sub nodes: **address** (module address), **linkageAddress** (linked module address), **moduleChannel** (module channel No.), **moduleType** (module type), **moduleStatus** (module status), **CheckTimeList** (range list of detector offline time), **sensitivity** (zone sensitivity), **resistor** (EOL resistor), **tamperType** (tampering type), and **zoneAttrib** (zone attribute) to the node **ZonesCap** (wireless zone capability).

3. Extended the parameter message of all zones **JSON List Zone** (related URL: [/ISAPI/SecurityCP/Configuration/zones?format=json](#) ; related API: **NET_DVR_STDXMLConfig**): added 10 sub nodes: **address** (module address), **linkageAddress** (linked module address), **moduleChannel** (module channel No.), **moduleType** (module type), **moduleStatus** (module status), **checkTime** (detector offline duration), **sensitivity** (zone sensitivity), **resistor** (EOL resistor), **tamperType** (tampering type), and **zoneAttrib** (zone attribute) to the node **Zone** (zone parameters).
4. Added the function of getting a specific zone parameters (related API: **NET_DVR_STDXMLConfig**): GET [/ISAPI/SecurityCP/Configuration/zones/<ID>?format=json](#) .
5. Extended a specific zone parameter message **JSON Zone** (related URL: [/ISAPI/SecurityCP/Configuration/zones/<ID>?format=json](#) ; related API: **NET_DVR_STDXMLConfig**): added 10 nodes: **address** (module address), **linkageAddress** (linked module address), **moduleChannel** (module channel No.), **moduleType** (module type), **moduleStatus** (module status), **checkTime** (detector offline duration), **sensitivity** (zone sensitivity), **resistor** (EOL resistor), **tamperType** (tampering type), and **zoneAttrib** (zone attribute).
6. Added the function of getting unlinked zones (related API: **NET_DVR_STDXMLConfig**): Get the capability: GET [/ISAPI/SecurityCP/Configuration/notRelateZones/capabilities?format=json](#) ;
Get unlinked zones: GET [/ISAPI/SecurityCP/Configuration/notRelateZones?format=json](#) .
7. Added the function of partition configuration, refer to **Partition Configuration** for details.
8. Added the function of configuring the security control panel timer (related API: **NET_DVR_STDXMLConfig**): Get the capability: GET [/ISAPI/SecurityCP/Configuration/deviceTime/capabilities?format=json](#) ;
Get parameters: GET [/ISAPI/SecurityCP/Configuration/deviceTime?format=json](#) ;
Set parameters: PUT [/ISAPI/SecurityCP/Configuration/deviceTime?format=json](#) .
9. Extended the siren configuration capability message **JSON SirenCap** (related URL: [/ISAPI/SecurityCP/Configuration/wirelessSiren/capabilities?format=json](#) ; related API: **NET_DVR_STDXMLConfig**): added seven nodes: **supportVolumeIDList** (ID list of sirens that support volume configuration), **address** (module address), **linkageAddress** (linked module address), **checkTime** (offline duration), **sirenAttrib** (siren attribute), **linkage** (event type linked to the siren), and **zoneEvent** (zone event type).
10. Extended the parameter message of all sirens **JSON List Siren** and parameter message of a specific siren **JSON Siren** (related URLs: [/ISAPI/SecurityCP/Configuration/wirelessSiren?format=json](#) and [/ISAPI/SecurityCP/Configuration/wirelessSiren/<ID>?format=json](#) ; related API: **NET_DVR_STDXMLConfig**): added six nodes: **address** (module address), **linkageAddress** (linked module address), **checkTime** (offline duration), **sirenAttrib** (siren attribute), **linkage** (event linkage type), and **zoneEvent** (zone event type).
11. Extended the repeater configuration capability message **JSON RepeaterCap** , parameter message of all repeaters **JSON List Repeater** , and parameter message of a specific repeater **JSON Repeater** (related URLs: [/ISAPI/SecurityCP/Configuration/repeaters/capabilities?](#)

format=json , */ISAPI/SecurityCP/Configuration/repeaters?format=json* , and */ISAPI/SecurityCP/Configuration/repeaters/<ID>?format=json* ; related API:

NET_DVR_STDXMLConfig):

added a node **checkTime** (offline duration).

12. Extended the configuration capability message of output module *JSON_OutputModuleCap* , parameter message of all output modules *JSON_List_OutputModule* , and parameter message of a specific output module *JSON_OutputModule* (related URLs: */ISAPI/SecurityCP/Configuration/outputModules/capabilities?format=json* , */ISAPI/SecurityCP/Configuration/outputModules?format=json* , and */ISAPI/SecurityCP/Configuration/outputModules/<ID>?format=json* ; related API: *NET_DVR_STDXMLConfig*):
added four nodes: **address** (module address), **linkageAddress** (linked module address), **attrib** (module attribute), and **checkTime** (offline duration).
13. Extended the configuration capability message of logical relay *JSON_OutputCap* (related URL: */ISAPI/SecurityCP/Configuration/outputs/capabilities?format=json* ; related API: *NET_DVR_STDXMLConfig*):
added a value "zone" to the node **linkage** (event types linked to the relay);
added six nodes: **zoneEvent** (zone event type), **moduleType** (module type), **address** (module address), **linkageAddress** (linked module address), **moduleChannel** (module channel No.), and **maxOutputsResults** (maximum number of returned results);
added a value "getCond" (get by conditions) to the node **method** (methods supported by the function).
14. Extended the parameter message of all logical relays *JSON_List_Output* and parameter message of a specific logical relay *JSON_Output* (related URLs: */ISAPI/SecurityCP/Configuration/outputs?format=json* and */ISAPI/SecurityCP/Configuration/outputs/<ID>?format=json* ; related API: *NET_DVR_STDXMLConfig*):
added a value "zone" to the node **linkage** (event types linked to the relay);
added five nodes: **zoneEvent** (zone event type), **moduleType** (module type), **address** (module address), **linkageAddress** (linked module address), and **moduleChannel** (module channel No.).
15. Added a URL to get logical relays' parameters by condition (related API: *NET_DVR_STDXMLConfig*): POST */ISAPI/SecurityCP/Configuration/outputs?format=json* .
16. Extended the keyfob configuration capability message *JSON_RemoteCtrlCap* (related URL: */ISAPI/SecurityCP/Configuration/remoteCtrl/capabilities?format=json* ; related API: *NET_DVR_STDXMLConfig*):
added two nodes **subSystem** (supported number of partitions that can be linked) and **subSystemNo** (range of partition No.);
added a value "currentAddAsyn" (asynchronously add) to the node **method** (methods supported by the function).
17. Extended the parameter message of all keyfobs *JSON_List_RemoteCtrl* and parameter message of a specific keyfob *JSON_RemoteCtrl* (related URLs: */ISAPI/SecurityCP/Configuration/remoteCtrl?format=json* and */ISAPI/SecurityCP/Configuration/remoteCtrl/<ID>?format=json* ; related API: *NET_DVR_STDXMLConfig*):
added a node **subSystem** (linked partitions).

18. Added the function of controlling asynchronous keyfob adding mode (related API: **NET_DVR_STDXMLConfig**):
Get the capability: GET **/ISAPI/SecurityCP/Configuration/remoteCtrl/mode/capabilities?format=json** ;
Set the parameters: PUT **/ISAPI/SecurityCP/Configuration/remoteCtrl/mode?format=json** .
19. Added a URL to get parameters of the asynchronously added keyfob (related API: **NET_DVR_STDXMLConfig**): GET **/ISAPI/SecurityCP/Configuration/remoteCtrl/currentAddAsyn?format=json** .
20. Extended the card configuration capability message **JSON_CardCap** (related URL: **/ISAPI/SecurityCP/Configuration/card/capabilities?format=json** ; related API: **NET_DVR_STDXMLConfig**):
added four nodes: **subSystem** (supported number of partitions that can be linked), **subSystemNo** (partition No. range), **cardType** (card type), and **method** (methods supported by the function).
21. Extended the parameter message of a card **JSON_Card** and parameter message of all cards **JSON_List_Card** (related URLs: **/ISAPI/SecurityCP/Configuration/card/<ID>?format=json** and **/ISAPI/SecurityCP/Configuration/card?format=json** ; related API: **NET_DVR_STDXMLConfig**):
added two nodes **subSystem** (linked partitions) and **cardType** (card type).
22. Added the function of controlling asynchronous card adding mode (related API: **NET_DVR_STDXMLConfig**):
Get the capability: GET **/ISAPI/SecurityCP/Configuration/card/mode/capabilities?format=json** ;
Set the parameters: PUT **/ISAPI/SecurityCP/Configuration/card/mode?format=json** .
23. Added a URL to get parameters of the asynchronously added card (related API: **NET_DVR_STDXMLConfig**): GET **/ISAPI/SecurityCP/Configuration/card/currentAddAsyn?format=json** .
24. Added the function of card reader configuration, refer to **Card Reader Configuration** for details.
25. Added the function of keypad configuration, refer to **Keypad Configuration** for details.
26. Added the URLs to configure advanced parameters of phone notification (related API: **NET_DVR_STDXMLConfig**):
Get the capability: GET **/ISAPI/SecurityCP/Configuration/messageSendPhoneAnvanced/capabilities?format=json** ;
Get the parameters: GET **/ISAPI/SecurityCP/Configuration/messageSendPhoneAnvanced?format=json** ;
Set the parameters: PUT **/ISAPI/SecurityCP/Configuration/messageSendPhoneAnvanced/<ID>?format=json** .
27. Extended security control panel alarm/event message **JSON_EventNotificationAlert_SecurityCPAlarmEventMsg** :
added three nodes: **uuid** (a unique ID to identify an event), **recheck** (mark of rechecking the alarm), and **videoURL** (video URL);
added three sub nodes: **cardReader** (card reader No.), **cardNo** (card No.), and **cardType** (card type) to the node **CIDEvent** (CID events).

28. Extended the capability message of security control panel status ***JSON HostStatusCap*** (related URL: ***/ISAPI/SecurityCP/status/capabilities?format=json*** ; related API: ***NET_DVR_STDXMLConfig***):
added 11 nodes: **isSptCondZones** (whether to support getting zone status by conditions), **maxZonesResults** (maximum number of zone status that can be obtained this time by calling the URL), **isSptOutputMod** (whether to support getting output module status), **isSptCondOutputs** (whether to support getting relay status by conditions), **maxOutputsResults** (maximum number of output module status that can be obtained this time by calling the URL), **isSptOutputs** (whether to support getting relay status), **isSptSirenMod** (whether to support getting siren status), **isSptRepeaterMod** (whether to support getting repeater status), **isSptCardReaderMod** (whether to support getting card reader status), **isSptExtensionModuleMod** (whether to support getting extension module status), and **isSptKeypadMod** (whether to support getting keypad status).
29. Extended arming status message of a specific partition ***JSON ArmStatusList*** (related URL: ***/ISAPI/SecurityCP/status/armStatus?format=json*** ; related API: ***NET_DVR_STDXMLConfig***):
added 9 values: "wirelessKeypadTamperEvident" (wireless keypad tampered), "wirelessCardReaderTamperEvident" (card reader tampered), "wirelessKeypadOffline" (wireless keypad offline), "wirelessCardReaderOffline" (card reader offline), "wKeypadOvertime" (keypad heartbeat timeout), "wCardReaderOvertime" (card reader heartbeat timeout), "keypadLowPower" (low keypad battery), "cardReaderLowPower" (low card reader battery), and "ARCUUploadFailed" (ARC uploading failed) to the node **info** (fault information).
30. Extended status message of all peripherals ***JSON ExDevStatus*** (related URL: ***/ISAPI/SecurityCP/status/exDevStatus?format=json*** ; related API: ***NET_DVR_STDXMLConfig***):
added three nodes: **CardReaderList** (card reader list), **ExtensionList** (extension module list), and **KeypadList** (keypad list).
31. Extended the fault parameter message of systems and partitions ***JSON ArmFault*** (related URL: ***/ISAPI/SecurityCP/status/systemFault?format=json*** ; related API: ***NET_DVR_STDXMLConfig***):
added 8 values: "wirelessKeypadTamperEvident" (wireless keypad tampered), "wirelessCardReaderTamperEvident" (card reader tampered), "wirelessKeypadOffline" (wireless keypad offline), "wirelessCardReaderOffline" (card reader offline), "wKeypadOvertime" (keypad heartbeat timeout), "wCardReaderOvertime" (card reader heartbeat timeout), "keypadLowPower" (low keypad battery), and "cardReaderLowPower" (low card reader battery) to the node **info** (fault information).
32. Added the URLs to get device (including output module, siren, repeater, card reader, extension module, and keypad), zone, and relay status by conditions, refer to ***Status Monitoring*** for details.
33. Extended parameter message of all statuses of security control panel ***JSON AlarmHostStatus*** (related URL: ***/ISAPI/SecurityCP/status/host?format=json*** ; related API: ***NET_DVR_STDXMLConfig***):
added three nodes: **CardReaderList** (card reader list), **ExtensionList** (extension module list), and **KeypadList** (keypad list).

34. Extended the configuration capability message of recording based on event ***JSON EventRecordCap*** (related URL: ***/ISAPI/SecurityCP/Configuration/eventRecord/channels/<ID>/capabilities?format=json*** ; related API: ***NET_DVR_STDXMLConfig***): added two nodes **recordTime** (supported recording schedule template) and **method** (methods supported by the function).
35. Added a URL to set the parameters of recording based on event (related API: ***NET_DVR_STDXMLConfig***): PUT ***/ISAPI/SecurityCP/Configuration/eventRecord/channels/<ID>?format=json*** .
36. Extended the configuration capability message of fault detection ***JSON FaultCheckParameterCap*** and fault detection parameter message ***JSON FaultCheckParameter*** (related URLs: ***/ISAPI/SecurityCP/Configuration/faultCheckCfg/capabilities?format=json*** and ***/ISAPI/SecurityCP/Configuration/faultCheckCfg?format=json*** ; related API: ***NET_DVR_STDXMLConfig***): added a node **ACcheckTime** (detection time when the AC is powered off).
37. Extended the capability message of security control panel ***JSON SecurityCPCap*** (related URL: ***/ISAPI/SecurityCP/capabilities?format=json*** ; related API: ***NET_DVR_STDXMLConfig***): added three nodes: **cardNum** (number of cards), **keypadNum** (number of keypads), and **cardReaderNum** (number of card readers).
38. Extended the configuration capability message of security control panel ***JSON HostConfigCap*** (related URL: ***/ISAPI/SecurityCP/Configuration/capabilities?format=json*** ; related API: ***NET_DVR_STDXMLConfig***): added five nodes: **isSptNotRelateZones** (whether the device supports getting unlinked zones), **isSptSubSys** (whether the device supports partition configuration), **isSptPublicSubSys** (whether the device supports public partition configuration), **isSptDeviceTime** (whether the device supports timer configuration of the security control panel), and **isSptRegisterMode** (whether the device supports registration mode configuration); added four nodes: **isSptExtensionModule** (whether the device supports extension module management), **isSptCardReader** (whether the device supports card reader configuration), **isSptKeypad** (whether the device supports keypad configuration), and **isSptKeypadAddList** (whether the device supports getting the list of keypads that can be added) to the node **ExDevice**; added two nodes **isSptPhoneAnvanced** (whether the device supports advanced configuration of phone notification and SMS notification) and **isSptPSTNCfg** (whether the device supports configuration of reporting by phone via PSTN (Public Switched Telephone Network)) to the node **MsgSend**.
39. Extended the device capability message ***XML DeviceCap*** (related URL: ***/ISAPI/System/capabilities*** ; related API: ***NET_DVR_STDXMLConfig***): added a node **<isSupportModuleLock>** (whether to support locking the module).
40. Extended the log types ***Log Types for ISAPI*** : added four alarm log types: "wirelessKeypadTamperEvident" (Wireless Keypad Tampered), "wirelessKeypadTamperEvidentReset" (Wireless Keypad Tamper Restored), "wirelessCardReaderTamperEvident" (Wireless Card Reader Tampered), and "wirelessCardReaderTamperEvidentReset" (Wireless Card Reader Tamper Restored);

added 12 exception logs: "wirelessKeypadOffline" (Wireless Keypad Disconnected), "wirelessKeypadOnline" (Wireless Keypad Connected), "wirelessCardReaderOffline" (Wireless Card Reader Disconnected), "wirelessCardReaderOnline" (Wireless Card Reader Connected), "keypadLowPower" (Low Keypad Battery), "keypadLowPowerRecovery" (Low Keypad Battery Recovered), "cardReaderLowPower" (Low Card Reader Battery), "cardReaderLowPowerRecovery" (Low Card Reader Battery Recovered), "wKeypadOvertime" (Wireless Keypad Heartbeat Timed Out), "wKeypadOvertimeRecovery" (Wireless Keypad Heartbeat Timeout Recovered), "wCardReaderOvertime" (Wireless Card Reader Heartbeat Timed Out), and "wCardReaderOvertimeRecovery" (Wireless Card Reader Heartbeat Timeout Recovered).

Summary of Changes in Version 5.3.6.26_Jan., 2019

1. Extended the structure of **NET_DVR_ALARMHOST_MAIN_STATUS_V51** (related API: **NET_DVR_GetDVRConfig**): added one zone arming/disarming status to **bySetupAlarmStatus**: 2-being armed; added one trigger status to **byAlarmOutStatus**: 4-heartbeat exception; added one partition arming/disarming status to **bySubSystemGuardStatus**: 2-being armed.
2. Extended the structure of **NET_DVR_ALARMHOST_OTHER_STATUS_V51** (related API: **NET_DVR_GetDVRConfig**): added one siren status to **bySirenStatus**: 4-heartbeat exception; added one detector connection status to **byDetectorConnection**: 3-heartbeat exception.

Summary of Changes in Version 5.3.5.55_May, 2019

1. Extended the structure of all modules information **NET_DVR_MODULE_INFO** (related API: **NET_DVR_GetNextRemoteConfig**): added one module type: 4-network module.
2. Extended the structure about data uploading configuration **NET_DVR_ALARMHOST_REPORT_CENTER_CFG_V40** (related API: **NET_DVR_GetDeviceConfig** **NET_DVR_SetDeviceConfig**): added one parameter **byAlarmNetCard** (alarm NIC center of central group)
3. Extended the zone parameters structure **NET_DVR_ALARMIN_PARAM_V50** (related API: **NET_DVR_GetDVRConfig** **NET_DVR_SetDVRConfig**): added four parameters **wTimeOut** (timeout), **byTimeOutRange** (timeout range), **byDetectorSignalIntensity** (detector signal strength), and **byTimeOutMethod** (timing method of over time zone); added two detector types to **wDetectorType**: **MEDICAL_HELP_BUTTON** (medical help button) and **OUTDOOR_DUAL_TECH** (outdoor dual-technology sensor); added two zone types to **byType**: 12-over time zone and 13-emergency zone; added one module type: 10-"8-zone wireless".
4. Extended the keyfob user parameters structure **NET_DVR_REMOTECONTROLLER_PERMISSION_CFG** (related API: **NET_DVR_GetDVRConfig** **NET_DVR_SetDVRConfig**): added one parameter **byEnableDel** (enable deleting keyfob user or not).

Summary of Changes in Version 5.3.6.25_Aug., 2018

New document.

Chapter 2 Typical Applications

2.1 System Configuration

Before implementing different applications, such as zone management, system status monitoring, functional key configuration, and so on, you should enable the functions via the system configuration by calling API to transmit the request URIs with messages.

Security Control System Settings

Function	Description
Get Security Control System's Configuration Capability	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET / <i>ISAPI/SecurityCP/Configuration/systemManage/capabilities?format=json</i> , and the capability is returned in the message <i>JSON_ManageCap</i> by <i>lpOutputParam</i> .
Get Security Control System Parameters	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET / <i>ISAPI/SecurityCP/Configuration/systemManage?format=json</i> , and the system configuration parameters are returned in the message <i>JSON_Manage</i> by <i>lpOutputParam</i> .
Set Security Control System Parameters	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: PUT / <i>ISAPI/SecurityCP/Configuration/systemManage?format=json</i> , and set <i>lpInputParam</i> to the message <i>JSON_Manage</i> .

Security Control Panel Capability

Function	Description
Get Security Control Panel's Capability	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET / <i>ISAPI/SecurityCP/capabilities?format=json</i> , and the capability is returned in the message <i>JSON_SecurityCPCap</i> by <i>lpOutputParam</i> .
Get Security Control Panel's Configuration Capability	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET / <i>ISAPI/SecurityCP/Configuration/capabilities?format=json</i> , and the configuration capability is returned in the message of <i>JSON_HostConfigCap</i> by <i>lpOutputParam</i> .


Audio Input and Output Settings


Function	Description
Get Configuration Capability of Audio Input and Output	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/SecurityCP/BasicParam/audiInOutCfg/capabilities</u> . The configuration capability is returned in the message <u>XML_Cap_AudiInOutCfg</u> by lpOutputParam .
Get Audio Input and Output Parameters	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/SecurityCP/BasicParam/audiInOutCfg</u> . The parameters are returned in the message <u>XML_AudiInOutCfg</u> by lpOutputParam .
Set Audio Input and Output Parameters	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: PUT / <u>ISAPI/SecurityCP/BasicParam/audiInOutCfg</u> and set lpInputParam to the message <u>XML_AudiInOutCfg</u> .

Note

To check whether the device supports configuring audio input and output parameters, you can call **NET_DVR_GetSTDAbility** and set **dwAbilityType** to "NET_DVR_GET_EMERGENCY_ALARM_PRODUCT_CAP" (macro definition value: 2212) for getting the capability of the one-touch panic alarm product.
The capability is returned in the message **XML_EmergencyAlarmProductCap** by **lpOutBuffer** of structure **NET_DVR_STD_ABILITY** . If this function is supported, the node <audiInOutCfg> will be returned and its value is true.

Audio File Management

Function	Description
Delete Audio File	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URL: DELETE / <u>ISAPI/SecurityCP/Control/audioFile/name=</u> .  Note To check whether the device supports deleting the audio file, you can call <u>NET_DVR_GetSTDAbility</u> and set dwAbilityType to "NET_DVR_GET_EMERGENCY_ALARM_PRODUCT_CAP" (macro

Function		Description
		definition value: 2212) for getting the capability of the one-touch panic alarm product. The capability is returned in the message <u>XML_EmergencyAlarmProductCap</u> by IpOutBuffer of the structure <u>NET_DVR_STD_ABILITY</u> . If this function is supported, the node <deleteAudioFile> will be returned and its value is true.
Get Audio File List	Get Capability of Getting Audio File List	<p>Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URL: GET <u>/ISAPI/SecurityCP/BasicParam/audioFileList/capabilities</u> .</p> <p>The capability is returned in the message <u>XML_Cap_AudioFileList</u> by the output parameter IpOutputParam.</p> <p> Note</p> <p>To check whether the device supports getting the audio file list, you can call <u>NET_DVR_GetSTDAbility</u> and set dwAbilityType to "NET_DVR_GET_EMERGENCY_ALARM_PRODUCT_CAP" (macro definition value: 2212) for getting the capability of the one-touch panic alarm product. The capability is returned in the message <u>XML_EmergencyAlarmProductCap</u> by IpOutBuffer of the structure <u>NET_DVR_STD_ABILITY</u>. If this function is supported, the node <getAudioFileListByType> will be returned and its value is true.</p>
	Get Audio File List	<p>Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URL: GET <u>/ISAPI/SecurityCP/BasicParam/audioFileList/type=</u> .</p> <p>The audio file list is returned in the message <u>XML_AudioFileList</u> by the output parameter IpOutputParam.</p>

Muting Schedule Configuration

Function	Description
Get Muting Schedule Configuration Capability	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET <i>/ISAPI/SecurityCP/Configuration/muteVoicePlanCFG/capabilities?format=json</i> . The capability is returned in the message <i>JSON_Cap_MuteVoicePlanCFG</i> by <i>IpOutputParam</i> .
Get Muting Schedule Parameters	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET <i>/ISAPI/SecurityCP/Configuration/muteVoicePlanCFG?format=json</i> . The parameters are returned in the message <i>JSON_MuteVoicePlanCFG</i> by <i>IpOutputParam</i> .
Set Muting Schedule Parameters	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: PUT <i>/ISAPI/SecurityCP/Configuration/muteVoicePlanCFG?format=json</i> and set <i>IpInputParam</i> to the message <i>JSON_MuteVoicePlanCFG</i> .

Voice Prompt Configuration

Function	Description
Get Voice Prompt Configuration Capability	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET <i>/ISAPI/SecurityCP/voicePrompt/capabilities?format=json</i> . The capability is returned in the message <i>JSON_Cap_voicePromptCfq</i> by <i>IpOutputParam</i> .
Get Voice Prompt Parameters	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET <i>/ISAPI/SecurityCP/voicePrompt?format=json</i> . The parameters are returned in the message <i>JSON_voicePromptCfq</i> by <i>IpOutputParam</i> .
Set Voice Prompt Parameters	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: PUT <i>/ISAPI/SecurityCP/voicePrompt?format=json</i> and set <i>IpInputParam</i> to the message <i>JSON_voicePromptCfq</i> .
Upload Audio File of Custom Voice Prompt	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: POST <i>/ISAPI/SecurityCP/videoBroadcast/customizeUpload?format=json</i> and set <i>IpInputParam</i> to the parameter <i>cycleTimes</i> if needed.

2.2 User Management

You can add, set, and delete users of the security control panel. When setting the users, you can assign the permissions of message notification, bypass, bypass recovery, and so on to the users.

User Permission

Four types of users with different permissions are adopted in the security specifications to control the security control panel.

Table 2-1 Permissions of Different User Types

Function	Permission			
	Operator	Administrator	Installer	Manufacturer
Arm	Depend on Configured Permissions	Yes	Yes	No
Disarm	Depend on Configured Permissions	Yes	Yes	No
Clear Alarms	Depend on Configured Permissions	Yes	Yes	No
Enable Pacing Mode	Depend on Configured Permissions	Yes	Yes	No
View Logs	Depend on Configured Permissions	Yes	Yes	No
Bypass Zone/ Disable Zone/ Force Arming	Depend on Configured Permissions	Yes	Yes	No
Add/Change Authorization Code	Depend on Configured Permissions	Yes	Yes	Yes
Add/Delete Users with Operator or Administrator Permission and	Depend on Configured Permissions	Yes	Yes	No

Authorization Code				
Add/Edit Configuration Parameters	No	No	Yes	No
Replace Software and Firmware	No	No	No	Yes

Function	Description
Get User Management Capability	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/SecurityCP/Configuration/users/capabilities?format=json</u> , and the user management capability is returned in the message of <u>JSON_UserCfgCap</u> .
User Configuration	Get All Users' Parameters Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/SecurityCP/Configuration/users?format=json</u> , and the configuration parameters are returned in the message of <u>JSON_List_UserCfg</u> .
	Set One User's Parameters Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: PUT / <u>ISAPI/SecurityCP/Configuration/users/<ID>?format=json</u> , and set the request message to <u>JSON_UserCfg</u> .
Add User	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: POST / <u>ISAPI/SecurityCP/Configuration/users?format=json</u> , and set the request message to <u>JSON_UserCfg</u> . The IDs of added users are returned by the message of <u>JSON_id</u> .
Delete A User	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: DELETE / <u>ISAPI/SecurityCP/Configuration/users/<ID>?format=json</u> .
Get Capability of Getting User Names of Users that Can Remotely Configure Devices	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/SecurityCP/Configuration/remoteCfgPermissonUserName/capabilities?format=json</u> . The capability is returned in the message <u>JSON_Cap_RemoteCfgUserName</u> by <u>lpOutBuffer</u> of <u>lpOutputParam</u> .
Get User Names of Users that Can Remotely Configure Devices	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/SecurityCP/Configuration/remoteCfgPermissonUserName?format=json</u> .

Function	Description
	The user names are returned in the message <u>JSON_RemoteCfgUserName</u> by IpOutBuffer of IpOutputParam .
Get Device User Configuration of Security Control Panels	<u>NET_DVR_GetAlarmDeviceUser</u>
Set Device User Configuration of Security Control Panels	<u>NET_DVR_SetAlarmDeviceUser</u>

2.3 Alarm Module Configuration

The alarm module refers to the zone, partition, siren, relay, repeater, keyfob, card, zone linked detector, and so on. To trigger and receive the alarms or events of security control system, you should firstly configure the alarm modules.

You can get all modules' information by the following steps.

1. Call **NET_DVR_StartRemoteConfig** with the command of "NET_DVR_GET_ALARMHOST_MODULE_LIST" (command No.: 1222) and set the input parameter **IpInBuffer** to "1"-keyboard, "2"-trigger, "3"-zone, or "4"-network module, for getting all modules information.
2. Call **NET_DVR_GetNextRemoteConfig** to get information one by one, and the information is returned in the structure **NET_DVR_MODULE_INFO**
3. When all data is obtained or getting failed, call **NET_DVR_StopRemoteConfig** to stop getting and release resources.

2.3.1 Detector Configuration

The detector is an alarm device that can be connected to a security system to provide notification of an alarm/event to the control panel.

General Detector Configuration

Function	Description
Get the capability of wired detectors according to the wired detector type	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: POST / <u>ISAPI/SecurityCP/Configuration/wiredDetector/capabilities?format=json</u> and set IpInputParam to the message <u>JSON_WiredDetectorType</u> .

Function	Description
	The capability is returned in the message <u>JSON_WiredDetectorCap</u> by IpOutputParam .
Get Detector Configuration Capability	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/SecurityCP/BasicParam/DetectorCfg/capabilities</u> , and the capability is returned in the message <u>XML_Cap_DetectorCfg</u> by IpOutputParam .
Get Detector's Parameters	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: POST / <u>ISAPI/SecurityCP/BasicParam/DetectorCfg</u> and set IpInputParam to the message <u>XML_ZoneCondList</u> . The detector configuration parameters are returned in the message <u>XML_DetectorCfg</u> by IpOutputParam .
Set Detector's Parameters	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: PUT / <u>ISAPI/SecurityCP/BasicParam/DetectorCfg</u> and set IpInputParam to the message <u>XML_DetectorCfg</u> .

Zone Linked Detector

Function	Description
Get Configuration Capability of Zone Linked Detector	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/SecurityCP/BasicParam/ZoneAssociatedDetectorCfg/capabilities</u> , and the capability is returned in the message <u>XML_Cap_ZoneAssociatedDetectorCfg</u> by IpOutputParam .
Get Zone Linked Detector's Parameters	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: POST / <u>ISAPI/SecurityCP/BasicParam/ZoneAssociatedDetectorCfg</u> and set IpInputParam to the message <u>XML_ZoneCondList</u> . The configuration parameters of zone linked detector are returned in the message <u>XML_ZoneAssociatedDetectorCfg</u> by IpOutputParam .
Set Zone Linked Detector's Parameters	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: PUT / <u>ISAPI/SecurityCP/BasicParam/ZoneAssociatedDetectorCfg</u> and set IpInputParam to the message <u>XML_ZoneAssociatedDetectorCfg</u> .

Composite Magnetic Contact Detector

Function	Description
Get configuration capability of composite magnetic contact detector	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET / <i>ISAPI/SecurityCP/Configuration/magneticContact/capabilities?format=json</i> , and the capability is returned in the message <i>JSON_MagneticContactCap</i> by <i>IpOutputParam</i> .
Get parameters of all composite magnetic contact detectors	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET / <i>ISAPI/SecurityCP/Configuration/magneticContact?format=json</i> , and the parameters are returned in the message <i>JSON_List_MagneticContact</i> by <i>IpOutputParam</i> .
Get parameters of the composite magnetic contact detector of a specific zone	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET / <i>ISAPI/SecurityCP/Configuration/magneticContact/zone/<ID>?format=json</i> , and the parameters are returned in the message <i>JSON_MagneticContact</i> by <i>IpOutputParam</i> .
Set parameters of the composite magnetic contact detector of a specific zone	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: PUT / <i>ISAPI/SecurityCP/Configuration/magneticContact/zone/<ID>?format=json</i> and set <i>IpInputParam</i> to the message <i>JSON_MagneticContact</i> .

Slim Magnetic Contact Detector

Function	Description
Get configuration capability of slim magnetic contact detector	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET / <i>ISAPI/SecurityCP/Configuration/slimMagneticContact/capabilities?format=json</i> , and the capability is returned in the message <i>JSON_SlimMagneticContactCap</i> by <i>IpOutputParam</i> .
Get parameters of all slim magnetic contact detectors	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET / <i>ISAPI/SecurityCP/Configuration/slimMagneticContact?format=json</i> , and the parameters are returned in the message <i>JSON_List_SlimMagneticContact</i> by <i>IpOutputParam</i> .
Get parameters of the slim magnetic contact detector of a specific zone	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET / <i>ISAPI/SecurityCP/Configuration/slimMagneticContact/zone/<ID>?format=json</i> , and the parameters are returned in the message <i>JSON_SlimMagneticContact</i> by <i>IpOutputParam</i> .
Set parameters of the slim magnetic contact detector of a specific zone	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: PUT / <i>ISAPI/SecurityCP/Configuration/slimMagneticContact/zone/</i>

Function	Description
	<u><ID>?format=json</u> and set IpInputParam to the message JSON_SlimMagneticContact .

PIR (Passive Infrared) Detector

Function	Description
Get configuration capability of PIR detector	Call NET_DVR_STDXMLConfig to transmit the request URI: GET / ISAPI/SecurityCP/Configuration/passiveInfraredDetector/capabilities?format=json , and the capability is returned in the message JSON_PassiveInfraredDetectorCap by IpOutputParam .
Get parameters of all PIR detectors	Call NET_DVR_STDXMLConfig to transmit the request URI: GET / ISAPI/SecurityCP/Configuration/passiveInfraredDetector?format=json , and the parameters are returned in the message JSON_List_PassiveInfraredDetector by IpOutputParam .
Get the PIR detector parameters of a specific zone	Call NET_DVR_STDXMLConfig to transmit the request URI: GET / ISAPI/SecurityCP/Configuration/passiveInfraredDetector/zone/<ID>?format=json , and the parameters are returned in the message JSON_PassiveInfraredDetector by IpOutputParam .
Set the PIR detector parameters of a specific zone	Call NET_DVR_STDXMLConfig to transmit the request URI: PUT / ISAPI/SecurityCP/Configuration/passiveInfraredDetector/zone/<ID>?format=json and set IpInputParam to the message JSON_PassiveInfraredDetector .

Curtain PIR (Passive Infrared) Detector

Function	Description
Get configuration capability of the curtain PIR detector	Call NET_DVR_STDXMLConfig to transmit the request URI: GET / ISAPI/SecurityCP/Configuration/curtainInfraredDetector/capabilities?format=json , and the capability is returned in the message JSON_CurtainInfraredDetectorCap by IpOutputParam .
Get parameters of all curtain PIR detectors	Call NET_DVR_STDXMLConfig to transmit the request URI: GET / ISAPI/SecurityCP/Configuration/curtainInfraredDetector?format=json , and the parameters are returned in the message JSON_List_CurtainInfraredDetector by IpOutputParam .
Get parameters of the curtain PIR detector of a specific zone	Call NET_DVR_STDXMLConfig to transmit the request URI: GET / ISAPI/SecurityCP/Configuration/curtainInfraredDetector/zone/

Function	Description
	<u><ID>?format=json</u> , and the parameters are returned in the message <u>JSON_CurtainInfraredDetector</u> by <u>IpOutputParam</u> .
Set parameters of the curtain PIR detector of a specific zone	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: PUT / <u>ISAPI/SecurityCP/Configuration/curtainInfraredDetector/zone/<ID>?format=json</u> and set <u>IpInputParam</u> to the message <u>JSON_CurtainInfraredDetector</u> .

Indoor Dual-Technology Detector

Function	Description
Get configuration capability of indoor dual-technology detector	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/SecurityCP/Configuration/indoorDualTechnologyDetector/capabilities?format=json</u> , and the capability is returned in the message <u>JSON_IndoorDualTechnologyDetectorCap</u> by <u>IpOutputParam</u> .
Get parameters of all indoor dual-technology detectors	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/SecurityCP/Configuration/indoorDualTechnologyDetector?format=json</u> , and the parameters are returned in the message <u>JSON_List_IndoorDualTechnologyDetector</u> by <u>IpOutputParam</u> .
Get parameters of the indoor dual-technology detector of a specific zone	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/SecurityCP/Configuration/indoorDualTechnologyDetector/zone/<ID>?format=json</u> , and the parameters are returned in the message <u>JSON_IndoorDualTechnologyDetector</u> by <u>IpOutputParam</u> .
Set parameters of the indoor dual-technology detector of a specific zone	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: PUT / <u>ISAPI/SecurityCP/Configuration/indoorDualTechnologyDetector/zone/<ID>?format=json</u> and set <u>IpInputParam</u> to the message <u>JSON_IndoorDualTechnologyDetector</u> .

Composite PIR (Passive Infrared) Glass-Break Detector

Function	Description
Get configuration capability of the composite PIR glass-break detector	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/SecurityCP/Configuration/glassBreakDetector/</u>

Function	Description
	<i>capabilities?format=json</i> , and the capability is returned in the message <i>JSON_GlassBreakDetectorCap</i> by IpOutputParam .
Get parameters of all composite PIR glass-break detectors	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET <i>/ISAPI/SecurityCP/Configuration/glassBreakDetector?format=json</i> , and the parameters are returned in the message <i>JSON_List_GlassBreakDetector</i> by IpOutputParam .
Get parameters of the composite PIR glass-break detector of a specific zone	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET <i>/ISAPI/SecurityCP/Configuration/glassBreakDetector/zone/<ID>?format=json</i> , and the parameters are returned in the message <i>JSON_GlassBreakDetector</i> by IpOutputParam .
Set parameters of the composite PIR glass-break detector of a specific zone	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: PUT <i>/ISAPI/SecurityCP/Configuration/glassBreakDetector/zone/<ID>?format=json</i> and set IpInputParam to the message <i>JSON_GlassBreakDetector</i> .

Pircam

The pircam is a kind of detector equipped with a camera which can capture pictures and record videos when the alarm is triggered.

Function	Description
Get pircam configuration capability	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET <i>/ISAPI/SecurityCP/Configuration/pircam/capabilities?format=json</i> , and the capability is returned in the message <i>JSON_PircamCap</i> by IpOutputParam .
Get pircam parameters of all zones	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET <i>/ISAPI/SecurityCP/Configuration/pircam/zone?format=json</i> , and the parameters are returned in the message <i>JSON_List_pircam</i> by IpOutputParam .
Get pircam parameters of a specific zone	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET <i>/ISAPI/SecurityCP/Configuration/pircam/zone/<ID>?format=json</i> , and the parameters are returned in the message <i>JSON_pircam</i> by IpOutputParam .
Set pircam parameters of a specific zone	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: PUT <i>/ISAPI/SecurityCP/Configuration/pircam/zone/<ID>?format=json</i> and set IpInputParam to the message <i>JSON_pircam</i> .

Function	Description
Get capability of exporting the picture captured by pircam	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET <i>/ISAPI/SecurityCP/FileExport/pircam/capabilities?format=json</i> , and the capability is returned in the message <i>JSON_FileExportCap</i> by <i>lpOutputParam</i> .
Export the picture captured by pircam	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: POST <i>/ISAPI/SecurityCP/FileExport/pircam?format=json</i> and set <i>lpInputParam</i> to the message <i>JSON_FileExportCond</i> . The URL of the exported picture is returned in the message <i>JSON_FileExportInfo</i> by <i>lpOutputParam</i> .

2.3.2 Zone Configuration

Zone is a basic concept in the security control panel system. It refers to a protection area in the system, and is regarded as the maximum recognizable unit to distinguish the alarm event. As a connection between the detector and the security control panel, it determines whether to trigger an alarm according to the resistance value of the alarm controller.

Function	Description
Get Zone Configuration Capability	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET <i>/ISAPI/SecurityCP/Configuration/zones/capabilities?format=json</i> . And the configuration capability is returned in the message <i>JSON_ZonesCap</i> by the output parameter (<i>lpOutputParam</i>).
Get All Zones' Parameters	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET <i>/ISAPI/SecurityCP/Configuration/zones?format=json</i> . And the configuration parameters are returned in the message <i>JSON_List_Zone</i> by the output buffer (<i>lpOutBuffer</i>) of the output parameter (<i>lpOutputParam</i>).
Get One Zone's Parameters	<ul style="list-style-type: none"> Option 1: Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET <i>/ISAPI/SecurityCP/Configuration/zones/<ID>?format=json</i> . And the zone's parameters are returned in the message <i>JSON_Zone</i> by the output parameter <i>lpOutputParam</i>. Option 2: Call <i>NET_DVR_GetDVRConfig</i> with "NET_DVR_GET_ALARMIN_PARAM_V50" (command No.: 1201). And the zone's parameters are returned in the structure <i>NET_DVR_ALARMIN_PARAM_V50</i> by the output parameter <i>lpOutBuffer</i>.

Function	Description
Set One Zone's Parameters	<ul style="list-style-type: none"> Option 1: Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: PUT <u>/ISAPI/SecurityCP/Configuration/zones/<ID>?format=json</u> , and set the input parameter IpInputParam to the message <u>JSON_Zone</u> . Option 2: Call <u>NET_DVR_SetDVRConfig</u> with "NET_DVR_SET_ALARMIN_PARAM_V50" (command No.: 1200) and set the input parameter IpInBuffer to the structure <u>NET_DVR_ALARMIN_PARAM_V50</u> .
Get Capability of Getting Unlinked Zones	<p>Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET <u>/ISAPI/SecurityCP/Configuration/notRelateZones/capabilities?format=json</u> .</p> <p>And the capability is returned in the message <u>JSON_NotRelateZonesCap</u> by the output parameter (IpOutputParam).</p>
Get Unlinked Zones	<p>Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET <u>/ISAPI/SecurityCP/Configuration/notRelateZones?format=json</u> .</p> <p>And the unlinked zones' parameters are returned in the message <u>JSON_NotRelateZones</u> by the output buffer (IpOutBuffer) of the output parameter (IpOutputParam).</p>

2.3.3 Partition Configuration

Partition, which is an independent control system of a security control panel, allows you to batch arm or disarm all zones in it. If the security control panel has two partitions, you have two independent systems for arming or disarming.

Partition

Function	Description
Get Partition Configuration Capability	<p>Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET <u>/ISAPI/SecurityCP/Configuration/subSys/capabilities?format=json</u> .</p> <p>And the configuration capability is returned in the message <u>JSON_SubSysCap</u> by the output parameter (IpOutputParam).</p>
Set Parameters of A Specific Partition	<p>Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: PUT <u>/ISAPI/SecurityCP/Configuration/subSys/<ID>?format=json</u> and set</p>

Function	Description
	the input buffer (IpInBuffer) of the input parameter (IpInputParam) to the message <i>JSON_SubSys</i> .
Get Parameters of All Partitions	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET / <i>ISAPI/SecurityCP/Configuration/subSys?format=json</i> . And the parameters are returned in the message <i>JSON_List_SubSys</i> by the output buffer (IpOutBuffer) of the output parameter (IpOutputParam).

Public Partition

Public partition is considered a special partition which can be shared by other partitions. It is usually applied to manage or control the public area related with other areas controlled by other partitions in one building. The public partition is armed automatically when all partitions linked with the public partition are in the arming status. The public partition is disarmed automatically when any of partitions linked with the public partition is in the disarming status. You can also arm or disarm the public partition independently.

Function	Description
Get Configuration Capability of Public Partition	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET / <i>ISAPI/SecurityCP/Configuration/publicSubSys/capabilities?format=json</i> . And the configuration capability is returned in the message <i>JSON_PublicSubSysCap</i> by the output parameter (IpOutputParam).
Set Parameters of A Specific Public Partition	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: PUT / <i>ISAPI/SecurityCP/Configuration/publicSubSys/<ID>?format=json</i> and set the input buffer (IpInBuffer) of the input parameter (IpInputParam) to the message <i>JSON_PublicSubSys</i> .
Get Parameters of All Public Partitions	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET / <i>ISAPI/SecurityCP/Configuration/publicSubSys?format=json</i> . And the parameters are returned in the message <i>JSON_List_PublicSubSys</i> by the output buffer (IpOutBuffer) of the output parameter (IpOutputParam).

2.3.4 Timer Configuration

The timer of security control panel or partition can be used to set arming or disarming schedule, control the enter delay or exit delay, and set the alarm delay time, audio warning duration, or heartbeat interval.

Partition Timer Configuration

Function	Description
Get Timer Configuration Capability	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/SecurityCP/Configuration/subSysTime/capabilities?format=json</u> And the capability is returned in the message <u>JSON_SubSysTimeCap</u> by the output parameter (lpOutputParam).
Get All Timers' Parameters	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/SecurityCP/Configuration/subSysTime?format=json</u> And the parameters are returned in the message of <u>JSON_List_SubSysTime</u> by the output buffer (lpOutBuffer) of the output parameter (lpOutputParam).
Set One Timer's Parameters	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: PUT / <u>ISAPI/SecurityCP/Configuration/subSysTime/<ID>?format=json</u> , and set the input buffer (lpInBuffer) of the input parameter (lpInputParam) to the message <u>JSON_SubSysTime</u> .

Security Control Panel Timer Configuration

Function	Description
Get Configuration Capability of Security Control Panel Timer	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/SecurityCP/Configuration/deviceTime/capabilities?format=json</u> . And the configuration capability is returned in the message <u>JSON_DeviceTimeCap</u> by the output parameter (lpOutputParam).
Get Parameters of Security Control Panel Timer	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/SecurityCP/Configuration/deviceTime?format=json</u> . And the parameters are returned in the message <u>JSON_DeviceTime</u> by the output buffer (lpOutBuffer) of the output parameter (lpOutputParam).
Set Parameters of Security Control Panel Timer	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: PUT / <u>ISAPI/SecurityCP/Configuration/deviceTime?format=json</u> and set the input buffer (lpInBuffer) of the input parameter (lpInputParam) to the message <u>JSON_DeviceTime</u> .

2.3.5 Siren Configuration

A siren is an output device that make a sound to alert people when alarm is triggered.

Function	Description
Get Siren Configuration Capability	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET <i>/ISAPI/SecurityCP/Configuration/wirelessSiren/capabilities?format=json</i> , and the capability is returned in the message of <i>JSON_SirenCap</i> by <i>lpOutputParam</i> .
Get All Sirens' Parameters	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET <i>/ISAPI/SecurityCP/Configuration/wirelessSiren?format=json</i> , and the configuration parameters are returned in the message of <i>JSON_List_Siren</i> by <i>lpOutputParam</i> .
Set One Siren's Parameters	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: PUT <i>/ISAPI/SecurityCP/Configuration/wirelessSiren/<ID>?format=json</i> , and set <i>lpInputParam</i> to the message <i>JSON_Siren</i> .
Get capability of controlling asynchronous siren adding mode	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET <i>/ISAPI/SecurityCP/Configuration/wirelessSiren/mode/capabilities?format=json</i> , and the capability is returned in the message <i>JSON_SirenModeCap</i> by <i>lpOutputParam</i> .
Control asynchronous siren adding mode	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: PUT <i>/ISAPI/SecurityCP/Configuration/wirelessSiren/mode?format=json</i> and set <i>lpInputParam</i> to the message <i>JSON_SirenMode</i> .
Get parameters of the currently added siren in asynchronous mode	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET <i>/ISAPI/SecurityCP/Configuration/wirelessSiren/currentAddAsyn?format=json</i> , and the parameters are returned in the message <i>JSON_Siren</i> by <i>lpOutputParam</i> .

2.3.6 Repeater Configuration

A repeater is an electronic device that receives a signal and retransmits it. It is used to extend transmissions so that the signal can cover longer distances or be received on the other side of an obstruction.

Function	Description
Get Repeater Configuration Capability	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET <i>/ISAPI/SecurityCP/Configuration/repeaters/capabilities?</i>

Function	Description
	<u><i>format=json</i></u> , and the capability is returned in the message of <u><i>JSON_RepeaterCap</i></u> .
Get All Repeaters' Parameters	Call <u><i>NET_DVR_STDXMLConfig</i></u> to transmit the request URI: GET / <u><i>ISAPI/SecurityCP/Configuration/repeaters?format=json</i></u> , and the configuration parameters are returned in the message of <u><i>JSON_List_Repeater</i></u> .
Set One Repeater's Parameters	Call <u><i>NET_DVR_STDXMLConfig</i></u> to transmit the request URI: PUT / <u><i>ISAPI/SecurityCP/Configuration/repeaters/<ID>?format=json</i></u> , and set the request message to <u><i>JSON_Repeater</i></u> .

2.3.7 Output Module Configuration

Here, the output module is a physical relay, an electrically operated switch. The terminal on the relay can be powered on or off immediately, which make normal close or normal open terminal pull in or out for opening or closing the door.

Function	Description
Get Output Module Configuration Capability	Call <u><i>NET_DVR_STDXMLConfig</i></u> to transmit the request URI: GET / <u><i>ISAPI/SecurityCP/Configuration/outputModules/capabilities?format=json</i></u> , and the capability is returned in the message of <u><i>JSON_OutputModuleCap</i></u> .
Get All Output Modules' Parameters	Call <u><i>NET_DVR_STDXMLConfig</i></u> to transmit the request URI: GET / <u><i>ISAPI/SecurityCP/Configuration/outputModules?format=json</i></u> , and the configuration parameters are returned in the message of <u><i>JSON_List_OutputModule</i></u> .
Set One Output Module's Parameters	Call <u><i>NET_DVR_STDXMLConfig</i></u> to transmit the request URI: PUT / <u><i>ISAPI/SecurityCP/Configuration/outputModules/<ID>?format=json</i></u> , and set the request message to <u><i>JSON_OutputModule</i></u> .

2.3.8 Relay Configuration

Here, the relay is a logical relay, which should be linked with the output module (physical module), and then you can control the output module via this relay.

Table 2-2 Relay Parameters

Function	Description
Get Relay Configuration Capability	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/SecurityCP/Configuration/outputs/capabilities?format=json</u> . And the configuration capability is returned in the message <u>JSON_OutputCap</u> by the output parameter (lpOutputParam).
Get All Relays' Parameters	<ul style="list-style-type: none"> Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/SecurityCP/Configuration/outputs?format=json</u> . And the parameters are returned in the message <u>JSON_List_Output</u> by the output buffer (lpOutBuffer) of the output parameter (lpOutputParam). 1. Call <u>NET_DVR_StartRemoteConfig</u> with "NET_DVR_GET_ALARMHOST_TRIGGER_LIST" (command No.: 2035) and set the input parameter (lpInBuffer) to <u>NET_DVR_LIST_INFO</u> for setting up the persistent connection and starting getting all relays' parameters. 2. Call <u>NET_DVR_SendRemoteConfig</u> to get search results one by one. The searched relay's parameters are returned in the structure <u>NET_DVR_ALARMOUT_PARAM</u> by the output pointer (lpOutBuff). 3. Call <u>NET_DVR_StopRemoteConfig</u> to stop getting all relays' parameters and disconnect the persistent connection.
Get Multiple Relays' Parameters by Specific Conditions	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: POST / <u>ISAPI/SecurityCP/Configuration/outputs?format=json</u> , and set the input buffer (lpInBuffer) of the input parameter (lpInputParam) to the message <u>JSON_OutputCond</u> .
Get One Relay's Parameters	Call <u>NET_DVR_GetDVRConfig</u> with the command "NET_DVR_GET_ALARMOUT_PARAM" (command No.: 1185) and set IChannel to the relay (trigger) No. (it starts from 0). The relay's parameters are returned in the structure <u>NET_DVR_ALARMOUT_PARAM</u> by the output parameter lpOutBuffer
Set One Relay's Parameters	<ul style="list-style-type: none"> Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: PUT / <u>ISAPI/SecurityCP/Configuration/outputs/<ID>?format=json</u> , and set the input buffer (lpInBuffer) of the input parameter (lpInputParam) to the message <u>JSON_Output</u> . Call <u>NET_DVR_SetDVRConfig</u> with the command "NET_DVR_SET_ALARMOUT_PARAM" (command No.: 1184), set IChannel to the

Function	Description
	relay (trigger) No. (it starts from 0), and set the input parameter IpInBuffer to the structure <u>NET_DVR_ALARMOUT_PARAM</u> .

Table 2-3 Relay Linkage Parameters (When the Relay is Closed/Open)

Function	Description
Get the relay's linkage configuration capability	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/SecurityCP/Configuration/outputsModule/capabilities?format=json</u> . The relay's linkage configuration capability is returned in the message <u>JSON_OutPutsModuleCap</u> by the output parameter (IpOutputParam).
Get linkage configuration parameters of all relays	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/SecurityCP/Configuration/outputsModule?format=json</u> . And the parameters are returned in the message <u>JSON_List_OutPutsModule</u> by the output buffer (IpOutBuffer) of the output parameter (IpOutputParam).
Search for linkage configuration parameters by condition	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: POST <u>/ISAPI/SecurityCP/Configuration/outputsModule?format=json</u> , and set the input buffer (IpInBuffer) of the input parameter (IpInputParam) to the message <u>JSON_OutputsModuleCond</u> .
Set relay's linkage configuration parameters	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: PUT / <u>ISAPI/SecurityCP/Configuration/outputsModule/<ID>?format=json</u> , and set the input buffer (IpInBuffer) of the input parameter (IpInputParam) to the message <u>JSON_OutPutsModule</u> .

2.3.9 Keyfob Configuration

The keyfob is an electronic key fob that ate used for controlling access.

Note

To check whether the device supports configuring keyfob, you can call **NET_DVR_GetDeviceAbility** , set the capability type **dwAbilityType** to "DEVICE_ABILITY_INFO" (macro definition value: 0x011), and set the input buffer (**pInBuf**) to **XML_Desc_AlarmHostAbility** for getting the network security control panel capability.


The capability is returned in the message **XML_AlarmHostAbility** by **pOutBuf**. If the node **<RemoteController>** is returned, it indicates that configuring keyfob is supported.

Basic Keyfob Settings

Function	Description
Get Keyfob Configuration Capability	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: <u>GET / ISAPI/SecurityCP/Configuration/remoteCtrl/capabilities?format=json</u> . And the configuration capability is returned in the message of <u>JSON_RemoteCtrlCap</u> by the output parameter (lpOutputParam).
Get Currently Added Keyfob's Parameters	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: <u>GET / ISAPI/SecurityCP/Configuration/remoteCtrl/currentAdd?format=json</u> . And the parameters of the currently added keyfob are returned in the message of <u>JSON_RemoteCtrl</u> .
Get One Keyfob's Parameters	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: <u>GET / ISAPI/SecurityCP/Configuration/remoteCtrl/<ID>?format=json</u> . And the parameters of a keyfob are returned in the message of <u>JSON_RemoteCtrl</u> .
Get All Keyfobs' Parameters	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: <u>GET / ISAPI/SecurityCP/Configuration/remoteCtrl?format=json</u> . And the parameters of all keyfobs are returned in the message <u>JSON_List_RemoteCtrl</u> by the output buffer (lpOutBuffer) of the output parameter (lpOutputParam).
Set One Keyfob's Parameters	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: <u>PUT / ISAPI/SecurityCP/Configuration/remoteCtrl/<ID>?format=json</u> , and set the input buffer (lpInBuffer) of the input parameter (lpInputParam) to the message <u>JSON_RemoteCtrl</u> .
Delete One Keyfob	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: <u>DELETE / ISAPI/SecurityCP/Configuration/remoteCtrl/<ID>?format=json</u> .

Keyfob User Settings

Funciton	Description
Get Keyfob User Parameters	Call <u>NET_DVR_GetDVRConfig</u> with "NET_DVR_GET_REMOTECONTROLLER_PERMISSION_CFG" (command No.: 2200).

Funciton	Description
	And the obtained keyfob user parameters are returned in the structure <u>NET_DVR_REMOTECONTROLLER_PERMISSION_CFG</u> by the output buffer (lpOutBuffer).
Set Keyfob User Parameters	Call <u>NET_DVR_SetDVRConfig</u> with "NET_DVR_SET_REMOTECONTROLLER_PERMISSION_CFG" (command No.: 2201) and set the input buffer (lpInBuffer) to the structure <u>NET_DVR_REMOTECONTROLLER_PERMISSION_CFG</u> .
Get All Keyfob Users	<ol style="list-style-type: none"> 1. Call <u>NET_DVR_StartRemoteConfig</u> with the command "NET_DVR_GET_ALL_REMOTECONTROLLER_LIST" (command No.: 2205) to get all keyfob users. 2. Call <u>NET_DVR_GetNextRemoteConfig</u> to get the keyfob user one by one. <p> Note</p> <p>The keyfob user information is returned in the structure <u>NET_DVR_REMOTECONTROLLER_PERMISSION_CFG</u>.</p> <ol style="list-style-type: none"> 3. Call <u>NET_DVR_StopRemoteConfig</u> to stop getting keyfob users and release resources.

Keyfob Adding

Function	Description
Get Capability of Controlling Asynchronous Keyfob Adding Mode	<p>Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: <u>GET /ISAPI/SecurityCP/Configuration/remoteCtrl/mode/capabilities?format=json</u>.</p> <p>And the capability is returned in the message <u>JSON_RemoteCtrlModeCap</u> by the output parameter (lpOutputParam).</p>
Control Asynchronous Keyfob Adding Mode	<p>Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: <u>PUT /ISAPI/SecurityCP/Configuration/remoteCtrl/mode?format=json</u> and set the input buffer (lpInBuffer) of the input parameter (lpInputParam) to the message <u>JSON_RemoteCtrlMode</u>.</p>
Get Asynchronously Added Kefob's Parameters	<p>Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: <u>GET /ISAPI/SecurityCP/Configuration/remoteCtrl/currentAddAsyn?format=json</u>.</p>

Function	Description
	And the parameters are returned in the message <i>JSON RemoteCtrl</i> by the output buffer (IpOutBuffer) of the output parameter (IpOutputParam).

2.3.10 Keypad Configuration

A keypad can be connected to the security control panel for programming and operating the security control panel.

Function	Description
Get Keypad Configuration Capability	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET / <i>ISAPI/SecurityCP/Configuration/keypad/capabilities?format=json</i> . And the configuration capability is returned in the message <i>JSON KeypadCap</i> by the output parameter (IpOutputParam).
Set Parameters of A Keypad	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: PUT / <i>ISAPI/SecurityCP/Configuration/keypad/<ID>?format=json</i> and set the input buffer (IpInBuffer) of the input parameter (IpInputParam) to the message <i>JSON Keypad</i> .
Get Parameters of All Keypads	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET / <i>ISAPI/SecurityCP/Configuration/keypad?format=json</i> . And the parameters are returned in the message <i>JSON List Keypad</i> by the output buffer (IpOutBuffer) of the output parameter (IpOutputParam).
Get Capability of Getting List of Keypads That Can Be Added	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET / <i>ISAPI/SecurityCP/Configuration/keypadAddList/capabilities?format=json</i> . And the capability is returned in the message <i>JSON KeypadAddListCap</i> by IpOutputParam .
Get List of Keypads That Can Be Added	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET / <i>ISAPI/SecurityCP/Configuration/keypadAddList?format=json</i> . And the list is returned in the message <i>JSON KeypadAddList</i> by IpOutBuffer of IpOutputParam .

2.3.11 Panic Button Configuration

The panic button can be pressed to send the alarm to the monitoring center when the emergency happens or someone asks for help.

Function	Description
Get configuration capability of panic button	Call <i>NET DVR STDXMLConfig</i> to transmit the request URI: GET / <i>ISAPI/SecurityCP/Configuration/panicButton/capabilities?format=json</i> , and the capability is returned in the message <i>JSON_PanicButtonCap</i> by <i>IpOutputParam</i> .
Get parameters of all panic buttons	Call <i>NET DVR STDXMLConfig</i> to transmit the request URI: GET / <i>ISAPI/SecurityCP/Configuration/panicButton?format=json</i> , and the parameters are returned in the message <i>JSON_List_PanicButton</i> by <i>IpOutputParam</i> .
Get panic button parameters of a specific button	Call <i>NET DVR STDXMLConfig</i> to transmit the request URI: GET / <i>ISAPI/SecurityCP/Configuration/panicButton/zone/<ID>?format=json</i> , and the parameters are returned in the message <i>JSON_PanicButton</i> by <i>IpOutputParam</i> .
Set panic button parameters of a specific button	Call <i>NET DVR STDXMLConfig</i> to transmit the request URI: PUT / <i>ISAPI/SecurityCP/Configuration/panicButton/zone/<ID>?format=json</i> and set <i>IpInputParam</i> to the message <i>JSON_PanicButton</i> .

2.3.12 Card Configuration

You can configure card number and name, and assign permissions to the card, and then you can arm and disarm by the card.

Function	Description
Get Card Configuration Capability	Call <i>NET DVR STDXMLConfig</i> to transmit the request URI: GET / <i>ISAPI/SecurityCP/Configuration/card/capabilities?format=json</i> . And the capability is returned in the message <i>JSON_CardCap</i> by the output parameter (<i>IpOutputParam</i>).
Get Currently Added Card's Parameters	Call <i>NET DVR STDXMLConfig</i> to transmit the request URI: GET / <i>ISAPI/SecurityCP/Configuration/card/currentAdd?format=json</i> . And the parameters of the currently added card are returned in the message <i>JSON_Card</i> by the output buffer (<i>IpOutBuffer</i>) of the output parameter (<i>IpOutputParam</i>).

Function	Description
Get One Card's Parameters	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/SecurityCP/Configuration/card/<ID>?format=json</u> . And the configuration parameters of a card are returned in the message <u>JSON_Card</u> by the output buffer (lpOutBuffer) of the output parameter (lpOutputParam).
Get All Cards' Parameters	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/SecurityCP/Configuration/card?format=json</u> . And the configuration parameters of all cards are returned in the message of <u>JSON_List_Card</u> by the output buffer (lpOutBuffer) of the output parameter (lpOutputParam).
Set One Card's Parameters	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: PUT / <u>ISAPI/SecurityCP/Configuration/card/<ID>?format=json</u> , and set the input buffer (lpInBuffer) of the input parameter (lpInputParam) to the message <u>JSON_Card</u> .
Delete One Card	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: DELETE / <u>ISAPI/SecurityCP/Configuration/card/<ID>?format=json</u> .
Get Capability of Controlling Asynchronous Card Adding Mode	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/SecurityCP/Configuration/card/mode/capabilities?format=json</u> . And the capability is returned in the message <u>JSON_Cap_CardMode</u> by the output parameter (lpOutputParam).
Control Asynchronous Card Adding Mode	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: PUT / <u>ISAPI/SecurityCP/Configuration/card/mode?format=json</u> and set the input buffer (lpInBuffer) of the input parameter (lpInputParam) to the message <u>JSON_CardMode</u> .
Get Asynchronously Added Card's Parameters	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/SecurityCP/Configuration/card/currentAddAsyn?format=json</u> . And the parameters are returned in the message <u>JSON_Card</u> by the output buffer (lpOutBuffer) of the output parameter (lpOutputParam).

2.3.13 Card Reader Configuration

A card reader in the security control system is used to read the code embedded in a card. The security control panel will determine whether to allow the user to access the area, arm or disarm the system according to the data uploaded by swiping the card.

Function	Description
Get Card Reader Configuration Capability	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/SecurityCP/Configuration/cardReader/capabilities?format=json</u> . And the configuration capability is returned in the message <u>JSON_CardReaderCap</u> by the output parameter (lpOutputParam).
Set Parameters of A Card Reader	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: PUT / <u>ISAPI/SecurityCP/Configuration/cardReader/<ID>?format=json</u> and set the input buffer (lpInBuffer) of the input parameter (lpInputParam) to the message <u>JSON_CardReader</u> .
Get Parameters of All Card Readers	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/SecurityCP/Configuration/cardReader?format=json</u> . And the parameters are returned in the message <u>JSON_List_CardReader</u> by the output buffer (lpOutBuffer) of the output parameter (lpOutputParam).
Get capability of controlling asynchronous card reader parameters adding mode	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/SecurityCP/Configuration/cardReader/mode/capabilities?format=json</u> , and the capability is returned in the message <u>JSON_Cap_CardReaderMode</u> by lpOutputParam .
Control asynchronous card reader parameters adding mode	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: PUT / <u>ISAPI/SecurityCP/Configuration/cardReader/mode?format=json</u> and set lpInputParam to the message <u>JSON_CardReaderMode</u> .
Get parameters of the currently added card reader in asynchronous mode	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/SecurityCP/Configuration/cardReader/currentAddAsyn?format=json</u> , and the parameters are returned in the message <u>JSON_CardReader</u> by lpOutputParam .

2.3.14 Extension Module Configuration

An extension module, such as wired or wireless output module, wireless receiver, wired zone, and so on, can be connected to a security control panel to extend functions.

Function	Description
Get Configuration Capability of Extension Module	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET / <i>ISAPI/SecurityCP/Configuration/extensionModule/capabilities?format=json</i> . And the configuration capability is returned in the message <i>JSON_ExtensionModuleCap</i> by <i>lpOutputParam</i> .
Set Parameters of A Specific Extension Module	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: PUT / <i>ISAPI/SecurityCP/Configuration/extensionModule/<ID>?format=json</i> and set <i>lpInBuffer</i> of <i>lpInputParam</i> to the message <i>JSON_ExtensionModule</i> .
Get Parameters of All Extension Modules	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET / <i>ISAPI/SecurityCP/Configuration/extensionModule?format=json</i> . And the parameters are returned in the message <i>JSON_List_ExtensionModule</i> by <i>lpOutBuffer</i> of <i>lpOutputParam</i> .

2.3.15 Module Locking and Unlocking

You can set the module locking parameters to lock the module after the maximum failure attempts. You can also unlock the module as needed.

Lock Module

Function	Description
Get Capability of Locking Module	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET / <i>ISAPI/System/moduleLock/config/capabilities?format=json</i> . And the capability is returned in the message <i>JSON_ModuleLockCap</i> by the output parameter (<i>lpOutputParam</i>).
Get Parameters of Locking Module	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET / <i>ISAPI/System/moduleLock/config?format=json</i> . And the parameters are returned in the message <i>JSON_List_ModuleLock</i> by the output buffer (<i>lpOutBuffer</i>) of the output parameter (<i>lpOutputParam</i>).
Set Parameters of Locking Module	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: PUT / <i>ISAPI/System/moduleLock/config?format=json</i> and set the input buffer (<i>lpInBuffer</i>) of the input parameter (<i>lpInputParam</i>) to the message <i>JSON_List_ModuleLock</i> .

Unlock Module

Call **NET_DVR_STDXMLConfig** to transmit the request URI: PUT **/ISAPI/System/moduleLock/unlockModule?format=json** and set the input buffer (**IpInBuffer**) of the input parameter (**IpInputParam**) to the message **JSON_List_ModuleInfo** .

2.3.16 Alarm Lamp Configuration

An alarm lamp is an output device that flickers to alert people when alarm is triggered. You can configure schedules to enable the alarm lamp flickering when the alarm is triggered, and you can also set the duration and interval of the alarm lamp flickering.

Alarm Lamp Settings

Function	Description
Get Alarm Lamp Configuration Capability	Call <u>NET_DVR_GetSTDAbility</u> and set dwAbilityType to "NET_DVR_GET_ALARM_LAMP_CFG_CAP" (macro definition value: 2216). The configuration capability is returned in the message <u>XML_Cap_AlarmLampConfig</u> by IpOutBuffer of the structure <u>NET_DVR_STD_ABILITY</u> .
Get Alarm Lamp Parameters	Call <u>NET_DVR_GetSTDConfig</u> with "NET_DVR_GET_ALARM_LAMP_CFG" (command No.: 2217). The parameters are returned in the structure <u>NET_DVR_ALARM_LAMP_CFG</u> by IpOutBuffer of the structure <u>NET_DVR_STD_CONFIG</u> .
Set Alarm Lamp Parameters	Call <u>NET_DVR_SetSTDConfig</u> with "NET_DVR_SET_ALARM_LAMP_CFG" (command No.: 2218) and set IpInBuffer of the structure <u>NET_DVR_STD_CONFIG</u> to <u>NET_DVR_ALARM_LAMP_CFG</u> .



Note

To check whether the device supports configuring alarm lamp parameters, you can call **NET_DVR_GetSTDAbility** and set **dwAbilityType** to "NET_DVR_GET_EMERGENCY_ALARM_PRODUCT_CAP" (macro definition value: 2212) for getting the capability of the one-touch panic alarm product.

The capability is returned in the message **XML_EmergencyAlarmProductCap** by **IpOutBuffer** of the structure **NET_DVR_STD_ABILITY** . If this function is supported, the node **<alarmLampConfig>** will be returned and its value is true.

Alarm Lamp Flickering Schedule

Function	Description
Get Configuration Capability of Alarm Lamp Flickering Schedule	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URL: GET <u>/ISAPI/SecurityCP/BasicParam/AlarmLampSchedTimeConfig/capabilities</u> . The configuration capability is returned in the message <u>XML_Cap_LampSchedTimeList</u> by the output parameter IpOutputParam .
Get Parameters of Alarm Lamp Flickering Schedule	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URL: GET <u>/ISAPI/SecurityCP/BasicParam/AlarmLampSchedTimeConfig</u> . The parameters are returned in the message <u>XML_LampSchedTimeList</u> by the output parameter IpOutputParam .
Set Parameters of Alarm Lamp Flickering Schedule	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URL: PUT <u>/ISAPI/SecurityCP/BasicParam/AlarmLampSchedTimeConfig</u> and set the input parameter IpInputParam to the message <u>XML_LampSchedTimeList</u> .

2.3.17 Access Module Configuration

Function	Description
Get the capability of adding zones, relays, or sirens to the access module according to the access module type	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: POST <u>/ISAPI/SecurityCP/Configuration/accessModule/addType/capabilities?format=json</u> and set IpInputParam to the message <u>JSON_accessModuleType</u> . The capability is returned in the message <u>JSON_AccessModuleAddTypeCap</u> by IpOutputParam .
Set parameters for adding zones, relays, or sirens to the access module	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: PUT <u>/ISAPI/SecurityCP/Configuration/accessModule/addType?format=json</u> and set IpInputParam to the message <u>JSON_accessModuleAddType</u> . The returned information after adding a zone/peripheral is in the message <u>JSON_accessModuleAddResult</u> by IpOutputParam .

2.4 Alarm and Event

You can configure different notification objects, i.e., alarm center (which can be configured), Hik-Connect, phone, and email, for events or alarms, and you can also receive the events or alarm in arming via the platform or system., and the received types of events or alarms can be subscribed.

- For configuring different notification objects, refer to **Alarm/Event Notification** .
- For receiving and subscribing events or alarms of security control panel,
 - You can set the command (**ICommand**) in the alarm callback function to "COMM_ISAPI_ALARM" (command No: 0x6009). And the alarm/event details are returned in the message **JSON_EventNotificationAlert_SecurityCPAlarmEventMsg** by the structure **NET_DVR_ALARM_ISAPI_INFO** .
 - You can set the command (**ICommand**) in the alarm callback function to "COMM_ALARMHOST_CID_ALARM" (command No: 0x1127). And the alarm/event details are returned in the structure **NET_DVR_CID_ALARM** .



Note

For detailed API calling flow of receiving and subscribing events or alarms, refer to chapters *Receive Alarm/Event in Arming Mode* and *Subscribe Alarm/Event in Arming Mode* in *User Manual of Device Network SDK (General)*.

2.4.1 Alarm/Event Notification

When the events or alarms of security control panel occurred or are triggered, you can choose to notify the events or alarms to different objects for different processes. The following objects are available: alarm center, Hik-Connect, phone, and email.

Alarm Center Notification

For Arming Mode

Arming mode refers to an passive method, that is, the platform connects to device automatically, when the alarm is triggered or event occurred, the platform sends uploading command to the device, and then the device will upload the alarm or event to the platform.

Function	Description
Get Configuration Capability of Alarm Center Notification	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/SecurityCP/Configuration/messageSendDirect/capabilities?format=json</u> , and the capability is returned in the message of <u>JSON_DirectCap</u> .
Get Parameters of Alarm Center Notification	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/SecurityCP/Configuration/messageSendDirect?format=json</u> ,

Function	Description
	and the notification configuration parameters are returned in the message of <i>JSON_Direct</i> .
Set Parameters of Alarm Center Notification	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: PUT <i>/ISAPI/SecurityCP/Configuration/messageSendDirect?format=json</i> , and set the request message to <i>JSON_Direct</i> .

For Listening Mode

Listening mode refers to an active method, that is, when alarm is triggered or event occurred, the device automatically uploads the alarm, and then the platform receives the uploaded alarm via the configured listening host (listening address and port should be configured).

Function	Description
Get Configuration Capability of Alarm Center Notification	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET <i>/ISAPI/SecurityCP/Configuration/messageSendARC/capabilities?format=json</i> , and the capability is returned in the message of <i>JSON_SendARCCap</i> .
Get Parameters of Alarm Center Notification	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET <i>/ISAPI/SecurityCP/Configuration/messageSendARC?format=json</i> , and the notification configuration parameters are returned in the message of <i>JSON_SendARC</i> .
Set Parameters of Alarm Center Notification	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: PUT <i>/ISAPI/SecurityCP/Configuration/messageSendARC?format=json</i> , and set the request message to <i>JSON_SendARC</i> .

Note

Before configuring alarm center notification, you should configure the alarm center, refer to ***Alarm Center Configuration*** for details.

Hik-Connect Notification

Function	Description
Get Configuration Capability of Hik-Connect Notification	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET <i>/ISAPI/SecurityCP/Configuration/messageSendCloud/capabilities?</i>

Function	Description
	<u><i>format=json</i></u> , and the capability is returned in the message of <u><i>JSON_CloudCap</i></u> .
Get Hik-Connect Notification Parameters	Call <u><i>NET_DVR_STDXMLConfig</i></u> to transmit the request URI: GET / <u><i>ISAPI/SecurityCP/Configuration/messageSendCloud?format=json</i></u> , and the notification configuration parameters are returned in the message of <u><i>JSON_Cloud</i></u> .
Set Hik-Connect Notification Parameters	Call <u><i>NET_DVR_STDXMLConfig</i></u> to transmit the request URI: PUT / <u><i>ISAPI/SecurityCP/Configuration/messageSendCloud?format=json</i></u> , and set the request message to <u><i>JSON_Cloud</i></u> .

Phone Notification

Function	Description
Get Configuration Capability of Phone Notification	Call <u><i>NET_DVR_STDXMLConfig</i></u> to transmit the request URI: GET / <u><i>ISAPI/SecurityCP/Configuration/messageSendPhone/capabilities?format=json</i></u> , and the capability is returned in the message of <u><i>JSON_PhoneCap</i></u> .
Get Phone Notification Parameters	Call <u><i>NET_DVR_STDXMLConfig</i></u> to transmit the request URI: GET / <u><i>ISAPI/SecurityCP/Configuration/messageSendPhone?format=json</i></u> , and the notification configuration parameters are returned in the message of <u><i>JSON_List_Phone</i></u> .
Set Parameters of One Phone Notification	Call <u><i>NET_DVR_STDXMLConfig</i></u> to transmit the request URI: PUT / <u><i>ISAPI/SecurityCP/Configuration/messageSendPhone/<ID>?format=json</i></u> , and set the request message to <u><i>JSON_Phone</i></u> .
Get Advanced Configuration Capability of Phone Notification	Call <u><i>NET_DVR_STDXMLConfig</i></u> to transmit the request URI: GET / <u><i>ISAPI/SecurityCP/Configuration/messageSendPhoneAdvanced/capabilities?format=json</i></u> . And the capability is returned in the message <u><i>JSON_PhoneAdvancedCap</i></u> by the output parameter (lpOutputParam).
Get Advanced Configuration Parameters of Phone Notification	Call <u><i>NET_DVR_STDXMLConfig</i></u> to transmit the request URI: GET / <u><i>ISAPI/SecurityCP/Configuration/messageSendPhoneAdvanced?format=json</i></u> . And the parameters are returned in the message <u><i>JSON_List_PhoneAdvanced</i></u> by the output buffer (lpOutBuffer) of the output parameter (lpOutputParam).

Function	Description
Set Advanced Configuration Parameters of Phone Notification	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: PUT / <u>ISAPI/SecurityCP/Configuration/messageSendPhoneAnvanced/<ID>?format=json</u> and set the input buffer (lpInBuffer) of the input parameter (lpInputParam) to the message <u>JSON_PhoneAnvanced</u> .
Get Configuration Capability of Phone Notification via PSTN (Public Switched Telephone Network)	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/SecurityCP/Configuration/PSTNCfg/capabilities?format=json</u> . And the configuration capability is returned in the message <u>JSON_PSTNCfgCap</u> by lpOutputParam .
Get Parameters of All Phone Notifications via PSTN (Public Switched Telephone Network)	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/SecurityCP/Configuration/PSTNCfg?format=json</u> . And the parameters are returned in the message <u>JSON_List_PSTNCfg</u> by lpOutBuffer of lpOutputParam .
Set Parameters of A Specific Phone Notification via PSTN (Public Switched Telephone Network)	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: PUT / <u>ISAPI/SecurityCP/Configuration/PSTNCfg/<ID>?format=json</u> and set lpInBuffer of lpInputParam to the message <u>JSON_PSTNCfg</u> .

Email Notification

Function	Description
Get Configuration Capability of Email Notification	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/SecurityCP/Configuration/messageSendMail/capabilities?format=json</u> , and the capability is returned in the message of <u>JSON_MailCap</u> .
Get Email Notification Parameters	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/SecurityCP/Configuration/messageSendMail?format=json</u> , and the notification configuration parameters are returned in the message of <u>JSON_List_Mail</u> .
Set Email Notification Parameters	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: PUT / <u>ISAPI/SecurityCP/Configuration/messageSendMail/<ID>?format=json</u> , and set the request message to <u>JSON_Mail</u> .

Zone Alarms Filtering

Function	Description
Get Configuration Capability of Filtering Duplicate Zone Alarms in the Configured Time Interval	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET / <i>ISAPI/SecurityCP/Configuration/zoneAlarmTimeFilter/capabilities?format=json</i> . The configuration capability is returned in the message <i>JSON_TimeCfgCap</i> by <i>lpOutBuffer</i> of <i>lpOutputParam</i> .
Get Parameters of Filtering Duplicate Zone Alarms in the Configured Time Interval	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET / <i>ISAPI/SecurityCP/Configuration/zoneAlarmTimeFilter?format=json</i> . The parameters are returned in the message <i>JSON_TimeCfg</i> by <i>lpOutBuffer</i> of <i>lpOutputParam</i> .
Set Parameters to Filter Duplicate Zone Alarms in the Configured Time Interval	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: PUT / <i>ISAPI/SecurityCP/Configuration/zoneAlarmTimeFilter?format=json</i> and set <i>lpInBuffer</i> of <i>lpInputParam</i> to the message <i>JSON_TimeCfg</i> .

Alarm Center Configuration

The alarm center is for receiving the events or alarms of security control panel and handling the alarms. Before notifying the events or alarms to alarm center, you must configure it.


Function	Description
Get Configuration Capability	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET / <i>ISAPI/SecurityCP/Configuration/ARC/capabilities?format=json</i> , and the capability is returned in the message of <i>JSON_ARCCap</i> .
Get Multiple Alarm Centers' Parameters	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET / <i>ISAPI/SecurityCP/Configuration/ARC?format=json</i> , and the configuration parameters are returned in the message of <i>JSON_List_ARC</i> .
Get An Alarm Center' Parameters	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET / <i>ISAPI/SecurityCP/Configuration/ARC/<ID>?format=json</i> , and the configuration parameters are returned in the message of <i>JSON_ARC</i> .
Set An Alarm Center's Parameters	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: PUT / <i>ISAPI/SecurityCP/Configuration/ARC/<ID>?format=json</i> , and set the request message to <i>JSON_ARC</i> .

Function	Description
Get Dial-Up Parameters of Security Control Panel	Call <i>NET_DVR_GetDVRConfig</i> with "NET_DVR_GET_ALARMMDIALMODECFG" (command No.: 1198) and set IChannel to 0. The parameters are returned in the structure <i>NET_DVR_ALARMHOSTDIALCFG</i> by IpOutBuffer .
Set Dial-Up Parameters of Security Control Panel	Call <i>NET_DVR_SetDVRConfig</i> with "NET_DVR_SET_ALARMMDIALMODECFG" (command No.: 1199), set IChannel to 0, and set IpInBuffer to the structure <i>NET_DVR_ALARMHOSTDIALCFG</i> .
Get the capability of ARC manual test	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET / <i>ISAPI/SecurityCP/Configuration/ARC/manualTest/capabilities?format=json</i> . And the capability is returned in the message <i>JSON_ARCManualTestCap</i> by IpOutputParam .
Set ACR manual test parameters	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: PUT / <i>ISAPI/SecurityCP/Configuration/ARC/manualTest?format=json</i> and set IpInBuffer of IpInputParam to the message <i>JSON_ARCManualTest</i> .
Get the manual test status of a single ARC	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: POST / <i>ISAPI/SecurityCP/Configuration/ARC/manualTest/status?format=json</i> and set IpInputParam to the message <i>JSON_ARCManualTestID</i> . The status is returned in the message <i>JSON_ARCManualTestStatus</i> by IpOutputParam .

2.4.2 Data Uploading

The alarm or event data from devices or detectors should be uploaded to control panel for management, such as triggering alarm, notifying to alarm center, and so on. You can set the uploading method and related parameters.

Function	Description
Get Data Uploading Parameters	Call <i>NET_DVR_GetDeviceConfig</i> with "NET_DVR_GET_ALARMHOST_REPORT_CENTER_V40" (command No.: 2064). And the data uploading parameters are returned in the structure of <i>NET_DVR_ALARMHOST_REPORT_CENTER_CFG_V40</i> .

Function	Description
	 Note For the security control panel in version 2.0 and lower, the value of parameter dwCount in the API <u>NET_DVR_GetDeviceConfig</u> is specified as the centers' number returned by the capability, and the parameter lpStatusList is invalid.
Set Data Uploading Parameters	Call <u>NET_DVR_SetDeviceConfig</u> with "NET_DVR_SET_ALARMHOST_REPORT_CENTER_V40" (command No.: 2065) and set the lpInParamBuffer to <u>NET_DVR_ALARMHOST_REPORT_CENTER_CFG_V40</u> .

2.5 Control and Operation

You can arming or disarm the partitions in the zones, perform bypass or bypass recovered on the zones of security control panels, clear alarms, and control the relays via different integration methods, i.e., calling API with command and calling API to transmit request URI with messages.

Function	Description	
Get Operation and Control Capability	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: <u>/ISAPI/SecurityCP/control/capabilities?format=json</u> , and the capability is returned in the response message of <u>JSON_HostControlCap</u> .	
Zone Bypass and Bypass Recovered	Bypass	Call <u>NET_DVR_BypassAlarmChan</u> and set the input parameter pointer (lpInter) to the structure <u>NET_DVR_ALARMIN_SETUP</u> .
		Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: PUT <u>/ISAPI/SecurityCP/control/bypass/<ID>?format=json</u> . This operation is for a single zone.
		Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: PUT <u>/ISAPI/SecurityCP/control/bypass?format=json</u> , and set the request message to <u>JSON_List_ID</u> . This operation is for multiple zones.
	Bypass Recovered	Call <u>NET_DVR_UnBypassAlarmChan</u> and set the input parameter pointer (lpInter) to the structure <u>NET_DVR_ALARMIN_SETUP</u> . Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: PUT <u>/ISAPI/SecurityCP/control/bypassRecover/</u>

Function	Description	
Arm and Disarm Partition		<u><ID>?format=json</u> . This operation is for a single zone.
		Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: PUT <u>/ISAPI/SecurityCP/control/bypassRecover?format=json</u> , and set the request message to <u>JSON_List_ID</u> . This operation is for multiple zones.
	Arm	(Recommended) Call <u>NET_DVR_RemoteControl</u> with the command of NET_DVR_ARM_ALARMHOST_SUBSYSTEM (command No.: 2036), and set the input parameter pointer (<u>lpInBuffer</u>) to the structure of <u>NET_DVR_CONTROL_PARAM</u> for stay arming, instant arming, or away arming.
		Call <u>NET_DVR_AlarmHostSubSystemSetupAlarmChan</u> .
		Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: PUT <u>/ISAPI/SecurityCP/control/arm/<ID>?ways=<string>&format=json</u> , and set the query parameter (<u>ways</u>) to "stay" or "away".
	Disarm	(Recommended) Call <u>NET_DVR_RemoteControl</u> with the command of NET_DVR_ALARMHOST_CLOSE_SUBSYSTEM (command No.: 2082), and set the input parameter pointer (<u>lpInBuffer</u>) to the structure of <u>NET_DVR_CONTROL_PARAM</u> .
		Call <u>NET_DVR_AlarmHostSubSystemCloseAlarmChan</u> .
		Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: PUT <u>/ISAPI/SecurityCP/control/disarm/<ID>?format=json</u> .
Clear Alarm	Call <u>NET_DVR_AlarmHostClearAlarm</u> .	
	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: PUT <u>/ISAPI/SecurityCP/control/clearAlarm/<ID>?format=json</u> .	
Control Relay	Call <u>NET_DVR_SetAlarmHostOut</u> .	
	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: PUT <u>/ISAPI/SecurityCP/control/outputs/<ID>?</u>	

Function		Description
		<u>format=json</u> , and set the request message to <u>JSON_OutputsCtrl</u> . This operation is for a single relay.
		Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: POST <u>/ISAPI/SecurityCP/control/outputs?format=json</u> , and set the request message to <u>JSON_OutputsCtrl</u> . This operation is for multiple relays.
Control Siren		Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: POST <u>/ISAPI/SecurityCP/control/siren/<ID>?format=json</u> , and set lpInBuffer of lpInputParam to the message <u>JSON_SirenCtrl</u> .
Pircam Capture	Get Picture Captured by Pircam in Synchronous Mode	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET <u>/ISAPI/SecurityCP/pircam/picture/channels/<ID>?format=json</u> , and the URI of the picture captured by the pircam is returned in the message <u>JSON_Picture</u> by lpOutputParam .
	Get Capability of Controlling Pircam to Capture Pictures or Record Videos in Asynchronous Mode	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET <u>/ISAPI/SecurityCP/pircam/picture/mode/capabilities?format=json</u> , and the capability is returned in the message <u>JSON_Cap_PircamMode</u> by lpOutputParam .
	Control Pircam to Capture Pictures in Asynchronous Mode	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: PUT <u>/ISAPI/SecurityCP/pircam/picture/channels/<ID>/mode?format=json</u> and set lpInputParam to the message <u>JSON_PircamMode</u> .
	Get Pircam Capture Parameters Being Added Currently in Asynchronous Mode	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET <u>/ISAPI/SecurityCP/pircam/picture/channels/<ID>/currentAddAsyn?format=json</u> , and the parameters are returned in the message <u>JSON_Pircam</u> by lpOutputParam .

2.6 Status Monitoring

Different integration methods, i.e., calling API with command and calling API to transmit request URI with messages, are provided to get the status of security control panels, peripherals, zones, partitions, storage battery, and communication.

Function	Description
Get Capability of Getting Security Control Panels' Status	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/SecurityCP/status/capabilities?format=json</u> , the capability is returned the message of <u>JSON_HostStatusCap</u> by lpOutputParam .
Get All Statuses of Security Control Panel	Call <u>NET_DVR_GetDVRConfig</u> with the command of "NET_DVR_GET_ALARMHOST_MAIN_STATUS_V51" (command No.: 2083), and all statuses of the security control panel are returned in the output parameter (lpOutBuffer) by the structure of <u>NET_DVR_ALARMHOST_MAIN_STATUS_V51</u> . Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/SecurityCP/status/host?format=json</u> , and all statuses of the security control panel are returned in the message of <u>JSON_AlarmHostStatus</u> by the output parameter lpOutputParam .
Get Status of Security Control Panel Itself	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/SecurityCP/status/hostItself?format=json</u> , and the status information is returned in the message <u>JSON_HostStatus</u> by lpOutputParam .
Get Peripherals' Status	Call <u>NET_DVR_GetDVRConfig</u> with the command of "NET_DVR_GET_ALARMHOST_OTHER_STATUS_V51" (command No.: 2236), the status of all peripherals is returned in the output parameter (lpOutBuffer) by the structure of <u>NET_DVR_ALARMHOST_OTHER_STATUS_V51</u> . Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/SecurityCP/status/exDevStatus?format=json</u> , the status of all peripherals is returned in the message of <u>JSON_ExDevStatus</u> by lpOutputParam .
Get Zones' Status	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/SecurityCP/status/zones?format=json</u> , the status of all zones is returned in the message of <u>JSON_ZoneList</u> by lpOutputParam .
Get Zones' Status by Specific Conditions	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: POST / <u>ISAPI/SecurityCP/status/zones?format=json</u> and set the input buffer (lpInBuffer) of the input parameter (lpInputParam) to the message <u>JSON_ZoneCond</u> .
Get Partitions' Status	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/SecurityCP/status/subSystems?format=json</u> , the status of all partitions is returned in the message of <u>JSON_SubSysList</u> by lpOutputParam .

Function	Description
Get Storage Battery's Voltage Status	Call <u>NET_DVR_GetBatteryVoltage</u> , the battery voltage status is returned by the parameter pVoltage .
	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/SecurityCP/status/batteries?format=json</u> , the battery voltage status is returned in the message of <u>JSON_BatteryList</u> by IpOutputParam .
Get AC Power Supply Status	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/SecurityCP/status/acPowerStatus?format=json</u> , and the status information is returned in the message <u>JSON_ACPowerStatus</u> by IpOutputParam .
Get Communication Status	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/SecurityCP/status/communication?format=json</u> , the communication status is returned in the message of <u>JSON_CommuniStatus</u> by IpOutputParam .
Get Output Module Status	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/SecurityCP/status/outputModStatus?format=json</u> . And the parameters are returned in the message <u>JSON_OutputModList</u> by the output buffer (IpOutBuffer) of the output parameter (IpOutputParam).
Get Relay Status by Specific Conditions	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: POST / <u>ISAPI/SecurityCP/status/outputStatus?format=json</u> and set the input buffer (IpInBuffer) of the input parameter (IpInputParam) to the message <u>JSON_OutputCond</u> .
Get Siren Status	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/SecurityCP/status/sirenStatus?format=json</u> . And the parameters are returned in the message <u>JSON_SirenList</u> by the output buffer (IpOutBuffer) of the output parameter (IpOutputParam).
Get Repeater Status	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/SecurityCP/status/repeaterStatus?format=json</u> . And the parameters are returned in the message <u>JSON_RepeaterList</u> by the output buffer (IpOutBuffer) of the output parameter (IpOutputParam).
Get Card Reader Status	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/SecurityCP/status/cardReaderStatus?format=json</u> .

Function	Description
	And the parameters are returned in the message <i>JSON_CardReaderList</i> by the output buffer (IpOutBuffer) of the output parameter (IpOutputParam).
Get Extension Module Status	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET / <i>ISAPI/SecurityCP/status/extensionModuleStatus?format=json</i> . And the parameters are returned in the message <i>JSON_ExtensionList</i> by the output buffer (IpOutBuffer) of the output parameter (IpOutputParam).
Get Keypad Status	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET / <i>ISAPI/SecurityCP/status/keypadStatus?format=json</i> . And the parameters are returned in the message <i>JSON_KeypadList</i> by the output buffer (IpOutBuffer) of the output parameter (IpOutputParam).

2.7 Capture and Recording

For some remarkable views, you can record the video segments and save the captured pictures or videos to the configured storage.

Capture

Function	Description
Get Parameters for Capturing Alarm Pictures	Call <i>NET_DVR_GetDVRConfig</i> with "NET_DVR_GET_ALARM_CAPTRUE_CFG" (command No.: 2074) and set IChannel to the channel No. The parameters are returned in the structure <i>NET_DVR_ALARM_CAPTRUE_CFG</i> by IpOutBuffer .
Set Parameters for Capturing Alarm Pictures	Call <i>NET_DVR_SetDVRConfig</i> with "NET_DVR_SET_ALARM_CAPTRUE_CFG" (command No.: 2075), set IChannel to the channel No., and set IpInBuffer to the structure <i>NET_DVR_ALARM_CAPTRUE_CFG</i> .

Recording

The pre-recorded and post-recorded time of recording based on event can be configured if the camera is added to the security control panel by calling API to transmit the request URIs with messages.

Function	Description
Get Capability of Recording Based on Event Configuration	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/SecurityCP/Configuration/eventRecord/channels/<ID>/capabilities?format=json</u> . And the capability is returned in the message of <u>JSON_EventRecordCap</u> .
Get Parameters of Recording Based on Event	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET / <u>ISAPI/SecurityCP/Configuration/eventRecord/channels/<ID>?format=json</u> . And the configuration parameters are returned in the message of <u>JSON_EventRecord</u> .
Set Parameters of Recording Based on Event	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: PUT / <u>ISAPI/SecurityCP/Configuration/eventRecord/channels/<ID>?format=json</u> and set the input buffer (IpInBuffer) of the input parameter (IpInputParam) to the message <u>JSON_EventRecord</u> .

2.8 Maintenance

You can call API to transmit request URIs with messages for detection and maintenance of the security control system, such as log search, fault detection, video and audio detection, and so on.

Log Search

Function	Description
Search for Security Control Panel's Logs	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: POST <u>/ISAPI/SecurityCP/Log/search?format=json</u> , and set IpInputParam to the message <u>JSON_SearchDescription</u> . The search results are returned in the message <u>JSON_SearchResult</u> by IpOutputParam .
	<ol style="list-style-type: none"> 1. Call <u>NET_DVR_FindAlarmHostLog</u> to search for security control panel logs. 2. Call <u>NET_DVR_FindNextAlarmHostLog</u> to search for the next security control panel log, i.e., get the searched log information of the security control panel one by one.

Fault Detection

Function		Description
Acknowledge System Faults		Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: PUT <u>/ISAPI/SecurityCP/control/systemFault?format=json</u> and set the input buffer (IpInBuffer) of the input parameter (IpInputParam) to the message <u>JSON_SubSysList</u> .
Get Faults of Systems and Partitions		Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: POST <u>/ISAPI/SecurityCP/status/systemFault?format=json</u> and set the input buffer (IpInBuffer) of the input parameter (IpInputParam) to the message <u>JSON_SubSysList</u> . And the fault parameters are returned in the message <u>JSON_ArmFault</u> by the output buffer (IpOutBuffer) of the output parameter (IpOutputParam).
Detect Faults	Get Configuration Capability of Fault Detection	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET <u>/ISAPI/SecurityCP/Configuration/faultCheckCfg/capabilities?format=json</u> . And the configuration capability is returned in the message <u>JSON_FaultCheckParameterCap</u> by the output parameter (IpOutputParam).
	Get Fault Detection Parameters	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET <u>/ISAPI/SecurityCP/Configuration/faultCheckCfg?format=json</u> . And the parameters are returned in the message <u>JSON_FaultCheckParameter</u> by the output buffer (IpOutBuffer) of the output parameter (IpOutputParam).
	Set Fault Detection Parameters	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: PUT <u>/ISAPI/SecurityCP/Configuration/faultCheckCfg?format=json</u> and set the input buffer (IpInBuffer) of the input parameter (IpInputParam) to the message <u>JSON_FaultCheckParameter</u> .
Configure Keypad Linkage of System Fault	Get Configuration Capability of Keypad Linkage of System Fault	Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET <u>/ISAPI/SecurityCP/Configuration/keypadFaultProcessCfg/capabilities?format=json</u> .

Function		Description
		And the configuration capability is returned in the message <i>JSON_KeypadFaultProcessCfgCap</i> by <i>IpOutputParam</i> .
	Get All Keypads' Linkage Parameters of System Fault	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET <i>/ISAPI/SecurityCP/Configuration/keypadFaultProcessCfg?format=json</i> . And the parameters are returned in the message <i>JSON_List_KeypadFaultProcessCfg</i> by <i>IpOutBuffer</i> of <i>IpOutputParam</i> .
	Set Linkage Parameters of A Specific Keypad for System Fault	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: PUT <i>/ISAPI/SecurityCP/Configuration/keypadFaultProcessCfg/<ID>?format=json</i> and set <i>IpInBuffer</i> of <i>IpInputParam</i> to the message <i>JSON_KeypadFaultProcessCfg</i> .

Audio and Video Detection

The audio and video detection function supports checking the video input, audio input, and audio output status. Two detection modes are available, i.e., automatic detection and manual detection. You can configure automatic detection schedules to automatically start detecting by schedule.

Function		Description
Automatic Audio and Video Detection	Get Configuration Capability of Automatic Audio and Video Detection	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET <i>/ISAPI/SecurityCP/sysAutoCheckTimeCfg/capabilities?format=json</i> . And the configuration capability is returned in the message <i>JSON_Cap_SysAutoCheckTimeCfg</i> by <i>IpOutputParam</i> .
	Get Automatic Audio and Video Detection Parameters	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET <i>/ISAPI/SecurityCP/sysAutoCheckTimeCfg?format=json</i> . And the parameters are returned in the message <i>JSON_SysAutoCheckTimeCfg</i> by <i>IpOutputParam</i> .
	Set Automatic Audio and Video Detection Parameters	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: PUT <i>/ISAPI/SecurityCP/sysAutoCheckTimeCfg?format=json</i> and set

Function		Description
		IpInputParam to the message <i>JSON_SysAutoCheckTimeCfg</i> .
Manual Audio and Video Detection	Get Configuration Capability of Manual Audio and Video Detection	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET <i>/ISAPI/SecurityCP/sysCheckManually/capabilities?format=json</i> . And the configuration capability is returned in the message <i>JSON_Cap_SysCheckManually</i> by IpOutputParam .
	Get Manual Audio and Video Detection Parameters	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET <i>/ISAPI/SecurityCP/sysCheckManually?format=json</i> . And the parameters are returned in the message <i>JSON_SysCheckManually</i> by IpOutputParam .
	Set Manual Audio and Video Detection Parameters	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: PUT <i>/ISAPI/SecurityCP/sysCheckManually?format=json</i> and set IpInputParam to the message <i>JSON_SysCheckManually</i> .
Get Detection Results	Get Capability of Getting Detection Results	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET <i>/ISAPI/SecurityCP/CheckResult/capabilities?format=json</i> . And the capability is returned in the message <i>JSON_Cap_CheckResult</i> by IpOutputParam .
	Get Detection Results	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET <i>/ISAPI/SecurityCP/CheckResult?format=json</i> . And the result parameters are returned in the message <i>JSON_CheckResult</i> by IpOutputParam .

Registration Mode

Function	Description
Get Configuration Capability of Registration Mode	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: GET <i>/ISAPI/SecurityCP/Configuration/registerMode/capabilities?format=json</i> .

Function	Description
	And the configuration capability is returned in the message <i>JSON_RegisterModeCap</i> by IpOutputParam .
Get Parameters of Registration Mode	Call <i>NET DVR STDXMLConfig</i> to transmit the request URI: GET / <i>ISAPI/SecurityCP/Configuration/registerMode?format=json</i> . And the parameters are returned in the message <i>JSON_RegisterMode</i> by IpOutBuffer of IpOutputParam .
Set Parameters of Registration Mode	Call <i>NET DVR STDXMLConfig</i> to transmit the request URI: PUT / <i>ISAPI/SecurityCP/Configuration/registerMode?format=json</i> and set IpInBuffer of IpInputParam to the message <i>JSON_RegisterMode</i> .
Get the Registration Status	Call <i>NET DVR STDXMLConfig</i> to transmit the request URI: GET / <i>ISAPI/SecurityCP/Configuration/registerMode/registerStatus?format=json</i> . And the status is returned in the message <i>JSON_WirelessRecv</i> by IpOutBuffer of IpOutputParam .

Device Environment

Function	Description
Get Device Environment Configuration Capability	Call <i>NET DVR STDXMLConfig</i> to transmit the request URI: GET / <i>ISAPI/SecurityCP/surroundEnvironmentCfg/capabilities?format=json</i> . And the configuration capability is returned in the message <i>JSON_SurrondParaCap</i> by IpOutputParam .
Get Device Environment Parameters	Call <i>NET DVR STDXMLConfig</i> to transmit the request URI: GET / <i>ISAPI/SecurityCP/surroundEnvironmentCfg?format=json</i> . And the parameters are returned in the message <i>JSON_SurrondParaCfg</i> by IpOutputParam .
Set Device Environment Parameters	Call <i>NET DVR STDXMLConfig</i> to transmit the request URI: PUT / <i>ISAPI/SecurityCP/surroundEnvironmentCfg?format=json</i> and set IpInputParam to the message <i>JSON_SurrondParaCfg</i> .

Signal Strength Detection

Function	Description
Get Configuration Capability of Signal Strength Detection in Asynchronous Mode	<p>Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET <u>/ISAPI/SecurityCP/Configuration/signalStrengthDetection/mode/capabilities?format=json</u> .</p> <p>And the configuration capability is returned in the message <u>JSON_SignalStrengthDetectionCap</u> by IpOutputParam.</p>
Start Signal Strength Detection in Asynchronous Mode	<p>Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: POST <u>/ISAPI/SecurityCP/Configuration/signalStrengthDetection/mode?format=json</u> and set IpInputParam to the message <u>JSON_SignalStrengthDetectionMode</u> .</p> <p>The result parameters are returned in the message <u>JSON_Result</u> by IpOutputParam.</p>
Stop Signal Strength Detection in Asynchronous Mode	<p>Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: PUT <u>/ISAPI/SecurityCP/Configuration/signalStrengthDetection/mode?format=json</u> and set IpInputParam to the message <u>JSON_SignalStrengthDetectionMode</u> .</p>
Get Current Signal Strength in Asynchronous Mode	<p>Call <u>NET_DVR_STDXMLConfig</u> to transmit the request URI: GET <u>/ISAPI/SecurityCP/Configuration/signalStrengthDetection/currentAsyn?format=json</u> .</p> <p>And the parameters are returned in the message <u>JSON_SignalStrengthDetection</u> by IpOutputParam.</p>

Chapter 3 API Reference

3.1 NET_DVR_AlarmHostClearAlarm

Clear alarms of partition.

API Definition

```
BOOL NET_DVR_AlarmHostClearAlarm(  
    LONG        lUserID,  
    DWORD       dwSubSystemNum  
) ;
```

Parameters

lUserID

[IN] Value returned by NET_DVR_Login_V40 .

dwSubSystemNum

[IN] Partition No., which starts from 0.

Return Values

Return *TRUE* for success, and return *FALSE* for failure.

If *FALSE* is returned, you can call NET_DVR_GetLastError to get the error code.

3.2 NET_DVR_AlarmHostSubSystemCloseAlarmChan

Disarm the partition.

API Definition

```
BOOL NET_DVR_AlarmHostSubSystemCloseAlarmChan(  
    LONG        lUserID,  
    DWORD       dwSubSystemNum  
) ;
```

Parameters

lUserID

[IN] Value returned by NET_DVR_Login_V40 .

dwSubSystemNum

[IN] Partition No., 0xffffffff-all partitions.

Return Values

Return *TRUE* for success, and return *FALSE* for failure.

If *FALSE* is returned, you can call **NET_DVR_GetLastError** to get the error code.

See Also

NET_DVR_AlarmHostSubSystemSetupAlarmChan

3.3 NET_DVR_AlarmHostSubSystemSetupAlarmChan

Arm the partition.

API Definition

```
BOOL NET_DVR_AlarmHostSubSystemSetupAlarmChan(  
    LONG        lUserID,  
    DWORD       dwSubSystemNum  
);
```

Parameters

lUserID

[IN] Value returned by **NET_DVR_Login_V40**.

dwSubSystemNum

[IN] Partition No., 0xffffffff-all partitions.

Return Values

Return *TRUE* for success, and return *FALSE* for failure.

If *FALSE* is returned, you can call **NET_DVR_GetLastError** to get the error code.

See Also

NET_DVR_AlarmHostSubSystemCloseAlarmChan

3.4 NET_DVR_BypassAlarmChan

Perform bypass on the zone.

API Definition

```
BOOL NET_DVR_BypassAlarmChan(  
    LONG        lUserID,  
  
    NET_DVR_ALARMIN_SETUP
```

```
);
```

Parameters

IUserID

[IN] Value returned by [**NET_DVR_Login_V40**](#).

lpInter

[IN] Zone parameters, see details in the structure [**NET_DVR_ALARMIN_SETUP**](#).

Return Values

Return *TRUE* for success, and return *FALSE* for failure.

If *FALSE* is returned, you can call [**NET_DVR_GetLastError**](#) to get the error code.

Remarks

Bypass is not allowed in the following situations: the zone is in arming status; the zone is a 24-hour zone; the security control panel is in programming mode and test mode. Otherwise, *FALSE* will be returned, and the corresponding error code is 11.

See Also

[**NET_DVR_UnBypassAlarmChan**](#)

3.5 NET_DVR_Cleanup

Release the resources after the program is ended.

API Definition

```
BOOL NET_DVR_Cleanup(  
);
```

Return Values

Returns *TRUE* for success, and returns *FALSE* for failure.

If *FALSE* is returned, you can call [**NET_DVR_GetLastError**](#) to get the error code.

The available error codes may be returned by this API are 0 and 3. See details in [**Device Network SDK Errors**](#).

Remarks

- When calling this API, you cannot call other APIs at the same time.
- [**NET_DVR_Init**](#) and this API should be called by pair. That is, once the [**NET_DVR_Init**](#) is called, you should call [**NET_DVR_Cleanup**](#) to release the resources when exiting the program.

3.6 NET_DVR_FindAlarmHostLog

Search for security control panel logs.

API Definition

```
LONG NET_DVR_FindAlarmHostLog(  
    LONG                                lUserID,  
    LONG                                lSelectMode,  
    NET_DVR_ALARMHOST_SEARCH_LOG_PARAM *lpSearchParam  
);
```

Parameters

lUserID

[IN] Value returned by NET_DVR_Login_V40.

lSelectMode

[IN] Searching mode: 0-all, 1-by type, 2-by time, 3-by time and type.

lpSearchParam

[IN] Structure about the conditions of searching for security control panel logs, see details in NET_DVR_ALARMHOST_SEARCH_LOG_PARAM.

Return Values

Return -1 for failure, and return other values as the parameter of NET_DVR_FindNextAlarmHostLog.

If -1 is returned, you can call NET_DVR_GetLastError to get the error code.

See Also

NET_DVR_FindNextAlarmHostLog

3.7 NET_DVR_FindNextAlarmHostLog

Search for the next security control panel log, i.e., get the searched log information of the security control panel one by one.

API Definition

```
LONG NET_DVR_FindNextAlarmHostLog(  
    LONG                                lFindHandle,  
    NET_DVR_ALARMHOST_LOG_RET          *lpFindData  
);
```

Parameters

IFindHandle

[IN] Handle for searching for logs, which is the value returned by [**NET_DVR_FindAlarmHostLog**](#) .

lpFindData

[OUT] Pointer of saving log information, see details in the structure [**NET_DVR_ALARMHOST_LOG_RET**](#) .

Return Values

Return -1 for failure, and return other values as the current getting status, see details in the following table.

Status	Value	Description
NET_DVR_FILE_SUCCESS	1000	Getting log information succeeded.
NET_DVR_FILE_NOFIND	1001	No log found.
NET_DVR_ISFINDING	1002	Searching. Please wait.
NET_DVR_NOMOREFILE	1003	No more log found. Searching ended.
NET_DVR_FILE_EXCEPTION	1004	Searching exception.

If -1 is returned, you can call [**NET_DVR_GetLastError**](#) to get the error code.

Remarks

Before getting the searched logs by calling this API, you should call [**NET_DVR_FindAlarmHostLog**](#) to get the current search handle.

See Also

[**NET_DVR_FindAlarmHostLog**](#)

3.8 NET_DVR_GetAlarmDeviceUser

Get device user parameters of security control panels.

API Definition

```
BOOL NET_DVR_GetAlarmDeviceUser(  
    LONG                lUserID,  
    LONG                lUserIndex,  
    NET_DVR_ALARM_DEVICE_USER *lpDeviceUser  
);
```

Parameters

IUserID

[IN] Value returned by **NET_DVR_Login_V40** .

IUserIndex

[IN] Device user index of security control panels.

lpDeviceUser

[OUT] Device user configuration, which is a pointer pointing to **NET_DVR_ALARM_DEVICE_USER** .

Return Values

Return *TRUE* for success, and return *FALSE* for failure.

If *FALSE* is returned, you can call **NET_DVR_GetLastError** to get the error code.

Remarks

- For security control panels, the device users refer to those log in to the device remotely via SDK, and the operation users refer to those perform operations on the device locally (for example, perform operations on the security control panel by using keypad).
- The device users (also referred to as network users) include admin users, administrators, and normal users. The maximum number of users and their permission can be obtained by calling the API **NET_DVR_GetDeviceAbility** (**dwAbilityType**: *DEVICE_USER_ABILITY*; related node: **<AlarmhostPermission>**).
- Admin users: By default the first user of the device is the admin user which is also a kind of administrators but has more permission than normal administrators. There is only one admin user for a device. The admin user can set or modify the permission of normal users and view all users' information. The admin user's permission cannot be modified.
- Administrators: For video security control panels, the administrators have all permission of the admin user except restoring to default settings, formatting HDD, upgrading the system program, and restarting. For other security control panels, the administrators can have all permission of the admin user. The permission of administrators cannot be modified by any users (including the admin user). Administrators can view information of normal users and themselves (that is, they cannot view information of the admin user and other administrators), and set or modify permission of normal users (they cannot modify permission of themselves).
- Normal users: By default they have permission of getting parameters. Other permission should be configured to take effect. The highest available permission level of normal users is that of administrators. Normal users can view information of themselves, but they cannot modify permission of themselves or view information of the admin user, administrators, and other normal users.

Related API

NET_DVR_GetAlarmDeviceUser

3.9 NET_DVR_GetBatteryVoltage

Get battery voltage.

API Definition

```
BOOL NET_DVR_GetBatteryVoltage(  
    LONG    lUserID,  
    float    *pVoltage  
);
```

Parameters

lUserID

[IN] Value returned by NET_DVR_Login_40.

pVoltage

[OUT] Battery voltage.

Return Values

Return *TRUE* for success, and return *FALSE* for failure.

If *FALSE* is returned, you can call NET_DVR_GetLastError to get the error code.

3.10 NET_DVR_GetDeviceAbility

Get the device capabilities.

API Definition

```
BOOL NET_DVR_GetDeviceAbility(  
    LONG    lUserID,  
    DWORD    dwAbilityType,  
    char    *pInBuf,  
    DWORD    dwInLength,  
    char    *pOutBuf,  
    DWORD    dwOutLength  
);
```

Parameters

lUserID

[IN] Value returned by NET_DVR_Login_V40.

dwAbilityType

[IN] Capability types, which are different according to different devices and functions.

pInBuf

[IN] Input parameter buffer pointer, which are different according to different devices and functions, and they are returned in the structure or messages.

dwInLength

[IN] Size of input buffer.

pOutBuf

[OUT] Output parameter buffer pointer, which are different according to different devices and functions, and they are returned in the structure or messages.

dwOutLength

[OUT] Size of buffer for receiving data.

Return Values

Returns *TRUE* for success, and returns *FALSE* for failure.

If *FALSE* is returned, you can call **NET_DVR_GetLastError** to get the error code.

3.11 NET_DVR_GetDeviceConfig

Get device configuration information in batch (with sending data).

API Definition

```
BOOL NET_DVR_GetDeviceConfig(  
    LONG        lUserID,  
    DWORD       dwCommand,  
    DWORD       dwCount,  
    LPVOID      lpInBuffer,  
    DWORD       dwInBufferSize,  
    LPVOID      lpStatusList,  
    LPVOID      lpOutBuffer,  
    DWORD       dwOutBufferSize  
) ;
```

Parameters

lUserID

[IN] Value returned by **NET_DVR_Login_V40**.

dwCommand

[IN] Device getting commands. The commands are different for different getting functions.

dwCount

[IN] Number of configurations (cameras) to get at a time. 0, 1-one camera, 2-two cameras, 3-three cameras, and so on. Up to 64 cameras' configuration information can be obtained at a time.

lpInBuffer

[IN] Pointer of configuration condition buffer, which specifies the number (**dwCount**) of configurations to get, and relates to the getting commands.

dwInBufferSize

[IN] Size of configuration condition buffer, which saves the obtained configuration information (the number is **dwCount**).

lpStatusList

[OUT] Error information list, and its memory is allocated by user, each error information contains 4 bytes (a unsigned 32-bit integer).

There is a one-to-one correspondence between the errors in the list and the cameras need to search, e.g., **lpStatusList[2]** corresponds to **lpInBuffer[2]**.

If the parameter value is 0 or 1, it refers to getting succeeded, otherwise, this parameter value is the error code.

lpOutBuffer

[OUT] Parameters returned by device, which relates to the getting commands. And there is a one-to-one correspondence between the parameters and the cameras need to search.

If the **lpStatusList** of one camera is larger than 1, the corresponding **lpOutBuffer** is invalid.

dwOutBufferSize

[IN] Total size of returned results (the number is **dwCount**).

Return Values

Returns *TRUE* for success, and returns *FALSE* for failure. If returns *TRUE*, it does not mean that all configurations are obtained, you can check the value of **lpStatusList[n]** to judge which one is succeeded.

If *FALSE* is returned, you can call **NET_DVR_GetLastError** to get the error code.

See Also

NET_DVR_SetDeviceConfig

3.12 NET_DVR_GetDVRConfig

Get the device configuration information.

API Definition

```
BOOL NET_DVR_GetDVRConfig(  
    LONG        lUserID,  
    DWORD       dwCommand,  
    LONG        lRuleID,  
    LONG        lChannel,  
    LPVOID       lpOutBuffer,  
    DWORD       dwOutBufferSize,
```

```
LPDWORD    lpBytesReturned
);
```

Parameters

lUserID

[IN] Value returned by **NET_DVR_Login_V40**.

dwCommand

[IN] Device getting commands, which are different according to different getting functions.

lRuleID

[IN] Rule ID.

lChannel

[IN] Channel No. (NIC No.), which varies with different commands. 0xffffffff-invalid or all channels, 1-main NIC, 2-extended NIC.

lpOutBuffer

[OUT] Pointer of buffer to receive data. For different getting functions, the structures of this parameter are different.

dwOutBufferSize

[IN] Size of buffer to receive data (unit: byte). It cannot be 0.

lpBytesReturned

[OUT] Pointer of actually received data size. It cannot be NULL.

Return Values

Returns *TRUE* for success, and returns *FALSE* for failure.

If *FALSE* is returned, you can call **NET_DVR_GetLastError** to get the error code.

The following error codes may be returned by this API: 0, 3, 6, 7, 8, 9, 10, 12, 17, 41, 43, 44, 47, 72, 73, and 76. See the corresponding error types and descriptions in the **Device Network SDK Errors**.

See Also

NET_DVR_SetDVRConfig

3.13 NET_DVR_GetErrorMsg

Return the error information of the last operation.

API Definition

```
char *NET_DVR_GetErrorMsg(
    LONG    *pErrorNo
);
```

Parameters

pErrorNo

[OUT] Error code pointer.

Return Values

The return values are the pointers of error information, see [***Device Network SDK Errors***](#) for details.

Remarks

You can call [***NET_DVR_GetLastError***](#) to get the error codes.

3.14 NET_DVR_GetLastError

Return the error code of the last operation.

API Definition

```
DWORD NET_DVR_GetLastError(  
);
```

Return Values

The return values are error codes, see [***Device Network SDK Errors***](#) for details.

Remarks

You can also call [***NET_DVR_GetErrorMsg***](#) to directly get the error information.

3.15 NET_DVR_GetNextRemoteConfig

Get the next search result.

API Definition

```
LONG NET_DVR_GetNextRemoteConfig(  
    LONG        lHandle,  
    void        *lpOutBuff,  
    DWORD       dwOutBuffSize  
);
```

Parameters

lHandle

[IN] Search handle, which is the value returned by [***NET_DVR_StartRemoteConfig***](#).

lpOutBuff

[OUT] Output parameter buffer pointer, which relates to the commands (**dwCommand**) of **NET_DVR_StartRemoteConfig**.

dwOutBuffSize

[IN] Buffer size.

Return Values

Returns **-1** for failure, and returns other values for the current statuses, see details in the following table.

Status	Value	Description
NET_SDK_GET_NEXT_STATUS_SUCCESS	1000	The data is obtained. The API NET_DVR_GetNextRemoteConfig should be called again to get the next item of data.
NET_SDK_GET_NETX_STATUS_NEED_WAIT	1001	Waiting. The API NET_DVR_GetNextRemoteConfig can be called again.
NET_SDK_GET_NEXT_STATUS_FINISH	1002	All data is obtained. The API <u>NET_DVR_StopRemoteConfig</u> can be called to end.
NET_SDK_GET_NEXT_STATUS_FAILED	1003	Getting data exception. The API <u>NET_DVR_StopRemoteConfig</u> can be called to end.

If **-1** is returned, you can call **NET_DVR_GetLastError** to get the error code.

Remarks

To get all information, you should call this API repeatedly.

3.16 NET_DVR_GetSDKLocalCfg

Get the HCNetSDK's local configuration parameters.

API Definition

```

BOOL NET_DVR_GetSDKLocalCfg(
    NET_SDK_LOCAL_CFG_TYPE    enumType,
    void                      *lpOutBuff
);

```

Parameters

enumType

[IN] Configuration options. Different values of configuration options correspond to different parameters, see details in **NET_SDK_LOCAL_CFG_TYPE**.

lpOutBuff

[OUT] Output parameters. For different configuration options, the structures of output parameters are different, see details in [NET_SDK_LOCAL_CFG_TYPE](#).

Return Values

Returns *TRUE* for success, and returns *FALSE* for failure. If *FALSE* is returned, you can call [NET_DVR_GetLastError](#) to get the error code.

See Also

[NET_DVR_SetSDKLocalCfg](#)

3.17 NET_DVR_GetSTDAbility

Get the device capabilities.

API Definition

```
BOOL NET_DVR_GetSTDAbility(  
    LONG                lUserID,  
    DWORD               dwAbilityType,  
    NET_DVR_STD_ABILITY lpAbilityParam  
);
```

Parameters

lUserID

[IN] Value returned by [NET_DVR_Login_V40](#).

dwAbilityType

[IN] Capability types, which are different according to different functions.

lpAbilityParam

[IN/OUT] Capability details, including condition parameter, input parameter, output parameter, and so on (see details in the structure [NET_DVR_STD_ABILITY](#)), which are different according to different capability types.

Return Values

Returns *TRUE* for success, and returns *FALSE* for failure.

If *FALSE* is returned, you can call [NET_DVR_GetLastError](#) to get the error code.

3.18 NET_DVR_GetSTDConfig

Get the device configuration information.

API Definition

```
BOOL NET_DVR_GetSTDConfig(  
    LONG                lUserID,  
    DWORD               dwCommand,  
    NET_DVR_STD_CONFIG  lpConfigParam  
);
```

Parameters

lUserID

[IN] Value returned by [NET_DVR_Login_V40](#).

dwCommand

[IN] Device configuration commands, which are different according to different configuration functions.

lpConfigParam

[IN][OUT] Set input and output parameters, which are different according to different configuration functions. For different configuration functions, the **lpCondBuffer** and **lpOutBuffer** in the **lpConfigParam** are also different. See the structure [NET_DVR_STD_CONFIG](#) for details.



Note

When getting configuration parameters, the **lpInBuffer** in the **lpConfigParam** is invalid, you can set it to NULL.

Return Values

Returns *TRUE* for success, and returns *FALSE* for failure.

If *FALSE* is returned, you can call [NET_DVR_GetLastError](#) to get the error code.

See Also

[NET_DVR_SetSTDConfig](#)

3.19 NET_DVR_Init

Initialize the programming environment before calling other APIs.

API Definition

```
BOOL NET_DVR_Init(  
);
```

Return Values

Returns *TURE* for success, and returns *FALSE* for failure.

If *FALSE* is returned, you can call [NET_DVR_GetLastError](#) to get the error code.

The available error codes of this API are 0, 41, and 53. See details in [Device Network SDK Errors](#) .

Remarks

Before initializing, you can call [NET_DVR_SetSDKInitCfg](#) to set the initialization parameters, such as supported capabilities, loading path of component libraries (only supported by Linux system), and so on.

See Also

[NET_DVR_Cleanup](#)

3.20 NET_DVR_Login_V40

Log in to the device (supports asynchronous login).

API Definition

```
LONG NET_DVR_Login_V40 (
    NET_DVR_USER_LOGIN_INFO    pLoginInfo,
    NET_DVR_DEVICEINFO_V40     lpDeviceInfo
);
```

Parameters

pLoginInfo

[IN] Login parameters, including device address, user name, password, and so on. See details in the structure [NET_DVR_USER_LOGIN_INFO](#) .

lpDeviceInfo

[OUT] Device information. See details in the structure [NET_DVR_DEVICEINFO_V40](#) .

Return Values

- For asynchronous login, the callback function ([fLoginResultCallBack](#)) configured in the structure ([NET_DVR_USER_LOGIN_INFO](#)) returns the asynchronous login status, user ID and device information.
- For synchronous login, this API returns -1 for logging failed, and returns other values for the returned user IDs. The user ID is unique, and it helps to realize the further device operations.
- If -1 is returned, you can call [NET_DVR_GetLastError](#) to get the error code.

Remarks

- When **bUseAsynLogin** in **pLoginInfo** is 0, it indicates that login is in synchronous mode; when **bUseAsynLogin** in **pLoginInfo** is 1, it indicates that login is in asynchronous mode.
- Up to 2048 users are allowed to log in to HCNetsDK at same time, and the values of returned **UserID** are ranging from 0 to 2047.

See Also

[**NET_DVR_Logout**](#)

3.21 NET_DVR_Logout

Log out from devices.

API Definitions

```
BOOL NET_DVR_Logout (
    LONG    lUserID
);
```

Parameters

lUserID

[IN] User ID, which is returned by [**NET_DVR_Login_V40**](#).

Return Values

Returns *TRUE* for success, and returns *FALSE* for failure.

If *FALSE* is returned, you can call [**NET_DVR_GetLastError**](#) to get the error code.

The available error codes may be returned by this API are 0, 3, 7, 8, 9, 10, 14, 17, 41, 44, 47, 72, and 73. See details in [**Device Network SDK Errors**](#).

3.22 NET_DVR_RemoteControl

Implement remote control.

API Definition

```
BOOL NET_DVR_RemoteControl (
    LONG        lUserID,
    DWORD       dwCommand,
    LPVOID      lpInBuffer,
    DWORD       dwInBufferSize
);
```

Parameters

lUserID

[IN] Value returned by [**NET_DVR_Login_V40**](#).

dwCommand

[IN] Control commands. To realize different functions, the commands are different.

IpInBuffer

[IN] Input parameters, which vary with different control commands.

dwInBufferSize

[IN] Size of input parameters.

Return Values

Returns *TRUE* for success, and returns *FALSE* for failure.

If *FALSE* is returned, you can call **NET_DVR_GetLastError** to get the error code.

3.23 NET_DVR_SendRemoteConfig

Send data via the persistent connection.

API Definition

```
BOOL NET_DVR_SendRemoteConfig(  
    LONG        lHandle,  
    DWORD       dwDataType,  
    char        *pSendBuf,  
    DWORD       dwBufSize  
);
```

Parameters

lHandle

Persistent configuration handle, which is returned by **NET_DVR_StartRemoteConfig**.

dwDataType

[IN] Data type, which relates to the commands of **NET_DVR_StartRemoteConfig**.

pSendBuf

[IN] Buffer for saving data to be sent, which relates to **dwDataType**.

dwBufSize

[IN] Size of data to be sent.

Return Values

Returns *TRUE* for success, and returns *FALSE* for failure.

If *FALSE* is returned, you can call **NET_DVR_GetLastError** to get the error code.

Remarks

Before calling this API, you must call **NET_DVR_StartRemoteConfig** to get the persistent connection handle.

3.24 NET_DVR_SetAlarmDeviceUser

Set device user parameters of security control panels.

API Definition

```
BOOL NET_DVR_SetAlarmDeviceUser (
    LONG                lUserID,
    LONG                lUserIndex,
    NET_DVR_ALARM_DEVICE_USER *lpDeviceUser
);
```

Parameters

lUserID

[IN] Value returned by NET_DVR_Login_V40 .

lUserIndex

[IN] Device user index of security control panels.

lpDeviceUser

[IN] Device user configuration, which is a pointer pointing to NET_DVR_ALARM_DEVICE_USER .

Return Values

Return *TRUE* for success, and return *FALSE* for failure.

If *FALSE* is returned, you can call NET_DVR_GetLastError to get the error code.

Remarks

- For security control panels, the device users refer to those log in to the device remotely via SDK, and the operation users refer to those perform operations on the device locally (for example, perform operations on the security control panel by using keypad).
- The device users (also referred to as network users) include admin users, administrators, and normal users. The maximum number of users and their permission can be obtained by calling the API NET_DVR_GetDeviceAbility (**dwAbilityType**: DEVICE_USER_ABILITY; related node: <AlarmhostPermission>).
 - Admin users: By default the first user of the device is the admin user which is also a kind of administrators but has more permission than normal administrators. There is only one admin user for a device. The admin user can set or modify the permission of normal users and view all users' information. The admin user's permission cannot be modified.
 - Administrators: For video security control panels, the administrators have all permission of the admin user except restoring to default settings, formatting HDD, upgrading the system program, and restarting. For other security control panels, the administrators can have all permission of the admin user. The permission of administrators cannot be modified by any users (including the admin user). Administrators can view information of normal users and themselves (that is, they cannot view information of the admin user and other

administrators), and set or modify permission of normal users (they cannot modify permission of themselves).

- Normal users: By default they have permission of getting parameters. Other permission should be configured to take effect. The highest available permission level of normal users is that of administrators. Normal users can view information of themselves, but they cannot modify permission of themselves or view information of the admin user, administrators, and other normal users.

Related API

NET_DVR_GetAlarmDeviceUser

3.25 NET_DVR_SetAlarmHostOut

Set alarm output for security control panel.

API Definition

```
BOOL NET_DVR_SetAlarmHostOut (
    LONG      lUserID,
    LONG      lAlarmOutPort,
    LONG      lAlarmOutStatic
);
```

Parameters

lUserID

[IN] Value returned by **NET_DVR_Login_V40** .

lAlarmOutPort

[IN] Alarm output No., which starts from 0, and 0xffffffff indicates all alarm outputs.

lAlarmOutStatic

[IN] Alarm output status: 0-no output, 1-output.

Return Values

Return *TRUE* for success, and return *FALSE* for failure.

If *FALSE* is returned, you can call **NET_DVR_GetLastError** to get the error code.

3.26 NET_DVR_SetConnectTime

Set network connection timeout and connection attempts.

API Definition

```
BOOL NET_DVR_SetConnectTime(  
    DWORD    dwWaitTime,  
    DWORD    dwTryTimes  
);
```

Parameters

dwWaitTime

[IN] Timeout, unit: ms, value range: [300,75000]; the maximum timeout varies with different operating systems.

dwTryTimes

[IN] Connection attempts (reserved).

Return Values

Return *TRUE* for success, and return *FALSE* for failure.

If *FALSE* is returned, you can call ***NET_DVR_GetLastError*** to get the error code.

Remarks

- For Windows operating system, the default connection timeout is 3000 ms; for Linux operating system with version 5.2.7.2 and above, the default connection timeout is 3500 ms.
- For HCNetsDK with version 4.0 and above, when the configured timeout is larger than or smaller than the limit value, this API will not return *FALSE*, it will automatically use the timeout that is closest to the limit value as the actual timeout.

3.27 NET_DVR_SetDeviceConfig

Set device parameters in batch (sending data is supported).

API Definition

```
BOOL NET_DVR_SetDeviceConfig(  
    LONG        lUserID,  
    DWORD       dwCommand,  
    DWORD       dwCount,  
    LPVOID      lpInBuffer,  
    DWORD       dwInBufferSize,  
    LPVOID      lpStatusList,  
    LPVOID      lpInParamBuffer,  
    DWORD       dwInParamBufferSize  
);
```

Parameters

lUserID

[IN] Value returned by **NET_DVR_Login_V40** .

dwCommand

[IN] Device configuration commands, which are different according to different configurations.

dwCount

[IN] Number of cameras to be set at a time. 0,1-one camera, 2-two cameras, 3-three cameras, and so on. Up to 256 cameras can be configured at a time.

lpInBuffer

[IN] Pointer of configuration condition buffer, e.g., stream ID, which specifies the number (**dwCount**) of cameras to set, and relates to the configuration commands.

dwInBufferSize

[IN] Size of configuration condition buffer, which saves the configured information of cameras with the number of **dwCount**.

lpStatusList

[OUT] Error information list, and its memory is allocated by user, each error information contains 4 bytes (a unsigned 32-bit integer).

There is a one-to-one correspondence between the errors in the list and the cameras that need to be searched, e.g., **lpStatusList[2]** corresponds to **lpInBuffer[2]**.

If the parameter value is 0, it refers to setting succeeded, otherwise, this parameter value is the error code.

lpInParamBuffer

[IN] Device parameters to set, which relates to the configuration commands. And there is a one-to-one correspondence between the parameters and the cameras that need to be searched.

dwInParamBufferSize

[IN] Set the size of content buffer.

Return Values

Returns *TRUE* for success, and returns *FALSE* for all failed. If returns *TRUE*, it does not indicate that all settings are succeeded, you can get the value of **lpStatusList[n]** to check which one is succeeded.

If *FALSE* is returned, you can call **NET_DVR_GetLastError** to get the error code.

See Also

NET_DVR_GetDeviceConfig

3.28 NET_DVR_SetDVRConfig

Set the device parameters.

API Definition

```
BOOL NET_DVR_SetDVRConfig(  
    LONG        lUserID,  
    DWORD       dwCommand,  
    LONG        lChannel,  
    LPVOID      lpInBuffer,  
    DWORD       dwInBufferSize  
);
```

Parameters

lUserID

[IN] Value returned by **NET_DVR_Login_V40** .

dwCommand

[IN] Device configuration commands, which are different according to different configuration functions.

lChannel

[IN] Channel No. (NIC No.), which varies with different commands. 0xFFFFFFFF-invalid, 1-main NIC, 2-extended NIC.

lpInBuffer

[IN] Pointer of input data buffer. For different configuration functions, the structures of this parameter are different.

dwInBufferSize

[IN] Size of input data buffer (unit: byte).

Return Values

Returns *TRUE* for success, and returns *FALSE* for failure.

If *FALSE* is returned, you can call **NET_DVR_GetLastError** to get the error code.

The following error codes may be returned by this API: 0, 3, 6, 7, 8, 9, 10, 12, 17, 41, 43, 44, 47, 72, 73, and 76. See the corresponding error types and descriptions in the **Device Network SDK Errors** .

See Also

NET_DVR_GetDVRConfig

3.29 NET_DVR_SetSDKInitCfg

Set initialization parameters.

API Parameters

```
BOOL NET_DVR_SetSDKInitCfg(  
    NET_SDK_INIT_CFG_TYPE    enumType,  
    void* const               lpInBuff  
);
```

Parameters

enumType

[IN] Initialization parameter type. Different type values correspond to different parameters, see details in the table below.

Table 3-1 NET_SDK_INIT_CFG_TYPE

enumType	Value	Description	lpInBuff
NET_SDK_INIT_CFG_ABILITY	1	Capability supported by SDK.	<u>NET_DVR_INIT_CFG_ABILITY</u>
NET_SDK_INIT_CFG_SDK_PATH	2	Set loading path for component libraries (supported by both Linux and Windows system).	<u>NET_DVR_LOCAL_SDK_PATH</u>
NET_SDK_INIT_CFG_LIBEAY_PATH	3	Set path (including library name) for libeay32.dll (Windows), libcrypto.so (Linux), and libcrypto.dylib (Mac) of OpenSSL in version 1.1.1 and 1.0.2.	Path in string format, e.g., C:\\libeay32.dll .
NET_SDK_INIT_CFG_SSLEAY_PATH	4	Set path (including library name) for ssleay32.dll (Windows), libssl.so (Linux), libssl.dylib (Mac) of OpenSSL in version 1.1.1 and 1.0.2.	Path in string format, e.g., C:\\ssleay32.dll .

lpInBuff

[IN] Input parameter. Different parameter types correspond to different structures, see details in the table above.

Return Values

Returns *TRUE* for success, and returns *FALSE* for failure.

If *FALSE* is returned, you can call **NET_DVR_GetLastError** to get the error code.

Remarks

This API should be called before calling **NET_DVR_Init** to initialize and check the dependent libraries or capabilities. This API only takes effect for POSIX. For Windows, it takes no effect but success will be returned.

3.30 NET_DVR_SetSDKLocalCfg

Set the local parameters.

API Definition

```
BOOL NET_DVR_SetSDKLocalCfg(  
    NET_SDK_LOCAL_CFG_TYPE    enumType,  
    void* const                lpInBuff  
);
```

Parameters

enumType

[IN] Configuration options. Different values of configuration options correspond to different SDK parameters, see details in **NET_SDK_LOCAL_CFG_TYPE**.

lpInBuff

[IN] Input parameters. For different configuration options, the structures of input parameters are different, see details in **NET_SDK_LOCAL_CFG_TYPE**.

Return Values

Returns *TRUE* for success, and returns *FALSE* for failure. If *FALSE* is returned, you can call **NET_DVR_GetLastError** to get the error code.

Before setting parameters for this function, make sure no device has logged in.

See Also

NET_DVR_GetSDKLocalCfg

3.31 NET_DVR_SetSTDConfig

Set the device parameters.

API Definition

```
BOOL NET_DVR_SetSTDConfig(  
    LONG                lUserID,  
    DWORD               dwCommand,  
    NET_DVR_STD_CONFIG  lpConfigParam  
);
```

Parameters

lUserID

[IN] Value returned by [NET_DVR_Login_V40](#).

dwCommand

[IN] Device configuration commands, which are different according to different configuration functions.

lpConfigParam

[IN][OUT] Set input and output parameters, which are different according to different configuration functions. For different configuration functions, the **lpCondBuffer** and **lpInBuffer** in the **lpConfigParam** are also different. See the structure [NET_DVR_STD_CONFIG](#) for details.



Note

When getting configuration parameters, the **lpOutBuffer** in the **lpConfigParam** is invalid, you can set it to "NULL".

Return Values

Returns *TRUE* for success, and returns *FALSE* for failure.

If *FALSE* is returned, you can call [NET_DVR_GetLastError](#) to get the error code.

See Also

[NET_DVR_GetSTDConfig](#)

3.32 NET_DVR_StartRemoteConfig

Enable remote configuration.

API Definition

```
LONG NET_DVR_StartRemoteConfig(  
    LONG                lUserID,
```

```

DWORD          dwCommand,
LPVOID         lpInBuffer,
DWORD          dwInBufferLen,
fRemoteConfigCallback cbStateCallback,
LPVOID         pUserData
);

```

Parameters

lUserID

[IN] Value returned by **NET_DVR_Login_V40** .

dwCommand

[IN] Configuration commands. For different functions, the commands and **lpInBuffer** are different, see the detailed relation in the table below:

dwCommand Macro Definition	Value	Description	lpInBuffer Related Structure	lpBuffer Related Structure
NET_DVR_GET_ALL_RECORD_PASSBACK_TASK_MANUAL	6235	Get tasks of manually copying back videos	<u>NET_DVR_RECORD_PASSBACK_MANUAL_COND</u>	<u>NET_DVR_RECORD_PASSBACK_MANUAL_TASK_RET</u>

lpInBuffer

Input parameter buffer pointer, which relates to the configuration command.

dwInBufferLen

[IN] Size of input buffer.

cbStateCallback

[IN] Status callback function, see the definition in **fRemoteConfigCallback** .

pUserData

[OUT] User data.

Return Values

Returns -1 for failure, and returns other values for the handles of **NET_DVR_GetNextRemoteConfig** and **NET_DVR_StopRemoteConfig** .

If -1 is returned, you can call **NET_DVR_GetLastError** to get the error code.

Remarks

This API specifies the information to search. After calling this API, you can call **NET_DVR_GetNextRemoteConfig** to get the information one by one.

3.32.1 fRemoteConfigCallback

Function for calling back the persistent connection status and data to be transmitted.

Callback Function Definition

```
void(CALLBACK *fRemoteConfigCallback) (
    DWORD        dwType,
    void          *lpBuffer,
    DWORD        dwBufLen,
    void          *pUserData
);
```

Parameters

dwType

[OUT] Connection statuses, see the macro definitions below:

```
enum _NET_SDK_CALLBACK_TYPE_ {
    NET_SDK_CALLBACK_TYPE_STATUS    = 0,
    NET_SDK_CALLBACK_TYPE_PROGRESS  = 1,
    NET_SDK_CALLBACK_TYPE_DATA      = 2,
}NET_SDK_CALLBACK_TYPE
```

NET_SDK_CALLBACK_TYPE_STATUS

Connection status.

NET_SDK_CALLBACK_TYPE_PROGRESS

Connection progress.

NET_SDK_CALLBACK_TYPE_DATA

Related data to be called back.

lpBuffer

[OUT] Pointer of buffer for saving progress, status, and related data to be called back, which relates to **dwType**, see details in the following table.

dwType	lpBuffer
NET_SDK_CALLBACK_TYPE_STATUS	If dwBufLen is 4, lpBuffer is 4-byte connection status; if dwBufLen is 8, lpBuffer consists of 4-byte connection status and 4-byte error code.

dwType	lpBuffer
	The connection status is enumerated in <u>NET_SDK_CALLBACK_STATUS_NORMAL</u>
NET_SDK_CALLBACK_TYPE_PROGRESS	Connection progress value.
NET_SDK_CALLBACK_TYPE_DATA	Data structures to be returned, which are different according to different commands (dwCommand) in <u>NET_DVR_StartRemoteConfig</u> .

dwBufLen

[OUT] Buffer size.

pUserData

[OUT] User data.

3.33 NET_DVR_STDXMLConfig

Transmit request URL with XML or JSON format to implement some typical functions.

API Definition

```

BOOL NET_DVR_STDXMLConfig(
    LONG                                     lUserID,
    const NET_DVR_XML_CONFIG_INPUT          *lpInputParam,
    NET_DVR_XML_CONFIG_OUTPUT               *lpOutputParam
);

```

Parameters

lUserID

[IN] Value returned by **NET_DVR_Login_V40** .

lpInputParam

[IN] Input parameters, refer to the structure **NET_DVR_XML_CONFIG_INPUT** for details.

lpOutputParam

[IN][OUT] Output parameters, refer to the structure **NET_DVR_XML_CONFIG_OUTPUT** for details.

Return Values

Return *TRUE* for success, and return *FALSE* for failure.

If *FALSE* is returned, you can call **NET_DVR_GetLastError** to get the error code.

Remarks

The input parameter **IpInputParam** and output parameter **IpOutputParam** are different when transmitting text protocol for implementing different functions, and each parameter corresponds to a component of text protocol, see the relations below:

Parameter of NET_DVR_STDXMLConfig		Component of Text Protocol
IpInputParam	IpRequestUrl (see in structure <u>NET_DVR_XML_CONFIG_INPUT</u>)	Method+URL E.g., GET /ISAPI/System/capabilities
	IpInBuffer (see in structure <u>NET_DVR_XML_CONFIG_INPUT</u>)	Request Message
IpOutputParam	IpOutBuffer (see in structure <u>NET_DVR_XML_CONFIG_OUTPUT</u>)	Response Message
	IpStatusBuffer (see in structure <u>NET_DVR_XML_CONFIG_OUTPUT</u>)	Response Message

3.34 NET_DVR_StopRemoteConfig

Disconnect the persistent connection to stop remote configuration, and release resources.

API Definition

```
BOOL NET_DVR_StopRemoteConfig(  
    LONG    lHandle  
);
```

Parameters

lHandle

[IN] Handle, which is returned by **NET_DVR_StartRemoteConfig**.

Return Values

Returns *TRUE* for success, and returns *FALSE* for failure.

If *FALSE* is returned, you can call **NET_DVR_GetLastError** to get the error code.

3.35 NET_DVR_UnBypassAlarmChan

Perform bypass recovered on the zone.

API Definition

```
BOOL NET_DVR_UnBypassAlarmChan(  
    LONG                lUserID,  
  
                                NET_DVR_ALARMIN_SETUP  
                                *lpInter  
) ;
```

Parameters

lUserID

[IN] Value returned by NET_DVR_Login_V40 .

lpInter

[IN] Zone parameters, see details in the structure NET_DVR_ALARMIN_SETUP .

Return Values

Return *TRUE* for success, and return *FALSE* for failure.

If *FALSE* is returned, you can call NET_DVR_GetLastError to get the error code.

See Also

NET_DVR_BypassAlarmChan

Appendix A. Appendixes

A.1 Data Structure

A.1.1 CHAR_ENCODE_CONVERT

Encoding type conversion callback function.

Callback Function Definition

```
typedef int (CALLBACK *CHAR_ENCODE_CONVERT) (  
    char            *pInput,  
    DWORD           dwInputLen,  
    DWORD           dwInEncodeType,  
    char            *pOutput,  
    DWORD           dwOutputLen,  
    DWORD           dwOutEncodeType  
);
```

Parameters

pInput

[IN] Input string, whose memory and size is applied and provided by the third-party platform

dwInputLen

[IN] Input buffer size.

dwInEncodeType

[IN] Encoding types of input string: 0-no encoding information, 1-GB2312 (Simplified Chinese), 2-GBK, 3-BIG5 (Traditional Chinese), 4-Shift_JIS (Japanese), 5-EUC-KR (Korean), 6-UTF-8, 7-ISO8859-1, 8-ISO8859-2, 9-ISO8859-3, ..., 21-ISO8859-15 (Western Europe).

pOutput

[OUT] Output string, whose memory is applied by the third-party platform.

dwOutputLen

[OUT] Output buffer size.

dwOutEncodeType

[OUT] Encoding tupes of output string: 0-no encoding information, 1-GB2312 (Simplified Chinese), 2-GBK, 3-BIG5 (Traditional Chinese), 4-Shift_JIS (Japanese), 5-EUC-KR (Korean), 6-UTF-8, 7-ISO8859-1, 8-ISO8859-2, 9-ISO8859-3, ..., 21-ISO8859-15 (Western Europe).

Return Values

Return -1 for failure, and return 0 for success.

A.1.2 DETECTOR_TYPE

Detection Type Enumeration

Enumeration Type	Macro Definition Value	Description
PANIC_BUTTON	0	Panic button
MAGNETIC_CONTACT	/	Magnetic switch
SMOKE_DETECTOR	/	Smoke detector
ACTIVE_INFRARED_DETECTOR	/	Active infrared detector
PASSIVE_INFRARED_DETECTOR	/	Passive infrared detector
GLASS_BREAK_DETECTOR	/	Glass break detector
VIBRATION_DETECTOR	/	Vibration detector
DUAL_TECHNOLOGY_PIR_DETECTOR	/	Dual-technology PIR detector
TRIPLE_TECHNOLOGY_PIR_DETECTOR	/	Triple-technology PIR detector
HUMIDITY_DETECTOR	/	Humidity detector
TEMPERATURE_DETECTOR	/	Temperature detector
COMBUSTIBLE_GAS_DETECTOR	/	Combustible gas detector
DYNAMIC_SWITCH	/	Dynamic switch
CONTROL_SWITCH	/	Control switch
SMART_LOCK	/	Smart lock
WATER_DETECTOR	/	Water detector
DISPLACEMENT_DETECTOR	/	Motion detector
SINGLE_INFRARED_DETECTOR	/	Open-Close detector
SINGLE_ZONE_MODULE	/	Wireless zone module
CURTAIN_INFRARED_DETECTOR	19	Curtain sensor PIR detector
DOORBELL_SWITCH	21	Doorbell switch
MEDICAL_HELP_BUTTON	/	Medical help button

Enumeration Type	Macro Definition Value	Description
OUTDOOR_DUAL_TECH	23	Outdoor dual-technology sensor
OTHER_DETECTOR	0xffff	Other detector

A.1.3 NET_DVR_ALARMHOST_CID_ALL_MINOR_TYPE

Minor type enumeration of CID alarms.

Enumeration Definition

```
enum NET_DVR_ALARMHOST_CID_ALL_MINOR_TYPE{
    CID_TYPE_MEDICAL_ALARM           = 1100,    //Personal Rescue Alarm
    CID_TYPE_MEDICAL_ALARM_RESET     = 3100,    //Personal Rescue Alarm
    Restored
    CID_TYPE_ALARM                   = 1103,    //Instant Alarm
    CID_TYPE_ALARM_RESET             = 3103,    //Instant Alarm Recovery
    CID_TYPE_FIRE_ALARM              = 1110,    //Fire Alarm
    CID_TYPE_FIRE_ALARM_RESET        = 3110,    //Fire Alarm Recovery
    CID_TYPE_ABDUCT_REPORT           = 1121,    //Duress Report
    CID_TYPE_SILENT_24               = 1122,    //24-hour Non-Voiced Alarm
    CID_TYPE_SILENT_24_RESET         = 3122,    //24-hour Non-Voiced Alarm
    Recovery
    CID_TYPE_AUDIO_24                = 1123,    //24-hour Voiced Alarm
    CID_TYPE_AUDIO_24_RESET          = 3123,    //24-hour Voiced Alarm
    Recovery
    CID_TYPE_AUXILIARY_24            = 1124,    //24-hour Aux Alarm
    CID_TYPE_AUXILIARY_24_RESET      = 3124,    //24-hour Aux Alarm
    Recovery
    CID_TYPE_SHOCK_24                = 1125,    //24-hour Vibration Alarm
    CID_TYPE_SHOCK_24_RESET          = 3125,    //24-hour Vibration Alarm
    Recovery
    CID_TYPE_TIMEOUT                 = 1126,    //Timeout Alarm
    CID_TYPE_TIMEOUT_RESET           = 3126,    //Timeout Alarm Recovery
    CID_TYPE_EMERGENCE_CALL_HELP     = 1129,    //Panic Alarm
    CID_TYPE_EMERGENCE_CALL_HELP_RESET = 3129,    //Panic Alarm Recovery
    CID_TYPE_PERIMETER_ALARM         = 1131,    //Perimeter Alarm
    CID_TYPE_PERIMETER_ALARM_RESET   = 3131,    //Perimeter Alarm Recovery
    CID_TYPE_INNET_ALARM             = 1132,    //Internal Delay Alarm
    CID_TYPE_INNET_ALARM_RESET       = 3132,    //Internal Delay Alarm
    Recovery
    CID_TYPE_ENTER_EXIT              = 1134,    //Delay Alarm
    CID_TYPE_ENTER_EXIT_RESET        = 3134,    //Delay Alarm Recovery
    CID_TYPE_DEVICE_OPEN             = 1137,    //Device Tampering Alarm
    CID_TYPE_DEVICE_OPEN_RESET       = 3137,    //Device Tampering Alarm
    Recovery
    CID_TYPE_ZONE_BUS_BREAK          = 1141,    //Bus Open Circuit Alarm
}
```

Device Network SDK (Security Control) Developer Guide

CID_TYPE_ZONE_BUS_BREAK_RESET	= 3141,	//Bus Open Circuit Alarm
Recovery		
CID_TYPE_ZONE_BUS_SHORT	= 1142,	//Bus Short Circuit Alarm
CID_TYPE_ZONE_BUS_SHORT_RESET	= 3142,	//Bus Short Circuit Alarm
Recovery		
CID_TYPE_AC_LOSS	= 1301,	//AC Power Off
CID_TYPE_AC_LOSS_RESET	= 3301,	//AC Recovery
CID_TYPE_LOW_BATT_VOL	= 1302,	//Low Battery Voltage
CID_TYPE_LOW_BATT_VOL_NORMAL	= 3302,	//Normal Battery Voltage
CID_TYPE_DEV_RESET	= 1305,	//Security Control Panel
Reset		
CID_TYPE_MBUS_MODEL_FAULT	= 1333,	//Extended Module Fault
CID_TYPE_MBUS_MODEL_RESET	= 3333,	//Extended Module Recovery
CID_TYPE_PRINTER_OFF	= 1336,	//Printer Offline
CID_TYPE_PRINTER_ON	= 3336,	//Printer Recovery
CID_TYPE_EXTEND_MODULE_VOL_LOW	= 1338,	//Extended Module Low
Battery Voltage		
CID_TYPE_EXTEND_MODULE_VOL_NORMAL	= 3338,	//Extended Module Normal
Battery Voltage		
CID_TYPE_EXTEND_MODULE_REMOVE	= 1341,	//Extended Module
Tampering Triggered		
CID_TYPE_EXTEND_MODULE_RECOVER	= 3341,	//Extended Module
Tampering Recovery		
CID_TYPE_EXTEND_MODULE_AC_LOSS	= 1342,	//Extended Module AC Power
Off		
CID_TYPE_EXTEND_MODULE_AC_LOSS_RESET	= 3342,	//Extended Module AC
Recovery		
CID_TYPE_LINE_LOSS_RESET	= 3354,	//Telephone Line
Disconnected		
CID_TYPE_BUS_LOSS	= 1382,	//Extended Bus Module
Offline		
CID_TYPE_BUS_LOSS_RESET	= 3382,	//Extended Bus Module
Offline Recovery		
CID_TYPE_SENSOR_TAMPER	= 1383,	//Zone Sensor Tampering
CID_TYPE_SENSOR_TAMPER_RESET	= 3383,	//Zone Sensor Tampering
Recovery		
CID_TYPE_DISARM	= 1401,	//Disarming
CID_TYPE_ARM	= 3401,	//Away Arming
CID_TYPE_DISARM_AUTO	= 1403,	//Auto Disarming
CID_TYPE_ARM_AUTO	= 3341,	//Auto Arming
CID_TYPE_CANCEL_ARM	= 1406,	//Clear Alarm
CID_TYPE_ARM_IMME	= 3408,	//Instant Arming
CID_TYPE_KEY_ZONE_ARM	= 1409,	//Key Zone Arming
CID_TYPE_KEY_ZONE_DISARM	= 3409,	//Key Zone Disarming
CID_TYPE_GUARD_STAY	= 3441,	//Stay Arming
CID_TYPE_FORCED_ARM	= 3442,	//Forced Arming
CID_TYPE_AUTOCTRL_TRIG_ON	= 1443,	//Timing Enable Trigger
CID_TYPE_AUTOCTRL_TRIG_OFF	= 3443,	//Timing Disable Trigger
CID_TYPE_AUTOGUARD_FAIL	= 1455,	//Auto-arming or auto-
disarming failed.		
CID_TYPE_AOPEN_TRIG_FAIL	= 1460,	//Timing Enable Trigger
Failed		

Device Network SDK (Security Control) Developer Guide

CID_TYPE_ACLOSE_TRIG_FAIL	= 1461,	//Timing Disable Trigger
Failed		
CID_TYPE_AUTOUNGUARD_FAIL	= 1462,	//Auto-disarming failed.
CID_TYPE_BYPASS	= 1570,	//Bypass
CID_TYPE_BYPASS_RESET	= 3570,	//Bypass Recovery
CID_TYPE_GROUP_BYPASS	= 1574,	//Subsystem Group Bypass
CID_TYPE_GROUP_BYPASS_RESET	= 3574,	//Subsystem Group Bypass
Recovery		
CID_TYPE_MANUAL_TEST_RPT	= 1601,	//Manual Test Reported
CID_TYPE_AUTO_TEST_RPT	= 1602,	//Scheduled Test Reported
CID_TYPE_LINE_TEST	= 1617,	//Telephone Link Test
CID_TYPE_ENTER_PROG	= 1627,	//Enter Programming
CID_TYPE_EXIT_PROG	= 1628,	//Exit Programming
CID_TYPE_SOFT_INSTAND	= 1810,	//Virtual Zone Emergency
Alarm		
CID_TYPE_SOFT_FIRE	= 1811,	//Virtual Zone Fire Alarm
CID_TYPE_SOFT_MOBS	= 1812,	//Virtual Zone Robbery
Alarm		
CID_TYPE_KEYPAD_LOCK	= 1862,	//Keyboard Locked
CID_TYPE_KEYPAD_UNLOCK	= 3862,	//Keyboard Unlocked
CID_TYPE_KEY_FAIL	= 1910,	//Keyboard Offline
CID_TYPE_KEY_RESET	= 3910,	//Keyboard Recovery
CID_TYPE_TRIGGER_FAIL	= 1911,	//Trigger of Keyboard Bus
Offline		
CID_TYPE_TRIGGER_RESET	= 3911,	//Trigger of Keyboard Bus
Recovery		
CID_TYPE_GPK_FAIL	= 1912,	//GP/K of Keyboard Bus
Offline		
CID_TYPE_GPK_RESET	= 3912,	//GP/K of Keyboard Bus
Recovery		
CID_TYPE_MODULE_FAIL	= 1913,	//MN/K of Keyboard Bus
Offline		
CID_TYPE_MODULE_RESET	= 3913,	//MN/K of Keyboard Bus
Recovery		
CID_TYPE_WIRELESS_DETECTOR_FAIL	= 1914,	//Wireless Detector Offline
CID_TYPE_WIRELESS_DETECTOR_RESET	= 3914,	//Wireless Detector
Offline Recovery		
CID_TYPE_WIRELESS_DETECTOR_BATTERY_LOW	= 1915,	//Wireless Detector Low
Battery Voltage		
CID_TYPE_WIRELESS_DETECTOR_BATTERY_NORML	= 3915,	//Wireless Detector Normal
Battery Voltage		
CID_TYPE_EXTEND_MODULE_LOSS	= 1916,	//Extended Module Offline
CID_TYPE_EXTEND_MODULE_LOSS_RESET	= 3916,	//Extended Module Offline
Recovery		
CID_TYPE_WRIELESS_NET_FAULT	= 1920,	//Wireless Network Fault
CID_TYPE_WRIELESS_NET_RESET	= 3920,	//Wireless Network Fault
Recovery		
CID_TYPE_SIM_FAULT	= 1921,	//SIM Card Exception
CID_TYPE_SIM_RESET	= 3921,	//SIM Card Exception
Recovery		
CID_TYPE_WIFI_ABNORMAL	= 1922,	//Wi-Fi Communication
Exception		


```

    CID_TYPE_WIFI_NORMAL                = 3922,    //WIFI Communication
Exception Recovery
    CID_TYPE_RF_ABNORMAL                = 1923,    //RF Signal Exception
    CID_TYPE_RF_NORMAL                  = 3923,    //RF Signal Normal
    CID_TYPE_DATE_TRAFFIC_OVERFLOW      = 1924,    //Traffic Overflow
    CID_TYPE_IPADDR_CONFLICT            = 1930,    //IP Address Conflicted
    CID_TYPE_IPADDR_NORMAL               = 3930,    //IP Address Normal
    CID_TYPE_ETHERNET_BROKEN             = 1931,    //Wired Network Fault
    CID_TYPE_ETHERNET_NORMAL             = 3931,    //Wired Network Recovery
    CID_TYPE_MOTION_DETECT_START        = 1940,    //Motion Detection Alarm
Started
    CID_TYPE_MOTION_DETECT_STOP         = 3940,    //Motion Detection Alarm
Ended
    CID_TYPE_MASK_ALARM_START           = 1941,    //Video Tampering
Detection Alarm Started
    CID_TYPE_MASK_ALARM_STOP            = 3941,    //Video Tampering
Detection Alarm Stopped
    CID_TYPE_VI_LOST_START              = 1942,    //Video Loss
    CID_TYPE_VI_LOST_STOP               = 3942,    //Video Signal Recovered
    CID_TYPE_VS_MISMATCH                = 1943,    //Input/Output Video
Standard Mismatch
    CID_TYPE_VS_MATCH                   = 3943,    //Input/Output Video
Standard Recovery
    CID_TYPE_VI_EXCEPTION               = 1944,    //Video Input Exception
    CID_TYPE_VI_NORMAL                  = 3944,    //Video Input Recovery
    CID_TYPE_HD_FULL                    = 1945,    //HDD Full
    CID_TYPE_HD_FREE                    = 3945,    //HDD is free.
    CID_TYPE_HD_ERROR                   = 1946,    //HDD Exception
    CID_TYPE_HD_RESET                   = 3946,    //HDD Recovery
    CID_TYPE_PIC_SEND_FAILED            = 1947,    //Uploading picture failed.
    CID_TYPE_ZONE_BUS_SEARCH             = 1970,    //Bus Search
    CID_TYPE_ZONE_BUS_REGIST            = 1971,    //Bus Registration
    CID_TYPE_GUARD_SINGLE_ARM            = 1973,    //Single-Zone Disarming
    CID_TYPE_GUARD_SINGLE_DISARM        = 3973,    //Single-Zone Arming
    CID_TYPE_ZONE_ASSOCIATED_DETECTOR_ADD = 1975,    //Detector Added
    CID_TYPE_ZONE_ASSOCIATED_DETECTOR_DEL = 3975,    //Detector Deleted
    CID_TYPE_ZONE_CONSULT               = 1976,    //Service Inquiry
    CID_TYPE_ZONE_CONSULT_STOP           = 3976,    //Service Inquiry Ended
    CID_TYPE_EXTEND_MODULE_DEL           = 1977,    //Deleted Extension Module
    CID_TYPE_EXTEND_MODULE_ADD           = 3977,    //Added Extension Module
    CID_TYPE_WIRELESS_REPEATER_DEL       = 1978,    //Deleted Wireless Repeater
    CID_TYPE_WIRELESS_REPEATER_ADD       = 3978,    //Added Wireless Repeater
    CID_TYPE_WIRELESS_SIREN_DEL          = 1979,    //Deleted Wireless Siren
    CID_TYPE_WIRELESS_SIREN_ADD          = 3979,    //Added Wireless Siren
}NET_DVR_ALARMHOST_CID_ALL_MINOR_TYPE;

```

A.1.4 NET_DVR_ALARMHOST_LOG_RET

Table A-1 Structure about Result of Searching for Security Control Panel Logs

Member	Data Type	Description
struLogTime	<i>NET_DVR_TIME</i>	Log time.
sUserName	Array [BYTE]	Operator user. The maximum size is 32 bytes (the value of the macro definition "NAME_LEN").
struIPAddr	<i>NET_DVR_IPADDR_UNION</i>	Operator IP address.
wMajorType	WORD	Major type, see details in <i>Table 4-2</i> .
wMinorType	WORD	For minor types of alarms, see details in <i>Table 4-3</i> ; for minor types of exceptions, see details in <i>Table 4-4</i> ; for minor types of operations, see details in <i>Table 4-5</i> ; for minor types of events, see details in <i>Table 4-6</i> .
wParam	WORD	Operation parameter.
byRes	Array [BYTE]	Reserved. The maximum size is 10 bytes.
dwInfoLen	DWORD	Length of the description information.
sInfo	Array [char]	Description information. The maximum size is 11840 bytes (the value of the macro definition "LOG_INFO_LEN").

Table A-2 Detailed Definitions of Different Major Types

Macro Definition	Macro Definition Value	Description
ALARMHOST_MAJOR_ALARM	1	Alarm
ALARMHOST_MAJOR_EXCEPTION	2	Exception

Macro Definition	Macro Definition Value	Description
ALARMHOST_MAJOR_OPERATE	3	Operation
ALARMHOST_MAJOR_EVENT	4	Event

Table A-3 Minor Types of Alarms

Macro Definition	Macro Definition Value	Description
MINOR_SHORT_CIRCUIT	0x01	Short Circuit Alarm
MINOR_BROKEN_CIRCUIT	0x02	Open Circuit Alarm
MINOR_ALARM_RESET	0x03	Alarm Reset
MINOR_ALARM_NORMAL	0x04	Return to Normal
MINOR_PASSWORD_ERROR	0x05	Incorrect Password (Three Failed Attempts)
MINOR_ID_CARD_ILLEGALLY	0x06	Invalid Card ID
MINOR_KEYPAD_REMOVE	0x07	Keypad Tampered
MINOR_KEYPAD_REMOVE_RESTORE	0x08	Keypad Restored
MINOR_DEV_REMOVE	0x09	Device Tampered
MINOR_DEV_REMOVE_RESTORE	0x0a	Device Restored
MINOR_BELOW_ALARM_LIMIT1	0x0b	Sensor Value is Lower than Alarm Limit Value 1
MINOR_BELOW_ALARM_LIMIT2	0x0c	Sensor Value is Lower than Alarm Limit Value 2
MINOR_BELOW_ALARM_LIMIT3	0x0d	Sensor Value is Lower than Alarm Limit Value 3
MINOR_BELOW_ALARM_LIMIT4	0x0e	Sensor Value is Lower than Alarm Limit Value 4
MINOR_ABOVE_ALARM_LIMIT1	0x0f	Sensor Value is Higher than Alarm Limit Value 1
MINOR_ABOVE_ALARM_LIMIT2	0x10	Sensor Value is Higher than Alarm Limit Value 2
MINOR_ABOVE_ALARM_LIMIT3	0x11	Sensor Value is Higher than Alarm Limit Value 3

Macro Definition	Macro Definition Value	Description
MINOR_ABOVE_ALARM_LIMIT4	0x12	Sensor Value is Higher than Alarm Limit Value 4
MINOR_URGENCYBTN_ON	0x13	Panic Button Triggered
MINOR_URGENCYBTN_OFF	0x14	Panic Button Restored
MINOR_VIRTUAL_DEFENCE_BANDIT	0x15	Virtual Zone Burglary Alarm
MINOR_VIRTUAL_DEFENCE_FIRE	0x16	Virtual Zone Fire Alarm
MINOR_VIRTUAL_DEFENCE_URGENT	0x17	Virtual Zone Panic Alarm
MINOR_ALARMHOST_MOTDET_START	0x18	Motion Detection Alarm Started
MINOR_ALARMHOST_MOTDET_STOP	0x19	Motion Detection Alarm Stopped
MINOR_ALARMHOST_HIDE_ALARM_START	0x1a	Device Blocked
MINOR_ALARMHOST_HIDE_ALARM_STOP	0x1b	Device Blocking Alarm Restored
MINOR_ALARMHOST_UPS_ALARM	0x1c	UPS Alarm
MINOR_ALARMHOST_ELECTRICITY_METER_ALARM	0x1d	Coulombmeter Alarm
MINOR_ALARMHOST_SWITCH_POWER_ALARM	0x1e	Switch Power Supply Alarm
MINOR_ALARMHOST_GAS_DETECT_SYS_ALARM	0x1f	Gas Detection Alarm
MINOR_ALARMHOST_TRANSFORMER_TEMPRATURE_ALARM	0x20	Transformer Temperature Alarm
MINOR_ALARMHOST_TEMP_HUMI_ALARM	0x21	Temperature and Humidity Sensor Alarm
MINOR_ALARMHOST_UPS_ALARM_RESTORE	0x22	UPS Alarm Restored
MINOR_ALARMHOST_ELECTRICITY_METER_ALARM_RESTORE	0x23	Coulombmeter Alarm Restored
MINOR_ALARMHOST_SWITCH_POWER_ALARM_RESTORE	0x24	Switch Power Supply Alarm Restored

Macro Definition	Macro Definition Value	Description
MINOR_ALARMHOST_GAS_DETECT_SYS_ALARM_RESTORE	0x25	Gas Detection Alarm Restored
MINOR_ALARMHOST_TRANSFORMER_TEMPRATURE_ALARM_RESTORE	0x26	Transformer Temperature Alarm Restored
MINOR_ALARMHOST_TEMP_HUMI_ALARM_RESTORE	0x27	Temperature-Humidity Sensor Alarm Restored
MINOR_ALARMHOST_WATER_LEVEL_SENSOR_ALARM	0x28	Flood Sensor Alarm
MINOR_ALARMHOST_WATER_LEVEL_SENSOR_ALARM_RESTORE	0x29	Flood Sensor Restored
MINOR_ALARMHOST_DUST_NOISE_ALARM	0x2a	Dust and Noise Sensor Alarm
MINOR_ALARMHOST_DUST_NOISE_ALARM_RESTORE	0x2b	Dust and Noise Sensor Alarm Restored
MINOR_ALARMHOST_ENVIRONMENTAL_LOGGER_ALARM	0x2c	Environmental Data Collector Alarm
MINOR_ALARMHOST_ENVIRONMENTAL_LOGGER_ALARM_RESTORE	0x2d	Environmental Data Collector Restored
MINOR_ALARMHOST_TRIGGER_TAMPER	0x2e	Detector Tampered
MINOR_ALARMHOST_TRIGGER_TAMPER_RESTORE	0x2f	Detector Restored
MINOR_ALARMHOST_EMERGENCY_CALL_HELP_ALARM	0x30	Panic Alarm
MINOR_ALARMHOST_EMERGENCY_CALL_HELP_ALARM_RESTORE	0x31	Panic Alarm Restored
MINOR_ALARMHOST_CONSULTING_ALARM	0x32	Consultation Alarm
MINOR_ALARMHOST_CONSULTING_ALARM_RESTORE	0x33	Consultation Alarm Restored
MINOR_ZONE_MODULE_REMOVE	0x34	Zone Module Tampered
MINOR_ZONE_MODULE_RESET	0x35	Zone Module Tampering Reset

Macro Definition	Macro Definition Value	Description
MINOR_WIRELESS_OUTPUT_MODULE_REMOVE	0x48	Wireless Output Module Tampered
MINOR_WIRELESS_OUTPUT_MODULE_RESET	0x49	Wireless Output Module Tampering Restored
MINOR_WIRELESS_REPEATER_MODULE_REMOVE	0x4a	Wireless Repeater Tampered
MINOR_WIRELESS_REPEATER_MODULE_RESET	0x4b	Wireless Repeater Tampering Restored
MINOR_WIRELESS_SIREN_MODULE_REMOVE	0x4c	Wireless Siren Tampered
MINOR_WIRELESS_SIREN_MODULE_RESET	0x4d	Wireless Siren Tampering Restored

Table A-4 Minor Types of Exceptions

Macro Definition	Macro Definition Value	Description
MINOR_POWER_ON	0x01	Power On
MINOR_POWER_OFF	0x02	Power Off
MINOR_WDT_RESET	0x03	WDT Reset
MINOR_LOW_BATTERY_VOLTAGE	0x04	Low Battery Voltage
MINOR_AC_LOSS	0x05	AC Power Disconnected
MINOR_AC_RESTORE	0x06	AC Power Restored
MINOR_RTC_EXCEPTION	0x07	RTC Real-Time Clock Exception
MINOR_NETWORK_CONNECT_FAILURE	0x08	Network Disconnected
MINOR_NETWORK_CONNECT_RESTORE	0x09	Network Connected
MINOR_TEL_LINE_CONNECT_FAILURE	0x0a	Telephone Line Disconnected
MINOR_TEL_LINE_CONNECT_RESTORE	0x0b	Telephone Line Connected
MINOR_EXPANDER_BUS_LOSS	0x0c	Bus Expander Disconnected
MINOR_EXPANDER_BUS_RESTORE	0x0d	Bus Expander Connected
MINOR_KEYPAD_BUS_LOSS	0x0e	Keypad Expander Disconnected

Macro Definition	Macro Definition Value	Description
MINOR_KEYPAD_BUS_RESTORE	0x0f	Keypad Expander Connected
MINOR_SENSOR_FAILURE	0x10	Analog Sensor Fault
MINOR_SENSOR_RESTORE	0x11	Analog Sensor Restored
MINOR_RS485_CONNECT_FAILURE	0x12	RS-485 Channel Disconnected
MINOR_RS485_CONNECT_RESTORE	0x13	RS-485 Channel Connected
MINOR_BATTERT_VOLTAGE_RESTORE	0x14	Battery Voltage Restored
MINOR_WIRED_NETWORK_ABNORMAL	0x15	Wired Network Exception
MINOR_WIRED_NETWORK_RESTORE	0x16	Wired Network Restored
MINOR_GPRS_ABNORMAL	0x17	GPRS Exception
MINOR_GPRS_RESTORE	0x18	GPRS Restored
MINOR_3G_ABNORMAL	0x19	3G Network Exception
MINOR_3G_RESTORE	0x1a	3G Network Restored
MINOR_SIM_CARD_ABNORMAL	0x1b	SIM Card Exception
MINOR_SIM_CARD_RESTORE	0x1c	SIM Card Restored
MINOR_ALARMHOST_VI_LOST	0x1d	Video Loss
MINOR_ALARMHOST_ILLEGAL_ACCESS	0x1e	Illegal Login
MINOR_ALARMHOST_HD_FULL	0x1f	HDD Full
MINOR_ALARMHOST_HD_ERROR	0x20	HDD Error
MINOR_ALARMHOST_DCD_LOST	0x21	MODEM Disconnected (Reserved)
MINOR_ALARMHOST_IP_CONFLICT	0x22	IP Address Conflicted
MINOR_ALARMHOST_NET_BROKEN	0x23	Network Disconnected
MINOR_ALARMHOST_REC_ERROR	0x24	Recording Error
MINOR_ALARMHOST_VI_EXCEPTION	0x25	Video Input Exception (Only for Analog Channel)
MINOR_ALARMHOST_FORMAT_HDD_ERROR	0x26	Remote HDD Formatting Failed
MINOR_ALARMHOST_USB_ERROR	0x27	USB Communication Error

Macro Definition	Macro Definition Value	Description
MINOR_ALARMHOST_USB_RESTORE	0x28	USB Communication Error Restored
MINOR_ALARMHOST_PRINT_ERROR	0x29	Printer Error
MINOR_ALARMHOST_PRINT_RESTORE	0x2a	Printer Error Restored
MINOR_SUBSYSTEM_COMMUNICATION_ERROR	0x2b	Sub-Board Communication Error
MINOR_ALARMHOST_IPC_NO_LINK	0x2c	Network Camera Disconnected
MINOR_ALARMHOST_IPC_IP_CONFLICT	0x2d	Network Camera IP Address Conflicted
MINOR_ALARMHOST_VI_MISMATCH	0x2e	Video Standard Mismatches
MINOR_ALARMHOST_MCU_RESTART	0x2f	MCU Restarted
MINOR_ALARMHOST_GPRS_MODULE_FAULT	0x30	GPRS Module Fault
MINOR_ALARMHOST_TELEPHONE_MODULE_FAULT	0x31	Telephone Module Fault
MINOR_ALARMHOST_WIFI_ABNORMAL	0x32	Wi-Fi Exception
MINOR_ALARMHOST_WIFI_RESTORE	0x33	Wi-Fi Restored
MINOR_ALARMHOST_RF_ABNORMAL	0x34	RF Exception
MINOR_ALARMHOST_RF_RESTORE	0x35	RF Restored
MINOR_ALARMHOST_DETECTOR_ONLINE	0x36	Detector Connected
MINOR_ALARMHOST_DETECTOR_OFFLINE	0x37	Detector Disconnected
MINOR_ALARMHOST_DETECTOR_BATTERY_NORMAL	0x38	Detector Battery Restored
MINOR_ALARMHOST_DETECTOR_BATTERY_LOW	0x39	Detector Battery Low
MINOR_ALARMHOST_DATA_TRAFFIC_OVERFLOW	0x3a	Cellular Network Data Exceeded
MINOR_ZONE_MODULE_LOSS	0x3b	Zone Module Disconnected
MINOR_ZONE_MODULE_RESTORE	0x3c	Zone Module Connected

Macro Definition	Macro Definition Value	Description
MINOR_ALARMHOST_WIRELESS_OUTPUT_LOSS	0x3d	Wireless Output Module Offline
MINOR_ALARMHOST_WIRELESS_OUTPUT_RESTORE	0x3e	Wireless Output Module Online
MINOR_ALARMHOST_WIRELESS_REPEATER_LOSS	0x3f	Wireless Repeater Offline
MINOR_ALARMHOST_WIRELESS_REPEATER_RESTORE	0x40	Wireless Repeater Online
MINOR_TRIGGER_MODULE_LOSS	0x41	Trigger Module Disconnected
MINOR_TRIGGER_MODULE_RESTORE	0x42	Trigger Module Connected
MINOR_WIRELESS_SIREN_LOSS	0x43	Wireless Siren Offline
MINOR_WIRELESS_SIREN_RESTORE	0x44	Wireless Siren Online

Table A-5 Minor Types of Operations

Macro Definition	Macro Definition Value	Description
MINOR_GUARD	0x01	Normal Arming
MINOR_UNGUARD	0x02	Normal Disarming
MINOR_BYPASS	0x03	Bypass
MINOR_DURESS_ACCESS	0x04	Duress
MINOR_ALARMHOST_LOCAL_REBOOT	0x05	Local Reboot
MINOR_ALARMHOST_REMOTE_REBOOT	0x06	Remote Reboot
MINOR_ALARMHOST_LOCAL_UPGRADE	0x07	Local Upgrade
MINOR_ALARMHOST_REMOTE_UPGRADE	0x08	Remote Upgrade
MINOR_RECOVERY_DEFAULT_PARAM	0x09	Restore Default Settings
MINOR_ALARM_OUTPUT	0x0a	Alarm Output Control
MINOR_ACCESS_OPEN	0x0b	Access Control: Open
MINOR_ACCESS_CLOSE	0x0c	Access Control: Closed
MINOR_SIREN_OPEN	0x0d	Siren: On

Macro Definition	Macro Definition Value	Description
MINOR_SIREN_CLOSE	0x0e	Siren: Off
MINOR_MOD_ZONE_CONFIG	0x0f	Edit Zone Settings
MINOR_MOD_ALARMOUT_CONFIG	0x10	Edit Alarm Output Settings
MINOR_MOD_ANALOG_CONFIG	0x11	Edit Analog Sensor Settings
MINOR_RS485_CONFIG	0x12	Edit RS-485 Channel Settings
MINOR_PHONE_CONFIG	0x13	Edit Dialing Settings
MINOR_ADD_ADMIN	0x14	Added Administrator
MINOR_MOD_ADMIN_PW	0x15	Edited Administrator Password
MINOR_DEL_ADMIN	0x16	Deleted Administrator
MINOR_ADD_NETUSER	0x17	Added DVR/NVR Operator
MINOR_MOD_NETUSER_PW	0x18	Edited DVR/NVR Operator Password
MINOR_DEL_NETUSER	0x19	Deleted DVR/NVR Operator
MINOR_ADD_OPERATORUSER	0x1a	Added Camera Operator
MINOR_MOD_OPERATORUSER_PW	0x1b	Edited Camera Operator Password
MINOR_DEL_OPERATORUSER	0x1c	Deleted Camera Operator
MINOR_ADD_KEYPADUSER	0x1d	Added Keypad/Card Reader User
MINOR_DEL_KEYPADUSER	0x1e	Deleted Keyboard/Card Reader User
MINOR_REMOTEUSER_LOGIN	0x1f	Remote Login
MINOR_REMOTEUSER_LOGOUT	0x20	Remote Logout
MINOR_REMOTE_GUARD	0x21	Remote Arming
MINOR_REMOTE_UNGUARD	0x22	Remote Disarming
MINOR_MOD_HOST_CONFIG	0x23	Edited Control Panel Settings
MINOR_RESTORE_BYPASS	0x24	Bypass Restored
MINOR_ALARMOUT_OPEN	0x25	Turned on Output
MINOR_ALARMOUT_CLOSE	0x26	Turned off Output

Macro Definition	Macro Definition Value	Description
MINOR_MOD_SUBSYSTEM_PARAM	0x27	Edited Partition Parameters
MINOR_GROUP_BYPASS	0x28	Group Bypass
MINOR_RESTORE_GROUP_BYPASS	0x29	Group Bypass Restored
MINOR_MOD_GRPS_PARAM	0x2a	Edited GPRS Parameters
MINOR_MOD_NET_REPORT_PARAM	0x2b	Edited Network Report Settings
MINOR_MOD_REPORT_MOD	0x2c	Edited Uploading Mode
MINOR_MOD_GATEWAY_PARAM	0x2d	Edited Access Control Settings
MINOR_ALARMHOST_REMOTE_START_REC	0x2e	Remote: Started Recording
MINOR_ALARMHOST_REMOTE_STOP_REC	0x2f	Remote: Stopped Recording
MINOR_ALARMHOST_START_TRANS_CHAN	0x30	Transparent Transmission Started
MINOR_ALARMHOST_STOP_TRANS_CHAN	0x31	Transparent Transmission Stopped
MINOR_ALARMHOST_START_VT	0x32	Two-way Audio Started
MINOR_ALARMHOST_STOP_VTM	0x33	Two-way Audio Terminated
MINOR_ALARMHOST_REMOTE_PLAYBYFILE	0x34	Remote: Playback or Downloaded by File
MINOR_ALARMHOST_REMOTE_PLAYBYTIME	0x35	Remote: Playback by Time
MINOR_ALARMHOST_REMOTE_PTZCTRL	0x36	Remote: PTZ Control
MINOR_ALARMHOST_REMOTE_FORMAT_HDD	0x37	Remote: Formatted HDD
MINOR_ALARMHOST_REMOTE_LOCKFILE	0x38	Remote: Locked File
MINOR_ALARMHOST_REMOTE_UNLOCKFILE	0x39	Remote: Unlocked File
MINOR_ALARMHOST_REMOTE_CFGFILE_OUTPUT	0x3a	Remote: Exported Configuration Files
MINOR_ALARMHOST_REMOTE_CFGFILE_INPUT	0x3b	Remote: Imported Configuration Files

Macro Definition	Macro Definition Value	Description
MINOR_ALARMHOST_REMOTE_RECFILE_OUTPUT	0x3c	Remote: Exported Video File
MINOR_ALARMHOST_STAY_ARM	0x3d	Stay Arming
MINOR_ALARMHOST_QUICK_ARM	0x3e	Instant Arming
MINOR_ALARMHOST_AUTOMATIC_ARM	0x3f	Automatic Arming
MINOR_ALARMHOST_AUTOMATIC_DISARM	0x40	Automatic Disarming
MINOR_ALARMHOST_KEYSWITCH_ARM	0x41	Key Zone Arming
MINOR_ALARMHOST_KEYSWITCH_DISARM	0x42	Key Zone Disarming
MINOR_ALARMHOST_CLEAR_ALARM	0x43	Alarm Cleared
MINOR_ALARMHOST_MOD_FAULT_CFG	0x44	Edited System Fault Settings
MINOR_ALARMHOST_MOD_EVENT_TRIGGER_ALARMOUT_CFG	0x45	Edited Event Alarm Output Settings
MINOR_ALARMHOST_SEARCH_EXTERNAL_MODULE	0x46	Searched for External Module
MINOR_ALARMHOST_REGISTER_EXTERNAL_MODULE	0x47	Re-registered External Module
MINOR_ALARMHOST_CLOSE_KEYBOARD_ALARM	0x48	Disabled Keypad Beep
MINOR_ALARMHOST_MOD_3G_PARAM	0x49	Edited 3G Parameters
MINOR_ALARMHOST_MOD_PRINT_PARAM	0x4a	Edited Printer Parameters
MINOR_SD_CARD_FORMAT	0x4b	Formatted SD Card
MINOR_SUBSYSTEM_UPGRADE	0x4c	Upgraded Sub-board
MINOR_ALARMHOST_PLAN_ARM_CFG	0x4d	Arming/Disarming Schedule Configuration
MINOR_ALARMHOST_PHONE_ARM	0x4e	SMS Arming
MINOR_ALARMHOST_PHONE_STAY_ARM	0x4f	SMS Stay Arming
MINOR_ALARMHOST_PHONE_QUICK_ARM	0x50	SMS Instant Arming

Macro Definition	Macro Definition Value	Description
MINOR_ALARMHOST_PHONE_DISARM	0x51	SMS Disarming
MINOR_ALARMHOST_PHONE_CLEAR_ALARM	0x52	SMS Alarm Cleared
MINOR_ALARMHOST_ALLOWLIST_CFG	0x53	Allowlist Settings
MINOR_ALARMHOST_TIME_TRIGGER_CFG	0x54	Enabled/Disabled Trigger Configuration by Schedule
MINOR_ALARMHOST_CAPTRUE_CFG	0x55	Capture Settings
MINOR_ALARMHOST_TAMPER_CFG	0x56	Zone Tamper-Proof Settings
MINOR_ALARMHOST_REMOTE_KEYPAD_UPGRADE	0x57	Remote: Upgraded Keypad
MINOR_ALARMHOST_ONETOUCH_AWAY_ARMING	0x58	One-Touch Away Arming
MINOR_ALARMHOST_ONETOUCH_STAY_ARMING	0x59	One-Touch Stay Arming
MINOR_ALARMHOST_SINGLE_PARTITION_ARMING_OR_DISARMING	0x5a	Single-Zone Arming/Disarming
MINOR_ALARMHOST_CARD_CONFIGURATION	0x5b	Card Settings
MINOR_ALARMHOST_CARD_ARMING_OR_DISARMING	0x5c	Arming/Disarming by Card
MINOR_ALARMHOST_EXPENDING_NETCENTER_CONFIGURATION	0x5d	Extension Network Center Settings
MINOR_ALARMHOST_NETCARD_CONFIGURATION	0x5e	NIC Settings
MINOR_ALARMHOST_DDNS_CONFIGURATION	0x5f	DDNS Settings
MINOR_ALARMHOST_RS485BUS_CONFIGURATION	0x60	RS-485 Bus Settings
MINOR_ALARMHOST_RS485BUS_RE_REGISTRATION	0x61	RS-485 Bus Re-registration
MINOR_ALARMHOST_REMOTE_OPEN_ELECTRIC_LOCK	0x62	Remote: Unlocked

Macro Definition	Macro Definition Value	Description
MINOR_ALARMHOST_REMOTE_CLOSE_ELECTRIC_LOCK	0x63	Remote: Locked
MINOR_ALARMHOST_LOCAL_OPEN_ELECTRIC_LOCK	0x64	Local: Unlocked
MINOR_ALARMHOST_LOCAL_CLOSE_ELECTRIC_LOCK	0x65	Local: Locked
MINOR_ALARMHOST_OPEN_ALARM_LAMP	0x66	Remote: Turned On Alarm Lamp
MINOR_ALARMHOST_CLOSE_ALARM_LAMP	0x67	Remote: Turned Off Strobe
MINOR_ALARMHOST_TEMPORARY_PASSWORD	0x68	Operation Record of Temporary Password
MINOR_ALARMHOST_ONEKEY_AWAY_ARM	0x69	One-Touch Away Arming
MINOR_ALARMHOST_ONEKEY_STAY_ARM	0x6a	One-Touch Stay Arming
MINOR_ALARMHOST_SINGLE_ZONE_ARM	0x6b	Single-Zone Arming
MINOR_ALARMHOST_SINGLE_ZONE_DISARM	0x6c	Single-Zone Disarming
MINOR_ALARMHOST_HIDDNS_CONFIG	0x6d	HiDDNS Settings
MINOR_ALARMHOST_REMOTE_KEYBOARD_UPDATA	0x6e	Remote: Upgraded Keypad
MINOR_ALARMHOST_ZONE_ADD_DETECTOR	0x6f	Added Detector
MINOR_ALARMHOST_ZONE_DELETE_DETECTOR	0x70	Deleted Detector
MINOR_ALARMHOST_QUERY_DETECTOR_SIGNAL	0x71	Checked Detector Signal Strength on Security Control Panel
MINOR_ALARMHOST_QUERY_DETECTOR_BATTERY	0x72	Checked Detector Remaining Battery on Security Control Panel
MINOR_ALARMHOST_SET_DETECTOR_GUARD	0x73	Detector Arming

Macro Definition	Macro Definition Value	Description
MINOR_ALARMHOST_SET_DETECTOR_UNGUARD	0x74	Detector Disarming
MINOR_ALARMHOST_SET_WIFI_PARAMETER	0x75	Wi-Fi Settings
MINOR_ALARMHOST_OPEN_VOICE	0x76	Audio On
MINOR_ALARMHOST_CLOSE_VOICE	0x77	Mute
MINOR_ALARMHOST_ENABLE_FUNCTION_KEY	0x78	Enabled Function Key
MINOR_ALARMHOST_DISABLE_FUNCTION_KEY	0x79	Disabled Panel Function Button
MINOR_ALARMHOST_READ_CARD	0x7a	Swiped Patrol Card
MINOR_ALARMHOST_START_BROADCAST	0x7b	Start Audio Broadcast
MINOR_ALARMHOST_STOP_BROADCAST	0x7c	Stop Audio Broadcast
MINOR_ALARMHOST_REMOTE_ZONE_MODULE_UPGRADE	0x7d	Upgrade Zone Module Remotely
MINOR_ALARMHOST_NETWORK_MODULE_EXTEND	0x7e	Network Module Settings
MINOR_ALARMHOST_ADD_CONTROLLER	0x7f	Added Keyfob User
MINOR_ALARMHOST_DELETE_CONTROLLER	0x80	Deleted Keyfob User
MINOR_ALARMHOST_REMOTE_NETWORKMODULE_UPGRADE	0x81	Upgrade Network Module Remotely
MINOR_ALARMHOST_WIRELESS_OUTPUT_ADD	0x82	Registered Wireless Output Module
MINOR_ALARMHOST_WIRELESS_OUTPUT_DEL	0x83	Deleted Wireless Output Module
MINOR_ALARMHOST_WIRELESS_REPEATER_ADD	0x84	Registered Wireless Repeater
MINOR_ALARMHOST_WIRELESS_REPEATER_DEL	0x85	Deleted Wireless Repeater
MINOR_ALARMHOST_PHONELIST_CFG	0x86	Phone List Settings

Macro Definition	Macro Definition Value	Description
MINOR_ALARMHOST_RF_SIGNAL_CHECK	0x87	RF Signal Search
MINOR_ALARMHOST_USB_UPGRADE	0x88	Upgrade via USB
MINOR_ALARMHOST_DOOR_TIME_REMINDER_CFG	0x89	Scheduled Magnetic Contact Prompt Settings
MINOR_ALARMHOST_WIRELESS_SIREN_ADD	0x8a	Registered Wireless Siren
MINOR_ALARMHOST_WIRELESS_SIREN_DEL	0x8b	Deleted Wireless Siren
MINOR_ALARMHOST_LOCAL_SET_DEVICE_ACTIVE	0xf0	Activated Device Locally
MINOR_ALARMHOST_REMOTE_SET_DEVICE_ACTIVE	0xf1	Activated Device Remotely
MINOR_ALARMHOST_LOCAL_PARA_FACTORY_DEFAULT	0xf2	Restored Factory Settings Locally
MINOR_ALARMHOST_REMOTE_PARA_FACTORY_DEFAULT	0xf3	Restored Factory Settings Remotely
MINOR_ALARMHOST_TIME_ZONE_CFG	/	Edited Time Zone
MINOR_ALARMHOST_NTP_START_AND_PARAMETERS_CFG	/	Enabled NTP and Edited Parameters
MINOR_ALARMHOST_DST_START_AND_PARAMETERS_CFG	/	Enabled DST and Edited Parameters
MINOR_ALARMHOST_DEVINFO_CFG	/	Device Information Settings
MINOR_ALARMHOST_VIDEO_OVERLAP_CFG	/	Video Overwriting Settings
MINOR_ALARMHOST_SSH_CFG	/	SSH Enabling Settings
MINOR_ALARMHOST_PASSWORD_MANAGE_CFG	/	Password Management Settings
MINOR_ALARMHOST_RESTORE_DEFAULT_PARAMETERS	/	Restore Default Parameters
MINOR_ALARMHOST_RESTORECOMPLETELY_DEFAULT_PARAMETERS	/	Completely Restore Default Parameters

Macro Definition	Macro Definition Value	Description
MINOR_ALARMHOST_AUDIO_AUTO_DETECT_CFG	/	Automatic Detection Settings
MINOR_ALARMHOST_AUDIO_MANUAL_DETECT_CFG	/	Manual Detection
MINOR_ALARMHOST_NET_PARAMETERS_CFG	/	Network Parameter Settings
MINOR_ALARMHOST_MTU_CFG	/	MTU Settings
MINOR_ALARMHOST_PORT_CFG	/	Port Settings
MINOR_ALARMHOST_DEFAULT_ROUTER_CFG	/	Default Route Settings
MINOR_ALARMHOST_DNS_PARAMETERS_CFG	/	DNS Settings
MINOR_ALARMHOST_UNPNP_PARAMETERS_CFG	/	UNPNP Settings
MINOR_ALARMHOST_SIP_PARAMETERS_CFG	/	SIP Settings
MINOR_ALARMHOST_FLOW_LIMIT_CFG	/	Data Usage Limit Settings
MINOR_ALARMHOST_APN_PARAMETERS_CFG	/	APN Settings
MINOR_ALARMHOST_MESSAGE_TELEPHONENO_CFG	/	SMS Phone Number Settings
MINOR_ALARMHOST_EZVIZ_PARAMETERS_CFG	/	EZVIZ Settings
MINOR_ALARMHOST_ISUP_PARAMETERS_CFG	/	ISUP Settings
MINOR_ALARMHOST_SIP_SWITCH_CFG	/	Switched between SIP Standard and Private Protocol
MINOR_ALARMHOST_INFO_UPLOAD_TO_PLATFORM_CFG	/	Platform Information Settings for Uploading
MINOR_ALARMHOST_ONVIF_CONTROL	/	ONVIF Control (Enable or Disable)
MINOR_ALARMHOST_ONVIF_USER_ADD	/	Added ONVIF User
MINOR_ALARMHOST_ONVIF_USER_MOD	/	Edited ONVIF User

Macro Definition	Macro Definition Value	Description
MINOR_ALARMHOST_ONVIF_USER_DELETE	/	Deleted ONVIF User
MINOR_ALARMHOST_TIME_OF_BELLS_CFG	/	Ringing Duration
MINOR_ALARMHOST_CALL_WAITTIME_CFG	/	Calling Waiting Duration
MINOR_ALARMHOST_PROMPT_PARAMETERS_CFG	/	Prompt Sound Settings
MINOR_ALARMHOST_MUTEPLAN_PARAMETERS_CFG	/	Muting Schedule Settings
MINOR_ALARMHOST_SD_PARTITION_CFG	/	SD Card Partition Settings
MINOR_ALARMHOST_AUDIO_PARAMETERS_CFG	/	Audio and Video Settings
MINOR_ALARMHOST_VOICETALK_AUDIO_ENCODING_CFG	/	Two-Way Audio Encoding
MINOR_ALARMHOST_RECORD_PLAN_PARAMETERS_CFG	/	Recording Schedule Settings
MINOR_ALARMHOST_RECORD_ADVANCE_PARAMETERS_CFG	/	Advanced Recording Settings
MINOR_ALARMHOST_PICTURE_PLAN_PARAMETERS_CFG	/	Capture Schedule Settings
MINOR_ALARMHOST_PICTURE_ADVANCE_PARAMETERS_CFG	/	Advanced Capture Settings
MINOR_ALARMHOST_AUDIO_EXCEPTION_PARAMETERS_CFG	/	Audio Exception Settings
MINOR_ALARMHOST_PATROL_CARD_CFG	/	Patrol Card Settings
MINOR_ALARMHOST_VOICE_VOLUME_CFG	/	Sound Settings
MINOR_ALARMHOST_VOICE_MODE_CFG	/	Sound Mode Settings (Input, Output, Broadcast)
MINOR_ALARMHOST_AUDIO_OR_MATERIALS_UPLOAD	/	Uploaded Audio File

Macro Definition	Macro Definition Value	Description
MINOR_ALARMHOST_AUDIO_OR_MATERIALS_DELETE	/	Deleted Audio File
MINOR_ALARMHOST_ALARM_LAMP_FLASH_TIME_CFG	/	Flicking Duration
MINOR_ALARMHOST_ALARM_LAMP_FLASH_PLAN_CFG	/	Scheduled Flicking Settings
MINOR_ALARMHOST_FRONT_END_VIDEO_PARAMETERS_CFG	/	Front-End Camera Video Settings
MINOR_ALARMHOST_WDR_CFG	/	WDR Settings
MINOR_ALARMHOST_BPFRAME_CFG	/	PN Frame Settings
MINOR_ALARMHOST_PASSWORD_RESET_CFG	/	Password Resetting Settings
MINOR_ALARMHOST_ACCOUNT_LOCK	/	Account Locked
MINOR_ALARMHOST_ACCOUNT_UNLOCK	/	Account Unlocked
MINOR_ALARMHOST_START_LIVEVIEW_REMOTELY	/	Start Live View Remotely
MINOR_ALARMHOST_STOP_LIVEVIEW_REMOTELY	/	Stop Live View Remotely
MINOR_ALARMHOST_TELEPHONE_CENTER_SETTINGS	/	Phone Call Center Settings

Table A-6 Minor Types of Events

Macro Definition	Macro Definition Value	Description
MINOR_SCHOOLTIME_IRGI_B	0x01	B Code Synchronization
MINOR_SCHOOLTIME_SDK	0x02	SDK Synchronization
MINOR_SCHOOLTIME_SELFTEST	0x03	Time Synchronization by Schedule
MINOR_SUBSYSTEM_ABNORMALINSERT	0x04	Plugged in Sub-Board
MINOR_SUBSYSTEM_ABNORMALPULLOUT	0x05	Pulled out Sub-Board
MINOR_AUTO_ARM	0x06	Automatic Arming

Macro Definition	Macro Definition Value	Description
MINOR_AUTO_DISARM	0x07	Automatic Disarming
MINOR_TIME_TRIGGER_ON	0x08	Activated Trigger by Schedule
MINOR_TIME_TRIGGER_OFF	0x09	Deactivated Trigger by Schedule
MINOR_AUTO_ARM_FAILED	0x0a	Automatic Arming Failed
MINOR_AUTO_DISARM_FAILED	0x0b	Automatic Disarming Failed
MINOR_TIME_TRIGGER_ON_FAILED	0x0c	Activating Trigger by Schedule Failed
MINOR_TIME_TRIGGER_OFF_FAILED	0x0d	Deactivating Trigger by Schedule Failed
MINOR_MANDATORY_ALARM	0x0e	Forced Arming
MINOR_KEYPAD_LOCKED	0x0f	Keypad Locked
MINOR_USB_INSERT	0x10	Plugged in USB Flash Drive
MINOR_USB_PULLOUT	0x11	Removed USB Flash Drive

A.1.5 NET_DVR_ALARMHOST_MAIN_STATUS_V51

Structure about Security Control Status

Member	Data Type	Description
dwSize	DWORD	Structure size.
bySetupAlarmStatus	Array of BYTE	Zone arming/disarming status, 0xff-invalid, 0-disarmed, 1-armed, 2-being armed. You can get the arming status of up to 512 zones. The maximum size is 512 bytes (value of macro definition "MAX_ALARMHOST_ALARMIN_NUM").
byAlarmInStatus	Array of BYTE	Zone alarm triggering status, 0xff-invalid, 0-no alarm triggered, 1-alarm triggered.

Member	Data Type	Description
		You can get the alarm status of up to 512 zones. The maximum size is 512 bytes (value of macro definition "MAX_ALARMHOST_ALARMIN_NUM").
byAlarmOutStatus	Array of BYTE	Trigger status, 0xff-invalid, 0-not triggered, 1-triggered, 2-no linked trigger, 3-the trigger is offline, 4-heartbeat exception. You can get the trigger status of up to 512 zones. The maximum size is 512 bytes (value of macro definition "MAX_ALARMHOST_ALARMOUT_NUM").
byBypassStatus	Array of BYTE	Zone bypass/bypass recovered status, 0xff-invalid, 0-bypass recovered, 1-bypass. If the array's subscript is 0, the zone No. is 1. The maximum size is 512 bytes (value of macro definition "MAX_ALARMHOST_ALARMIN_NUM").
bySubSystemGuardStatus	Array of BYTE	Partition arming/disarming status, 0xff-invalid, 0-disarmed, 1-armed, 2-being armed. The maximum size is 32 bytes (value of macro definition "MAX_ALARMHOST_SUBSYSTEM").
byAlarmInFaultStatus	Array of BYTE	Zone fault status, 0xff-invalid, 0-normal, 1-fault. The maximum size is 512 bytes (value of macro definition "MAX_ALARMHOST_ALARMIN_NUM").

Member	Data Type	Description
byAlarmInMemoryStatus	Array of BYTE	Zone alarm status, 0xff-invalid, 0-no alarm, 1-in alarm. The maximum size is 512 bytes (value of macro definition "MAX_ALARMHOST_ALARMIN_NUM").
byAlarmInTamperStatus	Array of BYTE	Zone tampering status, 0xff-invalid, 0-no tampering alarm, 1-in tampering alarm. The maximum size is 512 bytes (value of macro definition "MAX_ALARMHOST_ALARMIN_NUM").
byEnableSubSystem	Array of BYTE	Partition enabling/disabling status, 0-invalid, 1-disabled, 2-enabled. The maximum size is 32 bytes (value of macro definition "MAX_ALARMHOST_SUBSYSTEM").
bySubSystemGuardType	Array of BYTE	Partition arming types, 0-invalid, 1-away arming, 2-instant arming, 3-stay arming. The maximum size is 32 bytes (value of macro definition "MAX_ALARMHOST_SUBSYSTEM").
bySubSystemAlarm	Array of BYTE	Partition alarm status, 0-invalid, 1-no alarm, 2-in alarm. The maximum size is 32 bytes (value of macro definition "MAX_ALARMHOST_SUBSYSTEM").
byAlarmOutCharge	Array of BYTE	Trigger battery status, 0-invalid, 1-normal, 2-low battery. You can get the battery status of up to 512 triggers. The maximum size is 512 bytes (value of

Member	Data Type	Description
		macro definition "MAX_ALARMHOST_ALARMOUT_NUM").
byAlarmOutTamperStatus	Array of BYTE	Trigger anti-tampering status, 0-invalid, 1-anti-tampering enabled, 2-anti-tampering disabled. The maximum size is 512 bytes (value of macro definition "MAX_ALARMHOST_ALARMOUT_NUM").
byAlarmInShieldedStatus	Array of BYTE	Zone disabling status, 0-invalid, 1-disabled, 2-not disabled. The maximum size is 512 bytes (value of macro definition "MAX_ALARMHOST_ALARMIN_NUM").
byAlarmOutLinkage	Array of BYTE	Linkage actions of trigger, 0-invalid, 1-trigger alarm, 2-arm, 3-disarm, 4-manual control. You can get the linkage actions of up to 512 triggers. The maximum size is 512 bytes (value of macro definition "MAX_ALARMHOST_ALARMOUT_NUM").
byRes	Array of BYTE	Reserved. The maximum size is 512 bytes.

A.1.6 NET_DVR_ALARMHOST_OTHER_STATUS_V51

Peripherals Status Structure

Member	Data Type	Description
dwSize	DWORD	Structure size.
bySirenStatus	Array of BYTE	Siren status, 0xff-invalid, 0-siren is not triggered, 1-siren triggered, 2-no siren linked, 3-the siren is offline, 4-heartbeat exception. The maximum size is 8 bytes (value of macro definition "ALARMHOST_MAX_SIREN_NUM").
byDetetorPower	Array of BYTE	Detector battery status, 0xff-invalid, the power value is between 0 and 100. By default, if the power value is below 20, the battery status is in low battery, it is valid only when byDetetorPowerType is 0. The maximum size is 256 bytes (value of macro definition "MAX_DETECTOR_NUM_V51").
byDetetorConnection	Array of BYTE	Detector connection status, 0xff-invalid, 0-unregistered, 1-offline, 2-online, 3-heartbeat exception. The maximum size is 256 bytes (value of macro definition "MAX_DETECTOR_NUM_V51").
bySirenPower	Array of BYTE	Siren battery status, 0-invalid, 1-normal, 2-low battery. The maximum size is 8 bytes (value of macro definition "ALARMHOST_MAX_SIREN_NUM").
bySirenTamperStatus	Array of BYTE	Siren anti-tampering status, 0-invalid, 1-anti-tampering is enabled, 2-anti-tampering is disabled. The maximum size is

Member	Data Type	Description
		8 bytes (value of macro definition "ALARMHOST_MAX_SIREN_NUM").
byPowerStausEnabled	Array of BYTE	Detector battery status, 0-invalid, 1-valid. The maximum size is 256 bytes (value of macro definition "MAX_DETECTOR_NUM_V51").
byDetetorPowerStatus	Array of BYTE	Detector battery status, 0-normal, 1-low battery. The maximum size is 256 bytes (value of macro definition "MAX_DETECTOR_NUM_V51").
byDetetorPowerType	BYTE	Detector battery display modes, 0-by power value (byDetetorPower), 1-by battery status (byPowerStausEnabled and byDetetorPowerStatus).
byRes	Array of BYTE	Reserved. The maximum size is 431 bytes.

A.1.7 NET_DVR_ALARMHOST_REPORT_CENTER_CFG_V40

Structure about Data Uploading Configuration

Member	Data Type	Description
dwSize	DWORD	Structure size.
byValid	BYTE	Enable or not: 0-disable, 1-enable.
byDataType	BYTE	Uploaded data type: 1-all alarm data, 2-all non-alarm data, 3-all data, 4-zone alarm report, 5-non-zone alarm report.

Member	Data Type	Description
byRes	Array of BYTE	Reserved. The maximum size is 2 bytes.
byChanAlarmMode	Array of BYTE	Central group alarm channels: 1-T1, 2-T2, 3-N1, 4-N2, 5-G1, 6-G2, 7-N3, 8-N4 (If the device supports 3G, G1 and G2 mean 3G module; If not support, G1 and G2 mean GPRS module. Only one of the 3G module and GPRS module can exist in the device at the same time.). The maximum size is 4 bytes (value of macro definition "MAX_CHAN_NUM").
byDealFailCenter	Array of BYTE	Send failure report to the specified central group, array subscript means the group number, the value: 0-not select, 1-select. byDealFailCenter[0]==1-uploading data to central group 1, byDealFailCenter[1]==1-uploading data to central group 2, and so on. The maximum size is 16 bytes (value of macro definition "MAX_CENTERGROUP_NUM").
byZoneReport	Array of BYTE	Zone alarm report: 0-not upload, 1-upload. byZoneReport[0]==1-uploading the alarm report of zone 1, byZoneReport[1]==1-uploading the alarm report of zone 2, and so on. The maximum size is 512 bytes (value of macro definition "MAX_ALARMHOST_ALARMNUM").
byNonZoneReport	Array of BYTE	Non-zone alarm report, 0-not upload, 1-upload.

Member	Data Type	Description
		<p>Each array means a kind of event type, see below:</p> <p>byNonZoneReport[0]-virtual zone report;</p> <p>byNonZoneReport[1]-system status report;</p> <p>byNonZoneReport[2]-cancel reporting;</p> <p>byNonZoneReport[3]-test report;</p> <p>byNonZoneReport[4]-arming report;</p> <p>byNonZoneReport[5]-disarming report;</p> <p>byNonZoneReport[6]-duress report;</p> <p>byNonZoneReport[7]-alarm recovery report;</p> <p>byNonZoneReport[8]-bypass report;</p> <p>byNonZoneReport[9]-bypass recovery report;</p> <p>byNonZoneReport[10]-detector connection status report (online, offline);</p> <p>byNonZoneReport[11]-detector power status report (power is normal, undervoltage);</p> <p>byNonZoneReport[12]-video alarm report.</p> <p>The maximum size is 32 bytes (value of macro definition "MAX_EVENT_NUM").</p>
byAlarmNetCard	Array of BYTE	Alarm NIC center of central group: 0-"main NIC center 1", 1-"main NIC center 2", 2-"extended NIC center 1",

Member	Data Type	Description
		3-"extended network card center 2" (valid when alarm channel is N1, N2, N3, N4). The maximum size is 4 bytes (value of macro definition "MAX_REPORTCHAN_NUM").
byRes2	Array of BYTE	Reserved, and set to 0. The maximum size is 252 bytes.

A.1.8 NET_DVR_ALARMHOST_SEARCH_LOG_PARAM

Structure about Conditions of Searching for Security Control Panel Logs.

Table A-7 Structure about Conditions of Searching for Security Control Panel Logs

Member	Data Type	Description
wMajorType	WORD	Major type, see details in Table 4-8 .
wMinorType	WORD	Minor type. For minor types of alarms, see details in Table 4-9 ; for minor types of exceptions, see details in Table 4-10 ; for minor types of operations, see details in Table 4-11 ; for minor types of events, see details in Table 4-12 .
struStartTime	<u>NET_DVR_TIME</u>	Start time.
struEndTime	<u>NET_DVR_TIME</u>	End time.
byRes	Array [BYTE]	Reserved. The maximum size is 8 bytes.

Table A-8 Detailed Definitions of Different Major Types

Macro Definition	Macro Definition Value	Description
ALARMHOST_MAJOR_ALARM	1	Alarm
ALARMHOST_MAJOR_EXCEPTION	2	Exception
ALARMHOST_MAJOR_OPERATE	3	Operation
ALARMHOST_MAJOR_EVENT	4	Event

Table A-9 Minor Types of Alarms

Macro Definition	Macro Definition Value	Description
MINOR_SHORT_CIRCUIT	0x01	Short Circuit Alarm
MINOR_BROKEN_CIRCUIT	0x02	Open Circuit Alarm
MINOR_ALARM_RESET	0x03	Alarm Reset
MINOR_ALARM_NORMAL	0x04	Return to Normal
MINOR_PASSWORD_ERROR	0x05	Incorrect Password (Three Failed Attempts)
MINOR_ID_CARD_ILLEGALLY	0x06	Invalid Card ID
MINOR_KEYPAD_REMOVE	0x07	Keypad Tampered
MINOR_KEYPAD_REMOVE_RESTORE	0x08	Keypad Restored
MINOR_DEV_REMOVE	0x09	Device Tampered
MINOR_DEV_REMOVE_RESTORE	0x0a	Device Restored
MINOR_BELOW_ALARM_LIMIT1	0x0b	Sensor Value is Lower than Alarm Limit Value 1
MINOR_BELOW_ALARM_LIMIT2	0x0c	Sensor Value is Lower than Alarm Limit Value 2
MINOR_BELOW_ALARM_LIMIT3	0x0d	Sensor Value is Lower than Alarm Limit Value 3
MINOR_BELOW_ALARM_LIMIT4	0x0e	Sensor Value is Lower than Alarm Limit Value 4
MINOR_ABOVE_ALARM_LIMIT1	0x0f	Sensor Value is Higher than Alarm Limit Value 1

Macro Definition	Macro Definition Value	Description
MINOR_ABOVE_ALARM_LIMIT2	0x10	Sensor Value is Higher than Alarm Limit Value 2
MINOR_ABOVE_ALARM_LIMIT3	0x11	Sensor Value is Higher than Alarm Limit Value 3
MINOR_ABOVE_ALARM_LIMIT4	0x12	Sensor Value is Higher than Alarm Limit Value 4
MINOR_URGENCYBTN_ON	0x13	Panic Button Triggered
MINOR_URGENCYBTN_OFF	0x14	Panic Button Restored
MINOR_VIRTUAL_DEFENCE_BANDIT	0x15	Virtual Zone Burglary Alarm
MINOR_VIRTUAL_DEFENCE_FIRE	0x16	Virtual Zone Fire Alarm
MINOR_VIRTUAL_DEFENCE_URGENT	0x17	Virtual Zone Panic Alarm
MINOR_ALARMHOST_MOTDET_START	0x18	Motion Detection Alarm Started
MINOR_ALARMHOST_MOTDET_STOP	0x19	Motion Detection Alarm Stopped
MINOR_ALARMHOST_HIDE_ALARM_START	0x1a	Device Blocked
MINOR_ALARMHOST_HIDE_ALARM_STOP	0x1b	Device Blocking Alarm Restored
MINOR_ALARMHOST_UPS_ALARM	0x1c	UPS Alarm
MINOR_ALARMHOST_ELECTRICITY_METER_ALARM	0x1d	Coulombmeter Alarm
MINOR_ALARMHOST_SWITCH_POWER_ALARM	0x1e	Switch Power Supply Alarm
MINOR_ALARMHOST_GAS_DETECT_SYS_ALARM	0x1f	Gas Detection Alarm
MINOR_ALARMHOST_TRANSFORMER_TEMPRATURE_ALARM	0x20	Transformer Temperature Alarm
MINOR_ALARMHOST_TEMP_HUMI_ALARM	0x21	Temperature and Humidity Sensor Alarm
MINOR_ALARMHOST_UPS_ALARM_RESTORE	0x22	UPS Alarm Restored

Macro Definition	Macro Definition Value	Description
MINOR_ALARMHOST_ELECTRICITY_METER_ALARM_RESTORE	0x23	Coulombmeter Alarm Restored
MINOR_ALARMHOST_SWITCH_POWER_ALARM_RESTORE	0x24	Switch Power Supply Alarm Restored
MINOR_ALARMHOST_GAS_DETECT_SYS_ALARM_RESTORE	0x25	Gas Detection Alarm Restored
MINOR_ALARMHOST_TRANSFORMER_TEMPRATURE_ALARM_RESTORE	0x26	Transformer Temperature Alarm Restored
MINOR_ALARMHOST_TEMP_HUMI_ALARM_RESTORE	0x27	Temperature-Humidity Sensor Alarm Restored
MINOR_ALARMHOST_WATER_LEVEL_SENSOR_ALARM	0x28	Flood Sensor Alarm
MINOR_ALARMHOST_WATER_LEVEL_SENSOR_ALARM_RESTORE	0x29	Flood Sensor Restored
MINOR_ALARMHOST_DUST_NOISE_ALARM	0x2a	Dust and Noise Sensor Alarm
MINOR_ALARMHOST_DUST_NOISE_ALARM_RESTORE	0x2b	Dust and Noise Sensor Alarm Restored
MINOR_ALARMHOST_ENVIRONMENTAL_LOGGER_ALARM	0x2c	Environmental Data Collector Alarm
MINOR_ALARMHOST_ENVIRONMENTAL_LOGGER_ALARM_RESTORE	0x2d	Environmental Data Collector Restored
MINOR_ALARMHOST_TRIGGER_TAMPER	0x2e	Detector Tampered
MINOR_ALARMHOST_TRIGGER_TAMPER_RESTORE	0x2f	Detector Restored
MINOR_ALARMHOST_EMERGENCY_CALL_HELP_ALARM	0x30	Panic Alarm
MINOR_ALARMHOST_EMERGENCY_CALL_HELP_ALARM_RESTORE	0x31	Panic Alarm Restored
MINOR_ALARMHOST_CONSULTING_ALARM	0x32	Consultation Alarm

Macro Definition	Macro Definition Value	Description
MINOR_ALARMHOST_CONSULTING_ALARM_RESTORE	0x33	Consultation Alarm Restored
MINOR_ZONE_MODULE_REMOVE	0x34	Zone Module Tampered
MINOR_ZONE_MODULE_RESET	0x35	Zone Module Tampering Reset
MINOR_WIRELESS_OUTPUT_MODULE_REMOVE	0x48	Wireless Output Module Tampered
MINOR_WIRELESS_OUTPUT_MODULE_RESET	0x49	Wireless Output Module Tampering Restored
MINOR_WIRELESS_REPEATER_MODULE_REMOVE	0x4a	Wireless Repeater Tampered
MINOR_WIRELESS_REPEATER_MODULE_RESET	0x4b	Wireless Repeater Tampering Restored
MINOR_WIRELESS_SIREN_MODULE_REMOVE	0x4c	Wireless Siren Tampered
MINOR_WIRELESS_SIREN_MODULE_RESET	0x4d	Wireless Siren Tampering Restored

Table A-10 Minor Types of Exceptions

Macro Definition	Macro Definition Value	Description
MINOR_POWER_ON	0x01	Power On
MINOR_POWER_OFF	0x02	Power Off
MINOR_WDT_RESET	0x03	WDT Reset
MINOR_LOW_BATTERY_VOLTAGE	0x04	Low Battery Voltage
MINOR_AC_LOSS	0x05	AC Power Disconnected
MINOR_AC_RESTORE	0x06	AC Power Restored
MINOR_RTC_EXCEPTION	0x07	RTC Real-Time Clock Exception
MINOR_NETWORK_CONNECT_FAILURE	0x08	Network Disconnected
MINOR_NETWORK_CONNECT_RESTORE	0x09	Network Connected
MINOR_TEL_LINE_CONNECT_FAILURE	0x0a	Telephone Line Disconnected

Macro Definition	Macro Definition Value	Description
MINOR_TEL_LINE_CONNECT_RESTORE	0x0b	Telephone Line Connected
MINOR_EXPANDER_BUS_LOSS	0x0c	Bus Expander Disconnected
MINOR_EXPANDER_BUS_RESTORE	0x0d	Bus Expander Connected
MINOR_KEYPAD_BUS_LOSS	0x0e	Keypad Expander Disconnected
MINOR_KEYPAD_BUS_RESTORE	0x0f	Keypad Expander Connected
MINOR_SENSOR_FAILURE	0x10	Analog Sensor Fault
MINOR_SENSOR_RESTORE	0x11	Analog Sensor Restored
MINOR_RS485_CONNECT_FAILURE	0x12	RS-485 Channel Disconnected
MINOR_RS485_CONNECT_RESTORE	0x13	RS-485 Channel Connected
MINOR_BATTERT_VOLTAGE_RESTORE	0x14	Battery Voltage Restored
MINOR_WIRED_NETWORK_ABNORMAL	0x15	Wired Network Exception
MINOR_WIRED_NETWORK_RESTORE	0x16	Wired Network Restored
MINOR_GPRS_ABNORMAL	0x17	GPRS Exception
MINOR_GPRS_RESTORE	0x18	GPRS Restored
MINOR_3G_ABNORMAL	0x19	3G Network Exception
MINOR_3G_RESTORE	0x1a	3G Network Restored
MINOR_SIM_CARD_ABNORMAL	0x1b	SIM Card Exception
MINOR_SIM_CARD_RESTORE	0x1c	SIM Card Restored
MINOR_ALARMHOST_VI_LOST	0x1d	Video Loss
MINOR_ALARMHOST_ILLEGAL_ACCESS	0x1e	Illegal Login
MINOR_ALARMHOST_HD_FULL	0x1f	HDD Full
MINOR_ALARMHOST_HD_ERROR	0x20	HDD Error
MINOR_ALARMHOST_DCD_LOST	0x21	MODEM Disconnected (Reserved)
MINOR_ALARMHOST_IP_CONFLICT	0x22	IP Address Conflicted
MINOR_ALARMHOST_NET_BROKEN	0x23	Network Disconnected
MINOR_ALARMHOST_REC_ERROR	0x24	Recording Error

Macro Definition	Macro Definition Value	Description
MINOR_ALARMHOST_VI_EXCEPTION	0x25	Video Input Exception (Only for Analog Channel)
MINOR_ALARMHOST_FORMAT_HDD_ERROR	0x26	Remote HDD Formatting Failed
MINOR_ALARMHOST_USB_ERROR	0x27	USB Communication Error
MINOR_ALARMHOST_USB_RESTORE	0x28	USB Communication Error Restored
MINOR_ALARMHOST_PRINT_ERROR	0x29	Printer Error
MINOR_ALARMHOST_PRINT_RESTORE	0x2a	Printer Error Restored
MINOR_SUBSYSTEM_COMMUNICATION_ERROR	0x2b	Sub-Board Communication Error
MINOR_ALARMHOST_IPC_NO_LINK	0x2c	Network Camera Disconnected
MINOR_ALARMHOST_IPC_IP_CONFLICT	0x2d	Network Camera IP Address Conflicted
MINOR_ALARMHOST_VI_MISMATCH	0x2e	Video Standard Mismatches
MINOR_ALARMHOST_MCU_RESTART	0x2f	MCU Restarted
MINOR_ALARMHOST_GPRS_MODULE_FAULT	0x30	GPRS Module Fault
MINOR_ALARMHOST_TELEPHONE_MODULE_FAULT	0x31	Telephone Module Fault
MINOR_ALARMHOST_WIFI_ABNORMAL	0x32	Wi-Fi Exception
MINOR_ALARMHOST_WIFI_RESTORE	0x33	Wi-Fi Restored
MINOR_ALARMHOST_RF_ABNORMAL	0x34	RF Exception
MINOR_ALARMHOST_RF_RESTORE	0x35	RF Restored
MINOR_ALARMHOST_DETECTOR_ONLINE	0x36	Detector Connected
MINOR_ALARMHOST_DETECTOR_OFFLINE	0x37	Detector Disconnected
MINOR_ALARMHOST_DETECTOR_BATTERY_NORMAL	0x38	Detector Battery Restored
MINOR_ALARMHOST_DETECTOR_BATTERY_LOW	0x39	Detector Battery Low

Macro Definition	Macro Definition Value	Description
MINOR_ALARMHOST_DATA_TRAFFIC_OVERFLOW	0x3a	Cellular Network Data Exceeded
MINOR_ZONE_MODULE_LOSS	0x3b	Zone Module Disconnected
MINOR_ZONE_MODULE_RESTORE	0x3c	Zone Module Connected
MINOR_ALARMHOST_WIRELESS_OUTPUT_LOSS	0x3d	Wireless Output Module Offline
MINOR_ALARMHOST_WIRELESS_OUTPUT_RESTORE	0x3e	Wireless Output Module Online
MINOR_ALARMHOST_WIRELESS_REPEATER_LOSS	0x3f	Wireless Repeater Offline
MINOR_ALARMHOST_WIRELESS_REPEATER_RESTORE	0x40	Wireless Repeater Online
MINOR_TRIGGER_MODULE_LOSS	0x41	Trigger Module Disconnected
MINOR_TRIGGER_MODULE_RESTORE	0x42	Trigger Module Connected
MINOR_WIRELESS_SIREN_LOSS	0x43	Wireless Siren Offline
MINOR_WIRELESS_SIREN_RESTORE	0x44	Wireless Siren Online

Table A-11 Minor Types of Operations

Macro Definition	Macro Definition Value	Description
MINOR_GUARD	0x01	Normal Arming
MINOR_UNGUARD	0x02	Normal Disarming
MINOR_BYPASS	0x03	Bypass
MINOR_DURESS_ACCESS	0x04	Duress
MINOR_ALARMHOST_LOCAL_REBOOT	0x05	Local Reboot
MINOR_ALARMHOST_REMOTE_REBOOT	0x06	Remote Reboot
MINOR_ALARMHOST_LOCAL_UPGRADE	0x07	Local Upgrade
MINOR_ALARMHOST_REMOTE_UPGRADE	0x08	Remote Upgrade
MINOR_RECOVERY_DEFAULT_PARAM	0x09	Restore Default Settings
MINOR_ALARM_OUTPUT	0x0a	Alarm Output Control

Macro Definition	Macro Definition Value	Description
MINOR_ACCESS_OPEN	0x0b	Access Control: Open
MINOR_ACCESS_CLOSE	0x0c	Access Control: Closed
MINOR_SIREN_OPEN	0x0d	Siren: On
MINOR_SIREN_CLOSE	0x0e	Siren: Off
MINOR_MOD_ZONE_CONFIG	0x0f	Edit Zone Settings
MINOR_MOD_ALARMOUT_CONFIG	0x10	Edit Alarm Output Settings
MINOR_MOD_ANALOG_CONFIG	0x11	Edit Analog Sensor Settings
MINOR_RS485_CONFIG	0x12	Edit RS-485 Channel Settings
MINOR_PHONE_CONFIG	0x13	Edit Dialing Settings
MINOR_ADD_ADMIN	0x14	Added Administrator
MINOR_MOD_ADMIN_PW	0x15	Edited Administrator Password
MINOR_DEL_ADMIN	0x16	Deleted Administrator
MINOR_ADD_NETUSER	0x17	Added DVR/NVR Operator
MINOR_MOD_NETUSER_PW	0x18	Edited DVR/NVR Operator Password
MINOR_DEL_NETUSER	0x19	Deleted DVR/NVR Operator
MINOR_ADD_OPERATORUSER	0x1a	Added Camera Operator
MINOR_MOD_OPERATORUSER_PW	0x1b	Edited Camera Operator Password
MINOR_DEL_OPERATORUSER	0x1c	Deleted Camera Operator
MINOR_ADD_KEYPADUSER	0x1d	Added Keypad/Card Reader User
MINOR_DEL_KEYPADUSER	0x1e	Deleted Keyboard/Card Reader User
MINOR_REMOTEUSER_LOGIN	0x1f	Remote Login
MINOR_REMOTEUSER_LOGOUT	0x20	Remote Logout
MINOR_REMOTE_GUARD	0x21	Remote Arming
MINOR_REMOTE_UNGUARD	0x22	Remote Disarming
MINOR_MOD_HOST_CONFIG	0x23	Edited Control Panel Settings

Macro Definition	Macro Definition Value	Description
MINOR_RESTORE_BYPASS	0x24	Bypass Restored
MINOR_ALARMOUT_OPEN	0x25	Turned on Output
MINOR_ALARMOUT_CLOSE	0x26	Turned off Output
MINOR_MOD_SUBSYSTEM_PARAM	0x27	Edited Partition Parameters
MINOR_GROUP_BYPASS	0x28	Group Bypass
MINOR_RESTORE_GROUP_BYPASS	0x29	Group Bypass Restored
MINOR_MOD_GRPS_PARAM	0x2a	Edited GPRS Parameters
MINOR_MOD_NET_REPORT_PARAM	0x2b	Edited Network Report Settings
MINOR_MOD_REPORT_MOD	0x2c	Edited Uploading Mode
MINOR_MOD_GATEWAY_PARAM	0x2d	Edited Access Control Settings
MINOR_ALARMHOST_REMOTE_START_REC	0x2e	Remote: Started Recording
MINOR_ALARMHOST_REMOTE_STOP_REC	0x2f	Remote: Stopped Recording
MINOR_ALARMHOST_START_TRANS_CHAN	0x30	Transparent Transmission Started
MINOR_ALARMHOST_STOP_TRANS_CHAN	0x31	Transparent Transmission Stopped
MINOR_ALARMHOST_START_VT	0x32	Two-way Audio Started
MINOR_ALARMHOST_STOP_VTM	0x33	Two-way Audio Terminated
MINOR_ALARMHOST_REMOTE_PLAYBYFILE	0x34	Remote: Playback or Downloaded by File
MINOR_ALARMHOST_REMOTE_PLAYBYTIME	0x35	Remote: Playback by Time
MINOR_ALARMHOST_REMOTE_PTZCTRL	0x36	Remote: PTZ Control
MINOR_ALARMHOST_REMOTE_FORMAT_HDD	0x37	Remote: Formatted HDD
MINOR_ALARMHOST_REMOTE_LOCKFILE	0x38	Remote: Locked File
MINOR_ALARMHOST_REMOTE_UNLOCKFILE	0x39	Remote: Unlocked File

Macro Definition	Macro Definition Value	Description
MINOR_ALARMHOST_REMOTE_CFGFILE_OUTPUT	0x3a	Remote: Exported Configuration Files
MINOR_ALARMHOST_REMOTE_CFGFILE_INPUT	0x3b	Remote: Imported Configuration Files
MINOR_ALARMHOST_REMOTE_RECFILE_OUTPUT	0x3c	Remote: Exported Video File
MINOR_ALARMHOST_STAY_ARM	0x3d	Stay Arming
MINOR_ALARMHOST_QUICK_ARM	0x3e	Instant Arming
MINOR_ALARMHOST_AUTOMATIC_ARM	0x3f	Automatic Arming
MINOR_ALARMHOST_AUTOMATIC_DISARM	0x40	Automatic Disarming
MINOR_ALARMHOST_KEYSWITCH_ARM	0x41	Key Zone Arming
MINOR_ALARMHOST_KEYSWITCH_DISARM	0x42	Key Zone Disarming
MINOR_ALARMHOST_CLEAR_ALARM	0x43	Alarm Cleared
MINOR_ALARMHOST_MOD_FAULT_CFG	0x44	Edited System Fault Settings
MINOR_ALARMHOST_MOD_EVENT_TRIGGER_ALARMOUT_CFG	0x45	Edited Event Alarm Output Settings
MINOR_ALARMHOST_SEARCH_EXTERNAL_MODULE	0x46	Searched for External Module
MINOR_ALARMHOST_REGISTER_EXTERNAL_MODULE	0x47	Re-registered External Module
MINOR_ALARMHOST_CLOSE_KEYBOARD_ALARM	0x48	Disabled Keypad Beep
MINOR_ALARMHOST_MOD_3G_PARAM	0x49	Edited 3G Parameters
MINOR_ALARMHOST_MOD_PRINT_PARAM	0x4a	Edited Printer Parameters
MINOR_SD_CARD_FORMAT	0x4b	Formatted SD Card
MINOR_SUBSYSTEM_UPGRADE	0x4c	Upgraded Sub-board
MINOR_ALARMHOST_PLAN_ARM_CFG	0x4d	Arming/Disarming Schedule Configuration

Macro Definition	Macro Definition Value	Description
MINOR_ALARMHOST_PHONE_ARM	0x4e	SMS Arming
MINOR_ALARMHOST_PHONE_STAY_ARM	0x4f	SMS Stay Arming
MINOR_ALARMHOST_PHONE_QUICK_ARM	0x50	SMS Instant Arming
MINOR_ALARMHOST_PHONE_DISARM	0x51	SMS Disarming
MINOR_ALARMHOST_PHONE_CLEAR_ALARM	0x52	SMS Alarm Cleared
MINOR_ALARMHOST_ALLOWLIST_CFG	0x53	Allowlist Settings
MINOR_ALARMHOST_TIME_TRIGGER_CFG	0x54	Enabled/Disabled Trigger Configuration by Schedule
MINOR_ALARMHOST_CAPTRUE_CFG	0x55	Capture Settings
MINOR_ALARMHOST_TAMPER_CFG	0x56	Zone Tamper-Proof Settings
MINOR_ALARMHOST_REMOTE_KEYPAD_UPGRADE	0x57	Remote: Upgraded Keypad
MINOR_ALARMHOST_ONETOUCH_AWAY_ARMING	0x58	One-Touch Away Arming
MINOR_ALARMHOST_ONETOUCH_STAY_ARMING	0x59	One-Touch Stay Arming
MINOR_ALARMHOST_SINGLE_PARTITION_ARMING_OR_DISARMING	0x5a	Single-Zone Arming/Disarming
MINOR_ALARMHOST_CARD_CONFIGURATION	0x5b	Card Settings
MINOR_ALARMHOST_CARD_ARMING_OR_DISARMING	0x5c	Arming/Disarming by Card
MINOR_ALARMHOST_EXPENDING_NETCENTER_CONFIGURATION	0x5d	Extension Network Center Settings
MINOR_ALARMHOST_NETCARD_CONFIGURATION	0x5e	NIC Settings
MINOR_ALARMHOST_DDNS_CONFIGURATION	0x5f	DDNS Settings
MINOR_ALARMHOST_RS485BUS_CONFIGURATION	0x60	RS-485 Bus Settings

Macro Definition	Macro Definition Value	Description
MINOR_ALARMHOST_RS485BUS_RE_REGISTRATION	0x61	RS-485 Bus Re-registration
MINOR_ALARMHOST_REMOTE_OPEN_ELECTRIC_LOCK	0x62	Remote: Unlocked
MINOR_ALARMHOST_REMOTE_CLOSE_ELECTRIC_LOCK	0x63	Remote: Locked
MINOR_ALARMHOST_LOCAL_OPEN_ELECTRIC_LOCK	0x64	Local: Unlocked
MINOR_ALARMHOST_LOCAL_CLOSE_ELECTRIC_LOCK	0x65	Local: Locked
MINOR_ALARMHOST_OPEN_ALARM_LAMP	0x66	Remote: Turned On Alarm Lamp
MINOR_ALARMHOST_CLOSE_ALARM_LAMP	0x67	Remote: Turned Off Strobe
MINOR_ALARMHOST_TEMPORARY_PASSWORD	0x68	Operation Record of Temporary Password
MINOR_ALARMHOST_ONEKEY_AWAY_ARM	0x69	One-Touch Away Arming
MINOR_ALARMHOST_ONEKEY_STAY_ARM	0x6a	One-Touch Stay Arming
MINOR_ALARMHOST_SINGLE_ZONE_ARM	0x6b	Single-Zone Arming
MINOR_ALARMHOST_SINGLE_ZONE_DISARM	0x6c	Single-Zone Disarming
MINOR_ALARMHOST_HIDDNS_CONFIG	0x6d	HiDDNS Settings
MINOR_ALARMHOST_REMOTE_KEYBOARD_UPDATA	0x6e	Remote: Upgraded Keypad
MINOR_ALARMHOST_ZONE_ADD_DETECTOR	0x6f	Added Detector
MINOR_ALARMHOST_ZONE_DELETE_DETECTOR	0x70	Deleted Detector
MINOR_ALARMHOST_QUERY_DETECTOR_SIGNAL	0x71	Checked Detector Signal Strength on Security Control Panel

Macro Definition	Macro Definition Value	Description
MINOR_ALARMHOST_QUERY_DETECTOR_BATTERY	0x72	Checked Detector Remaining Battery on Security Control Panel
MINOR_ALARMHOST_SET_DETECTOR_GUARD	0x73	Detector Arming
MINOR_ALARMHOST_SET_DETECTOR_UNGUARD	0x74	Detector Disarming
MINOR_ALARMHOST_SET_WIFI_PARAMETER	0x75	Wi-Fi Settings
MINOR_ALARMHOST_OPEN_VOICE	0x76	Audio On
MINOR_ALARMHOST_CLOSE_VOICE	0x77	Mute
MINOR_ALARMHOST_ENABLE_FUNCTION_KEY	0x78	Enabled Function Key
MINOR_ALARMHOST_DISABLE_FUNCTION_KEY	0x79	Disabled Panel Function Button
MINOR_ALARMHOST_READ_CARD	0x7a	Swiped Patrol Card
MINOR_ALARMHOST_START_BROADCAST	0x7b	Start Audio Broadcast
MINOR_ALARMHOST_STOP_BROADCAST	0x7c	Stop Audio Broadcast
MINOR_ALARMHOST_REMOTE_ZONE_MODULE_UPGRADE	0x7d	Upgrade Zone Module Remotely
MINOR_ALARMHOST_NETWORK_MODULE_EXTEND	0x7e	Network Module Settings
MINOR_ALARMHOST_ADD_CONTROLLER	0x7f	Added Keyfob User
MINOR_ALARMHOST_DELETE_CONTROLLER	0x80	Deleted Keyfob User
MINOR_ALARMHOST_REMOTE_NETWORKMODULE_UPGRADE	0x81	Upgrade Network Module Remotely
MINOR_ALARMHOST_WIRELESS_OUTPUT_ADD	0x82	Registered Wireless Output Module
MINOR_ALARMHOST_WIRELESS_OUTPUT_DEL	0x83	Deleted Wireless Output Module
MINOR_ALARMHOST_WIRELESS_REPEATER_ADD	0x84	Registered Wireless Repeater

Macro Definition	Macro Definition Value	Description
MINOR_ALARMHOST_WIRELESS_REPEATER_DEL	0x85	Deleted Wireless Repeater
MINOR_ALARMHOST_PHONELIST_CFG	0x86	Phone List Settings
MINOR_ALARMHOST_RF_SIGNAL_CHECK	0x87	RF Signal Search
MINOR_ALARMHOST_USB_UPGRADE	0x88	Upgrade via USB
MINOR_ALARMHOST_DOOR_TIME_REMINDER_CFG	0x89	Scheduled Magnetic Contact Prompt Settings
MINOR_ALARMHOST_WIRELESS_SIREN_ADD	0x8a	Registered Wireless Siren
MINOR_ALARMHOST_WIRELESS_SIREN_DEL	0x8b	Deleted Wireless Siren
MINOR_ALARMHOST_LOCAL_SET_DEVICE_ACTIVE	0xf0	Activated Device Locally
MINOR_ALARMHOST_REMOTE_SET_DEVICE_ACTIVE	0xf1	Activated Device Remotely
MINOR_ALARMHOST_LOCAL_PARA_FACTORY_DEFAULT	0xf2	Restored Factory Settings Locally
MINOR_ALARMHOST_REMOTE_PARA_FACTORY_DEFAULT	0xf3	Restored Factory Settings Remotely

Table A-12 Minor Types of Events

Macro Definition	Macro Definition Value	Description
MINOR_SCHOOLTIME_IRGI_B	0x01	B Code Synchronization
MINOR_SCHOOLTIME_SDK	0x02	SDK Synchronization
MINOR_SCHOOLTIME_SELFTEST	0x03	Time Synchronization by Schedule
MINOR_SUBSYSTEM_ABNORMALINSERT	0x04	Plugged in Sub-Board
MINOR_SUBSYSTEM_ABNORMALPULLOUT	0x05	Pulled out Sub-Board
MINOR_AUTO_ARM	0x06	Automatic Arming
MINOR_AUTO_DISARM	0x07	Automatic Disarming

Macro Definition	Macro Definition Value	Description
MINOR_TIME_TIGGER_ON	0x08	Activated Trigger by Schedule
MINOR_TIME_TIGGER_OFF	0x09	Deactivated Trigger by Schedule
MINOR_AUTO_ARM_FAILED	0x0a	Automatic Arming Failed
MINOR_AUTO_DISARM_FAILED	0x0b	Automatic Disarming Failed
MINOR_TIME_TIGGER_ON_FAILED	0x0c	Activating Trigger by Schedule Failed
MINOR_TIME_TIGGER_OFF_FAILED	0x0d	Deactivating Trigger by Schedule Failed
MINOR_MANDATORY_ALARM	0x0e	Forced Arming
MINOR_KEYPAD_LOCKED	0x0f	Keypad Locked
MINOR_USB_INSERT	0x10	Plugged in USB Flash Drive
MINOR_USB_PULLOUT	0x11	Removed USB Flash Drive

A.1.9 NET_DVR_ALARMHOSTDIALCFG

Dial-up parameter structure.

Structure Definition

```
struct{
    DWORD                dwSize;
    NET_DVR_PHONECENTERDIALCFG  struPhoneCenterParam;
    DWORD                wReportPeriod;
    DWORD                wFirstReportTime;
    BYTE                 byReportValid;
    BYTE                 byRes[19];
}NET_DVR_ALARMHOSTDIALCFG, *LPNET_DVR_ALARMHOSTDIALCFG;
```

Members

dwSize

Structure size.

struPhoneCenterParam

Center parameter, see details in the structure [NET_DVR_PHONECENTERDIALCFG](#) .

wReportPeriod

Period of uploading test reports, which is between 1 and 168 hours (7 days).

wFirstReportTime

Time from the device starting up to uploading the first test report, which is between 1 and 3600, unit: minute.

byReportValid

Whether to enable uploading the test report: 0-disable, 1-enable.

byRes

Reserved, set to 0.

A.1.10 NET_DVR_ALARMIN_PARAM_V50

Zone Parameter Structure

Member	Data Type	Description
dwSize	DWORD	Structure size.
byName	Array [BYTE]	Zone name. The maximum size is 32 bytes (value of macro definition "NAME_LEN").
wDetectorType	WORD	Zone detector type, for details see <i>DETECTOR_TYPE</i> .
byType	BYTE	Alarm type of zone: 0-instant zone, 1-24-hour zone, 2-delay zone, 3-interior zone, 4-key zone, 5-fire zone, 6-perimeter zone, 7-24-hour silent zone, 8-24-hour auxiliary zone, 9-24-hour vibration zone, 10-emergency open door zone, 11-emergency close door zone, 12-over time zone, 13-emergency zone, 14-gas zone, 0xff-none.
byUploadAlarmRecoveryReport	BYTE	Whether to upload zone alarm recovery report: 0-no, 1-yes.
dwParam	DWORD	Zone parameter, the delay time of delay zone. The delay time of power monitoring system

Member	Data Type	Description
		and self-service bank alarm host is set by this parameter.
struAlarmTime	<u>NET DVR SCHEDTIME</u>	Arming time period. It is a two-dimensional array consisting of 7 rows (value of macro definition "MAX_DAYS") and 4 columns (value of macro definition "MAX_TIMESEGMENT").
byAssociateAlarmOut	Array [BYTE]	Linked alarm output. The maximum size is 512 bytes (value of macro definition "MAX_ALARMHOST_ALARMOUT_NUM").
byAssociateSirenOut	Array [BYTE]	Linked siren output, for each array element: 1-output, 0-no output. Array[0] indicates siren No.1, array[1] indicates siren No.2, and so on. The maximum size is 8 bytes.
bySensitivityParam	BYTE	Zone sensitivity: 0-10ms, 1-250ms, 2-500ms, 3-750ms.
byArrayBypass	BYTE	Whether to join the bypass group: 0-no, 1-yes.
byJointSubSystem	BYTE	No. of the partition which the zone belongs to, this parameter can only be obtained.
byModuleStatus	BYTE	External zone status: 1-online, 2-offline, this parameter can only be obtained.
wModuleAddress	WORD	Module address, the extension module is from 1 to 255, 0xFFFF-invalid, and this parameter can only be obtained.

Member	Data Type	Description
byModuleChan	BYTE	Module channel No., starting from 1, and the upper limit is determined by the module type, 0xFF-null, this parameter can only be obtained.
byModuleType	BYTE	Module type: 1-local zone, 2-single zone, 3-dual zone, 4-8-zone, 5-8-ch analog zone, 6-single zone trigger, 7-1-door distributed access controller, 8-2-door distributed access controller, 9-4-door distributed access controller, 10-8-zone wireless, 11-keypad, 12-8-zone wired, 13-extended zone.
wZoneIndex	WORD	Zone No., this parameter can only be obtained.
wInDelay	WORD	Enter delay, which ranges from 0 to 255 seconds.
wOutDelay	WORD	Exit delay, which ranges from 0 to 255 seconds.
byAlarmType	BYTE	Alarm type: 0-invalid, 1-remain open, 2-remain closed.
byZoneResistor	BYTE	Zone resistance, unit: kilo-ohm, value: 0-invalid, 1-2.2, 2-3.3, 3-4.7, 4-5.6, 5-8.2, 0xff-custom.
fZoneResistorManual	float	Zone manual resistance, value range: [1.0, 10.0], it is accurate to one decimal place, unit: kilo-ohm, and it is valid when byZoneResistor is 0xff.
byDetectorSerialNo	Array [BYTE]	Detector serial No (read-only). The maximum size is 16 bytes (value of macro definition "ALARMHOST_DETECTOR_SERIAL_LEN_V50").

Member	Data Type	Description
byZoneSignalType	BYTE	Zone signal transmission type (read-only), 0-wired, 1-wireless.
byDisableDetectorTypeCfg	BYTE	Whether detector type can be configured: 0-yes, 1-no.
wTimeOut	WORD	Timeout, unit: second.
byAssociateLampOut	Array [BYTE]	Siren output, array[0]-siren 1, and so forth. For each array element: 1-output, 0-no output. The maximum size is 8 bytes.
byVoiceFileName	Array [BYTE]	Voice file name. The maximum size is 32 bytes.
byTimeOutRange	BYTE	Timeout range, 0-from 1 to 599s, 1-from 1 to 65535s.
byDetectorSignalIntensity	BYTE	Detector signal strength (read-only), which ranges from 0 to 100.
byTimeOutMethod	BYTE	Timing method of over time zone: 0-timing when the alarm is triggered, 1-timing when the alarm is recovered.
byAssociateFlashLamp	BYTE	Flashing light output: 0-invalid, 1-not output, 2-output.
byStayAwayEnabled	BYTE	Whether to enable stay arming bypass for the instant zone: 0-invalid, 1-disable, 2-enable.
bySilentModeEnabled	BYTE	Whether to enable muting mode: 0-invalid, 1-disable, 2-enable.
byRelativeChannel	Array [BYTE]	Linked channel No. Each array element refers to a channel No. For each element, 0 means invalid. The maximum size is 2 bytes (value of macro

Member	Data Type	Description
		definition "RELATIVE_CHANNEL_LEN").
byDetectorVersion	Array [BYTE]	Detector version (read-only). The maximum size is 32 bytes (value of macro definition "VERSION_INFO_LEN").
byDetectorMAC	Array [BYTE]	Detector's MAC address (read-only). The maximum size is 6 bytes (value of macro definition "MACADDR_LEN").
byLinkageAlarmType	BYTE	Linked alarm type: 1-fire alarm, 2-supervision, 3-linkage, 4-shielding, 5-fault.
byRes3	Array [BYTE]	Reserved. The maximum size is 465 bytes.

Remarks

- For local zones, **byJointSubSystem**, **byModuleStatus**, **wModuleAddress**, **byModuleChan**, **byModuleType**, and **wZoneIndex** cannot be edited.
- Zone parameters cannot be edited when the zone is armed, the partition which the zone belongs to is armed, or the security control panel is in programming mode or pacing mode.
- For 24-hour zones, **byArrayBypass** cannot be set to 1, which means group bypass is not supported.

A.1.11 NET_DVR_ALARM_CAPTRUE_CFG

Structure about parameters for capturing alarm pictures.

Structure Definition

```
struct{
    DWORD    dwSize;
    BYTE     byBeforeAlarmPic;
    BYTE     byAfterAlarmPic;
    WORD     wInterval;
    BYTE     byResolution;
    BYTE     byRes[63];
}NET_DVR_ALARM_CAPTRUE_CFG, *LPNET_DVR_ALARM_CAPTRUE_CFG;
```


Members

dwSize

Structure size.

byBeforeAlarmPic

Maximum number of pictures of the corresponding resolution that can be captured before the alarm: 4CIF-10, 2CIF-20, CIF-40, QCIF-80, WD1-10, VGA-10, XVGA-10, 720P-10, 1080P-10.

byAfterAlarmPic

Maximum number of pictures of the corresponding resolution that can be captured after the alarm: 4CIF-10, 2CIF-20, CIF-40, QCIF-80, WD1-10, VGA-10, XVGA-10, 720P-10, 1080P-10.

wInterval

Time interval, unit: second.

byResolution

Picture resolution: 1-CIF, 2-QCIF, 3-4CIF, 4-2CIF, 5-WD1, 6-VGA, 7-XVGA, 8-720P, 9-1080P.

byRes

Reserved, set to 0.

A.1.12 NET_DVR_ALARM_DEVICE_USER

Structure about User Configuration of Security Control Devices

Member	Data Type	Description
dwSize	DWORD	Structure size.
sUserName	BYTE[]	User name. The array length is 32 bytes (macro definition value of "NAME_LEN").
sPassword	BYTE[]	Password. The array length is 16 bytes (macro definition value of "PASSWD_LEN").
struUserIP	<u>NET_DVR_IPADDR</u>	IP address of the user. If it is 0, it indicates that all addresses are allowed.
byMACAddr	BYTE[]	MAC address. The array length is 6 bytes (macro definition value of "MACADDR_LEN").

Member	Data Type	Description
byUserType	BYTE	User type: 0-normal user, 1-admin user.
byAlarmOnRight	BYTE	Whether the person has arming permission: 0-no, 1-yes.
byAlarmOffRight	BYTE	Whether the person has disarming permission: 0-no, 1-yes.
byBypassRight	BYTE	Whether the person has bypass permission: 0-no, 1-yes.
byOtherRight	BYTE[]	<p>Whether the person has other permission: 0-no, 1-yes. The array length is 32 bytes (macro definition value of "MAX_RIGHT").</p> <ul style="list-style-type: none"> • byOtherRight[0]: log permission. • byOtherRight[1]: restart and shut down. • byOtherRight[2]: permission of setting parameters. • byOtherRight[3]: permission of getting parameters. • byOtherRight[4]: permission of restoring to default settings. • byOtherRight[5]: siren output permission. • byOtherRight[6]: PTZ control permission. For video security control panel, this parameter is invalid, and the PTZ control permission is configured by byNetPTZRight. • byOtherRight[7]: permission of upgrading remotely. • byOtherRight[8]: alarm output control permission.

Member	Data Type	Description
		<ul style="list-style-type: none"> • byOtherRight[9]: serial port control permission. • byOtherRight[10]: access control permission. • byOtherRight[11]: two-way audio permission. • byOtherRight[12]: permission of controlling local output remotely. • byOtherRight[13]: permission of configuring HDD. • byOtherRight[14]: permission of formatting HDD. • byOtherRight[15]: permission of controlling analog sensor.
byNetPreviewRight	BYTE[]	Channels that support remote live view. The array length is 8 bytes (macro definition value of "MAX_ALARMHOST_VIDEO_CHAN" divided by 8). Each element in the array indicates the corresponding channel. For each element, 1 means having permission, and 0 means having no permission.
byNetRecordRight	BYTE[]	Channels that support remote recording. The array length is 8 bytes (macro definition value of "MAX_ALARMHOST_VIDEO_CHAN" divided by 8). Each element in the array indicates the corresponding channel. For each element, 1 means having permission, and 0 means having no permission.

Member	Data Type	Description
byNetPlaybackRight	BYTE[]	Channels that support remote playback. The array length is 8 bytes (macro definition value of "MAX_ALARMHOST_VIDEO_CHAN" divided by 8). Each element in the array indicates the corresponding channel. For each element, 1 means having permission, and 0 means having no permission.
byNetPTZRight	BYTE[]	Channels that support remote PTZ control. The array length is 8 bytes (macro definition value of "MAX_ALARMHOST_VIDEO_CHAN" divided by 8). Each element in the array indicates the corresponding channel. For each element, 1 means having permission, and 0 means having no permission.
sOriginalPassword	BYTE[]	Original password. The array length is 16 bytes (macro definition value of "PASSWD_LEN").
byRes2	BYTE[]	Reserved. The array length is 152 bytes.

A.1.13 NET_DVR_ALARM_ISAPI_INFO

Structure about Alarm Information Transmitted Based on Text Protocol

Member	Data Type	Description
pAlarmData	char*	Alarm information based on text protocol (XML or JSON message without binary data).
dwAlarmDataLen	DWORD	Alarm data length.

Member	Data Type	Description
byDataType	BYTE	Alarm data type: 0-invalid, 1-XML, 2-JSON.
byPicturesNumber	BYTE	The number of pictures (number of pPicPackData returned). When this member is 1, only one structure of <u>NET_DVR_ALARM_ISAPI_PICDATA</u> will be returned by pPicPackData . When this member is larger than 1, multiple structures of <u>NET_DVR_ALARM_ISAPI_PICDATA</u> will be returned by pPicPackData .
byRes	Array of BYTE	Reserved, set to 0. The maximum size is 2 bytes.
pPicPackData	void*	Alarm picture structure, see <u>NET_DVR_ALARM_ISAPI_PICDATA</u> for details.
byRes	Array of BYTE	Reserved. The maximum size is 32 bytes.

Remarks

When enabling the listening mode, you should call the network configuration API based on text protocol to set the IP address for the listening service.

A.1.14 NET_DVR_ALARM_ISAPI_PICDATA

Structure about Alarm Picture Data Transmitted Based on Text Protocol

Member	Data Type	Description
dwPicLen	DWORD	Alarm picture data length.
byRes	Array of BYTE	Reserved, set to 0. The maximum size is 4 bytes.

Member	Data Type	Description
szFilename	Array of char	Picture file saving path, including file name. The maximum size is 256 bytes.
pPicData	BYTE*	Pointer that pointing to the uploaded image data.

A.1.15 NET_DVR_ALARM_LAMP_CFG

Alarm lamp parameter structure.

Structure Definition

```
struct{
    DWORD    dwSize;
    BYTE     byEnable;
    BYTE     byRes1;
    WORD     wFlashDuration;
    WORD     wFlashIntervalTime;
    BYTE     byRes[510];
}NET_DVR_ALARM_LAMP_CFG, *LPNET_DVR_ALARM_LAMP_CFG;
```

Members

dwSize

Structure size.

byEnable

Whether to enable scheduled alarm lamp flickering: 0-no, 1-yes.

byRes1

Reserved, set to 0.

wFlashDuration

Scheduled flickering duration of the alarm lamp, unit: second.

wFlashIntervalTime

Alarm lamp flickering interval, unit: second.

byRes

Reserved, set to 0.

A.1.16 NET_DVR_ALARMIN_SETUP

Zone Parameter Structure

Member	Data Type	Description
byAssociateAlarmIn	Array of BYTE	No. of zone to be armed, if the array subscript is 0, the zone No. is 1. E.g., byAssociateAlarmIn[i]==1, the zone No.(i+1) will be armed. The maximum size is 512 bytes (value of macro definition "MAX_ALARMHOST_ALARMIN_NUM").
byRes	Array of BYTE	Reserved. The maximum size is 100 bytes.

A.1.17 NET_DVR_ALARMOUT_PARAM

Trigger (relay) parameter structure.

Structure Definition

```
struct{
    DWORD          dwSize;
    BYTE           byName [NAME_LEN/*32*/];
    WORD           wDelay;
    WORD           wTriggerIndex;
    BYTE           byAssociateAlarmIn [MAX_ALARMHOST_ALARMIN_NUM/*512*/];
    BYTE           byModuleType;
    BYTE           byModuleStatus;
    WORD           wModuleAddress;
    BYTE           byModuleChan;
    BYTE           byWorkMode;
    BYTE           byAlarmOutMode;
    BYTE           byTimeOn;
    BYTE           byTimeOff;
    BYTE           byRes2 [51];
}NET_DVR_ALARMOUT_PARAM, *LPNET_DVR_ALARMOUT_PARAM;
```

Members

dwSize

Structure size.

byName

Trigger (relay) name.

wDelay

Output delay, unit: second, value range:

Network alarm host, trunk network alarm host, video alarm host: from 0 to 5999, 0 means output when alarm is triggered and no output when there is no alarm triggered. Power supply monitoring alarm host V2.0: 0 to 65535, 0 means continuous output. Power supply monitoring alarm host V1.0: 0 to 65535, 0 means continuous output. Self-service bank alarm host: 0 to 5999, 0 means continuous output when alarm is triggered.

wTriggerIndex

Trigger No. which can only be obtained.

byAssociateAlarmIn

Alarm input channel of the siren (multiple alarm inputs trigger the same siren to output simultaneously). The **byAssociateAlarmIn[0]** refers to alarm input 1, and so forth. For each array element, 0 means not triggering siren, 1 means triggering siren.

byModuleType

External trigger type: 1-local trigger, 2-4-channel trigger, 3-8-channel trigger, 4-single-zone trigger, 5-32-channel trigger, 6-1-door distributed access controller, 7-2-door distributed access controller, 8-4-door distributed access controller, 9-2-channel trigger.

byModuleStatus

External trigger status: 1-online, 2-offline.

wModuleAddress

External trigger address. For the extension module, it ranges from 1 to 253, 0xFFFF means invalid.

byModuleChan

External trigger channel No., it starts from 1, and the maximum value depends on the module type, 0xFF means invalid.

byWorkMode

Working mode: 1-link, 2-dynamic.

byAlarmOutMode

Output mode: 1-non-pulse mode, 2-pulse mode.

byTimeOn

On time, value range: [1, 60], unit:second.

byTimeOff

Off duration, value range: [1, 60], unit:second.

byRes2

Reserved.

Remarks

- can only be obtained but cannot be modified.
The **wTriggerIndex**, **byModuleType**, **byModuleStatus**, **wModuleAddress**, and **byModuleChan** of the local trigger cannot be edited.
- The trigger parameters cannot be edited when the security control panel is in programming mode or pacing mode.
- For the single-zone trigger, the zone channel No. is 1 and the trigger channel No. is 2.

A.1.18 NET_DVR_CETTIFICATE_INFO

Certificate information structure

Structure Definition

```
struct{
    DWORD          dwSize;
    char           szIssuer[MAX_CERTIFICATE_ISSUER_LEN/*64*/];
    char           szSubject[MAX_CERTIFICATE_SUBJECT_LEN/*64*/];
    NET_DVR_TIME   struStartTime;
    NET_DVR_TIME   struEndTime;
    BYTE           byRes1[1024];
}NET_DVR_CETTIFICATE_INFO, *LPNET_DVR_CETTIFICATE_INFO;
```

Members

dwSize

Structure size.

szIssuer

Certificate issuer.

szSubject

Certificate holder.

struStartTime

Start time of expiry date, refer to the structure **NET_DVR_TIME** for details.

struEndTime

End time of expiry date, refer to the structure **NET_DVR_TIME** for details.

byRes1

Reserved.

A.1.19 NET_DVR_CID_ALARM

Table A-13 Structure about CID Alarm Information

Member	Data Type	Description
dwSize	DWORD	Structure size.
sCIDCode	Array [BYTE]	CID event No., refer to <u>NET_DVR_ALARMHOST_CID_ALL_MINOR_TYPE</u> for details. The maximum size is 4 bytes (the value of the macro definition "CID_CODE_LEN").
sCIDDescribe	Array [BYTE]	CID event name. The maximum size is 32 bytes (the value of the macro definition "NAME_LEN").
struTriggerTime	<u>NET_DVR_TIME_EX</u>	Alarm triggering time.
struUploadTime	<u>NET_DVR_TIME_EX</u>	Alarm uploading time.
sCenterAccount	Array [BYTE]	Center account which is valid when byCenterType is 0 or 1. The maximum size is 6 bytes (the value of the macro definition "ACCOUNTNUM_LEN").
byReportType	BYTE	Report type: <pre>enum _NET_DVR_ALARMHOST_REPORT_TYPE_ { NET_DVR_DEFENCE_ALARM = 1, //Zone alarm NET_DVR_VIDEO_ALARM, //Video alarm NET_DVR_VIRTUAL_DEFENCE_ALARM, //Virtual zone alarm NET_DVR_HOSTAGE_ALARM, //Duress alarm NET_DVR_KNOCK_DOWN_ALARM, //Tampering alarm NET_DVR_OPERATE_</pre>

Member	Data Type	Description
		ALARM, // Operation report NET_DVR_OHTER_ABNORMAL_ ALARM // Exception report }NET_DVR_ALARMHOST_ REPORT_TYPE
byUserType	BYTE	User Type: 0-network user, 1-keypad user, 2-mobile phone user, 3-system user.
sUserName	Array [BYTE]	Network user name. The maximum size is 32 bytes (the value of the macro definition "NAME_LEN").
wKeyUserNo	WORD	Keypad user No., 0xFFFF-invalid.
byKeypadNo	BYTE	Keypad No., 0xFF-invalid.
bySubSysNo	BYTE	Partition No., 0xFF-invalid.
wDefenceNo	WORD	Zone No., 0xFFFF-invalid.
byVideoChanNo	BYTE	Video channel No., 0xFF-invalid.
byDiskNo	BYTE	HDD No., 0xFF-invalid.
wModuleAddr	WORD	Module address, 0xFFFF-invalid.
byCenterType	BYTE	Center account type: 0-invalid, 1-center account (the length is 6), 2-extended center account (the length is 32).
byRelativeChannel	BYTE	External video channel No., 0-invalid.
sCenterAccountV40	Array [BYTE]	Extended center account which is valid when byCenterType is 2. When this member is configured, the member sCenterAccount is invalid. The maximum size is 32 bytes (the

Member	Data Type	Description
		value of the macro definition "ACCOUNTNUM_LEN_32").
byDevSerialNo	BYTE	Product serial No. The maximum size is 9 bytes (the value of the macro definition "DEV_SERIAL_LEN").
byRes3	BYTE	Reserved. The maximum size is 3 bytes.
dwIOTChannelNo	DWORD	IOT channel No.
standardCIDcode	BYTE	Standard CID code.
byRes2	Array [BYTE]	Reserved. The maximum size is 11 bytes.

A.1.20 NET_DVR_CONTROL_PARAM

Control Parameter Structure

Member	Data Type	Description
dwSize	DWORD	Structure size.
sDeviceID	Array of BYTE	ID of device to be controlled, you can set it to "NULL". The maximum size is 32 bytes (value of macro definition "NAME_LEN").
wChan	WORD	Channel to be controlled, it is invalid, set to 0.
byIndex	BYTE	No. of different control objects, it varies with different control commands (dwCommand) in the API <i>NET_DVR_RemoteControl</i> , see details in <i>Table 4-14</i> .
byRes1	BYTE	Reserved, set to 0.

Member	Data Type	Description
dwControlParam	DWORD	Control parameters, which are different according to different control commands, see details in Table 4-15 .
byMandatoryAlarm	BYTE	Whether to force arming: 0-no, 1-yes.
byRes2	BYTE	Reserved, set to 0.
wZoneIndex	WORD	Zone No.
byOperatorCode	Array of BYTE	Control No. The maximum size is 16 bytes.
dwPlanNo	DWORD	4-byte plan No.
byRes3	Array of BYTE	Reserved, set to 0. The maximum size is 8 bytes.

Table A-14 Control Object No. Corresponding to Different Control Commands

dwCommand	byIndex	Command Description
NET_DVR_OPEN_PLAN	Plan No.	Control plan.
NET_DVR_TURNON_LED	Invalid	Enable LED.
NET_DVR_TURNOFF_LED	Invalid	Disable LED.
NET_DVR_SET_LED_BRIGHTNESS	Serial No., which is ranging from 0 to n (determined by device capability).	Manually set the brightness of LED.
NET_DVR_CLOSE_SUBSYSTEM_FAULT_ALARM	Partition No., which starts from 1, 0xffffffff-all partitions.	Turn off the prompt sound of partition fault.
NET_DVR_SET_SUBSYSTEM_BYPASS	Partition No., which starts from 1, 0xffffffff-all partitions.	Perform bypass on partition.
NET_DVR_CANCEL_SUBSYSTEM_BYPASS	Partition No., which starts from 1, 0xffffffff-all partitions.	Perform bypass recovered on partition.
NET_DVR_ARM_ALARMHOST_SUBSYSTEM	Partition No., which starts from 1, 0xffffffff-all partitions.	Arm the partition by different arming types.

Table A-15 Control Parameters Corresponding to Different Control Commands

dwCommand	dwControlParam
NET_DVR_OPEN_PLAN	1-plan started, 2-plan ended
NET_DVR_TURNON_LED	Invalid, set to 0.
NET_DVR_TURNOFF_LED	Invalid, set to 0.
NET_DVR_SET_LED_BRIGHTNESS	Brightness value, which is ranging from 0 to 15.
NET_DVR_CLOSE_SUBSYSTEM_FAULT_ALARM	Invalid, set to 0.
NET_DVR_SET_SUBSYSTEM_BYPASS	Invalid, set to 0.
NET_DVR_CANCEL_SUBSYSTEM_BYPASS	Invalid, set to 0.
NET_DVR_ARM_ALARMHOST_SUBSYSTEM	Arming types: 1-away arming, 2-instant arming, 3-stay arming.

A.1.21 NET_DVR_DEVICEINFO_V40

Device Parameter Structure (V40)

Member	Data Type	Description
struDeviceV30	<u>NET_DVR_DEVICEINFO_V30</u>	Device parameters
bySupportLock	BYTE	Whether supports locking function: 1-support.
byRetryLoginTime	BYTE	Remaining login attempts, it is valid when the user name or password is incorrect and the bySupportLock is 1.
byPasswordLevel	BYTE	Password strength: 0-invalid, 1-default password, 2-valid password, 3-risky password. For default password or risky password, the users are reminded to change password.
byProxyType	BYTE	Proxy type: 0-no proxy, 1-standard proxy, 2-EHome proxy.
dwSurplusLockTime	DWORD	Remaining locking time, unit: second. It is valid only when bySupportLock is 1. During the

Member	Data Type	Description
		locking time, if the user try to log in to again, the remaining locking time will resume to 30 minutes.
byCharEncodeType	BYTE	Character encodings. 0-no decoding information, 1-GB2312 (Simplified Chinese), 2-GBK, 3-BIG5 (Traditional Chinese), 4-Shift_JIS (Japanese), 5-EUC-KR (Korean), 6-UTF-8, 7-ISO8859-1, 8-ISO8859-2, 9-ISO8859-3, ..., 21-ISO8859-15 (Western European)
bySupportDev5	BYTE	Whether to support getting the parameters of devices that support HCNetsdk version 5.0 or above, the size of device name and type name are extended to 64 bytes.
bySupport	BYTE	Whether it supports uploading changes, it depends on the result of bitwise AND (&) operation: 0-not support, 1-support. The result of bySupport&0x1 indicates that this member is reserved; the result of bySupport&0x2 indicates that whether it supports uploading changes: 0-not support, 1-support. This member is the capability set extension.
byLoginMode	BYTE	Login mode: 0-login via private protocol, 1-login via text protocol. For private protocol, the default login port number is 8000, and for text protocol, the default login port number is 80 or 443.
dwOEMCode	DWORD	OEM code.
iResidualValidity	int	Remaining valid days of the user's password, unit: day. If the negative number is returned, it indicates that the password being used has expired. For example, if -3 is returned, it indicates that the password being used has expired for three days.
byResidualValidity	BYTE	Whether the member iResidualValidity is valid: 0-invalid, 1-valid.

Member	Data Type	Description
bySingleStartDTalkChan	BYTE	Start channel No. for connecting independent audio tracks to the device. The value 0 is reserved and invalid. The channel No. of audio tracks cannot start from 0.
bySingleDTalkChanNums	BYTE	Total number of channels of the device connected with independent tracks, 0-not support.
byPassWordResetLevel	BYTE	Whether to prompt the non-admin user to change the password: 0 (invalid), 1 (If the administrator creates a non-admin user account with an initial password, the non-admin user will be prompted "Please change the initial password" each time he/she logs in to the device until he/she changes the initial password), 2(If the non-admin user's password has been changed by the administrator, the non-admin user will be prompted "Please set a new password" each time he/she logs in to the device until he/she changes the password).
bySupportStreamEncrypt	BYTE	Whether it supports stream encryption, it depends on the result of bitwise AND (&) operation: 0-no, 1-yes. The result of bySupportStreamEncrypt&0x1 indicates whether to support RTP/TLS streaming, the result of bySupportStreamEncrypt&0x2 indicates whether to support SRTP/UDP streaming, and the result of bySupportStreamEncrypt&0x4 indicates whether to support SRTP/MULTICAST streaming.
byRes2	Array of BYTE	Reserved, set to 0.

Remarks

- Four character types are allowed in the password, including digits, lowercase letters, uppercase letters and symbols. The maximum password length is 16 bits, and there are four password strength levels, see details below:

- Level 0 (Risky Password): The password length is less than 8 bits, or only contains one kind of the character types. Or the password is the same with the user name, or is the mirror writing of the user name.
- Level 1 (Weak Password): The password length is more than or equal to 8 bits, and contains two kinds of the character types. Meanwhile, the combination should be (digits + lowercase letters) or (digits + uppercase letters).
- Level 2 (Medium Password): The password length is more than or equal to 8 bits, and contains two kinds of the character types. Meanwhile, the combination cannot be (digits + lowercase letters) and (digits + uppercase letters).
- Level 3 (Strong Password): The password length is more than or equal to 8 bits, and at least contains three kinds of the character types.
- For login via text protocol, the following parameters are not supported: **bySupportLock**, **byRetryLoginTime**, **byPasswordLevel**, **byProxyType**, **dwSurplusLockTime**, **byCharEncodeType**, and **bySupportDev5**.

A.1.22 NET_DVR_DEVICEINFO_V30

Device parameter structure (V30).

Device Parameter Structure (V30)

Member	Data Type	Description
sSerialNumber	BYTE	Device serial No.
byAlarmInPortNum	BYTE	Number of analog alarm inputs
byAlarmOutPortNum	BYTE	Number of analog alarm outputs
byDiskNum	BYTE	Number of HDDs
byDVRType	BYTE	Device type
byChanNum	BYTE	Number of analog channels
byStartChan	BYTE	Start No. of analog channel, which starts from 1.
byAudioChanNum	BYTE	Number of two-way audio channels
byIPChanNum	BYTE	Number of digital channels, low 8-bit.
byZeroChanNum	BYTE	Number of channel-zero
byMainProto	BYTE	Transmission protocol type of main stream: 0-private protocol (default), 1-RTSP, 2-private protocol+RTSP

Member	Data Type	Description
bySubProto	BYTE	Transmission protocol type of sub-stream: 0-private protocol (default), 1-RTSP, 2-private protocol+RTSP
bySupport	BYTE	<p>Capabilities, if the result of bitwise operation is 0, it refers that the capability is not supported, if the result is 1, it indicates that the capability is supported.</p> <ul style="list-style-type: none"> • bySupport&0x1: whether supports VCA search. • bySupport&0x2: whether supports backup. • bySupport&0x4: whether supports getting encoding parameters. • bySupport&0x8: whether supports dual-NIC. • bySupport&0x10: whether supports remote SADP. • bySupport&0x20: whether supports RAID card. • bySupport&0x40: whether supports searching in IPSAN directory. • bySupport&0x80: whether supports RTP over RTSP.
bySupport1	BYTE	<p>Extended capabilities, if the result of bitwise operation is 0, it refers that the capability is not supported, if the result is 1, it indicates that the capability is supported.</p> <ul style="list-style-type: none"> • bySupport1&0x1: whether supports SNMP with version 30. • bySupport1&0x2: whether supports playback and downloading video files. • bySupport1&0x4: whether supports setting the arming priority. • bySupport1&0x8: whether supports extending the arming time period. • bySupport1&0x10: whether supports multiple HDDs (more than 33). • bySupport1&0x20: whether supports RTP over RTSP. • bySupport1&0x80: whether supports license plate recognition alarm.

Member	Data Type	Description
bySupport2	BYTE	<p>Extended capabilities, if the result of bitwise operation is 0, it refers that the capability is not supported, if the result is 1, it indicates that the capability is supported.</p> <ul style="list-style-type: none"> bySupport2&0x1: whether supports getting stream via URL. bySupport2&0x2: whether supports FTP with version 40. bySupport2&0x4: whether supports ANR. bySupport2&0x20: whether supports getting device status. bySupport2&0x40: whether supports encrypting stream.
wDevType	WORD	Device model
bySupport3	BYTE	<p>Extended capabilities, if the result of bitwise operation is 0, it refers that the capability is not supported, while, if the result is 1, it indicates that the capability is supported.</p> <ul style="list-style-type: none"> bySupport3&0x1: whether supports multi-stream. bySupport3&0x4: whether supports configuring by group (e.g., image, alarm input, alarm output, user, device status, JPEG picture capture, continuous and scheduled capture, .HDD group management, and so on). bySupport3&0x20: whether supports getting stream via DDNS.
byMultiStreamProto	BYTE	<p>Whether supports multi-stream, if the result of bitwise operation is 0, it refers to not support, if the result is 1, it refers to support.</p> <ul style="list-style-type: none"> byMultiStreamProto&0x1: whether supports third-stream. byMultiStreamProto&0x2: whether supports fourth-stream.

Member	Data Type	Description
		<ul style="list-style-type: none"> byMultiStreamProto&0x40: whether supports main stream. byMultiStreamProto&0x80: whether supports sub-stream.
byStartDChan	BYTE	Start No. of digital channel, 0-no digital channel (e.g., DVR, network camera).
byStartDTalkChan	BYTE	Start No. of two-way audio channel, 0-no two-way audio channel.
byHighDChanNum	BYTE	Number of digital channels, high 8-bit.
bySupport4	BYTE	<p>Extended capabilities, if the result of bitwise operation is 0, it refers that the capability is not supported, if the result is 1, it indicates that the capability is supported.</p> <ul style="list-style-type: none"> bySupport4&0x01: whether all stream types support RTSP and private protocol. bySupport4&0x02: whether the device supports transmitting form format data via API (NET_DVR_STDXMLConfig). bySupport4&0x10: whether supports loading network disk by domain name.
byLanguageType	BYTE	<p>Supported language types, if the result of bitwise operation is 0, it refers to not support, if the result is 1, it refers to support.</p> <ul style="list-style-type: none"> byLanguageType ==0: this field is not supported by device. byLanguageType&0x1: whether supports Chinese. byLanguageType&0x2: whether supports English.
byVoiceInChanNum	BYTE	Number of audio input channels
byStartVoiceInChanNo	BYTE	Start No. of audio input channel, 0-invalid.
byRes3	Array of BYTE	Reserved, set to 0.
byMirrorChanNum	BYTE	Number of mirror channels
wStartMirrorChanNo	WORD	Start No. of mirror channel
byRes2	Array of BYTE	Reserved, set to 0.

Remarks

- The maximum number of digital channels equal to `byIPChanNum+byHighDChanNum*256`.
- For login via text protocol, the following parameters are not supported: **byMainProto**, **bySubProto**, **bySupport**, **bySupport1**, **bySupport2**, **bySupport3**, **bySupport4**, **bySupport5**, **bySupport6**, **bySupport7**, **byMultiStreamProto**, **byStartDTalkChan**, **byVoiceInChanNum**, **byStartVoiceInChanNo**, **byMirrorChanNum**, and **wStartMirrorChanNo**.

See Also

NET_DVR_DEVICEINFO_V40

A.1.23 NET_DVR_INIT_CFG_ABILITY**Initialization Capability Structure**

Member	Data Type	Description
enumMaxLoginUsersNum	INIT_CFG_MAX_NUM	Maximum number of users can log in, see details below: <pre>enum _INIT_CFG_MAX_NUM_{ INIT_CFG_NUM_2048 = 2048, INIT_CFG_NUM_5120 = 5120, INIT_CFG_NUM_10240 = 10240, INIT_CFG_NUM_15360 = 15360, INIT_CFG_NUM_20480 = 20480 }_INIT_CFG_MAX_NUM</pre>
enumMaxAlarmNum	INIT_CFG_MAX_NUM	Maximum number of alarm channels, see details below: <pre>enum _INIT_CFG_MAX_NUM_{ INIT_CFG_NUM_2048 = 2048, INIT_CFG_NUM_5120 = 5120, INIT_CFG_NUM_10240 = 10240, INIT_CFG_NUM_15360 = 15360, INIT_CFG_NUM_20480 = 20480 }_INIT_CFG_MAX_NUM</pre>
byRes	Array of BYTE	Reserved, set to 0.

Remarks

By default, up to 2048 channels are supported. More channels require higher computer performance and network bandwidth.

See Also

[NET_DVR_SetSDKInitCfg](#)

A.1.24 NET_DVR_IPADDR

Structure about IP Address

Member	Data Type	Description
slpV4	char[]	IPv4 address of the device. The array length is 16 bytes.
slpV6	BYTE[]	IPv6 address of the device. The array length is 128 bytes.

A.1.25 NET_DVR_IPADDR_UNION

IP Address Union

Member	Data Type	Description
szIPv4	char[]	IPv4 address. The maximum length is 16 bytes.
szIPv6	char[]	IPv6 address. The maximum length is 256 bytes.

A.1.26 NET_DVR_LIST_INFO

Structure about the partition information list.

Structure Definition

```
struct{  
    DWORD    dwSize;  
    BYTE     byIndex;  
    BYTE     byRes[63];  
}NET_DVR_LIST_INFO, *LPNET_DVR_LIST_INFO;
```

Members

dwSize

Structure size.

byIndex

Partition No., 0xff indicates all partitions (the access control devices only support 0xff to get all partitions).

byRes

Reserved.

A.1.27 NET_DVR_LOCAL_ABILITY_PARSE_CFG

Structure about capability of analysis library configuration.

Structure Definition

```
struct{  
    BYTE        byEnableAbilityParse;  
    BYTE        byRes[127];  
}NET_DVR_LOCAL_ABILITY_PARSE_CFG, *LPNET_DVR_LOCAL_ABILITY_PARSE_CFG;
```

Members

byEnableAbilityParse

Whether to enable capability analysis library: 0-disable, 1-enable (default).

byRes

Reserved, set to 0.

Remarks

By default, the analog capability is disabled, you can enable the analog capability via this structure, and then call **NET_DVR_GetDeviceAbility** and load the "LocalXml.zip" to the directory of HCNetSDK to get the capabilities of devices.

A.1.28 NET_DVR_LOCAL_ASYNC_CFG

Structure about Asynchronous Configuration Parameter

Member	Data Type	Description
bEnable	BOOL	Whether to enable asynchronous configuration: "TRUE"-yes, "FALSE"-no (default).
byRes	Array of BYTE	Reserved, set to 0. The maximum size is 60 bytes.

Remarks

- After enabling asynchronous configuration, the notifications about disconnection and reconnection of devices will be received in asynchronous mode. This function can be adopted when you need to manage tens of thousands of devices. By default, this function is disabled.
- After enabling asynchronous configuration, the interval configuration of heartbeat interaction turns invalid (related command: "NET_SDK_LOCAL_CFG_TYPE_CHECK_DEV").
- After enabling asynchronous configuration, the API ***NET_DVR_SetConnectTime*** for setting network connection timeout turns invalid.

A.1.29 NET_DVR_LOCAL_BYTE_ENCODE_CONVERT

Structure about encoding format conversion configuration.

Structure Definition

```
struct{
    CHAR_ENCODE_CONVERT    fnCharConvertCallback
    BYTE                   byRes[256];
}NET_DVR_LOCAL_BYTE_ENCODE_CONVERT, *LPNET_DVR_LOCAL_BYTE_ENCODE_CONVERT;
```

Members

fnCharConvertCallback

Callback function of encoding type conversion, see details in ***CHAR_ENCODE_CONVERT***.

byRes

Reserved, set to 0.

Remarks

- The device character encoding type is returned by the login API.
- By default, the encoding type conversion is realized by the "libiconv.dll" of HCNetSDK, but the users can set the encoding type conversion callback via this structure and call their own encoding API to convert the encoding type.

A.1.30 NET_DVR_LOCAL_CERTIFICATION

Certificate configuration parameter structure

Structure Definition

```
struct{
    char                   szLoadPath[MAX_FILE_PATH_LEN/*256*/];
    fnCertVerifyResultCallback fnCB;
    void                   *pUserData;
```



```
    BYTE                                     byRes[64];  
}NET_DVR_LOCAL_CERTIFICATION, *LPNET_DVR_LOCAL_CERTIFICATION;
```

Members

szLoadPath

Certificate saving path.

fnCB

Certificate verification callback function, see details below.

```
typedef BOOL(CALLBACK *fnCertVerifyResultCallBack) (  
    DWORD                                     uiResult,  
    NET_DVR_CETTIFICATE_INFO                 lpCertificateInfo,  
    char                                     *pUserData  
);
```

uiResult

Certificate verification results: 0-verification failed, other values-verified.

lpCertificateInfo

Certificate information, see details in [**NET_DVR_CETTIFICATE_INFO**](#).

pUserData

User data pointer.

pUserData

User data.

byRes

Reserved, set to 0.

See Also

[**NET_SDK_LOCAL_CFG_TYPE**](#)

A.1.31 NET_DVR_LOCAL_CFG_TYPE_PTZ

PTZ interaction configuration structure.

Structure Definition

```
struct{  
    BYTE     byWithoutRecv;  
    BYTE     byRes[63];  
}NET_DVR_LOCAL_PTZ_CFG, *LPNET_DVR_LOCAL_PTZ_CFG;
```

Members

byWithoutRecv

Whether to receive the response from device: 0=yes, 1=no.

byRes

Reserved, set to 0

Remarks

This configuration is applicable to 3G network.

A.1.32 NET_DVR_LOCAL_CHECK_DEV

Heartbeat time interval configuration structure.

Structure Definition

```
struct{
    DWORD          dwCheckOnlineTimeout;
    DWORD          dwCheckOnlineNetFailMax;
    BYTE           byRes[256];
}NET_DVR_LOCAL_CHECK_DEV, *LPNET_DVR_LOCAL_CHECK_DEV;
```

Members

dwCheckOnlineTimeout

Online health monitoring time interval, unit: ms, range: 30-120 (s), 0-120s (default), the recommended value is 30s.

dwCheckOnlineNetFailMax

The maximum number of network failure attempts, if the failure attempts are larger than this threshold, exception message will be called back. 0-1 (default), the recommended value is 3.

byRes

Reserved, set to 0.

A.1.33 NET_DVR_LOCAL_GENERAL_CFG

General configurations structure.

Structure Definition

```
struct{
    BYTE           byExceptionCbDirectly;
    BYTE           byNotSplitRecordFile;
    BYTE           byResumeUpgradeEnable;
    BYTE           byAlarmJsonPictureSeparate;
    BYTE           byRes[4];
    UINT64         i64FileSize;
    DWORD          dwResumeUpgradeTimeout;
```

```
BYTE      byAlarmReconnectMode;  
BYTE      byStdXmlBufferSize;  
BYTE      byMultiplexing;  
BYTE      byFastUpgrade;  
BYTE      byRes[232];  
}NET_DVR_LOCAL_GENERAL_CFG, *LPNET_DVR_LOCAL_GENERAL_CFG;
```

Members

byExceptionCbDirectly

Exception callback type: 0-callback via thread pool, 1-callback via upper-layer.

byNotSplitRecordFile

Whether to subpackage the local video files: 0-yes (default), 1-no.

byResumeUpgradeEnable

Whether to enable upgrading ANR (Automatic Network Replenishment): 0-disable (default), 1-enable.

byAlarmJsonPictureSeparate

Whether to separate the alarm data and the alarm picture which will be transmitted in JSON format: 0-not separate, 1-separate (the **lCommand** in the callback function will be "COMM_ISAPI_ALARM").

byRes

Reserved.

i64FileSize

Maximum file size, unit: byte. When subpackaging is enabled, if the saved video file size is larger than the value of this parameter, the file will be subpackaged to multiple file segments for storage.

dwResumeUpgradeTimeout

ANR reconnection timeout, unit: millisecond.

byAlarmReconnectMode

Reconnection mode: 0-dependent thread reconnection (default), 1-thread pool reconnection.

byStdXmlBufferSize

Buffer size for receiving data transmitted by ISAPI: 1-1 MB, other values-default.

byMultiplexing

Whether to enable multiplexing of normal link (non-TLS link): 0-disable, 1-enable.

byFastUpgrade

Upgrading mode: 1-normal upgrading, 2-fast upgrading.

byRes1

Reserved.

A.1.34 NET_DVR_LOCAL_LOG_CFG

Log configuration structure.

Structure Definition

```
struct{  
    WORD        wSDKLogNum;  
    BYTE        byRes[254];  
}NET_DVR_LOCAL_LOG_CFG, *LPNET_DVR_LOCAL_LOG_CFG;
```

Members

wSDKLogNum

Number of log files in overwritten mode, "0"-10 log files (default).

byRes

Reserved, set to 0.

A.1.35 NET_DVR_LOCAL_MEM_POOL_CFG

Local configuration structure of storage pool.

Structure Definition

```
struct{  
    DWORD        dwAlarmMaxBlockNum;  
    DWORD        dwAlarmReleaseInterval;  
    BYTE        byRes[60];  
}NET_DVR_LOCAL_MEM_POOL_CFG, *LPNET_DVR_LOCAL_MEM_POOL_CFG;
```

Members

dwAlarmMaxBlockNum

The maximum number of memory blocks can be applied, the maximum size of each applied block is 64MB, if the required memory block size is larger than the threshold, do not apply for it from the system. If the value of this parameter is set to 0, it refers that the number of memory block can be applied is not limited.

dwAlarmReleaseInterval

The time interval between each free memory blocks to be released, unit: s, 0-not release the free memory.

byRes

Reserved, set to 0.

A.1.36 NET_DVR_LOCAL_MODULE_RECV_TIMEOUT_CFG

Structure about timeout configuration by module.

Structure Definition

```
struct{
    DWORD    dwPreviewTime;
    DWORD    dwAlarmTime;
    DWORD    dwVodTime;
    DWORD    dwElse;
    BYTE     byRes[512];
}NET_DVR_LOCAL_MODULE_RECV_TIMEOUT_CFG,
*LPNET_DVR_LOCAL_MODULE_RECV_TIMEOUT_CFG;
```

Members

dwPreviewTime

Live view module receiving timeout, unit: millisecond, range: 0-3000,000, 0-restore to default settings.

dwAlarmTime

Alarm module receiving timeout, unit: millisecond, range: 0-3000,000, 0-restore to default settings.

dwVodTime

Playback module receiving timeout, unit: millisecond, range: 0-3000,000, 0-restore to default settings.

dwElse

Other modules' receiving timeout, unit: millisecond, range: 0-3000,000, 0-restore to default settings.

byRes

Reserved, set to 0.

A.1.37 NET_DVR_LOCAL_PORT_MULTI_CFG

Configuration parameter structure of port multiplier.

Structure Definition

```
struct{
    BOOL     bEnable;
    BYTE     byRes[60];
}NET_DVR_LOCAL_PORT_MULTI_CFG, *LPNET_DVR_LOCAL_PORT_MULTI_CFG;
```

Members

bEnable

Whether to enable port multiplier: true=yes.

byRes

Reserved, set to 0.

See Also

[NET_SDK_LOCAL_CFG_TYPE](#)

A.1.38 NET_DVR_LOCAL_PROTECT_KEY_CFG

Key Parameter Structure

Member	Data Type	Description
byProtectKey	Array of BYTE	Key, the default value is 0. The maximum size is 128 bytes.
byRes	Array of BYTE	Reserved, set to 0. The maximum size is 128 bytes.

A.1.39 NET_DVR_LOCAL_SDK_PATH

Path Information Structure for Loading Component Libraries

Member	Data Type	Description
sPath	Array of char	Component libraries' addresses
byRes	Array of BYTE	Reserved.

Remarks

If the path of HCNetSDKCom folder and HCNetSDK libraries are same, but the path of executable programs are different, you can call [NET_DVR_SetSDKInitCfg](#) to specify the path of HCNetSDKCom folder to make sure the component libraries are loaded normally.

A.1.40 NET_DVR_LOCAL_STREAM_CALLBACK_CFG

Key Parameter Structure

Member	Data Type	Description
byPlayBackEndFlag	BYTE	Whether to call back playback end flag:0-No, 1-Yes
byRes	Array of BYTE	Reserved, set to 0. The maximum size is 255 bytes.

A.1.41 NET_DVR_LOCAL_TALK_MODE_CFG

Two-way audio configuration structure.

Structure Definition

```
struct{
    BYTE    byTalkMode;
    BYTE    byRes[127];
}NET_DVR_LOCAL_TALK_MODE_CFG, *LPNET_DVR_LOCAL_TALK_MODE_CFG;
```

Members

byTalkMode

Two-way audio mode: 0-enable two-way audio library (default), 1-enable Windows API mode.

byRes

Reserved, set to 0.

Remarks

If the two-way audio library is enabled, you must load the "AudioIntercom.dll" and "OpenAL32.dll".

A.1.42 NET_DVR_LOCAL_TCP_PORT_BIND_CFG

Local binding configuration structure of TCP port.

Structure Definition

```
struct{
    WORD    wLocalBindTcpMinPort;
    WORD    wLocalBindTcpMaxPort;
    BYTE    byRes[60];
}NET_DVR_LOCAL_TCP_PORT_BIND_CFG, *LPNET_DVR_LOCAL_TCP_PORT_BIND_CFG;
```

Members

wLocalBindTcpMinPort

The minimum TCP port number to be bound locally.

wLocalBindTcpMaxPort

The maximum TCP port number to be bound locally.

byRes

Reserved, set to 0.

Remarks

- Port bind strategy: provide a port number segment to ensure all used port numbers are in the segment (except multicast); the ports from port pool are tried to bind one by one until the port is not occupied, if all ports are occupied, error will be returned; binding the system reserved ports (form 1 to 1024) is not suggested.
- The maximum port number to be bound should be equal to or larger than the minimum port number, [0,0]: clear the binding; [0,non-0]: setting failed, as 0 can't be bound.

A.1.43 NET_DVR_LOCAL_UDP_PORT_BIND_CFG

Local binding configuration structure of UDP port.

Structure Definition

```
struct{
    WORD        wLocalBindUdpMinPort;
    WORD        wLocalBindUdpMaxPort;
    BYTE        byRes[60];
}NET_DVR_LOCAL_UDP_PORT_BIND_CFG, *LPNET_DVR_LOCAL_UDP_PORT_BIND_CFG;
```

Members

wLocalBindUdpMinPort

The minimum UDP port number to be bound locally.

wLocalBindUdpMaxPort

The maximum UDP port number to be bound locally.

byRes

Reserved, set to 0.

Remarks

- Port bind strategy: provide a port number segment to ensure all used port numbers are in the segment (except multicast); the ports from port pool are tried to bind one by one until the port

is not occupied, if all ports are occupied, error will be returned; binding the system reserved ports (form 1 to 1024) is not suggested.

- The maximum port number to be bound should be equal to or larger than the minimum port number, [0,0]: clear the binding; [0,non-0]: setting failed, as 0 can't be bound.

A.1.44 NET_DVR_MESSAGE_CALLBACK_PARAM_V51

Alarm Callback Configuration Parameters

Key Parameter Structure

Member	Data Type	Description
byVcaAlarmJsonType	BYTE	JSON format for alarm transmission (COMM_VCA_ALARM): 0-new JSON format, 1-old JSON format.
byRes	Array of BYTE	Reserved, set to 0. The maximum size is 63 bytes.

A.1.45 NET_DVR_MIME_UNIT

Input Content Details Structure of Message Transmission API (NET_DVR_STDXMLConfig)

Member	Data Type	Description
szContentType	Array of char	Content type (corresponds to Content-Type field in the message), e.g., text/json. text/xml, and so on. The content format must be supported by HTTP.
szName	Array of char	Content name (corresponds to name field in the message), e.g., name="upload".
szFilename	Array of char	Content file name (corresponds to filename field in the message), e.g., filename="C:\Users\test\Desktop\11.txt".
dwContentLen	DWORD	Content size
pContent	char*	Data point

Member	Data Type	Description
bySelfRead	BYTE	0-External file, 1-Internal data, whose address is specified by szFilename .
byRes	Array of BYTE	Reserved. Set to 0. Maximum: 15 bytes.

See Also

NET_DVR_XML_CONFIG_INPUT

A.1.46 NET_DVR_MODULE_INFO

Structure about All Modules Information

Member	Data Type	Description
dwSize	DWORD	Structure size.
byModuleType	BYTE	Module type: 1-keyboard, 2-trigger, 3-zone, 4-network module.
byKeyBoardType	BYTE	Keyboard type: 1-LCD, 2-LED.
byTriggerType	BYTE	Trigger type: 1-"local trigger", 2-"4-channel trigger", 3-"8-channel trigger", 4-"single zone trigger", 5-"32-channel trigger".
byZoneType	BYTE	Zone type: 1-"local zone", 2-"single zone", 3-"double zone", 4-"8 zone", 5-"8-channel analog zone".
wModuleAddress	WORD	External trigger address, extended module: from 0 to 255, 0xFFFF indicates invalid address.
byRes2	Array of BYTE	Reserved, and set to 0. The maximum size is 2 bytes.
sModelInfo	Array of char	Module information. The maximum size is 32 bytes.

Member	Data Type	Description
sDeviceVersionInfo	Array of char	Version information. The maximum size is 32 bytes.
byRes	Array of BYTE	Reserved, and set to 0. The maximum size is 128 bytes.

Related API

NET_DVR_GetNextRemoteConfig

A.1.47 NET_DVR_PHONECENTERDIALCFG

Center parameter structure.

Structure Definition

```
struct{
    BYTE          sCenterName[NAME_LEN/*32*/];
    BYTE          byPhoneNum[MAX_PHONE_NUM/*32*/];
    BYTE          byRepeatCall;
    BYTE          byPstnProtocol;
    BYTE          byDialDelay;
    BYTE          byPstnTransMode;
    BYTE          byEnable;
    BYTE          byRes1[5];
    BYTE          byReceiverId[6];
    BYTE          byRes2[32];
}NET_DVR_PHONECENTERDIALCFG, *LPNET_DVR_PHONECENTERDIALCFG;
```

Members

sCenterName

Center name.

byPhoneNum

Telephone number.

byRepeatCall

Repeated dial-up times, value range: 1~15 times.

byPstnProtocol

Communication protocol: 0-CID.

byDialDelay

Dial-up delay, value range: 0~150s.

byPstnTransMode

Transmission mode: 0-DTMF 5/S, 1-DTMF 10/S.

byEnable

Whether to enable, 0-disable, 1-enable.

byRes1

Reserved, set to 0.

byReceiverId

Receiver account ID. It consists of up to 6 characters including 0 to 9, a to f, and A to F. When the protocol is "CID", the valid length is 4.

byRes2

Reserved, set to 0.

Remarks

- Up to two centers' phone numbers are supported, and they can be the same one. For two centers (a main center and a backup center), the dial-up will only be transmitted to the first center (accounts of the two centers can be different).
- The phone number of each center consists of 31 characters including 0 to 9 and F, and the end character is "E".
- For the two centers, their default phone numbers are both "E000000000000000000000000000000" (the number of 0 is 31).

A.1.48 NET_DVR_RECORD_PASSBACK_MANUAL_COND

Structure About Conditions of Getting Task of Manually Copying Back Videos

Member	Data Type	Description
dwSize	DWORD	Structure size.
byType	BYTE	Method of getting the task information: 0 (get remaining tasks), 1 (get remaining tasks by stream ID), 2 (get all tasks), 3 (get all tasks by stream ID).
byRes1	BYTE	Reserved, set to 0. The size is 3 bytes.
struStreamInfo	<u>NET_DVR_STREAM_INFO</u>	Stream information structure. This member is valid when getting the task information by stream ID.
byRes	Array of BYTE	Reserved, set to 0. The size is 128 bytes.

A.1.49 NET_DVR_RECORD_PASSBACK_MANUAL_TASK_RET

Structure About Results of Getting Task of Manually Copying Back Videos

Member	Data Type	Description
dwSize	DWORD	Structure size.
struStreamInfo	<u>NET_DVR_STREAM_INFO</u>	Stream information structure. This member is valid when getting the task information by stream ID.
dwTaskID	DWORD	Task ID
struStartTime	<u>NET_DVR_TIME_EX</u>	Start time of video copy-back
struStopTime	<u>NET_DVR_TIME_EX</u>	End time of video copy back
byTaskStatus	BYTE	Task status: 0 (not executed), 1 (pausing), 2 (executed), 3 (copying back), 4 (copy-back failed), 5 (succeeded, but only some videos are copied back), 6 (succeeded, but there is no video in the camera).
byRes1	Array of BYTE	Reserved, set to 0. The size is 3 bytes.
struExecuteStartTime	<u>NET_DVR_TIME_EX</u>	Actual start time of executing the task. This member is valid when the value of byTaskStatus is 1 or 2.
struExecuteStopTime	<u>NET_DVR_TIME_EX</u>	Actual end time of executing the task. This member is valid when the value of byTaskStatus is 1 or 2.
byRes	Array of BYTE	Reserved, set to 0. The size is 128 bytes.

A.1.50 NET_DVR_REMOTECONTROLLER_PERMISSION_CFG

Keyfob User Parameters Structure

Member	Data Type	Description
dwSize	DWORD	Structure size.
byEnable	BYTE	Enable or not: 0-no, 1-yes.

Member	Data Type	Description
byRes1	BYTE	Reserved, set to 0.
wRemoteCtrllerID	WORD	Keyfob ID, starts from 1.
sDevSn	Array of BYTE	Product serial No. The maximum size is 16 bytes.
byArmRight	BYTE	With arming permission or not? 0-no, 1-yes.
byDisArmRight	BYTE	With disarming permission or not? 0-no, 1-yes.
byArmReportRight	BYTE	With permission of uploading arming report or not? 0-no, 1-yes.
byDisArmReportRight	BYTE	With permission of uploading disarming report or not? 0-no, 1-yes.
byClearAlarmRight	BYTE	With alarm clearing permission or not? 0-no, 1-yes.
bySubSystemID	BYTE	Keyfob subsystem No., ranges from 1 to 8.
byKeyboardAddr	BYTE	Keyfob address.
byEnableDel	BYTE	Enable deleting keyfob user: 0-no, 1-yes.
byAlwaysOpenRight	BYTE	Whether door remaining open is allowed: 0-invalid, 1-no, 2-yes.
byOpeningDirection	BYTE	Door opening direction: 0-invalid, 1-entrance, 2-exit.
byRes2	Array of BYTE	Reserved, set to 0. The maximum size is 62 bytes.

A.1.51 NET_DVR_RTSP_PARAMS_CFG

RTSP Parameter Structure

Member	Data Type	Description
dwMaxBuffRoomNum	DWORD	Maximum number of buffers for RTP over UDP sorting, the default value is 20. If the value is 0, it indicates that the member is invalid. One buffer size is about 10 KB, more number of buffers indicates higher sorting ability, more fluent, and longer delay.
byUseSort	BYTE	Whether to enable RTP over UDP sorting: 0-no, 1-yes.
byRes	Array of BYTE	Reserved, set to 0. The maximum size is 123 bytes.

A.1.52 NET_DVR_SCHEDTIME

Structure About Start and End Time Parameters

Member	Data Type	Description
byStartHour	BYTE	Start time: hour.
byStartMin	BYTE	Start time: minute.
byStopHour	BYTE	End time: hour.
byStopMin	BYTE	End time: minute.

A.1.53 NET_DVR_SIMXML_LOGIN

Structure about Complement Fields by Stimulation Capability

Member	Data Type	Description
byLoginWithSimXml	BYTE	Whether to complement fields by stimulation capability: 0-no, 1-yes.
byRes	Array of BYTE	Reserved, set to 0. The maximum size is 127 bytes.

A.1.54 NET_DVR_SIP_CFG_V50

SIP (Session Initiation Protocol) parameter structure.

Structure Definition

```
struct{
    DWORD                dwSize;
    BYTE                 byEnableAutoLogin;
    BYTE                 byLoginStatus;
    BYTE                 byRes1[2];
    NET_DVR_IPADDR       stuServerIP;
    WORD                 wServerPort;
    BYTE                 byRes2[2];
    BYTE                 byUserName[NAME_LEN/*32*/];
    BYTE                 byPassWord[PASSWD_LEN/*16*/];
    BYTE                 byLocalNo[MAX_NUMBER_LEN/*32*/];
    BYTE                 byDispalyName[MAX_NAME_LEN/*128*/];
    WORD                 wLocalPort;
    BYTE                 byLoginCycle;
    BYTE                 byRes3;
    BYTE                 bySIPServerDomain[MAX_DOMAIN_NAME/*64*/];
    NET_DVR_IPADDR       stuSTUNServerIP;
    BYTE                 bySTUNServerDomain[MAX_DOMAIN_NAME/*64*/];
    WORD                 wSTUNServerPort;
    BYTE                 byRes4[2];
    NET_DVR_IPADDR       stuProxyServerIP;
    BYTE                 byProxyServerDomain[MAX_DOMAIN_NAME/*64*/];
    WORD                 wProxyServerPort;
    BYTE                 byNetWork;
    BYTE                 byRes5;
    BYTE                 byCalledTargetName[NET_SDK_MAX_CALLEDTARGET_NAME/*32*/];
    BYTE                 byRes[224];
}NET_DVR_SIP_CFG_V50, *LPNET_DVR_SIP_CFG_V50;
```

Members

dwSize

Structure size.

byEnableAutoLogin

Whether to enable registering automatically: 0-disable, 1-enable.

byLoginStatus

Login status: 0-unregistered,1-registered. This parameter can only be obtained.

byRes1

Reserved, set to 0.

stuServerIP

IP address of the SIP server.

wServerPort

Port No. of the SIP server.

byRes2

Reserved, set to 0.

byUserName

User name.

byPassWord

Password.

byLocalNo

Local No.

byDispalyName

Displayed name.

wLocalPort

Local port.

byLoginCycle

Registration period, value range: [1, 99], unit: minute.

byRes3

Reserved, set to 0.

bySIPServerDomain

SIP server domain name

Domain name of the SIP server. Only one member of **stuServerIP** and **bySIPServerDomain** should be configured; if both are configured, **stuServerIP** will be preferred.

stuSTUNServerIP

IP address of the STUN server.

bySTUNServerDomain

Domain name of the STUN server.

wSTUNServerPort

Port No. of the STUN server.

byRes4

Reserved, set to 0.

stuProxyServerIP

IP address of the proxy server.

byProxyServerDomain

Domain name of the proxy server. Only one member of **stuProxyServerIP** and **byProxyServerDomain** should be configured; if both are configured, the **stuProxyServerIP** will be preferred.

wProxyServerPort

Port No. of the proxy server.

byNetWork

Network type: 0-invalid, 1-wired network 1, 2-wired network 2, 3-wireless network. If this member is set to wired network, only the wired network will be used no matter whether the wireless network is normal or not; if this member is set to wireless network, only the wireless network will be used.

byRes5

Reserved, set to 0.

byCalledTargetName

User name of the called person.

byRes

Reserved, set to 0.

A.1.55 NET_DVR_STD_ABILITY

Input and Output Parameter Structure for Getting Capabilities

Member	Data Type	Description
lpCondBuffer	LPVOID	Condition parameters (ASCII character format), e.g., the channel No., it can be set to "null".
dwCondSize	DWORD	Buffer size of condition parameters.
lpOutBuffer	LPVOID	Output parameters buffer (the parameter is returned in the message with XML format), it cannot be set to "null".
dwOutSize	DWORD	Output buffer size.
lpStatusBuffer	LPVOID	Get the returned status parameters (<i><u>XML_ResponseStatus</u></i>) when getting capabilities failed. It can be set to null.
dwStatusSize	DWORD	Status buffer size.
dwRetSize	DWORD	Obtained data size (if the capability is obtained, the value refers to the size of lpOutBuffer ; if

Member	Data Type	Description
		getting failed, the value refers to the size of lpStatusBuffer).
byRes	Array [BYTE]	Reserved. The maximum size is 32 bytes.

Remarks

For different capability types (which depend on the parameter **dwAbilityType** in the API **NET_DVR_GetSTDAbility**), the condition parameter **lpCondBuffer** and output parameter **lpOutBuffer** are different. For details, refer to the typical applications.

A.1.56 NET_DVR_STD_CONFIG

Structure About Configuring Input and Output Parameters

Member	Data Type	Description
lpCondBuffer	LPVOID	Condition parameters, e.g., channel No., it can be set to "NULL".
dwCondSize	DWORD	Size of buffer for storing condition parameters
lpInBuffer	LPVOID	Input parameters (a structure)
dwInSize	DWORD	Size of buffer for storing input parameters
lpOutBuffer	LPVOID	Output parameters (a structure)
dwOutSize	DWORD	Size of buffer for storing output parameters
lpStatusBuffer	LPVOID	Returned status parameters in XML format, it can be set to NULL.
dwStatusSize	DWORD	Size of buffer for storing status parameters
lpXmlBuffer	LPVOID	Request or response message in XML format, it is valid when byDataType is 1.
dwXmlSize	DWORD	Size of memory pointed by lpXmlBuffer .
byDataType	BYTE	Input or output parameter type: 0-valid when the input or output parameters is a structure; 1-valid when the input or output parameters is a XML message.
byRes	Array [BYTE]	Reserved, set to 0. The maximum size is 32 bytes.

A.1.57 NET_DVR_STREAM_INFO

Stream information structure.

Structure Definition

```
struct{  
    DWORD      dwSize;  
    BYTE       byID[STREAM_ID_LEN/*32*/];  
    DWORD      dwChannel;  
    BYTE       byRes[32];  
}NET_DVR_STREAM_INFO, *LPNET_DVR_STREAM_INFO;
```

Members

dwSize

Structure size.

byID

Stream ID, which consists of letters, digits, and dashes, 0-invalid.

dwChannel

Linked device channel. When it is 0xffffffff, if setting the stream source, this parameter indicates that no device channel is linked; if setting configuration condition, this parameter is invalid.

byRes

Reserved, set to 0.

Remarks

- If the device does not support marking stream ID, e.g., DVR, the parameter **byID** should be set to 0.
- For transcoder, when setting the stream source, only one of **byID** and **dwChannel** can be valid; when transcoding, both the **byID** and **dwChannel** can be invalid, the transcoding channel or stream ID is automatically allocated by device.
- For other devices (e.g., CVR), when this structure is inputted as configuration condition, if both the **byID** and **dwChannel** are invalid, error code (17) will be returned, if they are valid, but mismatched, error may also be returned, so only setting one of these two parameters is suggested.

A.1.58 NET_DVR_TIME

Time Parameter Structure

Member	Data Type	Description
dwYear	DWORD	Year
dwMonth	DWORD	Month
dwDay	DWORD	Day
dwHour	DWORD	Hour
dwMinute	DWORD	Minute
dwSecond	DWORD	Second

A.1.59 NET_DVR_TIME_EX

Extended Time Parameter Structure

Member	Data Type	Description
wYear	WORD	Year
byMonth	BYTE	Month
byDay	BYTE	Day
byHour	BYTE	Hour
byMinute	BYTE	Minute
bySecond	BYTE	Second
byRes	BYTE	Reserved.

A.1.60 NET_DVR_USER_LOGIN_INFO

Structure About Login Parameters

Member	Data Type	Description
sDeviceAddress	char	Device IP address, or domain name.
byUseTransport	BYTE	Enable capability transmission or not: 0-no (default), 1-yes.

Member	Data Type	Description
wPort	WORD	Device port number, e.g., 8000 (when login by private protocol), 80 (when login by text protocol).
sUserName	char	User name for logging in to device.
sPassword	char	Login password.
cbLoginResult	<u>fLoginResultCallback</u>	Callback function used to return login status, it is valid only when bUseAsynLogin is "1".
pUser	void*	User data.
bUseAsynLogin	BOOL	Whether to enable asynchronous login: 0-no, 1-yes.
byProxyType	BYTE	Proxy server type: 0-no proxy, 1-standard proxy, 2-EHome proxy.
byUseUTCTime	BYTE	0-not convert (default), 1-input or output UTC time, 2-input or output local time.
byLoginMode	BYTE	Login mode: 0-login by private protocol, 1-login by text protocol, 2-self-adaptive (it is available when the protocol type supported by device is unknown, and this mode does not support asynchronous login).
byHttps	BYTE	Whether to enable TLS for login (by private protocol or by text protocol): 0-no, 1-yes, 2-self-adaptive (which is usually used when the protocol type supported by device is unknown. Both HTTP and HTTPS requests will be sent).
iProxyID	LONG	Proxy server No.
byVerifyMode	BYTE	Whether to enable verification mode: 0-no, 1-bidirectional verification (currently not available), 2-unidirectional verification (it is valid when byLoginMode is 0 and byHttps is 1); when byVerifyMode is 0, CA certificate is not required, when byVerifyMode is 2, you should call NET_DVR_SetSDKLocalCfg to load CA certificate, and the enumeration value is "NET_SDK_LOCAL_CFG_CERTIFICATION".
byRes3	BYTE[]	Reserved, the maximum length is 119 bytes.

fLoginResultCallback**Login Status Callback Function**

Member	Data Type	Description
lUserID	LONG	User ID, which is returned by <u>NET_DVR_Login_V40</u> .
dwResult	DWORD	Login status: 0-asynchronously logging in failed, 1-asynchronously logged in.
lpDeviceInfo	<u>NET_DVR_DEVICEINFO_V40</u>	Device information, such as serial No., channel, capability, and so on.
pUser	void*	User data.

A.1.61 NET_DVR_XML_CONFIG_INPUT**Input Parameter Structure of Message Transmission API (NET_DVR_STDXMLConfig)**

Member	Data Type	Description
dwSize	DWORD	Structure size.
lpRequestUrl	void*	Request URL (command) for implement different functions, and it is in string format.
dwRequestUrlLen	DWORD	Request URL size.
lpInBuffer	void*	Buffer for storing input parameters (request messages), see the input content details structure in <u>NET_DVR_MIME_UNIT</u> .
dwInBufferSize	DWORD	Input buffer size.
dwRecvTimeOut	DWORD	Receiving timeout, unit: ms, 0-5000ms (default).
byForceEncript	BYTE	Whether to enable force encryption (the messages will be encrypted by AES algorithm for transmission): 0-no, 1-yes.
byNumOfMultiPart	BYTE	Number of message segments: 0-invalid; other values-number of message segments, which is

Member	Data Type	Description
		transmitted by the parameter lpInBuffer in the structure <i>NET_DVR_MIME_UNIT</i> .
byRes	Array of BYTE	Reserved, set to 0.

Related API

NET_DVR_STDXMLConfig

A.1.62 NET_DVR_XML_CONFIG_OUTPUT

Output Parameter Structure of Message Transmission API (NET_DVR_STDXMLConfig)

Member	Data Type	Description
dwSize	DWORD	Structure size.
lpOutBuffer	void*	Buffer for storing output parameters (response messages), which is allocated when passing through URL by GET method.
dwOutBufferSize	DWORD	Output buffer size.
dwReturnedXMLSize	DWORD	Actual size of response message.
lpStatusBuffer	void*	Response status (ResponseStatus message). This parameter will not be assigned if performing GET operation succeeded, and you can also set it to "NULL" if not required.
dwStatusSize	DWORD	Size of response status buffer.
lpDataBuffer	HPR_VOIDPTR	Buffer for transmitted data. This parameter is valid when the value of byNumOfMultiPart is larger than 0.
byNumOfMultiPart	HPR_UINT8	Number of parts that the message is divided into.
byRes [23]	BYTE	Reserved, set to 0.

Related API

NET_DVR_STDXMLConfig

A.1.63 NET_SDK_CALLBACK_STATUS_NORMAL

Enumeration About Persistent Connection Status

Enumeration Type	Marco Definition Value	Description
NET_SDK_CALLBACK_STATUS_SUCCESS	1000	Succeeded.
NET_SDK_CALLBACK_STATUS_PROCESSING	1001	Connecting. The lpBuffer is 4-byte status.
NET_SDK_CALLBACK_STATUS_FAILED	1002	Failed. The lpBuffer is the value of 4-byte status and 4-byte error code.

A.1.64 NET_SDK_LOCAL_CFG_TYPE

Enumerate the local configuration types of device network SDK.

Enumeration Definition

```
enum{
    NET_SDK_LOCAL_CFG_TYPE_TCP_PORT_BIND          =0,
    NET_SDK_LOCAL_CFG_TYPE_UDP_PORT_BIND          =1,
    NET_SDK_LOCAL_CFG_TYPE_MEM_POOL                =2,
    NET_SDK_LOCAL_CFG_TYPE_MODULE_RECV_TIMEOUT    =3,
    NET_SDK_LOCAL_CFG_TYPE_ABILITY_PARSE          =4,
    NET_SDK_LOCAL_CFG_TYPE_TALK_MODE              =5,
    NET_SDK_LOCAL_CFG_TYPE_PROTECT_KEY            =6,
    NET_SDK_LOCAL_CFG_TYPE_CFG_VERSION            =7,
    NET_SDK_LOCAL_CFG_TYPE_RTSP_PARAMS            =8,
    NET_SDK_LOCAL_CFG_TYPE_SIMXML_LOGIN           =9,
    NET_SDK_LOCAL_CFG_TYPE_CHECK_DEV              =10,
    NET_SDK_LOCAL_CFG_TYPE_SECURITY               =11,
    NET_SDK_LOCAL_CFG_TYPE_EZVIZLIB_PATH          =12,
    NET_SDK_LOCAL_CFG_TYPE_CHAR_ENCODE            =13,
    NET_SDK_LOCAL_CFG_TYPE_PROXYS                 =14,
    NET_DVR_LOCAL_CFG_TYPE_LOG                    =15,
    NET_DVR_LOCAL_CFG_TYPE_STREAM_CALLBACK        =16,
    NET_DVR_LOCAL_CFG_TYPE_GENERAL               =17,
    NET_DVR_LOCAL_CFG_TYPE_PTZ                   =18,
    NET_DVR_LOCAL_CFG_MESSAGE_CALLBACK_V51        =19,
    NET_SDK_LOCAL_CFG_CERTIFICATION               =20,
    NET_SDK_LOCAL_CFG_PORT_MULTIPLEX              =21,
```

```
NET_SDK_LOCAL_CFG_ASYNC                =22
}NET_SDK_LOCAL_CFG_TYPE
```

Members

NET_SDK_LOCAL_CFG_TYPE_TCP_PORT_BIND

Local binding configuration of TCP port, see details in [**NET_DVR_LOCAL_TCP_PORT_BIND_CFG**](#).

NET_SDK_LOCAL_CFG_TYPE_UDP_PORT_BIND

Binding configuration of local UDP port, see details in [**NET_DVR_LOCAL_UDP_PORT_BIND_CFG**](#).

NET_SDK_LOCAL_CFG_TYPE_MEM_POOL

Local configuration of storage pool, see details in [**NET_DVR_LOCAL_MEM_POOL_CFG**](#).

NET_SDK_LOCAL_CFG_TYPE_MODULE_RECV_TIMEOUT

Timeout configuration by module, see details in [**NET_DVR_LOCAL_MODULE_RECV_TIMEOUT_CFG**](#).

NET_SDK_LOCAL_CFG_TYPE_ABILITY_PARSE

Capability analysis library configuration, see details in [**NET_DVR_LOCAL_ABILITY_PARSE_CFG**](#).

NET_SDK_LOCAL_CFG_TYPE_TALK_MODE

Two-way audio configuration, see details in [**NET_DVR_LOCAL_TALK_MODE_CFG**](#).

NET_SDK_LOCAL_CFG_TYPE_PROTECT_KEY

Key configuration, see details in [**NET_DVR_LOCAL_PROTECT_KEY_CFG**](#).

NET_SDK_LOCAL_CFG_TYPE_CFG_VERSION

Check the device compatibility when setting parameters.

NET_SDK_LOCAL_CFG_TYPE_RTSP_PARAMS

RTSP parameters, see details in [**NET_DVR_RTSP_PARAMS_CFG**](#).

NET_SDK_LOCAL_CFG_TYPE_SIMXML_LOGIN

Parameters of using stimulation capability to complement fields, see details in [**NET_DVR_SIMXML_LOGIN**](#).

NET_SDK_LOCAL_CFG_TYPE_CHECK_DEV

Heartbeat time interval, see details in [**NET_DVR_LOCAL_CHECK_DEV**](#).

NET_SDK_LOCAL_CFG_TYPE_SECURITY

SDK security parameters.

NET_SDK_LOCAL_CFG_TYPE_EZVIZLIB_PATH

Communication library address of EZVIZ cloud.

NET_SDK_LOCAL_CFG_TYPE_CHAR_ENCODE

Encoding format conversion configuration, see details in [**NET_DVR_LOCAL_BYTE_ENCODE_CONVERT**](#).

NET_SDK_LOCAL_CFG_TYPE_PROXYS

Proxy types.

NET_DVR_LOCAL_CFG_TYPE_LOG

Log parameters, see details in [NET_DVR_LOCAL_LOG_CFG](#) .

NET_DVR_LOCAL_CFG_TYPE_STREAM_CALLBACK

Stream callback parameters, see details in [NET_DVR_LOCAL_STREAM_CALLBACK_CFG](#) .

NET_DVR_LOCAL_CFG_TYPE_GENERAL

General parameters, see details in [NET_DVR_LOCAL_GENERAL_CFG](#) .

NET_DVR_LOCAL_CFG_TYPE_PTZ

PTZ interaction parameters, see details in [NET_DVR_LOCAL_CFG_TYPE_PTZ](#) .

NET_DVR_LOCAL_CFG_MESSAGE_CALLBACK_V51

Local parameters of alarm callback, see details in [NET_DVR_MESSAGE_CALLBACK_PARAM_V51](#) .

NET_SDK_LOCAL_CFG_CERTIFICATION

Certificate parameters, see details in [NET_DVR_LOCAL_CERTIFICATION](#) .

NET_SDK_LOCAL_CFG_PORT_MULTIPLEX

Port multiplier parameters, see details in [NET_DVR_LOCAL_PORT_MULTI_CFG](#) .

NET_SDK_LOCAL_CFG_ASYNC

Asynchronous mode parameters, see details in [NET_DVR_LOCAL_ASYNC_CFG](#) .

A.2 Request URIs

Description	URI	Method	Request and Response Message
Get device information.	/ISAPI/System/deviceInfo	GET	XML_DeviceInfo XML_ResponseStatus
Edit device information.	/ISAPI/System/deviceInfo	PUT	-
Control PTZ.	/ISAPI/PTZCtrl/channels/<ID>/continuous	PUT	XML_ResponseStatus
Get preset list.	/ISAPI/PTZCtrl/channels/<ID>/presets	GET	XML_PTZPresetList XML_ResponseStatus
Manage all configured presets.	/ISAPI/PTZCtrl/channels/<ID>/presets	POST	-

Delete all presets.	/ISAPI/PTZCtrl/channels/<ID>/presets	DELETE	-
Add a preset.	/ISAPI/PTZCtrl/channels/<ID>/presets/<ID>	PUT	XML_ResponseStatus
Delete a preset.	/ISAPI/PTZCtrl/channels/<ID>/presets/<ID>	DELETE	XML_ResponseStatus
Get a preset.	/ISAPI/PTZCtrl/channels/<ID>/presets/<ID>	GET	-
Call a preset.	/ISAPI/PTZCtrl/channels/<ID>/presets/<ID>/goto	PUT	XML_ResponseStatus
Get partition status.	/ISAPI/SecurityCP/status/subSystems?format=json	GET	JSON_SubSysList JSON_ResponseStatus
Arm a partition.	/ISAPI/SecurityCP/control/arm/<ID>?ways=<string>&format=json	PUT	JSON_ResponseStatus
Disarm a partition.	/ISAPI/SecurityCP/control/disarm/<ID>?format=json	PUT	JSON_ResponseStatus
Clear partition alarms.	/ISAPI/SecurityCP/control/clearAlarm/<ID>?format=json	PUT	JSON_ResponseStatus
Get zone status	/ISAPI/SecurityCP/status/zones?format=json	GET	JSON_ZoneList JSON_ResponseStatus
Search partition status according to conditions.	/ISAPI/SecurityCP/status/zones?format=json	POST	-
Zone bypass.	/ISAPI/SecurityCP/control/bypass?format=json	PUT	JSON_ResponseStatus
Recover bypass of multiple zones.	/ISAPI/SecurityCP/control/bypassRecover?format=json	PUT	JSON_ResponseStatus
Get relay status by specific conditions.	/ISAPI/SecurityCP/status/outputStatus?format=json	POST	JSON_OutputSearch JSON_ResponseStatus
Control relay in batch.	/ISAPI/SecurityCP/control/outputs?format=json	POST	JSON_ResponseStatus
Get the information of all I/O output ports.	/ISAPI/System/IO/outputs	GET	XML_IOOutputPortList XML_ResponseStatus

Get status of a specific alarm output.	/ISAPI/System/IO/outputs/<ID>/status	GET	XML_IOPortStatus XML_ResponseStatus
Manually trigger a specific alarm output.	/ISAPI/System/IO/outputs/<ID>/trigger	PUT	XML_ResponseStatus
Get device time zone.	/ISAPI/System/time	GET	XML_TimeData XML_ResponseStatus
Get or set device time parameters.	/ISAPI/System/time	PUT	-
Operations about management of all digital channels.	/ISAPI/ContentMgmt/InputProxy/channels	GET	XML_InputProxyChannelList XML_ResponseStatus
Configure operations about management of all digital channels.	/ISAPI/ContentMgmt/InputProxy/channels	PUT	-
Create digital channels	/ISAPI/ContentMgmt/InputProxy/channels	POST	-
Get status of all digital channels.	/ISAPI/ContentMgmt/InputProxy/channels/status	GET	XML_InputProxyChannelStatusList XML_ResponseStatus
Refresh the video mode manually before playback.	/ISAPI/ContentMgmt/record/control/manualRefresh/channels/<ID>	PUT	XML_ResponseStatus
Search for access control events.	/ISAPI/AccessControl/AcsEvent?format=json	POST	JSON_AcsEvent XML_ResponseStatus
Search for person information.	/ISAPI/AccessControl/UserInfo/Search?format=json	POST	JSON_UserInfoSearch XML_ResponseStatus

A.2.1 /ISAPI/SecurityCP/BasicParam/AlarmLampSchedTimeConfig

Get or set the parameters of the alarm lamp flickering schedule.

Request URI Definition

Table A-16 GET /ISAPI/SecurityCP/BasicParam/AlarmLampSchedTimeConfig

Method	GET
Description	Get the parameters of the alarm lamp flickering schedule.
Query	None.
Request	None.
Response	Succeeded: <u><i>XML_LampSchedTimeList</i></u> Failed: <u><i>XML_ResponseStatus</i></u>

Table A-17 PUT /ISAPI/SecurityCP/BasicParam/AlarmLampSchedTimeConfig

Method	PUT
Description	Set the parameters of the alarm lamp flickering schedule.
Query	None.
Request	<u><i>XML_LampSchedTimeList</i></u>
Response	<u><i>XML_ResponseStatus</i></u>

A.2.2 /ISAPI/SecurityCP/BasicParam/AlarmLampSchedTimeConfig/capabilities

Get the configuration capability of the alarm lamp flickering schedule.

Request URI Definition

Table A-18 GET /ISAPI/SecurityCP/BasicParam/AlarmLampSchedTimeConfig/capabilities

Method	GET
Description	Get the configuration capability of the alarm lamp flickering schedule.
Query	None.
Request	None.
Response	Succeeded: <u><i>XML_Cap_LampSchedTimeList</i></u> Failed: <u><i>XML_ResponseStatus</i></u>

A.2.3 /ISAPI/SecurityCP/BasicParam/audioFileList/capabilities

Get the capability of getting the audio file list.

Request URI Definition

Table A-19 GET /ISAPI/SecurityCP/BasicParam/audioFileList/capabilities

Method	GET
Description	Get the capability of getting the audio file list.
Query	None.
Request	None.
Response	Succeeded: <u><i>XML_Cap_AudioFileList</i></u> Failed: <u><i>XML_ResponseStatus</i></u>

A.2.4 /ISAPI/SecurityCP/BasicParam/audioFileList/type=

Get the audio file list.

Request URI Definition

Table A-20 GET /ISAPI/SecurityCP/BasicParam/audioFileList/type=

Method	GET
Description	Get the audio file list.
Query	type : string, audio file type, the values available can be obtained from the capability set.
Request	None.
Response	Succeeded: <u><i>XML_AudioFileList</i></u> Failed: <u><i>XML_ResponseStatus</i></u>

A.2.5 /ISAPI/SecurityCP/BasicParam/audioInOutCfg

Get or set the audio input and output parameters.

Request URI Definition

Table A-21 GET /ISAPI/SecurityCP/BasicParam/audiolnOutCfg

Method	GET
Description	Get the audio input and output parameters.
Query	None.
Request	None.
Response	Succeeded: <u><i>XML_AudioInOutCfg</i></u> Failed: <u><i>XML_ResponseStatus</i></u>

Table A-22 PUT /ISAPI/SecurityCP/BasicParam/audiolnOutCfg

Method	PUT
Description	Set the audio input and output parameters.
Query	None.
Request	<u><i>XML_AudioInOutCfg</i></u>
Response	<u><i>XML_ResponseStatus</i></u>

A.2.6 /ISAPI/SecurityCP/BasicParam/audiolnOutCfg/capabilities

Get the configuration capability of the audio input and output.

Request URI Definition

Table A-23 GET /ISAPI/SecurityCP/BasicParam/audiolnOutCfg/capabilities

Method	GET
Description	Get the configuration capability of the audio input and output.
Query	None.
Request	None.
Response	Succeeded: <u><i>XML_Cap_AudioInOutCfg</i></u> Failed: <u><i>XML_ResponseStatus</i></u>

A.2.7 /ISAPI/SecurityCP/BasicParam/DetectorCfg

Get or set the detector configuration parameters.

Request URI Definition

Table A-24 POST /ISAPI/SecurityCP/BasicParam/DetectorCfg

Method	POST
Description	Get the detector configuration parameters.
Query	None.
Request	<u><i>XML_ZoneCondList</i></u>
Response	Succeeded: <u><i>XML_DetectorCfg</i></u> Failed: <u><i>XML_ResponseStatus</i></u>

Table A-25 PUT /ISAPI/SecurityCP/BasicParam/DetectorCfg

Method	PUT
Description	Set the detector configuration parameters.
Query	None.
Request	<u><i>XML_DetectorCfg</i></u>
Response	<u><i>XML_ResponseStatus</i></u>

A.2.8 /ISAPI/SecurityCP/BasicParam/DetectorCfg/capabilities

Get configuration capability set of detector parameters.

Request URI Definition

Table A-26 GET /ISAPI/SecurityCP/BasicParam/DetectorCfg/capabilities

Method	GET
Description	Get configuration capability set of detector parameters.
Query	None.
Request	None.
Response	Succeeded: <u><i>XML_Cap_DetectorCfg</i></u>

	Failed: <u><i>XML_ResponseStatus</i></u>
--	------------------------------------------

A.2.9 /ISAPI/SecurityCP/BasicParam/ZoneAssociatedDetectorCfg

Get or set the configuration parameters of zone linked detector.

Request URI Definition

Table A-27 POST /ISAPI/SecurityCP/BasicParam/ZoneAssociatedDetectorCfg

Method	POST
Description	Get the configuration parameters of zone linked detector.
Query	None.
Request	<u><i>XML_ZoneCondList</i></u>
Response	Succeeded: <u><i>XML_ZoneAssociatedDetectorCfg</i></u> Failed: <u><i>XML_ResponseStatus</i></u>

Table A-28 PUT /ISAPI/SecurityCP/BasicParam/ZoneAssociatedDetectorCfg

Method	PUT
Description	Set the parameters of zone linked detector.
Query	None.
Request	<u><i>XML_ZoneAssociatedDetectorCfg</i></u>
Response	<u><i>XML_ResponseStatus</i></u>

A.2.10 /ISAPI/SecurityCP/BasicParam/ZoneAssociatedDetectorCfg/capabilities

Get the capability set of zone linked detector configuration.

Request URI Definition

Table A-29 GET /ISAPI/SecurityCP/BasicParam/ZoneAssociatedDetectorCfg/capabilities

Method	GET
Description	Get the capability set of zone linked detector configuration.
Query	None.

Request	None.
Response	Succeeded: <u><i>XML_Cap_ZoneAssociatedDetectorCfg</i></u> Failed: <u><i>XML_ResponseStatus</i></u>

A.2.11 /ISAPI/SecurityCP/capabilities?format=json

Get the capability of security control panel.

Request URI Definition

Table A-30 GET /ISAPI/SecurityCP/capabilities?format=json

Method	GET
Description	Get the capability of security control panel.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_SecurityCPCap</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.12 /ISAPI/SecurityCP/CheckResult/capabilities?format=json

Get the capability of getting detection results.

Request URI Definition

Table A-31 GET /ISAPI/SecurityCP/CheckResult/capabilities?format=json

Method	GET
Description	Get the capability of getting detection results.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_Cap_CheckResult</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.13 /ISAPI/SecurityCP/CheckResult?format=json

Get the detection results.

Request URI Definition

Table A-32 GET /ISAPI/SecurityCP/CheckResult?format=json

Method	GET
Description	Get the detection results.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_CheckResult</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.14 /ISAPI/SecurityCP/Configuration/accessModule/addType/capabilities?format=json

Get the capability of adding zones, relays, or sirens to access modules.

Request URI Definition

Table A-33 POST /ISAPI/SecurityCP/Configuration/accessModule/addType/capabilities?format=json

Method	POST
Description	Get the capability of adding zones, relays, or sirens to access modules.
Query	format: determine the format of request or response message.
Request	<u><i>JSON_accessModuleType</i></u>
Response	Succeeded: <u><i>JSON_AccessModuleAddTypeCap</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.15 /ISAPI/SecurityCP/Configuration/accessModule/addType?format=json

Add zones, relays, or sirens to access modules.

Request URI Definition

Table A-34 PUT /ISAPI/SecurityCP/Configuration/accessModule/addType?format=json

Method	PUT
Description	Add zones, relays, or sirens to access modules.
Query	format: determine the format of request or response message.
Request	<u><i>JSON_accessModuleAddType</i></u>
Response	Succeeded: <u><i>JSON_accessModuleAddResult</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.16 /ISAPI/SecurityCP/Configuration/ARC/capabilities?format=json

Get alarm receiving center configuration capability.

Request URI Definition

Table A-35 GET /ISAPI/SecurityCP/Configuration/ARC/capabilities?format=json

Method	GET
Description	Get alarm receiving center configuration capability.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_ARCCap</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.17 /ISAPI/SecurityCP/Configuration/ARC/<ID>?format=json

Operations about the configuration of an alarm receiving center.

Request URI Definition

Table A-36 GET /ISAPI/SecurityCP/Configuration/ARC/<ID>?format=json

Method	GET
Description	Get the parameters of an alarm receiving center.
Query	format: determine the format of request or response message.

Request	None.
Response	<u><i>JSON_ARC</i></u>

Table A-37 PUT /ISAPI/SecurityCP/Configuration/ARC/<ID>?format=json

Method	PUT
Description	Set the parameters of an alarm receiving center.
Query	format: determine the format of request or response message.
Request	<u><i>JSON_ARC</i></u>
Response	<u><i>JSON_ResponseStatus</i></u>

Remarks

The <ID> in the request URI refers to the alarm receiving center ID.

A.2.18 /ISAPI/SecurityCP/Configuration/ARC/manualTest/capabilities?format=json

Get the capability of ARC manual test.

Request URI Definition**Table A-38 GET /ISAPI/SecurityCP/Configuration/ARC/manualTest/capabilities?format=json**

Method	GET
Description	Get the capability of ARC manual test.
Query	format: determine the format of request or response message.
Request	None
Response	Succeeded: <u><i>JSON_ARCManualTestCap</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.19 /ISAPI/SecurityCP/Configuration/ARC/manualTest/status?format=json

Get the manual test status of a single ARC.

Request URI Definition

Table A-39 POST /ISAPI/SecurityCP/Configuration/ARC/manualTest/status?format=json

Method	POST
Description	Get the manual test status of a single ARC.
Query	format : determine the format of request or response message.
Request	<u><i>JSON_ARCManualTestID</i></u>
Response	<u><i>JSON_ARCManualTestStatus</i></u>

A.2.20 /ISAPI/SecurityCP/Configuration/ARC/manualTest?format=json

Set the parameters of ARC manual test.

Request URI Definition

Table A-40 PUT /ISAPI/SecurityCP/Configuration/ARC/manualTest?format=json

Method	PUT
Description	Set the parameters of ARC manual test.
Query	format : determine the format of request or response message.
Request	<u><i>JSON_ARCManualTest</i></u>
Response	<u><i>JSON_ResponseStatus</i></u>

A.2.21 /ISAPI/SecurityCP/Configuration/ARC?format=json

Get the parameters of all alarm receiving centers.

Request URI Definition

Table A-41 GET /ISAPI/SecurityCP/Configuration/ARC?format=json

Method	GET
Description	Get the parameters of all alarm receiving centers.
Query	format : determine the format of request or response message.

Request	None.
Response	<u><i>JSON_List_ARC</i></u>

A.2.22 /ISAPI/SecurityCP/Configuration/capabilities?format=json

Get the configuration capability of security control panel.

Request URI Definition

Table A-42 GET /ISAPI/SecurityCP/Configuration/capabilities?format=json

Method	GET
Description	Get the configuration capability of security control panel.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_HostConfigCap</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.23 /ISAPI/SecurityCP/Configuration/card/capabilities?format=json

Get card configuration capability.

Request URI Definition

Table A-43 GET /ISAPI/SecurityCP/Configuration/card/capabilities?format=json

Method	GET
Description	Get card configuration capability.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_CardCap</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.24 /ISAPI/SecurityCP/Configuration/card/currentAdd?format=json

Get the parameters of the currently added card in synchronous mode.

Request URI Definition

Table A-44 GET /ISAPI/SecurityCP/Configuration/card/currentAdd?format=json

Method	GET
Description	Get the parameters of the currently added card in synchronous mode.
Query	format : determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_Card</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

Remarks

The parameters will be returned after they are obtained. The timeout is 20 seconds.

A.2.25 /ISAPI/SecurityCP/Configuration/card/currentAddAsyn?format=json

Get the parameters of the currently added card in asynchronous mode.

Request URI Definition

Table A-45 GET /ISAPI/SecurityCP/Configuration/card/currentAddAsyn?format=json

Method	GET
Description	Get the parameters of the currently added card in asynchronous mode.
Query	format : determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_Card</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

Remarks

- Before calling this URI, you should call */ISAPI/SecurityCP/Configuration/card/mode?format=json* by PUT method and set the **mode** to "enter" to enable the asynchronous card adding mode for the device.
- The result will be returned immediately after calling this URI. If **status** in the response message is "processing", you should continue calling this URI until **status** is "success" and the card parameters are returned. If **status** is "failed", you should stop calling this URI recursively.

A.2.26 /ISAPI/SecurityCP/Configuration/card/mode/capabilities?format=json

Get the capability of controlling asynchronous card parameters adding mode.

Request URI Definition**Table A-46 GET /ISAPI/SecurityCP/Configuration/card/mode/capabilities?format=json**

Method	GET
Description	Get the capability of controlling asynchronous card parameters adding mode.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_Cap_CardMode</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.27 /ISAPI/SecurityCP/Configuration/card/mode?format=json

Control the asynchronous card parameters adding mode.

Request URI Definition**Table A-47 PUT /ISAPI/SecurityCP/Configuration/card/mode?format=json**

Method	PUT
Description	Control the asynchronous card parameters adding mode.
Query	format: determine the format of request or response message.
Request	<u><i>JSON_CardMode</i></u>
Response	<u><i>JSON_ResponseStatus</i></u>

A.2.28 /ISAPI/SecurityCP/Configuration/card/<ID>?format=json

Operations about the configuration of a card.

Request URI Definition

Table A-48 GET /ISAPI/SecurityCP/Configuration/card/<ID>?format=json

Method	GET
Description	Get the parameters of a card.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <i><u>JSON_Card</u></i> Failed: <i><u>JSON_ResponseStatus</u></i>

Table A-49 PUT /ISAPI/SecurityCP/Configuration/card/<ID>?format=json

Method	PUT
Description	Set parameters for a card.
Query	format: determine the format of request or response message.
Request	<i><u>JSON_Card</u></i>
Response	<i><u>JSON_ResponseStatus</u></i>

Table A-50 DELETE /ISAPI/SecurityCP/Configuration/card/<ID>?format=json

Method	DELETE
Description	Delete a card.
Query	format: determine the format of request or response message.
Request	None.
Response	<i><u>JSON_ResponseStatus</u></i>

A.2.29 /ISAPI/SecurityCP/Configuration/card?format=json

Get parameters of all cards.

Request URI Definition

Table A-51 GET /ISAPI/SecurityCP/Configuration/card?format=json

Method	GET
Description	Get parameters of all cards.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON List Card</i></u> Failed: <u><i>JSON ResponseStatus</i></u>

A.2.30 /ISAPI/SecurityCP/Configuration/cardReader/<ID>?format=json

Set the parameters of a card reader.

Request URI Definition

Table A-52 PUT /ISAPI/SecurityCP/Configuration/cardReader/<ID>?format=json

Method	PUT
Description	Set the parameters of a card reader.
Query	format: determine the format of request or response message.
Request	<u><i>JSON CardReader</i></u>
Response	<u><i>JSON ResponseStatus</i></u>

A.2.31 /ISAPI/SecurityCP/Configuration/cardReader/capabilities?format=json

Get the card reader configuration capability.

Request URI Definition

Table A-53 GET /ISAPI/SecurityCP/Configuration/cardReader/capabilities?format=json

Method	GET
Description	Get the card reader configuration capability.
Query	format: determine the format of request or response message.

Request	None.
Response	Succeeded: <u><i>JSON_CardReaderCap</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.32 /ISAPI/SecurityCP/Configuration/cardReader/currentAddAsyn?format=json

Get the parameters of the currently added card reader in asynchronous mode.

Request URI Definition

Table A-54 GET /ISAPI/SecurityCP/Configuration/cardReader/currentAddAsyn?format=json

Method	GET
Description	Get the parameters of the currently added card reader in asynchronous mode.
Query	format : determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_CardReader</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

Remarks

- Before calling this URI, you should call */ISAPI/SecurityCP/Configuration/cardReader/mode?format=json* by PUT method and set the **mode** to "enter" to enable the asynchronous card reader adding mode for the device.
- The result will be returned immediately after calling this URI. If **status** in the response message is "processing", you should continue calling this URI until **status** is "success" and the card reader parameters are returned. If **status** is "failed", you should stop calling this URI recursively.

A.2.33 /ISAPI/SecurityCP/Configuration/cardReader/mode/capabilities?format=json

Get the capability of controlling asynchronous mode of adding card reader parameters.

Request URI Definition

Table A-55 GET /ISAPI/SecurityCP/Configuration/cardReader/mode/capabilities?format=json

Method	GET
Description	Get the capability of controlling asynchronous mode of adding card reader parameters.
Query	format : determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_Cap_CardReaderMode</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.34 /ISAPI/SecurityCP/Configuration/cardReader/mode?format=json

Control the asynchronous mode of adding card reader parameters.

Request URI Definition

Table A-56 PUT /ISAPI/SecurityCP/Configuration/cardReader/mode?format=json

Method	PUT
Description	Control the asynchronous mode of adding card reader parameters.
Query	format : determine the format of request or response message.
Request	<u><i>JSON_CardReaderMode</i></u>
Response	<u><i>JSON_ResponseStatus</i></u>

A.2.35 /ISAPI/SecurityCP/Configuration/cardReader?format=json

Get configuration parameters of all card readers.

Request URI Definition

Table A-57 GET /ISAPI/SecurityCP/Configuration/cardReader?format=json

Method	GET
Description	Get configuration parameters of all card readers.
Query	format : determine the format of request or response message.

Request	None.
Response	Succeeded: <u><i>JSON_List_CardReader</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.36 /ISAPI/SecurityCP/Configuration/curtainInfraredDetector/capabilities?format=json

Get the configuration capability of the curtain PIR (Passive Infrared) detector.

Request URI Definition

Table A-58 GET /ISAPI/SecurityCP/Configuration/curtainInfraredDetector/capabilities?format=json

Method	GET
Description	Get the configuration capability of the curtain PIR (Passive Infrared) detector.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_CurtainInfraredDetectorCap</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.37 /ISAPI/SecurityCP/Configuration/curtainInfraredDetector/zone/<ID>?format=json

Get or set the parameters of the curtain PIR (Passive Infrared) detector of a specific zone.

Request URI Definition

Table A-59 GET /ISAPI/SecurityCP/Configuration/curtainInfraredDetector/zone/<ID>?format=json

Method	GET
Description	Get the parameters of the curtain PIR (Passive Infrared) detector of a specific zone.
Query	format: determine the format of request or response message.

Request	None.
Response	Succeeded: <u><i>JSON_CurtainInfraredDetector</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

Table A-60 PUT /ISAPI/SecurityCP/Configuration/curtainInfraredDetector/zone/<ID>?format=json

Method	PUT
Description	Set the parameters of the curtain PIR (Passive Infrared) detector of a specific zone.
Query	format: determine the format of request or response message.
Request	<u><i>JSON_CurtainInfraredDetector</i></u>
Response	<u><i>JSON_ResponseStatus</i></u>

Remarks

The <ID> in the request URI refers to the zone No. which starts from 0.

A.2.38 /ISAPI/SecurityCP/Configuration/curtainInfraredDetector?format=json

Get the parameters of all curtain PIR (Passive Infrared) detectors.

Request URI Definition

Table A-61 GET /ISAPI/SecurityCP/Configuration/curtainInfraredDetector?format=json

Method	GET
Description	Get the parameters of all curtain PIR (Passive Infrared) detectors.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_List_CurtainInfraredDetector</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.39 /ISAPI/SecurityCP/Configuration/deviceTime/capabilities?format=json

Get the configuration capability of the security control panel timer.

Request URI Definition

Table A-62 GET /ISAPI/SecurityCP/Configuration/deviceTime/capabilities?format=json

Method	GET
Description	Get the configuration capability of the security control panel timer.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <i><u>JSON DeviceTimeCap</u></i> Failed: <i><u>JSON ResponseStatus</u></i>

A.2.40 /ISAPI/SecurityCP/Configuration/deviceTime?format=json

Operations about the configuration of the security control panel timer.

Request URI Definition

Table A-63 GET /ISAPI/SecurityCP/Configuration/deviceTime?format=json

Method	GET
Description	Get the parameters of the security control panel timer.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <i><u>JSON DeviceTime</u></i> Failed: <i><u>JSON ResponseStatus</u></i>

Table A-64 PUT /ISAPI/SecurityCP/Configuration/deviceTime?format=json

Method	PUT
Description	Set the parameters of the security control panel timer.
Query	format: determine the format of request or response message.
Request	<i><u>JSON DeviceTime</u></i>
Response	<i><u>JSON ResponseStatus</u></i>

A.2.41 /ISAPI/SecurityCP/Configuration/eventRecord/channels/<ID>/capabilities?format=json

Get the capability about the configuration of recording based on event.

Request URI Definition

Table A-65 GET /ISAPI/SecurityCP/Configuration/eventRecord/channels/<ID>/capabilities?format=json

Method	GET
Description	Get the capability about the configuration of recording based on event.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON EventRecordCap</i></u> Failed: <u><i>JSON ResponseStatus</i></u>

A.2.42 /ISAPI/SecurityCP/Configuration/eventRecord/channels/<ID>?format=json

Operations about the configuration of recording based on event.

Request URI Definition

Table A-66 GET /ISAPI/SecurityCP/Configuration/eventRecord/channels/<ID>?format=json

Method	GET
Description	Get the configuration parameters of recording based on event.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON EventRecord</i></u> Failed: <u><i>JSON ResponseStatus</i></u>

Table A-67 PUT /ISAPI/SecurityCP/Configuration/eventRecord/channels/<ID>?format=json

Method	PUT
Description	Set the parameters of recording based on event.

Query	format: determine the format of request or response message.
Request	<u><i>JSON_EventRecord</i></u>
Response	<u><i>JSON_ResponseStatus</i></u>

A.2.43 /ISAPI/SecurityCP/Configuration/extensionModule/<ID>?format=json

Set the parameters of a specific extension module.

Request URI Definition

Table A-68 PUT /ISAPI/SecurityCP/Configuration/extensionModule/<ID>?format=json

Method	PUT
Description	Set the parameters of a specific extension module.
Query	format: determine the format of request or response message.
Request	<u><i>JSON_ExtensionModule</i></u>
Response	<u><i>JSON_ResponseStatus</i></u>

Remarks

The <ID> in the request URI refers to the extension module No.

A.2.44 /ISAPI/SecurityCP/Configuration/extensionModule/capabilities?format=json

Get the configuration capability of the extension module.

Request URI Definition

Table A-69 GET /ISAPI/SecurityCP/Configuration/extensionModule/capabilities?format=json

Method	GET
Description	Get the configuration capability of the extension module.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_ExtensionModuleCap</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.45 /ISAPI/SecurityCP/Configuration/extensionModule?format=json

Get the configuration of all extension modules.

Request URI Definition

Table A-70 GET /ISAPI/SecurityCP/Configuration/extensionModule?format=json

Method	GET
Description	Get the configuration of all extension modules.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_List_ExtensionModule</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.46 /ISAPI/SecurityCP/Configuration/faultCheckCfg/capabilities?format=json

Get the configuration capability of fault detection.

Request URI Definition

Table A-71 GET /ISAPI/SecurityCP/Configuration/faultCheckCfg/capabilities?format=json

Method	GET
Description	Get the configuration capability of fault detection.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_FaultCheckParameterCap</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.47 /ISAPI/SecurityCP/Configuration/faultCheckCfg?format=json

Operations about the fault detection parameters.

Request URI Definition

Table A-72 GET /ISAPI/SecurityCP/Configuration/faultCheckCfg?format=json

Method	GET
Description	Get the fault detection parameters.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_FaultCheckParameter</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

Table A-73 PUT /ISAPI/SecurityCP/Configuration/faultCheckCfg?format=json

Method	PUT
Description	Set the fault detection parameters.
Query	format: determine the format of request or response message.
Request	<u><i>JSON_FaultCheckParameter</i></u>
Response	<u><i>JSON_ResponseStatus</i></u>

A.2.48 /ISAPI/SecurityCP/Configuration/glassBreakDetector/capabilities?format=json

Get the configuration capability of the composite PIR (Passive Infrared) glass-break detector.

Request URI Definition

Table A-74 GET /ISAPI/SecurityCP/Configuration/glassBreakDetector/capabilities?format=json

Method	GET
Description	Get the configuration capability of the composite PIR (Passive Infrared) glass-break detector.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_GlassBreakDetectorCap</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.49 /ISAPI/SecurityCP/Configuration/glassBreakDetector?format=json

Get parameters of all composite PIR (Passive Infrared) glass-break detectors.

Request URI Definition**Table A-75 GET /ISAPI/SecurityCP/Configuration/glassBreakDetector?format=json**

Method	GET
Description	Get parameters of all composite PIR (Passive Infrared) glass-break detectors.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON List GlassBreakDetector</i></u> Failed: <u><i>JSON ResponseStatus</i></u>

A.2.50 /ISAPI/SecurityCP/Configuration/glassBreakDetector/zone/<ID>?format=json

Get or set parameters of the composite PIR (Passive Infrared) glass-break detector of a specific zone.

Request URI Definition**Table A-76 GET /ISAPI/SecurityCP/Configuration/glassBreakDetector/zone/<ID>?format=json**

Method	GET
Description	Get parameters of the composite PIR (Passive Infrared) glass-break detector of a specific zone.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON GlassBreakDetector</i></u> Failed: <u><i>JSON ResponseStatus</i></u>

Table A-77 PUT /ISAPI/SecurityCP/Configuration/glassBreakDetector/zone/<ID>?format=json

Method	PUT
Description	Set parameters of the composite PIR (Passive Infrared) glass-break detector of a specific zone.
Query	format: determine the format of request or response message.
Request	<i><u>JSON_GlassBreakDetector</u></i>
Response	<i><u>JSON_ResponseStatus</u></i>

Remarks

The <ID> in the request URI refers to the zone No. which starts from 0.

A.2.51 /ISAPI/SecurityCP/Configuration/indoorDualTechnologyDetector/capabilities?format=json

Get the configuration capability of the indoor dual-technology detector.

Request URI Definition
Table A-78 GET /ISAPI/SecurityCP/Configuration/indoorDualTechnologyDetector/capabilities?format=json

Method	GET
Description	Get the configuration capability of the indoor dual-technology detector.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <i><u>JSON_IndoorDualTechnologyDetectorCap</u></i> Failed: <i><u>JSON_ResponseStatus</u></i>

A.2.52 /ISAPI/SecurityCP/Configuration/indoorDualTechnologyDetector/zone/<ID>?format=json

Get or set the parameters of the indoor dual-technology detector of a specific zone.

Request URI Definition

Table A-79 GET /ISAPI/SecurityCP/Configuration/indoorDualTechnologyDetector/zone/<ID>?format=json

Method	GET
Description	Get the parameters of the indoor dual-technology detector of a specific zone.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON IndoorDualTechnologyDetector</i></u> Failed: <u><i>JSON ResponseStatus</i></u>

Table A-80 PUT /ISAPI/SecurityCP/Configuration/indoorDualTechnologyDetector/zone/<ID>?format=json

Method	PUT
Description	Set the parameters of the indoor dual-technology detector of a specific zone.
Query	format: determine the format of request or response message.
Request	<u><i>JSON IndoorDualTechnologyDetector</i></u>
Response	<u><i>JSON ResponseStatus</i></u>

Remarks

The <ID> in the request URI refers to the zone No. which starts from 0.

A.2.53 /ISAPI/SecurityCP/Configuration/indoorDualTechnologyDetector?format=json

Get the parameters of all indoor dual-technology detectors.

Request URI Definition

Table A-81 GET /ISAPI/SecurityCP/Configuration/indoorDualTechnologyDetector?format=json

Method	GET
Description	Get the parameters of all indoor dual-technology detectors.

Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_List_IndoorDualTechnologyDetector</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.54 /ISAPI/SecurityCP/Configuration/keypad/<ID>?format=json

Set the parameters of a keypad.

Request URI Definition

Table A-82 PUT /ISAPI/SecurityCP/Configuration/keypad/<ID>?format=json

Method	PUT
Description	Set the parameters of a keypad.
Query	format: determine the format of request or response message.
Request	<u><i>JSON_Keypad</i></u>
Response	<u><i>JSON_ResponseStatus</i></u>

A.2.55 /ISAPI/SecurityCP/Configuration/keypad/capabilities?format=json

Get the keypad configuration capability.

Request URI Definition

Table A-83 GET /ISAPI/SecurityCP/Configuration/keypad/capabilities?format=json

Method	GET
Description	Get the keypad configuration capability.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_KeypadCap</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.56 /ISAPI/SecurityCP/Configuration/keypad?format=json

Get configuration parameters of all keypads.

Request URI Definition

Table A-84 GET /ISAPI/SecurityCP/Configuration/keypad?format=json

Method	GET
Description	Get configuration parameters of all keypads.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_List_Keypad</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.57 /ISAPI/SecurityCP/Configuration/keypadAddList/capabilities?format=json

Get the capability of getting the list of keypads that can be added.

Request URI Definition

Table A-85 GET /ISAPI/SecurityCP/Configuration/keypadAddList/capabilities?format=json

Method	GET
Description	Get the capability of getting the list of keypads that can be added.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_KeypadAddListCap</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.58 /ISAPI/SecurityCP/Configuration/keypadAddList?format=json

Get the list of keypads that can be added.

Request URI Definition

Table A-86 GET /ISAPI/SecurityCP/Configuration/keypadAddList?format=json

Method	GET
Description	Get the list of keypads that can be added.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <i><u>JSON KeypadAddList</u></i> Failed: <i><u>JSON ResponseStatus</u></i>

A.2.59 /ISAPI/SecurityCP/Configuration/keypadFaultProcessCfg/<ID>?format=json

Set the linkage parameters of a specific keypad for the system fault.

Request URI Definition

Table A-87 PUT /ISAPI/SecurityCP/Configuration/keypadFaultProcessCfg/<ID>?format=json

Method	PUT
Description	Set the linkage parameters of a specific keypad for the system fault.
Query	format: determine the format of request or response message.
Request	<i><u>JSON KeypadFaultProcessCfg</u></i>
Response	<i><u>JSON ResponseStatus</u></i>

Remarks

The <ID> in the request URI refers to the keypad address.

A.2.60 /ISAPI/SecurityCP/Configuration/keypadFaultProcessCfg/capabilities?format=json

Get the configuration capability of keypad linkage parameters of the system fault.

Request URI Definition

Table A-88 GET /ISAPI/SecurityCP/Configuration/keypadFaultProcessCfg/capabilities?format=json

Method	GET
Description	Get the configuration capability of keypad linkage parameters of the system fault.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON KeypadFaultProcessCfgCap</i></u> Failed: <u><i>JSON ResponseStatus</i></u>

A.2.61 /ISAPI/SecurityCP/Configuration/keypadFaultProcessCfg?format=json

Get all keypads' linkage parameters of the system fault.

Request URI Definition

Table A-89 GET /ISAPI/SecurityCP/Configuration/keypadFaultProcessCfg?format=json

Method	GET
Description	Get all keypads' linkage parameters of the system fault.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON List KeypadFaultProcessCfg</i></u> Failed: <u><i>JSON ResponseStatus</i></u>

A.2.62 /ISAPI/SecurityCP/Configuration/magneticContact/capabilities?format=json

Get the configuration capability of the composite magnetic contact detector.

Request URI Definition

Table A-90 GET /ISAPI/SecurityCP/Configuration/magneticContact/capabilities?format=json

Method	GET
Description	Get the configuration capability of the composite magnetic contact detector.
Query	format : determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_MagneticContactCap</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.63 /ISAPI/SecurityCP/Configuration/magneticContact/zone/<ID>?format=json

Get or set the parameters of the composite magnetic contact detector of a specific zone.

Request URI Definition

Table A-91 GET /ISAPI/SecurityCP/Configuration/magneticContact/zone/<ID>?format=json

Method	GET
Description	Get the parameters of the composite magnetic contact detector of a specific zone.
Query	format : determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_MagneticContact</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

Table A-92 PUT /ISAPI/SecurityCP/Configuration/magneticContact/zone/<ID>?format=json

Method	PUT
Description	Set the parameters of the composite magnetic contact detector of a specific zone.
Query	format : determine the format of request or response message.
Request	<u><i>JSON_MagneticContact</i></u>
Response	<u><i>JSON_ResponseStatus</i></u>

Remarks

The <ID> in the request URI refers to the zone No. which starts from 0.

A.2.64 /ISAPI/SecurityCP/Configuration/magneticContact?format=json

Get parameters of all composite magnetic contact detectors.

Request URI Definition

Table A-93 GET /ISAPI/SecurityCP/Configuration/magneticContact?format=json

Method	GET
Description	Get parameters of all composite magnetic contact detectors.
Query	format : determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_List_MagneticContact</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.65 /ISAPI/SecurityCP/Configuration/messageSendARC/capabilities?format=json

Get the configuration capability of alarm receiving center notification in listening mode.

Request URI Definition

Table A-94 GET /ISAPI/SecurityCP/Configuration/messageSendARC/capabilities?format=json

Method	GET
Description	Get the configuration capability of alarm receiving center notification in listening mode.
Query	format : determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_SendARCCap</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.66 /ISAPI/SecurityCP/Configuration/messageSendARCList?format=json

Get notification parameters of all alarm receiving centers.

Request URI Definition**Table A-95 GET /ISAPI/SecurityCP/Configuration/messageSendARCList?format=json**

Method	GET
Description	Get notification parameters of all alarm receiving centers.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_SendARCList</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.67 /ISAPI/SecurityCP/Configuration/messageSendARC?format=json

Operations about the configurations of alarm receiving center notification in listening mode.

Request URI Definition**Table A-96 GET /ISAPI/SecurityCP/Configuration/messageSendARC?format=json**

Method	GET
Description	Get the configuration parameters of alarm receiving center notification in listening mode.
Query	id: alarm receiving center No. If this query parameter does not exist, the default value is 1. format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_SendARC</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

Table A-97 PUT /ISAPI/SecurityCP/Configuration/messageSendARC?format=json

Method	PUT
Description	Set the parameters of alarm receiving center notification in listening mode.

Query	format: determine the format of request or response message. id: alarm receiving center No. If this query parameter does not exist, the default value is 1.
Request	<u><i>JSON SendARC</i></u>
Response	<u><i>JSON ResponseStatus</i></u>

A.2.68 /ISAPI/SecurityCP/Configuration/messageSendCloud/capabilities?format=json

Get the capability of Hik-Connect notification configuration.

Request URI Definition

Table A-98 GET /ISAPI/SecurityCP/Configuration/messageSendCloud/capabilities?format=json

Method	GET
Description	Get the capability of Hik-Connect notification configuration.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON CloudCap</i></u> Failed: <u><i>JSON ResponseStatus</i></u>

A.2.69 /ISAPI/SecurityCP/Configuration/messageSendCloud?format=json

Operations about Hik-Connect notification configurations.

Request URI Definition

Table A-99 GET /ISAPI/SecurityCP/Configuration/messageSendCloud?format=json

Method	GET
Description	Get the configuration parameters of Hik-Connect notification.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON Cloud</i></u> Failed: <u><i>JSON ResponseStatus</i></u>

Table A-100 PUT /ISAPI/SecurityCP/Configuration/messageSendCloud?format=json

Method	PUT
Description	Set the parameters of Hik-Connect notification.
Query	format: determine the format of request or response message.
Request	<u><i>JSON_Cloud</i></u>
Response	<u><i>JSON_ResponseStatus</i></u>

A.2.70 /ISAPI/SecurityCP/Configuration/messageSendDirect/capabilities?format=json

Get configuration capability of alarm receiving center notification in arming mode.

Request URI Definition

Table A-101 GET /ISAPI/SecurityCP/Configuration/messageSendDirect/capabilities?format=json

Method	GET
Description	Get configuration capability of alarm receiving center notification in arming mode.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_DirectCap</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.71 /ISAPI/SecurityCP/Configuration/messageSendDirect?format=json

Operations about the configurations of alarm receiving center notification in arming mode.

Request URI Definition

Table A-102 GET /ISAPI/SecurityCP/Configuration/messageSendDirect?format=json

Method	GET
Description	Get the configuration parameters of alarm receiving center notification in arming mode.

Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_Direct</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

Table A-103 PUT /ISAPI/SecurityCP/Configuration/messageSendDirect?format=json

Method	PUT	
Description	Set the parameters of alarm receiving center notification in arming mode.	
Query	format: determine the format of request or response message.	
Request	<u><i>JSON_Direct</i></u>	
Response	<u><i>JSON_ResponseStatus</i></u>	

A.2.72 /ISAPI/SecurityCP/Configuration/messageSendMail/<ID>?format=json

Set the parameters of one email notification.

Request URI Definition

Table A-104 PUT /ISAPI/SecurityCP/Configuration/messageSendMail/<ID>?format=json

Method	PUT
Description	Set the parameters of one email notification.
Query	format: determine the format of request or response message.
Request	<u><i>JSON_Mail</i></u>
Response	<u><i>JSON_ResponseStatus</i></u>

A.2.73 /ISAPI/SecurityCP/Configuration/messageSendMail?format=json

Get the email notification parameters.

Request URI Definition

Table A-105 GET /ISAPI/SecurityCP/Configuration/messageSendMail?format=json

Method	GET
Description	Get the email notification parameters.
Query	format: determine the format of request or response message.
Request	None.
Response	<i><u>JSON List Mail</u></i>

A.2.74 /ISAPI/SecurityCP/Configuration/messageSendMail/capabilities?format=json

Get the email notification configuration capability.

Request URI Definition

Table A-106 GET /ISAPI/SecurityCP/Configuration/messageSendMail/capabilities?format=json

Method	GET
Description	Get the email notification configuration capability.
Query	format: determine the format of request or response message.
Request	None.
Response	<i><u>JSON MailCap</u></i>

A.2.75 /ISAPI/SecurityCP/Configuration/messageSendPhone/<ID>?format=json

Set the parameters of one phone notification.

Request URI Definition

Table A-107 PUT /ISAPI/SecurityCP/Configuration/messageSendPhone/<ID>?format=json

Method	PUT
Description	Set the parameters of one phone notification.
Query	format: determine the format of request or response message.

Request	<u><i>JSON_Phone</i></u>
Response	<u><i>JSON_ResponseStatus</i></u>

A.2.76 /ISAPI/SecurityCP/Configuration/messageSendPhone/capabilities?format=json

Get the phone notification configuration capability.

Request URI Definition

Table A-108 GET /ISAPI/SecurityCP/Configuration/messageSendPhone/capabilities?format=json

Method	GET
Description	Get the phone notification configuration capability.
Query	format: determine the format of request or response message.
Request	None.
Response	<u><i>JSON_PhoneCap</i></u>

A.2.77 /ISAPI/SecurityCP/Configuration/messageSendPhone?format=json

Get the phone notification parameters.

Request URI Definition

Table A-109 GET /ISAPI/SecurityCP/Configuration/messageSendPhone?format=json

Method	GET
Description	Get the phone notification parameters.
Query	format: determine the format of request or response message.
Request	None.
Response	<u><i>JSON_List_Phone</i></u>

A.2.78 /ISAPI/SecurityCP/Configuration/messageSendPhoneAnvanced/<ID>?format=json

Set advanced notification parameters of a specific phone number.

Request URI Definition

Table A-110 PUT /ISAPI/SecurityCP/Configuration/messageSendPhoneAnvanded/<ID>?format=json

Method	PUT
Description	Set advanced notification parameters of a specific phone number.
Query	format: determine the format of request or response message.
Request	<u><i>JSON_PhoneAnvanded</i></u>
Response	<u><i>JSON_ResponseStatus</i></u>

Remarks

- After the phone notification is enabled, you can configure parameters to filter duplicate zone alarms in the configured time interval, configure phone call times, and configure event uploading parameters.
- After the SMS notification is enabled, you can configure event uploading parameters.

A.2.79 /ISAPI/SecurityCP/Configuration/messageSendPhoneAnvanded/capabilities?format=json

Get the advanced configuration capability of the phone notification.

Request URI Definition

Table A-111 GET /ISAPI/SecurityCP/Configuration/messageSendPhoneAnvanded/capabilities?format=json

Method	GET
Description	Get the advanced configuration capability of the phone notification.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_PhoneAnvandedCap</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

Remarks

- After the phone notification is enabled, you can configure parameters to filter duplicate zone alarms in the configured time interval, configure phone call times, and configure event uploading parameters.
- After the SMS notification is enabled, you can configure event uploading parameters.

A.2.80 /ISAPI/SecurityCP/Configuration/messageSendPhoneAnvanced?format=json

Get advanced notification parameters of all phone numbers.

Request URI Definition

Table A-112 GET /ISAPI/SecurityCP/Configuration/messageSendPhoneAnvanced?format=json

Method	GET
Description	Get advanced notification parameters of all phone numbers.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON List PhoneAnvanced</i></u> Failed: <u><i>JSON ResponseStatus</i></u>

Remarks

- After the phone notification is enabled, you can configure parameters to filter duplicate zone alarms in the configured time interval, configure phone call times, and configure event uploading parameters.
- After the SMS notification is enabled, you can configure event uploading parameters.

A.2.81 /ISAPI/SecurityCP/Configuration/muteVoicePlanCFG/capabilities?format=json

Get the muting schedule configuration capability.

Request URI Definition

Table A-113 GET /ISAPI/SecurityCP/Configuration/muteVoicePlanCFG/capabilities?format=json

Method	GET
Description	Get the muting schedule configuration capability.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <i><u>JSON_Cap_MuteVoicePlanCFG</u></i> Failed: <i><u>JSON_ResponseStatus</u></i>

A.2.82 /ISAPI/SecurityCP/Configuration/muteVoicePlanCFG?format=json

Get or set the muting schedule parameters.

Request URI Definition

Table A-114 GET /ISAPI/SecurityCP/Configuration/muteVoicePlanCFG?format=json

Method	GET
Description	Get the muting schedule parameters.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <i><u>JSON_MuteVoicePlanCFG</u></i> Failed: <i><u>JSON_ResponseStatus</u></i>

Table A-115 PUT /ISAPI/SecurityCP/Configuration/muteVoicePlanCFG?format=json

Method	PUT
Description	Set the muting schedule parameters.
Query	format: determine the format of request or response message.
Request	<i><u>JSON_MuteVoicePlanCFG</u></i>
Response	<i><u>JSON_ResponseStatus</u></i>

A.2.83 /ISAPI/SecurityCP/Configuration/notRelateZones/capabilities?format=json

Get the capability of getting unlinked zones.

Request URI Definition**Table A-116 GET /ISAPI/SecurityCP/Configuration/notRelateZones/capabilities?format=json**

Method	GET
Description	Get the capability of getting unlinked zones.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_NotRelateZonesCap</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.84 /ISAPI/SecurityCP/Configuration/notRelateZones?format=json

Get the unlinked zones.

Request URI Definition**Table A-117 GET /ISAPI/SecurityCP/Configuration/notRelateZones?format=json**

Method	GET
Description	Get the unlinked zones.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_NotRelateZones</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.85 /ISAPI/SecurityCP/Configuration/outputModules/<ID>?format=json

Set parameters for an output module (physical relay).

Request URI Definition

Table A-118 PUT /ISAPI/SecurityCP/Configuration/outputModules/<ID>?format=json

Method	PUT
Description	Set parameters for an output module (physical relay).
Query	format : determine the format of request or response message.
Request	<i><u>JSON_OutputModule</u></i>
Response	<i><u>JSON_ResponseStatus</u></i>

Remarks

The <ID> in the request URI refers to the output module (physical relay) No., and it starts from 1.

A.2.86 /ISAPI/SecurityCP/Configuration/outputModules/capabilities?format=json

Get output module (physical relay) configuration capability.

Request URI Definition

Table A-119 GET /ISAPI/SecurityCP/Configuration/outputModules/capabilities?format=json

Method	GET
Description	Get output module (physical relay) configuration capability.
Query	format : determine the format of request or response message.
Request	None.
Response	Succeeded: <i><u>JSON_OutputModuleCap</u></i> Failed: <i><u>JSON_ResponseStatus</u></i>

A.2.87 /ISAPI/SecurityCP/Configuration/outputModules?format=json

Get all output modules' (physical relays') parameters.

Request URI Definition

Table A-120 GET /ISAPI/SecurityCP/Configuration/outputModules?format=json

Method	GET
Description	Get all output modules' (physical relays') parameters.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <i><u>JSON List OutputModule</u></i> Failed: <i><u>JSON ResponseStatus</u></i>

A.2.88 /ISAPI/SecurityCP/Configuration/outputs/<ID>?format=json

Set parameters for a logical relay.

Request URI Definition

Table A-121 PUT /ISAPI/SecurityCP/Configuration/outputs/<ID>?format=json

Method	PUT
Description	Set parameters for a logical relay.
Query	format: determine the format of request or response message.
Request	<i><u>JSON Output</u></i>
Response	<i><u>JSON ResponseStatus</u></i>

Remarks

The <ID> in the request URI refers to the logical relay No., and it starts from 0.

A.2.89 /ISAPI/SecurityCP/Configuration/outputs/capabilities?format=json

Get the configuration capability of logical relay.

Request URI Definition

Table A-122 GET /ISAPI/SecurityCP/Configuration/outputs/capabilities?format=json

Method	GET
Description	Get the configuration capability of logical relay.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON OutputCap</i></u> Failed: <u><i>JSON ResponseStatus</i></u>

Remarks

The logical relay should be linked to a physical relay to control the physical relay. The physical relay is the channel of the output module.

A.2.90 /ISAPI/SecurityCP/Configuration/outputs?format=json

Get all logical relays' parameters or logical relays' parameters by specific conditions.

Request URI Definition

Table A-123 GET /ISAPI/SecurityCP/Configuration/outputs?format=json

Method	GET
Description	Get all logical relays' parameters.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON List Output</i></u> Failed: <u><i>JSON ResponseStatus</i></u>

Table A-124 POST /ISAPI/SecurityCP/Configuration/outputs?format=json

Method	POST
Description	Get logical relays' parameters by specific conditions.
Query	format: determine the format of request or response message.

Request	<u><i>JSON_OutputCond</i></u>
Response	<u><i>JSON_OutputSearch_Config</i></u>

A.2.91 /ISAPI/SecurityCP/Configuration/outputsModule/<ID>?format=json

Set relay's linkage configuration parameters (when the relay is closed/open).

Request URI Definition

Table A-125 PUT /ISAPI/SecurityCP/Configuration/outputsModule/<ID>?format=json

Method	PUT
Description	Set relay's linkage configuration parameters (when the relay is closed/open).
Query	format: determine the format of request or response message.
Request	<u><i>JSON_OutPutsModule</i></u>
Response	<u><i>JSON_ResponseStatus</i></u>

Remarks

The <ID> in the request URI refers to relay ID.

A.2.92 /ISAPI/SecurityCP/Configuration/outputsModule/capabilities?format=json

Get the relay's linkage configuration capability (when the relay is closed/open).

Request URI Definition

Table A-126 GET /ISAPI/SecurityCP/Configuration/outputsModule/capabilities?format=json

Method	GET
Description	Get the relay's linkage configuration capability (when the relay is closed/open).
Query	format: determine the format of request or response message.
Request	None
Response	Succeeded: <u><i>JSON_OutPutsModuleCap</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.93 /ISAPI/SecurityCP/Configuration/outputsModule?format=json

Get linkage configuration parameters (when the relay is closed/open) of all relays or search for linkage configuration parameters by condition.

Request URI Definition

Table A-127 GET /ISAPI/SecurityCP/Configuration/outputsModule?format=json

Method	GET
Description	Get linkage configuration parameters (when the relay is closed/open) of all relays.
Query	format : determine the format of request or response message.
Request	None
Response	Succeeded: <u><i>JSON_List_OutPutsModule</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

Table A-128 POST /ISAPI/SecurityCP/Configuration/outputsModule?format=json

Method	POST
Description	Search for linkage configuration parameters (when the relay is closed/open) by condition.
Query	format : determine the format of request or response message.
Request	<u><i>JSON_OutputsModuleCond</i></u>
Response	Succeeded: <u><i>JSON_OutputsModuleSearch</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.94 /ISAPI/SecurityCP/Configuration/panicButton/capabilities?format=json

Get the configuration capability of the panic button.

Request URI Definition

Table A-129 GET /ISAPI/SecurityCP/Configuration/panicButton/capabilities?format=json

Method	GET
Description	Get the configuration capability of the panic button.

Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_PanicButtonCap</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.95 /ISAPI/SecurityCP/Configuration/panicButton/zone/<ID>?format=json

Get or set panic button parameters of a specific zone.

Request URI Definition

Table A-130 GET /ISAPI/SecurityCP/Configuration/panicButton/zone/<ID>?format=json

Method	GET
Description	Get panic button parameters of a specific zone.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_PanicButton</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

Table A-131 PUT /ISAPI/SecurityCP/Configuration/panicButton/zone/<ID>?format=json

Method	PUT
Description	Set panic button parameters of a specific zone.
Query	format: determine the format of request or response message.
Request	<u><i>JSON_PanicButton</i></u>
Response	<u><i>JSON_ResponseStatus</i></u>

Remarks

The <ID> in the request URI refers to the zone No. which starts from 0.

A.2.96 /ISAPI/SecurityCP/Configuration/panicButton?format=json

Get parameters of all panic buttons.

Request URI Definition

Table A-132 GET /ISAPI/SecurityCP/Configuration/panicButton?format=json

Method	GET
Description	Get parameters of all panic buttons.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_List_PanicButton</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.97 /ISAPI/SecurityCP/Configuration/passiveInfraredDetector/capabilities?format=json

Get the configuration capability of the PIR (Passive Infrared) detector.

Request URI Definition

Table A-133 GET /ISAPI/SecurityCP/Configuration/passiveInfraredDetector/capabilities?format=json

Method	GET
Description	Get the configuration capability of the PIR (Passive Infrared) detector.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_PassiveInfraredDetectorCap</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.98 /ISAPI/SecurityCP/Configuration/passiveInfraredDetector/zone/<ID>?format=json

Get or set the PIR (Passive Infrared) detector parameters of a specific zone.

Request URI Definition

Table A-134 GET /ISAPI/SecurityCP/Configuration/passiveInfraredDetector/zone/<ID>?format=json

Method	GET
Description	Get the PIR (Passive Infrared) detector parameters of a specific zone.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_PassiveInfraredDetector</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

Table A-135 PUT /ISAPI/SecurityCP/Configuration/passiveInfraredDetector/zone/<ID>?format=json

Method	PUT
Description	Set the PIR (Passive Infrared) detector parameters of a specific zone.
Query	format: determine the format of request or response message.
Request	<u><i>JSON_PassiveInfraredDetector</i></u>
Response	<u><i>JSON_ResponseStatus</i></u>

Remarks

The <ID> in the request URI refers to the zone No. which starts from 0.

A.2.99 /ISAPI/SecurityCP/Configuration/passiveInfraredDetector?format=json

Get the parameters of all PIR (Passive Infrared) detectors.

Request URI Definition

Table A-136 GET /ISAPI/SecurityCP/Configuration/passiveInfraredDetector?format=json

Method	GET
Description	Get the parameters of all PIR (Passive Infrared) detectors.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_List_PassiveInfraredDetector</i></u>

	Failed: <u><i>JSON_ResponseStatus</i></u>
--	-------------------------------------------

A.2.100 /ISAPI/SecurityCP/Configuration/pircam/capabilities?format=json

Get the pircam (detector equipped with camera) configuration capability.

Request URI Definition

Table A-137 GET /ISAPI/SecurityCP/Configuration/pircam/capabilities?format=json

Method	GET
Description	Get the pircam (detector equipped with camera) configuration capability.
Query	format : determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_PircamCap</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.101 /ISAPI/SecurityCP/Configuration/pircam/zone/<ID>?format=json

Get or set the pircam (detector equipped with camera) parameters of a specific zone.

Request URI Definition

Table A-138 GET /ISAPI/SecurityCP/Configuration/pircam/zone/<ID>?format=json

Method	GET
Description	Get the pircam (detector equipped with camera) parameters of a specific zone.
Query	format : determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_pircam</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

Table A-139 PUT /ISAPI/SecurityCP/Configuration/pircam/zone/<ID>?format=json

Method	PUT
Description	Set the pircam (detector equipped with camera) parameters of a specific zone.
Query	format: determine the format of request or response message.
Request	<i><u>JSON_pircam</u></i>
Response	<i><u>JSON_ResponseStatus</u></i>

Remarks

The <ID> in the request URI refers to the zone No.

A.2.102 /ISAPI/SecurityCP/Configuration/pircam/zone?format=json

Get the pircam (detector equipped with camera) parameters of all zones.

Request URI Definition**Table A-140 GET /ISAPI/SecurityCP/Configuration/pircam/zone?format=json**

Method	GET
Description	Get the pircam (detector equipped with camera) parameters of all zones.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <i><u>JSON_List_pircam</u></i> Failed: <i><u>JSON_ResponseStatus</u></i>

A.2.103 /ISAPI/SecurityCP/Configuration/PSTNCfg/<ID>?format=json

Set the parameters of a specific phone notification via PSTN (Public Switched Telephone Network).

Request URI Definition

Table A-141 PUT /ISAPI/SecurityCP/Configuration/PSTNCfg/<ID>?format=json

Method	PUT
Description	Set the parameters of a specific phone notification via PSTN (Public Switched Telephone Network).
Query	format: determine the format of request or response message.
Request	<u><i>JSON_PSTNCfg</i></u>
Response	<u><i>JSON_ResponseStatus</i></u>

A.2.104 /ISAPI/SecurityCP/Configuration/PSTNCfg/capabilities?format=json

Get the configuration capability of phone notification via PSTN (Public Switched Telephone Network).

Request URI Definition

Table A-142 GET /ISAPI/SecurityCP/Configuration/PSTNCfg/capabilities?format=json

Method	GET
Description	Get the configuration capability of phone notification via PSTN (Public Switched Telephone Network).
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_PSTNCfgCap</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.105 /ISAPI/SecurityCP/Configuration/PSTNCfg?format=json

Get the configuration of all phone notifications via PSTN (Public Switched Telephone Network).

Request URI Definition

Table A-143 GET /ISAPI/SecurityCP/Configuration/PSTNCfg?format=json

Method	GET
Description	Get the parameters of all phone notifications via PSTN (Public Switched Telephone Network).
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_List_PSTNCfg</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.106 /ISAPI/SecurityCP/Configuration/publicSubSys/<ID>?format=json

Set the configuration parameters of a specific public partition.

Request URI Definition

Table A-144 PUT /ISAPI/SecurityCP/Configuration/publicSubSys/<ID>?format=json

Method	PUT
Description	Set the parameters of a specific public partition.
Query	format: determine the format of request or response message.
Request	<u><i>JSON_PublicSubSys</i></u>
Response	<u><i>JSON_ResponseStatus</i></u>

A.2.107 /ISAPI/SecurityCP/Configuration/publicSubSys/capabilities?format=json

Get the configuration capability of the public partition.

Request URI Definition

Table A-145 GET /ISAPI/SecurityCP/Configuration/publicSubSys/capabilities?format=json

Method	GET
Description	Get the configuration capability of the public partition.
Query	format: determine the format of request or response message.

Request	None.
Response	Succeeded: <u><i>JSON_PublicSubSysCap</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.108 /ISAPI/SecurityCP/Configuration/publicSubSys?format=json

Get configurations of all public partitions.

Request URI Definition

Table A-146 GET /ISAPI/SecurityCP/Configuration/publicSubSys?format=json

Method	GET
Description	Get configuration parameters of all public partitions.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_List_PublicSubSys</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.109 /ISAPI/SecurityCP/Configuration/registerMode/capabilities?format=json

Get the configuration capability of the registration mode.

Request URI Definition

Table A-147 GET /ISAPI/SecurityCP/Configuration/registerMode/capabilities?format=json

Method	GET
Description	Get the configuration capability of the registration mode.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_RegisterModeCap</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.110 /ISAPI/SecurityCP/Configuration/registerMode/registerStatus?format=json

Get the registration status.

Request URI Definition**Table A-148 GET /ISAPI/SecurityCP/Configuration/registerMode/registerStatus?format=json**

Method	GET
Description	Get the registration status.
Query	format: determine the format of request or response message.
Request	None
Response	Succeeded: <u><i>JSON_WirelessRecv</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.111 /ISAPI/SecurityCP/Configuration/registerMode?format=json

Operations about the configuration of the registration mode.

Request URI Definition**Table A-149 GET /ISAPI/SecurityCP/Configuration/registerMode?format=json**

Method	GET
Description	Get the parameters of the registration mode.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_RegisterMode</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

Table A-150 PUT /ISAPI/SecurityCP/Configuration/registerMode?format=json

Method	PUT
Description	Set the parameters of the registration mode.
Query	format: determine the format of request or response message.

Request	<i><u>JSON_RegisterMode</u></i>
Response	<i><u>JSON_ResponseStatus</u></i>

A.2.112 /ISAPI/SecurityCP/Configuration/remoteCfgPermissonUserName/capabilities?format=json

Get the capability of getting user names of users that can remotely configure devices.

Request URI Definition

Table A-151 GET /ISAPI/SecurityCP/Configuration/remoteCfgPermissonUserName/capabilities?format=json

Method	GET
Description	Get the capability of getting user names of users that can remotely configure devices.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <i><u>JSON_Cap_RemoteCfgUserName</u></i> Failed: <i><u>JSON_ResponseStatus</u></i>

A.2.113 /ISAPI/SecurityCP/Configuration/remoteCfgPermissonUserName?format=json

Get the user names of users that can remotely configure the devices.

Table A-152 GET /ISAPI/SecurityCP/Configuration/remoteCfgPermissonUserName?format=json

Method	GET
Description	Get the user names of users that can remotely configure the devices.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <i><u>JSON_RemoteCfgUserName</u></i> Failed: <i><u>JSON_ResponseStatus</u></i>

A.2.114 /ISAPI/SecurityCP/Configuration/remoteCtrl/capabilities?format=json

Get the keyfob configuration capability.

Request URI Definition**Table A-153 GET /ISAPI/SecurityCP/Configuration/remoteCtrl/capabilities?format=json**

Method	GET
Description	Get the keyfob configuration capability.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_RemoteCtrlCap</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.115 /ISAPI/SecurityCP/Configuration/remoteCtrl/currentAdd?format=json

Get the parameters of the currently added keyfob in synchronous mode.

Request URI Definition**Table A-154 GET /ISAPI/SecurityCP/Configuration/remoteCtrl/currentAdd?format=json**

Method	GET
Description	Get the parameters of the currently added keyfob.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_RemoteCtrl</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

Remarks

The parameters will be returned after they are obtained. The timeout is 20 seconds.

A.2.116 /ISAPI/SecurityCP/Configuration/remoteCtrl/currentAddAsyn?format=json

Get the parameters of the currently added keyfob in asynchronous mode.

Request URI Definition

Table A-155 GET /ISAPI/SecurityCP/Configuration/remoteCtrl/currentAddAsyn?format=json

Method	GET
Description	Get the parameters of the currently added keyfob in asynchronous mode.
Query	format : determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_RemoteCtrl</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

Remarks

- Before calling this URI, you should call */ISAPI/SecurityCP/Configuration/remoteCtrl/mode?format=json* by PUT method and set the **mode** to "enter" to enable the asynchronous keyfob adding mode for the device.
- The result will be returned immediately after calling this URI. If **status** in the response message is "processing", you should continue calling this URI until **status** is "success" and the keyfob parameters are returned. If **status** is "failed", you should stop calling this URI recursively.

A.2.117 /ISAPI/SecurityCP/Configuration/remoteCtrl/mode/capabilities?format=json

Get the capability of controlling asynchronous keyfob adding mode.

Request URI Definition

Table A-156 GET /ISAPI/SecurityCP/Configuration/remoteCtrl/mode/capabilities?format=json

Method	GET
Description	Get the capability of controlling asynchronous keyfob adding mode.
Query	format : determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_RemoteCtrlModeCap</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.118 /ISAPI/SecurityCP/Configuration/remoteCtrl/mode?format=json

Control the asynchronous keyfob adding mode.

Request URI Definition**Table A-157 PUT /ISAPI/SecurityCP/Configuration/remoteCtrl/mode?format=json**

Method	PUT
Description	Control the asynchronous keyfob adding mode.
Query	format : determine the format of request or response message.
Request	<u><i>JSON RemoteCtrlMode</i></u>
Response	<u><i>JSON ResponseStatus</i></u>

A.2.119 /ISAPI/SecurityCP/Configuration/remoteCtrl/<ID>?format=json

Operations about the configuration of a keyfob.

Request URI Definition**Table A-158 GET /ISAPI/SecurityCP/Configuration/remoteCtrl/<ID>?format=json**

Method	GET
Description	Get the parameters of a keyfob.
Query	format : determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON RemoteCtrl</i></u> Failed: <u><i>JSON ResponseStatus</i></u>

Table A-159 PUT /ISAPI/SecurityCP/Configuration/remoteCtrl/<ID>?format=json

Method	PUT
Description	Set the parameters for a keyfob.
Query	format : determine the format of request or response message.
Request	<u><i>JSON RemoteCtrl</i></u>
Response	<u><i>JSON ResponseStatus</i></u>

Table A-160 DELETE /ISAPI/SecurityCP/Configuration/remoteCtrl/<ID>?format=json

Method	DELETE
Description	Delete a keyfob.
Query	format : determine the format of request or response message.
Request	None.
Response	<u>JSON_ResponseStatus</u>

A.2.120 /ISAPI/SecurityCP/Configuration/remoteCtrl?format=json

Get the parameters of all keyfobs.

Request URI Definition

Table A-161 GET /ISAPI/SecurityCP/Configuration/remoteCtrl?format=json

Method	GET
Description	Get the parameters of all keyfobs.
Query	format : determine the format of request or response message.
Request	None.
Response	Succeeded: <u>JSON_List_RemoteCtrl</u> Failed: <u>JSON_ResponseStatus</u>

A.2.121 /ISAPI/SecurityCP/Configuration/repeaters/capabilities?format=json

Get repeater configuration capability.

Request URI Definition

Table A-162 GET /ISAPI/SecurityCP/Configuration/repeaters/capabilities?format=json

Method	GET
Description	Get repeater configuration capability.
Query	format : determine the format of request or response message.
Request	None.
Response	Succeeded: <u>JSON_RepeaterCap</u>

	Failed: <u><i>JSON_ResponseStatus</i></u>
--	-------------------------------------------

A.2.122 /ISAPI/SecurityCP/Configuration/repeaters/<ID>?format=json

Set parameters for a repeater.

Request URI Definition

Table A-163 PUT /ISAPI/SecurityCP/Configuration/repeaters/<ID>?format=json

Method	PUT
Description	Set parameters for a repeater.
Query	format: determine the format of request or response message.
Request	<u><i>JSON_Repeater</i></u>
Response	<u><i>JSON_ResponseStatus</i></u>

Remarks

The <ID> in the request URI refers to the repeater No., and it starts from 1.

A.2.123 /ISAPI/SecurityCP/Configuration/repeaters?format=json

Get all repeaters' parameters.

Request URI Definition

Table A-164 GET /ISAPI/SecurityCP/Configuration/repeaters?format=json

Method	GET
Description	Get all repeaters' parameters.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_List_Repeater</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.124 /ISAPI/SecurityCP/Configuration/signalStrengthDetection/currentAsyn?format=json

Get the current signal strength in asynchronous mode.

Request URI Definition

Table A-165 GET /ISAPI/SecurityCP/Configuration/signalStrengthDetection/currentAsyn?format=json

Method	GET
Description	Get the current signal strength in asynchronous mode.
Query	format : determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_SignalStrengthDetection</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

Remarks

- Before calling this URI, you should call */ISAPI/SecurityCP/Configuration/signalStrengthDetection/mode?format=json* by PUT method and set the **mode** to "enter" to enable signal strength detection for the device.
- The detection result will be returned immediately after calling this URI. If **status** in the response message is "processing", you should continue calling this URI until **status** is "success" and the signal strength is returned. If **status** is "failed", you should stop calling this URI recursively.

A.2.125 /ISAPI/SecurityCP/Configuration/signalStrengthDetection/mode/capabilities?format=json

Get the configuration capability of signal strength detection in asynchronous mode.

Request URI Definition

Table A-166 GET /ISAPI/SecurityCP/Configuration/signalStrengthDetection/mode/capabilities?format=json

Method	GET
Description	Get the configuration capability of signal strength detection in asynchronous mode. The signal strength detection is used to detect

	the signal strength of the detector linked with the zone, and the detection result will be returned asynchronously.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_SignalStrengthDetectionCap</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.126 /ISAPI/SecurityCP/Configuration/signalStrengthDetection/mode?format=json

Start or stop the signal strength detection in asynchronous mode.

Request URI Definition

Table A-167 POST /ISAPI/SecurityCP/Configuration/signalStrengthDetection/mode?format=json

Method	POST
Description	Start the signal strength detection in asynchronous mode.
Query	format: determine the format of request or response message.
Request	<u><i>JSON_SignalStrengthDetectionMode</i></u>
Response	Succeeded: <u><i>JSON_Result</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

Table A-168 PUT /ISAPI/SecurityCP/Configuration/signalStrengthDetection/mode?format=json

Method	PUT
Description	Stop the signal strength detection in asynchronous mode.
Query	format: determine the format of request or response message.
Request	<u><i>JSON_SignalStrengthDetectionMode</i></u>
Response	<u><i>JSON_ResponseStatus</i></u>

A.2.127 /ISAPI/SecurityCP/Configuration/slimMagneticContact/capabilities?format=json

Get the configuration capability of the slim magnetic contact detector.

Request URI Definition

Table A-169 GET /ISAPI/SecurityCP/Configuration/slimMagneticContact/capabilities?format=json

Method	GET
Description	Get the configuration capability of the slim magnetic contact detector.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_SlimMagneticContactCap</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.128 /ISAPI/SecurityCP/Configuration/slimMagneticContact/zone/<ID>?format=json

Get or set the parameters of the slim magnetic contact detector of a specific zone.

Request URI Definition

Table A-170 GET /ISAPI/SecurityCP/Configuration/slimMagneticContact/zone/<ID>?format=json

Method	GET
Description	Get the parameters of the slim magnetic contact detector of a specific zone.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_SlimMagneticContact</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

Table A-171 PUT /ISAPI/SecurityCP/Configuration/slimMagneticContact/zone/<ID>?format=json

Method	PUT
Description	Set the parameters of the slim magnetic contact detector of a specific zone.
Query	format: determine the format of request or response message.

Request	<i><u>JSON SlimMagneticContact</u></i>
Response	<i><u>JSON_ResponseStatus</u></i>

Remarks

The <ID> in the request URI refers to the zone No. which starts from 0.

A.2.129 /ISAPI/SecurityCP/Configuration/slimMagneticContact?format=json

Get the parameters of all slim magnetic contact detectors.

Request URI Definition

Table A-172 GET /ISAPI/SecurityCP/Configuration/slimMagneticContact?format=json

Method	GET
Description	Get the parameters of all slim magnetic contact detectors.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <i><u>JSON_List_SlimMagneticContact</u></i> Failed: <i><u>JSON_ResponseStatus</u></i>

A.2.130 /ISAPI/SecurityCP/Configuration/subSys/<ID>?format=json

Set the parameters of a specific partition.

Request URI Definition

Table A-173 PUT /ISAPI/SecurityCP/Configuration/subSys/<ID>?format=json

Method	PUT
Description	Set the parameters of a specific partition.
Query	format: determine the format of request or response message.
Request	<i><u>JSON_SubSys</u></i>
Response	<i><u>JSON_ResponseStatus</u></i>

A.2.131 /ISAPI/SecurityCP/Configuration/subSys/capabilities?format=json

Get the partition configuration capability.

Request URI Definition**Table A-174 GET /ISAPI/SecurityCP/Configuration/subSys/capabilities?format=json**

Method	GET
Description	Get the partition configuration capability.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_SubSysCap</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.132 /ISAPI/SecurityCP/Configuration/subSys?format=json

Get configuration parameters of all partitions.

Request URI Definition**Table A-175 GET /ISAPI/SecurityCP/Configuration/subSys?format=json**

Method	GET
Description	Get configuration parameters of all partitions.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_List_SubSys</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.133 /ISAPI/SecurityCP/Configuration/subSysTime/capabilities?format=json

Get the configuration capability of the partition timer.

Request URI Definition

Table A-176 GET /ISAPI/SecurityCP/Configuration/subSysTime/capabilities?format=json

Method	GET
Description	Get the configuration capability of the partition timer.
Query	format : determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_SubSysTimeCap</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.134 /ISAPI/SecurityCP/Configuration/subSysTime/<ID>?format=json

Set parameters of the timer of a specific partition.

Request URI Definition

Table A-177 PUT /ISAPI/SecurityCP/Configuration/subSysTime/<ID>?format=json

Method	PUT
Description	Set parameters of the timer of a specific partition.
Query	format : determine the format of request or response message.
Request	<u><i>JSON_SubSysTime</i></u>
Response	<u><i>JSON_ResponseStatus</i></u>

Remarks

The <ID> in the request URI refers to the partition No., and it starts from 1.

A.2.135 /ISAPI/SecurityCP/Configuration/subSysTime?format=json

Get the parameters of timers of all partitions.

Request URI Definition

Table A-178 GET /ISAPI/SecurityCP/Configuration/subSysTime?format=json

Method	GET
Description	Get the parameters of timers of all partitions.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <i><u>JSON_List_SubSysTime</u></i> Failed: <i><u>JSON_ResponseStatus</u></i>

A.2.136 /ISAPI/SecurityCP/Configuration/systemManage/capabilities?format=json

Get the configuration capability of security control system.

Request URI Definition

Table A-179 GET /ISAPI/SecurityCP/Configuration/systemManage/capabilities?format=json

Method	GET
Description	Get the configuration capability of security control system.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <i><u>JSON_ManageCap</u></i> Failed: <i><u>JSON_ResponseStatus</u></i>

A.2.137 /ISAPI/SecurityCP/Configuration/systemManage?format=json

Operations about the configuration of security control system.

Request URI Definition

Table A-180 GET /ISAPI/SecurityCP/Configuration/systemManage?format=json

Method	GET
Description	Get the configuration parameters of security control system.
Query	format: determine the format of request or response message.

Request	None.
Response	Succeeded: <u><i>JSON_Manage</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

Table A-181 PUT /ISAPI/SecurityCP/Configuration/systemManage?format=json

Method	PUT
Description	Set the parameters for security control system.
Query	format: determine the format of request or response message.
Request	<u><i>JSON_Manage</i></u>
Response	<u><i>JSON_ResponseStatus</i></u>

A.2.138 /ISAPI/SecurityCP/Configuration/users/capabilities?format=json

Get user configuration capability.

Request URI Definition

Table A-182 GET /ISAPI/SecurityCP/Configuration/users/capabilities?format=json

Method	GET
Description	Get user configuration capability.
Query	format: determine the format of request or response message.
Request	None.
Response	<u><i>JSON_UserCfgCap</i></u>

A.2.139 /ISAPI/SecurityCP/Configuration/users/<ID>?format=json

Operations about the configurations of a specific user.

Request URI Definition

Table A-183 PUT /ISAPI/SecurityCP/Configuration/users/<ID>?format=json

Method	PUT
Description	Set parameters of a specific user.

Query	format: determine the format of request or response message.
Request	<u><i>JSON_UserCfg</i></u>
Response	<u><i>JSON_ResponseStatus</i></u>

Table A-184 DELETE /ISAPI/SecurityCP/Configuration/users/<ID>?format=json

Method	DELETE
Description	Delete a specific user.
Query	format: determine the format of request or response message.
Request	None.
Response	<u><i>JSON_ResponseStatus</i></u>

A.2.140 /ISAPI/SecurityCP/Configuration/users?format=json

Operations about getting parameters of multiple users and adding a user.

Request URI Definition

Table A-185 GET /ISAPI/SecurityCP/Configuration/users?format=json

Method	GET
Description	Get the parameters of multiple users.
Query	format: determine the format of request or response message.
Request	None.
Response	<u><i>JSON_List_UserCfg</i></u>

Table A-186 POST /ISAPI/SecurityCP/Configuration/users?format=json

Method	POST
Description	Add a user.
Query	format: determine the format of request or response message.
Request	<u><i>JSON_UserCfg</i></u>
Response	<u><i>JSON_id</i></u>

A.2.141 /ISAPI/SecurityCP/Configuration/wiredDetector/capabilities?format=json

Get the capability of wired detectors according to the detector type.

Request URI Definition

Table A-187 POST /ISAPI/SecurityCP/Configuration/wiredDetector/capabilities?format=json

Method	POST
Description	Get the capability of wired detectors according to the wired detector type.
Query	format: determine the format of request or response message.
Request	<u><i>JSON_WiredDetectorType</i></u>
Response	Succeeded: <u><i>JSON_WiredDetectorCap</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.142 /ISAPI/SecurityCP/Configuration/wirelessSiren/capabilities?format=json

Get siren configuration capability.

Request URI Definition

Table A-188 GET /ISAPI/SecurityCP/Configuration/wirelessSiren/capabilities?format=json

Method	GET
Description	Get siren configuration capability.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_SirenCap</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.143 /ISAPI/SecurityCP/Configuration/wirelessSiren/<ID>?format=json

Set parameters for a siren.

Request URI Definition

Table A-189 PUT /ISAPI/SecurityCP/Configuration/wirelessSiren/<ID>?format=json

Method	PUT
Description	Set parameters for a siren.
Query	format : determine the format of request or response message.
Request	<u><i>JSON_Siren</i></u>
Response	<u><i>JSON_ResponseStatus</i></u>

Remarks

The <ID> in the request URI refers to the siren No., and it starts from 1.

A.2.144 /ISAPI/SecurityCP/Configuration/wirelessSiren?format=json

Get all sirens' parameters.

Request URI Definition

Table A-190 GET /ISAPI/SecurityCP/Configuration/wirelessSiren?format=json

Method	GET
Description	Get all sirens' parameters.
Query	format : determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_List_Siren</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.145 /ISAPI/SecurityCP/Configuration/wirelessSiren/currentAddAsyn?format=json

Get the parameters of the currently added siren in asynchronous mode.

Request URI Definition

Table A-191 GET /ISAPI/SecurityCP/Configuration/wirelessSiren/currentAddAsyn?format=json

Method	GET
Description	Get the parameters of the currently added siren in asynchronous mode.
Query	format : determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_Siren</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

Remarks

- Before calling this URI, you should call */ISAPI/SecurityCP/Configuration/wirelessSiren/mode?format=json* by PUT method and set the **mode** to "enter" to enable the asynchronous siren adding mode for the device.
- The result will be returned immediately after calling this URI. If **status** in the response message is "processing", you should continue calling this URI until **status** is "success" and the siren parameters are returned. If **status** is "failed", you should stop calling this URI recursively.

A.2.146 /ISAPI/SecurityCP/Configuration/wirelessSiren/mode/capabilities?format=json

Get the capability of controlling asynchronous mode of adding the siren.

Request URI Definition

Table A-192 GET /ISAPI/SecurityCP/Configuration/wirelessSiren/mode/capabilities?format=json

Method	GET
Description	Get the capability of controlling asynchronous mode of adding the siren.
Query	format : determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_SirenModeCap</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.147 /ISAPI/SecurityCP/Configuration/wirelessSiren/mode?format=json

Control the asynchronous mode of adding the siren.

Request URI Definition**Table A-193 PUT /ISAPI/SecurityCP/Configuration/wirelessSiren/mode?format=json**

Method	PUT
Description	Control the asynchronous mode of adding the siren.
Query	format: determine the format of request or response message.
Request	<u><i>JSON SirenMode</i></u>
Response	<u><i>JSON ResponseStatus</i></u>

A.2.148 /ISAPI/SecurityCP/Configuration/zoneAlarmTimeFilter/capabilities?format=json

Get the configuration capability of filtering duplicate zone alarms in the configured time interval.

Request URI Definition**Table A-194 GET /ISAPI/SecurityCP/Configuration/zoneAlarmTimeFilter/capabilities?format=json**

Method	GET
Description	Get the configuration capability of filtering duplicate zone alarms in the configured time interval.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON TimeCfgCap</i></u> Failed: <u><i>JSON ResponseStatus</i></u>

A.2.149 /ISAPI/SecurityCP/Configuration/zoneAlarmTimeFilter?format=json

Get or set parameters to filter duplicate zone alarms in the configured time interval.

Request URI Definition

Table A-195 GET /ISAPI/SecurityCP/Configuration/zoneAlarmTimeFilter?format=json

Method	GET
Description	Get parameters to filter duplicate zone alarms in the configured time interval.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_TimeCfg</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

Table A-196 PUT /ISAPI/SecurityCP/Configuration/zoneAlarmTimeFilter?format=json

Method	PUT
Description	Set parameters to filter duplicate zone alarms in the configured time interval.
Query	format: determine the format of request or response message.
Request	<u><i>JSON_TimeCfg</i></u>
Response	<u><i>JSON_ResponseStatus</i></u>

A.2.150 /ISAPI/SecurityCP/Configuration/zones/capabilities?format=json

Get the zone configuration capability.

Request URI Definition

Table A-197 GET /ISAPI/SecurityCP/Configuration/zones/capabilities?format=json

Method	GET
Description	Get the zone configuration capability.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_ZonesCap</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.151 /ISAPI/SecurityCP/Configuration/zones/currentAddAsyn?format=json

Get the parameters of the currently added zones in asynchronous mode.

Request URI Definition**Table A-198 GET /ISAPI/SecurityCP/Configuration/zones/currentAddAsyn?format=json**

Method	GET
Description	Get the parameters of the currently added zones in asynchronous mode.
Query	format: determine the format of request or response message.
Request	None
Response	Succeeded: <u><i>JSON Zone</i></u> Failed: <u><i>JSON ResponseStatus</i></u>

A.2.152 /ISAPI/SecurityCP/Configuration/zones/<ID>?format=json

Operations about the configuration of a specific zone.

Request URI Definition**Table A-199 GET /ISAPI/SecurityCP/Configuration/zones/<ID>?format=json**

Method	GET
Description	Get the parameters of a specific zone.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON Zone</i></u> Failed: <u><i>JSON ResponseStatus</i></u>

Table A-200 PUT /ISAPI/SecurityCP/Configuration/zones/<ID>?format=json

Method	PUT
Description	Set the parameters of a specific zone.
Query	format: determine the format of request or response message.

Request	<u>JSON_Zone</u>
Response	<u>JSON_ResponseStatus</u>

Remarks

The <ID> in the request URI refers to the zone No., and it starts from 0.

A.2.153 /ISAPI/SecurityCP/Configuration/zones?format=json

Get all zones' parameters.

Request URI Definition

Table A-201 GET /ISAPI/SecurityCP/Configuration/zones?format=json

Method	GET
Description	Get all zones' parameters.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u>JSON_List_Zone</u> Failed: <u>JSON_ResponseStatus</u>

A.2.154 /ISAPI/SecurityCP/control/bypassRecover/<ID>?format=json

Recover bypass of a zone.

Request URI Definition

Table A-202 PUT /ISAPI/SecurityCP/control/bypassRecover/<ID>?format=json

Method	PUT
Description	Recover bypass of a zone.
Query	format: determine the format of request or response message.
Request	None.
Response	<u>JSON_ResponseStatus</u>

Remarks

The <ID> in the request URI refers to zone No., and it starts from 0.

A.2.155 /ISAPI/SecurityCP/control/outputs?format=json

Control relay in batch.

Request URI Definition

Table A-203 POST /ISAPI/SecurityCP/control/outputs?format=json

Method	POST
Description	Control relay in batch.
Query	format: determine the format of request or response message.
Request	<u><i>JSON_OutputsCtrl</i></u>
Response	<u><i>JSON_ResponseStatus</i></u>

A.2.156 /ISAPI/SecurityCP/control/outputs/<ID>?format=json

Control a specific relay.

Request URI Definition

Table A-204 PUT /ISAPI/SecurityCP/control/outputs/<ID>?format=json

Method	PUT
Description	Control a specific relay.
Query	format: determine the format of request or response message.
Request	<u><i>JSON_OutputsCtrl</i></u>
Response	<u><i>JSON_ResponseStatus</i></u>

Remarks

The <ID> in this URI refers to the relay No., which starts from 0.

A.2.157 /ISAPI/SecurityCP/control/clearAlarm/<ID>?format=json

Clear alarms for a partition.

Request URI Definition

Table A-205 PUT /ISAPI/SecurityCP/control/clearAlarm/<ID>?format=json

Method	PUT
Description	Clear alarms for a partition.
Query	format : determine the format of request or response message.
Request	<u>JSON_Operate</u>
Response	<u>JSON_ResponseStatus</u>

Remarks

The <ID> in the request URI refers to the partition No., which starts from 1, and 0xffffffff indicates all partitions.

A.2.158 /ISAPI/SecurityCP/control/arm/<ID>?ways=<string>&format=json

Arm the partition.

Request URI Definition

Table A-206 PUT /ISAPI/SecurityCP/control/arm/<ID>?ways=<string>&format=json

Method	PUT
Description	Arm the partition.
Query	format : determine the format of request or response message. ways : the arming types, including "stay"-stay arming, and "away"-away arming, e.g., ways=stay or ways=away.
Request	<u>JSON_Operate</u>
Response	<u>JSON_ResponseStatus</u>

Remarks

- The <ID> in the request URI refers to the partition No., which starts from 1, and 0xffffffff refers to all partitions.
- If **armProcess** is returned in the response message, it indicates that the device supports arming, and the arming process will be executed.

A.2.159 /ISAPI/SecurityCP/Control/audioFile/name=

Delete the audio file.

Request URI Definition**Table A-207 DELETE /ISAPI/SecurityCP/Control/audioFile/name=**

Method	DELETE
Description	Delete the audio file.
Query	name: string, audio file name.
Request	None.
Response	<u><i>XML_ResponseStatus</i></u>

A.2.160 /ISAPI/SecurityCP/control/disarm/<ID>?format=json

Disarm the partition.

Request URI Definition**Table A-208 PUT /ISAPI/SecurityCP/control/disarm/<ID>?format=json**

Method	PUT
Description	Disarm the partition.
Query	format: determine the format of request or response message.
Request	<u><i>JSON_Operate</i></u>
Response	<u><i>JSON_ResponseStatus</i></u>

Remarks

The <ID> in the request URI refers to partition No., which starts from 1, and 0xffffffff indicates all partitions.

A.2.161 /ISAPI/SecurityCP/control/bypassRecover?format=json

Recover bypass of multiple zones.

Request URI Definition

Table A-209 PUT /ISAPI/SecurityCP/control/bypassRecover?format=json

Method	PUT
Description	Recover bypass of multiple zones.
Query	format: determine the format of request or response message.
Request	<u><i>JSON List ID</i></u>
Response	<u><i>JSON ResponseStatus</i></u>

A.2.162 /ISAPI/SecurityCP/control/bypass?format=json

Perform bypass on multiple zones.

Request URI Definition

Table A-210 PUT /ISAPI/SecurityCP/control/bypass?format=json

Method	PUT
Description	Perform bypass on multiple zones.
Query	format: determine the format of request or response message.
Request	<u><i>JSON List ID</i></u>
Response	<u><i>JSON ResponseStatus</i></u>

A.2.163 /ISAPI/SecurityCP/control/bypass/<ID>?format=json

Perform bypass on a specific zone.

Request URI Definition

Table A-211 PUT /ISAPI/SecurityCP/control/bypass/<ID>?format=json

Method	PUT
Description	Perform bypass on a specific zone.
Query	format: determine the format of request or response message.

Request	None.
Response	<u><i>JSON_ResponseStatus</i></u>

Remarks

The <ID> in the request URI refers to the zone No., and it starts from 0.

A.2.164 /ISAPI/SecurityCP/control/capabilities?format=json

Get the operation and control capability.

Request URI Definition

Table A-212 GET /ISAPI/SecurityCP/control/capabilities?format=json

Method	GET
Description	Get the operation and control capability.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_HostControlCap</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.165 /ISAPI/SecurityCP/control/siren/<ID>?format=json

Control a specific siren.

Request URI Definition

Table A-213 PUT /ISAPI/SecurityCP/control/siren/<ID>?format=json

Method	PUT
Description	Control a specific siren.
Query	format: determine the format of request or response message.
Request	<u><i>JSON_SirenCtrl</i></u>
Response	<u><i>JSON_ResponseStatus</i></u>

A.2.166 /ISAPI/SecurityCP/control/systemFault?format=json

Acknowledge the system faults.

Request URI Definition**Table A-214 PUT /ISAPI/SecurityCP/control/systemFault?format=json**

Method	PUT
Description	Acknowledge the system faults.
Query	format: determine the format of request or response message.
Request	<u><i>JSON SubSysList</i></u>
Response	<u><i>JSON ResponseStatus</i></u>

A.2.167 /ISAPI/SecurityCP/FileExport/pircam/capabilities?format=json

Get the capability of exporting pictures captured by pircam (detector equipped with camera).

Request URI Definition**Table A-215 GET /ISAPI/SecurityCP/FileExport/pircam/capabilities?format=json**

Method	GET
Description	Get the capability of exporting the picture captured by pircam (detector equipped with camera).
Query	format: determine the format of request or response message.
Request	None
Response	Succeeded: <u><i>JSON FileExportCap</i></u> Failed: <u><i>JSON ResponseStatus</i></u>

A.2.168 /ISAPI/SecurityCP/FileExport/pircam?format=json

Exporting the picture captured by pircam (detector equipped with camera).

Request URI Definition

Table A-216 POST /ISAPI/SecurityCP/FileExport/pircam?format=json

Method	POST
Description	Exporting the picture captured by pircam (detector equipped with camera).
Query	format : determine the format of request or response message.
Request	<u><i>JSON_FileExportCond</i></u>
Response	Succeeded: <u><i>JSON_FileExportInfo</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.169 /ISAPI/SecurityCP/Log/search?format=json

Search for logs of security control panel.

Request URI Definition

Table A-217 POST /ISAPI/SecurityCP/Log/search?format=json

Method	POST
Description	Search for logs of security control panel.
Query	format : determine the format of request or response message.
Request	<u><i>JSON_SearchDescription</i></u>
Response	<u><i>JSON_SearchResult</i></u>

A.2.170 /ISAPI/SecurityCP/pircam/picture/channels/<ID>/mode?format=json

Control the pircam (detector equipped with camera) to capture pictures or record videos in asynchronous mode.

Request URI Definition

Table A-218 PUT /ISAPI/SecurityCP/pircam/picture/channels/<ID>/mode?format=json

Method	PUT
Description	Control the pircam (detector equipped with camera) to capture pictures or record videos in asynchronous mode.
Query	format : determine the format of request or response message.
Request	<u><i>JSON_PircamMode</i></u>
Response	<u><i>JSON_ResponseStatus</i></u>

Remarks

The <ID> in the request URI refers to the channel No.

A.2.171 /ISAPI/SecurityCP/pircam/picture/channels/<ID>?format=json

Get the picture captured by the pircam (detector equipped with camera) in synchronous mode.

Request URI Definition

Table A-219 GET /ISAPI/SecurityCP/pircam/picture/channels/<ID>?format=json

Method	GET
Description	Get the picture captured by the pircam (detector equipped with camera) in synchronous mode.
Query	format : determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_Picture</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

Remarks

The <ID> in the request URI refers to the channel No.

A.2.172 /ISAPI/SecurityCP/pircam/picture/mode/capabilities?format=json

Get the capability of controlling the pircam (detector equipped with camera) to capture pictures or record videos in asynchronous mode.

Request URI Definition

Table A-220 GET /ISAPI/SecurityCP/pircam/picture/mode/capabilities?format=json

Method	GET
Description	Get the capability of controlling the pircam (detector equipped with camera) to capture pictures or record videos in asynchronous mode.
Query	format : determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_Cap_PircamMode</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.173 /ISAPI/SecurityCP/pircam/picture/channels/<ID>/currentAddAsyn?format=json

Get the pircam (detector equipped with camera) capture parameters being added currently in asynchronous mode.

Request URI Definition

Table A-221 GET /ISAPI/SecurityCP/pircam/picture/channels/<ID>/currentAddAsyn?format=json

Method	GET
Description	Get the pircam (detector equipped with camera) capture parameters being added currently in asynchronous mode.
Query	format : determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_Pircam</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

Remarks

- Before calling this URI, you should call */ISAPI/SecurityCP/pircam/picture/channels/<ID>/mode?format=json* by PUT method and set the **mode** to "enter" to enable the asynchronous mode for adding pircam capture parameters for the device.
- The result will be returned immediately after this URI is called. If **status** in the response message is "processing", you should continue calling this URI until **status** is "success" and the pircam capture parameters are returned. If **status** is "failed", you should stop calling this URI recursively.

A.2.174 /ISAPI/SecurityCP/status/acPowerStatus?format=json

Get the AC power supply status.

Request URI Definition**Table A-222 GET /ISAPI/SecurityCP/status/acPowerStatus?format=json**

Method	GET
Description	Get the AC power supply status.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_ACPowerStatus</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.175 /ISAPI/SecurityCP/status/armStatus?format=json

Get the arming status of a specific partition.

Request URI Definition**Table A-223 POST /ISAPI/SecurityCP/status/armStatus?format=json**

Method	POST
Description	Get the arming status of a specific partition.
Query	format: determine the format of request or response message.
Request	<u><i>JSON_SubSysList</i></u>
Response	Succeeded: <u><i>JSON_ArmStatusList</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.176 /ISAPI/SecurityCP/status/batteries?format=json

Get the storage battery voltage status.

Request URI Definition

Table A-224 GET /ISAPI/SecurityCP/status/batteries?format=json

Method	GET
Description	Get the storage battery voltage status.
Query	format: determine the format of request or response message.
Request	None.
Response	<u><i>JSON BatteryList</i></u>

A.2.177 /ISAPI/SecurityCP/status/capabilities?format=json

Get the capability of getting security control panels' status.

Request URI Definition

Table A-225 GET /ISAPI/SecurityCP/status/capabilities?format=json

Method	GET
Description	Get the capability of getting security control panels' status.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_HostStatusCap</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.178 /ISAPI/SecurityCP/status/cardReaderStatus?format=json

Get the card reader status.

Request URI Definition

Table A-226 GET /ISAPI/SecurityCP/status/cardReaderStatus?format=json

Method	GET
Description	Get the card reader status.
Query	format: determine the format of request or response message.

Request	None.
Response	Succeeded: <u><i>JSON_CardReaderList</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.179 /ISAPI/SecurityCP/status/communication?format=json

Get communication status.

Request URI Definition

Table A-227 GET /ISAPI/SecurityCP/status/communication?format=json

Method	GET
Description	Get communication status.
Query	format: determine the format of request or response message.
Request	None.
Response	<u><i>JSON_CommuniStatus</i></u>

A.2.180 /ISAPI/SecurityCP/status/exDevStatus?format=json

Get the status of all peripherals.

Request URI Definition

Table A-228 GET /ISAPI/SecurityCP/status/exDevStatus?format=json

Method	GET
Description	Get the status of all peripherals.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_ExDevStatus</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.181 /ISAPI/SecurityCP/status/extensionModuleStatus?format=json

Get the extension module status.

Request URI Definition

Table A-229 GET /ISAPI/SecurityCP/status/extensionModuleStatus?format=json

Method	GET
Description	Get the extension module status.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_ExtensionList</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.182 /ISAPI/SecurityCP/status/host?format=json

Get all status of the security control panel or get the status of the security control panel according to conditions.

Request URI Definition

Table A-230 GET /ISAPI/SecurityCP/status/host?format=json

Method	GET
Description	Get all status of the security control panel.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_AlarmHostStatus</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

Table A-231 POST /ISAPI/SecurityCP/status/host?format=json

Method	POST
Description	Get the status of the security control panel according to conditions.
Query	format: determine the format of request or response message.
Request	<u><i>JSON_AlarmHostStatusCond</i></u>
Response	Succeeded: <u><i>JSON_AlarmHostStatus</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.183 /ISAPI/SecurityCP/status/hostItself?format=json

Get the status of the security control panel itself.

Request URI Definition**Table A-232 GET /ISAPI/SecurityCP/status/hostItself?format=json**

Method	GET
Description	Get the status of the security control panel itself.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_HostStatus</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

Remarks

Because there is only one security control panel, the status does not need to be obtained by array in the response message.

A.2.184 /ISAPI/SecurityCP/status/keypadStatus?format=json

Get the keypad status.

Request URI Definition**Table A-233 GET /ISAPI/SecurityCP/status/keypadStatus?format=json**

Method	GET
Description	Get the keypad status.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_KeypadList</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.185 /ISAPI/SecurityCP/status/outputModStatus?format=json

Get the output module status.

Request URI Definition

Table A-234 GET /ISAPI/SecurityCP/status/outputModStatus?format=json

Method	GET
Description	Get the output module status.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <i><u>JSON OutputModList</u></i> Failed: <i><u>JSON ResponseStatus</u></i>

A.2.186 /ISAPI/SecurityCP/status/outputStatus?format=json

Get the relay status by specific conditions.

Request URI Definition

Table A-235 POST /ISAPI/SecurityCP/status/outputStatus?format=json

Method	POST
Description	Get the relay status by specific conditions.
Query	format: determine the format of request or response message.
Request	<i><u>JSON OutputCond</u></i>
Response	Succeeded: <i><u>JSON OutputSearch Status</u></i> Failed: <i><u>JSON ResponseStatus</u></i>

A.2.187 /ISAPI/SecurityCP/status/repeaterStatus?format=json

Get the repeater status.

Request URI Definition

Table A-236 GET /ISAPI/SecurityCP/status/repeaterStatus?format=json

Method	GET
Description	Get the repeater status.
Query	format: determine the format of request or response message.

Request	None.
Response	Succeeded: <u><i>JSON_RepeaterList</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.188 /ISAPI/SecurityCP/status/sirenStatus?format=json

Get the siren status.

Request URI Definition

Table A-237 GET /ISAPI/SecurityCP/status/sirenStatus?format=json

Method	GET
Description	Get the siren status.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_SirenList</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.189 /ISAPI/SecurityCP/status/subSystems?format=json

Get the status of all partitions.

Request URI Definition

Table A-238 GET /ISAPI/SecurityCP/status/subSystems?format=json

Method	GET
Description	Get the status of all partitions.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_SubSysList</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.190 /ISAPI/SecurityCP/status/systemFault?format=json

Get the faults of systems and partitions.

Request URI Definition**Table A-239 POST /ISAPI/SecurityCP/status/systemFault?format=json**

Method	POST
Description	Get the faults of systems and partitions.
Query	format: determine the format of request or response message.
Request	<u><i>JSON_SubSysList</i></u>
Response	Succeeded: <u><i>JSON_ArmFault</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.191 /ISAPI/SecurityCP/status/zones?format=json

Get the zone status.

Request URI Definition**Table A-240 GET /ISAPI/SecurityCP/status/zones?format=json**

Method	GET
Description	Get the status of all zones.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_ZoneList</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

Table A-241 POST /ISAPI/SecurityCP/status/zones?format=json

Method	POST
Description	Get the zone status by specific conditions.
Query	format: determine the format of request or response message.

Request	<i><u>JSON_ZoneCond</u></i>
Response	Succeeded: <i><u>JSON_ZoneSearch</u></i> Failed: <i><u>JSON_ResponseStatus</u></i>

A.2.192 /ISAPI/SecurityCP/surroundEnvironmentCfg/capabilities?format=json

Get the configuration capability of the device environment.

Request URI Definition

Table A-242 GET /ISAPI/SecurityCP/surroundEnvironmentCfg/capabilities?format=json

Method	GET
Description	Get the configuration capability of the device environment.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <i><u>JSON_SurrondParaCap</u></i> Failed: <i><u>JSON_ResponseStatus</u></i>

A.2.193 /ISAPI/SecurityCP/surroundEnvironmentCfg?format=json

Get or set the device environment parameters.

Request URI Definition

Table A-243 GET /ISAPI/SecurityCP/surroundEnvironmentCfg?format=json

Method	GET
Description	Get the device environment parameters.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <i><u>JSON_SurrondParaCfg</u></i> Failed: <i><u>JSON_ResponseStatus</u></i>

Table A-244 PUT /ISAPI/SecurityCP/surroundEnvironmentCfg?format=json

Method	PUT
Description	Set the device environment parameters.
Query	format : determine the format of request or response message.
Request	<u>JSON_SurrondParaCfg</u>
Response	<u>JSON_ResponseStatus</u>

A.2.194 /ISAPI/SecurityCP/sysAutoCheckTimeCfg/capabilities?format=json

Get the configuration capability of automatic audio and video detection.

Request URI Definition

Table A-245 GET /ISAPI/SecurityCP/sysAutoCheckTimeCfg/capabilities?format=json

Method	GET
Description	Get the configuration capability of automatic audio and video detection.
Query	format : determine the format of request or response message.
Request	None.
Response	Succeeded: <u>JSON_Cap_SysAutoCheckTimeCfg</u> Failed: <u>JSON_ResponseStatus</u>

A.2.195 /ISAPI/SecurityCP/sysAutoCheckTimeCfg?format=json

Get or set the parameters of automatic audio and video detection.

Request URI Definiton

Table A-246 GET /ISAPI/SecurityCP/sysAutoCheckTimeCfg?format=json

Method	GET
Description	Get the parameters of automatic audio and video detection.
Query	format : determine the format of request or response message.

Request	None.
Response	Succeeded: <u><i>JSON_SysAutoCheckTimeCfg</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

Table A-247 PUT /ISAPI/SecurityCP/sysAutoCheckTimeCfg?format=json

Method	PUT
Description	Set the parameters of automatic audio and video detection.
Query	format: determine the format of request or response message.
Request	<u><i>JSON_SysAutoCheckTimeCfg</i></u>
Response	<u><i>JSON_ResponseStatus</i></u>

A.2.196 /ISAPI/SecurityCP/sysCheckManually/capabilities?format=json

Get the configuration capability of manual audio and video detection.

Request URI Definition

Table A-248 GET /ISAPI/SecurityCP/sysCheckManually/capabilities?format=json

Method	GET
Description	Get the configuration capability of manual audio and video detection.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_Cap_SysCheckManually</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

A.2.197 /ISAPI/SecurityCP/sysCheckManually?format=json

Get or set the manual audio and video detection parameters.

Request URI Definition

Table A-249 GET /ISAPI/SecurityCP/sysCheckManually?format=json

Method	GET
Description	Get the manual audio and video detection parameters.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON SysCheckManually</i></u> Failed: <u><i>JSON ResponseStatus</i></u>

Table A-250 PUT /ISAPI/SecurityCP/sysCheckManually?format=json

Method	PUT
Description	Set the manual audio and video detection parameters.
Query	format: determine the format of request or response message.
Request	<u><i>JSON SysCheckManually</i></u>
Response	<u><i>JSON ResponseStatus</i></u>

A.2.198 /ISAPI/SecurityCP/videoBroadcast/customizeUpload?format=json

Upload the audio file of custom voice prompt.

Request URI Definition

Table A-251 POST /ISAPI/SecurityCP/videoBroadcast/customizeUpload?format=json

Method	POST
Description	Upload the audio file of custom voice prompt.
Query	format: determine the format of request or response message.
Request	cycleTimes (optional, int, repetition times)+audio file of custom voice prompt in binary format.
Response	<u><i>JSON ResponseStatus</i></u>

Example

Sample Code of Uploading Audio File of Custom Voice Prompt

```
Content-Type: multipart/form-data; boundary=MIME_boundary
--MIME_boundary
Content-Type: application/json
Content-Length: 480

{
  "cycleTimes":1
/*optional, integer, repetition times*/
}
--MIME_boundary
Content-Type: image/jpeg
Content-Length: 516876
Content-ID: image

fefefwageegfqaeg...
--MIME_boundary--
```

A.2.199 /ISAPI/SecurityCP/voicePrompt/capabilities?format=json

Get the configuration capability of the voice prompt.

Request URI Definition

Table A-252 GET /ISAPI/SecurityCP/voicePrompt/capabilities?format=json

Method	GET
Description	Get the configuration capability of the voice prompt.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <i><u>JSON_Cap_voicePromptCfg</u></i> Failed: <i><u>JSON_ResponseStatus</u></i>

A.2.200 /ISAPI/SecurityCP/voicePrompt?format=json

Get or set the voice prompt parameters.

Request URI Definition

Table A-253 GET /ISAPI/SecurityCP/voicePrompt?format=json

Method	GET
Description	Get the voice prompt parameters.

Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON_voicePromptCfg</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

Table A-254 PUT /ISAPI/SecurityCP/voicePrompt?format=json

Method	PUT
Description	Set the voice prompt parameters.
Query	format: determine the format of request or response message.
Request	<u><i>JSON_voicePromptCfg</i></u>
Response	<u><i>JSON_ResponseStatus</i></u>

A.2.201 /ISAPI/System/capabilities

Get device capability.

Request URI Definition

Table A-255 GET /ISAPI/System/capabilities

Method	GET
Description	Get device capability.
Query	None
Request	None.
Response	Succeeded: <u><i>XML_DeviceCap</i></u> Failed: <u><i>XML_ResponseStatus</i></u>

A.2.202 /ISAPI/System/moduleLock/config/capabilities?format=json

Get the capability of locking the module.

Request URI Definition

Table A-256 GET /ISAPI/System/moduleLock/config/capabilities?format=json

Method	GET
Description	Get the capability of locking the module.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON ModuleLockCap</i></u> Failed: <u><i>JSON ResponseStatus</i></u>

A.2.203 /ISAPI/System/moduleLock/config?format=json

Operations about the configuration of locking the module.

Request URI Definition

Table A-257 GET /ISAPI/System/moduleLock/config?format=json

Method	GET
Description	Get the parameters of locking the module.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <u><i>JSON List ModuleLock</i></u> Failed: <u><i>JSON ResponseStatus</i></u>

Table A-258 PUT /ISAPI/System/moduleLock/config?format=json

Method	PUT
Description	Set the parameters of locking the module.
Query	format: determine the format of request or response message.
Request	<u><i>JSON List ModuleLock</i></u>
Response	<u><i>JSON ResponseStatus</i></u>

A.2.204 /ISAPI/System/moduleLock/unlockModule?format=json

Unlock the module.

Request URI Definition

Table A-259 PUT /ISAPI/System/moduleLock/unlockModule?format=json

Method	PUT
Description	Unlock the module.
Query	format : determine the format of request or response message.
Request	<u><i>JSON List ModuleInfo</i></u>
Response	<u><i>JSON ResponseStatus</i></u>

A.3 Request and Response Messages

A.3.1 JSON_accessModuleAddResult

JSON message about the returned information after adding a zone/peripheral

```
{
  "id": 1,
  /*required, int, the returned zone/peripheral ID*/
  "wiredDetectorType": "other"
  /*optional, string, detector type: "dualTechnologyPirDetector",
  "tripleTechnologyPirDetector", "glassBreakDetector", "activeInfraredDetector",
  "passiveInfraredDetector", "magneticContact", "panicButton",
  "waterLeakDetector", "humidityDetector", "temperatureDetector",
  "smokeDetector", "combustibleGasDetector", "vibrationDetector", "other",
  "tamperDetector"*/
}
```

A.3.2 JSON_accessModuleAddType

JSON message about the information of zones, relays, or sirens to be added to the access module

```
{
  "accessModuleType": "localTransmitter",
  /*required, string, access module type: "localTransmitter", "multiTransmitter",
  "localZone", "localRelay", "localSiren", "keypad"*/
  "accessModuleID": 1,
  /*optional, int, access module ID, this node is valid when accessModuleType is
```

```
"multiTransmitter" or "keypad"*/
  "type": "zone",
/*required, string, type: "zone", "relay", "siren"*/
  "zoneChannelID": 1,
/*optional, int, zone channel ID, this node is valid when type is "zone"*/
  "relayChannelID": 1
/*optional, int, relay channel ID, this node is valid when type is "relay",
range:[1,2]*/
}
```

A.3.3 JSON_AccessModuleAddTypeCap

JSON message about the information of zone, relay, or siren

```
{
  "type": {
/*required, object, type*/
    "@opt": ["zone", "relay", "siren"]
/*optional, array of string*/
  },
  "zoneChannelID": {
/*optional, object, zone channel ID*/
    "@opt": [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16]
/*optional, array of int*/
  },
  "relayChannelID": {
/*optional, object, relay channel ID*/
    "@opt": [1, 2]
/*optional, array of int*/
  },
  "id": {
/*ro, req, object, the returned zone/peripheral ID*/
    "@min": 1,
/*optional, int, the minimum value*/
    "@max": 96
/*optional, int, the maximum value*/
  },
  "wiredDetectorType": {
/*optional, object, wired detector type*/
    "@opt": ["dualTechnologyPirDetector", "tripleTechnologyPirDetector",
"glassBreakDetector", "activeInfraredDetector", "passiveInfraredDetector",
"magneticContact", "panicButton", "waterLeakDetector", "humidityDetector",
"temperatureDetector", "smokeDetector", "combustibleGasDetector",
"vibrationDetector", "other", "tamperDetector"]
/*optional, array of string*/
  }
}
```

A.3.4 JSON_accessModuleType

JSON message about the access module types

```
{
  "accessModuleType": "localTransmitter",
  /*required, string, access module type: "localTransmitter", "multiTransmitter",
  "localZone", "localRelay", "localSiren", "keypad"*/
  "accessModuleID": 1
  /*optional, int, access module ID, this node is valid when accessModuleType is
  "multiTransmitter" or "keypad"*/
}
```

A.3.5 JSON_ACPowerStatus

ACPowerStatus message in JSON format

```
{
  " ACPowerStatus ":{
    "connect":""
  /*optional, string, connection status of AC power supply: "normal", "break"-
  disconnected*/
  }
}
```

A.3.6 JSON_AlarmHostStatus

JSON message about the status of security control panel

```
{
  "AlarmHostStatus":{
    "HostStatus": {
  /*optional, object, host status*/
      "tamperEvident": true,
  /*optional, boolean, zone tampering alarm status, true (triggered), false (not
  triggered)*/
      "ACConnect": true,
  /*optional, boolean, AC status: true (connected), false (disconnected)*/
      "EzvizNetwork": "wired",
  /*optional, string, access EZVIZ network: "wired", "wifi", "mobile", "offline"*/
      "smokeDetectorPowerSupply": "normal",
  /*optional, string, status of power supply for the smoke detector: "normal",
  "shorted"*/
      "powerSupply": "normal"
  /*optional, string, the status of 12-volt power supply: "normal", "shorted"*/
    },
    "ZoneList":[
  /*required, see details in JSON_ZoneList*/
```

```
],
  "IPCZoneList": [
/*required, array, list of network camera zones*/
    {
      "channelID": 1,
/*required, int, channel ID, range:[1,4]*/
      "name": "test",
/*optional, string, channel name, the maximum length is 64 bytes*/
      "chanDetectResult": "connect",
/*optional, string, channel status: "connect", "overSysBandwidth",
"domainError", "ipcStreamFail", "connecting", "chanNoError",
"ipAddrConflictWithDev", "ipAddrConflicWithIpc", "errorUserNameOrPasswd",
"netUnreachable", "unknownError", "notExist", "ipcStreamTypeNotSupport",
"ipcResolutionNotSupport", "userLocked", "userNotExist", "ipcUnregistered",
"ipcNotActivated", "poePortDetecting", "uploadingCloudFile",
"certificateValidationError"*/
      "deviceNo": 1,
/*optional, int, device No., range:[1,1000], After installation, the installer
will upload the device No. and the corresponding peripheral/detector
information to the ARC for device type recognition*/
      "linkageSubSystem": [1, 2, 3]
    }
  ],
  "SubSysList":[
/*required, see details in JSON_SubSysList*/
  ],
  "ExDevStatus":{
/*required, see details in JSON_ExDevStatus*/
    "OutputModList"[
    ],
    "OutputList":[
    ],
    "SirenList"[
    ],
    "RepeaterList":[
    ],
    "CardReaderList":[
    ],
    "ExtensionList":[
    ],
    "KeypadList":[
    ],
    "RemoteList": [
    ],
    "TransmitterList":[
    ]
  },
  "BatteryList":[{
/*optional*/
    "Battery":{
      "id": ,
```



```
        "status": "",
/*optional, string, storage battery status: "normal", "miss"-storage battery
loss*/
        "percent": "",
        "voltage":
    }
}],
    "CommuniStatus": {
/*optional, communication status*/
        "wired": "",
/*optional, string type, wired network connection status, "normal"-connected,
"break"-disconnected, "ipConflict"-IP address conflicted*/
        "wifi": "",
/*optional, string type, wireless network connection status, "normal"-
connected, "break"-disconnected*/
        "wifiSignal": ,
/*optional, integer type, Wi-Fi strength, level options: 0, 1, 2, 3, 4*/
        "mobile": "",
/*optional, string type, mobile network connection status (GPRS/3G/4G),
"normal"-connected, "break"-disconnected*/
        "mobileSignal": ,
/*optional, string type, mobile network signal strength, level options: 0, 1,
2, 3, 4*/
        "flow": ,
/*optional, float type, network traffic flow usage, unit: MB*/
        "monFlowLimit": ,
/*optional, integer type, network traffic flow limit in current month, unit:
MB, read only*/
        "cloud": "",
/*optional, string type, cloud connection status, "normal"-connected, "break"-
disconnected*/
        "wifiName": "test",
/*optional, string, Wi-Fi name*/
        "SIMNum": "test",
/*optional, string, SIM card No.*/
        "R3AverageNoise": 1,
/*optional, int, R3 average noise value, unit: dBm*/
        "RXAverageNoise": 2
/*optional, int, RX average noise value, unit: dBm*/
    }
}
}
```

Remarks

For the description of zone status, partition status, and peripheral status, refer to the messages of **JSON_ZoneList**, **JSON_SubSysList**, and **JSON_ExDevStatus** for details.

A.3.7 JSON_AlarmHostStatusCond

JSON message about the conditions for getting the status of the security control panel

```
{
  "AlarmHostStatusCond": {
    /*required, object, conditions*/
    "hostStatus": true,
    /*optional, boolean, whether to get the host status, true (yes), false (no),
    the default value is false*/
    "zoneStatus": true,
    /*optional, boolean, whether to get the zone status, true (yes), false (no),
    the default value is false*/
    "zoneNo": [1, 2, 3],
    /*optional, array, zone No. list, this node is valid when zoneStatus is true*/
    "subSys": true,
    /*optional, boolean, whether to get the partition status, true (yes), false
    (no), the default value is false*/
    "subSysNo": [1, 2, 3],
    /*optional, array, partition No. list, this node is valid when subSys is true*/
    "outputMod": true,
    /*optional, boolean, whether to get the output module status, true (yes), false
    (no), the default value is false*/
    "outputModNo": [1, 2, 3],
    /*optional, array, output module No. list, this node is valid when outputMod is
    true*/
    "output": true,
    /*optional, boolean, whether to get the relay status, true (yes), false (no),
    the default value is false*/
    "outputNo": [1, 2, 3],
    /*optional, array, relay No. list, this node is valid when output is true*/
    "siren": true,
    /*optional, boolean, whether to get the siren status, true (yes), false (no),
    the default value is false*/
    "sirenNo": [1, 2, 3],
    /*optional, array, siren No. list, this node is valid when siren is true*/
    "repeater": true,
    /*optional, boolean, whether to get the repeater status, true (yes), false
    (no), the default value is false*/
    "repeaterNo": [1, 2, 3],
    /*optional, array, repeater No. list, this node is valid when repeater is true*/
    "cardReader": true,
    /*optional, boolean, whether to get the card reader status, true (yes), false
    (no), the default value is false*/
    "cardReaderNo": [1, 2, 3],
    /*optional, array, card reader No. list, this node is valid when cardReader is
    true*/
    "extensionModule": true,
    /*optional, boolean, whether to get the extension module status, true (yes),
    false (no), the default value is false*/
    "extensionModuleNo": [1, 2, 3],
```

```
/*optional, array, extension module No. list, this node is valid when
extensionModule is true*/
    "keypad": true,
/*optional, boolean, whether to get the keypad status, true (yes), false (no),
the default value is false*/
    "keypadNo": [1, 2, 3],
/*optional, array, keypad No. list, this node is valid when keypad is true*/
    "remote": true,
/*optional, boolean, whether to get the keyfob status, true (yes), false (no),
the default value is false*/
    "remoteNo": [1, 2, 3],
/*optional, array, keyfob No. list, this node is valid when remote is true*/
    "battery": true,
/*optional, boolean, whether to get the battery status, true (yes), false (no),
the default value is false*/
    "batteryNo": [1, 2, 3],
/*optional, array, battery No. list, this node is valid when battery is true*/
    "communiStatus": true,
/*optional, boolean, whether to get the communication status, true (yes), false
(no), the default value is false*/
    "transmitter": true,
/*optional, boolean, whether to get the transmitter status, true (yes), false
(no), the default value is false*/
    "transmitterNo": [1, 2, 3],
/*optional, array, transmitter No. list, this node is valid when transmitter is
true*/
    "IPCZoneStatus": true,
/*optional, boolean, whether to get the status of the network camera zones,
true (yes), false (no), the default value is false*/
    "channelID": [1, 2, 3]
/*optional, array, network camera list*/
    }
}
```

A.3.8 JSON_ARC

JSON message about parameters of an alarm receiving center

```
{
    "ARC":{
        "id": ,
/*required, integer type, alarm center No., which starts from 1*/
        "enabled": ,
/*required, boolean type, whether to enable alarm center configuration,
"true,false"*/
        "spareEnabled": ,
/*optional, boolean type, whether to enable backup alarm receiving center,
"true", "false"*/
        "enable": ,
/*optional, boolean type, whether to enable*/
        "addrType":"","
```

```
/*required, string type, alarm center address type*/
    "ipVersion": "",
/*string type, IP address version information, it is valid only when addrType
is "ipAddress"*/
    "ipAddress": "",
/*string type, IP address types (IPv4 or IPv6), it is valid only when addrType
is "ipAddress"*/
    "hostName": "",
/*string type, domain name, it is required when addrType is "hostName"*/
    "port": ,
/*optional, integer type, port number for receiving alarm or event by the alarm
center*/
    "centerAccount": "",
/*optional, string type, alarm center account, which is used to mark the
device*/
    "protocol": "",
/*optional, string type, protocol type*/
    "transMode": "",
/*optional, string type, transmission mode*/
    "timeout": ,
/*optional, integer type, timeout of waiting for acknowledgment from the alarm
receiving center after uploading the event, the event will be uploaded again
after timeout*/
    "retryTime": ,
/*optional, integer type, re-uploading times*/
    "heartBeatInterval": ,
/*optional, integer type, heartbeat interval*/
    "algorithm": "",
/*string type, encryption algorithm, it is valid when protocol is "SIA-DCS"
and "ADM-CID"*/
    "bits": ,
/*integer type, number of bits for the encryption key, it is valid when
protocol is "SIA-DCS" and "ADM-CID"*/
    "key": "",
/*string type, key, which is used to encrypt the uploaded messages, and it is
valid when protocol is "SIA-DCS" or "ADM-CID"*/
    "authEnabled": true,
/*dependent, boolean, whether to enable authentication, this node is required
when protocol is "CSV-IP"*/
    "userName": "",
/*dependent, string, user name, this node is required when protocol is "CSV-IP"
and authEnabled is true*/
    "password": "",
/*dependent, string, password, this node is required when protocol is "CSV-IP"
and authEnabled is true*/
    "ARCchannelList": [{
/*ARC channel list*/
        "ARCchannel": {
            "ARCid": 1,
/*optional, int, ARC channel ID*/
            "enabled": true,
/*optional, boolean, whether to enable the ARC channel*/
```

```
        "transMethod":"","
/*optional, string, communication method*/
        "addrType":"","
/*required, string, ARC address type*/
        "ipVersion":"","
/*optional, string, IP address version, it is "IPv4" by default. This node is
valid when addrType is "ipAddress"*/
        "ipAddress":"","
/*dependent, string, IPv4 address or IPv6 address. This node is valid when
addrType is "ipAddress"*/
        "hostName":"","
/*dependent, string, domain name. This node is valid when addrType is
"hostName"*/
        "port":2,
/*optional, int, port No.*/
        "centerAccount":"","
/*optional, string, ARC account*/
        "protocol":"","
/*optional, string, protocol type for uploading alarms*/
        "transMode":"","
/*optional, string, transmission mode*/
        "timeout":28,
/*optional, int, timeout duration for waiting for the ARC to confirm the event
after the event is uploaded, unit: second. After timeout, the event will be
uploaded again*/
        "retryTime":3,
/*optional, int, retry times*/
        "heartBeatInterval":12,
/*optional, int, heartbeat interval*/
        "algorithm":"","
/*optional, string, encryption algorithm. This node is valid when protocol is
"*SIA-DCS" or "*ADM-CID"*/
        "bits": ,
/*optional, int, number of bits of the encryption key. This node is valid when
protocol is "*SIA-DCS" or "*ADM-CID"*/
        "key":"","
        "periodicTestEnabled": ,
/*optional, boolean, whether to enable periodic test. After the periodic test
function is enabled and configured, the device will regularly upload test
reports according to the configured interval. This node is valid when protocol
is "DC-09" (including ADM-CID, SIA-DCS, *SIA-DCS, *ADM-CID)*/
        "periodicTestTimeCfg":
/*optional, int, periodic test interval, unit: second. This node is valid when
protocol is "DC-09"*/
    }
    },
    "subSystem":[1, 2, 3],
/*optional, array of int, linked partition*/
    "sysEvent":["", ""],
/*optional, array of string, system event: "IPCOfflineTest" (network camera
disconnected detection), "wiredNetTest" (wired network fault detection),
"mobileNetTest" (mobile network fault detection), "phoneOfflineTest" (phone
```

```
disconnected detection), "hostBatteryTest" (security control panel battery
detection), "WiFiNetTest" (Wi-Fi network fault detection), "SIMTest" (SIM card
fault detection), "RS485Test" (RS-485 exception detection)*/
    "timestampGMTEnabled":true
/*optional, boolean, whether to enable GMT time stamp: true-enable (default),
false-disable. After this function is enabled, the time stamp in the
information uploaded by the device is London time; otherwise, it is the device
time. This node is valid when protocol is "DC-09"*/
    "company": "None",
/*optional, string, company: "None", "Multi" (Hungary-Multi ARC), "FranceARC"
(France ARC, or Le central d'alarme en france). This node is valid when the
value of protocol is "SIA"*/
    "pircamUploadMode": "Picture",
/*optional, string, uploading method: "Video", "Picture", this node is valid
when the value of company is "FranceARC"*/
    "transMethod": "IP",
/*optional, object, transmission mode: "IP", "PSTN" (public switched telephone
network), "serialPort"*/
    "spareARCID": 2,
/*optional, int, the No. of spare ARC: 2, 4; when the No. of the main ARC is 1,
the No. of the spare ARC is 2; when the No. of the main ARC is 3, the No. of
the spare ARC is 4*/
    "FSKCfg": {
/*optional, object, FSK configuration*/
        "baudRate": "300",
/*optional, string, baud rate: 300, 600, 1200, 2400, 4800, 9600, 14400, 19200,
38400, 56000, 57600, 115200, 128000, 256000, 230400*/
        "dataBit": 6,
/*optional, int, data bit: 6, 7, 8*/
        "parityBit": "none",
/*optional, string, parity check: "none", "odd"(odd check)*/
        "stopBit": 1
/*optional, int, stop bit: 1, 2*/
    }
}
}
```

A.3.9 JSON_ARCCap

JSON message about configuration capability of alarm receiving center

```
{
  "ARCCap":{
    "id":{
/*required, range of alarm receiving center No.*/
      "@min": ,
      "@max":
    },
    "spareEnabled": ,
/*optional, boolean type, whether to support enabling backup alarm receiving
center: "true"-yes, "false"-no*/
  }
```

```
"enable":"true,false",
/*optional, boolean type, whether to enable*/
"addrType":{
/*required, alarm receiving center address type: "ipAddress", "hostName"-domain
name*/
"@opt":"ipAddress,hostName"
},
"ipVersion":{
/*optional, IP address version information, "v4"-IPv4, "v6"-IPv6*/
"@opt":"v4,v6"
},
"ipAddress":{
/*optional, range of IP address length*/
"@min": ,
"@max":
},
"hostName":{
/*optional, range of domain name length*/
"@min": ,
"@max":
},
"port":{
/*optional, port number range for receiving alarm or event by the alarm
receiving center*/
"@min": ,
"@max":
},
"ProtoList":[{
"protocol":{
/*optional, protocol type*/
"@opt":"HTTP,NAL2300,SIA-DCS,ADM-CID,*SIA-DCS,*ADM-CID,EHome,PSTN-
CID,CSV-IP"
},
"centerAccount":{
/*optional, range of alarm center account length, which varies according to
protocol type*/
"@min": ,
"@max":
},
"transMode": {
/*optional, supported transmission mode*/
"@opt": ["TCP", "UDP"]
}
}],
"transMode":{
/*optional, supported transmission mode*/
"@opt":[
"TCP",
"UDP"
]
},
"timeout":{
```

```
/*optional, timeout range of waiting for acknowledgment from the alarm
receiving center after uploading the event, the event will be uploaded again
after timeout, unit: second*/
    "min": ,
    "max":
},
    "retryTime":{
/*optional, range of re-uploading times*/
    "min": ,
    "max":
},
    "heartBeatInterval":{
/*optional, heartbeat interval range, unit: second*/
    "min": ,
    "max":
},
    "algorithm":{
/*encryption algorithm, it is valid when protocol contains "*SIA-DCS" and "*ADM-
CID"*/
    "@opt":[
        "AES"
    ]
},
    "bits":{
/*number of bits range for the encryption key, it is valid when protocol
contains "*SIA-DCS" and "*ADM-CID"*/
    "@opt":[
        128,
        192,
        256
    ]
},
    "key":{
/*optional, range of encryption key length, it is valid when protocol contains
"*SIA-DCS" and "*ADM-CID"*/
    "@min": ,
    "@max":
}
    "method":{
/*required, methods supported by the function: "put"-edit, "get", getAll*/
    "@opt": "put,get,getAll"
},
    "authEnabled":{
/*dependent, boolean, whether to enable authentication, this node is required
when protocol is "CSV-IP"*/
    "@opt": [true,false]
},
    "userName":{
/*dependent, string, user name, this node is required when protocol is "CSV-IP"
and authEnabled is true*/
    "@min":,
    "@max":
```



```

    },
    "password":{
/*dependent, string, password, this node is required when protocol is "CSV-IP"
and authEnabled is true*/
        "@min":,
        "@max":
    },
    "ARCchannelList":{
/*ARC channel list*/
        "ARCchannel":{
            "ARCcid":{
/*optional, int, ARC channel ID*/
                "@min":1,
                "@max":2
            },
            "enabled":{
/*optional, boolean, whether to enable the ARC channel*/
                "@opt":[true, false]
            },
            "transMethod":{
/*optional, string, communication method: "mobileNetwork" (mobile network),
"LAN&WLAN" (LAN and wireless LAN), "PSTN"*/
                "@opt":["mobileNetwork", "LAN&WLAN", "PSTN"]
            },
            "addrType":{
                "@opt":"ipAddress,hostName"
            },
            "ipVersion":{
                "@opt":"v4,v6"
            },
            "ipAddress":{
                "@min": ,
                "@max":
            },
            "hostName":{
                "@min": ,
                "@max":
            },
            "port":{
                "@min": ,
                "@max":
            },
            "ProtoList":{
                "protocol":{
                    "@opt":"HTTP,NAL2300,SIA-DCS,ADM-CID,*SIA-DCS,*ADM-CID,EHome,PSTN-
CID"
                },
                "centerAccount":{
                    "@min": ,
                    "@max":
                },
                "algorithm":{

```

```
/*optional, string, encryption algorithm. This node is valid when protocol
contains "*SIA-DCS" and "*ADM-CID"*/
    "@opt":["AES"]
},
    "List":[{
        "bits":{
/*optional, int, number of bits of the encryption key. This node is valid when
protocol contains "*SIA-DCS" and "*ADM-CID"*/
            "@opt":[128, 192, 256]
        },
        "key":{
/*optional, int, range of the encryption key length. This node is valid when
protocol contains "*SIA-DCS" and "*ADM-CID"*/
            "@min": ,
            "@max":
        }
    }]
},
    "transMode":{
/*optional, string, supported transmission mode*/
    "@opt":["TCP", "UDP"]
},
    "timeout":{
/*optional, int, range of timeout duration for waiting for the ARC to confirm
the event after the event is uploaded, unit: second. After timeout, the event
will be uploaded again*/
        "min": ,
        "max":
    },
    "retryTime":{
/*optional, int, range of retry times*/
        "min": ,
        "max":
    },
    "heartBeatInterval":{
/*optional, int, range of heartbeat interval, unit: second*/
        "min": ,
        "max":
    },
    "periodicTestEnabled":{
/*optional, boolean, whether to enable periodic test. After the periodic test
function is enabled and configured, the device will regularly upload test
reports according to the configured interval. This node is valid when protocol
is "DC-09" (including ADM-CID, SIA-DCS, *SIA-DCS, *ADM-CID)*/
        "@opt":[true,false]
    },
    "periodicTestTimeCfg":{
/*optional, int, periodic test interval, unit: second. This node is valid when
protocol is "DC-09"*/
        "@min": ,
        "@max":
    }
}
```

```

    }
  },
  "subSystem":{
/*linked partition*/
    "@min":1,
/*optional, int, minimum partition No. that can be linked*/
    "@max":2,
/*optional, int, maximum partition No. that can be linked*/
    "@size":2
/*optional, int, maximum number of partitions that can be linked*/
  },
  "sysEvent":{
/*system event*/
    "@opt":["IPCOOfflineTest", "wiredNetTest", "mobileNetTest"],
/*optional, string, system event: "IPCOOfflineTest" (network camera disconnected
detection), "wiredNetTest" (wired network fault detection), "mobileNetTest"
(mobile network fault detection), "phoneOfflineTest" (phone disconnected
detection), "hostBatteryTest" (security control panel battery detection),
"WiFiNetTest" (Wi-Fi network fault detection), "SIMTest" (SIM card fault
detection), "RS485Test" (RS-485 exception detection)*/
    "@size":2
/*optional, int, maximum number of system events*/
  },
  "timeStampGMTEnabled":{
/*optional, boolean, whether to enable GMT time stamp: true-enable (default),
false-disable. After this function is enabled, the time stamp in the
information uploaded by the device is London time; otherwise, it is the device
time. This node is valid when protocol is "DC-09"*/
    "@opt":[true,false]
  },
  "company": {
/*optional, object, company: "None", "Multi" (Hungary-Multi ARC), "FranceARC"
(France ARC, or Le central d'alarme en france). This node is valid when the
value of protocol is "SIA"*/
    "@opt": ["None", "Multi", "FranceARC"]
  },
  "PircamUploadModeList": [{
/*optional, array, pircam camera uploading method list, this node is valid when
the value of company is "FranceARC"*/
    "company": "FranceARC",
/*optional, string, company*/
    "pircamUploadMode": {
/*optional, object, uploading method*/
      "@opt": ["Video", "Picture"]
    }
  }],
  "transMethod": {
/*optional, object, transmission mode: "IP", "PSTN" (public switched telephone
network), "serialPort"*/
    "@opt": ["IP", "PSTN", "serialPort"]
/*optional, array of string*/
  },

```

```
"SpareARCList": [  
/*optional, array, spare ARC list*/  
  {  
    "mainARCID": 1,  
/*optional, int, the main ARC No., when the No. of the main ARC is 1, the No.  
of the spare ARC is 2; when the No. of the main ARC is 3, the No. of the spare  
ARC is 4*/  
    "spareARCID": 2  
/*optional, int, the No. of spare ARC: 2, 4*/  
  },  
  {  
    "FSKCfg": {  
/*optional, object, FSK configuration*/  
      "baudRate": {  
/*optional, object, baud rate*/  
        "@opt": [300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 38400, 56000,  
57600, 115200, 128000, 256000, 230400]  
/*optional, array of int*/  
      },  
      "dataBit": {  
/*optional, object, data bit*/  
        "@opt": [6, 7, 8]  
/*optional, array of int*/  
      },  
      "parityBit": {  
/*optional, object, parity check*/  
        "@opt": ["none", "odd"]  
/*optional, array of string, "none", "odd"(odd check). Odd check: if the number  
in the data bit "1" is an even number, the parity bit is "1"; if the number in  
the data bit "1" is an odd number, the parity bit is "0"*/  
      },  
      "stopBit": {  
/*optional, object, stop bit*/  
        "@opt": [1, 2]  
/*optional, array of int*/  
      }  
    }  
  }  
]
```

A.3.10 JSON_ARCManualTest

JSON message about the parameters of manual test

```
{  
  "id": 1  
/*required, int, ARC ID, range:[1,4]*/  
}
```

A.3.11 JSON_ARCManualTestCap

JSON message about the capability of ARC manual test

```
{
  "id": {
    /*required, object, ARC ID*/
    "@min": 1,
    /*optional, int, the minimum value*/
    "@max": 4
    /*optional, int, the maximum value*/
  },
  "status": {
    /*required, object, status*/
    "@opt": ["success", "processing", "failed"]
    /*optional, array of string: "success", "processing", "failed"*/
  }
}
```

A.3.12 JSON_ARCManualTestID

JSON message about the ID of the ARC

```
{
  "id": 1
  /*required, int, ARC ID, range:[1,4]*/
}
```

A.3.13 JSON_ARCManualTestStatus

JSON message about the manual test status of a specific ARC

```
{
  "status": "success"
  /*required, string, status: "success", "processing", "failed"*/
}
```

A.3.14 JSON_ArmFault

JSON message about faults of systems and partitions

```
{
  "ArmFault":{
    "status": "",
    /*required, string, fault detection status: "checking"-detecting, "checked"-
    detected*/
  }
}
```

```
"SysFault":{
/*optional, system fault, which can be set to NULL if there is no fault*/
  "FaultList":[{
/*required, fault list*/
    "Fault":{
/*required*/
      "id": ,
/*required, int, fault No., which starts from 1*/
      "info":"",
/*required, read-only, string, fault information:
"wirelessOutputModTamperEvident"-output module tampered,
"wirelessRepeaterTamperEvident"-repeater tampered,
"wirelessKeypadTamperEvident"-wireless keypad tampered,
"wirelessCardReaderTamperEvident"-card reader tampered,
"wirelessSirenTamperEvident"-siren tampered, "devRemove"-security control panel
tampered, "wirelessOutputModOffline"-output module offline,
"wirelessRepeaterOffline"-repeater offline, "wirelessKeypadOffline"-wireless
keypad offline, "wirelessCardReaderOffline"-card reader offline,
"wirelessSirenOffline"-siren offline, "wOutputOvertime"-output module heartbeat
timeout, "wRepeaterOvertime"-repeater heartbeat timeout, "wKeypadOvertime"-
keypad heartbeat timeout, "wCardReaderOvertime"-card reader heartbeat timeout,
"wSirenOvertime"-siren heartbeat timeout, "keyfobLowPower"-keyfob battery low,
"keypadLowPower"-low keypad battery, "cardReaderLowPower"-low card reader
battery, "sirenLowPower"-siren battery low, "lowBatteryVoltage"-low storage
battery voltage, "batteryMiss"-storage battery fault, "ACLoss"-AC powered off,
"wiredNetAbnormal"-wired network fault, "GPRSAbnormal"-GPRS network fault,
"3GAbnormal"-3G network fault, "SIMCardAbnormal"-SIM card exception,
"IPCIpconflict"-network cameras' IP addresses conflict, "wifiAbnormal"-Wi-Fi
communication fault, "RFAbnormal"-RF signal exception, "dataTrafficOverflow"-
cellular network data exceeded, "ipcDisconnect"-network camera disconnected,
"virtualDefenceBandit"-virtual zone burglary alarm, "virtualDefenceFire"-
virtual zone fire alarm, "virtualDefenceUrgent"-virtual zone emergency alarm,
"ARCUploadFailed"-ARC uploading failed, "RS485ZoneModTamperEvident"-RS-485 zone
module tampered, "RS485WirelessacceptorTamperEvident"-RS-485 wireless receiver
module tampered, "RS485ZoneModOffline"-RS-485 zone module offline,
"RS485OutputModOffline"-RS-485 output module offline,
"RS485WirelessacceptorOffline"-RS-485 wireless receiver module offline,
"telLineBroken"-telephone line disconnected, "RS485DisConnect"-RS-485 bus
exception, "keypadTamperEvident"-keypad tampered, "keypadOffline"-keypad
offline*/
      "List":[{
/*No. list of output modules, repeaters, sirens, or keyfobs, which is valid
when info is set to the fault about output module, repeater, siren, or keyfob*/
        "id": ,
/*required, int, No. of output module, repeater, siren or keyfob, which starts
from 1*/
        "deviceName": "test"
/*optional, string, name of output module, repeater, siren, or keyfob*/
      }
    ]
  }
},
```

```
"SubSysFaultList":[{
/*required, faults of all partitions, which can be set to NULL if there is no
fault*/
  "SubSysFault":{
/*required, partition fault*/
    "id": ,
/*required, int, partition No., which starts from 1*/
    "name": "",
/*optional, string, partition name*/
    "armWithFault": ,
/*optional, boolean, whether to allow arming with faults: "true"-yes, "false"-
no*/
    "FaultList":[{
/*required, list of partition faults*/
      "Fault":{
/*required, fault information*/
        "id": ,
/*required, int, fault No., which starts from 1*/
        "type": "",
/*required, string, fault type: "triggerTemper"-zone tampered,
"detectorOffline"-zone offline, "detectorOvertime"-heartbeat timeout,
"detectorBatteryLow"-zone battery low, "shortCircuit"-zone triggered*/
        "ZoneList":[{
/*required, list of partitions with faults*/
          "Zone":{
            "id": ,
/*required, int, zone No., which starts from 0*/
            "zoneName": "test"
/*optional, string, zone name*/
          }
        ]
      }
    ]
  }
}]
}
```

A.3.15 JSON_ArmStatusList

JSON message about arming status of a specific partition

```
{
  "ArmStatusList":[{
    "ArmStatus":{
      "id": ,
/*required, int, partition No., which starts from 1*/
      "name": "",
/*optional, string, partition name*/
      "status": "",
/*required, string, arming status, "armed"-armed, "arming"-arming, "armFailed"-
```

```
arming failed, "disarmed"-disarmed*/
    "exitDelayTime": ,
/*int, exiting delay time, which is valid when status is "arming", unit:
second*/
    "SysFault":{
/*system fault, which is valid when status is "armFailed"*/
    "FaultList":[{
/*required, list of faults*/
        "Fault":{
/*required*/
            "id": ,
/*required, int, fault No., which starts from 1*/
            "info": "",
/*required, read-only, string, fault information:
"wirelessOutputModTamperEvident"-output module tampered,
"wirelessRepeaterTamperEvident"-repeater tampered,
"wirelessKeypadTamperEvident"-wireless keypad tampered,
"wirelessCardReaderTamperEvident"-card reader tampered,
"wirelessSirenTamperEvident"-siren tampered, "devRemove"-security control panel
tampered, "wirelessOutputModOffline"-output module offline,
"wirelessRepeaterOffline"-repeater offline, "wirelessKeypadOffline"-wireless
keypad offline, "wirelessCardReaderOffline"-card reader offline,
"wirelessSirenOffline"-siren offline, "wOutputOvertime"-output module heartbeat
timeout, "wRepeaterOvertime"-repeater heartbeat timeout, "wKeypadOvertime"-
keypad heartbeat timeout, "wCardReaderOvertime"-card reader heartbeat timeout,
"wSirenOvertime"-siren heartbeat timeout, "keyfobLowPower"-keyfob battery low,
"keypadLowPower"-low keypad battery, "cardReaderLowPower"-low card reader
battery, "sirenLowPower"-siren battery low, "lowBatteryVoltage"-low storage
battery voltage, "batteryMiss"-storage battery fault, "ACLoss"-AC powered off,
"wiredNetAbnormal"-wired network fault, "GPRSAbnormal"-GPRS network fault,
"3GAbnormal"-3G network fault, "SIMCardAbnormal"-SIM card exception,
"IPCIPconflict"-network cameras' IP addresses conflict, "wifiAbnormal"-Wi-Fi
communication fault, "RFAbnormal"-RF signal exception, "dataTrafficOverflow"-
cellular network data exceeded, "ipcDisconnect"-network camera disconnected,
"virtualDefenceBandit"-virtual zone burglary alarm, "virtualDefenceFire"-
virtual zone fire alarm, "virtualDefenceUrgent"-virtual zone emergency alarm,
"ARCUploadFailed"-uploading to ARC failed*/
            "List":[{
/*No. list of output modules, repeaters, sirens, or keyfobs, which is valid
when info is set to the fault about output module, repeater, siren, or keyfob*/
                "id":
/*required, int, No. of output module, repeater, siren or keyfob, which starts
from 1*/
                "deviceName": "test"
/*optional, string, name of output module, repeater, siren or keyfob*/
            }]}
        ],
    },
    "FaultList":[{
/*list of partitions' faults, which is valid when status is "armFailed"*/
        "Fault":{
```



```
/*required, fault information*/
    "id": ,
/*required, int, fault No., which starts from 1*/
    "type": "",
/*required, string, fault type: "triggerTemper"-zone tampered,
"detectorOffline"-zone offline, "detectorOvertime"-heartbeat timeout,
"detectorBatteryLow"-zone battery low, "shortCircuit"-zone triggered*/
    "ZoneList": [{
/*required, list of zones with faults*/
        "Zone": {
            "id":
/*required, int, zone No., which starts from 1*/
            "zoneName": "test"
/*optional, string, zone name*/
        }
    }]
}
}
```

A.3.16 JSON_BatteryList

BatteryList message in JSON format

```
{
  "BatteryList": [{
    "Battery": {
      "id": ,
/*required, integer type, battery No.*/
      "status": "",
/*optional, string type, storage battery status, "normal", "miss"-battery loss*/
      "percent": ,
/*optional, integer type, power percentage, unit: %*/
      "voltage":
/*optional, integer type, battery voltage, unit: V*/
    }
  }]
}
```

A.3.17 JSON_Cap_CardMode

JSON message about the capability of controlling card addition mode in asynchronous mode

```
{
  "CardMode": {
    "mode": {
/*optional, mode, "enter", "exit"*/

```

```
    "@opt":["enter","exit"]
  },
  "keypadAddress":{
/*optional, keypad address, this node is valid when mode is "enter"*/
    "@opt":[1,3,5...]
  },
  "sequence": {
/*optional, object, the length of the peripheral serial No.*/
    "@min": 0,
/*optional, int, the minimum value*/
    "@max": 1
/*optional, int, the maximum value*/
  },
  "deviceType": {
/*optional, object, device type*/
    "@opt": ["DS-PK1-E-WE", "DS-PT1-WE"]
  },
  "wirelessRecvAddress": {
/*optional, object, address of the wireless receiving module, this node is
valid when mode is enter*/
    "@opt": [1, 3, 5]
  }
}
}
```

A.3.18 JSON_Cap_CardReaderMode

Message about the capability of controlling asynchronous mode of adding card reader parameters in JSON format

```
{
"CardReaderMode": {
  "mode": {
/*required, string, mode: "enter", "exit"*/
    "@opt":["enter","exit"]
  },
  "sequence":{
/*optional, int, length of the peripheral's serial No.*/
    "@min":0,
    "@max":1
  },
  "deviceType":{
/*optional, string, peripheral model. Peripheral models supported by the device
will be returned*/
    "@opt":["DS-PK1-E-WE","DS-PT1-WE"]
  }
}
}
```

A.3.19 JSON_Cap_CheckResult

Message about the detection results in JSON format.

```
{
  "CheckResult":{
    "checkResult":{
/*optional, string, detection result: "normal", "abnormal", "unknown",
"processing"*/
      "@opt":["normal", "abnormal", "unknown", "processing"]
    },
    "value":{
/*optional, integer, detection result which is displayed in the format of a
number*/
      "@min": ,
      "@max":
    },
    "videoSignal":{
/*optional, string, video signal status: "normal", "abnormal", "unknown"*/
      "@opt":["normal", "abnormal", "unknown"]
    },
    "audioInput":{
/*optional, string, audio input status: "normal", "abnormal", "unknown"*/
      "@opt":["normal", "abnormal", "unknown"]
    },
    "audioOutput":{
/*optional, string, audio output status: "normal", "abnormal", "unknown"*/
      "@opt":["normal", "abnormal", "unknown"]
    }
  }
}
```

A.3.20 JSON_Cap_MuteVoicePlanCFG

MuteVoicePlanCFG capability message in JSON format

```
{
  "MuteVoicePlanCFG":{
    "enable":"true,false",
/*required, boolean, whether to enable muting: "true"-enable, "false"-disable*/
    "WeekPlanCfg":{
/*optional, week schedule parameters*/
      "maxSize":56,
/*optional, maximum number of schedules that can be configured. Up to 56
schedules can be supported, which means that you can configure up to eight time
periods for each day in a week*/
      "week":{
/*optional, string, day of the week: "Monday", "Tuesday", "Wednesday",
"Thursday", "Friday", "Saturday", "Sunday"*/
```

```
    "@opt":["Monday", "Tuesday", "Wednesday", "Thursday", "Friday",
"Saturday", "Sunday"]
  },
  "id":{
/*optional, configured time period*/
    "@min":1,
    "@max":8
  },
  "enable":"true,false",
/*optional, whether to enable: "true"-enable, "false"-disable*/
  "TimeSegment":{
/*optional, time period parameters*/
    "beginTime":"",
/*start time (device's local time)*/
    "endTime":""
/*end time (device's local time)*/
  }
}
}
```

A.3.21 JSON_Cap_PircamMode

JSON message about the capability of controlling the pircam (detector equipped with camera) to capture pictures or record videos in asynchronous mode

```
{
  "PircamMode":{
    "mode":{
/*required, string, mode: "enter", "exit"*/
      "@opt":["enter", "exit"]
    }
  }
}
```

A.3.22 JSON_Cap_RemoteCfgUserName

RemoteCfgUserName capability message in JSON format

```
{
  "RemoteCfgUserName":{
    "userName":{
/*string, user names that can remotely configure devices*/
      "@min": ,
      "@max":
    }
  }
}
```

A.3.23 JSON_Cap_SysAutoCheckTimeCfg

Message about the automatic audio and video detection capability in JSON format.

```
{
  "SysAutoCheckTimeCfg":{
    "enable":"true,false",
    /*required, boolean, whether to enable automatic audio and video detection:
    true-enable, false-disable*/
    "dayOfWeek":{
      /*optional, string, day of the week: "Monday", "Tuesday", "Wednesday",
      "Thursday", "Friday", "Saturday", "Sunday", "Everyday", "Null"*/
      "@opt":["Null", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday",
      "Saturday", "Sunday", "Everyday"]
    },
    "TimeSegment":{
      /*optional, time period*/
      "beginTime":""
      /*optional, string, start time*/
    }
  }
}
```

A.3.24 JSON_Cap_SysCheckManually

Message about the configuration capability of manual audio and video detection.

```
{
  "SysCheckManually":{
    "enable":"true,false"
    /*required, boolean, whether to enable manual detection of the system's audio
    and video: true-enable, false-disable*/
  }
}
```

A.3.25 JSON_Cap_voicePromptCfg

JSON message about configuration capability of voice prompt

```
{
  "voicePromptCfg":{
    "callPrompt":{
      /*optional, voice prompt for calling*/
      "@min":1,
      "@max":2
    },
    "rejectPrompt":{
      /*optional, voice prompt for rejection*/
    }
  }
}
```

```
        "@min":1,
        "@max":2
    },
    "voiceTalkStopPrompt":{
/*optional, end voice prompt for two-way audio*/
        "@min":1,
        "@max":2
    }
}
}
```

A.3.26 JSON_Card

Card message in JSON format

```
{
  "Card":{
    "status":"","
/*required, string, current status: "processing", "success", "failed"*/
    "failedReason":"","
/*optional, string, reason for failure: "repeatAdd" (the keyfob has been added
by this or other control panel), this node is valid when status is "failed"*/
    "id": ,
/*required, int, card No., which starts from 1*/
    "enabled": ,
/*required, boolean, whether to enable card function: "true"-yes, "false"-no*/
    "seq":"","
/*required, string, card serial No.*/
    "name":"","
/*optional, string, card name*/
    "armEnabled": ,
/*optional, boolean, whether to have arming permission*/
    "disarmEnabled": ,
/*optional, boolean, whether to have disarming permission*/
    "subSystem": ,
/*optional, array, linked partitions, e.g., [1,2,3] indicates linking to
partition 1, partition 2, and partition 3*/
    "relatedNetUserName":"","
/*optional, string, linked network user name*/
    "cardType":""
/*optional, string, card type: "operateCard"-operation card, "patrolCard"-
patrol card*/
  }
}
```

A.3.27 JSON_CardCap

CardCap capability message in JSON format

```
{
  "CardCap":{
    "id":{
/*required, card No.*/
      "@min": ,
      "@max":
    },
    "enabled":{
/*required, whether to enable card function, "true"-yes, "false"-no*/
      "@opt":
    },
    "seq":{
/*required, length of the card serial No.*/
      "@min": ,
      "@max":
    },
    "name":{
/*optional, card name length*/
      "@min": ,
      "@max":
    },
    "armEnabled":{
/*optional, whether to support configuring arming permission*/
      "@opt":
    },
    "disarmEnabled":{
/*optional, whether to support configuring disarming permission*/
      "@opt":
    }
    "subSystem":{
/*optional, supported number of partitions that can be linked*/
      "@min":,
      "@max":
    },
    "subSystemNo":{
/*optional, partition No. range*/
      "@min":,
      "@max":
    },
    "cardType": {
/*optional, card type: "operateCard"-operation card, "patrolCard"-patrol card*/
      "@opt":["operateCard","patrolCard"]
    },
    "method":{
/*optional, methods supported by the function: "currentAddAsyn"-asynchronously
add*/
      "@opt":["currentAddAsyn"]
    }
  }
}
```

A.3.28 JSON_CardMode

JSON message about parameters of controlling card addition mode in asynchronous mode

```
{
  "CardMode":{
    "mode":"","
    /*optional, string, mode: "enter", "exit"*/
    "keypadAddress": ,
    /*optional, integer, keypad address, this node is valid when mode is "enter"*/
    "sequence": "test",
    /*optional, string, the peripheral serial No.*/
    "deviceType": "test",
    /*optional, string, device type*/
    "wirelessRecvAddress": 1
    /*optional, int, address of the wireless receiving module*/
  }
}
```

A.3.29 JSON_CardReader

JSON message about the parameters of a card reader

```
{
  "CardReader":{
    "status":"","
    /*required, string, current status: "processing", "success", "failed"*/
    "enabled": ,
    /*optional, boolean, whether to enable the card reader*/
    "id": ,
    /*optional, integer type, card reader No., it starts from 1*/
    "related": ,
    /*required, boolean type, whether to link to the physical card reader. For PUT
    method, this node is optional; for GET method, this node is required*/
    "seq":"","
    /*optional, string type, card reader serial No., this node is required when
    related is "true"*/
    "name":"","
    /*optional, string type, card reader name*/
    "subSystem": ,
    /*optional, array, partition No. range, e.g., [1,2.3] indicates linking to
    partition 1, partition 2, and partition 3*/
    "buzzerEnabled": ,
    /*optional, boolean type, whether to enable the buzzer: "true"-yes, "false"-no*/
    "checkTime": ,
    /*optional, integer type, offline time, unit: hour*/
    "LEDEnabled": ,
    /*optional, boolean, whether to enable the LED indicator*/
    "heartBeatInterval": ,
  }
}
```



```
/*optional, int, heartbeat interval of the security control panel and the
peripheral, unit: second*/
    "operationMode": "simple"
/*optional, object, operation mode: "simple", "standard"*/
}
}
```

A.3.30 JSON_CardReaderCap

JSON message about the card reader configuration capability

```
{
  "CardReaderCap":{
    "enabled":{
/*optional, boolean, whether to enable the card reader*/
      "@opt":[true,false]
    },
    "id":{
/*optional, card reader No.*/
      "@min": ,
      "@max":
    },
    "related":"true,false",
/*required, whether to link to the physical card reader*/
    "seq":{
/*optional, card reader serial No.*/
      "@min": ,
      "@max":
    },
    "name":{
/*optional, card reader name*/
      "@min": ,
      "@max":
    },
    "subSystem":{
/*optional, supported number of partitions that can be linked*/
      "@min": ,
      "@max":
    },
    "subSystemNo":{
/*optional, partition No. range*/
      "@min": ,
      "@max":
    },
    "buzzerEnabled":"true,false",
/*optional, whether to enable the buzzer: "true"-yes, "false"-no*/
    "checkTime":{
/*optional, offline time, unit: hour*/
      "@min": ,
      "@max":
    },
  },
}
```

```
"LEDEnabled": {
/*optional, boolean, whether to enable the LED indicator*/
  "@opt": [true, false]
},
"method": {
/*methods supported by the function: "put"-edit, "getAll"-get all, "add"*/
  "@opt": ["put", "getAll", "add"]
},
"heartbeatInterval": {
/*optional, int, heartbeat interval of the security control panel and the
peripheral, unit: second*/
  "@opt": [5, 10, 20, 30]
},
"isSupportSignalTest": true,
/*optional, boolean, whether it supports signal strength detection*/
"isSupportZoneTest": true,
/*optional, boolean, whether zone test is supported*/
"isSupportFindMe": true,
/*optional, boolean, whether to enable "find me" function*/
"operationMode": {
/*optional, object, operation mode*/
  "@opt": ["simple", "standard"]
}
}
}
```

A.3.31 JSON_CardReaderList

JSON message about the card reader status

```
{
  "CardReaderList": [{
/*optional, card reader list*/
    "CardReader": {
      "id": ,
/*required, int, card reader No.*/
      "seq": "",
/*required, string, peripheral serial No.*/
      "name": "",
/*optional, string, card reader name*/
      "status": "",
/*optional, card reader status: "notRelated"-not linked, "online", "offline",
"heartbeatAbnormal"-heartbeat exception*/
      "tamperEvident": ,
/*optional, boolean, tampering status: "true"-tampered, "false"-not tampered*/
      "charge": "",
/*optional, string, state of charge: "normal", "lowPower"-low battery*/
      "chargeValue": ,
/*optional, int, battery power value which is between 0 and 100*/
      "signal": ,
/*optional, int, signal strength, it is between 0 and 255*/

```

```
    "model": "",
/*optional, string, model*/
    "temperature": ,
/*optional, read-only, int, temperature*/
    "subSystemList": [1, 2, 3],
/*optional, array, list of linked partitions*/
    "isViaRepeater": true,
/*optional, boolean, whether the signal is transmitted via repeater*/
    "repeaterName": "test",
/*optional, string, repeater name, the maximum length is 64 bytes, this node is
valid when isViaRepeater is true*/
    "version": "test",
/*optional, string, 版本号, range:[1,32]*/
    "deviceNo": 1
/*optional, int, device No., range:[1,1000]*/
  }
}]
}
```

A.3.32 JSON_CardReaderMode

Message about parameters for controlling the asynchronous mode of adding card reader parameters in JSON format

```
{
  "CardReaderMode": {
    "mode": "",
/*required, string, mode: "enter", "exit"*/
    "sequence": "",
/*required, string, peripheral's serial No.*/
    "deviceType": ""
/*optional, string, peripheral model*/
  }
}
```

A.3.33 JSON_CheckResult

Message about the detection result parameters in JSON format.

```
{
  "CheckResult":{
    "checkResult": "",
/*optional, string, detection result: "normal", "abnormal", "unknown",
"processing"*/
    "Value": ,
/*optional, integer, detection result which is displayed in the format of a
number*/
    "VideoSignal": "",
/*optional, string, video signal status: "normal", "abnormal", "unknown"*/
  }
```

```
"AudioInput":"","  
/*optional, string, audio input status: "normal", "abnormal", "unknown"*/  
"AudioOutput":"","  
/*optional, string, audio output status: "normal", "abnormal", "unknown"*/  
}  
}
```

A.3.34 JSON_Cloud

Cloud message in JSON format

```
{  
  "Cloud":{  
    "alarmTamperEnabled": ,  
    /*optional, boolean type, whether to enable alarm and tampering event  
notification: "true"-yes, "false"-no*/  
    "lifeSecurityEnabled": ,  
    /*optional, boolean type, whether to enable life security event notification:  
"true"-yes, "false"-no*/  
    "systemStatusEnabled": ,  
    /*optional, boolean type, whether to enable system status event notification:  
"true"-yes, "false"-no*/  
    "operateEventEnabled": ,  
    /*optional, boolean type, whether to enable operation event notification:  
"true"-yes, "false"-no*/  
    "zoneAlarmTamperEnabled": ,  
    /*optional, boolean, whether to enable alarm and tampering event notification  
of the supported zone: "true"-yes, "false"-no*/  
    "exDevTamperEventEnabled": ,  
    /*optional, boolean, whether to enable peripheral tampering alarm notification:  
"true"-yes, "false"-no*/  
    "hostTamperEventEnabled": ,  
    /*optional, boolean, whether to enable tampering alarm notification of security  
control panel: "true"-yes, "false"-no*/  
    "emergencyEventEnabled": ,  
    /*optional, boolean, whether to enable panic alarm notification: "true"-yes,  
"false"-no*/  
    "medicalEventEnabled": ,  
    /*optional, boolean, whether to enable medical alarm notification: "true"-yes,  
"false"-no*/  
    "gasEventEnabled": ,  
    /*optional, boolean, whether to enable gas alarm notification: "true"-yes,  
"false"-no*/  
    "fireEventEnabled": ,  
    /*optional, boolean, whether to enable fire alarm notification: "true"-yes,  
"false"-no*/  
    "hostStatusEventEnabled": ,  
    /*optional, boolean, whether to enable notification of security control panel  
status: "true"-yes, "false"-no*/  
    "exDevStatusEventEnabled": ,  
    /*optional, boolean, whether to enable peripheral status notification: "true"-
```

```
yes, "false"-no*/
    "detectorStatusEventEnabled": ,
/*optional, boolean, whether to enable detector status notification: "true"-
yes, "false"-no*/
    "intelligentAlarmEnable":
/*optional, boolean, whether to enable smart alarm notification: "true"-yes,
"false"-no*/
    }
}
```

A.3.35 JSON_CloudCap

CloudCap message in JSON format

```
{
  "CloudCap":{
    "alarmTamperEnabled":true,
/*optional, boolean type, whether to enable alarm and tampering event
notification: "true"-yes, "false"-yes*/
    "lifeSecurityEnabled":true,
/*optional, boolean type, whether to enable life security event notification:
"true"-yes, "false"-yes*/
    "systemStatusEnabled":true,
/*optional, boolean type, whether to enable system status event notification:
"true"-yes, "false"-yes*/
    "operateEventEnabled":true,
/*optional, boolean type, whether to enable operation event notification:
"true"-yes, "false"-yes*/
    "zoneAlarmTamperEnabled":true,
/*optional, boolean, whether to enable alarm and tampering event notification
of the supported zone: "true"-yes, "false"-no*/
    "exDevTamperEventEnabled":true,
/*optional, boolean, whether to enable peripheral tampering alarm notification:
"true"-yes, "false"-no*/
    "hostTamperEventEnabled":true,
/*optional, boolean, whether to enable tampering alarm notification of security
control panel: "true"-yes, "false"-no*/
    "emergencyEventEnabled":true,
/*optional, boolean, whether to enable panic alarm notification: "true"-yes,
"false"-no*/
    "medicalEventEnabled":true,
/*optional, boolean, whether to enable medical alarm notification: "true"-yes,
"false"-no*/
    "gasEventEnabled":true,
/*optional, boolean, whether to enable gas alarm notification: "true"-yes,
"false"-no*/
    "fireEventEnabled":true,
/*optional, boolean, whether to enable fire alarm notification: "true"-yes,
"false"-no*/
    "hostStatusEventEnabled":true,
/*optional, boolean, whether to enable notification of security control panel
```

```
status: "true"-yes, "false"-no*/
    "exDevStatusEventEnabled":true,
/*optional, boolean, whether to enable peripheral status notification: "true"-
yes, "false"-no*/
    "detectorStatusEventEnabled":true,
/*optional, boolean, whether to enable detector status notification: "true"-
yes, "false"-no*/
    "intelligentAlarmEnable":true
/*optional, boolean, whether to enable smart alarm notification: "true"-yes,
"false"-no*/
    }
}
```

A.3.36 JSON_CommuniStatus

JSON message about communication status

```
{
  "CommuniStatus":{
    "wired":"","
/*optional, string type, wired network connection status, "normal"-connected,
"break"-disconnected, "ipConflict"-IP address conflicted*/
    "wifi":"","
/*optional, string type, wireless network connection status, "normal"-
connected, "break"-disconnected*/
    "wifiSignal": ,
/*optional, integer type, Wi-Fi strength, level options: 0, 1, 2, 3, 4*/
    "mobile":"","
/*optional, string type, mobile network connection status (GPRS/3G/4G),
"normal"-connected, "break"-disconnected*/
    "mobileSignal": ,
/*optional, string type, mobile network signal strength, level options: 0, 1,
2, 3, 4*/
    "flow": ,
/*optional, float type, network traffic flow usage, unit: MB*/
    "monFlowLimit": ,
/*optional, integer type, network traffic flow limit in current month, unit:
MB, read only*/
    "cloud":"","
/*optional, string type, cloud connection status, "normal"-connected, "break"-
disconnected*/
    "wifiName": "test",
/*optional, string, Wi-Fi name*/
    "SIMNum": "test",
/*optional, string, SIM card No.*/
    "SIMOperatorName": "test",
/*optional, string, SIM card operator, the maximum length is 64 bytes*/
    "R3AverageNoise": 1,
/*optional, int, R3 average noise value, unit: dBm*/
    "RXAverageNoise": 2
/*optional, int, RX average noise value, unit: dBm*/
```

```
}  
}
```

A.3.37 JSON_CurtainInfraredDetector

Message about the parameters of the curtain PIR (Passive Infrared) detector of a specific zone in JSON format.

```
{  
  "CurtainInfraredDetector": {  
    "LEDEnabled": ,  
    /*optional, boolean, whether to enable the LED indicator*/  
    "LEDLatchTime": ,  
    /*optional, int, delay time of the LED indicator, unit: second*/  
    "heartBeatInterval": ,  
    /*optional, int, heartbeat interval of the security control panel and the  
    peripheral, unit: second*/  
    "alwaysActiveEnabled": true,  
    /*optional, object, whether to keep the detector enabled when the zone is  
    disarmed*/  
    "triggerNumLimited": 5,  
    /*optional, object, alarm triggering times*/  
    "curtainDetectorType": "CurtainInfrared",  
    /*optional, object, curtain detector type: "CurtainInfrared" (IR curtain  
    detector), "R3PIRCurtain" (R3 PIR curtain detector), "R3DTAMCurtain" (R3 DTAM  
    curtain detector)*/  
    "directionIdentification": "LeftToRight",  
    /*optional, object, direction identification: "LeftToRight", "RightToLeft",  
    "Off". By default, it is "Off"*/  
    "microwaveSensitivity": 5,  
    /*optional, object, microwave sensitivity: 5, 8, 10. Unit: meter, by default,  
    it is 10 meters*/  
    "antiMaskingEnabled": true,  
    /*optional, object, whether to enable anti-masking, by default, it is enabled*/  
    "AMPulseInterval": 5  
    /*optional, object, AM pulse interval: 5, 30, 60, 120. Unit: second*/  
  }  
}
```

A.3.38 JSON_CurtainInfraredDetectorCap

JSON message about the configuration capability of the curtain PIR (Passive Infrared) detector

```
{  
  "CurtainInfraredDetectorCap": {  
    "zoneNo": {  
    /*optional, int, values that can be configured as the zone No.*/  
      "@opt": [1,3,5]  
    },  
  },  
}
```

```

    "supportZoneType":{
/*optional, string, zone types supported by the peripheral: "Instant"-instant
zone, "Delay"-delay zone, "Follow"-follow zone, "Perimeter"-perimeter zone,
"24hNoSound"-24-hour silent zone, "Emergency"-panic zone, "Fire"-fire zone,
"Gas"-gas zone, "Medical"-medical zone, "Timeout"-timeout zone, "Non-Alarm"-
disabled zone, "Key"-key zone, "24hSound"-24-hour annunciating zone. When
switching zone type, zone types supported by the peripheral can be obtained*/
    "@opt":["Instant","Delay","Follow"]
    },
    "alwaysActiveEnabled": {
/*optional, object, whether to keep the detector enabled when the zone is
disarmed*/
    "@opt": [true, false]
    },
    "heartBeatInterval":{
/*optional, int, heartbeat interval of the security control panel and the
peripheral, unit: second*/
    "@opt": [5,10,20,30]
    },
    "LEDEnabled": {
/*optional, boolean, whether to enable the LED indicator*/
    "@opt": [true,false]
    },
    "LEDLatchTime": {
/*optional, int, delay time of the LED indicator, unit: second*/
    "@min":0,
    "@max":0
    },
    "triggerNumLimited": {
/*optional, object, alarm triggering times*/
    "@min": 0,
    "@max": 10
    },
    "isSupportSignalTest": true,
/*optional, read-only, boolean, whether it supports signal strength detection,
if this node is not returned or if the value is false, it indicates that this
function is not supported*/
    "isSupportZoneTest": true,
/*optional, boolean, whether zone test is supported, if this node is not
returned or if the value is false, it indicates that this function is not
supported*/
    "isSupportFindMe": true,
/*optional, boolean, whether it supports FindMe detection, if this node is not
returned or if the value is false, it indicates that this function is not
supported*/
    "curtainDetectorType": {
/*optional, object, curtain detector type*/
    "@opt": ["CurtainInfrared", "R3PIRCurtain", "R3DTAMCurtain"]
    },
    "R3PIRCurtainNode": {
/*optional, object, node for R3 PIR curtain detector*/
    "directionIdentification": {

```



```
/*optional, object, direction identification, by default, it is "Off"*/
  "@opt": ["LeftToRight", "RightToLeft", "Off"]
},
"antiMaskingEnabled": {
/*optional, object, whether to enable anti-masking, by default, it is enabled*/
  "@opt": [true, false]
},
"AMPulseInterval": {
/*optional, object, AM pulse interval, unit: second*/
  "@opt": [5, 30, 60, 120]
}
},
"R3DTAMCurtainNode": {
/*optional, object, R3 DTAM curtain detector*/
  "directionIdentification": {
/*optional, object, direction identification, by default, it is "Off"*/
    "@opt": ["LeftToRight", "RightToLeft", "Off"]
  },
  "microwaveSensitivity": {
/*optional, object, microwave sensitivity, unit: meter, by default, it is 10
meters*/
    "@opt": [5, 8, 10]
  },
  "antiMaskingEnabled": {
/*optional, object, whether to enable anti-masking, by default, it is enabled*/
    "@opt": [true, false]
  },
  "AMPulseInterval": {
/*optional, object, AM pulse interval, unit: second*/
    "@opt": [5, 30, 60, 120]
  }
}
}
}
```

A.3.39 JSON_DeviceTime

DeviceTime message in JSON format

```
{
  "DeviceTime":{
    "perimeterDelayTime": ,
/*optional, delay time of perimeter alarm, unit: second. Alarms in the
perimeter zone will trigger siren output if not disarming or clearing alarms
after the delay time*/
    "sounderTime":
/*optional, alarm duration, which is the alarm sound duration of the alarm
linkage system, unit: second*/
  }
}
```

A.3.40 JSON_DeviceTimeCap

DeviceTimeCap capability message in JSON format

```
{
  "DeviceTimeCap":{
    "perimeterDelayTime":{
/*optional, delay time of perimeter alarm, unit: second*/
      "@min":5,
      "@max":600
    },
    "sounderTime":{
/*optional, alarm sound duration of the alarm linkage system, unit: second*/
      "@min":5,
      "@max":600
    }
  }
}
```

A.3.41 JSON_Direct

Direct message in JSON format

```
{
  "Direct":{
    "alarmTamperEnabled": ,
/*optional, boolean, whether to enable alarm and tampering event notification:
"true"-yes, "false"-no*/
    "lifeSecurityEnabled": ,
/*optional, boolean, whether to enable life security event notification: "true"-
yes, "false"-no*/
    "systemStatusEnabled": ,
/*optional, boolean, whether to enable system status event notification: "true"-
yes, "false"-no*/
    "operateEventEnabled": ,
/*optional, boolean, whether to enable operation event notification: "true"-
yes, "false"-no*/
    "zoneAlarmTamperEnabled": ,
/*optional, boolean, whether to enable alarm and tampering event notification
of the supported zone: "true"-yes, "false"-no*/
    "exDevTamperEventEnabled": ,
/*optional, boolean, whether to enable peripheral tampering alarm notification:
"true"-yes, "false"-no*/
    "hostTamperEventEnabled": ,
/*optional, boolean, whether to enable tampering alarm notification of security
control panel: "true"-yes, "false"-no*/
    "emergencyEventEnabled": ,
/*optional, boolean, whether to enable panic alarm notification: "true"-yes,
"false"-no*/
  }
}
```

```
    "medicalEventEnabled": ,
/*optional, boolean, whether to enable medical alarm notification: "true"-yes,
"false"-no*/
    "gasEventEnabled": ,
/*optional, boolean, whether to enable gas alarm notification: "true"-yes,
"false"-no*/
    "fireEventEnabled": ,
/*optional, boolean, whether to enable fire alarm notification: "true"-yes,
"false"-no*/
    "hostStatusEventEnabled": ,
/*optional, boolean, whether to enable notification of security control panel
status: "true"-yes, "false"-no*/
    "exDevStatusEventEnabled": ,
/*optional, boolean, whether to enable peripheral status notification: "true"-
yes, "false"-no*/
    "detectorStatusEventEnabled": ,
/*optional, boolean, whether to enable detector status notification: "true"-
yes, "false"-no*/
    "intelligentAlarmEnable":
/*optional, boolean, whether to enable smart alarm notification: "true"-yes,
"false"-no*/
    }
}
```

A.3.42 JSON_DirectCap

DirectCap capability message in JSON format

```
{
  "DirectCap":{
    "alarmTamperEnabled":true,
/*optional, boolean, whether to enable alarm and tampering event notification:
"true"-yes, "false"-no*/
    "lifeSecurityEnabled":true,
/*optional, boolean, whether to enable life security event notification: "true"-
yes, "false"-no*/
    "systemStatusEnabled":true,
/*optional, boolean, whether to enable system status event notification: "true"-
yes, "false"-no*/
    "operateEventEnabled":true,
/*optional, boolean, whether to enable operation event notification: "true"-
yes, "false"-no*/
    "zoneAlarmTamperEnabled":true,
/*optional, boolean, whether to enable alarm and tampering event notification
of the supported zone: "true"-yes, "false"-no*/
    "exDevTamperEventEnabled":true,
/*optional, boolean, whether to enable peripheral tampering alarm notification:
"true"-yes, "false"-no*/
    "hostTamperEventEnabled":true,
/*optional, boolean, whether to enable tampering alarm notification of security
control panel: "true"-yes, "false"-no*/
  }
```

```
"emergencyEventEnabled":true,
/*optional, boolean, whether to enable panic alarm notification: "true"-yes,
"false"-no*/
"medicalEventEnabled":true,
/*optional, boolean, whether to enable medical alarm notification: "true"-yes,
"false"-no*/
"gasEventEnabled":true,
/*optional, boolean, whether to enable gas alarm notification: "true"-yes,
"false"-no*/
"fireEventEnabled":true,
/*optional, boolean, whether to enable fire alarm notification: "true"-yes,
"false"-no*/
"hostStatusEventEnabled":true,
/*optional, boolean, whether to enable notification of security control panel
status: "true"-yes, "false"-no*/
"exDevStatusEventEnabled":true,
/*optional, boolean, whether to enable peripheral status notification: "true"-
yes, "false"-no*/
"detectorStatusEventEnabled":true,
/*optional, boolean, whether to enable detector status notification: "true"-
yes, "false"-no*/
"intelligentAlarmEnable":true
/*optional, boolean, whether to enable smart alarm notification: "true"-yes,
"false"-no*/
}
}
```

A.3.43 JSON_EventNotificationAlert_SecurityCPAlarmEventMsg

The security control panel's alarm or event details are uploaded in JSON format of EventNotificationAlert message, here shows an example.

```
{
  "ipAddress":"","
  "ipv6Address":"","
  "portNo": ,
  "protocol":"","
  "macAddress":"","
  "channelID":"","
  "dateTime":"","
  "activePostCount": ,
  "eventType":"cidEvent",
  "eventState":"","
  "eventDescription":"","
  "deviceID":"","
  "uuid":"","
  "recheck": ,
  "videoURL":"","
  "imageURL":"","
  "CIDEvent":{
```

```

"code": ,
"standardCIDcode": ,
"name": "",
"type": "",
"trigger": "",
"upload": "",
"centerAccount": "",
"keypad": ,
"system": ,
"zone": ,
"zoneCompatible": ,
"alarmCenterNo": ,
"repeater": ,
"siren": ,
"outputModule": ,
"extensionModule": ,
"ipcChannel": ,
"CameraList": [{
  "id": ,
  "ip": "",
  "port": ,
  "channel":
}],
"cardReader": ,
"cardNo": "",
"cardType": "",
"NVRLList": [{
  "id": ,
  "ip": "",
  "port": ,
  "channel":
}],
"AlarmLineRule": {
  "id": ,
  "alarmLineNo1": ,
  "alarmLineNo2": ,
  "ruleType":
},
"alarmLineNo": ,
"AlarmRule": {
  "alarmLineNo1": ,
  "alarmLineNo2": ,
  "ruleType":
},
"remoteCtrlNo": .
"userName": ""
}
}

```

Node Description

uuid

String type, a unique ID to identify an event, standard format: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxx (e.g., 8-4-4-4-12).

recheck

Optional, integer type, mark of whether to report the alarm twice: 1-report the alarm twice, 0 or this node does not exist-normal alarm.

videoURL

Optional, string, video URL, this node is valid when **recheck** is 1. This URL will be uploaded when the alarm is reported for the second time.

imageURL

Optional, string, picture URL, this node is valid when **recheck** is 1. This URL will be uploaded when the alarm is reported for the second time.

CIDEvent

CID alarm events.

code

Required, integer type, event No.

standardCIDcode

Optional, integer type, standard CID code.

name

Optional, string type, event name.

type

Required, string type, event types: "alarmAndDismantle"-alarm&tampering event, "lifeSafety"-lidi security event, "sysStatus"-system status event, "armAndDisarm"-arming or disarming event, "zoneAlarm"-zone alarm, "operateAlarm"-operation alarm, "dismantleAlarm"-tampering alarm, "motionAlarm"-motion detection alarm, "hideAlarm"-device blocking alarm, "exceptionAlarm"-exception alarm, "earlyWarningAlarm"-early-warning zone alarm, "overLineAlarm"-cross-line alarm, "zoneAlarmTamper"-alarm and tampering event of the supported zone, "exDevTamperEvent"-peripheral tampering alarm, "hostTamperEvent"-tampering alarm of security control panel, "emergencyEvent"-panic alarm, "medicalEvent"-medical alarm, "gasEvent"-gas alarm, "fireEvent"-fire alarm, "hostStatusEvent"-security control panel status, "exDevStatusEvent"-peripheral status, "detectorStatusEvent"-detector status, "intelligentAlarmEvent"-smart alarm (smart alarms refer to alarms triggered by network cameras no matter whether they are analyzed by detectors or streaming algorithms).

trigger

Required, string type, event occurred time in ISO8601 time format.

upload

Required, string type, event uploaded time in ISO8601 time format.

centerAccount

Optional, string type, center account, which is used to mark the device.

keypad

Optional, keypad No.

system

Optional, partition No.

zone

Optional, zone No.

zoneCompatible

Optional, boolean, whether to support zone compatibility. This field is only supported by Axiom hub devices to solve the problem of device compatibility. If this field is returned, the zone No. (**zone**) will start from 0 (new devices); if this field is not returned, the zone No. (**zone**) will start from 1 (old devices). For devices except Axiom hub devices, this field will not be returned, and the zone No. (**zone**) will start from 0.

alarmCenterNo

Optional, integer, alarm receiving center No.

repeater

Optional, repeater No.

siren

Optional, siren No.

outputModule

Optional, output module No.

extensionModule

Optional, extended module No.

ipcChannel

Optional, channel No. of the network camera added to the security control panel.

CameraList

Optional, network camera information.

id

Optional, integer type, network camera No.

ip

Optional, string type, network camera IP address.

port

Optional, integer type, network camera port number.

channel

Optional, integer type, channel No. of network camera.

cardReader

Optional, integer type, card reader No.

cardNo

Optional, string type, card No.

cardType

Optional, string type, card type: "operateCard"-operation card, "patrolCard"-patrol card.

NVRList

Optional, NVR (Network Video Recorder) information.

id

Optional, integer type, NVR No.

ip

Optional, string type, NVR address.

port

Optional, integer type, NVR port No.

channel

Optional, integer type, NVR channel No.

AlarmLineRule

Alarm rule.

id

Optional, integer type, alarm rule No.

alarmLineNo1

Optional, integer type, No. of trigger line 1.

alarmLineNo2

Optional, integer type, No. of the trigger line linked to **alarmLineNo1**.

ruleType

Optional, integer type, alarm rule: 1-from **alarmLineNo1** to **alarmLineNo2**, 2-from **alarmLineNo2** to **alarmLineNo1**, 3-from **alarmLineNo1** to **alarmLineNo2** or from **alarmLineNo2** to **alarmLineNo1**. If the No. of **alarmLineNo1** and **alarmLineNo2** is the same one, it indicates configuring a single trigger line: 1-from left to right, 2-from right to left, 3-bidirectional.

alarmLineNo

Optional, integer32 type, trigger line No.

AlarmRule

Alarm rule.

alarmLineNo1

Optional, integer, trigger line No. of **alarmLineNo1**.

alarmLineNo2

Optional, integer, No. of the trigger line linked to **alarmLineNo1**.

ruleType

Optional, integer, direction of crossing trigger lines that can trigger alarms: 1-**alarmLineNo1** to **alarmLineNo2**, 2-**alarmLineNo2** to **alarmLineNo1**, 3-bidirectional.

remoteCtrlNo

Optional, int, keyfob No.

userName

Optional, string, user name. This node indicates the user name used to log in to the device when the alarm is uploaded.

A.3.44 JSON_EventNotificationAlert_Alarm/EventInfo

EventNotificationAlert message with alarm or event information in JSON format.

```
{
  "ipAddress": "",
  /*required, device IPv4 address , string, the maximum size is 32 bytes*/
  "ipv6Address": "",
  /*optional, device IPv6 address, string, the maximum size is 128 bytes*/
  "portNo": ,
  /*optional, device port No., integer32*/
  "protocol": "",
  /*optional, protocol type, "HTTP, HTTPS", string, the maximum size is 32 bytes*/
  "macAddress": "",
  /*optional, MAC address, string, the maximum size is 32 bytes, e.g.,
  01:17:24:45:D9:F4*/
  "channelID": "",
  /*optional, device channel No., integer32*/
  "dateTime": "",
  /*optional, string, alarm/event triggered or occurred time based on ISO8601,
  the maximum size is 32 bytes, e.g., 2009-11-14T15:27Z*/
  "activePostCount": "",
  /*required, alarm/event frequency, integer32*/
  "eventType": "",
  /*required, alarm/event type, "captureResult, faceCapture,...", string, the
  maximum size is 128 bytes*/
  "eventState": "",
  /*required, string, the maximum size is 32 bytes, durative alarm/event status:
  "active"-valid, "inactive"-invalid*/
  "eventDescription": "",
  /*required, event description, string, the maximum size is 128 bytes*/
}
```

```
"deviceId":"","  
/*string type, device ID*/  
"uuid":"","  
/*string type, event UUID, which is used to uniquely identify an event, the  
standard UUID format is xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx*/  
...  
/*optional, for different alarm/event types, the nodes are different, see the  
message examples in different applications*/  
}
```

A.3.45 JSON_EventRecord

EventRecord message in JSON format

```
{  
  "EventRecord":{  
    "id": ,  
/*required, integer type, channel ID, e.g., 101, 102*/  
    "frontRecordTime": ,  
/*optional, integer type, pre-recording time range when alarm is triggered or  
event occurred, unit: second*/  
    "afterRecordTime":  
/*optional, integer type, post-recording time range when alarm is triggered or  
event occurred, unit: second*/  
  }  
}
```

A.3.46 JSON_EventRecordCap

EventRecordCap capability message in JSON format

```
{  
  "EventRecordCap":{  
    "frontRecordTime":{  
/*optional, pre-recording time range when alarm is triggered or event occurred,  
unit: second*/  
      "@min": ,  
      "@max":  
    },  
    "afterRecordTime":{  
/*optional, post-recording time range when alarm is triggered or event  
occurred, unit: second*/  
      "@min":,  
      "@max":  
    },  
    "recordTime":{  
/*supported recording time type: 1 (pre-record for 5 seconds and post-record  
for 2 seconds), 2 (pre-record for 2 seconds and post-record for 5 seconds)*/  
      "@opt":[1,2]  
    }  
  }  
}
```

```
    },
    "method":{
/*optional, methods supported by the function: "get", "put"-edit*/
    "@opt":["get","put"]
    }
  }
}
```

A.3.47 JSON_ExDevStatus

JSON message about peripheral status

```
{
  "ExDevStatus":{
    "OutputModList":[{
/*optional*/
    "OutputMod":{
      "id": ,
/*required, int, wireless output module No.*/
      "seq": "",
/*required, string, peripheral serial No.*/
      "status": "",
/*optional, string, wireless output module status, "notRelated"-no wireless
output is linked, "online", "offline", "heartbeatAbnormal"-heartbeat exception*/
      "tamperEvident": ,
/*optional, boolean, zone tampering alarm status, true-triggered, false-not
triggered*/
      "charge": "",
/*optional, string, battery status, "normal", "lowPower"-low battery*/
      "signal":
/*optional, int, signal strength, which ranges from 0 to 255*/
      "model": "DS-PM1-O8-WE",
/*optional, string, model: "DS-PM1-O8-WE" wireless output module with 8
channels, "DS-PM1-O2-WE" wireless output module with 2 channels*/
      "temperature": 1,
/*optional, int, temperature*/
      "isViaRepeater": true,
/*optional, boolean, whether the signal is forwarded via repeater*/
      "repeaterName": "test",
/*optional, string, repeater name, this node is valid when the value of
"isViaRepeater" is true*/
      "voltValue": 1,
/*optional, int, voltage value, unit: V*/
      "currentValue": 1,
/*optional, int, current value, unit: mA*/
      "powerLoad": 1,
/*optional, int, power load, unit: W*/
      "energySumVaule": 1,
/*optional, int, power consumption, unit: Wh*/
      "relayList": [{
/*optional, array, list of relays*/
```

```

        "id": 1,
/*required, int, input ID*/
        "status": "on",
/*optional, string, relay status, "on", "off"*/
        "name": "test",
/*optional, string, relay name*/
        "subSystem": [1, 2, 3],
/*optional, array, related partition*/
        "scenarioType": ["alarm"]
/*optional, array, scenario type*/
    }],
    "voltValueV20": 1.000
/*optional, float, voltage value (version 2.0) which is accurate to 3 decimal
places, unit: V, this node has higher priority over voltValue*/
    }
}
    "OutputList": [{
/*optional*/
        "Output": {
            "id": ,
/*required, int, relay No.*/
            "name": "",
/*optional, string, relay name*/
            "status": "",
/*optional, string, relay status, "notRelated"-no relay is linked, "on", "off",
"offline", "heartbeatAbnormal"-heartbeat exception*/
            "tamperEvident": ,
/*optional, boolean, zone tampering status: "true"-tampered, "false"-not
tampered*/
            "charge": "",
/*optional, string, battery status: "normal", "lowPower"-low battery*/
            "linkage": "",
/*optional, string, relay linked event types: "alarm", "arming", "disarming",
"manualCtrl"-manual control*/
            "signal": ,
/*optional, integer, signal strength, which ranges from 0 to 255*/
            "temperature": 1,
/*optional, int, temperature*/
            "version": "test",
/*optional, string, version No, the maximum length is 32 bytes*/
            "accessModuleType": "transmitter",
/*optional, string, access module type: "transmitter", "localTransmitter",
"multiTransmitter", "localRelay", "keypad"*/
            "relatedAccessModuleID": 1,
/*optional, int, linked access module ID*/
            "address": 254,
/*optional, int, wired (extended) module address, this node works with
accessModuleType*/
            "subSystemList": [1, 2, 3],
/*optional, array, list of linked partitions*/
            "scenarioType": ["alarm"],
/*optional, array, scenario type*/

```

```
        "relayAttrib": "wired",
/*optional, string, relay attribute: "wired", "wireless" (default)*/
        "deviceNo": 1
/*optional, int, device No., range:[1,1000]*/
    },
    ],
    "SirenList":[{
/*optional*/
        "Siren":{
            "id": ,
/*required, int, siren No.*/
            "seq": "",
/*required, string, peripheral serial No.*/
            "name": "",
/*optional, string, siren name*/
            "status": "",
/*optional, string, siren status, "notRelated"-no siren is linked, "online",
"offline", "heartbeatAbnormal"-heartbeat exception*/
            "tamperEvident": ,
/*optional, boolean, zone tampering alarm status, true-triggered, false-not
triggered*/
            "sirenAttrib": "",
/*string, siren attribute: "wired", "wireless"*/
            "charge": "",
/*optional, string, battery status, "normal", "lowPower"-low battery*/
            "signal": ,
/*optional, int, signal strength, which ranges from 0 to 255*/
            "deviceNo": 1,
/*optional, int, device No., range:[1,1000]. After installation, the installer
will upload the device No. and the corresponding peripheral/detector
information to the ARC for device type recognition*/
            "mainPowerSupply": true
/*optional, boolean, main power supply (external power supply is supported for
wireless siren): true, false*/
        }
    },
    "RepeaterList":[{
/*optional*/
        "Repeater":{
            "id": ,
/*required, int, repeater No.*/
            "seq": "",
/*required, string, peripheral serial No.*/
            "name": "",
/*optional, string, repeater name*/
            "status": "",
/*optional, string, repeater status, "notRelated"-no repeater is linked,
"online", "offline", "heartbeatAbnormal"-heartbeat exception*/
            "tamperEvident": ,
/*optional, boolean, zone tampering alarm status, true-triggered, false-not
triggered*/
            "charge": "",
```

```
/*optional, string, battery status, "normal", "lowPower"-low battery*/
    "signal":
/*optional, int, signal strength, which ranges from 0 to 255*/
    "chargeValue": 0,
/*optional, int, power value, which is between 0 to 100*/
    "model": "DS-PR1-WE",
/*optional, string, model: "DS-PR1-WE" (wireless repeater)*/
    "temperature": 0,
/*optional, int, temperature*/
    "connDevNum": 1,
/*optional, int, number of devices*/
    "mainPowerSupply": true,
/*optional, boolean, AC power supply status: true (connected), false
(disconnected)*/
    "batteryStatus": "normal",
/*optional, string, battery status: "normal", "miss"*/
    "version": "test" ,
/*optional, string, version No.*/
    "deviceNo": 1
/*optional, int, device No., range:[1,1000]. After installation, the installer
will upload the device No. and the corresponding peripheral/detector
information to the ARC for device type recognition*/
    }},
    "CardReaderList": [{
/*optional, card reader list*/
        "CardReader": {
            "id": ,
/*required, int, card reader No.*/
            "seq": "",
/*required, string, peripheral serial No.*/
            "name": "",
/*optional, string, card reader name*/
            "status": "",
/*optional, string, card reader status: "notRelated"-not linked, "online",
"offline", "heartbeatAbnormal"-heartbeat exception*/
            "tamperEvident": ,
/*optional, boolean, tampering status: "true"-tampered, "false"-not tampered*/
            "charge": "",
/*optional, string, state of charge: "normal", "lowPower"-low battery*/
            "signal": ,
/*optional, int, signal strength, it is between 0 and 255*/
            "model": "DS-PT1-WE",
/*optional, enum, model, subType:string, "DS-PT1-WE" (wireless card reader)*/
            "temperature": 1,
/*optional, int, temperature*/
            "subSystemList": [1, 2, 3],
/*optional, array, linked partitions*/
            "isViaRepeater": true,
/*optional, boolean, whether the signal is forwarded via repeater*/
            "repeaterName": "test",
/*optional, string, repeater name, the maximum length is 64 bytes, this node is
valid when isViaRepeater is true*/
```

```
        "version": "test",
/*optional, string, version, the maximum length is 32 bytes*/
        "deviceNo": 1
/*optional, int, device No., range:[1,1000]. After installation, the installer
will upload the device No. and the corresponding peripheral/detector
information to the ARC for device type recognition*/
    },
    ],
    "ExtensionList":[{
/*optional, extension module list*/
        "ExtensionModule":{
            "id": ,
/*required, int, extension module No.*/
            "name": "",
/*optional, string, extension module name*/
            "address": ,
/*optional, int, module address, this node is returned by wired modules*/
            "linkageAddress": ,
/*optional, int, linked module address, this node is returned by wireless
modules*/
            "type": "",
/*optional, string, module type: "wiredZone"-wired zone module, "wiredOutput"-
wired output module, "wirelessOutput"-wireless output module, "wirelessRecv"-
wireless receiver module (wired module)*/
            "status": "",
/*optional, string, keypad status: "online", "offline", "heartbeatAbnormal"-
heartbeat exception*/
            "tamperEvident": ,
/*optional, boolean, tampering status: "true"-tampered, "false"-not tampered*/
            "moduleAttrib": "",
/*string, module attribute: "wired", "wireless"*/
            "charge": ""
/*optional, string, state of charge: "normal", "lowPower"-low battery*/
        }
    },
    "KeypadList":[{
/*optional, keypad list*/
        "Keypad":{
            "id": ,
/*required, int, keypad No.*/
            "seq": "",
/*required, string, peripheral serial No.*/
            "name": "",
/*optional, string, keypad name*/
            "status": "",
/*optional, string, keypad status: "notRelated"-not linked, "online",
"offline", "heartbeatAbnormal"-heartbeat exception*/
            "tamperEvident": ,
/*optional, boolean, tampering status: "true"-tampered, "false"-not tampered*/
            "keypadAttrib": "",
/*string, keypad attribute: "wired", "wireless"*/
            "charge": "",
```

```
/*optional, string, state of charge: "normal", "lowPower"-low battery*/
    "signal": ,
/*optional, int, signal strength, it is between 0 and 255*/
    "address":
/*optional, int, keypad address, this node is only returned by wired keypads*/
    "model": "DS-PK1-E-WE",
/*optional, string, model: "DS-PK1-E-WE" (wireless LED keypad)*/
    "temperature": 1,
/*optional, int, temperature*/
    "subSystemList": [1, 2, 3],
/*optional, array, list of linked partitions*/
    "isViaRepeater": true,
/*optional, boolean, whether the signal is forwarded via repeater*/
    "repeaterName": "test",
/*optional, string, repeater name, this node is valid when the value of
isViaRepeater is true*/
    "version": "test" ,
/*optional, string, version No.*/
    "smokeDetectorAlarm": "alarm",
/*optional, string, whether the alarm of smoke detector is triggered: "normal",
"alarm"*/
    "smokeDetectorPowerSupply": "normal",
/*optional, string, status of power supply for the smoke detector: "normal",
"shorted"*/
    "powerSupply": "normal",
/*optional, string, the status of 12-volt power supply: "normal", "shorted"*/
    "mainPowerSupply": true,
/*optional, boolean, external power supply status: true (connected), false
(disconnected)*/
    "type": "transmitter",
/*optional, transmitter type: "transmitter", "multiTransmitter"*/
    "deviceNo": 1
/*optional, int, device No., range:[1,1000]*/
    }
}],
    "RemoteList": [
/*optional, array, list of remote controls*/
    {
        "Remote": {
/*optional, object, remote control*/
            "id": 1,
/*required, int, remote control ID, range:[1,64]*/
            "seq": "test",
/*required, string, remote control serial No., the maximum length is 32 bytes*/
            "name": "test",
/*optional, string, remote control name, the maximum length is 64 bytes*/
            "status": "notRelated",
/*optional, enum, remote control status: "notRelated", "online", "offline",
"heartbeatAbnormal"*/
            "charge": "normal",
/*optional, enum, power supply status: "normal", "lowPower"*/
            "chargeValue": 1,
```



```
/*optional, int, power supply value, range:[0,100]*/
    "model": "test",
/*optional, string, model, the maximum length is 32 bytes*/
    "isViaRepeater": true,
/*optional, boolean, whether to transmit signals via the repeater*/
    "repeaterName": "test",
/*optional, string, repeater name, this node is valid when isViaRepeater is
true*/
    "SelKeyList": [
/*optional, array, list of list of single keys*/
        {
            "SelKey": {
/*optional, object, custom function of the single key, this node is required
when SelKeyList exists*/
                "key": 1,
/*optional, int, remote control key ID, this node is required when SelKey
exists*/
                "func": "test",
/*optional, string, corresponding function, this node is required when SelKey
exists*/
                "outputNo": 1
/*optional, int, relay No., this node is required when func is
"operateOutputs"*/
            }
        },
        {
            "CombKeyList": [
/*optional, array, list of combination key*/
                {
                    "CombKey": {
/*optional, object, custom function of the combination key, this node is
required when CombKeyList exists*/
                        "keys": "test",
/*optional, string, remote control key ID, this node is required when CombKey
exists*/
                        "func": "test",
/*optional, string, corresponding function, this node is required when CombKey
exists*/
                        "outputNo": 1
/*optional, int, relay No., this node is required when func is
"operateOutputs"*/
                    }
                },
                {
                    "relatedNetUserName": "test",
/*optional, string, user name of the linked network, the maximum length is 64
bytes*/
                    "userNickName": "test",
/*optional, string, the user nickname of the linked network, the maximum length
is 64 bytes*/
                    "version": "test",
/*optional, string, version No., the maximum length is 32 bytes*/
```

```
        "deviceNo": 1
/*optional, int, device No., range:[1,1000]. After installation, the installer
will upload the device No. and the corresponding peripheral/detector
information to the ARC for device type recognition*/
    }
}
],
    "TransmitterList": [
/*required, array, list of transmitters*/
    {
        "Transmitter": {
/*required, object, transmitter*/
            "id": 1,
/*required, int, transmitter ID*/
            "name": "test",
/*optional, string, name, the maximum length is 32 bytes*/
            "subSystemList": [1, 2, 3],
/*optional, array, list of partitions linked to the transmitter*/
            "ZoneList": [
/*optional, array, list of zones linked to the transmitter*/
            {
                "zoneID": 1,
/*required, int, zone ID, range:[0,255]*/
                "detectorType": "other",
/*optional, string, detector type: "dualTechnologyPirDetector",
"tripleTechnologyPirDetector", "glassBreakDetector", "activeInfraredDetector",
"passiveInfraredDetector", "magneticContact", "panicButton",
"waterLeakDetector", "humidityDetector", "temperatureDetector",
"smokeDetector", "combustibleGasDetector", "vibrationDetector", "other"*/
                "isBypassed": true,
/*optional, boolean, whether to bypass the zone, true (bypassed), false (bypass
recovered)*/
                "subSystemList": [1, 2, 3],
/*optional, array, list of partitions linked to the zone*/
                "zoneType": "Instant",
/*optional, string, zone type: "Instant"-instant zone, "Delay"-delay zone,
"Follow"-follow zone, "Perimeter"-perimeter zone, "24hNoSound"-24-hour silent
zone, "Emergency"-panic zone, "Fire"-fire zone, "Gas"-gas zone, "Medical"-
medical zone, "Timeout"-timeout zone, "Non-Alarm"-disabled zone, "Key"-key
zone*/
                "tamperEvident": true,
/*optional, boolean, zone tampering alarm status, true (triggered), false (not
triggered)*/
                "enterDelay": 1,
/*optional, int, delay time (enter), unit: second, this node is valid for delay
zones*/
                "exitDelay": 5,
/*optional, int, delay time (exit), unit: second*/
                "alarm": true,
/*optional, boolean, whether the zone alarm is triggered, true (triggered),
false (not triggered)*/
                "magnetOpenStatus": true
            }
        }
    }
]
```

```
/*optional, boolean, whether the magnetic contact is open: true (open), false
(closed)*/
    }
  ],
  "OutputList": [
/*optional, array, list of linked relays*/
    {
      "outputID": 1,
/*optional, int, relay ID*/
      "subSystemList": [1, 2, 3],
/*optional, array, list of partitions linked to the relay*/
      "status": "on"
/*optional, enum, relay status: "on", "off"*/
    }
  ],
  "seq": "test",
/*required, string, serial No. of the peripheral*/
  "status": "online",
/*optional, enum, transmitter's status: "online", "offline", "notRelated",
"heartbeatAbnormal"*/
  "tamperEvident": true,
/*optional, boolean, tampering alarm status, true (triggered), false (not
triggered)*/
  "charge": "normal",
/*optional, enum, power status: "normal", "lowPower"*/
  "chargeValue": 1,
/*optional, int, power value, value: [0,100]*/
  "signal": 1,
/*optional, int, signal strength, value: [0,255]*/
  "model": "test",
/*optional, string, model*/
  "temperature": 1,
/*optional, int, temperature*/
  "isViaRepeater": true,
/*optional, boolean, whether to transmit signals via the repeater*/
  "repeaterName": "test",
/*optional, string, repeater name, this node is valid when isViaRepeater is
true*/
  "movedAlarmEnabled": true,
/*optional, boolean, whether to enable mobile alarms: true (yes), false (no)*/
  "tamperPortEnabled": true,
/*optional, object, whether to enable port tamper detection*/
  "voltageOutput": "open",
/*optional, string, whether to output the 3.3-volt voltage: "open"(yes),
"close" (no), "disarmedClose" (not to output the voltage when it is disarmed)*/
  "portCfg": "antiMasking",
/*optional, string, port configuration: "antiMasking", "output", "close"*/
  "version": "test",
/*optional, string, transmitter version, the maximum length is 32 bytes*/
  "smokeDetectorAlarm": "alarm",
/*optional, string, whether the alarm of smoke detector is triggered: "normal",
"alarm"*/
```

```
        "smokeDetectorPowerSupply": "normal",
/*optional, string, status of power supply for the smoke detector: "normal",
"shorted"*/
        "powerSupply": "normal",
/*optional, string, the status of 12-volt power supply: "normal", "shorted"*/
        "mainPowerSupply": true,
/*optional, boolean, external power supply status: true (connected), false
(disconnected)*/
        "type": "transmitter",
/*optional, transmitter type: "transmitter", "multiTransmitter"*/
        "deviceNo": 1
/*optional, int, device No., range:[1,1000]*/
    }
}
}]
}
}
```

A.3.48 JSON_ExtensionList

ExtensionList message in JSON format

```
{
  "ExtensionList":[{
/*optional, extension module list*/
    "ExtensionModule":{
        "id": ,
/*required, integer type, extension module No.*/
        "name": "",
/*optional, string type, extension module name*/
        "address": ,
/*optional, integer type, module address, this node is returned by wired
modules*/
        "linkageAddress": ,
/*optional, integer type, linked module address, this node is returned by
wireless modules*/
        "type": "",
/*optional, string type, module type: "wiredZone"-wired zone module,
"wiredOutput"-wired output module, "wirelessOutput"-wireless output module,
"wirelessRecv"-wireless receiver module (wired module)*/
        "status": "",
/*optional, string type, extension module status: "online", "offline",
"heartbeatAbnormal"-heartbeat exception*/
        "tamperEvident": ,
/*optional, boolean type, tampering status: "true"-tampered, "false"-not
tampered*/
        "moduleAttrib": "",
/*string type, module attribute: "wired", "wireless"*/
        "charge": ""
/*optional, string type, state of charge: "normal", "lowPower"-low battery*/
    }
  ]
}
```

```
    }  
}
```

A.3.49 JSON_ExtensionModule

ExtensionModule message in JSON format

```
{  
  "ExtensionModule":{  
    "id": ,  
    /*optional, integer type, module No.*/  
    "address": ,  
    /*optional, read-only, integer type, module address, this node is only returned  
    by wired modules*/  
    "linkageAddress": ,  
    /*optional, read-only, integer type, linked module address, this node is only  
    returned by wireless modules*/  
    "name": "",  
    /*optional, string type, module name*/  
    "type": "",  
    /*optional, read-only, string type, module type: "wiredZone"-wired zone module,  
    "wiredOutput"-wired output module, "wirelessOutput"-wireless output module,  
    "wirelessRecv"-wireless receiver module (wired module)*/  
    "detailType": "",  
    /*optional, read-only, string type, detailed module type: wired zone module:  
    "eightWiredZone"-8-channel wired zone; wired output module: "localWiredOutput"-  
    local relay, "fourWiredOutput"-4-channel wired relay, "eightWiredOutput"-8-  
    channel wired relay; wireless output module: "twoWirelessOutput"-2-channel  
    wireless relay, "eightWirelessOutput"-8-channel wireless relay; wireless  
    receiver module (wired module): "RS485WirelessRecv"-RS-485 wireless receiver*/  
    "model": "",  
    /*optional, read-only, string type, module model*/  
    "version": "",  
    /*optional, read-only, string type, module version*/  
    "related": ,  
    /*optional, boolean type, whether the extension module is linked to a physical  
    extension module*/  
    "seq": "",  
    /*optional, string type, module serial No., this node can only be configured by  
    wireless modules*/  
    "checkTime":  
    /*optional, integer type, offline duration, unit: hour, this node can only be  
    configured by wireless modules*/  
  }  
}
```

A.3.50 JSON_ExtensionModuleCap

ExtensionModuleCap capability message in JSON format

```
{
  "ExtensionModuleCap":{
    "id":{
      /*optional, module No.*/
      "@min": ,
      "@max":
    },
    "address":{
      /*optional, read-only, module address, this node is only returned by wired
modules*/
      "@opt":[1,2,3]
    },
    "linkageAddress":{
      /*optional, read-only, linked module address, this node is only returned by
wireless modules*/
      "@opt":[1,2,3]
    },
    "name":{
      /*optional, module name*/
      "@min": ,
      "@max":
    },
    "type":{
      /*optional, read-only, module type: "wiredZone"-wired zone module,
"wiredOutput"-wired output module, "wirelessOutput"-wireless output module,
"wirelessRecv"-wireless receiver module (wired module)*/
      "@opt":["wiredZone", "wiredOutput", "wirelessOutput", "wirelessRecv"]
    },
    "detailType":{
      /*optional, read-only, detailed module type: wired zone module:
"eightWiredZone"-8-channel wired zone; wired output module: "localWiredOutput"-
local relay, "fourWiredOutput"-4-channel wired relay, "eightWiredOutput"-8-
channel wired relay; wireless output module: "twoWirelessOutput"-2-channel
wireless relay, "eightWirelessOutput"-8-channel wireless relay; wireless
receiver module (wired module): "RS485WirelessRecv"-RS-485 wireless receiver*/
      "@opt":["eightWiredZone", "localWiredOutput", "fourWiredOutput",
"eightWiredOutput", "twoWirelessOutput", "eightWirelessOutput",
"RS485WirelessRecv"]
    },
    "model":{
      /*optional, read-only, module model*/
      "@min": ,
      "@max":
    },
    "version":{
      /*optional, read-only, module version*/
      "@min": ,
```

```
        "@max":
      },
      "related": "true,false",
      /*optional, whether the extension module is linked to a physical extension
      module*/
      "seq": {
      /*optional, module serial No., this node can only be configured by wireless
      modules*/
        "@min": ,
        "@max":
      },
      "checkTime": {
      /*optional, offline duration, unit: hour, this node can only be configured by
      wireless modules*/
        "@min": ,
        "@max":
      },
      "method": {
      /*required, methods supported by the function: "add", "put"-edit, "getAll"-get
      all*/
        "@opt": ["add", "put", "getAll"]
      }
    }
  }
}
```

A.3.51 JSON_FaultCheckParameter

JSON message about the fault detection parameters

```
{
  "FaultCheckParameter": {
    "systemfaultDetectEnabled": ,
    /*optional, boolean, whether to enable system fault report: "true"-yes, "false"-
    no. If this function is disabled, the system fault report will not be generated
    and notified*/
    "ipcDetectEnabled": ,
    /*optional, boolean, whether to enable network camera offline detection*/
    "batteryDetectionEnabled": ,
    /*optional, boolean, whether to enable security control panel storage battery
    detection*/
    "LANDetectionEnabled": ,
    /*optional, boolean, whether to enable wired network fault detection*/
    "WIFIDetectionEnabled": ,
    /*optional, boolean, whether to enable Wi-Fi network fault detection*/
    "cellularDetectionEnabled": ,
    /*optional, boolean, whether to enable cellular (mobile) network fault
    detection*/
    "SIMDetectionEnabled": ,
    /*optional, boolean, whether to enable SIM card fault detection*/
    "telLineBrokenEnabled": ,
    /*optional, boolean, whether to enable telephone line disconnection detection*/
  }
}
```

```
    "RS485AbnormalEnabled": ,
/*optional, boolean, whether to enable RS-485 exception detection*/
    "ACcheckTime": ,
/*optional, integer type, AC detection time*/
    "mainPowerDetectionEnabled": true,
/*optional, boolean, whether to enable main power detection*/
  }
}
```

A.3.52 JSON_FaultCheckParameterCap

JSON message about the configuration capability of fault detection

```
{
  "FaultCheckParameterCap":{
    "systemfaultDetectEnabled":"true,false",
/*optional, boolean type, whether to support enabling system fault report*/
    "ipcDetectEnabled":"true,false",
/*optional, boolean type, whether to support enabling network camera offline
detection*/
    "batteryDetectionEnabled":"true,false",
/*optional, boolean type, whether to enable security control panel storage
battery detection*/
    "LANDetectionEnabled":"true,false",
/*optional, boolean type, whether to enable wired network fault detection*/
    "WIFIDetectionEnabled":"true,false",
/*optional, boolean type, whether to enable Wi-Fi network fault detection*/
    "cellularDetectionEnabled":"true,false",
/*optional, boolean type, whether to enable cellular (mobile) network fault
detection*/
    "SIMDetectionEnabled":"true,false",
/*optional, boolean type, whether to enable SIM card fault detection*/
    "telLineBrokenEnabled":"true,false",
/*optional, boolean type, whether to enable telephone line disconnection
detection*/
    "RS485AbnormalEnabled":"true,false",
/*optional, boolean type, whether to enable RS-485 exception detection*/
    "ACcheckTime":{
/*optional, detection time when the AC is powered off, unit: second*/
      "@min":0,
      "@max":3600
    },
    "mainPowerDetectionEnabled": {
/*optional, object, whether to enable main power detection*/
      "@opt": [true, false]
    }
  }
}
```


A.3.53 JSON_FileExportCap

JSON message about the capability of exporting picture captured by pircam (detector equipped with camera)

```
{
  "FileExportCap": {
    "zoneNo": {
/*optional, zone No.*/
      "@opt": [0,1,2],
    }
  }
}
```

A.3.54 JSON_FileExportCond

JSON message about condition of exporting picture captured by pircam (detector equipped with camera)

```
{
  "FileExportCond":{
    "zoneNo":
/*optional, int, zone No., which starts from 0*/
  }
}
```

A.3.55 JSON_FileExportInfo

JSON message about URL of the exported picture captured by pircam (detector equipped with camera)

```
{
  "FileExportInfo":{
    "fileUrl":""
/*optional, string, URL of the exported picture*/
  }
}
```

A.3.56 JSON_GlassBreakDetector

JSON message about parameters of a composite PIR (Passive Infrared) glass-break detector

```
{
  "GlassBreakDetector":{
    "LEDEnabled": ,
  }
}
```

```
/*optional, boolean, whether to enable the LED indicator*/
    "LEDLatchTime": ,
/*optional, int, delay time of the LED indicator, unit: second*/
    "findMeEnabled": ,
/*optional, boolean, whether to enable the Fine Me function*/
    "sensitivityLevel": "",
/*optional, string, sensitivity level: "high", "auto"-automatic, "antiPet"-pet
immune*/
    "checkEnabled": ,
/*optional, boolean, whether to enable self-test: true=yes, false=no*/
    "distance": ,
/*optional, int, distance, unit: meter*/
    "alarmLogic": "",
/*optional, string, alarm logic: "and", "or"*/
    "heartBeatInterval":
/*optional, int, heartbeat interval of the security control panel and the
peripheral, unit: second*/
    }
}
```

A.3.57 JSON_GlassBreakDetectorCap

JSON message about the configuration capability of the composite PIR (Passive Infrared) glass-break detector

```
{
    "GlassBreakDetectorCap":{
        "zoneNo":{
/*optional, int, No. of zones that can be configured*/
            "@opt":[1, 3, 5]
        },
        "supportZoneType":{
/*optional, string, zone types supported by the peripheral: "Instant"-instant
zone, "Delay"-delay zone, "Follow"-follow zone, "Perimeter"-perimeter zone,
"24hNoSound"-24-hour silent zone, "Emergency"-panic zone, "Fire"-fire zone,
"Gas"-gas zone, "Medical"-medical zone, "Timeout"-timeout zone, "Non-Alarm"-
disabled zone, "Key"-key zone, "24hSound"-24-hour annunciating zone. When
switching zone type, zone types supported by the peripheral can be obtained*/
            "@opt":["Instant", "Delay", "Follow"]
        },
        "heartBeatInterval":{
/*optional, int, heartbeat interval of the security control panel and the
peripheral, unit: second*/
            "@opt":[5, 10, 20, 30]
        },
        "LEDEnabled":{
/*optional, boolean, whether to enable the LED indicator*/
            "@opt":[true, false]
        },
        "LEDLatchTime":{
```

```
/*optional, int, delay time of the LED indicator, unit: second*/
    "@min": ,
    "@max":
  },
  "findMeEnabled":{
/*optional, boolean, whether to enable the Find Me function*/
    "@opt":[true, false]
  },
  "sensitivityLevel":{
/*optional, string, sensitivity level: "high", "auto"-automatic, "antiPet"-pet
immune*/
    "@opt":["high", "auto", "antiPet"]
  },
  "checkEnabled":{
/*optional, boolean, whether to enable self-test: true=yes, false=no*/
    "@opt":[true, false]
  },
  "distance":{
/*optional, int, distance, unit: meter*/
    "@opt":[2, 4, 6, 8, 10, 12]
  },
  "alarmLogic":{
/*optional, string, alarm logic: "and", "or"*/
    "@opt":["and", "or"]
  }
}
}
```

A.3.58 JSON_HostConfigCap

JSON message about configuration capability of security control panel

```
{
  "HostConfigCap":{
    "isSptZone": ,
/*optional, boolean, whether it supports zone management*/
    "isSptNotRelateZones": ,
/*optional, boolean, whether it supports getting unlinked zones*/
    "isSptSubSys": ,
/*optional, boolean, whether it supports partition configuration*/
    "isSptPublicSubSys": ,
/*optional, boolean, whether it supports public partition configuration*/
    "isSptSubSysTime": ,
/*optional, boolean, whether it supports partition timer configuration*/
    "isSptDeviceTime": ,
/*optional, boolean, whether it supports timer configuration of the security
control panel*/
    "ExDevice":{
      "isSptSiren": ,
/*optional, boolean, whether it supports siren management*/
      "isSptRepeater": ,
```

```
/*optional, boolean, whether it supports repeater management*/
    "isSptOutputModule": ,
/*optional, boolean, whether it supports output module management*/
    "isSptOutput": ,
/*optional, boolean, whether it supports relay management*/
    "isSptRemoteCtrl": ,
/*optional, boolean, whether it supports keyfob management*/
    "isSptExtensionModule": ,
/*optional, boolean, whether it supports extension module management*/
    "isSptCardReader": ,
/*optional, boolean, whether it supports card reader configuration*/
    "isSptKeypad": ,
/*optional, boolean, whether it supports keypad configuration*/
    "isSptKeypadAddList": ,
/*optional, boolean, whether it supports getting the list of keypads that can
be added*/
    "isSptRemoteCfgPermissonUserType": ,
/*optional, boolean, whether it supports getting user names of users that have
the permission to remotely configure devices*/
    "isSptPircam": ,
/*optional, boolean, whether it supports configuring pircam (detector equipped
with camera) parameters, related URI: /ISAPI/SecurityCP/Configuration/pirCam/
capabilities?format=json*/
    "isSptMuteVoicePlanCFG": ,
/*optional, boolean, whether it supports configuring muting schedule, related
URI: /ISAPI/SecurityCP/Configuration/muteVoicePlanCFG*/
    "isSptVoicePromptCfg": ,
/*optional, boolean, whether it supports configuring voice prompt parameters,
related URI: /ISAPI/SecurityCP/voicePrompt?format=json*/
    "isSptSurroundEnvironmentCfg":
/*optional, boolean, whether it supports configuring device environment
parameters, related URI: /ISAPI/SecurityCP/surroundEnvironmentCfg/capabilities?
format=json*/
    },
    "MsgSend":{
        "isSptDirect": ,
/*optional, boolean, whether it supports configuring alarm center notification
for arming mode*/
        "isSptARC": ,
/*optional, boolean, whether it supports configuring alarm center notification
for listening mode*/
        "isSptCloud": ,
/*optional, boolean, whether it supports configuring Hik-Connect notification*/
        "isSptPhone": ,
/*optional, boolean, whether it supports configuring phone call and message
notification*/
        "isSptPhoneAnvanced": ,
/*optional, boolean, whether it supports advanced configuration of phone
notification and SMS notification*/
        "isSptMail": ,
/*optional, boolean, whether it supports configuring E-mail notification*/
        "isSptPSTNCfg":
```

```
/*optional, boolean, whether it supports configuration of uploading events or
alarms by phone call via PSTN (Public Switched Telephone Network)*/
    },
    "isSptAlarmUser": ,
/*optional, boolean, whether it supports user management*/
    "isSptSysManage": ,
/*optional, boolean, whether it supports system management*/
    "isSptCard": ,
/*optional, boolean, whether it supports card configuration*/
    "isSptEventRecord": ,
/*optional, boolean, whether it supports event recording configuration*/
    "isSptAdvanceCfg": ,
/*optional, boolean, whether it supports advanced configuration*/
    "isSptElectricLockCfg": ,
/*optional, boolean, whether it supports electric lock settings*/
    "isSptFaultCheckCfg": ,
/*optional, boolean, whether it supports enabling fault detection*/
    "isSptNetCfg": ,
/*optional, boolean, whether it supports network configuration*/
    "isSptReportCenterCfg": ,
/*optional, boolean, whether it supports configuring method to upload reports*/
    "isSptAlarmInCfg": ,
/*optional, boolean, whether the device supports configuring alarm input
parameters*/
    "isSptAlarmOutCfg": ,
/*optional, boolean, whether it supports configuring alarm output parameters*/
    "isSptSetAlarmHostOut": ,
/*optional, boolean, whether it supports setting alarm output*/
    "isSptControlAlarmChan": ,
/*optional, boolean, whether it supports arming or disarming the alarm input
port (zone)*/
    "isSptKeypadFaultProcessCfg": ,
/*optional, boolean, whether it supports configuring keypad linkage parameters
of the system fault*/
    "isSptRegisterMode": ,
/*optional, boolean, whether it supports registration mode configuration*/
    "isSptVideoStrategy": ,
/*optional, boolean, whether it supports configuring video recording strategy*/
    "isSptVideoLinkage": ,
/*optional, boolean, whether it supports configuring camera linkage*/
    "isSptArmSchedule": ,
/*optional, boolean, whether it supports configuring arming and disarming
schedule*/
    "isSptzoneAlarmTimeFilter": ,
/*optional, boolean, whether it supports filtering duplicate zone alarms in the
configured time interval*/
    "isSptMobCalibration": ,
/*optional, boolean, whether it supports map calibration for the radar*/
    "isSptSignalStrengthDetection": ,
/*optional, boolean, whether it supports signal strength detection, related
URI: /ISAPI/SecurityCP/Configuration/signalStrengthDetection/mode/capabilities?
format=json*/
```

```
"isSptFindMeDetector": true,
/*optional, boolean, whether it supports FindMe detection, related URI: /ISAPI/
SecurityCP/Configuration/zones/<ID>/findme?format=json*/
"isSptRepeatersMatch": true,
/*optional, boolean, whether it supports repeater match, related URI: /ISAPI/
SecurityCP/Configuration/repeaters/match?format=json*/
"isSptWirelessSmokeDetector": true,
/*optional, boolean, whether configuring wireless smoke detector is supported,
related URI: /ISAPI/SecurityCP/Configuration/wirelessSmokeDetector/capabilities?
format=json*/
"isSptEXDetect": true,
/*optional, boolean, whether it supports peripheral detection, related URI: /
ISAPI/SecurityCP/Configuration/EXDetect/zone/capabilities?format=json*/
"isSptTemperatureHumidityDetector": true,
/*optional, boolean, whether configuring temperature and humidity detector is
supported, related URI: /ISAPI/SecurityCP/Configuration/
temperaturaHumidityDetector/capabilities?format=json*/
"isSptHeatDetector": true,
/*optional, boolean, whether configuring heat detector is supported, related
URI: /ISAPI/SecurityCP/Configuration/heatDetector/capabilities?format=json*/
"isSptCODetector": true,
/*optional, boolean, whether configuring CO detector is supported, related
URI: /ISAPI/SecurityCP/Configuration/CODetector/capabilities?format=json*/
"isSptUSSD": true,
/*optional, boolean, whether it supports searching for USSD remainder, related
URI: /ISAPI/SecurityCP/USSD/capabilities?format=json*/
"isSptIPCLinkageSubSystem": true,
/*optional, boolean, whether it supports linking network camera to partition,
related URI: /ISAPI/SecurityCP/Configuration/IPC/linkageSubSystem/capabilities?
format=json*/
"isSptOutdoorDetector": true,
/*optional, boolean, whether it supports configuring outdoor tri-tech
detectors, related URI: /ISAPI/SecurityCP/Configuration/outdoorDetector/
capabilities?format=json*/
"isSptTransmitter": true,
/*optional, boolean, whether it supports configuring transmitter parameters,
related URI: /ISAPI/SecurityCP/Configuration/transmitter/paramCfg/capabilities?
format=json*/
"isSptARCMannualTest": true,
/*optional, boolean, whether it supports ARC manual test, related URI: /ISAPI/
SecurityCP/Configuration/ARC/manualTest/capabilities?format=json*/
"isSptLocalTransmitterDetectorReboot": true,
/*optional, boolean, whether it supports rebooting the detector connected to a
local transmitter, related URI: /ISAPI/SecurityCP/Configuration/accessModule/
detector/reboot?format=json*/
"isSptRepeaterManualForward": true,
/*optional, boolean, whether it supports manual transmission via repeaters,
related URI: /ISAPI/SecurityCP/Configuration/repeaters/manualForward/
capabilities?format=json*/
"isSptGetSingleOutputCfg": true,
/*optional, boolean, whether it supports getting the configuration of a single
relay, related URI: /ISAPI/SecurityCP/Configuration/outputs/<outputID>?
```

```
format=json*/
  "isSptIPCZoneCfg": true,
  /*optional, boolean, whether it supports network camera zone configuration,
  related URI: /ISAPI/SecurityCP/Zone/IPC/channels/capabilities?format=json*/
  "isSupportAddCustomAudio": true,
  /*optional, boolean, whether it supports importing custom audio files, related
  URI: /ISAPI/SecurityCP/customAudio/addCustomAudio?format=json*/
  "isSupportDeleteCustomAudio": true,
  /*optional, boolean, whether it supports deleting custom audio files, related
  URI: /ISAPI/SecurityCP/customAudio/deleteCustomAudio?format=json*/
  "isSupportSearchCustomAudioList": true,
  /*optional, boolean, whether it supports searching for the information of
  custom audio files, related URI: /ISAPI/SecurityCP/customAudio/
  searchCustomAudioList?format=json*/
  "isSupportAddCustomMessage": true,
  /*optional, boolean, whether it supports creating custom messages, related
  URI: /ISAPI/SecurityCP/customMessage/addCustomMessage?
  format=json&security=<security>&iv=<iv>*/
  "isSupportDeleteCustomMessage": true,
  /*optional, boolean, whether it supports deleting custom messages, related
  URI: /ISAPI/SecurityCP/customMessage/deleteCustomMessage?format=json*/
  "isSupportSearchCustomMessageList": true,
  /*optional, boolean, whether it supports searching for custom messages, related
  URI: /ISAPI/SecurityCP/customMessage/searchCustomMessageList?
  format=json&security=<security>&iv=<iv>*/
  "isSupportModifyCustomMessage": true,
  /*optional, boolean, whether it supports editing custom messages, related URI: /
  ISAPI/SecurityCP/customMessage/modifyCustomMessage?
  format=json&security=<security>&iv=<iv>*/
  "isSupportTamperDetectionEnabledCfg": true,
  /*optional, boolean, whether it supports the configuration of enabling tamper
  detection*/
  }
}
```

A.3.59 JSON_HostControlCap

JSON message about operation and control capability of security control panel

```
{
  "HostControlCap":{
    "SubSysCap":{
      /*optional, operation and control capability of partitions*/
      "isSptArm": ,
      /*optional, boolean, whether supports arming the partition*/
      "armType":{
        /*partition arming type, it is valid when the partition supports arming, "stay"-
        stay arming, "away"-away arming*/
        "@opt":"stay,away"
      },
      "isSptDisarm": ,
```

```
/*optional, boolean, whether supports disarming partition*/
    "isSptClearAlarm": ,
/*optional, boolean, whether supports clearing alarms for partitions*/
    "isSptBatchArm": ,
/*optional, boolean, whether to support arming in a batch*/
    "isSptBatchDisarm": ,
/*optional, boolean, whether to support disarming in a batch*/
    "isSptBatchClearAlarm": ,
/*optional, boolean, whether to support clearing alarms in a batch*/
    "isSptStatus":
/*optional, boolean, whether supports getting status of partitions*/
    },
    "ZoneCap":{
/*optional, operation and control capability of zones*/
    "isSptBypass": ,
/*optional, boolean, whether supports bypass*/
    "isSptBypassBatch": ,
/*optional, boolean, whether is supports batch bypass*/
    "isSptBypassRecover": ,
/*optional, boolean, whether supports bypass recovered*/
    "isSptBypassRecoverBatch":
/*optional, boolean, whether supports batch bypass recovered*/
    },
    "isSptOutputsCtrl": ,
/*optional, boolean, whether supports controlling relay*/
    "isSptOutputsCtrlBatch": ,
/*optional, boolean, whether supports controlling relay in batch*/
    "isSptSirenCtrl": ,
/*optional, boolean, whether to support controlling siren*/
    "sirenCtrlType":["wired"],
/*optional, type of siren that supports being controlled: "wired"*/
    "isSptOneKeyAlarmCtrl": ,
/*optional, boolean, whether it supports one-push alarm, related URI: /ISAPI/
SecurityCP/control/oneKeyAlarm?format=json*/
    "isSptSubSysFaultConfirm": ,
/*optional, boolean, whether to support partition fault acknowledgment*/
    "isSptInputsCtrl": true,
/*optional, boolean, whether it supports arming and disarming control of alarm
input*/
    "isSptAlarmStrobeCtrl": true,
/*optional, boolean, whether it supports alarm strobe control, related URI: /
ISAPI/SecurityCP/control/alarmStrobe/<ID>?format=json*/
    "isSptElectricLockCtrl": true,
/*optional, boolean, whether it supports electric lock control, related URI: /
ISAPI/SecurityCP/control/electricLock/<ID>?format=json*/
    "isSptRadarFieldPreviewCtrl": true,
/*optional, boolean, whether it supports radar field preview control, related
URI: /ISAPI/Radar/Configuration/radarFieldPreviewCtrl?format=json*/
    "Operate": {
/*optional, object, operation parameters*/
    "moduleOperateCode": {
/*required, object, operation code*/
```



```
        "@min": 1,
/*optional, int, the minimum length of the character*/
        "@max": 2,
/*optional, int, the maximum length of the character*/
        "character": ["1", "2", "3"]
/*optional, array, supported character types: "1", "2", "3"*/
    }
}
}
```

Remarks

The required node refers to the parameter that is supported by the device and must be configured. For the optional node, only when it is returned, it indicates that the device supports this parameter; otherwise, the optional node is invalid.

A.3.60 JSON_HostStatus

HostStatus message in JSON format

```
{
  "HostStatus":{
    "temperature": ,
/*optional, float, security control panel temperature*/
    "mode": "",
/*optional, string, current mode of the security control panel*/
    "tamperEvident": "",
/*optional, boolean, whether the security control panel is tampered: "true"-yes
(tampered), "false"-no (not tampered)*/
    "moduleType": "",
/*optional, string, module type: "USB"-USB module, "sub1G"-sub1G module,
"zigBee"-ZigBee module*/
    "moduleStatus": ""
/*optional, string, module status: "plugged", "free"*/
  }
}
```

A.3.61 JSON_HostStatusCap

JSON message about the capability of getting security control panel's status

```
{
  "HostStatusCap":{
    "isSptZones": ,
/*optional, boolean, whether it supports getting the status of all zones*/
    "isSptSubSystems": ,
/*optional, boolean, whether it supports getting the status of all partitions*/
    "isSptExDev": ,

```

```
/*optional, boolean, whether it supports getting the status of all peripherals*/
    "isSptBatteries": ,
/*optional, boolean, whether it supports getting the status of all storage
batteries*/
    "isSptCommunication": ,
/*optional, boolean, whether it supports getting the communication status of
the security control panel*/
    "isSptHostItself": ,
/*optional, boolean, whether it supports getting status of the security control
panel itself*/
    "isSptACPower": ,
/*optional, boolean, whether it supports getting AC power supply status*/
    "isSptAllHost": ,
/*optional, boolean, whether it supports getting the status of all security
control panels*/
    "isSptSubSysArmStatus": ,
/*optional, boolean, whether it supports partition's arming status*/
    "isSptSubSysFault": ,
/*optional, boolean, whether it supports getting partition faults*/
    "isSptCondZones": ,
/*optional, boolean, whether it supports getting zone status by conditions*/
    "maxZonesResults": ,
/*optional, integer, maximum number of zone status that can be obtained this
time by calling the URI*/
    "isSptOutputMod": ,
/*optional, boolean, whether it supports getting output module status*/
    "isSptCondOutputs": ,
/*optional, boolean, whether it supports getting relay status by conditions*/
    "maxOutputsResults": ,
/*optional, integer, maximum number of output module status that can be
obtained this time by calling the URI*/
    "isSptOutputs": ,
/*optional, boolean, whether it supports getting relay status*/
    "isSptSirenMod": ,
/*optional, boolean, whether it supports getting siren status*/
    "isSptRepeaterMod": ,
/*optional, boolean, whether it supports getting repeater status*/
    "isSptCardReaderMod": ,
/*optional, boolean, whether it supports getting card reader status*/
    "isSptExtensionModuleMod": ,
/*optional, boolean, whether it supports getting extension module status*/
    "isSptKeypadMod": ,
/*optional, boolean, whether it supports getting keypad status*/
    "isSptTransmitter": true,
/*optional, boolean, whether it supports getting transmitter status, related
URI: /ISAPI/SecurityCP/status/transmitterStatus?format=json*/
    "isSupportSubSystemsSearch": true,
/*optional, boolean, whether it supports searching for partition status (by
page)*/
    "isSupportSirenSearch": true,
/*optional, boolean, whether it supports searching for sounder status (by
page)*/
```

```
"isSptIPCZones": true
/*optional, boolean, whether it supports getting network camera zone status,
related URI: /ISAPI/SecurityCP/Zone/IPC/status?format=json*/
}
}
```

A.3.62 JSON_IndoorDualTechnologyDetector

Message about the parameters of the indoor dual-technology detector of a specific zone in JSON format.

```
{
  "IndoorDualTechnologyDetector": {
    "LEDEnabled": ,
    /*optional, boolean, whether to enable the LED indicator*/
    "LEDLatchTime": ,
    /*optional, boolean, delay time of the LED indicator, unit: second*/
    "findMeEnabled": ,
    /*optional, boolean, whether to enable the Find Me function*/
    "sensitivityLevel": "",
    /*optional, string, sensitivity level: "high", "auto", "antiPet"-pet immune*/
    "checkEnabled": ,
    /*optional, boolean, whether to enable self-test*/
    "alarmLogic": "",
    /*optional, string, alarm logic: "and", "or"*/
    "distance": "",
    /*optional, string, adjustment distance: "high", "low"*/
    "heartBeatInterval":
    /*optional, int, heartbeat interval of the security control panel and the
    peripheral, unit: second*/
  }
}
```

A.3.63 JSON_IndoorDualTechnologyDetectorCap

Message about the configuration capability of the indoor dual-technology detector in JSON format.

```
{
  "IndoorDualTechnologyDetectorCap": {
    "zoneNo": {
    /*optional, int, values that can be configured as the zone No.*/
      "@opt": [1,3,5]
    },
    "supportZoneType": {
    /*optional, string, zone types supported by the peripheral: "Instant"-instant
    zone, "Delay"-delay zone, "Follow"-follow zone, "Perimeter"-perimeter zone,
    "24hNoSound"-24-hour silent zone, "Emergency"-panic zone, "Fire"-fire zone,
    "Gas"-gas zone, "Medical"-medical zone, "Timeout"-timeout zone, "Non-Alarm"-
    disabled zone, "Key"-key zone, "24hSound"-24-hour annunciating zone. When
```

```
switching zone type, zone types supported by the peripheral can be obtained*/
    "@opt":["Instant","Delay","Follow"]
  },
  "heartBeatInterval":{
/*optional, int, heartbeat interval of the security control panel and the
peripheral, unit: second*/
    "@opt":[5,10,20,30]
  },
  "LEDEnabled": {
/*optional, boolean, whether to enable the LED indicator*/
    "@opt":[true,false]
  },
  "LEDLatchTime": {
/*optional, boolean, delay time of the LED indicator, unit: second*/
    "@min":0,
    "@max":0
  },
  "findMeEnabled": {
/*optional, boolean, whether to enable the Find Me function*/
    "@opt":[true,false]
  },
  "sensitivityLevel": {
/*optional, string, sensitivity level: "high","auto","antiPet"-pet immune*/
    "@opt":["high","auto","antiPet"]
  },
  "checkEnabled":{
/*optional, boolean, whether to enable self-test*/
    "@opt":[true,false]
  },
  "alarmLogic": {
/*optional, string, alarm logic: "and","or"*/
    "@opt":["and","or"]
  },
  "distance": {
/*optional, string, adjustment distance: "high","low"*/
    "@opt":["high","low"]
  }
}
}
```

A.3.64 JSON_Keypad

JSON message about parameters of a keypad

```
{
  "Keypad":{
    "id": ,
/*optional, int, keypad No., it starts from 1*/
    "related": ,
/*boolean, whether to link to the physical keypad. For PUT method, this node is
optional; for GET method, this node is required*/
```

```
"name":"","
/*optional, string, keypad name*/
"subSystem": ,
/*optional, array, linked partitions, e.g., [1,2,3] indicates linking to
partition 1, partition 2, and partition 3*/
"buzzerEnabled": ,
/*optional, boolean, whether to enable the buzzer: "true"-yes, "false"-no*/
"swipingCardEnabled": ,
/*optional, boolean, whether to enable the card swiping function: "true"-yes,
"false"-no*/
"armByKeyEnabled": ,
/*optional, boolean, whether to enable the function of arming and disarming by
pressing the key: "true"-yes (the virtual zone alarm will be enabled), "false"-
no (the virtual zone alarm will be disabled). This function cannot be disabled
for panic alarm and fire alarm*/
"seq":"","
/*optional, string, keypad serial No., this node is required when keypadAttrib
is "wireless"*/
"keypadAttrib":"","
/*optional, read-only, string, keypad attribute: "wired", "wireless"*/
"checkTime": ,
/*optional, int, offline time, unit: hour*/
"address": ,
/*optional, read-only, int, keypad address, this node is only returned by wired
keypads*/
"type": "",
/*optional, read-only, string, keypad model*/
"version": "",
/*optional, read-only, string, keypad version*/
"status":"","
/*optional, read-only, string, keypad status: "notRelated"-
unlinked,"online","offline","heartbeatAbnormal"-heartbeat exception*/
"tamperEvident": ,
/*optional, read-only, boolean, tampering status: "true"-tampered, "false"-not
tampered*/
"alarmVoicePromptEnabled": ,
/*optional, boolean, whether to enable voice prompt for panic alarm: true=yes
(the panic alarm of the keypad will be linked with voice prompt), false=no (the
panic alarm of the keypad will not be linked with voice prompt)*/
"SetLEDCFG":{
/*optional, schedule configuration to enable the backlight*/
  "TimeCFGList":[{
    "TimeSegment":{
      "id": ,
/*optional, int, time period No.*/
      "enabled": ,
/*optional, boolean, whether to enable: "true"-yes, "false"-no*/
      "beginTime":"","
/*optional, string, start time of the time period, e.g., 09:00*/
      "endTime":"","
/*optional, string, end time of the time period, e.g., 11:30*/
    }
  ]
}
```

```
    }}
  },
  "attribute": "",
  /*optional, string, read-only, keypad type: "LED" (LED keypad), "LCD" (LCD
  keypad)*/
  "company": "",
  /*optional, string, company name (for interface display), this node can be
  configured when attribute is "LED"*/
  "phoneNo": "",
  /*optional, string, phone number (for interface display), this node can be
  configured when attribute is "LED"*/
  "activeDelayEnabled": true,
  /*optional, boolean, whether to enable delayed activation*/
  "armAndDisarmAuthorityCfg": "tag",
  /*optional, string, permission for arming/disarming the partition: "tag"-by
  card, "password"-by password, "tagOrPassword"-by card or password,
  "tagAndPassword"-by card and password*/
  "alarmBuzzerEnabled": true,
  /*optional, boolean, whether to enable alarm buzzer: true, false*/
  "buttonBuzzerEnabled": true
  /*optional, boolean, whether to enable button buzzer: true, false*/
  }
}
```

A.3.65 JSON_KeypadAddList

KeypadAddList message in JSON format

```
{
  "KeypadAddList": [{
    "KeypadAdd": {
      "id": ,
      /*optional, integer type, keypad No.*/
      "address": ,
      /*optional, integer type, keypad address*/
      "isSupportRemoteCtrlAdd": ,
      /*optional, boolean type, whether to support adding keyfob: "true"-yes, "false"
      or this node is not returned-no*/
      "isSupportCardAdd":
      /*optional, boolean type, whether to support adding card: "true"-yes, "false"
      or this node is not returned-no*/
    }
  ]
}
```

A.3.66 JSON_KeypadAddListCap

KeypadAddListCap capability message in JSON format

```
{
  "KeypadAddListCap":{
    "id":{
/*optional, keypad No.*/
      "@min": ,
      "@max":
    },
    "address":{
/*optional, keypad address*/
      "@opt":[1,2,3]
    },
    "isSupportRemoteCtrlAdd":true,
/*optional, whether to support adding keyfob*/
    "isSupportCardAdd":true
/*optional, whether to support adding card*/
  }
}
```

A.3.67 JSON_KeypadCap

JSON message about keypad configuration capability

```
{
  "KeypadCap":{
    "id":{
/*optional, keypad No.*/
      "@min": ,
      "@max":
    },
    "related":"true,false",
/*whether to link to the physical keypad. For PUT method, this node is
optional; for GET method, this node is required*/
    "name":{
/*optional, keypad name*/
      "@min": ,
      "@max":
    },
    "subSystem":{
/*optional, number of linked partitions*/
      "@min": ,
      "@max":
    },
    "subSystemNo":{
/*optional, partition No. range*/
      "@min": ,
      "@max":
    },
    "buzzerEnabled":"true,false",
/*optional, whether to enable the buzzer: "true"-yes, "false"-no*/
    "swipingCardEnabled":"true,false",
  }
}
```

```
/*optional, whether to enable the card swiping function: "true"-yes, "false"-no*/
"alarmVoicePromptEnabled":{
/*optional, boolean, whether to enable voice prompt for panic alarm: true=yes
(the panic alarm of the keypad will be linked with voice prompt), false=no (the
panic alarm of the keypad will not be linked with voice prompt)*/
"@opt":[true,false]
},
"armByKeyEnabled":"true,false",
/*optional, boolean, whether to enable the function of arming and disarming by
pressing the key: "true"-yes (the virtual zone alarm will be enabled), "false"-
no (the virtual zone alarm will be disabled). This function cannot be disabled
for panic alarm and fire alarm*/
"seq":{
/*optional, keypad serial No., this node is required when keypadAttrib is
"wireless"*/
"@min": ,
"@max":
},
"keypadAttrib":{
/*optional, read-only, keypad attribute: "wired", "wireless"*/
"@opt":["wired","wireless"]
},
"checkTime":{
/*optional, offline duration, unit: hour*/
"@min": ,
"@max":
},
"address":{
/*optional, read-only, keypad address, this node is only returned by wired
keypads*/
"@min": ,
"@max":
},
"type":{
/*optional, read-only, keypad model*/
"@min": ,
"@max":
},
"version":{
/*optional, read-only, keypad version*/
"@min": ,
"@max":
},
"status":{
/*optional, read-only, keypad status: "notRelated"-
unlinked,"online","offline","heartbeatAbnormal"-heartbeat exception*/
"@opt":["notRelated","online","offline","heartbeatAbnormal"]
},
"tamperEvident":"true,false",
/*optional, read-only, tampering status: "true"-tampered, "false"-not tampered*/
"SetLED CFG":{
```



```
/*optional, schedule configuration to enable the backlight*/
    "maxSize":8,
/*optional, maximum number of time periods that can be configured for each day*/
    "TimeSegment":{
        "id":{
/*optional, time period No.*/
            "@min": ,
            "@max":
        },
        "enabled":"true,false",
/*optional, whether to enable: "true"-yes, "false"-no*/
        "beginTime":"true,false",
/*optional, start time of the time period*/
        "endTime":"true,false"
/*optional, end time of the time period*/
    }
},
    "attribute": {
/*optional, string, read-only, keypad type: "LED" (LED keypad), "LCD" (LCD
keypad)*/
        "@opt":["LED","LCD"]
    },
    "company": {
/*optional, string, company name (for interface display), this node can be
configured when attribute is "LED"*/
        "@min": ,
        "@max":
    },
    "phoneNo": {
/*optional, string, phone number (for interface display), this node can be
configured when attribute is "LED"*/
        "@min": ,
        "@max":
        "method":{
/*methods supported by the function: "put"-edit, "getAll"-get all, "add"*/
            "@opt":["put","getAll","add"]
        },
        "heartBeatInterval": {
/*optional, object, heartbeat interval between host and peripheral, unit:
second*/
            "@opt": [5, 10, 20, 30]
/*optional, int*/
        },
        "isSupportSignalTest": true,
/*optional, boolean, whether it supports signal strength detection; if the node
is not returned or the value is false, it indicates that this function is not
supported*/
        "isSupportZoneTest": true,
/*optional, boolean, whether it supports zone detection; if the node is not
returned or the value is false, it indicates that this function is not
supported*/
        "activeDelayEnabled": {
```

```
/*optional, object, whether to enable delayed activation: true, false; this
node is valid for delayed zone only*/
    "@opt": [true, false]
  },
  "armAndDisarmAuthorityCfg": {
/*optional, string, permission for arming/disarming the partition: "tag"-by
card, "password"-by password, "tagOrPassword"-by card or password,
"tagAndPassword"-by card and password*/
    "@opt": ["tag", "password", "tagOrPassword", "tagAndPassword"]
  },
  "alarmBuzzerEnabled": {
/*optional, boolean, whether to enable alarm buzzer: true, false*/
    "@opt": [true, false]
  },
  "buttonBuzzerEnabled": {
/*optional, boolean, whether to enable button buzzer: true, false*/
    "@opt": [true, false]
  }
}
```

A.3.68 JSON_KeypadFaultProcessCfg

KeypadFaultProcessCfg message in JSON format

```
{
  "KeypadFaultProcessCfg":{
    "JointLED":{
/*optional, keypad indicator output linked with the fault*/
      "tamper": ,
/*optional, boolean type, whether it is tampered*/
      "offline": ,
/*optional, boolean type, whether it is offline*/
      "lowPower": ,
/*optional, boolean type, whether it is low battery*/
      "trigger": ,
/*optional, boolean type, whether it is triggered by zone*/
      "power": ,
/*optional, boolean type, whether it is power supply fault*/
      "communication": ,
/*optional, boolean type, whether it is communication fault*/
      "all":
/*optional, boolean type, all faults. If this node is "true", all faults will
trigger keypad indicator output*/
    },
    "JointSound":{
/*optional, keypad sound output linked with the fault*/
      "tamper": ,
/*optional, boolean type, whether it is tampered*/
      "offline": ,
/*optional, boolean type, whether it is offline*/
    }
  }
}
```

```
    "lowPower": ,
/*optional, boolean type, whether it is low battery*/
    "trigger": ,
/*optional, boolean type, whether it is triggered by zone*/
    "power": ,
/*optional, boolean type, whether it is power supply fault*/
    "communication": ,
/*optional, boolean type, whether it is communication fault*/
    "all":
/*optional, boolean type, all faults. If this node is "true", all faults will
trigger keypad sound output*/
    }
  }
}
```

A.3.69 JSON_KeypadFaultProcessCfgCap

KeypadFaultProcessCfgCap capability message in JSON format

```
{
  "KeypadFaultProcessCfgCap":{
    "address":{
/*optional, keypad address*/
      "@opt":[1,2,3]
    },
    "JointLED":{
/*optional, keypad indicator output linked with the fault*/
      "tamper":"true,false",
/*optional, whether it is tampered*/
      "offline":"true,false",
/*optional, whether it is offline*/
      "lowPower":"true,false",
/*optional, whether it is low battery*/
      "trigger":"true,false",
/*optional, whether it is triggered by zone*/
      "power":"true,false",
/*optional, whether it is power supply fault*/
      "communication":"true,false",
/*optional, whether it is communication fault*/
      "all":"true,false"
/*optional, all faults. If this node is "true", all faults will trigger keypad
indicator output*/
    },
    "JointSound":{
/*optional, keypad sound output linked with the fault*/
      "tamper":"true,false",
/*optional, whether it is tampered*/
      "offline":"true,false",
/*optional, whether it is offline*/
      "lowPower":"true,false",
/*optional, whether it is low battery*/
```

```
    "trigger":"true,false",
/*optional, whether it is triggered by zone*/
    "power":"true,false",
/*optional, whether it is power supply fault*/
    "communication":"true,false",
/*optional, whether it is communication fault*/
    "all":"true,false"
/*optional, all faults. If this node is "true", all faults will trigger keypad
sound output*/
  }
}
}
```

A.3.70 JSON_KeypadList

JSON message about keypad status

```
{
  "KeypadList":[{
/*optional, keypad list*/
    "Keypad":{
      "id": ,
/*required, int, keypad No.*/
      "seq": "",
/*required, string, peripheral serial No.*/
      "name": "",
/*optional, string, keypad name*/
      "status": "",
/*optional, string, keypad status: "notRelated"-not linked, "online",
"offline", "heartbeatAbnormal"-heartbeat exception*/
      "tamperEvident": ,
/*optional, boolean, tampering status: "true"-tampered, "false"-not tampered*/
      "keypadAttrib": "",
/*string, keypad attribute: "wired", "wireless"*/
      "charge": "",
/*optional, string, state of charge: "normal", "lowPower"-low battery*/
      "signal": ,
/*optional, int, signal strength, it is between 0 and 255*/
      "address":
/*optional, int, keypad address, this node is only returned by wired keypads*/
      "model": "DS-PK1-E-WE",
/*optional, string, model: "DS-PK1-E-WE" (wireless LED keypad)*/
      "temperature": 1,
/*optional, int, temperature*/
      "subSystemList": [1, 2, 3],
/*optional, array, list of linked partitions*/
      "isViaRepeater": true,
/*optional, boolean, whether the signal is forwarded via repeater*/
      "repeaterName": "test",
/*optional, string, repeater name, this node is valid when the value of
isViaRepeater is true*/
```

```
    "version": "test",
/*optional, string, version No.*/
    "ZoneList": [
/*optional, array, list of linked zones*/
        {
            "zoneID": 1,
/*required, int, zone ID, range: [0,95]*/
            "detectorType": "other",
/*optional, string, detector type: "panicButton", "magneticContact",
"smokeDetector", "activeInfraredDetector", "passiveInfraredDetector",
"glassBreakDetector", "vibrationDetector", "dualTechnologyPirDetector",
"tripleTechnologyPirDetector", "humidityDetector", "temperatureDetector",
"combustibleGasDetector", "dynamicSwitch", "controlSwitch", "smartLock",
"waterDetector", "displacementDetector", "singleInfraredDetector",
"singleZoneModule", "curtainInfraredDetector", "pircam", "slimMagneticContact",
"indoorDualTechnologyDetector", "magnetShockDetector", "waterLeakDetector",
"wirelessSmokeDetector", "wirelessGlassBreakDetector",
"wirelessTemperatureHumidityDetector", "wirelessHeatDetector",
"wirelessCODetector", "wirelessPIRCeilingDetector",
"wirelessExternalMagnetDetector", "wirelessPIRCurtainDetector",
"wirelessDTAMCurtainDetector", "outdoorDetector", "other", "tamperDetector"
(tamper-proof detector)*/
            "isBypassed": true,
/*optional, boolean, whether to bypass the zone: true-bypassed, false-bypass
recovered*/
            "subSystemList": [1, 2, 3],
/*optional, array, list of the linked partitions*/
            "zoneType": "Instant",
/*optional, string, zone type: "Instant", "Delay", "Follow", "Perimeter",
"24hNoSound", "Emergency", "Fire", "Gas", "Medical", "Timeout", "Non-Alarm",
"24hSound", "24h"*/
            "tamperEvident": true,
/*optional, boolean, tamper status: true, false*/
            "enterDelay": 1,
/*optional, int, (enter) delay time, unit: second. You can set the delay time
for each zone if the device supports this function*/
            "exitDelay": 5,
/*optional, int, (exit) delay time, unit: second*/
            "alarm": true
/*optional, boolean, whether the alarm is triggered in the zone: true, false*/
        }
    ],
    "OutputList": [
/*optional, array, list of linked relays*/
        {
            "outputID": 1,
/*optional, int, relay ID*/
            "subSystemList": [1, 2, 3]
/*optional, array, list of linked partitions*/
        }
    ],
    "isSupportAddType": true,
```

```
/*optional, boolean, whether it supports adding types, if false is returned or
this node is not returned, it indicates that this function is not supported.
Currently, only adding zones and relays are supported; this node should be used
together with keypadAttrib*/
    "deviceNo": 1
/*optional, int, device No., range:[1,1000]*/
    }
  ]]
}
```

A.3.71 JSON_List_ARC

JSON message about parameters of all alarm receiving centers

```
{
  "List":[{
    "ARC":{
      "id": ,
/*required, integer type, alarm receiving center No., which starts from 1*/
      "enabled": ,
/*required, boolean type, whether to enable alarm receiving center
configuration, "true,false"*/
      "spareEnabled": ,
/*optional, boolean type, whether to enable backup alarm receiving center,
"true", "false"*/
      "enable": ,
/*optional, boolean type, whether to enable*/
      "addrType": "",
/*required, string type, alarm receiving center address type*/
      "ipVersion": "",
/*string type, IP address version information, it is valid only when addrType
is "ipAddress"*/
      "ipAddress": "",
/*string type, IP address types (IPv4 or IPv6), it is valid only when addrType
is "ipAddress"*/
      "hostName": "",
/*string type, domain name, it is required when addrType is "hostName"*/
      "port": ,
/*optional, integer type, port number for receiving alarm or event by the alarm
receiving center*/
      "centerAccount": "",
/*optional, string type, alarm receiving center account, which is used to mark
the device*/
      "protocol": "",
/*optional, string type, protocol type*/
      "transMode": "",
/*optional, string type, transmission mode*/
      "timeout": ,
/*optional, integer type, timeout of waiting for acknowledgment from the alarm
receiving center after uploading the event, the event will be uploaded again
after timeout*/
```

```
    "retryTime": ,
/*optional, integer type, re-uploading times*/
    "heartBeatInterval": ,
/*optional, integer type, heartbeat interval*/
    "algorithm": "",
/*string type, encryption algorithm, it is valid when protocol is "*SIA-DCS" or
"*ADM-CID"*/
    "bits": ,
/*integer type, number of bits for the encryption key, it is valid when
protocol is "*SIA-DCS" or "*ADM-CID"*/
    "key": "",
/*string type, key, which is used to encrypt the uploaded messages, and it is
valid when protocol is "*SIA-DCS" or "*ADM-CID"*/
    "authEnabled": true,
/*dependent, boolean, whether to enable authentication, this node is required
when protocol is "CSV-IP"*/
    "userName": "",
/*dependent, string, user name, this node is required when protocol is "CSV-IP"
and authEnabled is true*/
    "password": "",
/*dependent, string, password, this node is required when protocol is "CSV-IP"
and authEnabled is true*/
    "ARCchannelList": [{
/*ARC channel list*/
        "ARCchannel": {
            "ARCIid": 1,
/*optional, int, ARC channel ID*/
            "enabled": true,
/*optional, boolean, whether to enable the ARC channel*/
            "transMethod": "",
/*optional, string, communication method*/
            "addrType": "",
/*required, string, ARC address type*/
            "ipVersion": "",
/*optional, string, IP address version, it is "IPv4" by default. This node is
valid when addrType is "ipAddress"*/
            "ipAddress": "",
/*dependent, string, IPv4 address or IPv6 address. This node is valid when
addrType is "ipAddress"*/
            "hostName": "",
/*dependent, string, domain name. This node is valid when addrType is
"hostName"*/
            "port": 2,
/*optional, int, port No.*/
            "centerAccount": "",
/*optional, string, ARC account*/
            "protocol": "",
/*optional, string, protocol type for uploading alarms*/
            "transMode": "",
/*optional, string, transmission mode*/
            "timeout": 28,
/*optional, int, timeout duration for waiting for the ARC to confirm the event
```

```
after the event is uploaded, unit: second. After timeout, the event will be
uploaded again*/
    "retryTime":3,
/*optional, int, retry times*/
    "heartBeatInterval":12,
/*optional, int, heartbeat interval*/
    "algorithm":"",
/*optional, string, encryption algorithm. This node is valid when protocol is
"*SIA-DCS" or "*ADM-CID"*/
    "bits": ,
/*optional, int, number of bits of the encryption key. This node is valid when
protocol is "*SIA-DCS" or "*ADM-CID"*/
    "key":""," ,
    "periodicTestEnabled": ,
/*optional, boolean, whether to enable periodic test. After the periodic test
function is enabled and configured, the device will regularly upload test
reports according to the configured interval. This node is valid when protocol
is "DC-09" (including ADM-CID, SIA-DCS, *SIA-DCS, *ADM-CID)*/
    "periodicTestTimeCfg":
/*optional, int, periodic test interval, unit: second. This node is valid when
protocol is "DC-09"*/
    }
  }],
  "subSystem":[1, 2, 3],
/*optional, array of int, linked partition*/
  "sysEvent":["", ""],
/*optional, array of string, system event: "IPCOOfflineTest" (network camera
disconnected detection), "wiredNetTest" (wired network fault detection),
"mobileNetTest" (mobile network fault detection), "phoneOfflineTest" (phone
disconnected detection), "hostBatteryTest" (security control panel battery
detection), "WiFiNetTest" (Wi-Fi network fault detection), "SIMTest" (SIM card
fault detection), "RS485Test" (RS-485 exception detection)*/
  "timeStampGMTEnabled":true
/*optional, boolean, whether to enable GMT time stamp: true-enable (default),
false-disable. After this function is enabled, the time stamp in the
information uploaded by the device is London time; otherwise, it is the device
time. This node is valid when protocol is "DC-09"*/
  "company": "None",
/*optional, string, company: "None", "Multi" (Hungary-Multi ARC), "FranceARC"
(France ARC, or Le central d'alarme en france). This node is valid when the
value of protocol is "SIA"*/
  "pircamUploadMode": "Picture",
/*optional, string, uploading method: "Video", "Picture", this node is valid
when the value of company is "FranceARC"*/
  "transMethod": "IP",
/*optional, object, transmission mode: "IP", "PSTN" (public switched telephone
network), "serialPort"*/
  "spareARCID": 2,
/*optional, int, the No. of spare ARC: 2, 4; when the No. of the main ARC is 1,
the No. of the spare ARC is 2; when the No. of the main ARC is 3, the No. of
the spare ARC is 4*/
  "FSKCfg": {
```



```
/*optional, object, FSK configuration*/
    "baudRate": "300",
/*optional, string, baud rate: 300, 600, 1200, 2400, 4800, 9600, 14400, 19200,
38400, 56000, 57600, 115200, 128000, 256000, 230400*/
    "dataBit": 6,
/*optional, int, data bit: 6, 7, 8*/
    "parityBit": "none",
/*optional, string, parity check: "none", "odd"(odd check)*/
    "stopBit": 1
/*optional, int, stop bit: 1, 2*/
    }
}
}]
}
```

A.3.72 JSON_List_Card

List_Card message in JSON format

```
{
  "List":[{
    "Card":{
      "id": ,
/*required, integer type, card No., which starts from 1*/
      "enabled": ,
/*required, boolean type, whether to enable card function: "true"-yes, "false"-
no*/
      "seq": "",
/*required, string type, card serial No.*/
      "name": "",
/*optional, string type, card name*/
      "armEnabled": ,
/*optional, boolean type, whether to have arming permission*/
      "disarmEnabled": ,
/*optional, boolean type, whether to have disarming permission*/
      "subSystem": ,
/*optional, array, linked partitions, e.g., [1,2,3] indicates linking to
partition 1, partition 2, and partition 3*/
      "cardType": ""
/*optional, string, card type: "operateCard"-operation card, "patrolCard"-
patrol card*/
    }
  }]
}
```

A.3.73 JSON_List_CardReader

JSON message about the parameters of all card readers

```
{
  "List": [{
    "CardReader": {
      "enabled": ,
      /*optional, boolean, whether to enable the card reader*/
      "id": ,
      /*optional, integer type, card reader No., it starts from 1*/
      "related": ,
      /*required, boolean type, whether to link to the physical card reader. For PUT
method, this node is optional; for GET method, this node is required*/
      "seq": "",
      /*optional, string type, card reader serial No.*/
      "name": "",
      /*optional, string type, card reader name*/
      "subSystem": ,
      /*optional, array, partition No. range, e.g., [1,2.3] indicates linking to
partition 1, partition 2, and partition 3*/
      "buzzerEnabled": ,
      /*optional, boolean type, whether to enable the buzzer: "true"-yes, "false"-no*/
      "checkTime": ,
      /*optional, integer type, offline time, unit: hour*/
      "LEDEnabled": ,
      /*optional, boolean, whether to enable the LED indicator*/
      "heartBeatInterval":
      /*optional, int, heartbeat interval of the security control panel and the
peripheral, unit: second*/
      "operationMode": "simple"
      /*optional, object, operation mode: "simple", "standard"*/
    }
  ]
}
```

A.3.74 JSON_List_CurtainInfraredDetector

Message about the parameters of all curtain PIR (Passive Infrared) detectors in JSON format.

```
{
  "List": [{
    "CurtainInfraredDetector": {
      "zoneNo": ,
      /*optional, int, values that can be configured as the zone No.*/
      "LEDEnabled": ,
      /*optional, boolean, whether to enable the LED indicator*/
      "LEDLatchTime": ,
      /*optional, int, delay time of the LED indicator, unit: second*/
      "heartBeatInterval": ,
      /*optional, int, heartbeat interval of the security control panel and the
peripheral, unit: second*/
      "alwaysActiveEnabled": true,
      /*optional, object, whether to keep the detector enabled when the zone is
```

```
disarmed*/
    "triggerNumLimited": 5,
/*optional, object, alarm triggering times*/
    "curtainDetectorType": "CurtainInfrared",
/*optional, object, curtain detector type: "CurtainInfrared" (IR curtain
detector), "R3PIRCurtain" (R3 PIR curtain detector), "R3DTAMCurtain" (R3 DTAM
curtain detector)*/
    "directionIdentification": "LeftToRight",
/*optional, object, direction identification: "LeftToRight", "RightToLeft",
"Off". By default, it is "Off"*/
    "microwaveSensitivity": 5,
/*optional, object, microwave sensitivity: 5, 8, 10. Unit: meter, by default,
it is 10 meters*/
    "antiMaskingEnabled": true,
/*optional, object, whether to enable anti-masking, by default, it is enabled*/
    "AMPulseInterval": 5
/*optional, object, AM pulse interval: 5, 30, 60, 120. Unit: second*/
  }
}
}
```

A.3.75 JSON_List_ExtensionModule

List_ExtensionModule message in JSON format

```
{
  "List":[{
    "ExtensionModule":{
      "id": ,
/*optional, integer type, module No.*/
      "address": ,
/*optional, read-only, integer type, module address, this node is only returned
by wired modules*/
      "linkageAddress": ,
/*optional, read-only, integer type, linked module address, this node is only
returned by wireless modules*/
      "name": "",
/*optional, string type, module name*/
      "type": "",
/*optional, read-only, string type, module type: "wiredZone"-wired zone module,
"wiredOutput"-wired output module, "wirelessOutput"-wireless output module,
"wirelessRecv"-wireless receiver module (wired module)*/
      "detailType": "",
/*optional, read-only, string type, detailed module type: wired zone module:
"eightWiredZone"-8-channel wired zone; wired output module: "localWiredOutput"-
local relay, "fourWiredOutput"-4-channel wired relay, "eightWiredOutput"-8-
channel wired relay; wireless output module: "twoWirelessOutput"-2-channel
wireless relay, "eightWirelessOutput"-8-channel wireless relay; wireless
receiver module (wired module): "RS485WirelessRecv"-RS-485 wireless receiver*/
      "model": "",
/*optional, read-only, string type, module model*/
```

```
    "version": "",
    /*optional, read-only, string type, module version*/
    "related": ,
    /*optional, boolean type, whether the extension module is linked to a physical
    extension module*/
    "seq": "",
    /*optional, string type, module serial No., this node can only be configured by
    wireless modules*/
    "checkTime":
    /*optional, integer type, offline duration, unit: hour, this node can only be
    configured by wireless modules*/
    }
  ]]
}
```

A.3.76 JSON_List_GlassBreakDetector

JSON message about parameters of all composite PIR (Passive Infrared) glass-break detectors

```
{
  "List": [{
    "GlassBreakDetector": {
      "zoneNo": ,
      /*optional, int, zone No. which starts from 0*/
      "LEDEnabled": ,
      /*optional, boolean, whether to enable the LED indicator*/
      "LEDLatchTime": ,
      /*optional, int, delay time of the LED indicator, unit: second*/
      "findMeEnabled": ,
      /*optional, boolean, whether to enable the Fine Me function*/
      "sensitivityLevel": "",
      /*optional, string, sensitivity level: "high", "auto"-automatic, "antiPet"-pet
      immune*/
      "checkEnabled": ,
      /*optional, boolean, whether to enable self-test: true=yes, false=no*/
      "distance": ,
      /*optional, int, distance, unit: meter*/
      "alarmLogic": "",
      /*optional, string, alarm logic: "and", "or"*/
      "heartBeatInterval":
      /*optional, int, heartbeat interval of the security control panel and the
      peripheral, unit: second*/
    }
  ]
}
```

A.3.77 JSON_List_ID

JSON message about zone ID list

```
{
  "List": [{
    "id":
/*int, zone No., which starts from 0; it is required when performing bypass or
bypass recovered on multiple zones; it is not required when performing bypass
or bypass recovered on a zone*/
  }]
}
```

A.3.78 JSON_List_IndoorDualTechnologyDetector

Message about the parameters of all indoor dual-technology detectors in JSON format.

```
{
  "List": [{
    "IndoorDualTechnologyDetector": {
      "zoneNo": ,
/*optional, int, zone No. which starts from 0*/
      "LEDEnabled": ,
/*optional, boolean, whether to enable the LED indicator*/
      "LEDLatchTime": ,
/*optional, boolean, delay time of the LED indicator, unit: second*/
      "findMeEnabled": ,
/*optional, boolean, whether to enable the Find Me function*/
      "sensitivityLevel": "",
/*optional, string, sensitivity level: "high", "auto", "antiPet"-pet immune*/
      "checkEnabled": ,
/*optional, boolean, whether to enable self-test*/
      "alarmLogic": "",
/*optional, string, alarm logic: "and", "or"*/
      "distance": "",
/*optional, string, adjustment distance: "high", "low"*/
      "heartBeatInterval":
/*optional, int, heartbeat interval of the security control panel and the
peripheral, unit: second*/
    }
  }]
}
```

A.3.79 JSON_List_IPAddress

List_IPAddress message in JSON format

```
{
  "List": [{
    "IPAddress": {
      "ipVersion": "",
      /*required, string type, version information of IP address: "v4"-IPv4, "v6"-IPv6*/
      "ipAddress": ""
      /*required, string type, IP address*/
    }
  }]
}
```

A.3.80 JSON_List_Keypad

JSON message about parameters of all keypads

```
{
  "List": [{
    "Keypad": {
      "id": ,
      /*optional, int, keypad No., it starts from 1*/
      "related": ,
      /*boolean, whether to link to the physical keypad. For PUT method, this node is optional; for GET method, this node is required*/
      "name": "",
      /*optional, string, keypad name*/
      "subSystem": ,
      /*optional, array, linked partitions, e.g., [1,2,3] indicates linking to partition 1, partition 2, and partition 3*/
      "buzzerEnabled": ,
      /*optional, boolean, whether to enable the buzzer: "true"-yes, "false"-no*/
      "swipingCardEnabled": ,
      /*optional, boolean, whether to enable the card swiping function: "true"-yes, "false"-no*/
      "armByKeyEnabled": ,
      /*optional, boolean, whether to enable the function of arming and disarming by pressing the key: "true"-yes (the virtual zone alarm will be enabled), "false"-no (the virtual zone alarm will be disabled). This function cannot be disabled for panic alarm and fire alarm*/
      "seq": "",
      /*optional, string, keypad serial No., this node is required when keypadAttrib is "wireless"*/
      "keypadAttrib": "",
      /*optional, read-only, string, keypad attribute: "wired", "wireless"*/
      "checkTime": ,
      /*optional, int, offline time, unit: hour*/
      "address": ,
      /*optional, read-only, integer type, keypad address, this node is only returned by wired keypads*/
      "type": ""
    }
  }]
}
```

```
/*optional, read-only, string, keypad model*/
    "version": "",
/*optional, read-only, string, keypad version*/
    "status": "",
/*optional, read-only, string, keypad status: "notRelated"-
unlinked, "online", "offline", "heartbeatAbnormal"-heartbeat exception*/
    "tamperEvident": ,
/*optional, read-only, boolean, tampering status: "true"-tampered, "false"-not
tampered*/
    "alarmVoicePromptEnabled": ,
/*optional, boolean, whether to enable voice prompt for panic alarm: true=yes
(the panic alarm of the keypad will be linked with voice prompt), false=no (the
panic alarm of the keypad will not be linked with voice prompt)*/
    "SetLEDCFG":{
/*optional, schedule configuration to enable the backlight*/
        "TimeCFGList":[{
            "TimeSegment":{
                "id": ,
/*optional, integer type, time period No.*/
                "enabled": ,
/*optional, boolean, whether to enable: "true"-yes, "false"-no*/
                "beginTime": "",
/*optional, string, start time of the time period, e.g., 09:00*/
                "endTime": ""
/*optional, string, end time of the time period, e.g., 11:30*/
            }
        ]
    },
    "attribute": "",
/*optional, string, read-only, keypad type: "LED" (LED keypad), "LCD" (LCD
keypad)*/
    "company": "",
/*optional, string, company name (for interface display), this node can be
configured when attribute is "LED"*/
    "phoneNo": "",
/*optional, string, phone number (for interface display), this node can be
configured when attribute is "LED"*/
    "activeDelayEnabled": true,
/*optional, boolean, whether to enable delayed activation*/
    "armAndDisarmAuthorityCfg": "tag",
/*optional, string, permission for arming/disarming the partition: "tag"-by
card, "password"-by password, "tagOrPassword"-by card or password,
"tagAndPassword"-by card and password*/
    "alarmBuzzerEnabled": true,
/*optional, boolean, whether to enable alarm buzzer: true, false*/
    "buttonBuzzerEnabled": true
/*optional, boolean, whether to enable button buzzer: true, false*/
}
}]
}
```

A.3.81 JSON_List_KeypadFaultProcessCfg

List_KeypadFaultProcessCfg message in JSON format

```
{
  "List":[{
    "KeypadFaultProcessCfg":{
      "JointLED":{
/*optional, keypad indicator output linked with the fault*/
        "tamper": ,
/*optional, boolean type, whether it is tampered*/
        "offline": ,
/*optional, boolean type, whether it is offline*/
        "lowPower": ,
/*optional, boolean type, whether it is low battery*/
        "trigger": ,
/*optional, boolean type, whether it is triggered by zone*/
        "power": ,
/*optional, boolean type, whether it is power supply fault*/
        "communication": ,
/*optional, boolean type, whether it is communication fault*/
        "all":
/*optional, boolean type, all faults. If this node is "true", all faults will
trigger keypad indicator output*/
      },
      "JointSound":{
/*optional, keypad sound output linked with the fault*/
        "tamper": ,
/*optional, boolean type, whether it is tampered*/
        "offline": ,
/*optional, boolean type, whether it is offline*/
        "lowPower": ,
/*optional, boolean type, whether it is low battery*/
        "trigger": ,
/*optional, boolean type, whether it is triggered by zone*/
        "power": ,
/*optional, boolean type, whether it is power supply fault*/
        "communication": ,
/*optional, boolean type, whether it is communication fault*/
        "all":
/*optional, boolean type, all faults. If this node is "true", all faults will
trigger keypad sound output*/
      }
    }
  }]
}
```


A.3.82 JSON_List_MagneticContact

Message about parameters of all composite magnetic contact detectors in JSON format.

```
{
  "List": [{
    "MagneticContact": {
      "zoneNo": ,
      /*optional, int, zone No. which starts from 1*/
      "LEDEnabled": ,
      /*optional, boolean, whether to enable the LED indicator*/
      "LEDLatchTime": ,
      /*optional, int, delay time of the LED indicator, unit: second */
      "findMeEnabled": ,
      /*optional, boolean, whether to enable the Find Me function*/
      "sensitivityLevel": "",
      /*optional, string, sensitivity level: "high", "middle", "low"*/
      "checkEnabled": ,
      /*optional, boolean, whether to enable self-test*/
      "magneticEnabled": ,
      /*optional, boolean, whether to enable the magnetic contact*/
      "InputList": [{
        "Input": {
          "enabled": ,
          /*required, boolean, whether to enable the input*/
          "id": ,
          /*required, int, No. which starts from 1*/
          "mode": "",
          /*optional, string, magnetic contact input mode: "normalOpen"-remain open,
          "normalClose"-remain close, "customize"-custom (when the number of pulses and
          the timeout reach the configured thresholds, the alarm will be triggered),
          "tamper"-tampering mode*/
          "pulseNum": ,
          /*optional, int, number of pulses (connecting to the rolling door)*/
          "timeout": ,
          /*optional, int, timeout (connecting to the rolling door), unit: second*/
          "name": "test"
          /*optional, string, input name*/
        }
      }],
      "alwaysActiveEnabled": true,
      /*optional, object, whether to keep the detector enabled when the zone is
      disarmed*/
      "heartBeatInterval": ,
      /*optional, int, heartbeat interval of the security control panel and the
      peripheral, unit: second*/
      "magneticType": "MagneticContact"
      /*optional, object, magnetic contact detector type: "MagneticContact"
      (composite magnetic contact), "ExternalMagnetic" (outdoor magnetic contact)*/
    }
  ]
}
```

```
  ]]  
}
```

A.3.83 JSON_List_Mail

List_Mail message in JSON format

```
{  
  "List": [{  
    "Mail": {  
      "id": ,  
/*required, integer type, email No.*/  
      "alarmTamperEnabled": ,  
/*optional, boolean type, whether to enable alarm and tampering event  
notification, "true, false"*/  
      "ipcDisconnetEnabled":  
/*optional, boolean type, whether to enable network camera disconnected event  
notification, "true, false"*/  
    }  
  }]  
}
```

A.3.84 JSON_List_ModuleInfo

List_ModuleInfo message in JSON format

```
{  
  "operType": "",  
/*required, string, operation type: "unlock"-unlock one or more modules,  
"unlockAll"-unlock all modules*/  
  "List": [{  
    "ModuleInfo": {  
      "id": ,  
/*required, integer, No.*/  
      "type": ""  
/*required, string, type: "localKeypad"-local keypad, "keypad"-keypad,  
"cardReader"-card reader, "localCardReader"-local card reader*/  
    }  
  }]  
}
```

A.3.85 JSON_List_ModuleLock

List_ModuleLock message in JSON format

```
{  
  "List": [{
```

```
"ModuleLock":{
  "id": ,
  /*required, integer type, No.*/
  "type": ,
  /*required, string type, module type: "localKeypad"-local keypad, "keypad"-
  keypad, "cardReader"-card reader, "localCardReader"-local card reader*/
  "status": ,
  /*optional, read-only, string type, status: "lock"-locked, "unlock"-unlocked*/
  "maxTryTimes": ,
  /*optional, integer type, maximum number of failed attempts*/
  "lockedTime": ,
  /*optional, integer type, locking duration, unit: second. If this node is set
  to 0, it indicates remaining locked*/
  "address": ,
  /*optional, read-only, integer type, module address, this node is only returned
  by wired modules*/
  "moduleAttrib":""
  /*optional, read-only, string type, module attribute: "wired", "wireless"*/
}
}]
}
```

A.3.86 JSON_List_Output

JSON message about the parameters of all logical relays

```
{
  "List":[{
    "Output":{
      "id": ,
      /*required, int, relay No., which starts from 0*/
      "name":"","
      /*optional, string type, relay name*/
      "related": ,
      /*read-only, boolean type, whether the relay is linked to the output module,
      for PUT method, this node is optional; for GET method, this node is required*/
      "outputModuleNo": ,
      /*read-only, int, linked output module No., this node is required only when the
      related is "true"*/
      "channelNo": ,
      /*read-only, int, channel No. of output module, this node is required only when
      the related is "true"*/
      "linkage":"","
      /*optional, string type, linked event types: "alarm", "arming", "disarming",
      "manualCtrl"-manual control, "system"-system event, "zone"*/
      "minorType": ,
      /*optional, array, minor event type: "zoneAlarmTamper"-zone alarm and tampered
      event, "exDevTamper"-peripheral device tampered, "hostTamper"-control panel
      tampered, "emergency"-panic alarm, "medical"-medical alarm, "fire"-fire alarm,
      "gas"-gas event. When linkage is "alarm", one or all minor event types can be
      selected. The data type of elements in the array is string*/
```

```
    "subSystem": ,
/*optional, array, linked partitions. When linkage is "alarm", "arming", or
"disarming", one or more partitions can be configured to link. For example,
[1,2,3] indicates that partition 1, partition 2, and partition 3 are linked*/
    "LinkageList":[{
/*optional, linked event list. If the device supports multiple linkage event
types linking to multiple minor event types and multiple partitions, this node
can be used. For compatibility, one linkage event types linking to multiple
minor event types and multiple partitions should also be supported*/
        "linkage": "",
/*optional, string, event types that can be linked to the relay: "alarm",
"arming", "disarming", "manualCtrl"-manual control, "zone"*/
        "alarmMinorType": ,
/*optional, array, minor event type: "zoneAlarmTamper"-zone alarm and tampered
event, "exDevTamper"-peripheral device tampered, "hostTamper"-control panel
tampered, "emergency"-panic alarm, "medical"-medical alarm, "fire"-fire alarm,
"gas"-gas event. When linkage is "alarm", one or all minor event types can be
selected. The data type of elements in the array is string*/
        "zoneEvent": ,
/*optional, string, zone event type, this node is valid when linkage contains
"zone". [1,3] indicates that zone 1 and zone 3 are enabled. If this node is not
configured when linkage contains "zone", it indicates enabling all zones*/
        "subSystem":
/*optional, array, linked partitions. When linkage is "alarm", "arming", or
"disarming", one or more partitions can be configured to link. For example,
[1,2,3] indicates that partition 1, partition 2, and partition 3 are linked*/
        },
        "durationConstOutputEnable": ,
/*optional, boolean, whether it supports configuring relay output duration:
"true"-not support (the duration cannot be configured and the relay will output
continuously), "false"-support (the duration can be configured and it can be
between 5s and 600s)*/
        "alarmEvent": "",
/*optional, string type, alarm event types, this node is valid only when the
node linkage is "alarm", e.g., "1,3" indicates that the zone 1 and zone 3 is
enabled; if this node is not configured, it indicates that all zones are
enabled*/
        "zoneEvent": ,
/*optional, string, zone event type, this node is valid when linkage contains
"zone". [1,3] indicates that zone 1 and zone 3 are enabled. If this node is not
configured when linkage contains "zone", it indicates enabling all zones*/
        "systemEvent": "",
/*optional, string type, system event types: "ACOff"-AC power off,
"networkAbnormal"-network fault, "dismantleAlarm"-tampering alarm, "hidAlarm"-
device blocking alarm, "preventMoveEvent"-motion event, "radarAbnormal"-radar
fault. This node is valid only when the node linkage is "system"; if this node
is not configured, it indicates that all system event types are selected*/
        "duration": ,
/*optional, int, output duration*/
        "alarmLine": ,
/*optional, int, trigger line No., it starts from 1*/
        "followModeEnabled": ,
```

```
/*optional, boolean type, whether to enable following mode. If this node is set
to "true": 1) duration is invalid for zone alarms, and after the zone or
trigger line is triggered, the relay can remain open until the zone or trigger
line is restored; 2) only zone or trigger line events can be linked, system
events and arming and disarming events cannot be linked*/
    "moduleType": "",
/*optional, read-only, string type, module type: "localWired"-local wired
module, "extendWired"-extended wired module, "localWireless"-local wireless
module, "extendWireless"-extended wireless module*/
    "scenarioType": ["alarm", "schedule", "arm", "disarm", "clearAlarm",
"fault", "manual"],
/*optional, object, scenario type, you can select one or multiple types*/
    "alarmCfg": {
/*optional, object, alarm configuration*/
        "alarmType": ["zoneAlarmAndTamper", "exDevTamper", "panicAlarm",
"medicalAlarm", "hostTamper", "fireAlarm"],
/*optional, object, alarm type, you can set one or multiple types*/
        "relayMode": "pulse",
/*optional, enum, relay mode: "pulse", "latch", subType:string*/
        "pulseDuration": 0,
/*optional, int, pulse interval, this node is valid when relayMode is "pulse"*/
        "contactStatus": "normalOpen",
/*optional, enum, relay status after events occurred: "normalOpen",
"normalClose", subType:string*/
        "supportAssociatedZone": [1, 2, 3],
/*optional, array, zones that can be linked, this node is valid when alarmType
is "zoneAlarmAndTamper", subType:int*/
        "associateZoneCfg": [1, 2, 3],
/*optional, array, zones that have been linked with this relay, this node is
valid when alarmType is "zoneAlarmAndTamper", subType:int*/
        "alarmLogic": "and",
/*optional, enum, alarm logic: "and", "or", this node is valid when alarmType
is "zoneAlarmAndTamper", subType:string*/
        "supportLinkageChannelID": [1, 2, 3],
/*optional, array, No.s of the channels that can be linked with network
cameras, this node is valid when alarmType is "zoneAlarmAndTamper",
subType:int*/
        "linkageChannelID": [1, 2]
/*optional, array, No.s of the channels to be linked with network cameras, this
node is valid when alarmType is "zoneAlarmAndTamper", subType:int, range:[1,4]*/
    },
    "address": ,
/*optional, read-only, int, module address, this node is only returned by wired
modules*/
    "linkageAddress": ,
/*optional, read-only, int, linked module address, this node is only returned
by wireless modules*/
    "moduleChannel": ,
/*optional, int, module channel No.*/
    "alarmLine": [1, 2, 3],
/*optional, array, No. of the trigger lines, which starts from 1*/
    "followModeEnabled": true,
```

```
/*optional, boolean, whether to enable following mode. If this node is set to
"true": 1) duration is invalid for zone alarms, and after the zone or trigger
line is triggered, the relay can remain open until the zone or trigger line is
restored; 2) only zone or trigger line events can be linked, system events and
arming and disarming events cannot be linked*/
    "modifiedOutputNo": 1,
/*optional, int, relay No. that has been modified*/
    "accessModuleType": "transmitter" ,
/*optional, string, access module type: "transmitter", "multiTransmitter",
"localTransmitter", "localRelay", "keypad"*/
    "relatedAccessModuleID": 1,
/*optional, int, No. of the linked access module*/
    "relayAttrib": "wired"
/*optional, string, relay attribute: "wired", "wireless" (default)*/
  }
  ]]
}
```

Remarks

Zone parameters configured by **alarmEvent** and **zoneEvent** should be the same and they both start from 0.

A.3.87 JSON_List_OutputModule

List_OutputModule message in JSON format

```
{
  "List":[{
    "OutputModule":{
      "id": ,
/*required, integer type, output module No., which starts from 0*/
      "name":"",
/*optional, string type, output module name*/
      "related": ,
/*boolean type, whether the output module is linked, for PUT method, this node
is optional; for GET method, this node is required*/
      "seq":"",
/*string type, output module serial No., this node is required only when
related is "true"*/
      "ChanList":[{
/*read-only, channel list of output module, this node is optional for PUT
method, but it is required for GET method*/
        "Chan":{
/*channel information of output module, this node is required only when the
node ChanList exists*/
          "channelNo":
/*integer type, channel No. of output module, this node is required only when
the node Chan exists*/
        }
      ]
    }
  ]
}
```

```
    "address": ,
    /*optional, read-only, integer type, module address, this node is only returned
    by wired modules*/
    "linkageAddress": ,
    /*optional, read-only, integer, linked module address, this node is only
    returned by wireless modules*/
    "attrib": "",
    /*optional, read-only, module attribute: "wired", "wireless". If this node is
    not returned, the default module attribute is "wireless"*/
    "checkTime":
    /*optional, integer type, offline duration, unit: hour*/
    }
  ]]
}
```

A.3.88 JSON_List_OutPutsModule

JSON message about linkage configuration parameters of all relays

```
{
  "List": [{
    "id": 1,
    /*optional, int, relay ID*/
    "OutPutModule": {
      "name": "",
      /*optional, string, relay name*/
      "OutPutCloseLinkage": [{
        /*optional, event linkage information when the relay is closed*/
        "linkage": "",
        /*optional, string, event types that can be linked to the relay: "alarm",
        "arming", "disarming", "manualCtrl"-manual control, "zone", "sysEvent"*/
        "alarmMinorType": ["zoneAlarmTamper", "exDevTamper", "hostTamper",
        "emergency", "medical", "fire", "gas"],
        /*optional, array, minor event type: "zoneAlarmTamper"-zone alarm and tampered
        event, "exDevTamper"-peripheral device tampered, "hostTamper"-control panel
        tampered, "emergency"-panic alarm, "medical"-medical alarm, "fire"-fire alarm,
        "gas"-gas event. When linkage is "alarm", one or all minor event types can be
        selected. The data type of elements in the array is string*/
        "sysEventMinorType": ["ACOutage", "lowVoltageOfBattery",
        "telephoneOffLine", "networkAbnormal", "wirelessNetworkAbnormal",
        "harddiskException", "485Exception", "mBusException", "3G4GSignalAbnormal",
        "moduleOffline"],
        /*optional, array, minor type of system event: "ACOutage"-AC power outage,
        "lowVoltageOfBattery"-low voltage of battery, "telephoneOffLine"-telephone
        offline, "networkAbnormal"-wired network disconnected,
        "wirelessNetworkAbnormal"-wireless network disconnected, "harddiskException"-
        hard disk exception, "485Exception"-system keyboard exception, "mBusException"-
        MBUS exception, "3G4GSignalAbnormal"-3G/4G signal exception, "moduleOffline"-
        module offline. This node is valid when linkage contains "sysEvent"*/
        "zoneEvent": [1, 3],
        /*optional, array, zone event type, this node is valid when linkage contains
```

```
"zone". [1,3] indicates that zone 1 and zone 3 are enabled. If this node is not
configured when linkage contains "zone", it indicates enabling all zones*/
    "subSystem": [1, 2, 3]
/*optional, array, linked partitions. When linkage is "alarm", "arming", or
"disarming", one or more partitions can be configured to link. For example,
[1,2,3] indicates that partition 1, partition 2, and partition 3 are linked*/
    },
    "OutputOpenLinkage": [{
/*optional, event linkage information when the relay is open*/
        "linkage": "",
/*optional, string, event types that can be linked to the relay: "alarm",
"arming", "disarming", "manualCtrl"-manual control, "zone", "sysEvent"*/
        "alarmMinorType": ["zoneAlarmTamper", "exDevTamper", "hostTamper",
"emergency", "medical", "fire", "gas"],
/*optional, array, minor event type: "zoneAlarmTamper"-zone alarm and tampered
event, "exDevTamper"-peripheral device tampered, "hostTamper"-control panel
tampered, "emergency"-panic alarm, "medical"-medical alarm, "fire"-fire alarm,
"gas"-gas event. When linkage is "alarm", one or all minor event types can be
selected. The data type of elements in the array is string*/
        "sysEventMinorType": ["ACOutage", "lowVoltageOfBattery",
"telephoneOffLine", "networkAbnormal", "wirelessNetworkAbnormal",
"harddiskException", "485Exception", "mBusException", "3G4GSignalAbnormal",
"moduleOffline"],
/*optional, array, minor type of system event: "ACOutage"-AC power outage,
"lowVoltageOfBattery"-low voltage of battery, "telephoneOffLine"-telephone
offline, "networkAbnormal"-wired network disconnected,
"wirelessNetworkAbnormal"-wireless network disconnected, "harddiskException"-
hard disk exception, "485Exception"-system keyboard exception, "mBusException"-
MBUS exception, "3G4GSignalAbnormal"-3G/4G signal exception, "moduleOffline"-
module offline. This node is valid when linkage contains "sysEvent"*/
        "zoneEvent": [1, 3],
/*optional, array, zone event type, this node is valid when linkage contains
"zone". [1,3] indicates that zone 1 and zone 3 are enabled. If this node is not
configured when linkage contains "zone", it indicates enabling all zones*/
        "subSystem": [1, 2, 3]
/*optional, array, linked partitions. When linkage is "alarm", "arming", or
"disarming", one or more partitions can be configured to link. For example,
[1,2,3] indicates that partition 1, partition 2, and partition 3 are linked*/
        },
        "duration":1,
/*optional, int, output duration of the relay, and the range is between 5 and
600 seconds*/
        "durationConstOutputEnable": true
/*optional, boolean, whether the duration (output duration of the relay) can be
configured: true-output duration cannot be configured (continuous output),
false-output duration can be configured*/
    }
    }
}
```


A.3.89 JSON_List_PanicButton

Message about the parameters of all panic buttons in JSON format.

```
{
  "List":[{
    "PanicButton":{
      "zoneNo": ,
      /*optional, int, zone No. which starts from 0*/
      "LEDEnabled": ,
      /*optional, boolean, whether to enable the LED indicator*/
      "LEDLatchTime": ,
      /*optional, int, delay time of the LED indicator, unit: second*/
      "findMeEnabled": ,
      /*optional, boolean, whether to enable the Find Me function*/
      "alarmMode":"","
      /*optional, string, alarm mode. When the panic alarm is triggered, the alarms
      of the corresponding alarm mode will be uploaded*/
      "accidentalPressProtection":"","
      /*optional, string, protection method to avoid triggering unintentionally*/
      "panicButtonType":"","
      /*optional, read-only, string, panic button type*/
      "heartBeatInterval": ,
      /*optional, int, heartbeat interval of the security control panel and the
      peripheral, unit: second*/
      "pollingOptionEnable":
      /*optional, boolean, whether to disable detecting heartbeat of the security
      control panel and the peripheral*/
      "workMode": "detector",
      /*optional, string, work mode: "detector", "autoControl" (control repeater)*/
      "supportAssociatedRelay": [1, 2, 3],
      /*optional, int, the repeaters that can be linked; this node is valid when the
      value of workMode is "autoControl"*/
      "associateRelayCfg": [1, 2, 3],
      /*optional, array, the repeaters that have been linked with the zone; this node
      is valid when the value of workMode is "autoControl"*/
      "triggerMode": ["longPress"],
      /*optional, string, trigger mode of the panic button*/
      "confirmAlarmInterval": 8
      /*optional, int, time interval for uploading acknowledgment alarm, unit: hour,
      range:[8,20]*/
    }
  ]
}
```

A.3.90 JSON_List_PassiveInfraredDetector

JSON message about the parameters of all PIR (Passive Infrared) detectors

```
{
  "List": [{
    "PassiveInfraredDetector": {
      "zoneNo": ,
/*optional, int, zone No. which starts from 0*/
      "LEDEnabled": ,
/*optional, boolean, whether to enable the LED indicator*/
      "LEDLatchTime": ,
/*optional, int, delay time of the LED indicator, unit: second*/
      "findMeEnabled": ,
/*optional, boolean, whether to enable the Fine Me function*/
      "sensitivityLevel": "",
/*optional, string, sensitivity level: "high", "auto", "antiPet"-pet immune*/
      "alwaysActiveEnabled": true,
/*optional, boolean, whether to enable keeping detector working even after
disarming*/
      "heartBeatInterval": 12,
/*optional, int, heartbeat interval of the security control panel and the
peripheral, unit: second*/
      "triggerNumLimited": 5,
/*optional, int, limited number of trigger times*/
      "detectorType": "normal"
/*object, optional, read-only, detector type: "normal" (PIR detector),
"ceiling"(PIR ceiling detector)*/
    }
  ]
}
```

A.3.91 JSON_List_Phone

List_Phone message in JSON format

```
{
  "List": [{
    "Phone": {
      "id": ,
/*required, integer type, phone No.*/
      "numbers": "",
/*required, string type, phone number*/
      "messageEnabled": ,
/*optional,boolean type, whether to enable message notification, "true, false"*/
      "callEnabled": ,
/*optional, boolean type, whether to enable phone call notification, "true,
false"*/
      "alarmTamperEnabled": ,
/*optional, boolean type, whether to enable alarm and tampering event
notification, "true, false"*/
      "lifeSecurityEnabled": ,
/*optional, boolean type, whether to enable life security event notification,
"true, false"*/
    }
  ]
}
```

```
    "systemStatusEnabled": ,
/*optional, boolean type, whether to enable system status event notification,
"true, false"*/
    "operateEventEnabled":
/*optional, boolean type, whether to enable operation event notification,
"true, false"*/
    }
  ]]
}
```

A.3.92 JSON_List_PhoneAnvanded

JSON message about the advanced notification parameters of all phone numbers

```
{
  "List":[{
    "PhoneAnvanded":{
      "id": ,
/*required, integer type, phone number ID*/
      "numbers":"",
/*required, string type, phone number*/
      "messageEnabled": ,
/*optional, boolean type, whether to enable SMS notification*/
      "callEnabled": ,
/*optional, boolean type, whether to enable phone notification*/
      "Message":{
/*this node is valid when "messageEnabled" is "true"*/
        "alarmTamperEnabled": ,
/*optional, boolean type, whether to enable alarm and tampering event
notification*/
        "lifeSecurityEnabled": ,
/*optional, boolean type, whether to enable life safety event notification*/
        "systemStatusEnabled": ,
/*optional, boolean type, whether to enable system status event notification*/
        "operateEventEnabled": ,
/*optional, boolean type, whether to enable operation event notification*/
        "zoneAlarmTamperEnabled": ,
/*optional, boolean, whether to enable alarm and tampering event notification
of the supported zone: "true"-yes, "false"-no*/
        "exDevTamperEventEnabled": ,
/*optional, boolean, whether to enable peripheral tampering alarm notification:
"true"-yes, "false"-no*/
        "hostTamperEventEnabled": ,
/*optional, boolean, whether to enable tampering alarm notification of security
control panel: "true"-yes, "false"-no*/
        "emergencyEventEnabled": ,
/*optional, boolean, whether to enable panic alarm notification: "true"-yes,
"false"-no*/
        "medicalEventEnabled": ,
/*optional, boolean, whether to enable medical alarm notification: "true"-yes,
"false"-no*/
```

```
    "gasEventEnabled": ,
/*optional, boolean, whether to enable gas alarm notification: "true"-yes,
"false"-no*/
    "fireEventEnabled": ,
/*optional, boolean, whether to enable fire alarm notification: "true"-yes,
"false"-no*/
    "hostStatusEventEnabled": ,
/*optional, boolean, whether to enable notification of security control panel
status: "true"-yes, "false"-no*/
    "exDevStatusEventEnabled": ,
/*optional, boolean, whether to enable peripheral status notification: "true"-
yes, "false"-no*/
    "detectorStatusEventEnabled": ,
/*optional, boolean, whether to enable detector status notification: "true"-
yes, "false"-no*/
    "intelligentAlarmEnable": ,
/*optional, boolean, whether to enable smart alarm notification: "true"-yes,
"false"-no*/
    "arm": ,
/*optional, array, arming permission, e.g., [1,2,3] indicates having permission
to arm partition 1, partition 2, and partition 3*/
    "disarm": ,
/*optional, array, disarming permission, e.g., [1,2,3] indicates having
permission to disarm partition 1, partition 2, and partition 3*/
    "clearAlarm": ,
/*optional, array, alarm clearing permission, e.g., [1,2,3] indicates having
permission to clear alarms of partition 1, partition 2, and partition 3*/
    "timeFilterEnabled": true,
/*optional, boolean, whether to enable uploading event details by message only
within the configured period: true, false(uploading the messages all the time)*/
    "startTime": "10:00:00",
/*optional, time, start time, this node is valid when timeFilterEnabled is
true*/
    "endTime": "16:00:00",
/*optional, time, end time, this node is valid when timeFilterEnabled is true*/
    "WeekPlanCfg": [
/*optional, array, week schedule information, range:[1,7], this node is valid
when timeFilterEnabled is true*/
        {
            "dayOfWeek": 1,
/*required, int, day of the week, range:[1,7]*/
            "TimeRange": [
/*required, array, period, range:[1,8]*/
                {
                    "startTime": "10:00:00",
/*required, time, start time*/
                    "endTime": "16:00:00"
/*required, time, end time*/
                }
            ]
        }
    ]
}
```

```
    },
    "Call":{
/*this node is valid when "callEnabled" is "true"*/
        "alarmTamperEnabled": ,
/*optional, boolean type, whether to enable alarm and tampering event
notification*/
        "lifeSecurityEnabled": ,
/*optional, boolean type, whether to enable life safety event notification*/
        "systemStatusEnabled": ,
/*optional, boolean type, whether to enable system status event notification*/
        "operateEventEnabled": ,
/*optional, boolean type, whether to enable operation event notification*/
        "zoneAlarmTamperEnabled": ,
/*optional, boolean, whether to enable alarm and tampering event notification
of the supported zone: "true"-yes, "false"-no*/
        "exDevTamperEventEnabled": ,
/*optional, boolean, whether to enable peripheral tampering alarm notification:
"true"-yes, "false"-no*/
        "hostTamperEventEnabled": ,
/*optional, boolean, whether to enable tampering alarm notification of security
control panel: "true"-yes, "false"-no*/
        "emergencyEventEnabled": ,
/*optional, boolean, whether to enable panic alarm notification: "true"-yes,
"false"-no*/
        "medicalEventEnabled": ,
/*optional, boolean, whether to enable medical alarm notification: "true"-yes,
"false"-no*/
        "gasEventEnabled": ,
/*optional, boolean, whether to enable gas alarm notification: "true"-yes,
"false"-no*/
        "fireEventEnabled": ,
/*optional, boolean, whether to enable fire alarm notification: "true"-yes,
"false"-no*/
        "hostStatusEventEnabled": ,
/*optional, boolean, whether to enable notification of security control panel
status: "true"-yes, "false"-no*/
        "exDevStatusEventEnabled": ,
/*optional, boolean, whether to enable peripheral status notification: "true"-
yes, "false"-no*/
        "detectorStatusEventEnabled": ,
/*optional, boolean, whether to enable detector status notification: "true"-
yes, "false"-no*/
        "intelligentAlarmEnable": ,
/*optional, boolean, whether to enable smart alarm notification: "true"-yes,
"false"-no*/
        "numbersOfCalls": ,
/*optional, integer, phone call times*/
        "timeFilterEnabled": true,
/*optional, boolean, whether to enable uploading event details by call only
within the configured period: true, false(uploading the calls all the time)*/
        "startTime": "10:00:00",
/*optional, time, start time, this node is valid when timeFilterEnabled is
```

```
true*/
    "endTime": "16:00:00",
/*optional, time, end time, this node is valid when timeFilterEnabled is true*/
    "WeekPlanCfg": [
/*optional, array, week schedule information, range:[1,7], this node is valid
when timeFilterEnabled is true*/
        {
            "dayOfWeek": 1,
/*required, int, day of the week, range:[1,7]*/
            "TimeRange": [
/*required, array, period, range:[1,8]*/
                {
                    "startTime": "10:00:00",
/*required, time, start time*/
                    "endTime": "16:00:00"
/*required, time, end time*/
                }
            ]
        }
    ]
}
]]
}
```

A.3.93 JSON_List_pircam

JSON message about pircam parameters of all zones

```
{
  "List":[{
    "pircam":{
      "enabled": ,
/*required, boolean, whether to enable pircam (detector equipped with camera)
configuration*/
      "channelNo": ,
/*optional, integer, pircam (detector equipped with camera) channel No.*/
      "picColorResolution": "",
/*optional, string, picture resolution*/
      "zoneName": "",
/*optional, read-only, string, zone name*/
      "zoneNo": ,
/*optional, integer, zone No. which starts from 0*/
      "camEnable": ,
/*optional, boolean, whether to enable the camera*/
      "picNum": ,
/*optional, integer, number of pictures. For pictures whose resolution is
640*480, this field is between 0 and 10; for other resolution, this field is
between 0 and 20*/
      "picInterval": ,
/*optional, integer, picture capture interval, unit: second*/
```

```
"picQoc": "",
/*optional, string, picture quality: "20%", "40%", "60%"/
"detectInterval": ,
/*optional, detection time interval (valid alarm duration)*/
"LEDEnable": ,
/*optional, boolean, whether to enable LED*/
"LEDLatchTime": ,
/*optional, int, delay time of the LED indicator, unit: second*/
"devDetectEnable": ,
/*optional, boolean, whether to enable device detection*/
"signGainCfg": "",
/*optional, string, pircam detector gain*/
"petImmueFilter": "",
/*optional, boolean, whether the pet can trigger the pircam alarm: true=yes*/
"pulseFilterCfg": ,
/*optional, int, number of impulse waves*/
"holdOffTime": ,
/*optional, int, delay time, 0 indicates no delay, unit: second*/
"jpegModeCfg": "",
/*optional, string, format of pictures captured by pircam*/
"sensitivityLevel": "",
/*optional, string, sensitivity level: "high", "auto", "antiPet"-pet immune*/
"findMeEnabled": ,
/*optional, boolean, whether to enable the Find Me function*/
"climeEnabled": ,
/*optional, boolean, whether to enable muting*/
"videoResolution": "",
/*optional, string, video resolution*/
"picMode": "",
/*optional, string, picture mode: "blackAndWhite"-black and white, "color"/
"videoTime": ,
/*optional, int, video duration, unit: second*/
"triggerTime": ,
/*optional, int, alarm interval, unit: second*/
"triggerNum": ,
/*optional, int, alarm triggering times. When the number of detector alarms
reaches the configured limit and there is no alarm uploaded by any other zone
in the whole security control system, the pircam will not trigger alarm again;
if there are alarms uploaded by other zones, this field will be set to 0 and
the alarm triggering times will be calculated again*/
"linkageCaptureType": ,
/*optional, string, linkage action after the alarm is triggered: "picture"-the
pircam will capture pictures, "4seconds video"-record a 4-second video,
"8seconds video"-record an 8-second video*/
"heartBeatInterval": ,
/*optional, int, heartbeat interval of the security control panel and the
peripheral, unit: second*/
"frame": ,
/*optional, int, frame rate*/
"alwaysActiveEnabled": true,
/*optional, boolean, whether to enable detector working all the time even if
the zone is disarmed*/
```

```
    "operateTime": "1970-01-01T00:00:00+08:00",
    /*optional, datetime, operation time in ISO 8601 time format, this node should
    be used together with "nextArmTime"*/
    "nextArmTime": 3
    /*optional, int, time before the next arming: 3, 6, 12, 24, 48, 96, 192, unit:
    h*/
  }
}]
}
```

A.3.94 JSON_List_PSTNCfg

JSON message about configuration parameters of all phone notifications via PSTN

```
{
  "List":[{
    "PSTNCfg":{
      "id": ,
      /*required, int, uploading index*/
      "enabled": ,
      /*required, boolean, whether to enable*/
      "name": "",
      /*optional, string, alarm receiving center name*/
      "phoneNum": "",
      /*optional, string, phone number of alarm receiving center*/
      "repeatCall": "",
      /*optional, int, number of repeated callings*/
      "protocol": "",
      /*optional, string, communication protocol: "CID"-CID communication*/
      "transMode": ,
      /*optional, int, transmission mode: 0-DTMF 5/S, 1-DTMF 10/S*/
      "receiverId": "",
      /*optional, string, receiver user name*/
      "reportPeriod": ,
      /*optional, int, test report uploading period, unit: hour*/
      "reportPeriodEnabled": ,
      /*optional, boolean, whether to enable test report uploading period*/
      "firstReportTime":
      /*optional, int, duration from launching the device to uploading the first test
      report, unit: minute*/
    }
  ]
}
```

A.3.95 JSON_List_PublicSubSys

List_PublicSubSys message in JSON format


```
{
  "List": [{
    "PublicSubSys": {
      "id": ,
      /*required, integer, partition No., it starts from 1*/
      "enabled": ,
      /*required, boolean, whether to enable the public partition*/
      "linkageCommon":
      /*optional, array, normal partitions linked to the public partition. For
      example, [1,2] indicates that normal partitions 1 and 2 are linked to the
      public partition*/
    }
  }]
}
```

A.3.96 JSON_List_RemoteCtrl

List_RemoteCtrl message in JSON format

```
{
  "List": [{
    "RemoteCtrl": {
      "id": ,
      /*required, integer type, keyfob No., which starts from 1*/
      "enabled": ,
      /*required, boolean type, whether to enable keyfob*/
      "seq": "",
      /*required, string type, keyfob serial No.*/
      "name": "",
      /*optional, string type, keyfob name*/
      "factory": "",
      /*optional, string type, keyfob manufacturer*/
      "right": [],
      /*optional, array with string type, keyfob permission, if no value is assigned
      to the node, the device adopts the default permissions*/
      "SelKeyList": [{
        /*optional, key list, if no value is assigned to the node, the device adopts
        the default one*/
        "SelKey": {
          /*custom key, this node is required when the node SelKeyList exists*/
          "key": ,
          /*integer type, key No., this node is required when the node SelKeyList exists*/
          "func": "",
          /*integer type, keys' function, this node is required when the node SelKeyList
          exists*/
          "outputNo":
          /*integer type, relay No., this node is required when the fun is
          "operateOutputs"*/
        }
      }],
    }
  }],
}
```

```
"CombKeyList":[{
/*optional, combined key list, if no value is assigned to the node, the device
adopts the default keys*/
  "CombKey":{
/*combined keys, this node is required only when the CombKeyList exists*/
    "keys": "",
/*string, combined key, this node is required only when the CombKey exists*/
    "func": "",
/*string, combined keys' function, this node is required only when the CombKey
exists and its value can be empty*/
    "outputNo":
      }
  }
}],
"subSystem": ,
/*optional, array, linked partitions. For example, [1,2,3] indicates linking to
partition 1, partition 2, and partition 3*/
"relatedNetUserName": "",
/*optional, string, linked network user name*/
"alarmVoicePromptEnabled":
/*optional, boolean, whether to enable voice prompt for panic alarm: true=yes
(the panic alarm of the keyfob will have linked voice prompt), false=no (the
panic alarm of the keyfob will not have linked voice prompt)*/
  }
}]
}
```

A.3.97 JSON_List_Repeater

List_Repeater message in JSON format

```
{
  "List": [{
    "Repeater": {
      "id": ,
/*required, integer type, repeater No., which starts from 1*/
      "name": "",
/*optional, string type, repeater name*/
      "related": ,
/*boolean type, whether the repeater is linked, for PUT method, this node is
optional; for GET method, this node is required*/
      "seq": "",
/*string type, repeater serial No., this node is required when related is
"true"*/
      "linkageAddress": ,
/*optional, integer type, linked module address, this node is only returned by
wireless modules*/
      "checkTime":
/*optional, integer type, offline duration, unit: hour*/
    }
  ]
}
```

```
    }  
}
```

A.3.98 JSON_List_Siren

JSON message about parameters of all sirens

```
{  
  "List": [{  
    "Siren": {  
      "id": ,  
      /*required, int, siren No., which starts from 1*/  
      "name": "",  
      /*optional, string, siren name*/  
      "volume": ,  
      /*optional, int, siren volume*/  
      "related": ,  
      /*boolean, whether the physical siren is linked, for PUT method, this node is  
optional; but for GET method, this node is required*/  
      "seq": "",  
      /*string, siren serial No., this node is required only when related is "true"*/  
      "address": ,  
      /*optional, read-only, int, module address, this node is only returned by wired  
modules*/  
      "linkageAddress": ,  
      /*optional, read-only, int, linked module address, this node is only returned  
by wireless modules. If related is "false", this node can be configured*/  
      "checkTime": ,  
      /*optional, int, offline duration, unit: hour*/  
      "sirenAttrib": "",  
      /*optional, read-only, string, siren attribute: "wired", "wireless"*/  
      "linkage": "",  
      /*optional, string, event linkage type*/  
      "zoneEvent": "",  
      /*optional, string, zone event type, this node is valid when linkage contains  
"zone". [1,3] indicates that zone 1 and zone 3 are enabled. If this node is not  
configured when linkage contains "zone", it indicates enabling all zones*/  
      "subSystem": ,  
      /*optional, array, partitions linked to the siren. For example, [1,2,3]  
indicates that partition 1, partition 2, and partition 3 are linked to the  
siren*/  
      "LinkageList": [{  
      /*optional, linked event list. If the device supports linking multiple event  
types with multiple minor event types and multiple partitions, this node can be  
configured. For compatibility, linking an event type with multiple minor event  
types and multiple partitions should also be supported*/  
        "linkage": "",  
        /*optional, string, linked event type*/  
        "zoneEvent": [1,3],  
        /*optional, string, zone event type. This node is valid when linkage contains  
"zone". For example, [1,3] indicates enabling zone 1 and zone 3. If this node
```

```
is not configured, it indicates enabling all zones*/
    "subSystem": [1, 2, 3]
/*optional, array, linked partitions when linkage is "alarm", "arming", or
"disarming". For example, [1,2,3] indicates linking with partition 1, partition
2, and partition 3 are linked*/
    },
    "LEDEnabled": ,
/*optional, boolean, whether to enable the LED indicator*/
    "LEDLatchTime": ,
/*optional, int, delay time of the LED indicator, unit: second*/
    "findMeEnabled": ,
/*optional, boolean, whether to enable the Find Me function*/
    "location": "",
/*optional, string, siren location: "outdoor", "indoor"*/
    "ArmAndDisarmIndicatorCfg": {
/*optional, indicator settings for arming and disarming*/
    "LEDEnabled": ,
/*optional, boolean, whether to enable the LED indicator to flicker for arming
and disarming*/
    "LEDTimes": ,
/*optional, int, LED indicator flickering times*/
    "LEDFrequency": ,
/*optional, int, LED indicator flickering frequency, unit: Hz*/
    "buzzerEnabled": ,
/*optional, boolean, whether to enable the buzzer to buzz for arming and
disarming*/
    "buzzerTimes":
/*optional, int, buzzer buzzing times*/
    },
    "company": "",
/*optional, string, read-only, company name: "pyronix", "longhorn",
"hikvision". When the value of company is "hikvision", the siren supports test*/
    "tamperEnabled": ,
/*optional, boolean, whether to enable siren tampering*/
    "tryAlarmEnabled": ,
/*optional, boolean, whether to enable alarm attempt*/
    "preRegisterEnabled": ,
/*optional, boolean, whether to enable pre-registration*/
    "buzzEnabled": ,
/*optional, boolean, whether to enable linking the buzzer to buzz when the
alarm is triggered*/
    "disarmTamperEnabled": true,
/*optional, boolean, whether to enable tampering alarm when it is disarmed,
this node is valid when buzzEnabled is true*/
    "alarmStrobeFlashEnabled": ,
/*optional, boolean, whether to enable linking the alarm lamp to flicker when
the alarm is triggered*/
    "sounderAlarmDuration": ,
/*optional, int, siren's output duration when the alarm is triggered, unit:
second*/
    "heartBeatInterval":
/*optional, int, heartbeat interval of the security control panel and the
```

```
peripheral, unit: second*/
    "sirenColor": "red",
/*optional, string, siren color*/
    "alarmLinkedEventCfg": ["alarmTrigger"],
/*optional, array of string, linked event configuration*/
    "accessModuleType": "localSiren"
/*optional, string, access module type: "localSiren"*/
    }
  ]]
}
```

A.3.99 JSON_List_SlimMagneticContact

Message about the parameters of all slim magnetic contact detectors in JSON format.

```
{
  "List": [{
    "SlimMagneticContact": {
      "zoneNo": ,
/*optional, int, zone No. which starts from 0*/
      "LEDEnabled": ,
/*optional, boolean, whether to enable the LED indicator*/
      "LEDLatchTime": ,
/*optional, int, delay time of the LED indicator, unit: second*/
      "findMeEnabled": ,
/*optional, boolean, whether to enable the Find Me function*/
      "heartBeatInterval":
/*optional, boolean, heartbeat interval of the security control panel and the
peripheral, unit: second*/
    }
  ]
}
```

A.3.100 JSON_List_SubSys

List_SubSys message in JSON format

```
{
  "List": [{
    "SubSys": {
      "id": ,
/*required, integer type, partition No., it starts from 1*/
      "enabled": ,
/*required, boolean type, whether to enable the partition*/
      "linkageZones": ,
/*optional, array, zones linked to the partition*/
      "oneKeyArmEnabled": ,
/*optional, boolean type, whether to enable one-touch arming*/
      "isPublicSubSys":
```

```
/*optional, boolean, whether the partition is a public partition*/
    }
  ]]
}
```

A.3.101 JSON_List_SubSysTime

List_SubSysTime message in JSON format

```
{
  "List": [{
    "SubSysTime": {
      "id": ,
      /*required, integer type, partition No., which starts from 1*/
      "enteyDelay1": ,
      /*optional, integer type, entrance delay 1, unit: second*/
      "enteyDelay2": ,
      /*optional, integer type, entrance delay 2, unit: second*/
      "exitDelay": ,
      /*optional, integer type, exiting delay time, unit: second*/
      "autoArmingEnable": ,
      /*optional, boolean type, whether to enable automatic arming, "true"-yes,
      "false"-no*/
      "autoArming": "",
      /*time of automatic arming enabled, which is in 24-hour system and adopts the
      device local time zone, it is valid only when the autoArmingEnable is "true"*/
      "autoDisarmingEnable": ,
      /*optional, boolean type, whether to enable automatic disarming, "true"-yes,
      "false"-no*/
      "autoDisarming": "",
      /*time of automatic disarming enabled, which is in 24-hour system and adopts
      the device local time zone, it is valid only when the autoDisarmingEnable is
      "true"*/
      "lateWarningEnable": ,
      /*optional, boolean type, whether to enable late warning*/
      "lateWarning": "",
      /*time of late warning enabled, which is in 24-hour system and adopts the
      device local time zone, it is valid only when the lateWarningEnable is "true"*/
      "weekendsExceptEnable": ,
      /*optional, boolean type, whether to enable automatic arming or disarming
      except weekend, "true"-yes, "false"-no, and it is valid only when
      autoArmingEnable is "true" or autoDisarmingEnable is "true"*/
      "WeekCfg": [{
        "dayOfWeek": ""
        /*optional, string, day of the week: "Monday", "Tuesday", "Wednesday",
        "Thursday", "Friday", "Saturday", "Sunday". This node is valid when
        weekendsExceptEnable is "true"*/
      }],
      "HolidayExceptionsCfg": {
        "enable": ,
        /*optional, boolean, whether to support configuring holiday time period: "true"-
```

```
yes, "false"-no. Up to six holiday time periods are supported, and currently
only months and days of holiday time periods are required*/
    "HolidayCfg":[{
        "startDate":"",
/*optional, string, start date, e.g., "05-09" refers to May 9*/
        "endDate":""
/*optional, string, end date, e.g., "05-09" refers to May 9*/
    }],
    "perimeterDelayTime": ,
/*optional, delay time of perimeter zone alarm, unit: second, when the
perimeter zone alarm is triggered, the siren output will not start until the
delay time is ended (during the delay time, do not disarm or clear alarm,
otherwise, there is also no siren output)*/
    "sounderTime": ,
/*optional, audio alarm duration, unit: second*/
    "heartbeatInterval":"",
/*optional, string type, heartbeat interval of security control panel and
peripherals*/
    "ACcheckTime":
/*optional, integer type, AC detection time*/
    }
}]]
}
```

A.3.102 JSON_List_UserCfg

List_UserCfg message in JSON format

```
{
    "List":[{
        "UserCfg":{
            "id": ,
/*integer type, user No., it is optional for POST method, but it is required
for other methods*/
            "userName":"",
/*required, string type, user name*/
            "password":"",
/*required, string type, user password*/
            "keypadPassword":"",
/*optional, string type, keyboard password, write-only*/
            "messageSendEnabled":,
/*optional, boolean type, whether to enable message notification, "true,
false"*/
            "bypassEnabled":,
/*optional, boolean type, whether to enable bypass or bypass recovery*/
            "duressEnabled":,
/*optional, boolean type, whether to enable duress alarm*/
            "MacList":[{
                "Mac":""
/*optional, MAC address bound by the user*/
            }]
```

```
    }],
    "IpAddrList": [{
/*optional*/
        "IpAddr": {
            "ipVersion": "",
/*optional, string type, IP address version information: "v4"-IPv4, "v6"-IPv6*/
            "ipAddress": ""
/*optional, IPV4 or IPV6 address*/
        }
    }]
}
}]
}
```

A.3.103 JSON_List_Zone

JSON message about all zones' parameters

```
{
  "List": [{
/*required, array, list of all zones*/
    "Zone": {
/*optional, object, parameters of a zone, see details in JSON_Zone*/
    }
  }]
}
```

See Also

[JSON_Zone](#)

A.3.104 JSON_MagneticContact

Message about the parameters of the composite magnetic contact detector of a specific zone in JSON format.

```
{
  "MagneticContact": {
    "LEDEnabled": ,
/*optional, boolean, whether to enable the LED indicator*/
    "LEDLatchTime": ,
/*optional, int, delay time of the LED indicator, unit: second*/
    "findMeEnabled": ,
/*optional, boolean, whether to enable the Find Me function*/
    "sensitivityLevel": "",
/*optional, string, sensitivity level: "high", "middle", "low"*/
    "checkEnabled": ,
/*optional, boolean, whether to enable self-test*/
    "magneticEnabled": ,
  }
}
```



```
/*optional, boolean, whether to enable the magnetic contact*/
  "InputList":[{
    "Input":{
      "enabled": ,
/*required, boolean, whether to enable the input*/
      "id": ,
/*optional, int, No. which starts from 1*/
      "mode":"","
/*optional, string, magnetic contact input mode: "normalOpen"-remain open,
"normalClose"-remain close, "customize"-custom (when the number of pulses and
the timeout reach the configured thresholds, the alarm will be triggered),
"tamper"-tampering mode*/
      "pulseNum": ,
/*optional, int, number of pulses (connecting to the rolling door)*/
      "timeout": ,
/*optional, int, timeout (connecting to the rolling door), unit: second*/
      "name": "test"
/*optional, string, input name*/
    }
  }],
  "alwaysActiveEnabled": true,
/*optional, object, whether to keep the detector enabled when the zone is
disarmed*/
  "heartBeatInterval": ,
/*optional, int, heartbeat interval of the security control panel and the
peripheral, unit: second*/
  "magneticType": "MagneticContact"
/*optional, object, magnetic contact detector type: "MagneticContact"
(composite magnetic contact), "ExternalMagnetic" (outdoor magnetic contact)*/
}
}
```

A.3.105 JSON_MagneticContactCap

JSON message about the configuration capability of the composite magnetic contact detector

```
{
  "MagneticContactCap": {
    "zoneNo": {
/*optional, int, values that can be configured as the zone No.*/
      "@opt":[1,3,5]
    },
    "supportZoneType":{
/*optional, string, zone types supported by the peripheral: "Instant"-instant
zone, "Delay"-delay zone, "Follow"-follow zone, "Perimeter"-perimeter zone,
"24hNoSound"-24-hour silent zone, "Emergency"-panic zone, "Fire"-fire zone,
"Gas"-gas zone, "Medical"-medical zone, "Timeout"-timeout zone, "Non-Alarm"-
disabled zone, "Key"-key zone, "24hSound"-24-hour annunciating zone. When
switching zone type, zone types supported by the peripheral can be obtained*/
      "@opt":["Instant","Delay","Follow"]
    },
  },
}
```

```
"heartBeatInterval":{
/*optional, int, heartbeat interval of the security control panel and the
peripheral, unit: second*/
"@opt":[5,10,20,30]
},
"LEDEnabled": {
/*optional, boolean, whether to enable the LED indicator*/
"@opt":[true,false]
},
"LEDLatchTime": {
/*optional, int, delay time of the LED indicator, unit: second*/
"@min":0,
"@max":0
},
"findMeEnabled": {
/*optional, boolean, whether to enable the Find Me function*/
"@opt":[true,false]
},
"sensitivityLevel": {
/*optional, string, sensitivity level: "high","middle","low"*/
"@opt":["high","middle","low"]
},
"checkEnabled":{
/*optional, boolean, whether to enable self-test*/
"@opt":[true,false]
},
"magneticEnabled":{
/*optional, boolean, whether to enable the magnetic contact*/
"@opt":[true,false]
},
"InputList":{
"maxInputNum":2,
/*optional, int, the maximum number of inputs*/
"Input":{
"enabled":{
/*required, boolean, whether to enable the input*/
"@opt":[true,false]
},
"id":{
/*required, boolean, No. which starts from 1*/
"@min":1,
"@max":2
},
"mode":{
/*optional, string, magnetic contact input mode: "normalOpen"-remain open,
"normalClose"-remain close, "customize"-custom (when the number of pulses and
the timeout reach the configured thresholds, the alarm will be triggered),
"tamper"-tampering mode*/
"@opt":["normalOpen","normalClose","customize","tamper"]
},
"pulseNum":{
/*optional, int, number of pulses (connecting to the rolling door)*/
```

```
        "@opt": [2, 4, 6]
      },
      "timeout": {
        /*optional, int, timeout (connecting to the rolling door), unit: second*/
        "@opt": [10, 20, 30]
      },
      "name": {
        /*optional, object, input name*/
        "@min": 1,
        /*optional, int, the minimum value*/
        "@max": 48
        /*optional, int, the maximum value*/
      }
    },
    "isSupportSignalTest": true,
    /*optional, read-only, boolean, whether it supports signal strength detection,
    if this node is not returned or if the value is false, it indicates that this
    function is not supported*/
    "isSupportZoneTest": true,
    /*optional, boolean, whether zone test is supported, if this node is not
    returned or if the value is false, it indicates that this function is not
    supported*/
    "isSupportFindMe": true,
    /*optional, boolean, whether it supports FindMe detection, if this node is not
    returned or if the value is false, it indicates that this function is not
    supported*/
    "magneticType": {
      /*optional, object, magnetic contact detector type: "MagneticContact"
      (composite magnetic contact), "ExternalMagnetic" (outdoor magnetic contact)*/
      "@opt": ["MagneticContact", "ExternalMagnetic"]
    },
    "ExternalMagneticNode": {
      /*optional, object, node for outdoor magnetic contact detector*/
      "supportZoneType": {
        /*optional, object, zone types supported by outdoor magnetic contact detector:
        "Instant"-instant zone, "Delay"-delay zone, "Follow"-follow zone, "Perimeter"-
        perimeter zone, "24hNoSound"-24-hour silent zone, "Emergency"-panic zone,
        "Fire"-fire zone, "Gas"-gas zone, "Medical"-medical zone, "Timeout"-timeout
        zone, "Non-Alarm"-disabled zone, "Key"-key zone, "24hSound"-24-hour
        annunciating zone*/
        "@opt": ["Instant", "Delay", "24h", "Non-Alarm"]
      }
    },
    "isSupportFinalDoorExit": true
    /*optional, boolean, whether it supports Final Door Exit function. If Final
    Door Exit is enabled on a door magnetic contact (a detector), the area will be
    armed immediately after the magnetic contact detects door opening and door
    closing. If disabled, the area has to wait until a fixed countdown is over
    before being armed; if the node is not returned or the value is false, it
    indicates the function is not supported*/
```

```
}  
}
```

A.3.106 JSON_Mail

Mail message in JSON format

```
{  
  "Mail":{  
    "id": ,  
    /*required, integer type, email No.*/  
    "alarmTamperEnabled": ,  
    /*optional, boolean type, whether to enable alarm and tampering event  
notification, "true, false"*/  
    "ipcDisconnetEnabled":  
    /*optional, boolean type, whether to enable network camera disconnected event  
notification, "true, false"*/  
  }  
}
```

A.3.107 JSON_MailCap

MailCap message in JSON format

```
{  
  "MailCap":{  
    "alarmTamperEnabled": ,  
    /*optional, boolean type, whether to enable alarm and tampering event  
notification, "true, false"*/  
    "ipcDisconnetEnabled":  
    /*optional, boolean type, whether to enable network camera disconnected event  
notification, "true, false"*/  
  }  
}
```

A.3.108 JSON_Manage

JSON message about the configuration parameters of security control system

```
{  
  "Manage":{  
    "setterEnabled": ,  
    /*optional, boolean, whether to enable installer configuration, "true, false",  
if it is not supported, the installer cannot remotely connect to security  
control panel and program or operate the panel via keypad*/  
    "wirelessSuperVision": ,  
    /*optional, boolean, whether to enable wireless peripherals management, "true,
```

```
false", if it is enabled, you can check the peripherals status via security
control panel*/
    "zonesfaultArming": ,
/*optional, boolean, whether to enable forbidding arming zone when fault
occurred, "true, false". For security radar, normal arming must be enabled
first before usage; if enabling normal arming for the zone failed and the
device returned error message, then whether to enable forced arming can be
configured*/
    "systemfaultArming": ,
/*optional, boolean, whether to enable system fault notification, "true,
false"*/
    "realtimeStatus": ,
/*optional, boolean, whether to upload real-time system status, "true, false"*/
    "autoUpgrade": ,
/*optional, boolean, whether to enable auto upgrade, "true, false"*/
    "sysVolume": ,
/*optional, int, system volume*/
    "disableHostKey": ,
/*optional, boolean, whether to disable the functional keys of security control
panel*/
    "ipcDetectEnabled": ,
/*optional, boolean, whether supports enabling and configuring offline
detection for network camera, "true, false"*/
    "batteryDetectionEnabled": ,
/*optional, boolean, whether to enable security control panel battery
detection, "true", "false"*/
    "wordVoiceEnabled": ,
/*optional, boolean, whether to enable audio prompt: "true"-yes, "false"-no. If
audio prompt is disabled, the alarm prompt sound will be played normally*/
    "disArmAndClearAlarmVoicePrompt": ,
/*optional, boolean, whether the security control panel plays the audio prompt
of the fault once again when disarming or clearing alarms. If this function is
disabled, the audio prompt will not be played. This function cannot be
configured when the audio prompt is disabled*/
    "mandatoryArmEnabled": ,
/*optional, boolean, whether to enable forced arming regardless of the fault:
true=yes (if a fault occurs during automatic arming, the system will be armed
regardless of the fault), false=no*/
    "tamperLinkageAlarmEnabled": ,
/*optional, boolean, whether to enable tampering alarm linkage: true=yes (when
any device triggers a tampering alarm, it will link the siren and the security
control panel), false=no (when any device triggers the tampering alarm, only
the log is uploaded and the siren will not be linked)*/
    "oneKeyLockEnabled": ,
/*optional, boolean, whether to enable one-push locking panel (lock the
security control panel by pushing the Panel Lockup Button): true=yes (the
security control panel does not process any peripheral events, does not record
or upload any event reports, does not record any logs, does not support any
audio or voice function, and does not support any arming or disarming
operation), false=no (functions mentioned above are normal)*/
    "jammingSensitivity": "high",
/*optional, string, jamming sensitivity: "high" (default), "low", "close"*/
```

```
"UKLocalCertificationEnabled": true,
/*optional, boolean, whether to enable UK local certification*/
"ATPPFaultSendDelayTime": 1800,
/*optional, int, the delay time of reporting ATP malfunction to ARC, unit:
second*/
"faultIndicatorEnabled": true,
/*optional, boolean, whether to enable malfunction indicator light*/
"ezvizIndicatorEnabled": true,
/*optional, boolean, whether to enable EZVIZ indicator light*/
"motionDetectorRestore": "off",
/*optional, enum, types of restoring events to be uploaded: "off",
"alarmRestore"(arming restoring event), "disarmRestore" (disarming restoring
event)*/
"voicePromotType": {
/*optional, object, voice prompt types, this node is valid when the value of
wordVoiceEnabled is true*/
"armingPromot": true,
/*optional, boolean, whether to enable voice prompt in the process of arming*/
"armedPromot": true,
/*optional, boolean, whether to enable voice prompt when it is armed*/
"disarmedPromot": true,
/*optional, boolean, whether to enable voice prompt when it is disarmed*/
"alarmPromot": true
/*optional, boolean, whether to enable voice prompt for alarm*/
},
"armIndicatorLightAlwaysOnEnabled": true,
/*optional, boolean, whether it supports enabling keeping arming indicator
light always on*/
"energySavingMode": true,
/*optional, boolean, whether to enable energy saving mode*/
"tamperDetectionEnabled": true,
/*optional, boolean, whether to enable tampering detection, which will also
take effect after the installer logs in the device*/
"chimeEnabled": false,
/*optional, boolean, whether to enable doorbell, the keypad can make sound only
when the value of this node is true and doorbell function is enabled for the
zone*/
"tamperRestoreTime": 1
/*optional, int, tampering restore time, range:[1,86400]*/
}
}
```

A.3.109 JSON_ManageCap

JSON message about the configuration capability of security control system

```
{
  "ManageCap":{
    "setterEnabled":true,
/*optional, boolean, whether to enable installer configuration, "true, false"*/
    "wirelessSuperVision":true,
```

```
/*optional, boolean, whether to enable wireless peripherals management, "true,
false"*/
    "zonesfaultArming":true,
/*optional, boolean, whether to enable forbidding arming zone when fault
occurred, "true, false". For security radar, normal arming must be enabled
first before usage; if enabling normal arming for the zone failed and the
device returned error message, then whether to enable forced arming can be
configured*/
    "systemfaultArming":true,
/*optional, boolean, whether to enable zone fault notification, "true, false"*/
    "realtimeStatus":true,
/*optional, boolean, whether to enable system status reporting, "true, false"*/
    "autoUpgrade":true,
/*optional, boolean, whether to enable auto upgrade configuration, "true,
false"*/
    "sysVolume":{
/*optional, integer, system volume*/
        "@min":0,
        "@max":0
    },
    "disableHostKey":true,
/*optional, boolean, whether supports disabling the configuration pf functional
key on the security control panel, "true, false"*/
    "ipcDetectEnabled":true,
/*optional, boolean, whether supports enabling and configuring offline
detection for network camera, "true, false"*/
    "batteryDetectionEnabled":true,
/*optional, boolean, whether to enable security control panel battery
detection, "true", "false"*/
    "wordVoiceEnabled":true,
/*optional, boolean, whether to enable audio prompt: "true"-yes, "false"-no. If
audio prompt is disabled, the alarm prompt sound will be played normally*/
    "disArmAndClearAlarmVoicePrompt":true,
/*optional, boolean, whether the security control panel plays the audio prompt
of the fault once again when disarming or clearing alarms. If this function is
disabled, the audio prompt will not be played. This function cannot be
configured when the audio prompt is disabled*/
    "mandatoryArmEnabled": {
/*optional, boolean, whether to enable forced arming regardless of the fault:
true=yes (if a fault occurs during automatic arming, the system will be armed
regardless of the fault), false=no*/
        "@opt":[true,false]
    },
    "tamperLinkageAlarmEnabled": {
/*optional, boolean, whether to enable tampering alarm linkage: true=yes (when
any device triggers a tampering alarm, it will link the siren and the security
control panel), false=no (when any device triggers the tampering alarm, only
the log is uploaded and the siren will not be linked)*/
        "@opt":[true,false]
    },
    "oneKeyLockEnabled": {
/*optional, boolean, whether to enable one-push locking panel (lock the
```

```
security control panel by pushing the Panel Lockup Button): true=yes (the
security control panel does not process any peripheral events, does not record
or upload any event reports, does not record any logs, does not support any
audio or voice function, and does not support any arming or disarming
operation), false=no (functions mentioned above are normal)*/
    "@opt": [true, false]
  },
  "jammingSensitivity": {
/*optional, object, jamming sensitivity*/
/*scenario:
    1. Test the two RF communication channels (RX, R3) separately with the
default jamming sensitivity "high". If the jamming strength exceeds the
corresponding sensitivity for 30 seconds or more, a report on the communication
channel jamming will be generated according to the log record, and meanwhile
the system trouble indicator will be turned on; no more duplicate log and
report will be generated before the jamming is cleared;
    2. When a RF communication channel jamming occurs and release of the
channel is detected, the jamming clearing of the RF communication channel will
be recorded in the log to generate a report;
    3. Users can select the jamming sensitivity according to the real RF
jamming situation.
*/
    "@opt": ["high", "low", "close"]
  },
  "UKLocalCertificationEnabled": {
/*optional, object, whether to enable UK local certification: true-enable,
false-disable; if it is enabled, related configurations will be displayed
including alarm acknowledgment, delay reporting ATS communication malfunction
information to ARC, lid status and alarm acknowledgment for installer, and
selecting related events for wireless siren*/
    "@opt": [true, false]
  },
  "ATPFaultSendDelayTime": {
/*optional, object, the delay time of reporting ATP malfunction to ARC, unit:
second*/
    "@min": 1800,
/*optional, int, the minimum value*/
    "@max": 15000
/*optional, int, the maximum value*/
  },
  "faultIndicatorEnabled": {
/*optional, object, whether to enable malfunction indicator light*/
    "@opt": [true, false]
  },
  "ezvizIndicatorEnabled": {
/*optional, object, whether to enable EZVIZ indicator light*/
    "@opt": [true, false]
  },
  "motionDetectorRestore": {
/*optional, object, types of restoring events to be uploaded: "off",
"alarmRestore"(arming restoring event), "disarmRestore" (disarming restoring
event)*/
```



```
    "@opt": ["off", "alarmRestore", "disarmRestore"]
  },
  "voicePromotType": {
    /*optional, object, voice prompt types, this node is valid when the value of
    wordVoiceEnabled is true*/
    "armingPromot": {
      /*optional, object, whether to enable voice prompt in the process of arming*/
      "@opt": [true, false]
    },
    "armedPromot": {
      /*optional, object, whether to enable voice prompt when it is armed*/
      "@opt": [true, false]
    },
    "disarmedPromot": {
      /*optional, object, whether to enable voice prompt when it is disarmed*/
      "@opt": [true, false]
    },
    "alarmPromot": {
      /*optional, object, whether to enable voice prompt for alarm*/
      "@opt": [true, false]
    }
  },
  "armIndicatorLightAlwaysOnEnabled": true,
  /*optional, boolean, whether it supports enabling keeping arming indicator
  light always on: true=yes, false(or this node is not returned)-no*/
  "energySavingMode": {
    /*optional, boolean, whether to enable energy saving mode*/
    "@opt": [true, false]
  },
  "tamperDetectionEnabled": {
    /*optional, object, whether to enable tampering detection, which will also take
    effect after the installer logs in the device*/
    "@opt": [true, false]
  },
  "chimeEnabled": {
    /*optional, object, whether to enable doorbell, the keypad can make sound only
    when the value of this node is true and doorbell function is enabled for the
    zone*/
    "@opt": [true, false]
  },
  "tamperRestoreTime": {
    /*optional, object, tampering restore time*/
    "@min": 1,
    /*required, int, the minimum value, range:[1,86400]*/
    "@max": 86400
    /*required, int, the maximum value, range:[1,86400]*/
  }
}
```

A.3.110 JSON_ModuleLockCap

ModuleLockCap capability message in JSON format

```
{
  "ModuleLockCap":{
    "id":{
      /*required, No.*/
      "@min": ,
      "@max":
    },
    "type":{
      /*required, module type: "localKeypad"-local keypad, "keypad"-keypad,
      "cardReader"-card reader, "localCardReader"-local card reader*/
      "@opt":["localKeypad","keypad","cardReader","localCardReader"]
    },
    "status":{
      /*optional, read-only, status: "lock"-locked, "unlock"-unlocked*/
      "@opt":["lock","unlock"]
    },
    "maxTryTimes":{
      /*optional, maximum number of failed attempts*/
      "@min": ,
      "@max":
    },
    "lockedTime":{
      /*optional, locking duration, unit: second*/
      "@min": ,
      "@max":
    },
    "address":{
      /*optional, read-only, module address, this node is only returned by wired
      modules*/
      "@opt":[1,2,3]
    },
    "sirenAttrib":{
      /*optional, read-only, siren attribute: "wired", "wireless"*/
      "@opt":["wired","wireless"]
    },
    "isSupportAlwaysLocked":true
    /*whether to support configuration of remaining locked (whether to support
    setting lockedTime to 0): "true"-yes, this node is not returned-no*/
  }
}
```

A.3.111 JSON_MuteVoicePlanCFG

MuteVoicePlanCFG message in JSON format

```
{
  "MuteVoicePlanCFG":{
    "enable": ,
    /*required, boolean, whether to enable muting: "true"-enable, "false"-disable*/
    "WeekPlanCfg":[{
      /*optional, week schedule parameters*/
      "week": "",
      /*optional, string, day of the week: "Monday", "Tuesday", "Wednesday",
      "Thursday", "Friday", "Saturday", "Sunday"*/
      "id": ,
      /*optional, integer, time period No., it is between 1 and 8*/
      "enable": ,
      /*optional, boolean, whether to enable: "true"-enable, "false"-disable*/
      "TimeSegment":{
        "beginTime": "",
        /*optional, start time (device's local time)*/
        "endTime": ""
        /*optional, end time (device's local time)*/
      }
    }]
  }
}
```

A.3.112 JSON_NotRelateZones

NotRelateZones message in JSON format

```
{
  "NotRelateZones":{
    "id":
    /*required, array, No. of unlinked zones (No. of zones that are not linked to
    detectors)*/
  }
}
```

A.3.113 JSON_NotRelateZonesCap

NotRelateZonesCap capability message in JSON format

```
{
  "NotRelateZonesCap":{
    "id":{
      /*required, unlinked zone range (No. of zones that are not linked to
      detectors)*/
      "@min": ,
      "@max":
    }
  }
}
```

```
}  
}
```

A.3.114 JSON_Operate

JSON message about operation parameters

```
{  
  "Operate": {  
/*optional, object, operation parameters*/  
    "moduleOperateCode": "12345"  
/*optional, string, module operation code, which should be encrypted*/  
  }  
}
```

A.3.115 JSON_Output

JSON message about the parameters of a logical relay

```
{  
  "Output": {  
    "id": ,  
/*required, integer type, relay No., which starts from 0*/  
    "name": "",  
/*optional, string type, relay name*/  
    "related": ,  
/*read-only, boolean type, whether the relay is linked to the output module,  
for PUT method, this node is optional; for GET method, this node is required*/  
    "outputModuleNo": ,  
/*read-only, integer type, linked output module No., this node is required only  
when the related is "true"*/  
    "channelNo": ,  
/*read-only, integer type, channel No. of output module, this node is required  
only when the related is "true"*/  
    "linkage": "",  
/*optional, string type, linked event types: "alarm", "arming", "disarming",  
"manualCtrl"-manual control, "system"-system event, "zone"*/  
    "minorType": ,  
/*optional, array, minor event type: "zoneAlarmTamper"-zone alarm and tampered  
event, "exDevTamper"-peripheral device tampered, "hostTamper"-control panel  
tampered, "emergency"-panic alarm, "medical"-medical alarm, "fire"-fire alarm,  
"gas"-gas event. When linkage is "alarm", one or all minor event types can be  
selected. The data type of elements in the array is string*/  
    "subSystem": ,  
/*optional, array, linked partitions. When linkage is "alarm", "arming", or  
"disarming", one or more partitions can be configured to link. For example,  
[1,2,3] indicates that partition 1, partition 2, and partition 3 are linked*/  
    "LinkageList": [{  
/*optional, linked event list. If the device supports multiple linkage event
```

```
types linking to multiple minor event types and multiple partitions, this node
can be used. For compatibility, one linkage event types linking to multiple
minor event types and multiple partitions should also be supported*/
    "linkage": "",
    /*optional, string, event types that can be linked to the relay: "alarm",
    "arming", "disarming", "manualCtrl"-manual control, "zone"*/
    "alarmMinorType": ,
    /*optional, array, minor event type: "zoneAlarmTamper"-zone alarm and tampered
    event, "exDevTamper"-peripheral device tampered, "hostTamper"-control panel
    tampered, "emergency"-panic alarm, "medical"-medical alarm, "fire"-fire alarm,
    "gas"-gas event. When linkage is "alarm", one or all minor event types can be
    selected. The data type of elements in the array is string*/
    "zoneEvent": ,
    /*optional, string, zone event type, this node is valid when linkage contains
    "zone". [1,3] indicates that zone 1 and zone 3 are enabled. If this node is not
    configured when linkage contains "zone", it indicates enabling all zones*/
    "subSystem":
    /*optional, array, linked partitions. When linkage is "alarm", "arming", or
    "disarming", one or more partitions can be configured to link. For example,
    [1,2,3] indicates that partition 1, partition 2, and partition 3 are linked*/
    },
    "durationConstOutputEnable": ,
    /*optional, boolean, whether it supports configuring relay output duration:
    "true"-not support (the duration cannot be configured and the relay will output
    continuously), "false"-support (the duration can be configured and it can be
    between 5s and 600s)*/
    "alarmEvent": "",
    /*optional, string type, alarm event types, this node is valid only when the
    node linkage is "alarm", e.g., "1,3" indicates that the zone 1 and zone 3 is
    enabled; if this node is not configured, it indicates that all zones are
    enabled*/
    "zoneEvent": ,
    /*optional, string type, zone event type, this node is valid when linkage
    contains "zone". For example, [1,3] indicates that zone 1 and zone 3 are
    enabled. If this node is not configured when linkage contains "zone", it
    indicates enabling all zones*/
    "systemEvent": "",
    /*optional, string type, system event types: "ACOff"-AC power off,
    "networkAbnormal"-network fault, "dismantleAlarm"-tampering alarm, "hidAlarm"-
    device blocking alarm, "preventMoveEvent"-motion event, "radarAbnormal"-radar
    fault. This node is valid only when the node linkage is "system"; if this node
    is not configured, it indicates that all system event types are selected*/
    "duration": ,
    /*optional, integer type, output duration*/
    "alarmLine": ,
    /*optional, integer type, trigger line No., it starts from 1*/
    "followModeEnabled": ,
    /*optional, boolean type, whether to enable following mode. If this node is set
    to "true": 1) duration is invalid for zone alarms, and after the zone or
    trigger line is triggered, the relay can remain open until the zone or trigger
    line is restored; 2) only zone or trigger line events can be linked, system
    events and arming and disarming events cannot be linked*/
```

```

    "moduleType": "",
    /*optional, read-only, string type, module type: "localWired"-local wired
    module, "extendWired"-extended wired module, "localWireless"-local wireless
    module, "extendWireless"-extended wireless module*/
    "scenarioType": ["alarm", "schedule", "arm", "disarm", "clearAlarm",
    "fault", "manual"],
    /*optional, object, scenario type, you can select one or multiple types*/
    "alarmCfg": {
    /*optional, object, alarm configuration*/
        "alarmType": ["zoneAlarmAndTamper", "exDevTamper", "panicAlarm",
        "medicalAlarm", "hostTamper", "fireAlarm"],
        /*optional, object, alarm type, you can set one or multiple types*/
        "relayMode": "pulse",
        /*optional, enum, relay mode: "pulse", "latch", subType:string*/
        "pulseDuration": 0,
        /*optional, int, pulse interval, this node is valid when relayMode is "pulse"*/
        "contactStatus": "normalOpen",
        /*optional, enum, relay status after events occurred: "normalOpen",
        "normalClose", subType:string*/
        "supportAssociatedZone": [1, 2, 3],
        /*optional, array, zones that can be linked, this node is valid when alarmType
        is "zoneAlarmAndTamper", subType:int*/
        "associateZoneCfg": [1, 2, 3],
        /*optional, array, zones that have been linked with this relay, this node is
        valid when alarmType is "zoneAlarmAndTamper", subType:int*/
        "alarmLogic": "and",
        /*optional, enum, alarm logic: "and", "or", this node is valid when alarmType
        is "zoneAlarmAndTamper", subType:string*/
        "supportLinkageChannelID": [1, 2, 3],
        /*optional, array, No.s of the channels that can be linked with network
        cameras, this node is valid when alarmType is "zoneAlarmAndTamper",
        subType:int*/
        "linkageChannelID": [1, 2]
        /*optional, array, No.s of the channels to be linked with network cameras, this
        node is valid when alarmType is "zoneAlarmAndTamper", subType:int, range:[1,4]*/
    },
    "address": ,
    /*optional, read-only, integer type, module address, this node is only returned
    by wired modules*/
    "linkageAddress": ,
    /*optional, read-only, integer type, linked module address, this node is only
    returned by wireless modules*/
    "moduleChannel": ,
    /*optional, integer type, module channel No.*/
    "alarmLine": [1, 2, 3],
    /*optional, array, No. of the trigger lines, which starts from 1*/
    "followModeEnabled": true,
    /*optional, boolean, whether to enable following mode. If this node is set to
    "true": 1) duration is invalid for zone alarms, and after the zone or trigger
    line is triggered, the relay can remain open until the zone or trigger line is
    restored; 2) only zone or trigger line events can be linked, system events and
    arming and disarming events cannot be linked*/

```

```
    "modifiedOutputNo": 1,
    /*optional, int, relay No. that has been modified*/
    "accessModuleType": "transmitter",
    /*optional, string, access module type: "transmitter", "multiTransmitter",
    "localTransmitter", "localRelay", "keypad"*/
    "relatedAccessModuleID": 1,
    /*optional, int, No. of the linked access module*/
    "relayAttrib": "wired"
    /*optional, string, relay attribute: "wired", "wireless" (default)*/
  }
}
```

Remarks

Zone parameters configured by **alarmEvent** and **zoneEvent** should be the same and they both start from 0.

A.3.116 JSON_OutputCap

JSON message about the configuration capability of logical relays

```
{
  "OutputCap":{
    "id":{
      /*required, relay No. range*/
      "@min": ,
      "@max":
    },
    "name":{
      /*optional, range of relay name length*/
      "@min": ,
      "@max":
    },
    "related":{
      /*required, whether to link to the physical relay (the channel of the output
      module)*/
      "@opt":"true,false"
    },
    "outputModuleNo":{
      /*required, range of output module No. length*/
      "@min": ,
      "@max":
    },
    "channelNo":{
      /*required, range of channel No. of output module*/
      "@min": ,
      "@max":
    },
    "linkage":""{
      /*required, event types linked to the relay: "alarm", "arming", "disarming",
      "manualCtrl"-manual control, "system"-system event, "zone". When this field is
```

```
"arming", "disarming", or "alarm", one or more partitions can be configured to
link*/
    "@opt": "alarm,arming,disarming>manualCtrl,system,zone"
},
"minorType":{
/*optional, array, minor event type: "zoneAlarmTamper"-zone alarm and tampered
event, "exDevTamper"-peripheral device tampered, "hostTamper"-control panel
tampered, "emergency"-panic alarm, "medical"-medical alarm, "fire"-fire alarm,
"gas"-gas event. When linkage is "alarm", one or all minor event types can be
selected. The data type of elements in the array is string*/
    "@opt":["zoneAlarmTamper", "exDevTamper", "hostTamper", "emergency",
"medical", "fire", "gas"]
},
"subSystem":{
/*optional, number of linked partitions*/
    "@min": ,
    "@max":
},
"subSystemNo":{
/*optional, range of partition No.*/
    "@min": ,
    "@max":
},
"LinkageList":{
/*optional, linked event list. If the device supports multiple linkage event
types linking to multiple minor event types and multiple partitions, this node
can be used. For compatibility, one linkage event types linking to multiple
minor event types and multiple partitions should also be supported*/
    "@size":2,
/*int, maximum number of linkage event types*/
    "linkage":""{
/*optional, string, event types that can be linked to the relay: "alarm",
"arming", "disarming", "manualCtrl"-manual control, "zone"*/
        "@opt": "alarm,arming,disarming>manualCtrl,zone"
    },
    "alarmMinorType":{
/*optional, array, minor event type: "zoneAlarmTamper"-zone alarm and tampered
event, "exDevTamper"-peripheral device tampered, "hostTamper"-control panel
tampered, "emergency"-panic alarm, "medical"-medical alarm, "fire"-fire alarm,
"gas"-gas event. When linkage is "alarm", one or all minor event types can be
selected. The data type of elements in the array is string*/
        "@opt":
["zoneAlarmTamper", "exDevTamper", "hostTamper", "emergency", "medical", "fire", "gas"
]
    },
    "zoneEvent ":{
        "@min":1,
/*optional, int, minimum zone No.*/
        "@max":2,
/*optional, int, maximum zone No.*/
        "@size":2
/*optional, int, maximum number of zones that can be linked*/
```



```

    }
    "subSystem":{
        "@min":1,
/*optional, int, minimum partition No.*/
        "@max":2,
/*optional, int, maximum partition No.*/
        "@size":2
/*optional, int, maximum number of partitions that can be linked*/
    }
},
    "durationConstOutputEnable":{
/*optional, boolean, whether it supports configuring relay output duration:
"true"-not support (the duration cannot be configured and the relay will output
continuously), "false"-support (the duration can be configured and it can be
between 5s and 600s)*/
        "@opt":"true,false"
    },
    "isLinkageSupportMultiSelect": ,
/*optional, boolean type, whether the node linkage supports multiple
selections, "false" or this node is not returned-not support, "true"-support*/
    "alarmEvent":""{
/*optional, string type, alarm event types, this node is valid only when the
node linkage is "alarm", e.g., "1,3" indicates that the zone 1 and zone 3 is
enabled; if this node is not configured, it indicates that all zones are
enabled*/
        "@opt":"1,2,3,4"
    },
    "zoneEvent":{
/*optional, string, zone event type, this node is valid when linkage contains
"zone". For example, [1,3] indicates that zone 1 and zone 3 are enabled. If
this node is not configured when linkage contains "zone", it indicates enabling
all zones*/
        "@opt":[0,1,2,3]
    },
    "systemEvent":""{
/*optional, string type, system event types: "ACOff"-AC power off,
"networkAbnormal"-network fault, "dismantleAlarm"-tampering alarm, "hidAlarm"-
device blocking alarm, "preventMoveEvent"-motion event, "radarAbnormal"-radar
fault. This node is valid only when the node linkage is "system"; if this node
is not configured, it indicates that all system event types are selected*/

"@opt":"ACOff,networkAbnormal,dismantleAlarm,hideAlarm,preventMoveEvent,radarAbn
ormal"
    },
    "duration":{
/*optional, output duration of the relay*/
        "@min": ,
        "@max":
    },
    "moduleType":{
/*optional, read-only, module type: "localWired"-local wired module,
"extendWired"-extended wired module, "localWireless"-local wireless module,

```

```
"extendWireless"-extended wireless module*/
  "@opt":["localWired","extendWired","localWireless","extendWireless"]
},
"scenarioType": {
/*optional, object, scenario type, you can select one or multiple types*/
  "@opt": ["alarm", "schedule", "arm", "disarm", "clearAlarm", "fault",
"manual"]
},
"alarmCfg": {
/*optional, object, alarm configuration, this node is valid when scenarioType
is "alarm"*/
  "alarmType": {
/*optional, object, alarm type, you can set one or multiple types*/
    "@opt": ["zoneAlarmAndTamper", "exDevTamper", "panicAlarm",
"medicalAlarm", "hostTamper", "fireAlarm"]
  },
  "relayMode": {
/*optional, object, relay mode*/
    "@opt": ["pulse", "latch"]
  },
  "pulseDuration": {
/*optional, object, pulse interval, this node is valid when relayMode is
"pulse"*/
    "@min": 0,
/*optional, int, the minimum value*/
    "@max": 0
/*optional, int, the maximum value*/
  },
  "contactStatus": {
/*optional, object, relay status after events occurred*/
    "@opt": ["normalOpen", "normalClose"]
  },
  "supportAssociatedZone": {
/*optional, object, zones that can be linked, this node is valid when alarmType
is "zoneAlarmAndTamper"*/
    "@min": 1,
/*optional, int, the minimum value*/
    "@max": 2,
/*optional, int, the maximum value*/
    "@size": 1
/*optional, int, the maximum number of zones that can be linked*/
  },
  "associateZoneCfg": {
/*optional, object, zones that have been linked with this relay, this node is
valid when alarmType is "zoneAlarmAndTamper"*/
    "@min": 1,
/*optional, int, the minimum value*/
    "@max": 2,
/*optional, int, the maximum value*/
    "@size": 1
/*optional, int, the maximum number of zones that can be linked with this
relay*/
  }
}
```

```

    },
    "alarmLogic": {
/*optional, object, alarm logic: "and", "or", this node is valid when alarmType
is "zoneAlarmAndTamper"*/
        "@opt": ["and", "or"]
    },
    "supportLinkageChannelID": {
/*optional, object, No.s of the channels that can be linked with network
cameras, this node is valid when alarmType is "zoneAlarmAndTamper"*/
        "@min": 1,
/*optional, int, the minimum value, range:[1,4]*/
        "@max": 4,
/*optional, int, the maximum value, range:[1,4]*/
        "@size": 4
/*optional, int, the maximum number of channels that can be linked with network
cameras, range:[1,4]*/
    },
    "linkageChannelID": {
/*optional, object, No.s of the channels to be linked with network cameras,
this node is valid when alarmType is "zoneAlarmAndTamper"*/
        "@min": 1,
/*optional, int, the minimum value, range:[1,4]*/
        "@max": 4,
/*optional, int, the maximum value, range:[1,4]*/
        "@size": 4
/*optional, int, the maximum number of channels to be linked, range:[1,4]*/
    }
},
    "address":{
/*optional, read-only, module address, this node is only returned by wired
modules*/
        "@opt":[1,2,3]
    },
    "linkageAddress":{
/*optional, read-only, linked module address, this node is only returned by
wireless modules*/
        "@opt":[1,2,3]
    },
    "moduleChannel":{
/*optional, module channel No.*/
        "@min": ,
        "@max":
    },
    "method":{
/*required, methods supported by the function: "put"-edit, "getAll"-get all,
"getCond"-get by conditions*/
        "@opt":"put,getAll,getCond"
    },
    "alarmLine":{
/*optional, integer type, number of linked trigger lines' No.*/
        "@min": ,
        "@max":
    }
}

```

```
    },
    "alarmLineNo":{
/*optional, integer type, range of trigger line No., the minimum No. is 1*/
        "@min":1,
        "@max":
    },
    "followModeEnabled":{
/*optional, boolean type, whether to enable following mode. If this node is set
to "true": 1) duration is invalid for zone alarms, and after the zone or
trigger line is triggered, the relay can remain open until the zone or trigger
line is restored; 2) only zone or trigger line events can be linked, system
events and arming and disarming events cannot be linked*/
        "@opt":[true,false]
    },
    "maxOutputsResults": ,
/*optional, integer, maximum number of records that can be obtained each time
by calling this URI. This node is valid only when method is "getCond"*/
    "isSupportSignalTest": true,
/*optional, boolean, whether it supports signal strength detection, if the
value is false or the node is not returned, it indicates that this function is
not supported*/
    "isSupportZoneTest": true,
/*optional, boolean, whether it supports zone detection, if the value is false
or the node is not returned, it indicates that this function is not supported*/
    "notRelatedOutputNo": {
/*optional, object, No. of unlinked relay*/
        "@size": 1,
/*optional, int, the maximum number of unlinked relays*/
        "@min": 1,
/*optional, int, the minimum value*/
        "@max": 2
/*optional, int, the maximum value*/
    },
    "modifiedOutputNo": {
/*optional, object, relay No. that has been modified*/
        "@min": 1,
/*optional, int, the minimum value*/
        "@max": 2
/*optional, int, the maximum value*/
    },
    "accessModuleType": {
/*optional, object, access module type*/
        "@opt": ["transmitter", "multiTransmitter", "localTransmitter",
"localRelay", "keypad"]
    },
    "relatedAccessModuleID": {
/*optional, object, No. of the linked access module*/
        "@min": 0,
/*optional, int, the minimum value*/
        "@max": 255
/*optional, int, the maximum value*/
    },
    },
```

```
"relayAttrib": {  
/*optional, object, relay attribute: "wired", "wireless" (default)*/  
  "@opt": ["wired", "wireless"]  
}  
}  
}
```

Remarks

Zone parameters configured by **alarmEvent** and **zoneEvent** should be the same and they both start from 0.

A.3.117 JSON_OutputCond

JSON message about conditions of getting relay status

```
{  
  "OutputCond":{  
    "searchID":"","  
/*required, string, search ID, which is used to confirm the upper-level  
platform or system. If the platform or the system is the same one during two  
searching, the search history will be saved in the memory to speed up next  
searching*/  
    "searchResultPosition": ,  
/*required, integer32, the start position of the search result in the result  
list. When there are multiple records and you cannot get all search results at  
a time, you can search for the records after the specified position next time*/  
    "maxResults": ,  
/*required, integer32, maximum number of search results this time by calling  
this URI. If maxResults exceeds the range returned by the device capability,  
the device will return the maximum number of search results according to the  
device capability and will not return error message*/  
    "outputModuleNo": ,  
/*optional, int, linked output module No.*/  
    "moduleType":""  
/*optional, string, module type: "localWired"-local wired module, "extendWired"-  
extended wired module, "localWireless"-local wireless module, "extendWireless"-  
extended wireless module*/  
  }  
}
```

A.3.118 JSON_OutputModList

JSON message about status of output module

```
{  
  "OutputModList":[{  
/*optional, output module list*/  
    "OutputMod":{
```

```
    "id": ,
/*required, integer type, output module No.*/
    "seq": "",
/*required, string type, peripheral serial No.*/
    "status": "",
/*optional, string type, wireless output module status: "notRelated"-not
linked, "online", "offline", "heartbeatAbnormal"-heartbeat exception*/
    "tamperEvident": ,
/*optional, boolean type, zone tampering status: "true"-tampered, "false"-not
tampered*/
    "charge": "",
/*optional, string type, state of charge: "normal", "lowPower"-low battery*/
    "signal":
/*optional, integer type, signal strength, it is between 0 and 255*/
    "model": "DS-PM1-O8-WE",
/*optional, string, model: "DS-PM1-O8-WE" (wireless output module with 8
channels), "DS-PM1-O2-WE" (wireless output module with 2 channels)*/
    "temperature": 1,
/*optional, int, temperature*/
    "isViaRepeater": true,
/*optional, boolean, whether the signal is forwarded via repeater*/
    "repeaterName": "test",
/*optional, string, repeater name, this node is valid when the value of
isViaRepeater is true*/
    "voltValue": 1,
/*optional, int, voltage value, unit: V*/
    "currentValue": 1,
/*optional, int, current value, unit: mA*/
    "powerLoad": 1,
/*optional, int, power load, unit: W*/
    "energySumVaule": 1,
/*optional, int, power consumption, unit: Wh*/
    "relayList": [{
/*optional, array, list of relays*/
    "id": 1,
/*required, int, input ID*/
    "status": "on",
/*optional, string, relay status, "on", "off"*/
    "name": "test",
/*optional, string, relay name*/
    "subSystem": [1, 2, 3],
/*optional, array, related partition*/
    "scenarioType": ["alarm"]
/*optional, array, scenario type*/
    }],
    "voltValueV20": 1.000
/*optional, float, voltage value (version 2.0) which is accurate to 3 decimal
places, unit: V, this node has higher priority over voltValue*/
    }
  }
}
```

A.3.119 JSON_OutputModule

OutputModule message in JSON format

```
{
  "OutputModule":{
    "id": ,
    /*required, integer type, output module No., which starts from 0*/
    "name": "",
    /*optional, string type, output module name*/
    "related": ,
    /*boolean type, whether the output module is linked, for PUT method, this node
    is optional; for GET method, this node is required*/
    "seq": "",
    /*string type, output module serial No., this node is required only when
    related is "true"*/
    "ChanList":[{
    /*read-only, channel list of output module, this node is optional for PUT
    method, but it is required for GET method*/
      "Chan":{
    /*channel information of output module, this node is required only when the
    node ChanList exists*/
        "channelNo":
    /*integer type, channel No. of output module, this node is required only when
    the node Chan exists*/
      }
    }],
    "address": ,
    /*optional, read-only, integer type, module address, this node is only returned
    by wired modules*/
    "linkageAddress": ,
    /*optional, read-only, integer, linked module address, this node is only
    returned by wireless modules. If related is "true", this node cannot be
    configured*/
    "attrib": "",
    /*optional, read-only, module attribute: "wired", "wireless". If this node is
    not returned, the default module attribute is "wireless"*/
    "checkTime":
    /*optional, integer type, offline duration, unit: hour*/
  }
}
```

A.3.120 JSON_OutputModuleCap

OutputModuleCap capability message in JSON format

```
{
  "OutputModuleCap":{
    "id":{
```

```
/*required, range of output module No.*/
    "@min": ,
    "@max":
  },
  "name":{
/*optional, range of output module name length*/
    "@min": ,
    "@max":
  },
  "related":{
/*required, whether the output module is linked*/
    "@opt":"true,false"
  },
  "seq":{
/*required, range of output module serial No. length*/
    "@min": ,
    "@max":
  },
  "address":{
/*optional, read-only, module address, this node is only returned by wired
modules*/
    "@opt":[1,2,3]
  },
  "linkageAddress":{
/*optional, read-only, linked module address, this node is only returned by
wireless modules*/
    "@opt":[1,2,3]
  },
  "attrib":{
/*optional, read-only, module attribute: "wired", "wireless". If this node is
not returned, the default module attribute is "wireless"*/
    "@opt":["wired","wireless"]
  },
  "checkTime":{
/*optional, offline duration, unit: hour*/
    "@min": ,
    "@max":
  },
  "method":{
/*required, methods supported by the function: "add", "put"-edit, "getAll"-get
all*/
    "@opt":"add,put,getAll"
  }
}
```

A.3.121 JSON_OutputsCtrl

JSON message about relay control parameters


```
{
  "OutputsCtrl":{
    "switch":""
  /*required, string, "open"-enable relay, "close"-disable relay*/
    "List":[{
  /*it is required when control multiple relays, and it is not required when
control only one relay*/
    "id":
  /*int, relay No., which starts from 0*/
    }]
  }
}
```

A.3.122 JSON_OutputSearch_Config

JSON message about the parameters of the specified logical relays

```
{
  "OutputSearch":{
    "searchID":"","
  /*required, string type, search ID, which is used to confirm the upper-level
platform or system. If the platform or the system is the same one during two
searching, the search history will be saved in the memory to speed up next
searching*/
    "responseStatusStrg":"","
  /*required, string type, search status description: "OK"-search ended, "MORE"-
more data to be searched, "NO MATCH"-no data found*/
    "numOfMatches": ,
  /*required, integer32 type, number of matched results returned this time*/
    "totalMatches": ,
  /*required, integer32 type, total number of matched results*/
    "List":[{
      "Output":{
        "id": ,
  /*required, integer type, relay No., which starts from 0*/
        "name":"","
  /*optional, string type, relay name*/
        "related": ,
  /*read-only, boolean type, whether the relay is linked to the physical relay
(the channel of the output module). For PUT method, this node is optional; for
GET method, this node is required*/
        "outputModuleNo": ,
  /*read-only, integer type, linked output module No., this node is required only
when the related is "true"*/
        "channelNo": ,
  /*read-only, integer type, channel No. of output module, this node is required
only when the related is "true"*/
        "linkage":"","
  /*optional, string type, event linkage type: "alarm", "arming", "disarming",
"manualCtrl"-manual control, "system"-system event, "zone"*/
```

```
    "minorType": ,
/*optional, array, minor event type: "zoneAlarmTamper"-zone alarm and tampered
event, "exDevTamper"-peripheral device tampered, "hostTamper"-control panel
tampered, "emergency"-panic alarm, "medical"-medical alarm, "fire"-fire alarm,
"gas"-gas event. When linkage is "alarm", one or all minor event types can be
selected. The data type of elements in the array is string*/
    "subSystem": ,
/*optional, array, linked partitions. When linkage is "alarm", "arming", or
"disarming", one or more partitions can be configured to link. For example,
[1,2,3] indicates that partition 1, partition 2, and partition 3 are linked*/
    "LinkageList":[{
/*optional, linked event list. If the device supports multiple linkage event
types linking to multiple minor event types and multiple partitions, this node
can be used. For compatibility, one linkage event types linking to multiple
minor event types and multiple partitions should also be supported*/
        "linkage": "",
/*optional, string, event types that can be linked to the relay: "alarm",
"arming", "disarming", "manualCtrl"-manual control, "zone"*/
        "alarmMinorType": ,
/*optional, array, minor event type: "zoneAlarmTamper"-zone alarm and tampered
event, "exDevTamper"-peripheral device tampered, "hostTamper"-control panel
tampered, "emergency"-panic alarm, "medical"-medical alarm, "fire"-fire alarm,
"gas"-gas event. When linkage is "alarm", one or all minor event types can be
selected. The data type of elements in the array is string*/
        "zoneEvent": ,
/*optional, string, zone event type, this node is valid when linkage contains
"zone". [1,3] indicates that zone 1 and zone 3 are enabled. If this node is not
configured when linkage contains "zone", it indicates enabling all zones*/
        "subSystem":
/*optional, array, linked partitions. When linkage is "alarm", "arming", or
"disarming", one or more partitions can be configured to link. For example,
[1,2,3] indicates that partition 1, partition 2, and partition 3 are linked*/
        }],
    "durationConstOutputEnable": ,
/*optional, boolean, whether it supports configuring relay output duration:
"true"-not support (the duration cannot be configured and the relay will output
continuously), "false"-support (the duration can be configured and it can be
between 5s and 600s)*/
    "alarmEvent": "",
/*optional, string type, alarm event types, this node is valid only when
linkage contains "alarm". For example, "1,3" indicates that the zone 1 and zone
3 are enabled; if this node is not configured when linkage contains "alarm", it
indicates that all zones are enabled*/
    "zoneEvent": ,
/*optional, string type, zone event type, this node is valid when linkage
contains "zone". For example, [1,3] indicates that zone 1 and zone 3 are
enabled. If this node is not configured when linkage contains "zone", it
indicates enabling all zones*/
    "duration": ,
/*optional, output duration of the relay*/
    "moduleType": "",
/*optional, read-only, string type, module type: "localWired"-local wired
```

```
module, "extendWired"-extended wired module, "localWireless"-local wireless
module, "extendWireless"-extended wireless module*/
    "address": ,
/*optional, read-only, integer type, module address, this node is only returned
by wired modules*/
    "linkageAddress": ,
/*optional, read-only, integer type, linked module address, this node is only
returned by wireless modules*/
    "moduleChannel": ,
/*optional, integer type, module channel No.*/
    "alarmLine": [1, 2, 3],
/*optional, int, trigger line No., it starts from 1*/
    "followModeEnabled": true,
/*optional, boolean type, whether to enable following mode. If this node is set
to "true": 1) duration is invalid for zone alarms, and after the zone or
trigger line is triggered, the relay can remain open until the zone or trigger
line is restored; 2) only zone or trigger line events can be linked, system
events and arming and disarming events cannot be linked*/
    "notRelatedOutputNo": [1, 2, 3],
/*optional, int, No. of unlinked relay*/
    "modifiedOutputNo": 1,
/*optional, int, relay No. that has been modified*/
    "accessModuleType": "transmitter",
/*optional, string, access module type: "transmitter", "multiTransmitter",
"localTransmitter", "localRelay", "keypad"*/
    "relatedAccessModuleID": 1,
/*optional, int, No. of the linked access module*/
    "relayAttrib": "wired"
/*optional, string, relay attribute: "wired", "wireless" (default)*/
    }
  ]]
}
}
```

A.3.123 JSON_OutputSearch_Status

JSON message about results of getting relay status

```
{
  "OutputSearch":{
    "searchID":"","
/*required, string, search ID, which is used to confirm the upper-level
platform or system. If the platform or the system is the same one during two
searching, the search history will be saved in the memory to speed up next
searching*/
    "responseStatusStrg":"","
/*required, string, search status: "OK"-searching completed, "NO MATCH"-no
matched results, "MORE"-searching for more results*/
    "numOfMatches": ,
/*required, integer32, number of results returned this time*/
    "totalMatches": ,
```

```
/*required, integer32, total number of matched results*/
    "OutputList":[{
/*optional, relay list*/
        "Output":{
            "id": ,
/*required, int, relay No.*/
            "name": "",
/*optional, string, relay name*/
            "status": "",
/*optional, string, relay status: "notRelated"-not linked, "on", "off",
"offline", "heartbeatAbnormal"-heartbeat exception*/
            "tamperEvident": ,
/*optional, boolean, zone tampering status: true (tampered), false (not
tampered)*/
            "charge": "",
/*optional, string, state of charge: "normal", "lowPower"-low battery*/
            "linkage": "",
/*optional, string, event type linked to the relay: "alarm", "arming",
"disarming", "manualCtrl"-manually control*/
            "signal": ,
/*optional, int, signal strength, it is between 0 and 255*/
            "temperature": 1,
/*optional, int, temperature*/
            "devIndex": "test",
/*optional, string, device ID, the maximum length is 64 bytes*/
            "devName": "test",
/*optional, string, device name, the maximum length is 64 bytes*/
            "durationConstOutputEnable": true,
/*optional, boolean, whether to always keep the relay open*/
            "isAvailable": true,
/*optional, boolean, whether the relay is enabled, if this node is not
returned, it indicates that the relay is enabled by default*/
            "accessModuleType": "transmitter",
/*optional, enum, access module type: "transmitter", "localTransmitter",
"multiTransmitter", "localRelay", "keypad"*/
            "relatedAccessModuleID": 1,
/*optional, int, linked access module ID*/
            "address": 254,
/*optional, int, wired (extended) module address, this node works with
accessModuleType*/
            "subSystemList": [1, 2, 3],
/*optional, array, list of linked partitions*/
            "scenarioType": ["alarm"],
/*optional, array, scenario type*/
            "relayAttrib": "wired",
/*optional, string, relay attribute: "wired", "wireless" (default)*/
            "deviceNo": 1
/*optional, int, device ID, range:[1,1000]*/
        }
    }
}
```

A.3.124 JSON_OutPutsModule

JSON message about linkage configuration parameters of a single relay when the relay is closed/open

```
{
  "OutPutModule": {
    "name": "",
    /*optional, string, relay name*/
    "OutPutCloseLinkage": [{
    /*optional, event linkage information when the relay is closed*/
      "linkage": "",
      /*optional, string, event types that can be linked to the relay: "alarm",
      "arming", "disarming", "manualCtrl"-manual control, "zone", "sysEvent"*/
      "alarmMinorType": ["zoneAlarmTamper", "exDevTamper", "hostTamper",
      "emergency", "medical", "fire", "gas"],
      /*optional, array, minor event type: "zoneAlarmTamper"-zone alarm and tampered
      event, "exDevTamper"-peripheral device tampered, "hostTamper"-control panel
      tampered, "emergency"-panic alarm, "medical"-medical alarm, "fire"-fire alarm,
      "gas"-gas event. When linkage is "alarm", one or all minor event types can be
      selected. The data type of elements in the array is string*/
      "sysEventMinorType": ["ACOutage", "lowVoltageOfBattery",
      "telephoneOffLine", "networkAbnormal", "wirelessNetworkAbnormal",
      "harddiskException", "485Exception", "mBusException", "3G4GSignalAbnormal",
      "moduleOffline"],
      /*optional, array, minor type of system event: "ACOutage"-AC power outage,
      "lowVoltageOfBattery"-low voltage of battery, "telephoneOffLine"-telephone
      offline, "networkAbnormal"-wired network disconnected,
      "wirelessNetworkAbnormal"-wireless network disconnected, "harddiskException"-
      hard disk exception, "485Exception"-system keyboard exception, "mBusException"-
      MBUS exception, "3G4GSignalAbnormal"-3G/4G signal exception, "moduleOffline"-
      module offline. This node is valid when linkage contains "sysEvent"*/
      "zoneEvent": [1, 3],
      /*optional, array, zone event type, this node is valid when linkage contains
      "zone". [1,3] indicates that zone 1 and zone 3 are enabled. If this node is not
      configured when linkage contains "zone", it indicates enabling all zones*/
      "subSystem": [1, 2, 3]
      /*optional, array, linked partitions. When linkage is "alarm", "arming", or
      "disarming", one or more partitions can be configured to link. For example,
      [1,2,3] indicates that partition 1, partition 2, and partition 3 are linked*/
    }],
    "OutputOpenLinkage": [{
    /*optional, event linkage information when the relay is open*/
      "linkage": "",
      /*optional, string, event types that can be linked to the relay: "alarm",
      "arming", "disarming", "manualCtrl"-manual control, "zone", "sysEvent"*/
      "alarmMinorType": ["zoneAlarmTamper", "exDevTamper", "hostTamper",
      "emergency", "medical", "fire", "gas"],
      /*optional, array, minor event type: "zoneAlarmTamper"-zone alarm and tampered
      event, "exDevTamper"-peripheral device tampered, "hostTamper"-control panel
      tampered, "emergency"-panic alarm, "medical"-medical alarm, "fire"-fire alarm,
```

```
"gas"-gas event. When linkage is "alarm", one or all minor event types can be
selected. The data type of elements in the array is string*/
  "sysEventMinorType": ["ACOutage", "lowVoltageOfBattery",
"telephoneOffLine", "networkAbnormal", "wirelessNetworkAbnormal",
"harddiskException", "485Exception", "mBusException", "3G4GSignalAbnormal",
"moduleOffline"],
/*optional, array, minor type of system event: "ACOutage"-AC power outage,
"lowVoltageOfBattery"-low voltage of battery, "telephoneOffLine"-telephone
offline, "networkAbnormal"-wired network disconnected,
"wirelessNetworkAbnormal"-wireless network disconnected, "harddiskException"-
hard disk exception, "485Exception"-system keyboard exception, "mBusException"-
MBUS exception, "3G4GSignalAbnormal"-3G/4G signal exception, "moduleOffline"-
module offline. This node is valid when linkage contains "sysEvent"*/
  "zoneEvent": [1, 3],
/*optional, array, zone event type, this node is valid when linkage contains
"zone". [1,3] indicates that zone 1 and zone 3 are enabled. If this node is not
configured when linkage contains "zone", it indicates enabling all zones*/
  "subSystem": [1, 2, 3]
/*optional, array, linked partitions. When linkage is "alarm", "arming", or
"disarming", one or more partitions can be configured to link. For example,
[1,2,3] indicates that partition 1, partition 2, and partition 3 are linked*/
  },
  "duration":1,
/*optional, int, output duration of the relay, and the range is between 5 and
600 seconds*/
  "durationConstOutputEnable": true
/*optional, boolean, whether the duration (output duration of the relay) can be
configured: true-output duration cannot be configured (continuous output),
false-output duration can be configured*/
  }
}
```

A.3.125 JSON_OutPutsModuleCap

JSON message about relay's linkage configuration capability when the relay is closed/open

```
{
  "OutPutModuleCap": {
    "id": {
/*optional, int, relay ID*/
      "@min": 1,
      "@max": 1
    },
    "name": {
/*optional, string, relay name*/
      "@min": ,
      "@max":
    },
    "OutPutCloseLinkage": {
/*optional, event linkage information when the relay is closed*/
      "linkage": {
```

```

/*optional, string, event types that can be linked to the relay: "alarm",
"arming", "disarming", "manualCtrl"-manual control, "zone", "sysEvent"*/
    "@opt": "alarm,arming,disarming,manualCtrl,zone,sysEvent"
},
    "alarmMinorType": {
/*optional, array, minor event type: "zoneAlarmTamper"-zone alarm and tampered
event, "exDevTamper"-peripheral device tampered, "hostTamper"-control panel
tampered, "emergency"-panic alarm, "medical"-medical alarm, "fire"-fire alarm,
"gas"-gas event. When linkage is "alarm", one or all minor event types can be
selected. The data type of elements in the array is string*/
    "@opt": ["zoneAlarmTamper", "exDevTamper", "hostTamper", "emergency",
"medical", "fire", "gas"]
    },
    "sysEventMinorType": {
/*optional, array, minor type of system event: "ACOutage"-AC power outage,
"lowVoltageOfBattery"-low voltage of battery, "telephoneOffLine"-telephone
offline, "networkAbnormal"-wired network disconnected,
"wirelessNetworkAbnormal"-wireless network disconnected, "harddiskException"-
hard disk exception, "485Exception"-system keyboard exception, "mBusException"-
MBUS exception, "3G4GSignalAbnormal"-3G/4G signal exception, "moduleOffline"-
module offline. This node is valid when linkage contains "sysEvent"*/
    "@opt": ["ACOutage", "lowVoltageOfBattery", "telephoneOffLine",
"networkAbnormal", "wirelessNetworkAbnormal", "harddiskException",
"485Exception", "mBusException", "3G4GSignalAbnormal",
"moduleOffline","hostOffline"]
    },
    "zoneEvent": {
/*optional, array, zone event type, this node is valid when linkage contains
"zone". [1,3] indicates that zone 1 and zone 3 are enabled. If this node is not
configured when linkage contains "zone", it indicates enabling all zones*/
    "@min": 1,
    "@max": 2,
/*optional, int, the maximum value of zone No.*/
    "@size": 2
/*optional, int, the maximum number of zones that can be linked*/
    },
    "subSystem": {
    "@min": 1,
    "@max": 2,
/*optional, int, the maximum value of partition No.*/
    "@size": 2
/*optional, int, the maximum number of partitions that can be linked*/
    }
    },
    "OutputOpenLinkage": {
/*optional, event linkage information when the relay is open*/
    "linkage": {
/*optional, string, event types that can be linked to the relay: "alarm",
"arming", "disarming", "manualCtrl"-manual control, "zone", "sysEvent"*/
    "@opt": "alarm,arming,disarming,manualCtrl,zone,sysEvent"
    },
    "alarmMinorType": {

```

```

/*optional, array, minor event type: "zoneAlarmTamper"-zone alarm and tampered
event, "exDevTamper"-peripheral device tampered, "hostTamper"-control panel
tampered, "emergency"-panic alarm, "medical"-medical alarm, "fire"-fire alarm,
"gas"-gas event. When linkage is "alarm", one or all minor event types can be
selected. The data type of elements in the array is string*/
    "@opt": ["zoneAlarmTamper", "exDevTamper", "hostTamper", "emergency",
"medical", "fire", "gas"]
    },
    "sysEventMinorType": {
/*optional, array, minor type of system event: "ACOutage"-AC power outage,
"lowVoltageOfBattery"-low voltage of battery, "telephoneOffLine"-telephone
offline, "networkAbnormal"-wired network disconnected,
"wirelessNetworkAbnormal"-wireless network disconnected, "harddiskException"-
hard disk exception, "485Exception"-system keyboard exception, "mBusException"-
MBUS exception, "3G4GSignalAbnormal"-3G/4G signal exception, "moduleOffline"-
module offline. This node is valid when linkage contains "sysEvent"*/
    "@opt": ["ACOutage", "lowVoltageOfBattery", "telephoneOffLine",
"networkAbnormal", "wirelessNetworkAbnormal", "harddiskException",
"485Exception", "mBusException", "3G4GSignalAbnormal",
"moduleOffline", "hostOffline"]
    },
    "zoneEvent": {
/*optional, array, zone event type, this node is valid when linkage contains
"zone". [1,3] indicates that zone 1 and zone 3 are enabled. If this node is not
configured when linkage contains "zone", it indicates enabling all zones*/
    "@min": 1,
    "@max": 2,
/*optional, int, the maximum value of zone No.*/
    "@size": 2
/*optional, int, the maximum number of zones that can be linked*/
    },
    "subSystem": {
    "@min": 1,
    "@max": 2,
/*optional, int, the maximum value of partition No.*/
    "@size ": 2
/*optional, int, the maximum number of partitions that can be linked*/
    }
    },
    "duration":{
/*optional, int, output duration of the relay, and the range is between 5 and
600 seconds*/
    "@min": 1,
    "@max":
    },
    "durationConstOutputEnable":{
/*optional, boolean, whether the duration (output duration of the relay) can be
configured: true-output duration cannot be configured (continuous output),
false-output duration can be configured*/
    "@opt":[true,false]
    }
}

```



```
}  
}
```

A.3.126 JSON_OutputsModuleCond

JSON message about the condition of searching for relay's linkage configuration parameters

```
{  
  "OutputsModuleCond": {  
    "searchID": "",  
    /*required, string, search ID, which is used to check whether the current  
    search requester is the same as the previous one. If they are the same, the  
    search record will be stored in the device to speed up the next search*/  
    "searchResultPosition": 0,  
    /*required, integer32, the end position of search result in result list. In a  
    single search, if you cannot get all the records in the result list, you can  
    mark the end position and get the following records after the marked position  
    in the next search*/  
    "maxResults": 30,  
    /*required, integer32, the maximum number of results that can be obtained in  
    this search, if the value of maxResults exceeds the range in the capability  
    message, the maximum number will be returned according to that in the  
    capability message*/  
    "outputModuleNo": ,  
    /*optional, int, linked output module No.*/  
    "moduleType": ""  
    /*optional, string, module type: "localWired"-local wired module, "extendWired"-  
    extended wired module, "localWireless"-local wireless module, "extendWireless"-  
    extended wireless module*/  
  }  
}
```

A.3.127 JSON_OutputsModuleSearch

JSON message about the result of searching for relay's linkage configuration parameters

```
{  
  "OutputsModuleSearch": {  
    "searchID": "",  
    /*required, string, search ID, which is used to check whether the current  
    search requester is the same as the previous one. If they are the same, the  
    search record will be stored in the device to speed up the next search*/  
    "responseStatusStrg": "OK",  
    /*required, string, search status description: "OK"-search completed, "MORE"-  
    more data to be searched, "NO MATCH"-no data found*/  
    "numOfMatches": 1,  
    /*required, integer32, number of matched results returned this time*/  
    "totalMatches": 1,  
    /*required, integer32, total number of matched results*/  
  }  
}
```

```

    "List": [{
        "id": 1,
/*optional, int, relay ID*/
        "OutPutModule": {
            "name": "",
/*optional, string, relay name*/
            "related": ,
/*read-only, boolean, whether the relay is linked to the physical relay (the
channel of the output module). For PUT method, this node is optional; for GET
method, this node is required*/
            "outputModuleNo": ,
/*dependent,read-only, int, linked output module No., this node is required
only when the related is true*/
            "channelNo": ,
/*dependent, int, read-only, channel No. of output module, this node is
required only when the related is "true"*/
            "moduleType": "",
/*optional, string, module type: "localWired"-local wired module, "extendWired"-
extended wired module, "localWireless"-local wireless module, "extendWireless"-
extended wireless module*/
            "address": ,
/*optional, int, read-only, module address, this node is only returned by wired
modules*/
            "linkageAddress": ,
/*optional, int, read-only, linked module address, this node is only returned
by wireless modules*/
            "moduleChannel": ,
/*optional, int, read-only, module channel No.*/
            "OutPutCloseLinkage": [{
/*optional, event linkage information when the relay is closed*/
                "linkage": "",
/*optional, string, event types that can be linked to the relay: "alarm",
"arming", "disarming", "manualCtrl"-manual control, "zone", "sysEvent"*/
                "alarmMinorType": ["zoneAlarmTamper", "exDevTamper", "hostTamper",
"emergency", "medical", "fire", "gas"],
/*optional, array, minor event type: "zoneAlarmTamper"-zone alarm and tampered
event, "exDevTamper"-peripheral device tampered, "hostTamper"-control panel
tampered, "emergency"-panic alarm, "medical"-medical alarm, "fire"-fire alarm,
"gas"-gas event. When linkage is "alarm", one or all minor event types can be
selected. The data type of elements in the array is string*/
                "sysEventMinorType": ["ACOutage", "lowVoltageOfBattery",
"telephoneOffLine", "networkAbnormal", "wirelessNetworkAbnormal",
"harddiskException", "485Exception", "mBusException", "3G4GSignalAbnormal",
"moduleOffline"],
/*optional, array, minor type of system event: "ACOutage"-AC power outage,
"lowVoltageOfBattery"-low voltage of battery, "telephoneOffLine"-telephone
offline, "networkAbnormal"-wired network disconnected,
"wirelessNetworkAbnormal"-wireless network disconnected, "harddiskException"-
hard disk exception, "485Exception"-system keyboard exception, "mBusException"-
MBUS exception, "3G4GSignalAbnormal"-3G/4G signal exception, "moduleOffline"-
module offline. This node is valid when linkage contains "sysEvent"*/
                "zoneEvent": [1, 3],

```

```
/*optional, array, zone event type, this node is valid when linkage contains
"zone". [1,3] indicates that zone 1 and zone 3 are enabled. If this node is not
configured when linkage contains "zone", it indicates enabling all zones*/
    "subSystem": [1, 2, 3]
/*optional, array, linked partitions. When linkage is "alarm", "arming", or
"disarming", one or more partitions can be configured to link. For example,
[1,2,3] indicates that partition 1, partition 2, and partition 3 are linked*/
    },
    "OutputOpenLinkage": [{
/*optional, event linkage information when the relay is open*/
        "linkage": "",
/*optional, string, event types that can be linked to the relay: "alarm",
"arming", "disarming", "manualCtrl"-manual control, "zone", "sysEvent"*/
        "alarmMinorType": ["zoneAlarmTamper", "exDevTamper", "hostTamper",
"emergency", "medical", "fire", "gas"],
/*optional, array, minor event type: "zoneAlarmTamper"-zone alarm and tampered
event, "exDevTamper"-peripheral device tampered, "hostTamper"-control panel
tampered, "emergency"-panic alarm, "medical"-medical alarm, "fire"-fire alarm,
"gas"-gas event. When linkage is "alarm", one or all minor event types can be
selected. The data type of elements in the array is string*/
        "sysEventMinorType": ["ACOutage", "lowVoltageOfBattery",
"telephoneOffLine", "networkAbnormal", "wirelessNetworkAbnormal",
"harddiskException", "485Exception", "mBusException", "3G4GSignalAbnormal",
"moduleOffline"],
/*optional, array, minor type of system event: "ACOutage"-AC power outage,
"lowVoltageOfBattery"-low voltage of battery, "telephoneOffLine"-telephone
offline, "networkAbnormal"-wired network disconnected,
"wirelessNetworkAbnormal"-wireless network disconnected, "harddiskException"-
hard disk exception, "485Exception"-system keyboard exception, "mBusException"-
MBUS exception, "3G4GSignalAbnormal"-3G/4G signal exception, "moduleOffline"-
module offline. This node is valid when linkage contains "sysEvent"*/
        "zoneEvent": [1, 3],
/*optional, array, zone event type, this node is valid when linkage contains
"zone". [1,3] indicates that zone 1 and zone 3 are enabled. If this node is not
configured when linkage contains "zone", it indicates enabling all zones*/
        "subSystem": [1, 2, 3]
/*optional, array, linked partitions. When linkage is "alarm", "arming", or
"disarming", one or more partitions can be configured to link. For example,
[1,2,3] indicates that partition 1, partition 2, and partition 3 are linked*/
        },
        "duration":1,
/*optional, int, output duration of the relay, and the range is between 5 and
600 seconds*/
        "durationConstOutputEnable": true
/*optional, boolean, whether the duration (output duration of the relay) can be
configured: true-output duration cannot be configured (continuous output),
false-output duration can be configured*/
    }
    }
}
```

A.3.128 JSON_PanicButton

Message about panic button parameters of a specific zone in JSON format.

```
{
  "PanicButton":{
    "LEDEnabled": ,
    /*optional, boolean, whether to enable the LED indicator*/
    "LEDLatchTime": ,
    /*optional, int, delay time of the LED indicator, unit: second*/
    "findMeEnabled": ,
    /*optional, boolean, whether to enable the Find Me function*/
    "alarmMode":"","
    /*optional, string, alarm mode. When the panic alarm is triggered, alarms of
    the corresponding alarm mode will be uploaded*/
    "accidentalPressProtection":"","
    /*optional, string, protection method to avoid triggering unintentionally*/
    "panicButtonType":"","
    /*optional, read-only, string, panic button type*/
    "heartBeatInterval": ,
    /*optional, int, heartbeat interval of the security control panel and the
    peripheral, unit: second*/
    "pollingOptionEnable":
    /*optional, boolean, whether to disable detecting heartbeat of the security
    control panel and the peripheral*/
    "workMode": "detector",
    /*optional, string, work mode: "detector", "autoControl" (control repeater)*/
    "supportAssociatedRelay": [1, 2, 3],
    /*optional, int, the repeaters that can be linked; this node is valid when the
    value of workMode is "autoControl"*/
    "associateRelayCfg": [1, 2, 3],
    /*optional, array, the repeaters that have been linked with the zone; this node
    is valid when the value of workMode is "autoControl"*/
    "triggerMode": ["longPress"],
    /*optional, string, trigger mode of the panic button*/
    "confirmAlarmInterval": 8
    /*optional, int, time interval for uploading acknowledgment alarm, unit: hour,
    range:[8,20]*/
  }
}
```

A.3.129 JSON_PanicButtonCap

JSON message about the configuration capability of the panic button

```
{
  "PanicButtonCap":{
    "zoneNo":{
    /*optional, int, values that can be configured as the zone No.*/
```

```
        "@opt": [1, 3, 5]
    },
    "LEDEnabled": {
/*optional, boolean, whether to enable the LED indicator*/
        "@opt": [true, false]
    },
    "LEDLatchTime": {
/*optional, int, delay time of the LED indicator, unit: second*/
        "@min": 0,
        "@max": 0
    },
    "findMeEnabled": {
/*optional, boolean, whether to enable the Find Me function*/
        "@opt": [true, false]
    },
    "alarmMode": {
/*optional, string, alarm mode: "emergency"-panic alarm, "medical"-medical
alarm*/
        "@opt": ["emergency", "medical"]
    },
    "supportZoneType": {
/*optional, string, zone types supported by the peripheral: "Instant"-instant
zone, "Delay"-delay zone, "Follow"-follow zone, "Perimeter"-perimeter zone,
"24hNoSound"-24-hour silent zone, "Emergency"-panic zone, "Fire"-fire zone,
"Gas"-gas zone, "Medical"-medical zone, "Timeout"-timeout zone, "Non-Alarm"-
disabled zone, "Key"-key zone, "24hSound"-24-hour annunciating zone. When
switching zone type, zone types supported by the peripheral can be obtained*/
        "@opt": ["Instant", "Delay", "Follow"]
    },
    "PanicButtonTypeList": [{
        "panicButtonType": "",
/*optional, read-only, string, panic button type: "poratble"-portable,
"hanging"-wall mounting*/
        "pollingOptionEnable": {
/*optional, boolean, whether to disable detecting heartbeat of the security
control panel and the peripheral: true=yes. If this function is not supported,
this field will not be returned*/
            "@opt": [true, false]
        },
        "accidentalPressProtection": {
/*optional, string, protection method to avoid triggering unintentionally:
"double"-press twice to trigger, "delay"-press and hold to trigger, "off"-
disable*/
            "@opt": ["double", "delay", "off"]
        }
    }],
    "heartBeatInterval": {
/*optional, int, heartbeat interval of the security control panel and the
peripheral*/
        "@opt": [5, 10, 20, 30]
    },
    "triggerMode": {
```

```
/*optional, object, trigger mode of the panic button: "longPress",
"shortPress", "doublePress"*/
    "@size": 2,
/*optional, int, the maximum types of mode that can be configured at the same
time*/
    "@opt": ["longPress", "shortPress", "doublePress"]
/*optional, array of string*/
    },
    "confirmAlarmInterval": {
/*optional, object, time interval for uploading acknowledgment alarm, unit:
hour; if you have selected two trigger modes, when the button is triggered in
the first mode, a normal emergency alarm will be uploaded; if the button is
triggered in the second mode within the configured time interval, then an
acknowledgment emergency alarm will be uploaded*/
        "@min": 8,
/*optional, int, the minimum value*/
        "@max": 20
/*optional, int, the maximum value*/
    }
}
}
```

A.3.130 JSON_PassiveInfraredDetectorCap

JSON message about the configuration capability of the PIR (Passive Infrared) detector

```
{
  "PassiveInfraredDetectorCap": {
    "zoneNo": {
/*optional, int, values that can be configured as the zone No.*/
        "@opt": [1,3,5]
    },
    "supportZoneType":{
/*optional, string, zone types supported by the peripheral: "Instant"-instant
zone, "Delay"-delay zone, "Follow"-follow zone, "Perimeter"-perimeter zone,
"24hNoSound"-24-hour silent zone, "Emergency"-panic zone, "Fire"-fire zone,
"Gas"-gas zone, "Medical"-medical zone, "Timeout"-timeout zone, "Non-Alarm"-
disabled zone, "Key"-key zone, "24hSound"-24-hour annunciating zone. When
switching zone type, zone types supported by the peripheral can be obtained*/
        "@opt": ["Instant", "Delay", "Follow"]
    },
    "alwaysActiveEnabled": {
/*optional, object, read-only, whether to enable keeping detector working even
after disarming*/
        "@opt": [true, false]
    },
    "heartBeatInterval":{
/*optional, int, heartbeat interval of the security control panel and the
peripheral, unit: second*/
        "@opt": [5,10,20,30]
    },
  },
}
```

```
"LEDEnabled": {
/*optional, boolean, whether to enable the LED indicator*/
    "@opt": [true, false]
},
"LEDLatchTime": {
/*optional, int, delay time of the LED indicator, unit: second*/
    "@min": 0,
    "@max": 0
},
"sensitivityLevel": {
/*optional, string, sensitivity level: "high", "auto", "antiPet"-pet immune*/
    "@opt": ["high", "auto", "antiPet"]
},
"triggerNumLimited": {
/*optional, object, read-only, limited number of trigger times*/
    "@min": 0,
/*optional, int, read-only, the minimum value*/
    "@max": 10
/*optional, int, read-only, the maximum value*/
},
"isSupportSignalTest": true,
/*optional, read-only, boolean, whether it supports signal strength detection,
if this node is not returned or if the value is false, it indicates that this
function is not supported*/
"isSupportZoneTest": true,
/*optional, read-only, boolean, whether it supports zone detection, if this
node is not returned or if the value is false, it indicate that this function
is not supported*/
"isSupportFindMe": true,
/*optional, read-only, boolean, whether it supports FindMe detection, if this
node is not returned or id the value is false, it indicates that this function
is not supported*/
"detectorType": {
/*object, optional, read-only, detector type: "normal" (PIR detector),
"ceiling"(PIR ceiling detector)*/
    "@opt": ["normal", "ceiling"]
},
"isSupportDoubleKnock": true,
/*optional, boolean, read-only, whether it supports double knock, if this node
is not returned or if the value is false, it indicates that this function is
not supported*/
"CeilingNode": {
/*optional, object, read-only, node for PIR ceiling detector, this node is
valid when the value of detectorType is "ceiling"*/
    "sensitivityLevel": {
/*optional, object, read-only, sensitivity: "auto", "low"*/
        "@opt": ["auto", "low"]
    }
}
}
```

A.3.131 JSON_PassiveInfraredDetector

JSON message about the PIR (Passive Infrared) detector parameters of a specific zone

```
{
  "PassiveInfraredDetector": {
    "LEDEnabled": ,
    /*optional, boolean, whether to enable the LED indicator*/
    "LEDLatchTime": ,
    /*optional, int, delay time of the LED indicator, unit: second*/
    "findMeEnabled": ,
    /*optional, boolean, whether to enable the Fine Me function*/
    "sensitivityLevel": "",
    /*optional, string, sensitivity level: "high","auto","antiPet"-pet immune*/
    "alwaysActiveEnabled": true,
    /*optional, boolean, whether to enable keeping detector working even after
    disarming*/
    "heartBeatInterval": 12,
    /*optional, int, heartbeat interval of the security control panel and the
    peripheral, unit: second*/
    "triggerNumLimited": 5,
    /*optional, int, limited number of trigger times*/
    "detectorType": "normal"
    /*object, optional, read-only, detector type: "normal" (PIR detector),
    "ceiling"(PIR ceiling detector)*/
  }
}
```

A.3.132 JSON_Phone

Phone message in JSON format

```
{
  "Phone":{
    "id": ,
    /*required, integer type, phone No.*/
    "numbers": "",
    /**required, string type, phone number*/
    "messageEnabled": ,
    /*optional,boolean type, whether to enable message notification, "true, false"*/
    "callEnabled": ,
    /*optional, boolean type, whether to enable phone call notification, "true,
    false"*/
    "alarmTamperEnabled": ,
    /*optional, boolean type, whether to enable alarm and tampering event
    notification, "true, false"*/
    "lifeSecurityEnabled": ,
    /*optional, boolean type, whether to enable life security event notification,
    "true, false"*/
  }
}
```



```
    "systemStatusEnabled": ,
    /*optional, boolean type, whether to enable system status event notification,
    "true, false"*/
    "operateEventEnabled":
    /*optional, boolean type, whether to enable operation event notification,
    "true, false"*/
    }
}
```

A.3.133 JSON_Picture

Picture message in JSON format

```
{
  "Picture":{
    "captureURI":""
    /*optional, string, URI returned by the device after the picture is captured,
    and it is used for display and live view*/
  }
}
```

A.3.134 JSON_pircam

JSON message about the pircam parameters of a specific zone

```
{
  "pircam":{
    "enabled": ,
    /*required, boolean, whether to enable pircam (detector equipped with camera)
    configuration*/
    "channelNo": ,
    /*optional, integer, pircam (detector equipped with camera) channel No.*/
    "picColorResolution":"","
    /*optional, string, picture resolution*/
    "zoneName":"","
    /*optional, read-only, string, zone name*/
    "camEnable": ,
    /*optional, boolean, whether to enable the camera*/
    "picNum": ,
    /*optional, int, number of pictures. For pictures whose resolution is 640*480,
    this field is between 0 and 10; for other resolution, this field is between 0
    and 20*/
    "picInterval": ,
    /*optional, float, picture capture interval, unit: second*/
    "picQoc":"","
    /*optional, string, picture quality: "20%","40%","60%"/
    "detectInterval": ,
    /*optional, int, detection interval (valid alarm duration)*/
    "LEDEnable": ,
```

```
/*optional, boolean, whether to enable the LED indicator*/
    "LEDLatchTime": ,
/*optional, int, delay time of the LED indicator, unit: second*/
    "devDetectEnable": ,
/*optional, boolean, whether to enable device detection*/
    "signGainCfg": "",
/*optional, string, pircam detector gain*/
    "petImmueFilter": "",
/*optional, boolean, whether the pet can trigger the pircam alarm: true=yes*/
    "pulseFilterCfg": ,
/*optional, int, number of impulse waves*/
    "holdOffTime": ,
/*optional, int, delay time, 0 indicates no delay, unit: second*/
    "jpegModeCfg": "",
/*optional, string, format of pictures captured by pircam*/
    "sensitivityLevel": "",
/*optional, string, sensitivity level: "high", "auto", "antiPet"-pet immune*/
    "findMeEnabled": ,
/*optional, boolean, whether to enable the Find Me function*/
    "climeEnabled": ,
/*optional, boolean, whether to enable muting*/
    "videoResolution": "",
/*optional, string, video resolution*/
    "picMode": "",
/*optional, string, picture mode: "blackAndWhite"-black and white, "color"*/
    "videoTime": ,
/*optional, int, video duration, unit: second*/
    "triggerTime": ,
/*optional, int, alarm interval, unit: second*/
    "triggerNum": ,
/*optional, int, alarm triggering times. When the number of detector alarms
reaches the configured limit and there is no alarm uploaded by any other zone
in the whole security control system, the pircam will not trigger alarm again;
if there are alarms uploaded by other zones, this field will be set to 0 and
the alarm triggering times will be calculated again*/
    "linkageCaptureType": ,
/*optional, string, linkage action after the alarm is triggered: "picture"-the
pircam will capture pictures, "4seconds video"-record a 4-second video,
"8seconds video"-record an 8-second video*/
    "heartBeatInterval": ,
/*optional, int, heartbeat interval of the security control panel and the
peripheral, unit: second*/
    "frame": ,
/*optional, int, frame rate*/
    "alwaysActiveEnabled": true,
/*optional, boolean, whether to enable detector working all the time even if
the zone is disarmed*/
    "operateTime": "1970-01-01T00:00:00+08:00",
/*optional, datetime, operation time in ISO 8601 time format, this node should
be used together with "nextArmTime"*/
    "nextArmTime": 3
/*optional, int, time before the next arming: 3, 6, 12, 24, 48, 96, 192, unit:
```

```
h*/  
}  
}
```

A.3.135 JSON_Pircam

JSON message about the pircam (detector equipped with camera) capture parameters being added currently in asynchronous mode

```
{  
  "Pircam":{  
    "status": "",  
    /*required, string, current status: "processing", "success", "failed"*/  
    "picURL": ""  
    /*optional, string, picture URL captured by the pircam (detector equipped with  
    camera). This URL can be used by the upper-layer application to download the  
    picture after it is returned by the device*/  
  }  
}
```

A.3.136 JSON_PircamCap

JSON message about the pircam configuration capability

```
{  
  "PircamCap":{  
    "enabled":[true, false],  
    /*required, boolean, whether to enable pircam (detector equipped with camera)  
    settings*/  
    "picColorResolution":{  
      /*optional, string, picture resolution: "VGA (640x480)", "QVGA (320x240)",  
      "QQVGA (160x120)"*/  
      "@opt":["VGA (640x480)", "QVGA (320x240)", "QQVGA (160x120)"]  
    },  
    "channelNo":{  
      /*optional, int, value range of pircam (detector equipped with camera) channel  
      No.*/  
      "@min":0,  
      "@max":0  
    },  
    "zoneNo":{  
      /*optional, int, values that can be configured as the zone No.*/  
      "@opt":[1, 3, 5]  
    },  
    "zoneName":{  
      /*optional, int, zone name length*/  
      "@min":0,  
      "@max":0  
    }  
  },  
}
```

```
"picNum":{
/*number of pictures*/
  "160*120PicNum":{
/*optional, int, number of pictures whose resolution is 160*120*/
    "@min":0,
    "@max":0
  },
  "320*240PicNum":{
/*optional, int, number of pictures whose resolution is 320*240*/
    "@min":0,
    "@max":0
  },
  "640x480PicNum":{
/*optional, int, number of pictures whose resolution is 640x480*/
    "@min":0,
    "@max":0
  }
},
"picInterval":{
/*optional, int, picture capture interval*/
  "@opt":[0.5,1,2.5]
},
"picQoc":{
/*optional, string, picture quality: "20%", "40%", "60%"/
  "@opt":["20%", "40%", "60%"]
},
"detectInterval":{
/*optional, int, detection time interval (a valid alarm's duration)*/
  "@min":0,
  "@max":0
},
"LEDEnable":"true,false",
/*optional, boolean, whether to enable the LED indicator*/
"LEDLatchTime":{
/*optional, int, delay time of the LED indicator, unit: second*/
  "@min":0,
  "@max":0
},
"devDetectEnable":"true,false",
/*optional, boolean, whether to enable device detection*/
"signGainCfg":{
/*optional, string, pircam gain: "Lowest","Medium low","Medium","Medium
high","Highest"*/
  "@opt":["Lowest", "Medium low", "Medium", "Medium high", "Highest"]
},
"petImmueFilter":[true, false],
/*optional, boolean, whether the pet can trigger the pircam alarm: true=yes*/
"pulseFilterCfg":{
/*optional, int, number of impulse waves*/
  "@opt":[1, 2, 3]
},
"holdOffTime":{
```

```
/*optional, int, delay time, 0 indicates no delay, unit: second*/
    "@opt":[30, 45, 60, 75, 90]
},
"jpegModeCfg":{
/*optional, string, format of pictures captured by pircam*/
    "@opt":["Regular JPEG", "Differential JPEG"]
},
"sensitivityLevel":{
/*optional, string, sensitivity level: "high", "auto", "antiPet"-pet immune*/
    "@opt":["high", "auto", "antiPet"]
},
"findMeEnabled":{
/*optional, boolean, whether to enable the Find Me function*/
    "@opt":[true, false]
},
"climeEnabled":{
/*optional, boolean, whether to enable muting*/
    "@opt":[true, false]
},
"videoResolution":{
/*optional, string, video resolution*/
    "@opt":["VGA (640x480)", "QVGA (320x240)", "QQVGA (160x120)"]
},
"picMode":{
/*optional, string, picture mode: "blackAndWhite"-black and white, "color"*/
    "@opt":["blackAndWhite", "color"]
},
"videoTime":{
/*optional, int, video duration, unit: second*/
    "@min":0,
    "@max":0
},
"triggerNum":{
/*optional, int, alarm triggering times*/
    "@min":0,
    "@max":0
},
"triggerTime":{
/*optional, int, alarm interval, unit: second*/
    "@opt":[10, 20, 30]
},
"linkageCaptureType":{
/*optional, string, linkage action after the alarm is triggered: "picture"-the
pircam detector will capture pictures, "4seconds video"-record 4s' video,
"8seconds video"-record a 8s' video*/
    "@opt":["picture", "4seconds video", "8seconds video"]
},
"supportZoneType":{
/*optional, string, zone types supported by the peripheral: "Instant"-instant
zone, "Delay"-delay zone, "Follow"-follow zone, "Perimeter"-perimeter zone,
"24hNoSound"-24-hour silent zone, "Emergency"-panic zone, "Fire"-fire zone,
"Gas"-gas zone, "Medical"-medical zone, "Timeout"-timeout zone, "Non-Alarm"-
```

```
disabled zone, "Key"-key zone, "24hSound"-24-hour annunciating zone. When
switching zone type, zone types supported by the peripheral can be obtained*/
"@opt":["Instant", "Delay", "Follow"]
},
"heartbeatInterval":{
/*optional, int, heartbeat interval of the security control panel and the
peripheral, unit: second*/
"@opt":[5, 10, 20, 30]
},
"frame":{
/*optional, int, frame rate which refers to the number of pictures captured
each second*/
"@opt":[1, 2, 3],
"alwaysActiveEnabled": true,
/*optional, boolean, whether to enable detector working all the time even if
the zone is disarmed*/
"operateTime": "1970-01-01T00:00:00+08:00",
/*optional, datetime, operation time in ISO 8601 time format, this node should
be used together with "nextArmTime"*/
"nextArmTime": 3
/*optional, int, time before the next arming: 3, 6, 12, 24, 48, 96, 192, unit:
h*/
"@opt": [3, 6, 12, 24, 48, 96, 192]
}
}
}
```

A.3.137 JSON_PircamMode

JSON message about parameters of controlling the pircam (detector equipped with camera) to capture pictures or record videos in asynchronous mode

```
{
  "PircamMode":{
    "mode":""
/*required, string, mode: "enter", "exit"*/
  }
}
```

A.3.138 JSON_PSTNCfg

JSON message about configuration parameters of a specific phone notification via PSTN

```
{
  "PSTNCfg":{
    "id": ,
/*required, int, uploading index*/
    "enabled": ,
/*required, boolean, whether to enable*/
  }
}
```

```
    "name": "",
/*optional, string, alarm receiving center name*/
    "phoneNum": "",
/*optional, string, phone number of alarm receiving center*/
    "repeatCall": "",
/*optional, int, number of repeated callings*/
    "protocol": "",
/*optional, string, communication protocol: "CID"-CID communication*/
    "transMode": ,
/*optional, int, transmission mode: 0-DTMF 5/S, 1-DTMF 10/S*/
    "receiverId": "",
/*optional, string, receiver user name*/
    "reportPeriod": ,
/*optional, int, test report uploading period, unit: hour*/
    "reportPeriodEnabled": ,
/*optional, boolean, whether to enable test report uploading period*/
    "firstReportTime":
/*optional, int, duration from launching the device to uploading the first test
report, unit: minute*/
    }
}
```

A.3.139 JSON_PSTNCfgCap

JSON message about configuration capability of phone notification via PSTN

```
{
  "PSTNCfgCap":{
    "id":{
/*required, uploading index*/
      "@min": ,
      "@max":
    },
    "enabled":"true,false",
/*required, whether to enable*/
    "name":{
/*optional, alarm receiving center name*/
      "@min": ,
      "@max":
    },
    "phoneNum":{
/*optional, phone number of alarm receiving center*/
      "@min": ,
      "@max":
    },
    "repeatCall":{
/*optional, number of repeated callings*/
      "@min": ,
      "@max":
    },
    "protocol":{
```

```
/*optional, communication protocol: "CID"-CID communication*/
  "@opt":["CID"]
},
"transMode":{
/*optional, transmission mode: 0-DTMF 5/S, 1-DTMF 10/S*/
  "@opt":[0,1]
},
"receiverId":{
/*optional, string type, receiver user name*/
  "@min": ,
  "@max":
},
"reportPeriod":{
/*optional, test report uploading period, unit: hour*/
  "@min": ,
  "@max":
},
"reportPeriodEnabled": {
/*optional, whether to enable test report uploading period*/
  "@opt": [true,false]
},
"firstReportTime":{
/*optional, duration from launching the device to uploading the first test
report, unit: minute*/
  "@min": ,
  "@max":
}
}
}
```

A.3.140 JSON_PhoneAnvanced

JSON message about the advanced notification parameters of a specific phone number

```
{
  "PhoneAnvanced":{
    "id": ,
/*required, integer type, phone number ID*/
    "numbers":"",
/*required, string type, phone number*/
    "messageEnabled": ,
/*optional, boolean type, whether to enable SMS notification*/
    "callEnabled": ,
/*optional, boolean type, whether to enable phone notification*/
    "Message":{
/*this node is valid when "messageEnabled" is "true"*/
      "alarmTamperEnabled": ,
/*optional, boolean type, whether to enable alarm and tampering event
notification*/
      "lifeSecurityEnabled": ,
/*optional, boolean type, whether to enable life safety event notification*/
```



```
    "systemStatusEnabled": ,
/*optional, boolean type, whether to enable system status event notification*/
    "operateEventEnabled": ,
/*optional, boolean type, whether to enable operation event notification*/
    "zoneAlarmTamperEnabled": ,
/*optional, boolean, whether to enable alarm and tampering event notification
of the supported zone: "true"-yes, "false"-no*/
    "exDevTamperEventEnabled": ,
/*optional, boolean, whether to enable peripheral tampering alarm notification:
"true"-yes, "false"-no*/
    "hostTamperEventEnabled": ,
/*optional, boolean, whether to enable tampering alarm notification of security
control panel: "true"-yes, "false"-no*/
    "emergencyEventEnabled": ,
/*optional, boolean, whether to enable panic alarm notification: "true"-yes,
"false"-no*/
    "medicalEventEnabled": ,
/*optional, boolean, whether to enable medical alarm notification: "true"-yes,
"false"-no*/
    "gasEventEnabled": ,
/*optional, boolean, whether to enable gas alarm notification: "true"-yes,
"false"-no*/
    "fireEventEnabled": ,
/*optional, boolean, whether to enable fire alarm notification: "true"-yes,
"false"-no*/
    "hostStatusEventEnabled": ,
/*optional, boolean, whether to enable notification of security control panel
status: "true"-yes, "false"-no*/
    "exDevStatusEventEnabled": ,
/*optional, boolean, whether to enable peripheral status notification: "true"-
yes, "false"-no*/
    "detectorStatusEventEnabled": ,
/*optional, boolean, whether to enable detector status notification: "true"-
yes, "false"-no*/
    "intelligentAlarmEnable": ,
/*optional, boolean, whether to enable smart alarm notification: "true"-yes,
"false"-no*/
    "arm": ,
/*optional, array, arming permission, e.g., [1,2,3] indicates having permission
to arm partition 1, partition 2, and partition 3*/
    "disarm": ,
/*optional, array, disarming permission, e.g., [1,2,3] indicates having
permission to disarm partition 1, partition 2, and partition 3*/
    "clearAlarm": ,
/*optional, array, alarm clearing permission, e.g., [1,2,3] indicates having
permission to clear alarms of partition 1, partition 2, and partition 3*/
    "timeFilterEnabled": true,
/*optional, boolean, whether to enable uploading event details by message only
within the configured period: true, false(uploading the messages all the time)*/
    "startTime": "10:00:00",
/*optional, time, start time, this node is valid when timeFilterEnabled is
true*/
```

```
        "endTime": "16:00:00",
/*optional, time, end time, this node is valid when timeFilterEnabled is true*/
        "WeekPlanCfg": [
/*optional, array, week schedule information, range:[1,7], this node is valid
when timeFilterEnabled is true*/
            {
                "dayOfWeek": 1,
/*required, int, day of the week, range:[1,7]*/
                "TimeRange": [
/*required, array, period, range:[1,8]*/
                    {
                        "startTime": "10:00:00",
/*required, time, start time*/
                        "endTime": "16:00:00"
/*required, time, end time*/
                    }
                ]
            }
        ],
    },
    "Call":{
/*this node is valid when "callEnabled" is "true"*/
        "alarmTamperEnabled": ,
/*optional, boolean type, whether to enable alarm and tampering event
notification*/
        "lifeSecurityEnabled": ,
/*optional, boolean type, whether to enable life safety event notification*/
        "systemStatusEnabled": ,
/*optional, boolean type, whether to enable system status event notification*/
        "operateEventEnabled": ,
/*optional, boolean type, whether to enable operation event notification*/
        "zoneAlarmTamperEnabled": ,
/*optional, boolean, whether to enable alarm and tampering event notification
of the supported zone: "true"-yes, "false"-no*/
        "exDevTamperEventEnabled": ,
/*optional, boolean, whether to enable peripheral tampering alarm notification:
"true"-yes, "false"-no*/
        "hostTamperEventEnabled": ,
/*optional, boolean, whether to enable tampering alarm notification of security
control panel: "true"-yes, "false"-no*/
        "emergencyEventEnabled": ,
/*optional, boolean, whether to enable panic alarm notification: "true"-yes,
"false"-no*/
        "medicalEventEnabled": ,
/*optional, boolean, whether to enable medical alarm notification: "true"-yes,
"false"-no*/
        "gasEventEnabled": ,
/*optional, boolean, whether to enable gas alarm notification: "true"-yes,
"false"-no*/
        "fireEventEnabled": ,
/*optional, boolean, whether to enable fire alarm notification: "true"-yes,
"false"-no*/
    }
```

```

        "hostStatusEventEnabled": ,
/*optional, boolean, whether to enable notification of security control panel
status: "true"-yes, "false"-no*/
        "exDevStatusEventEnabled": ,
/*optional, boolean, whether to enable peripheral status notification: "true"-
yes, "false"-no*/
        "detectorStatusEventEnabled": ,
/*optional, boolean, whether to enable detector status notification: "true"-
yes, "false"-no*/
        "intelligentAlarmEnable": ,
/*optional, boolean, whether to enable smart alarm notification: "true"-yes,
"false"-no*/
        "numbersOfCalls": ,
/*optional, integer, phone call times*/
        "timeFilterEnabled": true,
/*optional, boolean, whether to enable uploading event details by call only
within the configured period: true, false(uploading the calls all the time)*/
        "startTime": "10:00:00",
/*optional, time, start time, this node is valid when timeFilterEnabled is
true*/
        "endTime": "16:00:00",
/*optional, time, end time, this node is valid when timeFilterEnabled is true*/
        "WeekPlanCfg": [
/*optional, array, week schedule information, range:[1,7], this node is valid
when timeFilterEnabled is true*/
            {
                "dayOfWeek": 1,
/*required, int, day of the week, range:[1,7]*/
                "TimeRange": [
/*required, array, period, range:[1,8]*/
                    {
                        "startTime": "10:00:00",
/*required, time, start time*/
                        "endTime": "16:00:00"
/*required, time, end time*/
                    }
                ]
            }
        ]
    }
}

```

A.3.141 JSON_PhoneAnvancedCap

JSON message about the advanced configuration capability of the phone notification

```

{
    "PhoneAnvancedCap":{
        "id":{
/*required, phone No. range*/

```

```
    "@min": ,
    "@max":
  },
  "messageEnabled":true,
  /*optional, whether to support enabling SMS notification*/
  "callEnabled":true,
  /*optional, whether to support enabling phone notification*/
  "numLength":32,
  /*required, maximum length of the phone number*/
  "Message":{
    "alarmTamperEnabled":true,
    /*optional, whether to support enabling alarm and tampering event
notification*/
    "lifeSecurityEnabled":true,
    /*optional, whether to support enabling life safety event
notification*/
    "systemStatusEnabled":true,
    /*optional, whether to support enabling system status event
notification*/
    "operateEventEnabled":true,
    /*optional, whether to support enabling operation event notification*/
    "zoneAlarmTamperEnabled":true,
    /*optional, boolean, whether to enable alarm and tampering event notification
of the supported zone: "true"-yes, "false"-no*/
    "exDevTamperEventEnabled":true,
    /*optional, boolean, whether to enable peripheral tampering alarm notification:
"true"-yes, "false"-no*/
    "hostTamperEventEnabled":true,
    /*optional, boolean, whether to enable tampering alarm notification of security
control panel: "true"-yes, "false"-no*/
    "emergencyEventEnabled":true,
    /*optional, boolean, whether to enable panic alarm notification: "true"-yes,
"false"-no*/
    "medicalEventEnabled":true,
    /*optional, boolean, whether to enable medical alarm notification: "true"-yes,
"false"-no*/
    "gasEventEnabled":true,
    /*optional, boolean, whether to enable gas alarm notification: "true"-yes,
"false"-no*/
    "fireEventEnabled":true,
    /*optional, boolean, whether to enable fire alarm notification: "true"-yes,
"false"-no*/
    "hostStatusEventEnabled":true,
    /*optional, boolean, whether to enable notification of security control panel
status: "true"-yes, "false"-no*/
    "exDevStatusEventEnabled":true,
    /*optional, boolean, whether to enable peripheral status notification: "true"-
yes, "false"-no*/
    "detectorStatusEventEnabled":true,
    /*optional, boolean, whether to enable detector status notification: "true"-
yes, "false"-no*/
    "intelligentAlarmEnable":true,
```

```
/*optional, boolean, whether to enable smart alarm notification: "true"-yes,
"false"-no*/
    "arm":{
/*optional, supported partition range of the arming permission that can be
configured*/
        "@min": ,
        "@max":
    },
    "disarm":{
/*optional, supported partition range of the disarming permission that can be
configured*/
        "@min": ,
        "@max":
    },
    "clearAlarm":{
/*optional, supported partition range of the alarm clearing permission that can
be configured*/
        "@min": ,
        "@max":
    },
    "timeFilterEnabled": {
/*optional, boolean, whether to enable uploading event details by message only
within the configured period: true, false(uploading the messages all the time)*/
        "@opt": [true, false]
    },
    "startTime": "10:00:00",
/*optional, time, start time, this node is valid when timeFilterEnabled is
true*/
    "endTime": "16:00:00",
/*optional, time, end time, this node is valid when timeFilterEnabled is true*/
    "WeekPlanCfg": {
/*optional, object, week schedule information, this node is valid when
timeFilterEnabled is true*/
        "@size": 7,
/*optional, int, the maximum number of days*/
        "dayOfWeek": {
/*required, object, day of the week*/
            "@min": 1,
/*optional, int, the minimum value, range:[1,7]*/
            "@max": 7
/*optional, int, the maximum value, range:[1,7]*/
        },
        "TimeRange": {
/*required, object, period*/
            "@size": 8,
/*optional, int, the maximum number of periods, range:[1,8]*/
            "startTime": "15:00:00",
/*required, time, start time*/
            "endTime": "20:00:00"
/*required, time, end time*/
        }
    }
}
```

```
    },
    "Call":{
        "alarmTamperEnabled":true,
        /*optional, whether to support enabling alarm and tampering event notification*/
        "lifeSecurityEnabled":true,
        /*optional, whether to support enabling life safety event notification*/
        "systemStatusEnabled":true,
        /*optional, whether to support enabling system status event notification*/
        "operateEventEnabled":true,
        /*optional, whether to support enabling operation event notification*/
        "zoneAlarmTamperEnabled":true,
        /*optional, boolean, whether to enable alarm and tampering event notification
        of the supported zone: "true"-yes, "false"-no*/
        "exDevTamperEventEnabled":true,
        /*optional, boolean, whether to enable peripheral tampering alarm notification:
        "true"-yes, "false"-no*/
        "hostTamperEventEnabled":true,
        /*optional, boolean, whether to enable tampering alarm notification of security
        control panel: "true"-yes, "false"-no*/
        "emergencyEventEnabled":true,
        /*optional, boolean, whether to enable panic alarm notification: "true"-yes,
        "false"-no*/
        "medicalEventEnabled":true,
        /*optional, boolean, whether to enable medical alarm notification: "true"-yes,
        "false"-no*/
        "gasEventEnabled":true,
        /*optional, boolean, whether to enable gas alarm notification: "true"-yes,
        "false"-no*/
        "fireEventEnabled":true,
        /*optional, boolean, whether to enable fire alarm notification: "true"-yes,
        "false"-no*/
        "hostStatusEventEnabled":true,
        /*optional, boolean, whether to enable notification of security control panel
        status: "true"-yes, "false"-no*/
        "exDevStatusEventEnabled":true,
        /*optional, boolean, whether to enable peripheral status notification: "true"-
        yes, "false"-no*/
        "detectorStatusEventEnabled":true,
        /*optional, boolean, whether to enable detector status notification: "true"-
        yes, "false"-no*/
        "intelligentAlarmEnable":true,
        /*optional, boolean, whether to enable smart alarm notification: "true"-yes,
        "false"-no*/
        "numbersOfCalls":{
        /*optional, integer, phone call times*/
            "@min":0,
            "@max":0
        },
        "timeFilterEnabled": {
        /*optional, boolean, whether to enable uploading event details by call only
        within the configured period: true, false(uploading the calls all the time)*/
            "@opt": [true, false]
```

```
    },
    "startTime": "10:00:00",
    /*optional, time, start time, this node is valid when timeFilterEnabled is
    true*/
    "endTime": "16:00:00",
    /*optional, time, end time, this node is valid when timeFilterEnabled is true*/
    "WeekPlanCfg": {
    /*optional, object, week schedule information, this node is valid when
    timeFilterEnabled is true*/
        "@size": 7,
    /*optional, int, the maximum number of days*/
        "dayOfWeek": {
    /*required, object, day of the week*/
            "@min": 1,
    /*optional, int, the minimum value, range:[1,7]*/
            "@max": 7
    /*optional, int, the maximum value, range:[1,7]*/
        },
        "TimeRange": {
    /*required, object, period*/
            "@size": 8,
    /*optional, int, the maximum number of periods, range:[1,8]*/
            "startTime": "15:00:00",
    /*required, time, start time*/
            "endTime": "20:00:00"
    /*required, time, end time*/
        }
    }
}
}
```

A.3.142 JSON_PhoneCap

PhoneCap message in JSON format

```
{
  "PhoneCap":{
    "id":{
    /*required, range of phone number*/
      "@min":,
      "@max":
    },
    "alarmTamperEnabled": ,
    /*optional, boolean type, whether to enable alarm and tampering event
    notification, "true, false"*/
    "lifeSecurityEnabled": ,
    /*optional, boolean type, whether to enable life security event notification,
    "true, false"*/
    "systemStatusEnabled": ,
    /*optional, boolean type, whether to enable system status event notification,
```

```
"true, false"*/
    "operateEventEnabled":
/*optional, boolean type, whether to enable operation event notification,
"true, false"*/
    "messageEnabled": ,
/*optional, boolean type, whether to enable message notification, "true,
false"*/
    "callEnabled": ,
/*optional, boolean type, whether to enable phone call notification, "true,
false"*/
    "numLength":
/*required, phone number length*/
    }
}
```

A.3.143 JSON_PublicSubSys

PublicSubSys message in JSON format

```
{
  "PublicSubSys":{
    "id": ,
/*required, integer, partition No., it starts from 1*/
    "enabled": ,
/*required, boolean, whether to enable the public partition*/
    "linkageCommon":
/*optional, array, normal partitions linked to the public partition. This node
is required when enabled is "true"*/
    }
}
```

A.3.144 JSON_PublicSubSysCap

PublicSubSysCap capability message in JSON format

```
{
  "PublicSubSysCap":{
    "id":{
/*required, public partition No. that can be configured*/
      "@opt":[1]
    },
    "enabled":"true,false",
/*required, whether to enable the public partition*/
    "linkageCommon":{
/*optional, number of normal partitions linked to the public partition*/
      "@min": ,
      "@max":
    },
    "subSystemNo":{
```



```
/*optional, range of the normal partition No. that can be configured*/
    "@min": ,
    "@max":
  },
  "method":{
/*required, methods supported by the function: "put"-edit, "getAll"-get all*/
    "@opt":["put", "getAll"]
  }
}
}
```

A.3.145 JSON_RegisterMode

JSON message about the parameters of the registration mode

```
{
  "RegisterMode":{
    "mode":"","
/*optional, string, registration mode: "enter", "exit"*/
    "wirelessRecvAddress": ,
/*optional, int, wireless receiving module address, it is valid when mode is
"enter"*/
    "exDevType":""
/*optional, string, peripheral module type: "detector", "wirelessSiren"-
wireless siren, "wirelessRepeater"-wireless repeater, "wirelessOutput"-wireless
output module. This node is valid when mode is "enter"*/
  }
}
```

A.3.146 JSON_RegisterModeCap

JSON message about the configuration capability of the registration mode

```
{
  "RegisterModeCap":{
    "mode":{
/*optional, registration mode: "enter", "exit"*/
      "@opt":["enter", "exit"]
    },
    "detetorRecvAddress":{
/*optional, wireless receiving module address of the detector, it is valid when
mode is "enter"*/
      "@opt":[1,3,5]
    },
    "wirelessSirenRecvAddress":{
/*optional, wireless receiving module address of the wireless siren, it is
valid when mode is "enter"*/
      "@opt":[1,3,5]
    },
  },
}
```

```
"wirelessRepeaterRecvAddress":{
/*optional, wireless receiving module address of the wireless repeater, it is
valid when mode is "enter"*/
"@opt": [1,3,5]
},
"wirelessOutputRecvAddress":{
/*optional, wireless receiving module address of the wireless output module, it
is valid when mode is "enter"*/
"@opt": [1,3,5]
},
"exDevType":{
/*optional, peripheral module type: "detector", "wirelessSiren"-wireless siren,
"wirelessRepeater"-wireless repeater, "wirelessOutput"-wireless output module,
"wirelessKeypad", "wirelessRemoteCtrl". This node is valid when mode is
"enter"*/
"@opt": ["detetor", "wirelessSiren", "wirelessRepeater", "wirelessOutput",
"wirelessKeypad", "wirelessRemoteCtrl"]
},
"wirelessKeypadRecvAddress": {
/*optional, object, wireless receiving module address of the wireless keypad*/
"@opt": [1, 3, 5]
},
"wirelessRemoteCtrlRecvAddress": {
/*optional, object, wireless receiving module address of the wireless keyfob*/
"@opt": [1, 3, 5]
}
}
}
```

A.3.147 JSON_RemoteCfgUserName

RemoteCfgUserName message in JSON format

```
{
  "RemoteCfgUserName":{
    "userName":""
/*string, user name*/
  }
}
```

A.3.148 JSON_RemoteCtrl

JSON message about the parameters of the currently added keyfob in asynchronous mode

```
{
  "RemoteCtrl":{
    "status":"","
/*required, string, current status: "processing", "success", "failed"*/
    "failedReason":"","
```

```
/*optional, string, reason for failure: "repeatAdd" (the keyfob has been added
by this or other control panel), this node is valid when status is "failed"*/
    "id": ,
/*required, int, keyfob No., which starts from 1*/
    "enabled": ,
/*required, boolean, whether to enable keyfob*/
    "seq": "",
/*required, string, keyfob serial No.*/
    "name": "",
/*optional, string, keyfob name*/
    "factory": "",
/*optional, string, keyfob manufacturer*/
    "right": [""],
/*optional, array with string type, keyfob permission, if no value is assigned
to the node, the device adopts the default permissions*/
    "SelKeyList": [{
/*optional, key list, if no value is assigned to the node, the device adopts
the default one*/
        "SelKey": {
/*custom key, this node is required when the node SelKeyList exists*/
            "key": ,
/*integer type, key No., this node is required when the node SelKeyList exists*/
            "func": "",
/*integer type, keys' function, this node is required when the node SelKeyList
exists*/
            "outputNo":
/*integer type, relay No., this node is required when the fun is
"operateOutputs"*/
            }
        }],
    "CombKeyList": [{
/*optional, combined key list, if no value is assigned to the node, the device
adopts the default keys*/
        "CombKey": {
/*combined keys, this node is required only when the CombKeyList exists*/
            "keys": "",
/*string, combined keys, this node is required only when the CombKey exists*/
            "func": "",
/*string, combined keys' function, this node is required only when the CombKey
exists and its value can be empty*/
            "outputNo":
            }
        }],
    "subSystem": ,
/*optional, array, linked partitions. For example, [1,2,3] indicates linking to
partition 1, partition 2, and partition 3*/
    "relatedNetUserName": "",
/*optional, string, linked network user name*/
    "alarmVoicePromptEnabled":
/*optional, boolean, whether to enable voice prompt for panic alarm: true=yes
(the panic alarm of the keyfob will have linked voice prompt), false=no (the
panic alarm of the keyfob will not have linked voice prompt)*/
```

```
}  
}
```

A.3.149 JSON_RemoteCtrlCap

RemoteCtrlCap capability message in JSON format

```
{  
  "RemoteCtrlCap":{  
    "id":{  
/*required, keyfob No.*/  
      "@min": ,  
      "@max":  
    },  
    "enabled":{  
/*required, whether to enable the keyfob*/  
      "@opt":"true,false"  
    },  
    "seq":{  
/*required, length of the keyfob serial No.*/  
      "@min": ,  
      "@max":  
    },  
    "name":{  
/*optional, keyfob name length*/  
      "@min": ,  
      "@max":  
    },  
    "factory":{  
/*optional, keyfob manufacturer: "pyronix"-Pyronix, "hik"-Hikvision*/  
      "@opt":"pyronix,hik"  
    },  
    "right":{  
/*optional, permissions supported by the keyfob: "awayArming"-away arming,  
"stayArming"-stay arming, "disarming"-disarm, "panic"-panic alarm, "clearAlarm"-  
clear alarms, "operateOutputs"-operate relays*/  
      "@opt":"awayArming,stayArming,disarming,panic,operateOutputs"  
    },  
    "selKey":{  
/*optional, custom keys*/  
      "@opt":"3,4,..."  
    },  
    "combKey":{  
/*optional, combined keys*/  
      "@opt":"12,..."  
    },  
    "func":{  
/*optional, functions supported by the keyfob's key: "awayArming"-away arming,  
"stayArming"-stay arming, "disarming"-disarm, "panic"-panic alarm, "clearAlarm"-  
clear alarms, "operateOutputs"-operate relays, "queryFault"-search for faults  
(it is used to report existing faults of the security control panel),
```

```
"queryArmState"-search for arming status (it is used to prompt the arming and
disarming status of the security control panel)*/

"@opt":"awayArming,stayArming,disarming,panic,clearAlarm,operateOutputs,queryFau
lt,queryArmState,mutePA,muteMedicalTreatment,medicalTreatment"
},
"combKeyFunc":{
/*optional, string, combination key function. If this field is returned, this
field indicates the capability set of the combination key and the field func
indicates that of a single key; if this field is not returned, the field func
indicates the capability set of both a single key and the combination key*/

"@opt":"awayArming,stayArming,disarming,panic,clearAlarm,operateOutputs,queryFau
lt,queryArmState,mutePA,muteMedicalTreatment,medicalTreatment"
},
"outputNo":{
/*optional, output channel No., this node is required only when the fun
contains "operateOutputs"*/
"@min": ,
"@max":
},
"subSystem":{
/*optional, supported number of partitions that can be linked*/
"@min": ,
"@max":
},
"subSystemNo":{
/*optional, range of partition No.*/
"@min": ,
"@max":
},
"method":{
/*required, methods supported by the function: "currentAdd"-add, "del"-delete,
"put"-edit, "get"-get one, "getAll"-get all, "currentAddAsyn"-asynchronously
add*/
"@opt":"currentAdd,del,put,get,getAll,currentAddAsyn"
},
"relatedNetUserName":{
/*optional, int, length of the linked network user name*/
"@min": ,
"@max":
},
"alarmVoicePromptEnabled":{
/*optional, boolean, whether to enable voice prompt for panic alarm: true=yes
(the panic alarm of the keyfob will have linked voice prompt), false-no (the
panic alarm of the keyfob will not have linked voice prompt)*/
"@opt":[true,false]
}
}
}
```

Remarks

- There are four keys on the keyfob: **Lock**, **Unlock**, **I**, and **II**, which corresponds to key No.1, 2, 3, and 4.
- The combined key is represented by the strings that consists of key No., e.g., if **keys** is "12", it indicates that the key is combined by key No.1 (**Lock**) and key No.2 (**Unlock**).
- The **Lock** and **Unlock** keys are predefined, so only the **I** and **II** can be customized.
- Customizing the key is not supported by the Hikvision keyfob.

A.3.150 JSON_RemoteCtrlMode

RemoteCtrlMode message in JSON format

```
{
  "RemoteCtrlMode":{
    "mode": "",
    /*optional, string type, mode: "enter", "exit"*/
    "wirelessRecvAddress": ,
    /*optional, integer type, wireless receiving module address, this node is valid
    when mode is "enter"*/
    "keypadAddress":
    /*optional, integer type, keypad address, this node is valid when mode is
    "enter"*/
  }
}
```

Remarks

The keyfob can be added by the wireless receiving module or by the keypad, so either **wirelessRecvAddress** or **keypadAddress** should be configured.

A.3.151 JSON_RemoteCtrlModeCap

RemoteCtrlModeCap capability message in JSON format

```
{
  "RemoteCtrlModeCap":{
    "mode":{
    /*optional, mode: "enter", "exit"*/
      "@opt":["enter","exit"]
    },
    "wirelessRecvAddress":{
    /*optional, wireless receiving module address, this node is valid when mode is
    "enter"*/
      "@opt":[1,3,5...]
    },
    "keypadAddress":{
    /*optional, keypad address, this node is valid when mode is "enter"*/
    }
  }
}
```

```
    "@opt": [1, 3, 5...]
  }
}
```

A.3.152 JSON_Repeater

Repeater message in JSON format

```
{
  "Repeater": {
    "id": ,
    /*required, integer type, repeater No., which starts from 1*/
    "name": "",
    /*optional, string type, repeater name*/
    "related": ,
    /*boolean type, whether the repeater is linked, for PUT method, this node is
    optional; for GET method, this node is required*/
    "seq": "",
    /*string type, repeater serial No., this node is required when related is
    "true"*/
    "linkageAddress": ,
    /*optional, integer type, linked module address, this node is only returned by
    wireless modules. If related is "true", this node cannot be configured*/
    "checkTime":
    /*optional, integer type, offline duration, unit: hour*/
  }
}
```

A.3.153 JSON_RepeaterCap

RepeaterCap capability message in JSON format

```
{
  "RepeaterCap": {
    "id": {
    /*required, repeater No. range*/
      "@min": ,
      "@max":
    },
    "name": {
    /*optional, range of repeater name length*/
      "@min": ,
      "@max":
    },
    "related": {
    /*required, whether to link the physical repeater*/
      "@opt": "true,false"
    },
  },
}
```

```
"seq":{
/*required, range of repeater serial No. length*/
  "@min":,
  "@max":
},
"linkageAddress":{
/*optional, linked module address, this node is only returned by wireless
modules*/
  "@opt":[1,2,3]
},
"checkTime":{
/*optional, offline duration, unit: hour*/
  "@min": ,
  "@max":
},
"method":{
/*required, methods supported by the function: "add", "put"-edit, "getAll"-get
all*/
  "@opt":"add,put,getAll"
}
}
```

A.3.154 JSON_RepeaterList

JSON message about repeater status

```
{
  "RepeaterList":[{
/*optional, repeater list*/
    "Repeater":{
      "id": ,
/*required, integer type, repeater No.*/
      "seq": "",
/*required, string type, peripheral serial No.*/
      "name": "",
/*optional, string type, repeater name*/
      "status": "",
/*optional, string type, repeater status: "notRelated"-not linked, "online",
"offline", "heartbeatAbnormal"-heartbeat exception*/
      "tamperEvident": ,
/*optional, boolean type, tampering status: "true"-tampered, "false"-not
tampered*/
      "charge": "",
/*optional, string type, state of charge: "normal", "lowPower"-low battery*/
      "signal":
/*optional, integer type, signal strength, it is between 0 and 255*/
      "model": "DS-PR1-WE",
/*optional, string, model: "DS-PR1-WE" (wireless repeater)*/
      "temperature": 0,
/*optional, int, temperature*/
```



```
    "connDevNum": 1,
/*optional, int, number of devices*/
    "mainPowerSupply": true,
/*optional, boolean, AC power supply status: true (connected), false
(disconnected)*/
    "batteryStatus": "normal",
/*optional, string, battery status, "normal", "miss"*/
    "version": "test",
/*optional, string, version No.*/
    "deviceNo": 1
/*optional, int, device No., range:[1,1000]*/
  }
}]
}
```

A.3.155 JSON_ResponseStatus

JSON message about response status

```
{
  "requestURL": "",
/*optional, string, request URL*/
  "statusCode": ,
/*optional, int, status code*/
  "statusString": "",
/*optional, string, status description*/
  "subStatusCode": "",
/*optional, string, sub status code*/
  "errorCode": ,
/*required, int, error code, which corresponds to subStatusCode, this field is
required when statusCode is not 1. The returned value is the transformed
decimal number*/
  "errorMsg": "",
/*required, string, error details, this field is required when statusCode is
not 1*/
  "MErrCode": "0xFFFFFFFF",
/*optional, string, error code categorized by functional modules*/
  "MErrDevSelfEx": "0xFFFFFFFF"
/*optional, string, extension of MErrCode. It is used to define the custom
error code, which is categorized by functional modules*/
}
```

A.3.156 JSON_Result

JSON message about result of starting signal strength detection in asynchronous mode

```
{
  "Result": {
    "detectionTime": 12
  }
}
```

```
/*optional, int, duration required for detection, unit: second*/
}
}
```

A.3.157 JSON_SearchDescription

SearchDescription message in JSON format

```
{
  "SearchDescription":{
    "searchID":"","
    /*required, string type, search ID, it is recommended to be same with the GUID
    OR UUID of ISO/IEC 9834-8/ITU X.667, e.g.,
    "812F04E0-4089-11A3-9A0C-0305E82C2906". And it is repeatedly returned with
    search results during the search*/
    "timeSpanList":[{
    /*start time and end time of search*/
      "timeSpan":{
        "startTime":"","
        /*start time, e,g, "2013-06-10T12:00:00+08:00"*/
        "endTime":"","
        /*end time, e,g, "2013-06-10T12:00:00+08:00"*/
      }
    }],
    "metaID":"","
    /*search range, the format of metaID is <domain>/<class>/<type>, e.g.,
    log.xxx.com/Alarm/motionstart, which indicates that the logs with minor type of
    "motionstart" and major type of "alarm" will be searched, if only the <domain>
    exists, it indicates search all*/
    "searchResultPostion": ,
    /*number of returned search results, when the number of search results is
    larger than the value of maxResults, this node is used to mark the received
    results position for next search*/
    "maxResults":
    /*maximum search results can be returned*/
  }
}
```

A.3.158 JSON_SearchResult

SearchResult message in JSON format

```
{
  "SearchResult":{
    "searchID":"","
    /*required, string type, search ID, it is recommended to be same with the GUID
    OR UUID of ISO/IEC 9834-8/ITU X.667, e.g.,
    "812F04E0-4089-11A3-9A0C-0305E82C2906". And it is repeatedly returned with
    search results during the search*/
```

```
    "responseStatus": ,
    /*required, response status, return "true" for success, and return "false" for
    failure*/
    "responseStatusStrg": "",
    /*required, response status which can be directly read, and it corresponds to
    responseStatus*/
    "numOfMatches": ,
    /*optional, number of matched results*/
    "matchList": [{
        "searchMatchItem": {
            "logDescriptor": {
                "metaId": "",
                /*search range, the format of metaID is <domain>/<class>/<type>, e.g.,
                log.xxx.com/Infomation/runStatusInfo, which indicates that the logs with minor
                type of "runStatusInfo" and major type of "Infomation" will be searched, if
                only the <domain> exists, it indicates search all.*/
                "startDateTime": "",
                /*required, start time of search, e.g., "2013-05-18T10:31:26+08:00"*/
                "userName": "",
                /*optional, local or remote user name*/
                "ipAddress": "",
                /*optional, IP address of remote security control panel*/
                "object": "",
                /*required, operating medium: "network", "keypad", "remoteCtrl"-keyfob, "card"*/
                "params": "",
                /*optional, zone No. or other parameters*/
                "seq": "",
                /*optional, serial No.*/
                "attachInfo": ""
            }
            /*optional, additional information/
            }
        }
    }
    }
    }
    }
```

Remarks

For details about log major and minor types, refer to .

A.3.159 JSON_SecurityCPCap

JSON message about the capability of security control panel

```
{
    "SecurityCPCap": {
        "partitionNum": ,
        /*optional, int, the number of partitions can be set, the default value is 1*/
        "localZoneNum": ,
        /*required, int, number of local zones*/
        "extendZoneNum": ,
```

```
/*optional, int, number of extended zones*/
    "wirelessZoneNum": ,
/*optional, int, number of wireless zones*/
    "localRelayNum": ,
/*optional, int, number of local relays*/
    "extendRelayNum": ,
/*optional, int, number of extended relays*/
    "wirelessRelayNum": ,
/*optional, int, number of wireless relays*/
    "repeater": ,
/*optional, int, number of repeaters*/
    "sirenNum": ,
/*optional, int, umber of sirens*/
    "userNum": ,
/*required, int, number of users*/
    "ARCNum": ,
/*optional, int, number of alarm receiving centers*/
    "phoneNum": ,
/*optional, int, number of phone number*/
    "outputModNum": ,
/*optional, int, number of output modules*/
    "cardNum": ,
/*optional, int, number of cards*/
    "keypadNum": ,
/*optional, int, number of keypads*/
    "cardReaderNum": ,
/*optional, int, number of card readers*/
    "electricLockNum": ,
/*optional, int, number of electric locks*/
    "alarmLampNum": ,
/*optional, int, number of alarm lamps*/
    "isSptLogSearch": ,
/*boolean, whether to support log search*/
    "isSptConfiguration": ,
/*optional, boolean, whether to support configuring security control panel*/
    "isSptControl": ,
/*optional, boolean, whether to support security control panel control*/
    "isSptStatus": ,
/*optional, boolean, whether to support monitoring security control panel
status*/
    "isSptPaceTest": ,
/*optional, boolean, whether to support pacing*/
    "isSptStandardCfg": ,
/*optional, boolean, whether to support standard configuration for security
control panel*/
    "isSptOneKeyAlarm": ,
/*optional, boolean, whether it supports one-push alarm, related URI: /ISAPI/
SecurityCP/control/oneKeyAlarm. For compatibility, the node
isSptOneKeyAlarmCtrl will also be returned in the capability message
JSON_HostControlCap after calling the URI /ISAPI/SecurityCP/control/
capabilities?format=json by GET method, and the device behavior will be the
same*/
```

```

    "isSptSysCheckManually": ,
    /*optional, boolean, whether it supports enabling system detection manually,
    related URI: /ISAPI/SecurityCP/syscheckManually*/
    "isSptSysCheckResult": ,
    /*optional, boolean, whether it supports getting the system detection result,
    related URI: /ISAPI/SecurityCP/checkResult*/
    "isSptSysAutoCheckTimeCfg": ,
    /*optional, boolean, whether it supports audio and video system detection,
    related URI: /ISAPI/SecurityCP/sysAutoCheckTimeCfg*/
    "isSptVideoFileUpload": ,
    /*optional, boolean, whether it supports uploading the audio file, related
    URI: /ISAPI/SecurityCP/videoFileUpload?format=json*/
    "isSptPircamCapture": ,
    /*optional, boolean, whether it supports pircam (detector equipped with camera)
    capture, related URI: /ISAPI/SecurityCP/pircam/channels/<ID>/picture?
    format=json*/
    "isSptCustomizeCfg": ,
    /*optional, boolean, whether it supports configuring the custom audio file,
    related URI: /ISAPI/SecurityCP/videoBroadcast/customizeCfg*/
    "isSptCustomizeUpload": ,
    /*optional, boolean, whether it supports uploading the custom file, related
    URI: /ISAPI/SecurityCP/videoBroadcast/customizeUpload?format=json*/
    "isSptPircamFileExport": ,
    /*optional, boolean, whether it supports exporting pictures captured by pircam
    (detector equipped with camera), related URI: ISAPI/SecurityCP/FileExport/
    pircam?format=json*/
    "transmitterNum": 1,
    /*optional, int, the number of transmitters*/
    "transmitterModel": {
    /*optional, object, supported model of transmitters*/
        "@opt": ["0x71001"]
    },
    "localAccessModuleType": {
    /*optional, object, local access module type*/
        "@opt": ["localTransmitter", "localZone", "localRelay", "localSiren"]
    /*optional, array of string: "localTransmitter" (local transmitter),
    "localZone" (local zone), "localRelay" (local relay), "localSiren" (local
    siren)*/
    }
    }
}

```

A.3.160 JSON_SendARC

SendARC message in JSON format

```

{
    "SendARC":{
        "alarmTamperEnabled": ,
        /*optional, boolean type, whether to enable alarm and tampering event
        notification: "true"-yes, "false"-no*/
    }
}

```

```
"lifeSecurityEnabled": ,
/*optional, boolean type, whether to enable life security event notification:
"true"-yes, "false"-no*/
"systemStatusEnabled": ,
/*optional, boolean type, whether to enable system status event notification:
"true"-yes, "false"-no*/
"operateEventEnabled": ,
/*optional, boolean type, whether to enable operation event notification:
"true"-yes, "false"-no*/
"zoneAlarmTamperEnabled": ,
/*optional, boolean, whether to enable alarm and tampering event notification
of the supported zone: "true"-yes, "false"-no*/
"exDevTamperEventEnabled": ,
/*optional, boolean, whether to enable peripheral tampering alarm notification:
"true"-yes, "false"-no*/
"hostTamperEventEnabled": ,
/*optional, boolean, whether to enable tampering alarm notification of security
control panel: "true"-yes, "false"-no*/
"emergencyEventEnabled": ,
/*optional, boolean, whether to enable panic alarm notification: "true"-yes,
"false"-no*/
"medicalEventEnabled": ,
/*optional, boolean, whether to enable medical alarm notification: "true"-yes,
"false"-no*/
"gasEventEnabled": ,
/*optional, boolean, whether to enable gas alarm notification: "true"-yes,
"false"-no*/
"fireEventEnabled": ,
/*optional, boolean, whether to enable fire alarm notification: "true"-yes,
"false"-no*/
"hostStatusEventEnabled": ,
/*optional, boolean, whether to enable notification of security control panel
status: "true"-yes, "false"-no*/
"exDevStatusEventEnabled": ,
/*optional, boolean, whether to enable peripheral status notification: "true"-
yes, "false"-no*/
"detectorStatusEventEnabled": ,
/*optional, boolean, whether to enable detector status notification: "true"-
yes, "false"-no*/
"intelligentAlarmEnable":
/*optional, boolean, whether to enable smart alarm notification: "true"-yes,
"false"-no*/
}
}
```

A.3.161 JSON_SendARCCap

SendARCCap capability message in JSON format

```
{
  "SendARCCap":{
```

```
"alarmTamperEnabled":true,
/*optional, boolean type, whether to enable alarm and tampering event
notification: "true"-yes, "false"-no*/
"lifeSecurityEnabled":true,
/*optional, boolean type, whether to enable life security event notification:
"true"-yes, "false"-no*/
"systemStatusEnabled":true,
/*optional, boolean type, whether to enable system status event notification:
"true"-yes, "false"-no*/
"operateEventEnabled":true,
/*optional, boolean type, whether to enable operation event notification:
"true"-yes, "false"-no*/
"isSupportSendARCList":true,
/*optional, boolean type, whether to support getting SendARCList message in
JSON format: "true, false"*/
"zoneAlarmTamperEnabled":true,
/*optional, boolean, whether to enable alarm and tampering event notification
of the supported zone: "true"-yes, "false"-no*/
"exDevTamperEventEnabled":true,
/*optional, boolean, whether to enable peripheral tampering alarm notification:
"true"-yes, "false"-no*/
"hostTamperEventEnabled":true,
/*optional, boolean, whether to enable tampering alarm notification of security
control panel: "true"-yes, "false"-no*/
"emergencyEventEnabled":true,
/*optional, boolean, whether to enable panic alarm notification: "true"-yes,
"false"-no*/
"medicalEventEnabled":true,
/*optional, boolean, whether to enable medical alarm notification: "true"-yes,
"false"-no*/
"gasEventEnabled":true,
/*optional, boolean, whether to enable gas alarm notification: "true"-yes,
"false"-no*/
"fireEventEnabled":true,
/*optional, boolean, whether to enable fire alarm notification: "true"-yes,
"false"-no*/
"hostStatusEventEnabled":true,
/*optional, boolean, whether to enable notification of security control panel
status: "true"-yes, "false"-no*/
"exDevStatusEventEnabled":true,
/*optional, boolean, whether to enable peripheral status notification: "true"-
yes, "false"-no*/
"detectorStatusEventEnabled":true,
/*optional, boolean, whether to enable detector status notification: "true"-
yes, "false"-no*/
"intelligentAlarmEnable":true
/*optional, boolean, whether to enable smart alarm notification: "true"-yes,
"false"-no*/
}
}
```

See Also

[JSON_SendARCList](#)

A.3.162 JSON_SendARCList

SendARCList message in JSON format

```
{
  "SendARCList": [{
    "SendARC": {
      "id": ,
      /*required, integer, alarm receiving center No.*/
      "alarmTamperEnabled": ,
      /*optional, boolean, whether to enable alarm and tampering event notification:
      "true"-yes, "false"-no*/
      "lifeSecurityEnabled": ,
      /*optional, boolean, whether to enable life security event notification: "true"-
      yes, "false"-no*/
      "systemStatusEnabled": ,
      /*optional, boolean, whether to enable system status event notification: "true"-
      yes, "false"-no*/
      "operateEventEnabled": ,
      /*optional, boolean, whether to enable operation event notification: "true"-
      yes, "false"-no*/
      "zoneAlarmTamperEnabled": ,
      /*optional, boolean, whether to enable alarm and tampering event notification
      of the supported zone: "true"-yes, "false"-no*/
      "exDevTamperEventEnabled": ,
      /*optional, boolean, whether to enable peripheral tampering alarm notification:
      "true"-yes, "false"-no*/
      "hostTamperEventEnabled": ,
      /*optional, boolean, whether to enable tampering alarm notification of security
      control panel: "true"-yes, "false"-no*/
      "emergencyEventEnabled": ,
      /*optional, boolean, whether to enable panic alarm notification: "true"-yes,
      "false"-no*/
      "medicalEventEnabled": ,
      /*optional, boolean, whether to enable medical alarm notification: "true"-yes,
      "false"-no*/
      "gasEventEnabled": ,
      /*optional, boolean, whether to enable gas alarm notification: "true"-yes,
      "false"-no*/
      "fireEventEnabled": ,
      /*optional, boolean, whether to enable fire alarm notification: "true"-yes,
      "false"-no*/
      "hostStatusEventEnabled": ,
      /*optional, boolean, whether to enable notification of security control panel
      status: "true"-yes, "false"-no*/
      "exDevStatusEventEnabled": ,
```



```
/*optional, boolean, whether to enable peripheral status notification: "true"-yes, "false"-no*/
    "detectorStatusEventEnabled": ,
/*optional, boolean, whether to enable detector status notification: "true"-yes, "false"-no*/
    "intelligentAlarmEnable":
/*optional, boolean, whether to enable smart alarm notification: "true"-yes, "false"-no*/
    }
  ]]
}
```

A.3.163 JSON_SignalStrengthDetection

JSON message about result parameters of signal strength detection in asynchronous mode

```
{
  "SignalStrengthDetection":{
    "status": "",
/*required, string, current stats: "processing", "success", "failed"*/
    "signal": ,
/*optional, int, signal strength*/
    "exDevType": "",
/*optional, string, peripheral type*/
    "id": 1
/*optional, int, ID*/
  }
}
```

A.3.164 JSON_SignalStrengthDetectionCap

JSON message about the configuration capability of signal strength detection in asynchronous mode

```
{
  "SignalStrengthDetectionCap":{
    "signal":{
/*optional, int, signal strength*/
      "@min": 1,
      "@max": 2
    },
    "mode":{
/*optional, string, operation: "enter", "exit"*/
      "@opt": ["enter", "exit"]
    },
    "detectionTime":{
/*optional, int, detection time*/
      "@min": 1,
      "@max": 2
    }
  }
}
```

```
    },
    "List": [{
        "supportedType": {
            /*required, string, module type that supports signal strength detection:
            "cardReader"-card reader, "keyPad"-keypad, "siren"-siren, "zone"-zone (if this
            node is "zone", it indicates detecting the signal strength of the detector
            linked with the zone. If the detector does not support signal strength
            detection, the related information will be returned in the configuration
            capability message of the corresponding detector)*/
            "@opt":[""]
        },
        "id": {
            /*required, int, ID*/
            "@min":1,
            "@max":2
        }
    }]
}
```

A.3.165 JSON_SignalStrengthDetectionMode

JSON message about the mode parameters of signal strength detection in asynchronous mode

```
{
  "SignalStrengthDetectionMode": {
    "mode": "",
    /*required, string, mode: "enter", "exit"*/
    "exDevType": "",
    /*required, string, peripheral type*/
    "id": 1
    /*required, int, ID*/
  }
}
```

A.3.166 JSON_Siren

JSON message about parameters of a siren

```
{
  "Siren": {
    "status": ,
    /*required, string, current status: "processing", "success", "failed"*/
    "id": ,
    /*required, int, siren No., which starts from 1*/
    "name": "",
    /*optional, string, siren name*/
    "volume": ,
    /*optional, int, siren volume*/
  }
}
```

```
"related": ,
/*boolean type, whether the physical siren is linked, for PUT method, this node
is optional; but for GET method, this node is required*/
"seq": "",
/*string, siren serial No., this node is required only when related is "true"*/
"address": ,
/*optional, read-only, int, module address, this node is only returned by wired
modules*/
"linkageAddress": ,
/*optional, read-only, int, linked module address, this node is only returned
by wireless modules. If related is "true", this node cannot be configured*/
"checkTime": ,
/*optional, int, offline duration, unit: hour*/
"sirenAttrib": "",
/*optional, read-only, string, siren attribute: "wired", "wireless"*/
"linkage": "",
/*optional, string, event linkage type*/
"zoneEvent": "",
/*optional, string, zone event type, this node is valid when linkage contains
"zone". [1,3] indicates that zone 1 and zone 3 are enabled. If this node is not
configured when linkage contains "zone", it indicates enabling all zones*/
"subSystem": ,
/*optional, array, partitions linked to the siren. For example, [1,2,3]
indicates that partition 1, partition 2, and partition 3 are linked to the
siren*/
"LinkageList": [{
/*optional, linked event list. If the device supports linking multiple event
types with multiple minor event types and multiple partitions, this node can be
configured. For compatibility, linking an event type with multiple minor event
types and multiple partitions should also be supported*/
"linkage": "",
/*optional, string, linked event type*/
"zoneEvent": [1,3],
/*optional, array, zone event type. This node is valid when linkage contains
"zone". For example, [1,3] indicates enabling zone 1 and zone 3. If this node
is not configured, it indicates enabling all zones*/
"subSystem": [1, 2, 3]
/*optional, array, linked partitions when linkage is "alarm", "arming", or
"disarming". For example, [1,2,3] indicates linking with partition 1, partition
2, and partition 3 are linked*/
}],
"LEDEnabled": ,
/*optional, boolean, whether to enable the LED indicator*/
"LEDLatchTime": ,
/*optional, int, delay time of the LED indicator, unit: second*/
"findMeEnabled": ,
/*optional, boolean, whether to enable the Find Me function*/
"location": "",
/*optional, string, siren location: "outdoor", "indoor"*/
"ArmAndDisarmIndicatorCfg": {
/*optional, indicator settings for arming and disarming*/
"LEDEnabled": ,
```

```
/*optional, boolean, whether to enable the LED indicator to flicker for arming
and disarming*/
    "LEDTimes": ,
/*optional, int, LED indicator flickering times*/
    "LEDFrequency": ,
/*optional, int, LED indicator flickering frequency, unit: Hz*/
    "buzzerEnabled": ,
/*optional, boolean, whether to enable the buzzer to buzz for arming and
disarming*/
    "buzzerTimes":
/*optional, int, buzzer buzzing times*/
    },
    "company": "",
/*optional, string, read-only, company name: "pyronix", "longhorn",
"hikvision". When the value of company is "hikvision", the siren supports test*/
    "tamperEnabled": ,
/*optional, boolean, whether to enable siren tampering*/
    "tryAlarmEnabled": ,
/*optional, boolean, whether to enable alarm attempt*/
    "preRegisterEnabled": ,
/*optional, boolean, whether to enable pre-registration*/
    "buzzEnabled": ,
/*optional, boolean, whether to enable linking the buzzer to buzz when the
alarm is triggered*/
    "disarmTamperEnabled": true,
/*optional, boolean, whether to enable tampering alarm when it is disarmed,
this node is valid when buzzEnabled is true*/
    "alarmStrobeFlashEnabled": ,
/*optional, boolean, whether to enable linking the alarm lamp to flicker when
the alarm is triggered*/
    "sounderAlarmDuration": ,
/*optional, int, siren's output duration when the alarm is triggered, unit:
second*/
    "heartBeatInterval":
/*optional, int, heartbeat interval of the security control panel and the
peripheral, unit: second*/
    "sirenColor": "red",
/*optional, string, siren color*/
    "alarmLinkedEventCfg": ["alarmTrigger"]
/*optional, array of string, linked event configuration*/
    }
}
```

A.3.167 JSON_SirenCap

JSON message about siren configuration capability

```
{
  "SirenCap":{
    "id":{
/*required, siren No. range*/
```

```
        "@min": ,
        "@max":
    },
    "name":{
/*optional, range of siren name length*/
        "@min": ,
        "@max":
    },
    "volume":{
/*optional, siren volume range: 0 (muted), 1 (low), 2 (medium), 3 (high). The
values in the capability message returned by the device are the supported
volume values*/
        "@min": ,
        "@max":
    },
    "supportVolumeIDList": ,
/*optional, ID list of sirens that support volume configuration. If this node
is not returned and volume is returned, it indicates that all sirens support
volume configuration; if this node is not returned and volume is not returned,
it indicates that all sirens do not support volume configuration; if this node
is returned, it indicates that sirens with the returned ID support volume
configuration (e.g., [1,3] indicates that siren 1 and siren 3 support volume
configuration)*/
    "related":{
/*required, whether the siren is linked*/
        "@opt":"true,false"
    },
    "seq":{
/*required, range of siren serial No. length*/
        "@min": ,
        "@max":
    },
    "address":{
/*optional, read-only, module address, this node is only returned by wired
modules*/
        "@opt":[1,2,3]
    },
    "linkageAddress":{
/*optional, read-only, linked module address, this node is only returned by
wireless modules*/
        "@opt":[1,2,3]
    },
    "checkTime":{
/*optional, offline duration, unit: hour*/
        "@min": ,
        "@max":
    },
    "sirenAttrib":{
/*optional, read-only, siren attribute: "wired", "wireless"*/
        "@opt":["wired","wireless"]
    },
    "sirenLinkageType":{
```

```
/*optional, type of siren that supports being linked: "wired"*/
    "@opt":["wired"]
},
"linkage":{
/*optional, event type linked to the siren: "alarm", "arming", "disarming",
"manualCtrl"-manually control, "zone"*/
    "@opt":["alarm","arming","disarming","manualCtrl","zone"]
},
"LinkageList":{
/*optional, linked event list. If the device supports linking multiple event
types with multiple minor event types and multiple partitions, this node can be
configured. For compatibility, linking an event type with multiple minor event
types and multiple partitions should also be supported*/
    "@size":2,
/*int, maximum number of event types that can be linked*/
    "linkage":""{
/*optional, string, event types that can be linked to the siren*/
        "@opt":"alarm,arming,disarming,manualCtrl,zone"
    },
    "zoneEvent ":{
        "@min":1,
        "@max":2,
/*optional, int, maximum No. of the zone*/
        "@size":2,
/*optional, int, maximum number of zones that can be linked*/
    }
    "subSystem":{
        "@min":1,
        "@max":2,
/*optional, int, maximum No. of the partition*/
        "@size":2,
/*optional, int, maximum number of partitions that can be linked*/
    }
},
"zoneEvent":{
/*optional, string, zone event type, this node is valid when linkage contains
"zone". For example, [1,3] indicates that zone 1 and zone 3 are enabled. If
this node is not configured when linkage contains "zone", it indicates enabling
all zones*/
    "@opt":["0,1,2,3"]
},
"method":{
/*required, methods supported by the function: "add", "put"-edit, "getAll"-get
all*/
    "@opt":"add,put,getAll"
},
"subSystem":{
/*optional, number of partitions that can be linked*/
    "@min": ,
    "@max":
},
"subSystemNo":{
```

```
/*optional, range of partition No.*/
    "@min": ,
    "@max":
},
    "LEDEnabled": {
/*optional, boolean, whether to enable the LED indicator*/
    "@opt":[true,false]
},
    "LEDLatchTime": {
/*optional, int, delay time of the LED indicator, unit: second*/
    "@min":0,
    "@max":0
},
    "findMeEnabled": {
/*optional, boolean, whether to enable the Find Me function*/
    "@opt":[true,false]
},
    "location":{
/*optional, string, siren location: "outdoor", "indoor"*/
    "@opt":["indoor","outdoor"]
},
    "ArmAndDisarmIndicatorCfg":{
/*optional, indicator settings for arming and disarming*/
    "LEDEnabled": {
/*optional, boolean, whether to enable the LED indicator to flicker for arming
and disarming*/
        "@opt":[true,false]
    },
    "LEDTimes": {
/*optional, int, LED indicator flickering times*/
        "@min": ,
        "@max":
    },
    "LEDFrequency": {
/*optional, int, LED indicator flickering frequency, unit: Hz*/
        "@min": ,
        "@max":
    },
    "buzzerEnabled": {
/*optional, boolean, whether to enable the buzzer to buzz for arming and
disarming*/
        "@opt":[true,false]
    },
    "buzzerTimes": {
/*optional, int, buzzer buzzing times*/
        "@min": ,
        "@max":
    }
},
    "company":{
/*optional, string, read-only, company name: "pyronix", "longhorn",
"hikvision". When the value of company is "hikvision", the siren supports test*/
```

```
    "@opt":["pyronix","longhorn","hikvision"]
  },
  "supportSirenCtrlIDList": [1,3]
/*optional, ID list of the sirens that support test (e.g., if [1,3] is
returned, sirens with IDs 1 and 3 support test)*/
  "tamperEnabled": {
/*optional, boolean, whether to enable siren tampering*/
    "@opt":[true,false]
  },
  "tryAlarmEnabled": {
/*optional, boolean, whether to enable alarm attempt*/
    "@opt":[true,false]
  },
  "preRegisterEnabled":{
/*optional, boolean, whether to enable pre-registration*/
    "@opt":[true,false]
  },
  "buzzEnabled":{
/*optional, boolean, whether to enable linking the buzzer to buzz when the
alarm is triggered*/
    "@opt":[true,false]
  },
  "disarmTamperEnabled": {
/*optional, object, whether to enable tampering alarm when it is disarmed*/
    "@opt": [true, false]
  },
  "alarmStrobeFlashEnabled":{
/*optional, boolean, whether to enable linking the alarm lamp to flicker when
the alarm is triggered*/
    "@opt":[true,false]
  },
  "sounderAlarmDuration":{
/*optional, int, siren's output duration when the alarm is triggered, unit:
second*/
    "@min":0,
    "@max":1
  },
  "heartBeatInterval":{
/*optional, int, heartbeat interval of the security control panel and the
peripheral, unit: second*/
    "@opt":[5,10,20,30]
  },
  "isSupportSignalTest": ,
/*optional, boolean, whether it supports signal strength detection: true-
support, false or this node is not returned-no*/
  "isSupportZoneTest": ,
/*optional, boolean, whether it supports zone detection: true-support, false or
this node is not returned-no*/
  "supportSirenCtrlIDList":
/*optional, array, ID list of sirens that support siren test alarm. For
example, [1,3] indicates that siren 1 and siren 3 support siren test alarm*/
  "alarmLinkedEventCfg": {
```



```
/*optional, object, linked event configuration: "alarmTrigger" (current alarm
event), "alarmConfirm" (confirm alarm event)*/
    "@size": 2,
    "@opt": ["alarmTrigger", "alarmConfirm"]
/*optional, array of string*/
    },
    "accessModuleType": {
/*optional, string, access module type: "localSiren"*/
        "@opt": ["localSiren"]
    }
}
}
```

A.3.168 JSON_SirenList

JSON message about the siren status

```
{
  "SirenList":[{
/*optional, siren list*/
    "Siren":{
      "id": ,
/*required, integer type, siren No.*/
      "seq": "",
/*required, string type, peripheral serial No.*/
      "name": "",
/*optional, string type, siren name*/
      "status": "",
/*optional, string type, siren status: "notRelated"-not linked, "on", "off",
"offline", "heartbeatAbnormal"-heartbeat exception*/
      "tamperEvident": ,
/*optional, boolean type, tampering status: "true"-tampered, "false"-not
tampered*/
      "sirenAttrib": "",
/*string type, siren attribute: "wired", "wireless"*/
      "charge": "",
/*optional, string type, state of charge: "normal", "lowPower"-low battery*/
      "chargeValue": ,
/*optional, int, battery power value which is between 0 and 100*/
      "signal": ,
/*optional, integer type, signal strength, it is between 0 and 255*/
      "model": "",
/*optional, string, model*/
      "temperature": ,
/*optional, read-only, int, temperature*/
      "subSystemList": [1, 2, 3],
/*optional, int, list of the linked partitions*/
      "powerSupplyStatus": "battery",
/*optional, string, power supply status: "battery", "DC12V"*/
      "sirenColor": "test",
/*optional, string, siren color*/
```

```
    "isViaRepeater": true,
    /*optional, boolean, whether the signal is transmitted via the repeater*/
    "repeaterName": "test",
    /*optional, string, repeater name, this node is valid when isViaRepeater is
    true*/
    "version": "test",
    /*optional, string, version No.*/
    "accessModuleType": "localSiren",
    /*optional, string, access module type: "localSiren", this node should be used
    together with sirenAttribute (for displaying wireless or wired conditions)*/
    "address": 254,
    /*optional, int, (wired) access module address*/
    "deviceNo": 1,
    /*optional, int, device No., range:[1,1000]. After installation, the installer
    will upload the device No. and the corresponding peripheral/detector
    information to the ARC for device type recognition*/
    "mainPowerSupply": true
    /*optional, boolean, main power supply (external power supply is supported for
    wireless siren): true, false*/
  }
}]
}
```

A.3.169 JSON_SirenMode

Message about the parameters for controlling the asynchronous mode of adding the siren in JSON format

```
{
  "SirenMode":{
    "mode": "",
    /*optional, string, mode: "enter", "exit"*/
    "wirelessRecvAddress": ,
    /*optional, int, wireless receiving module address, this field is valid when
mode is "enter"*/
    "keypadAddress": ,
    /*optional, int, keypad address, this field is valid when mode is "enter"*/
    "sequence": "",
    /*required, string, peripheral's serial No.*/
    "deviceType": ""
    /*optional, string, peripheral model*/
  }
}
```

Remarks

Sirens can be added by the wireless receiving module or the keypad, so either **wirelessRecvAddress** or **keypadAddress** should be configured.

A.3.170 JSON_SirenModeCap

Message about the capability of controlling asynchronous mode of adding the siren in JSON format

```
{
  "SirenModeCap": {
    "mode": {
      /*optional, string, mode: "enter", "exit"*/
      "@opt": ["enter", "exit"]
    },
    "wirelessRecvAddress": {
      /*optional, int, wireless receiving module address, this field is valid when
mode is "enter"*/
      "@opt": [1, 3, 5...]
    },
    "keypadAddress": {
      /*optional, int, keypad address, this field is valid when mode is "enter"*/
      "@opt": [1, 3, 5...]
    },
    "sequence": {
      /*optional, int, length of the peripheral's serial No.*/
      "@min": 0,
      "@max": 1
    },
    "deviceType": {
      /*optional, string, peripheral model, peripheral models supported by the device
      will be returned*/
      "@opt": ["DS-PK1-E-WE", "DS-PT1-WE"]
    }
  }
}
```

A.3.171 JSON_SlimMagneticContact

Message about the parameters of the slim magnetic contact detector of a specific zone in JSON format.

```
{
  "SlimMagneticContact": {
    "LEDEnabled": ,
    /*optional, boolean, whether to enable the LED indicator*/
    "LEDLatchTime": ,
    /*optional, int, delay time of the LED indicator, unit: second*/
    "findMeEnabled": ,
    /*optional, boolean, whether to enable the Find Me function*/
    "heartBeatInterval":
    /*optional, boolean, heartbeat interval of the security control panel and the
    peripheral, unit: second*/
  }
}
```

```
}  
}
```

A.3.172 JSON_SlimMagneticContactCap

JSON message about the configuration capability of the slim magnetic contact detector

```
{  
  "SlimMagneticContactCap": {  
    "zoneNo": {  
      /*optional, int, values that can be configured as the zone No.*/  
      "@opt": [1,3,5]  
    },  
    "supportZoneType": {  
      /*optional, string, zone types supported by the peripheral: "Instant"-instant  
      zone, "Delay"-delay zone, "Follow"-follow zone, "Perimeter"-perimeter zone,  
      "24hNoSound"-24-hour silent zone, "Emergency"-panic zone, "Fire"-fire zone,  
      "Gas"-gas zone, "Medical"-medical zone, "Timeout"-timeout zone, "Non-Alarm"-  
      disabled zone, "Key"-key zone, "24hSound"-24-hour annunciating zone. When  
      switching zone type, zone types supported by the peripheral can be obtained*/  
      "@opt": ["Instant","Delay","Follow"]  
    },  
    "heartBeatInterval": {  
      /*optional, int, heartbeat interval of the security control panel and the  
      peripheral, unit: second*/  
      "@opt": [5,10,20,30]  
    },  
    "LEDEnabled": {  
      /*optional, boolean, whether to enable the LED indicator*/  
      "@opt": [true,false]  
    },  
    "LEDLatchTime": {  
      /*optional, int, delay time of the LED indicator, unit: second*/  
      "@min": 0,  
      "@max": 0  
    },  
    "findMeEnabled": {  
      /*optional, boolean, whether to enable the Find Me function*/  
      "@opt": [true,false]  
    },  
    "isSupportSignalTest": true,  
    /*optional, boolean, whether it supports signal strength detection; if this  
    node is not returned or the value is false, it indicates that this function is  
    not supported*/  
    "isSupportZoneTest": true,  
    /*optional, boolean, whether it supports zone detection; if this node is not  
    returned or the value is false, it indicates that this function is not  
    supported*/  
    "isSupportFindMe": true,  
    /*optional, boolean, whether it supports FindMe detection; if this node is not  
    returned or the value is false, it indicates that this function is not
```

```
supported*/
    "isSupportFinalDoorExit": true
/*optional, boolean, whether it supports Final Door Exit function. If Final
Door Exit is enabled on a door magnetic contact (a detector), the area will be
armed immediately after the magnetic contact detects door opening and door
closing. If disabled, the area has to wait until a fixed countdown is over
before being armed; if this node is not returned or the value is false, it
indicates that this function is not supported*/
    }
}
```

A.3.173 JSON_SubSys

SubSys message in JSON format

```
{
  "SubSys":{
    "id": ,
/*required, integer type, partition No., it starts from 1*/
    "enabled": ,
/*required, boolean type, whether to enable the partition*/
    "linkageZones": ,
/*optional, array, zones linked to the partition*/
    "oneKeyArmEnabled": ,
/*optional, boolean type, whether to enable one-touch arming: "true"-yes,
"false"-no*/
    "isPublicSubSys":
/*optional, boolean, whether the partition is a public partition*/
  }
}
```

A.3.174 JSON_SubSysCap

SubSysCap capability message in JSON format

```
{
  "SubSysCap":{
    "id":{
/*required, integer, partition No., it starts from 1*/
      "@min": ,
      "@max":
    },
    "enabled":"true,false",
/*required, whether to enable the partition*/
    "publicNum":1,
/*optional, supported number of public partitions that can be configured*/
    "linkageZones":{
/*optional, number of linked zones supported by the partition*/
      "@min": ,

```

```
        "@max":
        },
        "zoneNo":{
/*optional, zone No. range*/
        "@min": ,
        "@max":
        },
        "oneKeyArmEnabled":"true,false",
/*optional, whether to enable one-touch arming: "true"-yes, "false"-no*/
        "method":{
/*required, methods supported by the function: "put"-edit, "getAll"-get all*/
        "@opt":["put","getAll"]
        },
        "isPublicSubSys":[true,false]
/*optional, boolean, whether the partition is a public partition. If this field
is "false" or is not returned, it indicates that the device does not support
getting or configuring this field by calling the configuration URI*/
    }
}
```

A.3.175 JSON_SirenCtrl

SirenCtrl message in JSON format

```
{
  "SirenCtrl":{
    "switch":"","
/*required, string type, switch: "open", "close"*/
    "List":[{
/*this node is only required when controlling multiple sirens*/
      "id":
/*integer type, siren No., it starts from 1, 0xffffffff-control all sirens*/
    }]
  }
}
```

A.3.176 JSON_SubSysList

JSON message about partition information list

```
{
  "SubSysList":[{
/*required, partition list*/
    "SubSys":{
/*optional, partition, it can be set to NULL if partitions are not needed*/
      "id": ,
/*required, int, partition No., which starts from 1*/
      "armType":"","
/*required, string, partition arming type: "stay"-stay arming, "away"-away
```

```
arming. This node is only valid for arming partitions in a batch*/
    "arming": "",
    /*optional, string, partition arming/disarming status, "stay"-stay arming,
    "away"-away arming, "disarm"-disarmed, "arming"*/
    "alarm": ,
    /*optional, boolean, whether the partition alarm is triggered: true, false*/
    "preventFaultArm": ,
    /*optional, boolean, whether to prevent fault arming: true, false*/
    "enabled": ,
    /*optional, boolean, whether to enable the partition*/
    "name": "",
    /*optional, string, partition name*/
    "moduleOperateCode": "12345",
    /*optional, string, module operation code, which should be encrypted*/
    "delayTime": 1
    /*optional, int, maximum remaining delay time of the delay zones in the current
    partition, unit:s; this node is valid when the value of the node arming is
    "stay" or "away"*/
  }
  ]]
}
```

A.3.177 JSON_SubSysTime

SubSysTime message in JSON format

```
{
  "SubSysTime":{
    "id": ,
    /*required, integer type, partition No., which starts from 1*/
    "enteyDelay1": ,
    /*optional, integer type, entrance delay 1, unit: second*/
    "enteyDelay2": ,
    /*optional, integer type, entrance delay 2, unit: second*/
    "exitDelay": ,
    /*optional, integer type, exiting delay time, unit: second*/
    "autoArmingEnable": ,
    /*optional, boolean type, whether to enable automatic arming, "true"-yes,
    "false"-no*/
    "autoArming": "",
    /*time of automatic arming enabled, which is in 24-hour system and adopts the
    device local time zone, it is valid only when the autoArmingEnable is "true"*/
    "autoDisarmingEnable": ,
    /*optional, boolean type, whether to enable automatic disarming, "true"-yes,
    "false"-no*/
    "autoDisarming": "",
    /*time of automatic disarming enabled, which is in 24-hour system and adopts
    the device local time zone, it is valid only when the autoDisarmingEnable is
    "true"*/
    "lateWarningEnable": ,
    /*optional, boolean type, whether to enable late warning*/
```

```

    "lateWarning": "",
    /*time of late warning enabled, which is in 24-hour system and adopts the
    device local time zone, it is valid only when the lateWarningEnable is "true"*/
    "weekendsExceptEnable": ,
    /*optional, boolean type, whether to enable automatic arming or disarming
    except weekend, "true"-yes, "false"-no, and it is valid only when
autoArmingEnable is "true" or autoDisarmingEnable is "true"*/
    "WeekCfg": [{
        "dayOfWeek": ""
    /*optional, string, day of the week: "Monday", "Tuesday", "Wednesday",
    "Thursday", "Friday", "Saturday", "Sunday". This node is valid when
weekendsExceptEnable is "true"*/
    }],
    "HolidayExceptionsCfg": {
        "enable": ,
    /*optional, boolean, whether to support configuring holiday time period: "true"-
    yes, "false"-no. Up to six holiday time periods are supported, and currently
    only months and days of holiday time periods are required*/
        "HolidayCfg": [{
            "startDate": "",
    /*optional, string, start date, e.g., "05-09" refers to May 9*/
            "endDate": ""
    /*optional, string, end date, e.g., "05-09" refers to May 9*/
        }],
    },
    "perimeterDelayTime": ,
    /*optional, delay time of perimeter zone alarm, unit: second, when the
    perimeter zone alarm is triggered, the siren output will not start until the
    delay time is ended (during the delay time, do not disarm or clear alarm,
    otherwise, there is also no siren output)*/
    "sounderTime": ,
    /*optional, audio alarm duration, unit: second*/
    "heartbeatInterval": "",
    /*optional, string type, heartbeat interval of security control panel and
    peripherals*/
    "ACcheckTime":
    /*optional, integer type, AC detection time*/
    }
}

```

A.3.178 JSON_SubSysTimeCap

SubSysTimeCap capability message in JSON format

```

{
    "SubSysTimeCap": {
        "id": {
    /*required, partition No. range*/
            "@min": ,
            "@max":
        },
    },
}

```



```
"enteyDelay1":{
/*optional, entrance delay 1, unit: second*/
    "@min": ,
    "@max":
},
"enteyDelay2":{
/*optional, entrance delay 2, unit: second*/
    "@min": ,
    "@max":
},
"exitDelay":{
/*optional, exiting delay time, unit: second*/
    "@min": ,
    "@max":
},
"autoArmingEnable": ,
/*optional, whether to enable automatic arming, "true"-yes, "false"-no*/
"autoDisarmingEnable": ,
/*optional, whether to enable automatic disarming, "true"-yes, "false"-no*/
"lateWarningEnable": ,
/*optional, whether to enable late warning, "true"-yes, "false"-no*/
"weekendsExceptEnable": ,
/*optional, whether to enable automatic arming or disarming except weekends,
"true"-yes, "false"-no*/
    "WeekCfg":{
        "dayOfWeek":{
/*day of the week, this node is valid when weekendsExceptEnable is "true"*/
            "@opt":["Monday", "Tuesday", "Wednesday", "Thursday", "Friday",
"Saturday", "Sunday"]
        }
    },
    "HolidayExceptionsCfg":{
        "enable":"true,false",
/*optional, boolean, whether to support configuring holiday time period: "true"-
yes, "false"-no. Up to six holiday time periods are supported, and currently
only months and days of holiday time periods are required*/
        "HolidayCfgNum":{
/*optional, integer, number of holiday configurations, it is the supported
maximum number of holiday configurations*/
            "@min":0,
            "@max":0
        },
        "dateLen":{
/*optional, integer, name length of the start date and the end date*/
            "@min":0,
            "@max":0
        }
    },
    "perimeterDelayTime":{
/*optional, the delay time of perimeter zone alarm, unit: second*/
        "@min": ,
        "@max":
```

```
    },
    "sounderTime":{
/*optional, audio alarm duration, unit: second*/
    "@min": ,
    "@max":
    }
    "heartbeatInterval":{
/*optional, heartbeat interval of security control panel and peripherals*/
    "@opt":"5min,10min,20min,30min,1h,2h,4h,6h"
    },
    "ACcheckTime":{
/*optional, detection time when the AC is powered off, unit: second*/
    "@min": ,
    "@max":
    }
    }
}
```

Remarks

- For the whole partition, only two delay time can be set, and either delay 1 or delay 2 can be set for each zone.
- The automatic arming and disarming time is applied to the whole partition, and it cannot be set for a single zone.

A.3.179 JSON_SurrondParaCap

JSON message about the configuration capability of the device environment

```
{
  "SurrondParaCap":{
    "temperDetectEnable":"true,false",
/*optional, boolean, whether to enable temperature detection*/
    "tempLowRange":{
/*optional, float, minimum temperature threshold, unit: degree Celsius*/
    "@min":0.1,
    "@max":0.2
    },
    "tempHighRange":{
/*optional, float, maximum temperature threshold, unit: degree Celsius*/
    "@min":0.1,
    "@max":0.2
    },
    "devTemperature":{
/*optional, float, device temperature, unit: degree Celsius*/
    "@min":0.1,
    "@max":0.2
    }
  }
}
```

A.3.180 JSON_SurrondParaCfg

JSON message about device environment parameters

```
{
  "SurrondParaCfg":{
    "temperDetectEnable": ,
    /*optional, boolean, whether to enable temperature detection*/
    "tempLowRange": ,
    /*optional, float, minimum temperature threshold, unit: degree Celsius*/
    "tempHighRange": ,
    /*optional, float, maximum temperature threshold, unit: degree Celsius*/
    "devTemperature":
    /*optional, float, device temperature, unit: degree Celsius. The device
    temperature can only be obtained*/
  }
}
```

A.3.181 JSON_SysAutoCheckTimeCfg

Message about the automatic audio and video detection parameters in JSON format.

```
{
  "SysAutoCheckTimeCfg":{
    "enable": ,
    /*required, boolean, whether to enable automatic audio and video detection:
    true-enable, false-disable*/
    "dayOfWeek": "",
    /*optional, string, day of the week: "Monday", "Tuesday", "Wednesday",
    "Thursday", "Friday", "Saturday", "Sunday", "Everyday", "Null"*/
    "TimeSegment":{
      "beginTime": ""
    }
    /*optional, string, start time. For example, "10:10" indicates that the start
    time is 10:10*/
  }
}
```

A.3.182 JSON_SysCheckManually

Message about the manual audio and video detection parameters in JSON format.

```
{
  "SysCheckManually":{
    "enable":
    /*required, boolean, whether to enable manual detection of the system's audio
    and video: true-enable, false-disable*/
  }
}
```

```
}  
}
```

A.3.183 JSON_TimeCfg

TimeCfg message in JSON format

```
{  
  "TimeCfg":{  
    "zoneAlarmEventTimeIntervalFilterCfg":  
/*optional, integer, time interval. Duplicate zone alarms in the configured  
time interval will be filtered*/  
  }  
}
```

A.3.184 JSON_TimeCfgCap

TimeCfgCap capability message in JSON format

```
{  
  "TimeCfgCap":{  
    "zoneAlarmEventTimeIntervalFilterCfg":{  
/*optional, integer, time interval. Duplicate zone alarms in the configured  
time interval will be filtered*/  
      "@opt":[5, 10, 15]  
    }  
  }  
}
```

A.3.185 JSON_id

ID message in JSON format.

```
{  
  "id":  
/*required, integer type, network user No., which starts from 1, "1"-installer,  
"2"-administrator, other No.-normal user*/  
}
```

A.3.186 JSON_UserCfg

UserCfg message in JSON format.

```
{  
  "UserCfg":{  
    "id": ,  
  }  
}
```

```
/*integer type, user No., it is optional for POST method, but it is required
for other methods*/
  "userName": "",
/*required, string type, user name*/
  "password": "",
/*required, string type, user password*/
  "keypadPassword": "",
/*optional, string type, keyboard password, write-only*/
  "messageSendEnabled":,
/*optional, boolean type, whether to enable message notification, "true,
false"*/
  "bypassEnabled":,
/*optional, boolean type, whether to enable bypass or bypass recovery*/
  "duressEnabled":,
/*optional, boolean type, whether to enable duress alarm*/
  "MacList": [{
    "Mac": ""
  }],
/*optional, MAC address bound by the user*/
  "IpAddrList": [{
/*optional*/
    "IpAddr": {
      "ipVersion": "",
/*optional, string type, IP address version information: "v4"-IPv4, "v6"-IPv6*/
      "ipAddress": ""
/*optional, IPV4 or IPV6 address*/
    }
  }]
}
```

A.3.187 JSON_UserCfgCap

UserCfgCap message in JSON format

```
{
  "UserCfgCap": {
    "userName": {
/*required, range of user name length*/
      "@min":,
      "@max":
    },
    "password": {
/*required, range of user password length*/
      "@min":,
      "@max":
    },
    "keypadPassword": {
/*optional, range of keyboard password length*/
      "@min":,
      "@max":
    }
  }
}
```

```
    },
    "messageSendEnabled":{
/*optional, whether to enable message notification*/
    "@opt":"true,false"
    },
    "bypassEnabled":{
/*optional, whether to enable bypass or bypass recovery*/
    "@opt":"true,false"
    },
    "duressEnabled":{
/*optional, whether to enable duress alarm*/
    "@opt":"true,false"
    },
    "MacNum": ,
/*optional, integer type, number of MAC addresses that can be bound by one
user*/
    "Mac":{
/*optional, range of MAC address's length*/
    "@min":,
    "@max":
    },
    "IPAddrNum":,
/*optional, integer type, number of IP addresses that can be bound by one user*/
    "ipVersion":{
/*optional, IP address version information*/
    "@opt":"v4,v6"
    },
    "ipAddress":{
/*optional, IP address length*/
    "@min":,
    "@max":
    },
    "method":{
    "@opt":"post,del,put,getAll"
    }
  }
}
```

A.3.188 JSON_voicePromptCfg

JSON message about voice prompt parameters

```
{
  "voicePromptCfg":{
    "callPrompt":"","
/*optional, string, voice prompt for calling*/
    "rejectPrompt":"","
/*optional, string, voice prompt for rejection*/
    "voiceTalkStopPrompt":"","
/*optional, string, end voice prompt for two-way audio*/
```

```
}  
}
```

A.3.189 JSON_WiredDetectorCap

JSON message about the capability of wired detectors

```
{  
  "zoneType": {  
    /*optional, object, zone type*/  
    "@opt": ["Instant", "Delay", "24h", "Non-Alarm", "Timeout", "Key",  
    "Emergency", "Medical", "Fire", "Gas"]  
    /*optional, array of string*/  
  },  
  "detectorContactMode": {  
    /*optional, object, detector mode*/  
    "@opt": ["NO", "NC", "rollerShutter", "EOL", "DEOL-NC", "DEOL-NO"]  
    /*optional, array of string*/  
  },  
  "impulseCountTime": {  
    /*optional, object, pulse count time, unit:second, this node is valid when  
    detectorContactMode is "rollerShutter"*/  
    "@opt": [10, 20, 30, 40, 50, 60]  
    /*optional, array of int*/  
  },  
  "impulsesBeforeAlarm": {  
    /*optional, object, number of pulses before alarm*/  
    "@opt": [2, 4, 6]  
    /*optional, array of int, the minimum value is 2 and the maximum value is 30,  
    this node is valid when detectorContactMode is "rollerShutter"*/  
  },  
  "detectorContactModeList": [  
    /*optional, array, detector contact mode list*/  
    {  
      "detectorContactMode": "NO",  
      /*optional, enum, detector contact mode: "NO", "NC", "A", "EOL", "DEOL-NO",  
      "DEOL-NC"*/  
      "pulseSensitivity": {  
        /*optional, int, pulse sensitivity, unit: ms, when detectorContactMode is NO/  
        NC, the value is 30,100,1000; when detectorContactMode is EOL/DEOL-NO/DEOL-NC,  
        the value is 10,100,250,500,750*/  
        "@opt": [10, 30, 100, 250, 500, 750, 1000]  
        /*optional, array of int*/  
      }  
    }  
  ],  
  "alarmResistance": {  
    /*optional, object, alarm resistance, unit: k, this node is valid when  
    detectorContactMode is EOL/DEOL-NO/DEOL-NC*/  
    "@opt": [1.0, 2.2, 4.7, 8.2]  
    /*optional, array of float*/  
  }  
}
```

```

    },
    "tamperResistance": {
/*optional, object, tamper resistance, unit: k, this node is valid when
detectorContactMode is DEOL-NO/DEOL-NC*/
        "@opt": [1.0, 2.2, 4.7, 8.2]
/*optional, array of float*/
    },
    "isSupportZoneTest": true,
/*optional, boolean, whether it supports zone detection*/
    "isNotSupportCrossZoneCfg": true,
/*optional, boolean, whether it does not support cross zone configuration*/
    "isNotSupportByPass": true,
/*optional, boolean, whether it does not support bypass*/
    "isSupportDoubleKnock": true,
/*optional, boolean, whether it supports double knock*/
    "isNotSupportSirenDelayTime": true,
/*optional, boolean, whether it does not support siren delay configuration*/
    "isSupportDoubleZoneCfg": true,
/*optional, boolean, whether it supports configuring double zones*/
    "isSupportFinalDoorExit": true,
/*optional, boolean, whether it supports "Final Door Exit"*/
    "isNotSupportChimeEnabled": true,
/*optional, boolean, whether it does not support enabling chime*/
    "isSupportTimeRestart": true,
/*optional, boolean, whether it supports restarting countdown*/
    "isSupportFindMe": true
/*optional, boolean, whether it supports Find Me detection, if the value is
false or is not returned, it indicates that this function is not supported*/
    }

```

A.3.190 JSON_WiredDetectorType

JSON message about the wired detector type

```

{
    "wiredDetectorType": "other",
/*required, string, detector type: "dualTechnologyPirDetector",
"tripleTechnologyPirDetector", "glassBreakDetector", "activeInfraredDetector",
"passiveInfraredDetector", "magneticContact", "panicButton",
"waterLeakDetector", "humidityDetector", "temperatureDetector",
"smokeDetector", "combustibleGasDetector", "vibrationDetector", "other",
"tamperDetector"*/
    "accessModuleType": "localTransmitter"
/*optional, string, access module type: "transmitter", "multiTransmitter",
"localTransmitter", "localZone", "keypad"*/
}

```


A.3.191 JSON_WirelessRecv

JSON message about the registration status

```
{
  "WirelessRecv": {
    /*optional, object*/
    "status": "processing",
    /*optional, string, current status: "processing", "success", "failed"*/
    "exDevType": "detector",
    /*optional, string, peripheral module type: "detector", "wirelessSiren"-
    wireless siren, "wirelessRepeater"-wireless repeater, "wirelessOutput"-wireless
    output module*/
    "detectorType": "panicButton",
    /*optional, string, detector type: "panicButton", "smokeDetector",
    "combustibleGasDetector", "temperatureDetector", this node is valid when the
    value of detectorType is "exDevType"*/
    "exDevSeq": "test"
    /*optional, string, the serial No. of the peripheral module*/
  }
}
```

A.3.192 JSON_Zone

JSON message about parameters of a specific zone

```
{
  "Zone":{
    "id": ,
    /*required, int, zone No., which starts from 0*/
    "enable": ,
    /*optional, boolean, whether to enable the zone*/
    "zoneName": "",
    /*optional, string, zone name*/
    "zoneType": "",
    /*optional, string, zone type*/
    "detectorType": "",
    /*optional, string, read-only, detector type: "panicButton", "magneticContact",
    "smokeDetector", "activeInfraredDetector", "passiveInfraredDetector",
    "glassBreakDetector", "vibrationDetector", "dualTechnologyPirDetector",
    "tripleTechnologyPirDetector", "humidityDetector", "temperatureDetector",
    "combustibleGasDetector", "dynamicSwitch", "controlSwitch", "smartLock",
    "waterDetector", "displacementDetector", "singleInfraredDetector",
    "singleZoneModule", "curtainInfraredDetector", "pircam", "slimMagneticContact",
    "indoorDualTechnologyDetector", "magnetShockDetector", "waterLeakDetector",
    "wirelessSmokeDetector", "wirelessGlassBreakDetector",
    "wirelessTemperatureHumidityDetector", "wirelessHeatDetector",
    "wirelessCODetector", "wirelessPIRCeilingDetector",
    "wirelessExternalMagnetDetector", "wirelessPIRCurtainDetector",
```

```
"wirelessDTAMCurtainDetector", "outdoorDetector", "other", "tamperDetector"*/
    "subSystemNo": ,
/*optional, integer type, read-only, partition No.*/
    "delayTime": ,
/*optional, select delay time as "entryDelay1" and "entryDelay2" in the timer
configuration of partition, and this node is valid only when the zoneType is
"Delay"*/
    "stayAwayEnabled": ,
/*optional, boolean, whether to enable stay arming bypass: "true"-yes, "false"-
no*/
    "chimeEnabled": ,
/*optional, boolean, whether to enable doorbell: "true"-yes, "false"-no*/
    "silentEnabled": ,
/*optional, boolean, whether to enable mute: "true"-yes, "false"-no*/
    "timeoutType": "",
/*optional, string, timeout type*/
    "timeoutLimit": ,
/*boolean, timeout threshold: "true"-short timeout, "false"-long timeout, and
this node is valid only when the zoneType is "Timeout"*/
    "timeout": ,
/*int, timeout, and this node is valid only when the zoneType is "Timeout"*/
    "relateDetector": "",
/*optional, boolean, linked detector*/
    "detectorSeq": "",
/*optional, string, detector No., and it is required when relateDetector is
"true"*/
    "RelatedChanList": [{
/*optional, linked channel list*/
        "RelatedChan": {
            "relator": "",
/*required, string, channel linkage types, which links the channel when alarm
is triggered or event occurred*/
            "cameraSeq": "",
/*optional, string, camera serial No.*/
            "relatedChan": ,
/*optional, int, linked channel No.*/
            "linkageCameraName": "test"
/*optional, string, name of the camera linked with the zone, the maximum length
is 64 bytes*/
        }
    }],
    "address": ,
/*optional, read-only, integer, module address, wireless module does not return
this node*/
    "linkageAddress": ,
/*optional, read-only, integer, linked module address, wired module does not
return this node*/
    "moduleChannel": ,
/*optional, integer, module channel No.*/
    "moduleType": "",
/*optional, read-only, string, module type: "localWired"-local wired module,
"extendWired"-extended wired module, "localWireless"-local wireless module,
```

```
"extendWireless"-extended wireless module*/
    "moduleStatus": "",
/*optional, read-only, string, module status: "online", "offline",
"heartbeatAbnormal"-heartbeat exception*/
    "checkTime": ,
/*optional, integer, detector offline duration, unit: hour*/
    "sensitivity": ,
/*optional, integer, zone sensitivity: 10-10 ms, 250-250 ms, 500-500 ms,
750-750 ms*/
    "resistor": ,
/*optional, float, EOL (End-of-Line) resistor: 2.2-2.2k, 3.3-3.3k, 5.6-5.6k,
8.2-8.2k*/
    "tamperType": "",
/*optional, string, tampering type: "disable", "normalOpen"-remain open,
"normalClose"-remain closed*/
    "zoneAttrib": "",
/*optional, read-only, string, zone attribute: "wired", "wireless". If this
node is not returned, the default zone attribute is "wireless"*/
    "linkagePircamCapCfg": ,
/*optional, boolean, whether it supports configuring pircam capture linkage*/
    "linkageFileName": "",
/*optional, string, name of the linked file*/
    "CrossZoneCfg": {
/*optional, object, cross-zone configuration*/
        "isAssociated": true,
/*required, boolean, whether this zone is linked with other zones: true, false*/
        "supportAssociatedZone": [1, 2, 3],
/*optional, array, detector-linked zones without being linked with other zones*/
        "alreadyAssociatedZone": [1, 2, 3],
/*optional, array, zones that have been linked with other zones*/
        "associateZoneCfg": [1, 2, 3],
/*optional, array, zone(s) that this zone is linked with*/
        "associateTime": 1,
/*optional, int, time of the linkage between zones*/
        "supportLinkageChannelID": [1, 2, 3],
/*optional, array, No.s of the channels that can be linked with network
cameras*/
        "alreadyLinkageChannelID": [1, 2, 3],
/*optional, array, No.s of the channels that have been linked with network
cameras*/
        "linkageChannelID": [1, 2]
/*optional, array, No.s of the channels to be linked with network cameras*/
    },
    "relay": "",
/*optional, linked relays, e.g., "1,3" indicates the linked relay 1 and relay
3*/
    "CoordinateList": [{
/*coordinate information list, the polar coordinates will be returned*/
        "angle": ,
/*optional, float, angle, unit: degree*/
        "distance":
/*optional, float, distance, unit: meter*/
```

```
    }},
    "doubleZoneCfgEnable": ,
    /*optional, boolean, whether to enable double-detector zone. If this field is
    "true", it indicates that the double-detector zone has been enabled for the
    zone and two detectors will be linked. For the upper level, two detectors
    require two zone numbers*/
    "extendZoneNo": ,
    /*optional, integer, read-only, extended zone No. after enabling double-
    detector zone. This field is valid when the configuration URI is called to
    configure parameters of the linked zone. For example, if the double-detector
    zone has been enabled for the linked zone (zone 1) and the extended zone No. of
    zone 1 is 9, when the configuration URI is called to configure parameters of
    zone 1, id in the message should be set to 1, extendZoneNo should be set to 9,
    and relatedZoneNo is invalid at this point*/
    "extendZoneName": ,
    /*optional, string, name of the extended zone, the maximum length is 32 bytes*/
    "relatedZoneNo": ,
    /*optional, integer, read-only, linked zone No. This field is valid when the
    configuration URI is called to configure parameters of the extended zone. For
    example, if the double-detector zone has been enabled for the linked zone (zone
    1) and the extended zone No. of zone 1 is 9, when the configuration URI is
    called to configure parameters of zone 9, id in the message should be set to 9,
    relatedZoneNo should be set to 1, and extendZoneNo is invalid at this point*/
    "relatedZoneName": ,
    /*optional, string, name of the linked zone, the maximum length is 32 bytes*/
    "keyZoneTriggerTypeCfg": "",
    /*optional, string, triggering type settings of the key zone:
    "secondTriggerDisarm"-the key zone will be disarmed after being triggered for
    the second time (before enabling this function, the key zone will be armed
    after being triggered; after enabling this function, the key zone will be armed
    after being triggered for the first time, and it will be disarmed after being
    triggered for the second time), "triggerArm"-the key zone will be armed every
    time being triggered*/
    "newKeyZoneTriggerTypeCfg": "",
    /*optional, string, triggering type settings of the key zone (new version):
    "triggerTimes"-by trigger times, "zoneStatus"-by zone status. If this node is
    supported by the device, the node keyZoneTriggerTypeCfg will not be supported*/
    "zoneStatusCfg": "",
    /*optional, dependent, string, zone status settings: "triggerArm"-the zone will
    be armed every time being triggered, "triggerDisArm"-the zone will be disarmed
    every time being triggered. This node is required when newKeyZoneTriggerTypeCfg
    is "zoneStatus"*/
    "AlarmSoundInterlink": {
    /*optional, object, alarm linkage*/
        "supportLinkageZones": [1, 3, 5],
    /*optional, array, zones that support alarm linkage*/
        "linkageZones": [1, 3, 5]
    /*optional, array, zones with alarm linkage configured*/
    },
    "RelatedPIRCAM": {
    /*optional, object, linked pircam information*/
        "supportLinkageZones": [1, 3, 5],
```

```
/*optional, array, zones that can be linked with pircam*/
    "linkageZone": [1],
/*optional, array, zones that have been linked with pircam*/
    "linkagePIRCAMName": "test"
/*optional, string, name of the pircam linked with the zone, the maximum length
is 64 bytes*/
    },
    "reportSendDelayTime": 30,
/*optional, int, delay time of uploading the alarm to ARC, range:[10,3600],
unit: second; this node is valid for delayed zones only*/
    "finalDoorExitEnabled": true,
/*optional, boolean, whether to enable Final Door Exit function: true-enable,
false-disable, If Final Door Exit is enabled on a door magnetic contact (a
detector), the area will be armed immediately after the magnetic contact
detects door opening and door closing. If disabled, the area has to wait until
a fixed countdown is over before being armed*/
    "timeRestartEnabled": true,
/*optional, boolean, whether to enable resetting delay time for exit; this node
is valid only for delayed zones*/
    "modifiedZoneNo": 2,
/*optional, int, Zone No. after it is modified, range:[0,255]*/
    "swingerLimitActivation": 5,
/*optional, int, allowed times of triggers*/
    "detectorContactMode": "NO",
/*optional, string, detector mode: "NO"-always open, "NC"-always closed,
"rollerShutter"(customized)*/
    "impulseCountTime": 10,
/*optional, int, pulse count time, unit:second*/
    "impulsesBeforeAlarm": 2,
/*optional, int, number of pulses before alarm*/
    "detectorInputMode": "pulse",
/*optional, string, detector input mode: "pulse", "latch";this node is valid
when the value of detectorContactMode is "NO" or "NC"*/
    "pulseDuration": 5,
/*optional, int, pulse interval, unit: second, this node is valid when the
value of detectorInputMode is "pulse"*/
    "detectorTamperMode": "NO",
/*optional, string, detector temper-resistant mode: "NO"-always enabled, "NC"-
always disabled*/
    "antiMasking": "NO",
/*optional, string, anti-masking mode: "NO"-always enabled, "NC"-always
disabled*/
    "pulseSensitivity": 10,
/*optional, int, pulse sensitivity, 10 (10ms),30 (30ms),100 (100ms),250 (250ms),
500 (500ms),750 (750ms),1000 (1000ms), unit: ms, when detectorContactMode is NO/
NC, the value is 30, 100, 1000; when detectorContactMode is EOL/DEOL-NO/DEOL-
NC, the value is 10,100,250,500,750*/
    "alarmResistance": 2.2,
/*optional, int, alarm resistance, 1.0 (1k),2.2 (2.2k),4.7 (4.7k),8.2 (8.2k),
unit: k, this node is valid when detectorContactMode is EOL/DEOL-NO/DEOL-NC*/
    "tamperResistance": 2.2,
/*optional, int, tamper resistance, 1.0 (1k),2.2 (2.2k),4.7 (4.7k),8.2 (8.2k),
```

```
unit: k, this node is valid when detectorContactMode is DEOL-NO/DEOL-NC*/
    "accessModuleType": "transmitter",
    /*optional, string, access module type: "transmitter" (transmitter peripheral),
    "multiTransmitter", "localTransmitter", "localZone", "keypad"*/
    "relatedAccessModuleID": 1
    /*optional, int, ID of linked access module*/
    "armMode": "and"
    /*optional, string, arming mode: "and", "or"*/
  }
}
```

A.3.193 JSON_ZoneCond

ZoneCond message in JSON format

```
{
  "ZoneCond":{
    "searchID":"","
    /*required, string type, search ID, which is used to confirm the upper-level
    platform or system. If the platform or the system is the same one during two
    searching, the search history will be saved in the memory to speed up next
    searching*/
    "searchResultPosition": ,
    /*required, integer32 type, the start position of the search result in the
    result list. When there are multiple records and you cannot get all search
    results at a time, you can search for the records after the specified position
    next time*/
    "maxResults":
    /*required, integer32 type, maximum number of search results that can be
    obtained this time by calling the URI. If maxResults exceeds the range returned
    by the device capability, the device will return the maximum number of search
    results according to the device capability and will not return error message*/
  }
}
```

A.3.194 JSON_ZoneList

JSON message about zone status list

```
{
  "ZoneList":[{
    "Zone":{
      "id": ,
      /*required, int, zone ID*/
      "name":"","
      /*optional, string, zone name*/
      "status":"","
      /*optional, string, zone status, "notRelated"-no zone linked, "online"-online,
      "offline"-offline, "trigger"-triggered, "breakDown"-fault, "heartbeatAbnormal"-
```

```
heartbeat exception*/
    "reason": "short",
/*optional, string, default reason: "short", "break"*/
    "tamperEvident": ,
/*optional, boolean, zone tampering alarm status, true-triggered, false-not
triggered*/
    "shielded": ,
/*optional, boolean, zone disabling status, true-disabled, false-not
disabled*/
    "bypassed": ,
/*optional, boolean, whether to bypass the zone, true-bypassed, false-bypass
recovered*/
    "armed": ,
/*required, boolean, whether to arm the zone, true-armed, false-disarmed*/
    "isArming": ,
/*optional, boolean, whether the zone is armed, this node can only be set to
"true"*/
    "alarm": ,
/*optional, boolean, whether the zone alarm is triggered, true-triggered, false-
not triggered*/
    "charge": "",
/*optional, string, zone battery status, "normal", "lowPower"-low battery*/
    "chargeValue": ,
/*optional, int, battery power value which is between 0 and 100*/
    "signal": ,
/*optional, int, signal strength, which ranges from 0 to 255*/
    "temperature": ,
/*optional, read-only, int, temperature*/
    "detectorType": "",
/*optional, string, type of the detector linked to the zone, see details about
the supported detector types in JSON_ZoneCap*/
    "model": "",
/*optional, string, model*/
    "zoneType": ""
/*optional, string, zone type: "Instant"-instant zone, "Delay"-delay zone,
"Follow"-follow zone, "Perimeter"-perimeter zone, "24hNoSound"-24-hour silent
zone, "Emergency"-panic zone, "Fire"-fire zone, "Gas"-gas zone, "Medical"-
medical zone, "Timeout"-timeout zone, "Non-Alarm"-disabled zone, "Key"-key
zone*/
    "humidity": 20,
/*optional, int, read-only, humidity, the value is between 10% and 90%*/
    "healthStatus": "normal",
/*optional, string, read-only, health status: "normal", "fault"*/
    "antiMaskingEnabled": true,
/*optional, boolean, read-only, whether to enable anti-masking: true-enable,
false-disable*/
    "mountingType": "wall",
/*optional, string, read-only, mounting type: "wall", "ceiling"*/
    "magnetOpenStatus": true,
/*optional, boolean, whether the magnetic contact is open: true (open), false
(closed)*/
    "version": "test",
```

```
/*optional, string, detector version No., the maximum length is 32 bytes*/
    "pirCamConnected": true,
/*optional, boolean, whether the outdoor tri-tech detector and PIR camera are
connected: true (yes), false (no)*/
    "accessModuleType": "transmitter",
/*optional, enum, access module type: "transmitter", "localTransmitter",
"multiTransmitter", "localZone", "keypad"*/
    "relatedAccessModuleID": 1,
/*optional, int, linked access module ID*/
    "address": 254,
/*optional, int, wired (extended) module address, this node works with
accessModuleType*/
    "zoneAttrib": "wired",
/*optional, enum, zone attribute: "wired", "wireless" (default)*/
    "voltage": 1,
/*optional, int, voltage of the zone*/
    "signalStrength": 1,
/*optional, int, signal strength, range: [-128,127]*/
    "mainCharge": "normal",
/*optional, enum, main (adapter) power status: "normal", "lowPower". Main power
and zone power refers to the zone's adapter power status and the zone's battery
status respectively*/
    "preheatStatus": "processing",
/*optional, enum, pre-heat status: "processing", "success"*/
    "usefulLifeStatus": "expire",
/*optional, enum, expiry status: "expire", "normal"*/
    "mazeStatus": "abnormal",
/*optional, enum, maze status: "abnormal", "normal"*/
    "sensorStatus": "abnormal",
/*optional, enum, sensor status: "abnormal", "normal"*/
    "deviceNo": 1
/*optional, int, device No., range: [1,1000]*/
    }
  }
}
```

A.3.195 JSON_ZonesCap

JSON message about the zone configuration capability

```
{
  "ZonesCap":{
/*wireless zone capability*/
    "id":{
/*required, zone No. range*/
      "@min": ,
      "@max":
    },
    "enable":"true,false",
/*optional, whether to enable the zone*/
    "zoneName":{
```



```

/*optional, zone name length*/
    "@min": ,
    "@max":
},
"ZoneTypeList": [{
    "ZoneType": {
        "type": "",
/*required, supported zone types: "Instant"-instant zone, "Delay"-delayed zone,
"Follow"-follow zone, "Perimeter"-perimeter zone, "24hNoSound"-24-hour silent
zone, "Emergency"-panic zone, "Fire"-fire zone, "Gas"-gas zone, "Medical"-
medical zone, "Timeout"-timeout zone, "Non-Alarm"-disabled zone, "EarlyWarning"-
early warning zone, "Warning"-warning zone, "Shield"-shielded zone, "Key"-key
zone, "24hSound"-24-hour annunciating zone*/
        "sptProp": [""]
/*zone properties supported by the zone: "delayTime"-delay, "awayBypass"-stay
arming bypass, "chime"-doorbell, "silent"-mute, "timeout"-timeout,
"doubleKnock"-double alarm, "crossZoneCfg"-cross-zone settings,
"newKeyZoneTriggerTypeCfg"-trigger type settings of the key zone (new version),
"armNoBypass"-no bypass during arming, "reportSendDelay"-delay alarm uploading,
"finalDoorExit"-start arming immediately after the magnetic contact detects
door opening and door closing*/
        "KeyZoneTriggerTypeCfg": {
/*optional, string, triggering type settings of the key zone:
"secondTriggerDisarm"-the key zone will be disarmed after being triggered for
the second time (before enabling this function, the key zone will be armed
after being triggered; after enabling this function, the key zone will be armed
after being triggered for the first time, and it will be disarmed after being
triggered for the second time), "triggerArm"-the key zone will be armed every
time being triggered*/
            "opt": ["secondTriggerDisarm", "triggerArm"]
        }
    }
}],
"newKeyZoneTriggerTypeCfg": {
/*optional, string, triggering type settings of the key zone (new version):
"triggerTimes"-by trigger times, "zoneStatus"-by zone status. If this node is
supported by the device, the node keyZoneTriggerTypeCfg will not be supported*/
    "@opt": ["triggerTimes", "zoneStatus"]
},
"zoneStatusCfg": {
/*optional, dependent, string, zone status settings: "triggerArm"-the zone will
be armed every time being triggered, "triggerDisArm"-the zone will be disarmed
every time being triggered. This node is required when newKeyZoneTriggerTypeCfg
is "zoneStatus"*/
    "@opt": ["triggerArm", "triggerDisArm"]
},
"detectorType": {
/*required, "panicButton"-panic button, "magneticContact"-door magnetic contact
detector, "smokeDetector"-smoke detector, "activeInfraredDetector"-active IR
detector, "passiveInfraredDetector"-PIR detector, "glassBreakDetector"-glass-
break detector, "vibrationDetector"-shock detector, "dualTechnologyPirDetector"-
dual-technology motion detector, "tripleTechnologyPirDetector"-triple-

```

```
technology detector, "humidityDetector"-humidity detector,
"temperatureDetector"-temperature detector, "combustibleGasDetector"-gas
detector, "dynamicSwitch"-dynamic switch, "controlSwitch"-control switch,
"smartLock"-smart lock, "waterDetector"-water detector, "displacementDetector"-
displacement detector, "singleInfraredDetector"-door contact,
"singleZoneModule"-wireless single input expander, "curtainInfraredDetector"-IR
curtain detector, "pircam"-pircam detector (detector equipped with camera),
"slimMagneticContact", "indoorDualTechnologyDetector", "magnetShockDetector",
"waterLeakDetector", "wirelessSmokeDetector",
"wirelessGlassBreakDetector", "wirelessTemperatureHumidityDetector",
"wirelessHeatDetector", "wirelessCODetector", "wirelessPIRCeilingDetector",
"wirelessExternalMagnetDetector", "wirelessPIRCurtainDetector",
"wirelessDTAMCurtainDetector", "other"-others*/

"@opt": "panicButton, magneticContact, smokeDetector, activeInfraredDetector, passive
InfraredDetector, glassBreakDetector,
vibrationDetector, dualTechnologyPirDetector, tripleTechnologyPirDetector, humidity
Detector, temperatureDetector, combustibleGasDetector, dynamicSwitch, controlSwitch,
smartLock, waterDetector, displacementDetector, singleInfraredDetector, singleZoneMod
ule, curtainInfraredDetector, pircam, slimMagneticContact, indoorDualTechnologyDetec
tor, magnetShockDetector, waterLeakDetector, wirelessSmokeDetector,
wirelessGlassBreakDetector, wirelessTemperatureHumidityDetector,
wirelessHeatDetector, wirelessCODetector, wirelessPIRCeilingDetector,
wirelessExternalMagnetDetector, wirelessPIRCurtainDetector,
wirelessDTAMCurtainDetector, other"
    },
    "subSystemNo": {
/*required, partition No. range*/
        "@min": ,
        "@max":
    },
    "linkageSubSystem": {
/*optional, object, partition that the network camera is linked to*/
        "@size": 1,
/*optional, int*/
        "@min": 1,
/*optional, int, the minimum value*/
        "@max": 2
/*optional, int, the maximum value*/
    },
    "supportLinkageSubSystemList": {
/*optional, object, partitions that the network camera can be linked to*/
        "@size": 1,
/*optional, int*/
        "@min": 1,
/*optional, int, the minimum value*/
        "@max": 2
/*optional, int, the maximum value*/
    },
    "stayArmDelayTime": {
/*optional, object*/
        "@min": 1,
```

```
/*optional, int, the minimum value*/
    "@min": 21
/*optional, int, the maximum value*/
    },
    "sirenDelayTime": {
/*optional, object, delay time for siren*/
        "@opt": [0, 10, 20, 30]
/*optional, int*/
    },
    "delayTime":{
/*optional, select delay 1 or delay 2 (which is in the timer configuration of
partition, and corresponds to "entryDelay1" and "entryDelay2")*/
        "@opt": "1,2"
    },
    "stayAwayEnabled": "true,false",
/*optional, whether to enable stay arming bypass*/
    "chimeEnabled": "true,false",
/*optional, whether to enable doorbell*/
    "chimeWarningType":{
/*doorbell tone types: "single"-single tone, "continuous"-continuous tone, this
node is valid only when chimeEnabled is "true"*/
        "@opt": "single,continuous "
    },
    "silentEnabled": "true,false",
/*optional, whether to support enabling mute*/
    "timeoutLimit":{
/*optional, configuration of zone timeout threshold*/
        "@opt": "true,false"
    },
    "timeoutType":{
/*optional, timeout type, "tigger"-triggering timed out, "recover"-restoring
timed out*/
        "@opt": "tigger,recover"
    },
    "limitTimeout":{
/*optional, time range within the zone timeout threshold*/
        "@min": ,
        "@max":
    },
    "timeout":{
/*optional, zone timeout range*/
        "@min": ,
        "@max":
    },
    "relateDetector":{
/*required, whether to support linking to detectors*/
        "@opt": "true,false"
    },
    "detectorSeq":{
/*optional, length of detector serial No.*/
        "@min": ,
        "@max":
```

```
    },
    "RelatedChan":{
        "relator":{
            /*required, channel linkage type: "host"-security control panel, "app"-
            application program*/
            "@opt":"host,app"
        },
        "cameraSeq":{
            /*optional, length of camera serial No.*/
            "@min": ,
            "@max":
        },
        "relatedChan":{
            /*required, linked channel No. range*/
            "@min": ,
            "@max":
        },
        "remoteRelatedChan":{
            /*optional, range of remotely linked channel No. (application linkage)*/
            "@min": ,
            "@max":
        },
        "linkageCameraName": {
            /*optional, object, name of the linked camera*/
            "@min": 1,
            /*optional, int, the maximum length, range:[1,64]*/
            "@max": 64
            /*optional, int, the maximum length, range:[1,64]*/
        }
    },
    "relay": ,
    /*optional, maximum number of relays that can be linked to the zone*/
    "maxCoordinateNum":32,
    /*optional, maximum number of coordinates that can be linked to the zone. At
    least three coordinates should be used for a zone*/
    "isSupportZonesOverlay":true,
    /*optional, whether to support zone overlay: "true"-yes, this node is not
    returned-no*/
    "isSptAddDetector":
    /*optional, boolean type, whether to support adding detectors*/
    "address":{
        /*optional, read-only, module address, wireless modules do not return this
        node*/
        "@opt":[1,2,3]
    },
    "linkageAddress":{
        /*optional, read-only, linked module address, wired modules do not return this
        node*/
        "@opt":[1,2,3]
    },
    "moduleChannel":{
        /*optional, module channel No.*/
```

```

        "@min": ,
        "@max":
    },
    "moduleType":{
/*optional, read-only, module type: "localWired"-local wired module,
"extendWired"-extended wired module, "localWireless"-local wireless module,
"extendWireless"-extended wireless module*/
        "@opt":["localWired", "extendWired", "localWireless", "extendWireless"]
    },
    "moduleStatus":{
/*optional, read-only, module status: "online", "offline", "heartbeatAbnormal"-
heartbeat exception*/
        "@opt":["online", "offline", "heartbeatAbnormal"]
    },
    "CheckTimeList":[{
/*optional, range list of detector offline duration, unit: hour. Different
detectors correspond to different offline duration ranges, so the offline
duration range should be listed separately for each detector supported by the
device*/
        "detectorType": "",
/*detector type, refer to detectorType node for details*/
        "@min": ,
        "@max":
    }],
    "sensitivity":{
/*optional, zone sensitivity: 10-10 ms, 250-250 ms, 500-500 ms, 750-750 ms*/
        "@opt":[10,250,500,750]
    },
    "resistor":{
/*optional, EOL (End-of-Line) resistor: 2.2-2.2k, 3.3-3.3k, 5.6-5.6k, 8.2-8.2k*/
        "@opt":[2.2,3.3,5.6,8.2]
    },
    "tamperType":{
/*optional, tampering type: "disable", "normalOpen"-remain open, "normalClose"-
remain closed*/
        "@opt":["disable", "normalOpen", "normalClose"]
    },
    "zoneAttrib":{
/*optional, read-only, zone attribute: "wired", "wireless". If this node is not
returned, the default zone attribute is "wireless"*/
        "@opt":["wired", "wireless"]
    },
    "linkagePircamCapCfg":"true,false",
/*optional, boolean, whether it supports configuring pircam capture linkage*/
    "linkageFileName":{
/*optional, name length of the linked file*/
        "@min": ,
        "@max":
    },
    "doubleKnockEnabled": {
/*optional, object, whether to enable double knock, an alarm will be triggered
when the detector is triggered twice within the configured time of

```

```
doubleKnockTime*/
    "@opt": [true, false]
/*optional, array, subType:bool*/
    },
    "doubleKnockTime": {
/*optional, object, interval of double knock, unit: second, range: [5,600]*/
        "@min": 5,
/*optional, int, the minimum value*/
        "@max": 600
/*optional, int, the maximum value*/
    },
    "CrossZoneCfg": {
/*optional, object, cross-zone configuration*/
        "isAssociated": {
/*required, object, whether this zone has been linked with other zones: true-
yes, false-no*/
            "@opt": [true, false]
        },
        "supportAssociatedZone": {
/*optional, object, zones that can be linked*/
            "@min": 1,
/*optional, int, the minimum value*/
            "@max": 2,
/*optional, int, the maximum value*/
            "@size": 1
/*optional, int, the maximum number of zones that can be linked*/
        },
        "alreadyAssociatedZone": {
/*optional, object, zones that have been linked with other zones*/
            "@min": 1,
/*optional, int, the minimum value*/
            "@max": 2,
/*optional, int, the maximum value*/
            "@size": 1
/*optional, int, the maximum number of linked zones*/
        },
        "associateZoneCfg": {
/*optional, object, other zones that have been linked with this zone*/
            "@min": 1,
/*optional, int, the minimum value*/
            "@max": 2,
/*optional, int, the maximum value*/
            "@size": 1
/*optional, int, the maximum number of linked zones*/
        },
        "associateTime": {
/*optional, object, link time between zones*/
            "@opt": [10, 20, 30]
        },
        "supportLinkageChannelID": {
/*optional, object, No.s of the channels that can be linked with network
cameras*/
```

```
        "@min": 1,
/*optional, int, the minimum value, range:[1,4]*/
        "@max": 4,
/*optional, int, the maximum value,, range:[1,4]*/
        "@size": 1
/*optional, int, the maximum number of network camera channels, range:[1,4]*/
    },
    "alreadyLinkageChannelID": {
/*optional, object, No.s of the channels that have been linked with network
cameras*/
        "@min": 1,
/*optional, int, the minimum value, range:[1,4]*/
        "@max": 4,
/*optional, int, the maximum value,, range:[1,4]*/
        "@size": 1
/*optional, int, the maximum number of linked network camera zones, range:
[1,4]*/
    },
    "linkageChannelID": {
/*optional, object, No.s of the channels to be linked with network cameras*/
        "@min": 1,
/*optional, int, the minimum value, range:[1,4]*/
        "@max": 4,
/*optional, int, the maximum value,, range:[1,4]*/
        "@size": 1
/*optional, int, the maximum number of network camera channels to be linked,
range:[1,4]*/
    }
},
    "armNoBypassEnabled": {
/*optional, object, whether to enable no bypass during arming*/
        "@opt": [true, false]
    },
    "AlarmSoundInterlink": {
/*optional, object, alarm linkage, which is supported by zones with R3 smoke
detectors, heat detectors, or CO detectors*/
        "supportLinkageZones": {
/*optional, object, zones that support alarm linkage, this node is required
when the node AlarmSoundInterlink exists*/
            "@size": 1,
/*optional, int, number of zones that support alarm linkage*/
            "@min": 1,
/*optional, int, the minimum value*/
            "@max": 2
/*optional, int, the maximum value*/
        },
        "linkageZones": {
/*optional, object, zones with alarm linkage configured, this node is required
when the node AlarmSoundInterlink exists*/
            "@size": 1,
/*optional, int, the number of zones with alarm linkage configured*/
            "@min": 1,
```

```
/*optional, int, the minimum value*/
    "@max": 2
/*optional, int, the maximum value*/
    }
},
    "RelatedPIRCAM": {
/*optional, object, linked pircam information, either this node or RelatedChan
takes effect, that is, a zone cannot be linked with both network camera and
pircam, and one pircam can only be linked with one zone*/
    "supportLinkageZones": {
/*optional, object, zones that can be linked with pircam; this node is required
when "RelatedPIRCAM" exists*/
        "@size": 1,
/*optional, int, the maximum number of zones that can be linked with pircam*/
        "@min": 1,
/*optional, int, the minimum value*/
        "@max": 2
/*optional, int, the maximum value*/
    },
    "linkageZone": {
/*optional, object, zones that have been linked with pircam; this node is
required when "RelatedPIRCAM" exists*/
        "@size": 1,
/*optional, int, the maximum number of zones that have been linked with pircam*/
        "@min": 1,
/*optional, int, the minimum value*/
        "@max": 2
/*optional, int, the maximum value*/
    },
    "linkagePIRCAMName": {
/*optional, object, name of the linked pircam*/
        "@min": 1,
/*optional, int, the minimum value, range:[1,64]*/
        "@max": 64
/*optional, int, the maximum value, range:[1,64]*/
    }
},
    "reportSendDelayTime": {
/*optional, object, delay time of uploading the alarm to ARC, unit: second;
this node is valid for delayed zones only*/
        "@min": 10,
/*optional, int, the minimum value*/
        "@max": 3600
/*optional, int, the maximum value*/
    },
    "finalDoorExitEnabled": {
/*optional, object, whether to enable Final Door Exit function: true-enable,
false-disable, If Final Door Exit is enabled on a door magnetic contact (a
detector), the area will be armed immediately after the magnetic contact
detects door opening and door closing. If disabled, the area has to wait until
a fixed countdown is over before being armed*/
        "@opt": [true, false]
```



```

    },
    "timeRestartEnabled": {
/*optional, boolean, whether to enable restarting time for delayed zone, which
is valid after exit delay is configured*/
        "@opt": [true, false]
    },
    "armMode": {
/*optional, string, arming mode, which is valid for public zones*/
        "@opt": ["and", "or"]
    }
},
"WiredZonesCap":{
/*wired zone capability*/
    "id":{
/*required, zone No. range*/
        "@min": ,
        "@max":
    },
    "zoneName":{
/*optional, zone name length*/
        "@min": ,
        "@max":
    },
    "ZoneTypeList":[{
        "ZoneType":{
            "type":"","
/*required, supported zone types: "Instant"-instant zone, "Delay"-delayed zone,
"Follow"-follow zone, "Perimeter"-perimeter zone, "24hNoSound"-24H silent zone,
"Emergency"-panic zone, "Fire"-fire zone, "Gas"-gas zone, "Medical"-medical
zone, "Timeout"-timeout zone, "Non-Alarm"-disabled zone, "EarlyWarning"-early-
warning zone, "Warning"-warning zone, "Shield"-shielded zone, "Key"-key zone*/
            "sptProp":
/*supported zone properties: "delayTime"-delay, "awayBypass"-stay arming
bypass, "chime"-doorbell, "silent"-mute, "timeout"-timeout*/
        }
    ]},
    "detectorType":{
/*required, "panicButton"-panic button, "magneticContact"-door magnetic contact
detector, "smokeDetector"-smoke detector, "activeInfraredDetector"-active IR
detector, "passiveInfraredDetector"-PIR detector, "glassBreakDetector"-glass-
break detector, "vibrationDetector"-shock detector, "dualTechnologyPirDetector"-
dual-technology motion detector, "tripleTechnologyPirDetector"-triple-
technology detector, "humidityDetector"-humidity detector,
"temperatureDetector"-temperature detector, "combustibleGasDetector"-gas
detector, "dynamicSwitch"-dynamic switch, "controlSwitch"-control switch,
"smartLock"-smart lock, "waterDetector"-water detector, "displacementDetector"-
displacement detector, "singleInfraredDetector"-door contact,
"singleZoneModule"-wireless single input expander, "curtainInfraredDetector"-IR
curtain detector, "pircam"-pircam detector (detector equipped with camera),
"slimMagneticContact", "indoorDualTechnologyDetector", "magnetShockDetector",
"waterLeakDetector", "wirelessSmokeDetector",
"wirelessGlassBreakDetector", "wirelessTemperatureHumidityDetector",

```

```
"wirelessHeatDetector", "wirelessCODetector", "wirelessPIRCeilingDetector",
"wirelessExternalMagnetDetector", "wirelessPIRCurtainDetector",
"wirelessDTAMCurtainDetector", "other", "tamperDetector"(tamper-proof
detector) */

"@opt": "panicButton, magneticContact, smokeDetector, activeInfraredDetector, passive
InfraredDetector, glassBreakDetector,
vibrationDetector, dualTechnologyPirDetector, tripleTechnologyPirDetector, humidity
Detector, temperatureDetector, combustibleGasDetector, dynamicSwitch, controlSwitch,
smartLock, waterDetector, displacementDetectorsingleInfraredDetector, singleZoneMod
ule, curtainInfraredDetector, pircam, slimMagneticContact, indoorDualTechnologyDetec
tor, magnetShockDetector, waterLeakDetector, wirelessSmokeDetector,
wirelessGlassBreakDetector, wirelessTemperatureHumidityDetector,
wirelessHeatDetector, wirelessCODetector, wirelessPIRCeilingDetector,
wirelessExternalMagnetDetector, wirelessPIRCurtainDetector,
wirelessDTAMCurtainDetector, other, tamperDetector"
    },
    "subSystemNo": {
/*required, partition No. range*/
        "@min": ,
        "@max":
    },
    "delayTime": {
/*optional, select delay 1 or delay 2 (which is in the timer configuration of
the partition, and corresponds to "entryDelay1" and "entryDelay2") */
        "@opt": "1, 2"
    },
    "stayAwayEnabled": "true, false",
/*optional, whether to enable stay arming bypass*/
    "chimeEnabled": "true, false",
/*optional, whether to enable doorbell*/
    "chimeWarningType": {
/*doorbell tone type: "single"-single tone, "continuous"-continuous tone, this
node is valid only when chimeEnabled is "true"*/
        "@opt": "single, continuous"
    },
    "silentEnabled": "true, false",
/*optional, whether to support enabling mute*/
    "timeoutLimit": {
/*optional, configuration of zone timeout threshold*/
        "@opt": "true, false"
    },
    "timeoutType": {
/*optional, timeout type, "tigger"-triggering timed out, "recover"-restoring
timed out*/
        "@opt": "tigger, recover"
    },
    "limitTimeout": {
/*optional, time range within the zone timeout threshold*/
        "@min": ,
        "@max":
    },
    },
```

```
"timeout":{
/*optional, zone timeout range*/
    "@min": ,
    "@max":
},
"relateDetector":{
/*required, whether to support linking to detectors*/
    "@opt":"true,false"
},
"detectorSeq":{
/*optional, length of detector serial No.*/
    "@min": ,
    "@max":
},
"RelatedChan":{
    "relator":{
/*required, channel linkage type: "host"-security control panel, "app"-
application program*/
        "@opt":"host,app"
    },
    "cameraSeq":{
/*optional, length of camera serial No.*/
        "@min": ,
        "@max":
    },
    "relatedChan":{
/*required, linked channel No. range*/
        "@min": ,
        "@max":
    },
    "remoteRelatedChan":{
/*optional, range of remotely linked channel No. (application linkage)*/
        "@min": ,
        "@max":
    },
    "linkageCameraName": {
/*optional, object, name of the linked camera*/
        "@min": 1,
/*optional, int, the minimum value, range:[1,64]*/
        "@max": 64
/*optional, int, the maximum value, range:[1,64]*/
    }
},
"isSptAddDetector": ,
optional, boolean type, whether to support adding detectors
"address":{
/*optional, read-only, module address, wireless module does not return this
node*/
    "@opt":[1,2,3]
},
"linkageAddress":{
/*optional, read-only, linked module address, this node can only be configured
```

```
by wireless modules*/
    "@opt": [1, 2, 3]
  },
  "moduleChannel": {
    /*optional, module channel No.*/
    "@min": ,
    "@max":
  },
  "moduleType": {
    /*optional, read-only, module type: "localWired"-local wired module,
    "extendWired"-extended wired module, "localWireless"-local wireless module,
    "extendWireless"-extended wireless module*/
    "@opt": ["localWired", "extendWired", "localWireless", "extendWireless"]
  },
  "moduleStatus": {
    /*optional, read-only, module status: "online", "offline", "heartbeatAbnormal"-
    heartbeat exception*/
    "@opt": ["online", "offline", "heartbeatAbnormal"]
  },
  "CheckTimeList": [{
    /*optional, range list of detector offline duration, unit: hour. Different
    detectors correspond to different offline duration ranges, so the offline
    duration range should be listed separately for each detector supported by the
    device*/
    "detectorType": "",
    /*detectory type, refer to <detectorType> node for details*/
    "@min": ,
    "@max":
  }],
  "sensitivity": {
    /*optional, zone sensitivity: 10-10 ms, 250-250 ms, 500-500 ms, 750-750 ms*/
    "@opt": [10, 250, 500, 750]
  },
  "resistor": {
    /*optional, EOL (End-of-Line) resistor: 2.2-2.2k, 3.3-3.3k, 5.6-5.6k, 8.2-8.2k*/
    "@opt": [2.2, 3.3, 5.6, 8.2]
  },
  "tamperType": {
    /*optional, tampering type: "disable", "normalOpen"-remain open, "normalClose"-
    remain closed*/
    "@opt": ["disable", "normalOpen", "normalClose"]
  },
  "zoneAttrib": {
    /*optional, read-only, zone attribute: "wired", "wireless". If this node is not
    returned, the default zone attribute is "wireless"*/
    "@opt": ["wired", "wireless"]
  },
  "method": {
    /*optional, methods supported by the function: "put"-edit, "getAll"-get all,
    "get"-get one. If this node is not returned, it indicates that the device
    supports "put" and "getAll" methods by default*/
    "@opt": ["put", "getAll", "get"]
  }
}
```

```
    },
    "linkagePircamCapCfg":"true,false",
    /*optional, boolean, whether it supports configuring pircam capture linkage*/
    "linkageFileName":{
    /*optional, name length of the linked file*/
        "@min": ,
        "@max":
    },
    "extendZoneNo":{
    /*optional, integer, read-only, extended zone No. after enabling double-
    detector zone. This field is valid when the configuration URI is called to
    configure parameters of the linked zone. For example, if the double-detector
    zone has been enabled for the linked zone (zone 1) and the extended zone No. of
    zone 1 is 9, when the configuration URI is called to configure parameters of
    zone 1, id in the message should be set to 1, extendZoneNo should be set to 9,
    and relatedZoneNo is invalid at this point*/
        "@min": ,
        "@max":
    },
    "extendZoneName": {
    /*optional, object, name of the extended zone*/
        "@min": 1,
    /*required, int, the minimum value, range:[1,32]*/
        "@max": 32
    /*required, int, the maximum value, range:[1,32]*/
    },
    "relatedZoneNo":{
    /*optional, integer, read-only, linked zone No. This field is valid when the
    configuration URI is called to configure parameters of the extended zone. For
    example, if the double-detector zone has been enabled for the linked zone (zone
    1) and the extended zone No. of zone 1 is 9, when the configuration URI is
    called to configure parameters of zone 9, id in the message should be set to 9,
    relatedZoneNo should be set to 1, and extendZoneNo is invalid at this point*/
        "@opt":[1,2,3]
    },
    "relatedZoneName": {
    /*optional, object, name of the linked zone*/
        "@min": 1,
    /*optional, int, the minimum value, range:[1,32]*/
        "@max": 32
    /*optional, int, the maximum value, range:[1,32]*/
    },
    "SupportedDoubleZoneNos": ,
    /*optional, array of integer, No. list of zones that support double-detector
    zone settings. For example, [1,2,3,4] indicates that zone 1, zone 2, zone 3,
    and zone 4 support double-detector zone settings*/
    "doubleZoneCfgEnable":"true,false"
    /*optional, boolean, whether to enable double-detector zone. If this field is
    "true", it indicates that the double-detector zone has been enabled for the
    zone and two detectors will be linked. For the upper level, two detectors
    require two zone numbers*/
    "linkagePircamCapCfg": "true,false",
```

```
/*optional, string, picture capture configuration of pircam*/
    "linkageFileName ": {
/*optional, object, length of the linked file name*/
        "@min": 1,
/*optional, int, the minimum value*/
        "@max": 1
/*optional, int, the maximum value*/
    },
    "doubleKnockEnabled": {
/*optional, object, whether to enable double knock*/
        "@opt": [true, false]
/*optional, array of boolean*/
    },
    "doubleKnockTime": {
/*optional, object, interval of double knock, unit: second, range: [5,600]*/
        "@min": 5,
/*optional, int, the minimum value*/
        "@max": 600
/*optional, int, the maximum value*/
    },
    "CrossZoneCfg": {
/*optional, object, cross-zone configuration*/
        "isAssociated": {
/*required, object, whether this zone has been linked with other zones: true-
yes, false-no*/
            "@opt": [true, false]
/*required, array of boolean*/
        },
        "supportAssociatedZone": {
/*optional, object, zones that can be linked*/
            "@min": 1,
/*optional, int, the minimum value*/
            "@max": 2,
/*optional, int, the maximum value*/
            "@size": 1
/*optional, int, the maximum number of zones that can be linked*/
        },
        "alreadyAssociatedZone": {
/*optional, object, zones that have been linked with other zones*/
            "@min": 1,
/*optional, int, the minimum value*/
            "@max": 2,
/*optional, int, the maximum value*/
            "@size": 1
/*optional, int, the maximum number of linked zones*/
        },
        "associateZoneCfg": {
/*optional, object, other zones that have been linked with this zone*/
            "@min": 1,
/*optional, int, the minimum value*/
            "@max": 2,
/*optional, int, the maximum value*/
        }
    }
}
```

```
        "@size": 1
/*optional, int, the maximum number of linked zones*/
    },
    "associateTime": {
/*optional, object, link time between zones*/
        "@opt": [10, 20, 30]
/*optional, int*/
    },
    "supportLinkageChannelID": {
/*optional, object, No.s of the channels that can be linked with network
cameras*/
        "@min": 1,
/*optional, int, the minimum value, range:[1,4]*/
        "@max": 4,
/*optional, int, the maximum value,, range:[1,4]*/
        "@size": 1
/*optional, int, the maximum number of network camera channels, range:[1,4]*/
    },
    "alreadyLinkageChannelID": {
/*optional, object, No.s of the channels that have been linked with network
cameras*/
        "@min": 1,
/*optional, int, the minimum value, range:[1,4]*/
        "@max": 4,
/*optional, int, the maximum value,, range:[1,4]*/
        "@size": 1
/*optional, int, the maximum number of linked network camera zones, range:
[1,4]*/
    },
    "linkageChannelID": {
/*optional, object, No.s of the channels to be linked with network cameras*/
        "@min": 1,
/*optional, int, the minimum value, range:[1,4]*/
        "@max": 4,
/*optional, int, the maximum value,, range:[1,4]*/
        "@size": 1
/*optional, int, the maximum number of network camera channels to be linked,
range:[1,4]*/
    }
},
    "armNoBypassEnabled": {
/*optional, object, whether to enable no bypass during arming*/
        "@opt": "true,false"
/*optional, string*/
    },
    "isSupportSignalTest": true,
/*optional, boolean, whether it supports signal strength detection, if the node
is not returned or the value is false, it indicates "no"*/
    "isSupportZoneTest": true,
/*optional, boolean, whether it supports zone detection, if the node is not
returned or the value is false, it indicates "no"*/
    "RelatedPIRCAM": {
```

```
/*optional, object, linked pircam information, either this node or RelatedChan
takes effect, that is, a zone cannot be linked with both network camera and
pircam, and one pircam can only be linked with one zone*/
    "supportLinkageZones": {
/*optional, object, zones that can be linked with pircam; this node is required
when "RelatedPIRCAM" exists*/
        "@size": 1,
/*optional, int, the maximum number of zones that can be linked with pircam*/
        "@min": 1,
/*optional, int, the minimum value*/
        "@max": 2
/*optional, int, the maximum value*/
    },
    "linkageZone": {
/*optional, object, zones that have been linked with pircam; this node is
required when "RelatedPIRCAM" exists*/
        "@size": 1,
/*optional, int, the maximum number of zones that have been linked with pircam*/
        "@min": 1,
/*optional, int, the minimum value*/
        "@max": 2
/*optional, int, the maximum value*/
    },
    "linkagePIRCAMName": {
/*optional, object, name of the linked pircam*/
        "@min": 1,
/*optional, int, the minimum value, range:[1,64]*/
        "@max": 64
/*optional, int, the maximum value, range:[1,64]*/
    }
},
    "reportSendDelayTime": {
/*optional, object, delay time of uploading the alarm to ARC, unit: second;
this node is valid for delayed zones only*/
        "@min": 10,
/*optional, int, the minimum value*/
        "@max": 3600
/*optional, int, the maximum value*/
    },
    "timeRestartEnabled": {
/*optional, object, whether to enable resetting delay time for exit; this node
is valid only for delayed zones*/
        "@opt": [true, false]
/*optional, array of boolean*/
    },
    "notRelatedZoneNo": {
/*optional, object, No. of the unlinked zones*/
        "@size": 1,
/*optional, int, the maximum number of unlinked zones*/
        "@min": 1,
/*optional, int, the minimum value*/
        "@max": 2
    }
}
```



```
/*optional, int, the maximum value*/
    },
    "modifiedZoneNo": {
/*optional, object, Zone No. after it is modified*/
        "@min": 0,
/*optional, int, the minimum value*/
        "@max": 255
/*optional, int, the maximum value*/
    },
    "finalDoorExitEnabled": {
/*optional, object, whether to enable Final Door Exit function: true-enable,
false-disable, If Final Door Exit is enabled on a door magnetic contact (a
detector), the area will be armed immediately after the magnetic contact
detects door opening and door closing. If disabled, the area has to wait until
a fixed countdown is over before being armed*/
        "@opt": [true, false]
/*optional, array of boolean*/
    },
    "swingerLimitActivation": {
/*optional, object, allowed times of triggers*/
        "@min": 0,
/*optional, int, the minimum value*/
        "@max": 10
/*optional, int, the maximum value*/
    },
    "detectorContactMode": {
/*optional, object, detector mode: "NO"-always open, "NC"-always closed,
"rollerShutter"(customized)*/
        "@opt": ["NO", "NC", "rollerShutter"]
/*optional, array of string*/
    },
    "impulseCountTime": {
/*optional, object, pulse count time, unit:second*/
        "@opt": [10, 20, 30, 40, 50, 60]
/*optional, array of int*/
    },
    "impulsesBeforeAlarm": {
/*optional, object, number of pulses before alarm*/
        "@opt": [2, 4, 6]
/*optional, array of int, range:[2,30]*/
    },
    "detectorInputMode": {
/*optional, object, detector input mode: "pulse", "latch";this node is valid
when the value of detectorContactMode is "NO" or "NC"*/
        "@opt": ["pulse", "latch"]
    },
    "pulseDuration": {
/*optional, object, pulse interval, unit: second, this node is valid when the
value of detectorInputMode is "pulse"*/
        "@min": 5,
/*optional, int, the minimum value*/
        "@max": 600
```

```
/*optional, int, the maximum value*/
},
"detectorTamperMode": {
/*optional, object, detector tamper mode: "NO"-always enabled, "NC"-always
disabled*/
"@opt": ["NO", "NC"]
},
"antiMasking": {
/*optional, object, anti-masking mode: "NO"-always enabled, "NC"-always
disabled*/
"@opt": ["NO", "NC"]
},
"detectorContactModeList": [
/*optional, array, list of detector contact modes*/
{
"detectorContactMode": "NO",
/*optional, enum, detector contact mode: "NO" (always open), "NC" (always
closed), "A" (custom), "EOL", "DEOL-NO", "DEOL-NC"*/
"pulseSensitivity": {
/*optional, object, pulse sensitivity, unit: ms, when detectorContactMode is
"NO" or "NC", the values can be 30,100,1000; when detectorContactMode is "EOL",
"DEOL-NO", or "DEOL-NC", the values can be 10,100,250,500,750, and the default
value is 250*/
"@opt": [10, 30, 100, 250, 500, 750, 1000]
/*optional, array of int, unit: ms*/
}
}
],
"alarmResistance": {
/*optional, object, alarm resistance, unit: k, this node is valid when
detectorContactMode is "EOL", "DEOL-NO", or "DEOL-NC"*/
"@opt": [1.0, 2.2, 4.7, 8.2]
/*optional, array of float*/
},
"tamperResistance": {
/*optional, object, tamper resistance, unit: k, this node is valid when
detectorContactMode is "DEOL-NO", "DEOL-NC"*/
"@opt": [1.0, 2.2, 4.7, 8.2]
/*optional, array of float*/
},
"accessModuleType": {
/*optional, object, access module type: "transmitter"-transmitter peripheral,
"multiTransmitter", "localTransmitter", "localZone", "keypad"*/
"@opt": ["transmitter", "multiTransmitter", "localTransmitter",
"localZone", "keypad"]
},
"relatedAccessModuleID": {
/*optional, object, ID of linked access module*/
"@min": 0,
/*optional, int, the minimum value*/
"@max": 255
/*optional, int, the maximum value*/
```

```
    },
    "armMode": {
/*optional, object, arming mode, which is valid for public zone*/
        "@opt": ["and", "or"]
    },
    "AccessModuleNotSupportDetectorList": [
/*optional, array, list of detectors that are not supported by the access
module*/
        {
            "accessModuleType": "transmitter",
/*optional, enum, access module type*/
            "notSupportDetectorType": {
/*optional, object, detector type that is not supported*/
                "@opt": ["tamperDetector"]
/*optional, array, "tamperDetector"*/
            }
        }
    ]
}
```

Remarks

- The "trigger" timeout type (**timeoutType**) indicates that the alarm will not be triggered until the configured timeout of detector is ended, e.g., for the regions that only the passing is allowed, but the loitering is not allowed, if you loiter for the certain time duration, the alarm will be triggered.
- The "recover" timeout type (**timeoutType**) indicates that the target should be detected by the detector continuously, if the detector cannot detect the target for the configured timeout, the alarm will be triggered, e.g., if the security is absent for the certain time duration, the alarm will be triggered.

A.3.196 JSON_ZoneSearch

JSON message about the result of zone status

```
{
  "ZoneSearch":{
    "searchID": "",
/*required, string type, search ID, which is used to confirm the upper-level
platform or system. If the platform or the system is the same one during two
searching, the search history will be saved in the memory to speed up next
searching*/
    "responseStatusStrg": "",
/*required, string type, search status: "OK"-searching completed, "NO MATCH"-no
matched results, "MORE"-searching for more results*/
    "numOfMatches": ,
/*required, integer32, number of returned results this time*/
    "totalMatches": ,
```

```
/*required, integer32, total number of matched results*/
    "ZoneList":[{
        "Zone":{
            "id": ,
/*required, integer type, zone No.*/
            "name": "",
/*optional, string type, zone name*/
            "status": "",
/*optional, string type, zone status: "notRelated"-not linked, "online",
"offline", "trigger", "breakDown"-fault, "heartbeatAbnormal"-heartbeat
exception*/
            "tamperEvident": ,
/*optional, boolean type, zone tampering status: "true"-tampered, "false"-not
tampered*/
            "shielded": ,
/*optional, boolean type, zone shielding status: "true"-shielded, "false"-not
shielded*/
            "bypassed": ,
/*optional, boolean type, whether the zone is bypassed: "true"-yes, "false"-no*/
            "armed": ,
/*required, boolean type, whether the zone is armed: "true"-yes, "false"-no*/
            "isArming": ,
/*optional, boolean type, whether the zone is armed, this node can only be set
to "true"*/
            "alarm": ,
/*optional, boolean type, whether the alarm is triggered in the zone: "true"-
yes, "false"-no*/
            "charge": "",
/*optional, string type, state of charge of the zone: "normal", "lowPower"-low
battery*/
            "signal": ,
/*optional, integer type, signal strength, it is between 0 and 255*/
            "subSystemNo ": ,
/*optional, integer type, partition No.*/
            "zoneAttrib": "",
/*optional, string, zone attribute: "wired", "wireless". If this node is not
returned, the default zone attribute is "wireless"*/
            "RelatedChanList":[{
/*optional, list of linked channel No.*/
                "RelatedChan":{
                    "relator": "",
/*required, string type, device linked to the channel when the alarm is
triggered*/
                    "cameraSeq": "",
/*optional, string type, camera serial No.*/
                    "relatedChan":
/*optional, integer type, linked channel No.*/
                }
            }],
            "detectorType": "test",
/*optional, string, type of the detector linked to the zone*/
            "model": "DS-PM1-O8-WE",
```

```
/*optional, enum, model, subType:string, "DS-PM1-O8-WE", "DS-PM1-O2-WE"*/
    "zoneType": "Instant",
/*optional, string, zone type: "Instant"-instant zone, "Delay"-delay zone,
"Follow"-follow zone, "Perimeter"-perimeter zone, "24hNoSound"-24-hour silent
zone, "Emergency"-panic zone, "Fire"-fire zone, "Gas"-gas zone, "Medical"-
medical zone, "Timeout"-timeout zone, "Non-Alarm"-disabled zone, "Key"-key
zone*/
    "InputList": [
/*optional, array, list of input status*/
        {
            "id": 1,
/*required, int, input ID*/
            "enabled": true,
/*required, boolean, whether it is enabled*/
            "mode": "NO"
/*optional, enum, input type: "rolling shutter", "NC" (always closed), "NO"
(always open)*/
        }
    ],
    "humidity": 10,
/*optional, int, humidity, the value is between 10% and 90%*/
    "healthStatus": "normal",
/*optional, string, read-only, health status: "normal", "fault"*/
    "antiMaskingEnabled": true,
/*optional, boolean, read-only, whether to enable anti-masking: true-enable,
false-disable*/
    "mountingType": "wall",
/*optional, string, read-only, mounting type: "wall", "ceiling"*/
    "magnetOpenStatus": true,
/*optional, boolean, whether the magnetic contact is open: true (open), false
(closed)*/
    "devIndex": "test",
/*optional, string, device ID, the maximum length is 64 bytes*/
    "devName": "test",
/*optional, string, device name, the maximum length is 64 bytes*/
    "isAvailable": true,
/*optional, boolean, whether the partition is available: true (default), false*/
    "isBypassedAvailable": true,
/*optional, boolean, whether bypass is configurable: true (yes), false (no). By
default, it is configurable*/
    "version": "test",
/*optional, string, detector version No., the maximum length is 32 bytes*/
    "pirCamConnected": true,
/*optional, boolean, whether the outdoor tri-tech detector and PIR camera are
connected: true (yes), false (no)*/
    "accessModuleType": "transmitter",
/*optional, enum, access module type: "transmitter", "localTransmitter",
"multiTransmitter", "localZone", "keypad"*/
    "relatedAccessModuleID": 1,
/*optional, int, linked access module ID*/
    "address": 254,
/*optional, int, wired (extended) module address, this node works with
```

```
accessModuleType*/
    "deviceNo": 1
/*optional, int, device ID, range:[1,1000]*/
    }
    ]]
    }
}
```

A.3.197 XML_AlarmHostAbility

AlarmHostAbility capability message in XML format

```
<AlarmHostAbility version="2.0"> <!--req, Description for Capability Set of
Network Security Control Panel -->
    <keyboardNo min="" max=""/>
    <!--req,keyboard No.-->
    <keyboardAddr min="" max=""/>
    <!--req,keyboard address-->
    <operatorUserNo min="" max=""/>
    <!--req,operator user number of keyboard -->
    <operatorPasswordLen min="" max=""/>
    <!--req,password length-->
    <operatorPassword opt="0,1,2,3,4,5,6,7,8,9,F"/>
    <!--req,the supported characters of operator password-->
    <subSystemPermission opt="0,1,2,3,4,5 "/>
    <!--req, 0- single arming, no bypass; 1- single disarming, no bypass; 2-
arming/disarming, no bypass; 3- single arming, bypass; 4- single disarming,
bypass; 5- arming/disarming, bypass-->
    <!--Note: display all capabilities of device without return capability.
Original capability is: 1- single arming, no arming report, no bypass; 2-
single disarming, no arming report, no bypass;
    3- arming/disarming, no arming report, no bypass; 4- single arming, arming
report, no bypass; 5- single disarming, arming report, no bypass;
    6- arming/disarming, arming report, no bypass; 7- single arming, no arming
report, bypass; 8- single disarming, no disarming report, bypass;
    9- arming/disarming, no arming report, bypass; 10- single arming, arming
report, bypass; 11- single disarming, arming report, bypass; 12- arming/
disarming, arming report, bypass-->
    <installerNo min="" max=""/>
    <!--req,installer number-->
    <installPasswordLen min="" max=""/>
    <!--req,password length of installer password -->
    <installPassword min="" max=""/>
    <!--req, the supported string of installer password-->
    <Zone>
        <ZoneConfig>
            <enabled>true</enabled>
            <!--req,support parameter configuration of zone (alarm input parameter
configuration) -->
            <delayInParam>true</delayInParam>
            <!--req, true,the client set dwell time by dwParam of
```

```
NET_DVR_ALARMIN_PARAM, rotating ring alarm host and self-service alarm host
using this dwell time configuration;
    false means set via the wEnterDelay, wExitDelay of
NET_DVR_ALARMSUBSYSTEMPARAM-->
    <detectorType
opt="panicButton,magneticContact,smokeDetector,activeInfraredDetector,passiveInf
raredDetector,glassBreakDetector,vibrationDetector,dualTechnologyPirDetector,tri
pleTechnologyPirDetector,humidityDetector,temperatureDetector,combustibleGasDete
cto,dynamicSwitch,controlSwitch,otherDetector"/>
    <!--req, supported detector type: emergency switch, magnetic switch,
smoke detector, initiative infrared detector, passive infrared detector, glass
broken detector, vibration detector, dual technology motion detector, triple
technology PIR detector, humidity detector, temperature detector, combustible
gas detector, dynamic switch, control switch, other detector-->
    <zoneType opt="instantZone,
24hourAudibleZone,delayZone,interiorWithDelayZone,keyswitchZone,
    supervisedFireZone,perimeterZone,24hourSlientZone,24hourAuxiliaryAlarmIn,
24hourVibrationAlarmIn, timeOutZone,

doorEmergencyOpenProtectionZone,doorEmergencyShutdownProtectionZone,disable"/>
    <!--,Zone types supported by device-->
    <uploadAlarmRecoveryReport>true</uploadAlarmRecoveryReport>
    <!--req,Upload alarm recovery report configuration -->
    <zoneDelayTime min="" max=""/>
    <!--req, delay time of delay zone-->
    <sensitivity opt="10ms,250ms,500ms,750ms"/>
    <!--,sensitivity-->
    <arrayBypass>true<!--req, support zone bypass configuration --></
arrayBypass>
    <moduleStatus attri="readonly" opt="offline,online"/>
    <!--module status-->
    <moduleAddress min="" max=""/>
    <!--module address-->
    <moduleChannel>true</moduleChannel>
    <!--module channel-->
    <moduleType opt="localZone, 1zoneExpander,2zoneExpander,8ZoneExpander,
8sensorZoneExpander,1ZoneAndTrigger,1DoorController,2DoorsController,
4DoorsController"/>
    <!--supported arming region type-->
    <zoneNo attri="readonly" min="" max=""/>
    <!--zone ID, get only-->
    <subsystemNo attri="readonly">1</subsystemNo>
    <!--subsystem of zone, get only-->
    <InDelayTime min="0" max="255"/>
    <!--req, enter delay time, unit:s-->
    <OutDelayTime min="0" max="255"/>
    <!--req, exit delay time, unit:s-->
    <alarmType opt="open,close" default="open"/>
    <!--req, alarm type, open- normally open, close- normally closed-->
    <zoneResistor opt="2.2,4.7 ,5.6 ,8.2 ,manual"/>
    <!--req, zone resistor, unit: kilo-ohm)-->
    <zoneResistorManual min="0" max="10"/>
```

```

    <!--req, custom zone resistor, accurate to one decimal place, unit: kilo-
ohm-->
    <timeout min="" max=""/><!--opt, Timeout, unit: s-->
    <timeoutRange opt="1-599s,1-65535s"/><!--opt, Timeout range-->
    <detectorSerialNo min="0" max="9"/><!--req, xs: string, Detector serial
No.-->
    <zoneSignalType opt="0,1"/><!--req, Zone signal transmission type: 0-
Wired Zone, 1- Wireless Zone-->
    <enableDetectorTypeCfg>true</enableDetectorTypeCfg><!--req, Whether the
detector type is matched.-->
    <zoneTypeWithDelayCfg opt="instantZone,
24hourAudibleZone,delayZone,interiorWithDelayZone,keyswitchZone,supervisedFireZo
ne,perimeterZone,24hourSilentZone,24hourAuxiliaryAlarmIn,
24hourVibrationAlarmIn,disable"/><!--, 支持配置延迟防区的类型, 反向能力, 当不返回按照以
前默认处理-->
    </ZoneConfig>
    <ZoneTamperConfig>
    <!--req,arming region anti-tamper-->
    <tamperType opt="0,1,2"/>
    <!--req, 0-Null,1- Remain open,2- Remain closed-->
    <uploadAlarmRecoveryReport>true</uploadAlarmRecoveryReport>
    <!--req, upload alarm recovery report configuration-->
    <associateAlarmOut min="0" max="512"/>
    <!--req,zone linked trigger-->
    <associateSirenOut min="0" max="8"/>
    <!--req,zone linked siren output-->
    <tamperResistor opt="2.2,4.7 ,5.6 ,8.2 ,manual"/>
    <!--req, anti-tamper resistor, unit: kilo-ohm-->
    <tamperResistorManual min="0" max="10"/>
    <!--req, custom anti-tamper resistor, accurate to one decimal place,
unit: kilo-ohm-->
    <supportZoneNo min="" max=""/>
    <!--req, anti-tamper supported arming region-->
    </ZoneTamperConfig>

    <GetZoneList>
    <enabled>true</enabled>
    <!--req, zone list supported-->
    </GetZoneList>

    <ZoneArmDisarm>
    <enabled>true</enabled>
    <!--req, support zone arming and disarming-->
    </ZoneArmDisarm>

    <ZoneGroupBypass>
    <enabled>true</enabled>
    <!--req, support zone by-path-->
    </ZoneGroupBypass>
    <associateLampOut min="0" max="8"/>
    <!--req,zone linked alarm lamp output-->
    <emergencyCallZone opt="3"/>

```



```
<!--req,emergency arming region No.-->
<consultZone opt="4"/>
<!--req, service inquiry arming region No.-->
</Zone>

<Trigger>
  <TriggerConfig>
    <enabled>true</enabled>
    <!--req, support parameter configuration of trigger(parameter
configuration of alarm input)-->
    <TriggerIndex>attri=readonly,true</TriggerIndex>
    <ModuleType opt="localTrigger,4-wayTrigger,8-wayTrigger,1ZoneAndTrigger,
32Trigger,ger" />-->
    <ModuleAddress min="1" max="253" />
    <ModuleChan>true</ModuleChan>
    <outputDelay min="1" max="65535"/>
    <!--req, time of duration of alarm input, uint: s -->
    <workMode opt="linkage,dynamic"/>
    <!--req, link, dynamic-->
    <alarmOutMode opt="pulseMode,nonePulseMode"/>
    <!--req, pulse, non-pulse-->
    <timeOn min="1" max="60"/>
    <!--req, open time 1-60-->
    <timeOff min="1" max="60"/>
    <!--req, close time 1-60-->
  </TriggerConfig>

  <TriggerControl>
    <enabled>true</enabled>
    <!--req,support trigger control (alarm output control)-->
  </TriggerControl>

  <GetTriggerList>
    <enabled>true</enabled>
    <!--req,support getting trigger list-->
  </GetTriggerList>
</Trigger>

<AutoRegisterExternalModule>
  <enabled>true</enabled>
  <!--req, support automatic register circumscribed mode-->
</AutoRegisterExternalModule>
<AutoSearchExternalModule>
  <enabled>true</enabled>
  <!--req, support automatic searching circumscribed mode-->
</AutoSearchExternalModule>

<moduleNo min="" max=""/>
<!--req,Module No.-->
<ModuleConfig>
  <enabled>true</enabled>
  <!--req, module configuration-->
```

```

<moduleType opt="zone,trigger,keyBoard"/>
<!--req,zone, trigger,keyboard-->
<ModuleInfo>
    <enable>true</enable>
</ModuleInfo>
<!--req, module information-->
<deviceVersionInfo>true</deviceVersionInfo >
<!--req, device version information-->
</ModuleConfig>
<isNotSupportZoneAddrEdit>true</isNotSupportZoneAddrEdit>
<!-- No return means the field can be exited. Otherwise, it cannot be
edited.-->
<isNotSupportTriggerAddrEdit>true</isNotSupportTriggerAddrEdit >
<!--No return means the field can be exited. Otherwise, it cannot be edited.--
>

<assistantControl opt="electricLock, mobileGate,audioOut,siren,alarmLamp"/>
<!--req,aux function control, electric lock, mobile door, audio output,
siren, alarm lamp-->

<GatewayConfig>
    <enabled>true</enabled><!--, support access control parameter
configuration-->
    <delayTime min="" max=""/><!--, delay time, uint: s. the door will
automatic close when the door opened delayTime -->
    <enabledConfig>true</enabledConfig><!--req, support enable /disable
entrance guard parameter-->
    <enabledLocal>true</enabledLocal><!--req,support local enable /disable
entrance guard parameter-->
    <lockWorkMode opt="setUpEnabled,interruptEnabled"/><!--req, electric lock
mode, enable when the power on, else interrupt-->
</GatewayConfig>

<SirenConfig>
    <enabled>true</enabled>
    <!--req,support siren parameter configuration-->
    <outputDelay min="1" max="65535"/>
    <!--req, output delay time of warning signal, uint: s -->
    <sirenNameLength min="" max=""/>
    <!--support name configuration of siren-->
    <overallEventTriggerSirenOn
opt="tamperAlarm,overallKeypadEmergencyAlarm,ACPower,lowBatteryVoltage,phoneOffLi
ne,wireNetWorkException,wirelessNetWorkException,keyBoard485Break"/>
    <!--Enable siren linkage output of global event, host tampering, panic
alarm of global keyboard, AC power off, low battery voltage, call dropped,
wired network exception,wireless network exception, keyboard 485 disconnected--
>

    <overallEventTriggerSirenOff opt=""/>
    <!--enable siren linkage of global event-->
    <subsystemEventTriggerSirenOn opt="emergencyKeypadAlarm,arm,disarm"/>
    <!--enable siren linkage of subsystem event, Emergency Alarm, arming and
disarming-->

```

```

    <subsystemEventTriggerSirenOff opt=""/>
    <!--enable siren linkage of subsystem event-->
</SirenConfig>

<SearchMainStatus>
    <enabled>true</enabled>
    <!--main status searching supported-->
    <mainStatusType opt="zoneArmed, zoneAlarm, trigger, zoneBypass,
subSystemArmed, zoneFault, zoneMemory, zoneTamper"/>
    <!--req,main status searching supported: arming status of defence area,
Emergency Alarm status of defence area, alarm status of trigger, by-pass status
of defence area, subsystem arming status, defence area malfunction status,
arming region alarm memory status, arming region anti-tamper status -->
    <enableSubSystem opt="disable,enable"/>
    <!--, whether to enable sub system-->
    <subSystemGuardType opt="armAway,armInstant,armStay"/>
    <!--, arming type: away arming, instant arming, stay arming-->
</SearchMainStatus>
<SearchOtherStatus>
    <enabled>true</enabled>
    <!--, support other status searching-->
    <otherStatusType
opt="siren,electricLock,alarmLamp,detectorPower,detectorConnection"/>
    <!--, support other status: siren status, electric lock, alarm lamp,
detector power status, detector online status -->
    <detectorPower min="0" max="100"/>
    <!--Detector power-->
    <detectorConnection opt="unregedit,disconnect,connect"/>
    <!--Detector connection status, unregedit-unregistered, disconnect-
Offline, connect-Online-->
</SearchOtherStatus>
<EnableConfig>
    <enabled>true</enabled>
    <!--req, support enabled configuration-->
    <enableType opt="audioOut, electricLock,mobileGate,siren"/>
    <!--req,enabled supported, audio output enable , electric lock enable ,
mobile door enable, warning signal enable -->
    <serialPurpose opt="cascade, PTZ"/>
    <!--req,serial port application: Cascade alarm host, PTZ control-->
</EnableConfig>

<TransparencySerial>
    <enabled>true</enabled>
    <!--req, support alarm host transparent channel-->
</TransparencySerial>

<Get485DeviceList>
    <enabled>true</enabled>
    <!--req, support getting 485 external devices list-->
</Get485DeviceList>
<Get485DeviceProtocolList>
    <enabled>true</enabled>

```

```
<!--req, support getting 485 external devices protocol list -->
</Get485DeviceProtocolList>
<RS485Config>
  <enabled>true</enabled>
  <!--req, support 485 parameter configuration, 485 interface of video alarm
host using the interface, for the 485 communication capability, you can refer
to "RS232 and RS485 serial port capability port"-->
</RS485Config>
<AlarmhostRS485Config>
  <enabled>true</enabled>
  <!--req, support 485 alarm configuration, for the 485 communication
capability, you can refer to "RS232 and RS485 serial port capability port -->
  <deviceNameLength min="0" max="32"/>
  <!--req, 485 name -->
  <wDeviceType>true</wDeviceType><!--support device type configuration, the
specific device type, you get it from external devices interface list -->
  <deviceProtocol>true</deviceProtocol><!--support device type configuration,
the specific device protocol type, you get it from external devices protocol
interface list-->
  <baudRate>true</baudRate >
  <!--Baud rate-->
</AlarmhostRS485Config>

<RS485SlotConfig>
  <enabled>true</enabled>
  <!--req, 485 slot parameter configuration-->
  <deviceNameLength min="0" max="32"/>
  <!--req, 485 name -->
  <wDeviceType>true</wDeviceType><!--support device type configuration, the
specific device type, you get it from external devices interface list-->
  <deviceProtocol>true</deviceProtocol><!--support device type configuration,
the specific device protocol type, you get it from external devices protocol
interface list -->
  <deviceAddress min="0" max="65535"/>
  <!--device address-->
  <channel>true</channel>
  <!--req, channel No.-->
  <slotChan>true</slotChan>
  <!--req, slot No.-->
</RS485SlotConfig>

<UploadExternalDeviceAlarm>
  <enabled>true</enabled>
  <!--req, 485 external device alarm uploading-->
</UploadExternalDeviceAlarm>
<GetExternalDeviceStatus>
  <enabled>true</enabled>
  <!--req, get 485 external devices status-->
</GetExternalDeviceStatus>
<ExternalDeviceLimitValueConfig>
  <enabled>true</enabled>
  <!--req, limiting value configuration of 485 xternal devices-->
```

```
</ExternalDeviceLimitValueConfig>
<RS485ProtocolVersion>
  <enabled>true</enabled>
  <!--req, Supports getting RS485 protocol version information-->
</RS485ProtocolVersion>

<Sensor>
  <SensorConfig>
    <enabledConfig>true</enabledConfig>
    <!--req, support analog quantity configuration -->
    <sensorNameLength min="" max=""/>
    <!--req, length of analog quantity name -->
    <enabledSensor>true</enabledSensor>
    <!--req, analog quantity enable-->
    <normalWork>true</normalWork>
    <!--req, work status display of sensor-->
    <alarmMode opt="HHHH,HHHL,HHLL,HLLL,LLLL"/>
    <!--req, alarm mode supported-->
    <sensorType opt="none,temperature, humidity, windspeed,
gas,ACVoltage,ACCurrent,DCVoltage,
DCCurrent,waterPressure,pressureTransmitter,flowTransmitter,waterLeakage,
intergratedTemperatureDetector, isolationTemperatureDetector,
residualChlorine,others"/>
    <!--req,alarm type supported: none, Temperature sensor, humidity sensor,
speed sensor, gas sensor, voltage sensor, Ac current sensor, dc voltage sensor,
dc current sensor, pressure sensor, pressure transmitter, flow transmitter,
leakage sensors, Integrated temperature sensor, isolation type temperature
sensor, online residual chlorine, custom -->
    <measureHigh min="-10000.000" max="10000.000"/>
    <!--req, Upper limitation of Range Value-->
    <measureLow min="-10000.000" max="10000.000"/>
    <!--req, Lower limitation of Range Value-->
    <alarmLimitValue1 min="-10000.000" max="10000.000"/>
    <!--req, limited value 1 of alarm -->
    <alarmLimitValue2 min="-10000.000" max="10000.000"/>
    <!--req, limited value 2 of alarm-->
    <alarmLimitValue3 min="-10000.000" max="10000.000"/>
    <!--req, limited value 3 of alarm -->
    <alarmLimitValue4 min="-10000.000" max="10000.000"/>
    <!--req, limited value 4 of alarm-->
    <osd>true</osd>
    <!--req, analog quantity string overlay -->
    <sensitive min="0.010" max="1.000"/>
    <!--req, sensitivity-->
    <sensorStandard opt="4~20mA,0~5V"/>
    <!--req, sensor specification-->
  </SensorConfig>
  <SensorValueUpload>
    <enabled>true</enabled>
    <!--req, data uploading of analog quantity-->
  </SensorValueUpload>
```

```

<GetSensorValue>
  <enabled>true</enabled>
  <!--req, get real time data of analog quantity -->
  <absTime>true</absTime>
  <!--req, absolute time -->
  <sensorName>true</sensorName>
  <!--req, analog quantity name -->
  <sensorChannel>true</sensorChannel>
  <!--req, analog quantity channel -->
  <sensorType opt="none,temperature,
humidity,windspeed,gas,ACVoltage,ACCurrent,DCVoltage,DCCurrent,
waterPressure,pressureTransmitter,flowTransmitter,waterLeakage,
intergratedTemperatureDetector,
isolationTemperatureDetector, residualChlorine,others"/>
  <!--req,sensor type supported: none, Temperature sensor, humidity sensor,
speed sensor, gas sensor, voltage sensor, Ac current sensor, dc voltage sensor,
dc current sensor, pressure sensor, pressure transmitter, flow transmitter,
leakage sensors, Integrated temperature sensor, isolation type temperature
sensor, online residual chlorine, custom -->
  <alarmType opt="upper4, upper3, upper2, upper1,lower1, lower2, lower3,
lower4,fault"/>
  <!--req, alarm type-->
  <alarmMode opt="HHHH,HHHL,HHLL,HLLL,LLLL"/>
  <!--req,alarm type supported-->
  <value>true</value>
  <!--req, real time data of analog quantity -->
  <originalValue>true</originalValue>
  <!--req, Original current and voltage value, judge whether the value is
current or voltage according to the sensor bySensorStandard type.-->
</GetSensorValue>

<SensorAlarmUpload>
  <enabled>true</enabled>
  <!--req, alarm uploading of analog quantity-->
</SensorAlarmUpload>
<SensorJointConfig>
  <enabled>true</enabled>
  <!--req, configuration linkage of analog quantity-->
  <sensorJointAlarmout opt="joint,notJoint"/>
  <!--req, alarm input linkage of analog quantity (defence area) , no
linkage -->
  <sensorJointSiren opt="joint,notJoint"/>
  <!--req, warning signal linkage of analog quantity-->
  <sensorAlarmTypeJointAlarmOut opt="alarmLimitValue1, alarmLimitValue2,
alarmLimitValue3, alarmLimitValue4,fault"/>
  <!--req, alarm output linkage by analog quantity alarm type (trigger),
limitation 1, limitation 2, limitation 3, limitation 4, malfunction -->
  <sensorAlarmTypeJointSiren opt="alarmLimitValue1, alarmLimitValue2,
alarmLimitValue3, alarmLimitValue4,fault"/>
  <!--req, alarm output linkage by analog quantity alarm type, limitation
1, limitation 2, limitation 3, limitation 4, malfunction -->
  <chan>true</chan>

```

```

        <!--req, channel No.-->
    </SensorJointConfig>
</Sensor>
<SwitchAlarmUpload>
    <enabled>true</enabled>
    <!--req, switching value alarm uploading-->
</SwitchAlarmUpload>

<SubSystem>
    <subSystemNo opt="1,2,3,4,5,6,7,8,0xffffffff"/>
    <!--req, subsystem number-->
    <SubSystemConfig>
        <Part1Config>
            <enterDelayTime min="10" max="150"/>
            <!--req, enter delay time,uint: s -->
            <exitDelayTime min="10" max="300"/>
            <!--req, exit delay time,uint: s -->
            <hostageReportEnable>true</hostageReportEnable>
            <!--req, hostage report enable-->
            <keyboardWarmingOfArmDisarmReport>true</
keyboardWarmingOfArmDisarmReport>
            <!--req, keyboard warning of sending disarming report success -->
            <keyboardWarmingOfTestReport>true</keyboardWarmingOfTestReport>
            <!--req, keyboard warning of sending test report success-->
            <sirenDelayTime min="" max=""/>
            <!--req, continuous output time of warning signal, uint: s -->
            <publicSubSystemNum>1</publicSubSystemNum>
            <!--req, number of subsystem -->
            <keySwitchZoneArm>true</keySwitchZoneArm>
            <!--req, support key arming and disarming of subsystem defence area-->
            <keySwitchZoneArmReport>true</keySwitchZoneArmReport>
            <!--req, support key arming and disarming to send arming report of
subsystem defence area-->
            <keySwitchZoneDisarm>true</keySwitchZoneDisarm>
            <!--req, support key arming and disarming to disarm operation of
subsystem defence area-->
            <keySwitchZoneDisarmReport>true</keySwitchZoneDisarmReport>
            <!--req, support key arming and disarming to send disarming report of
subsystem defence area-->
            <notSupportSubsystemEnable>false</notSupportSubsystemEnable>
            <!--, Subsystem enable reverse capability, return true on not support,
no return or return false means support-->
        </Part1Config>
        <Part2Config>
            <armTime>true</armTime>
            <!--req, time settings for arming and disarming -->
            <alarmInAdvance min="" max=""/>
            <!--req, remind the time before automatic arm and disarm, uint: minute
-->
            <jointZone>true</jointZone>
            <!--req, support subsystem linking defence area-->
            <jointKeyboard>true</jointKeyboard>

```

```

    <!--req, support subsystem linking Keyboard -->
    <jointOperaterUser>true</jointOperaterUser>
    <!--req, support subsystem linking operator user -->
    <alarmRemindTime>
        <enable>true</enable>
        <!--req, whether to support arming reminder time settings-->
        <maxDay>7</maxDay>
        <!--req, max days-->
        <timesegment>8</timesegment>
        <!--req, time periods-->
    </alarmRemindTime>
</Part2Config>
</SubSystemConfig>
<subSystemArmType opt="armAway,armInstant,armStay"/>
<!--req, subsystem arm type, go out arming, Real-time protection and Left-
behind protection -->
<SubSystemDisarm>
    <enabled>true</enabled>
    <!--req, support subsystem disarming-->
</SubSystemDisarm>
<SubSystemClearAlarm>
    <enabled>true</enabled>
    <!--req, support subsystem dis-warning-->
</SubSystemClearAlarm>
<SubSystemCloseWarning>
    <enabled>true</enabled>
    <!--req, support close Fault prompt ring of subsystem-->
</SubSystemCloseWarning>
<SubSystemGroupBypass>
    <enabled>true</enabled>
    <!--req, support by-pass of subsystem group, bypass recovery -->
</SubSystemGroupBypass>

<subSystemIDLength min="" max=""/>
<!--req, The length of subsystem ID-->
<subSystemID
opt="0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z,A,B
,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z"/>
    <!--req, The characters supported by subsystem ID-->
    <keyZoneArmReportEnable>true</keyZoneArmReportEnable >
    <!--req, Enable uploading arming report of key zone-->
    <keyZoneArmEnable>true</keyZoneArmEnable >
    <!--req, Enale key zone-->
    <oneKeyAlarmEnable>true</oneKeyAlarmEnable >
    <!--req, Enable one-touch arming-->

    <singleZoneAlarmEnable>true</singleZoneAlarmEnable >
    <!--req, Enable single zone arming and disarming -->
    <centerType>true</centerType >
    <!--req, center account type-->
    <centerAccount
opt="0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z,A,B

```



```
,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z"/>
  <!--req, character supported by center account-->
  <centerAccountV40
opt="0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z,A,B
,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z"/>
  <!--req, character supported by center account-->
  <SubSystemControl>
    <mandatoryAlarm>true</mandatoryAlarm >
    <!--req, support subsystem forced arming-->
  </SubSystemControl>
</SubSystem>

<telModuleNo min="" max=""/>
<!--req, telephone module ID-->
<DialConfig>
  <PhoneCenterParameter>
    <centerPhoneNumberLength min="" max=""/>
    <!--req, Length of center Phone Number -->
    <centerPhoneNumber opt="0,1,2,3,4,5,6,7,8,9"/>
    <!--req, string supported of center Phone Number -->
    <repeatCall min="" max=""/>
    <!--req, dial number-->
    <dialDelay min="" max=""/>
    <!--req, dial delay, uint: s -->
    <pstnTransMode opt="DTMF 5/S,DTMF 10/S"/>
    <!--, Transmission Mode-->
    <receiveID opt="0,1,2,3,4,5,6,7,8,9,F,E"/>
    <!--, string supported of receiving account -->
    <receiveIDLength min="" max=""/>
    <!--req, length of receiving account-->
    <enabled opt="true,false"/>
    <!--opt ,whether to support enable, the node will not be returned if not
supported-->
  </PhoneCenterParameter>
  <reportEnable>
    <reportPeried min="" max=""/>
    <!--req, Upload cycle of test report, unit: hour -->
    <firstReportTime min="" max=""/>
    <!--req, the uploading time of the first test report, unit: minute -->
    true<!--req, whether enable test report uploading-->
  </reportEnable>
</DialConfig>

<DialMode>
  <enabled>true</enabled>
  <!--req, dial enable type-->
</DialMode>

<DialSchedule>
  <hideConfig>true</hideConfig>
  <!--opt, hide the dial schedule configuration or not.-->
</DialSchedule >
```

```

<GPRSModuleNo min="" max=""/>
<!--req, GPRS module ID-->
<!--<3GModuleNo min="" max=""/>-->
<!--req, 3G module ID-->
<WirelessNetworkConfig>
  <NetParameter>
    <domainName>true</domainName>
    <!--req, domain-->
    <reportProtocol opt="private,NAL2300, EHome"/>
    <!--,alarm uploading protocol supported-->
    <deviceIDLength min="" max=""/>
    <!--req, name length of devices ID -->
    <deviceID
opt="0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z,
A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z"/>
    <!--,characters that devices ID supported-->
    <protocolList>
      <!--req, Alarm upload protocol capability-->
      <private>
        <deviceIDLength min="" max=""/>
        <!--req, length of device ID name-->
      </private>
      <NAL2300>
        <deviceIDLength min="" max=""/>
        <!--req, length of device ID name-->
      </NAL2300>
    </protocolList>
    <addressType opt="IP/IPV6, domain"/>
    <!--req,Supported address type-->
    <protocolVersion opt="v2.0,v4.0,v5.0"/>
    <!--opt, EHome protocol version-->
  </NetParameter>
  <netType opt="1, 2"/>
  <!--req, NIC type (1-Main NIC, 2-Extended NIC)-->
  <APNName>true</APNName>
  <!--req, APN name-->
  <APNUserName>true</APNUserName>
  <!--req, APN user name-->
  <APNPassword>true</APNPassword>
  <!--req, APN password-->
  <reconnectTime min="" max=""/>
  <!--req, reconnection time when connect failed, uint: 10s -->
  <overTime min="" max=""/>
  <!--req, timeout , reconnection when it does not receive the valid data,
uint: 30s-->
  <detectLinkTime min="" max=""/>
  <!--req, detect time, whether the detect periodic line is hold, uint: 10s--
>
  <SIMNumberLength min="" max=""/>
  <!--req, SIM ID, Phone number -->

```

```
<SIMIP>true</SIMIP>
<!--req, The IP address of the landing after the network to allocate, get
only -->
</WirelessNetworkConfig>
<netModuleNumber min="" max=""/>
<!--req, Network module number-->
<netModuleAddr min="" max=""/>
<!--req, Network module address-->

<netModuleNo min="" max=""/>
<!--req, serial number of Cable network module -->
<NetworkConfig>
  <NetParameter>
    <domainName>true</domainName>
    <!--req, domain-->
    <reportProtocol opt="private,NAL2300"/>
    <!--,alarm uploading protocol supported-->
    <deviceIDLength min="" max=""/>
    <!--req, name length of devices ID.If device return protocolList, it
means the node is invalid, no return. -->
    <deviceID
opt="0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z"/>
    <!--,characters that devices ID supported-->
    <protocolList>
      <private>
        <deviceIDLength min="" max=""/>
        <!--, Device ID length-->
      </private>
      <NAL2300>
        <deviceIDLength min="" max=""/><!--, Device ID length-->
      </NAL2300>
    </protocolList>
    <addressType opt="IP/IPV6, domain"/>
    <!--req, supported address type-->
    <protocolVersion opt="v2.0,v4.0,v5.0"/>
    <!--opt, EHome protocol version-->
  </NetParameter>
  <netType opt="1,2"/><!--NIC type, 1-main NIC, 2-extension NIC-->
  <NetModuleUpgrade>
    <enabled>true</enabled>
    <!--Network module supports upgrade-->
  </NetModuleUpgrade >
</NetworkConfig>

<centerGroupNo min="" max=""/>
<!--req, center Group number-->
<ReportModeConfig>
  <enabled>true</enabled>
  <!--req, alarm uploading configuration -->
  <valid>true</valid>
  <!--req whether enable-->
  <alarmMode opt="T1, T2,N1,N2,G1,G2,N3,N4"/>
```

```

    <!--opt, Alarm channel-->
    <dataType
opt="allAlarmData,allNonAlarmData,allData,zoonAlarmData,zoonNonAlarmData"/>
    <!--data type-->
    <alarmChannelMode>true</alarmChannelMode>
    <!--alarm channel configuration supported-->
    <dealFailCenter>true</dealFailCenter>
    <!--req send report to the specific group-->
    <uploadZoneReport opt="upload, notUpload" />
    <uploadNonZoneReport
opt="softZoneReport,systemStatusReport,cancelReport,testReport,armReport,disarmR
eport,

duressReport,alarmRestoreReport,bypassReport,bypassRestoreReport,detectorConnect
Report,
    detectorPowerReport, videoAlarmReport"/>
    <!--req,Non-zone alarm report, system status report, cancelling report,
test report, arming report, disarming report, duress report, alarm recovery
report,
    bypass report, bypass recovery report, detector connection status report
(online/offline), detector power status report (normal/under voltage), video
alarm report -->
    <alarmChanNum min="" max=""/>
    <!--channel number, one main channel, the others are spare channels.
Reverse capability, if the device did not return this capability, it means
supported by 4 channels.-->
    <alarmNetCard opt="primaryCard-1, primaryCard-2, extendCard-1,
extendCard-2"/>
    <!--opt, Alarm network card center-->
</ReportModeConfig>

<ProcessFaultConfig>
    <supportFaultType
opt="ACOutage,lowVoltageOfBattery,devicePreventDisassemble,telephoneOffLine,RS48
5busAbnormal,

networkAbnormal,wirelessAbnormal,expandbusAbnormal,hardDiskAbnormal,FPGAFault,se
nsorFault"/>
    <!--, Support the fault types, ac power-off, battery undervoltage, host
apart, telephone line drops, 485 abnormal equipment, network fault, wireless
abnormalities, abnormal expansion bus, hard disk exception, FPGA and analog
fault-->
    <checkFault>true</checkFault>
    <!--req, fault detect-->
    <overallFaultJointLED>true<!--req, Fault associated global keyboard light
output--></overallFaultJointLED>
    <overallFaultJointSound>true<!--req, Fault associated global keyboard audio
output--></overallFaultJointSound>
    <subsystemFaultJointLED>true<!--req, Fault associated subsystem keyboard
light output--></subsystemFaultJointLED>
    <subsystemFaultJointSound>true<!--req, Fault associated subsystem keyboard
audio output--></subsystemFaultJointSound>

```

```
<faultJointFaultLight>true<!--req, Fault associated fault keyboard light
output--></faultJointFaultLight>
</ProcessFaultConfig>

<OverallFaultAlarmNo min="" max=""/>
<!--req, global keyboard number-->
<CloseOverallFaultAlarm>
  <enabled>true</enabled>
  <!--req, close fault warning tone of global keyboard -->
</CloseOverallFaultAlarm>
<TriggerEventConfig>
  <supportOverallEvent
opt="ACOutage,lowVoltageOfBattery,telephoneOffLinek,networkAbnormal,wirelessNetw
orkAbnormal"/>
  <!--,global event supported-->
  <supportSubsystemEvent opt="enterDelayTime,exitDelayTime,arm,disarm,alarm,
clearAlarm,AlarmRestore"/>
  <!--, subsystem event supported -->
  <overallEventTriggerAlarmoutOn>true</overallEventTriggerAlarmoutOn><!--req,
Global event trigger flip-flop open -->
  <overallEventTriggerAlarmoutOff>true</overallEventTriggerAlarmoutOff><!--
req, Global event trigger flip-flop close -->
  <subsystemEventTriggerAlarmoutOn>true</subsystemEventTriggerAlarmoutOn><!--
req, subsystem event trigger flip-flop open-->
  <subsystemEventTriggerAlarmoutOff>true</
subsystemEventTriggerAlarmoutOff><!--req, subsystem event trigger flip-flop
close-->
</TriggerEventConfig>

<BatteryVoltage>
  <enabled>true</enabled>
  <!--req, support the query of battery voltage-->
</BatteryVoltage>
<AlarmHostLog>
  <enabled>true</enabled>
  <!--req, support the query of alarm host log-->
</AlarmHostLog>
<FaultAlarmUpload>
  <enabled>true</enabled>
  <!--req, support Fault alarm uploading-->
</FaultAlarmUpload>
<RemoteUpgrade>
  <enabled>true</enabled>
  <!--req, support remote upgrade-->
</RemoteUpgrade>
<VoiceUpload>
  <enabled>true</enabled>
  <!--req, support audio uploading-->
</VoiceUpload>
<VoiceDownload>
  <enabled>true</enabled>
  <!--req, support audio download-->
```

```
</VoiceDownload>
<voiceNo min="" max=""/>
<!--,audio serial number-->
<VoiceControl>
  <enabled>true</enabled>
  <!--req, support audio control-->
</VoiceControl>
<UploadSafetyCabinState>
  <enabled>true</enabled>
  <!--req, Protective tank status uploading-->
</UploadSafetyCabinState>
<UploadAlarmoutStatus>
  <enabled>true</enabled>
  <!--req, support initiative uploading alarm output status-->
</UploadAlarmoutStatus>
<UploadSirenStatus>
  <enabled>true</enabled>
  <!--req, upport initiative uploading warning signal status-->
</UploadSirenStatus>
<AudioAssociateAlarmEventConfig>
  <enabled>true</enabled>
  <!--req, audio associated alarm event -->
</AudioAssociateAlarmEventConfig>
<CIDReportUpload>
  <enabled>true</enabled>
  <!--req, CID report uploading-->
</CIDReportUpload>

<RS485ExternalDeviceAlarmUpload>
  <enabled>true</enabled>
  <!--req, 485 external devices alarm uploading-->
</RS485ExternalDeviceAlarmUpload>
<GetExternalDeviceState>
  <enabled>true</enabled>
  <!--req, get external devices status-->
</GetExternalDeviceState>
<GetVariableList>
  <enabled>true</enabled>
  <!--req, get variable element list-->
  <LocalSensorChannel>
    <enable>true</enable>
    <!--support variable element list of local analog quantity -->
    <channel>true</channel>
    <!--channel number, the specific channel range id depend on the channel
type, such as when the type is local analog quantity , channel range will be
get from local analog quantity node. the node of the channel means that whether
to send parameter when getting variable element list-->
    <subChannel>true</subChannel>
    <!-- subchannel, slot number-->
  </LocalSensorChannel>
  <LocalSwitchChannel>
    <enable>true</enable>
```

```
<!--support getting variable list of local switching value -->
<channel>true</channel>
<!--channel number-->
<subChannel>true</subChannel>
<!--subchannel, slot number-->
</LocalSwitchChannel>
<RS485Channel>
  <enable>true</enable>
  <!--support getting variable list of 485 channel-->
  <channel>true</channel>
  <!--channel number-->
  <subChannel>true</subChannel>
  <!-- subchannel, slot number-->
</RS485Channel>
<AlarmVariableConfig>
  <variableIndex attri="readonly">true</variableIndex>
  <!--variable number-->
  <variableType attri="readonly" opt="sensor,switch"/>
  <!--variable type, analog quantity, switching value-->
  <variableDescribe attri="readonly">true</variableDescribe>
  <!-- variable description-->
  <limitLineType attri="readonly"
opt="notSupport,twoLimitLines,fourLimitLines"/>
  <!--variable limitation type-->
</AlarmVariableConfig>
</GetVariableList>
<AlarmPointConfig>
  <enable>true</enable>
  <!--alarm point configuration-->
  <pointNo min="" max=""/>
  <!--point range-->
  <pointDescribe>true</pointDescribe>
  <!--point description -->
  <pointType opt="sensor,switch"/>
  <!-- point type supported-->
  <PointParam>
    <enable>true</enable>
    <!--point parameter-->
    <SensorPoint><!--telemetering point parameter-->
      <measureHigh min="-10000.000" max="10000.000"/>
      <!--req, Upper limitation of Range Value -->
      <measureLow min="-10000.000" max="10000.000"/>
      <!--req, Lower limitation of Range Value-->
      <alarmMode opt="HHHH,HHHL,HHL,LLLL,LLLL"/>
      <!--,alarm mode supported-->
      <alarmLimitValue1 min="-10000.000" max="10000.000"/>
      <!--req, limitation value 1 of alarm-->
      <alarmLimitValue2 min="-10000.000" max="10000.000"/>
      <!--req, limitation value 2 of alarm-->
      <alarmLimitValue3 min="-10000.000" max="10000.000"/>
      <!--req, limitation value 3 of alarm-->
      <alarmLimitValue4 min="-10000.000" max="10000.000"/>
```

```

        <!--req, limitation value 4 of alarm-->
        <osd>true</osd>
        <!--req, analog quantity string overlay-->
        <sensitive min="0.010" max="1.000"/>
        <!--req, sensitivity-->
    </SensorPoint>
    <SwitchPoint>
        <enable>true</enable>
        <!--telemetry point parameter-->
    </SwitchPoint>
</PointParam>
<chanType opt="analogChan,switchChan,485Chan,netChan"/>
<!--req, access type: 1- local analog channel, 2- local relay channel, 3-
485 channel, 4- network channel-->
<chanNo>true</chanNo>
<!--req, channel No.-->
<subChanNo>true</subChanNo>
<!--req, slot No.-->
<variableNo>true</variableNo>
<!--req, variable No.-->
</AlarmPointConfig>
<AlarmModeConfig>
    <enable>true</enable>
    <!--support alarm configuration mode interface-->
    <dataUploadMode opt="uploadByChannel,uploadByPoint"/>
</AlarmModeConfig>
<AlarmHostDataUpload>
    <enable>true</enable>
    <!--support alarm data uploading interface -->
</AlarmHostDataUpload>
<PrinterConfig>
    <enablePrinter>true</enablePrinter>
    <!--enable printer-->
    <printTime>true</printTime>
    <!--whether enable print time-->
    <faultDetect>true</faultDetect>
    <!-- whether support fault detection -->
    <alarmInfo
opt="zoneAlarm,zoneAlarmRestore,emergencyKeypadAlarm,duressAlarm"/>
    <!--alarm event, defence area alarm, defence area alarm recovery, Emergency
Alarm, coerce alarm-->
    <deviceInfo
opt="ACLoss,ACLossRestore,systemLowBattery,systemLowBatteryRestore,PSTNFault,PST
NFaultRestore,testReport,tamperAlarm,tamperAlarmRestore,RS485deviceFault,RS485de
viceFaultRestore,wirelessFault,wirelessFaultRestore,networkFault,networkFaultRes
tore,BUSFault,BUSFaultRestore,harddiskFault,hardDiskFaultRestore"/>
    <!--device info, ac power-off, ac power-off recovery, battery
undervoltage, battery undervoltage recovery, telephone line drops, telephone
line drops recovery,Test report, tamper, tamper recovery, 485 abnormal
equipment,485 abnormal equipment recovery, network fault, network fault
recovery, wireless abnormalities, wireless abnormalities recovery, Cable
network anomalies, Cable network anomalies recovery, Abnormal expansion bus,

```



```
Abnormal expansion bus recovery-->
  <operateInfo
opt="arm,disarm,clearAlarm,bypass,bypassRestore,enterProgram,exitProgram,restart
"/>
  <!--operation event, arm, disarm. dis-warning, by-pass, by0pass recovery,
enter programming, exit programming, reboot-->
  </PrinterConfig>
  <SearchHistoryData>
    <SearchCondition>
      <majorType opt="all,time,type,timeAndtype"/>
      <!--main type: all, by time, by type, time + type-->
      <minorType opt="point,variable"/>
      <!--type: point number, channel+slot+variable number-->
      <startTime>true</startTime>
      <!--start time support-->
      <stopTime>true</stopTime>
      <!--end time support -->
      <chanType opt="sensor,switch,RS485"/>
      <!--channel type: analog quantity, switching value, 485 channel-->
      <sensorChan min="1" max="16"/>
      <RS485Chan>
        <chanNo min="1" max="8"/>
        <!--channel number-->
        <subChanNo min="1" max=""/>
        <!--slot number-->
      </RS485Chan>
      <variableNo min="1" max=""/>
      <!--variable number-->
      <pointNo min="1" max=""/>
      <!--point number-->
    </SearchCondition>
    <HistoryData>
      <struTime>true</struTime>
      <!--time-->
      <chanType opt="sensor,switch,RS485"/>
      <!--channel type: analog quantity, switching value, 485 channel-->
      <chanNo min="1" max=""/>
      <!--channel -->
      <subChanNo min="1" max=""/>
      <!--slot number-->
      <variableNo min="1" max=""/>
      <!--variable number-->
      <pointNo min="1" max=""/>
      <!--point number -->
      <data>true</data>
      <!--historical data-->
    </HistoryData>
  </SearchHistoryData>
</OutputScheduleRuleCfg>
  <OutputScheduleRuleList>
    <ruleNum>3</ruleNum>
    <!--req, regulation amount-->
```

```

<OutputScheduleRule>
<!--req, regulation of time output control-->
  <enabled>true</enabled>
  <!--req,enable regulation  -->
  <scheduleDate>true</scheduleDate>
  <!--req,date schedule-->
  <OutputScheduleList>
    <scheduleNum>3</scheduleNum>
    <!--output amount of time control-->
    <OutputSchedule>
      <!--req,output parameter of time control-->
      <scheduleTime>true</scheduleTime>
      <!--req, time schedule-->
      <state opt = "off,on"/>
      <!--req,off-close,on-open-->
      <triggerIndex min="0" max="512"/>
      <!--req, link trigger-->
    </OutputSchedule>
  </OutputScheduleList>
</OutputScheduleRule>
</OutputScheduleRuleList>
</OutputScheduleRuleCfg>
<LED>
  <LEDScreen>
    <nameLength min="1" max="32"/>
    <!--req,length of LED screen name-->
    <transMode opt="serial"/>
    <!--req,communication mode,serial-serial port communication-->
    <protocol>true</protocol>
    <!--req,LED screen protocol-->
    <color opt = "monochrome,bicolour,256GrayDualColor,fullcolor"/>
    <!--req,monochrome-monochrome,bicolour-two-tone,256GrayDualColor-256 gray
level double base colour ,fullcolor-full color-->
    <dataPolarity opt="negative,positive"/>
    <!--req,data polarity ,negative-cathode,positive-anode-->
    <OEPolarity opt="low,high"/>
    <!--req, OE(Output Enable) electrical level,low- low valid,high- high
valid-->
    <scanMode opt="1/16,1/8,1/4,1/2,static"/>
    <!--req, scanning mode-->
    <width min="16" max="2048"/>
    <!--req,width of LED screen -->
    <height min="16" max="2048"/>
    <!--req,height of LED screenLED-->
  </LEDScreen>
  <LEDContent>
    <contentLength min="32" max="512"/>
    <!--req,length of subtitle content-->
    <contentAct opt="static,quick-play,left,right,up,down"/>
    <!--req,subtitle action,static-static,quick-play- quick play,left-move
left,right-move right ,up-move up,down-move down-->
    <contentSpeed min="1" max="24"/>
  </LEDContent>
</LED>

```

```
<!--req,speed show-->
<contentStayTime min="0" max="127.5"/>
<!--req, dwell time-->
</LEDContent>
<LEDSwitch>
  <manual>true</manual>
  <!--req,manual switch-->
  <Auto>
    <!--req,automatic switch-->
    <switchTimeNum>3</switchTimeNum>
    <!--req,number of time quantum-->
    <SwitchTime>
      <valid>true</valid>
      <!--req,effectiveness-->
      <onTime>true</onTime>
      <!--req,uptime-->
      <offTime>true</offTime>
      <!--req,off time-->
    </SwitchTime>
  </Auto>
</LEDSwitch>
<LEDTimeAdjustment>true</LEDTimeAdjustment>
<!--LED timing-->
<LEDBrightnessAdjustment>
  <!--req,LED intensity control -->
  <manual>true</manual>
  <!--req, hand regulation-->
  <Auto>
    <!--req,automatic adjustment-->
    <BrightnessStep>
      <valid>true</valid>
      <!--req,whether the time division intensity control is valid -->
      <timeSegNum>48</timeSegNum>
      <!--req,number of time quantum-->
    </BrightnessStep>
  </Auto>
</LEDBrightnessAdjustment>
<LEDStatus>
  <!--req,device status-->
  <switchState opt="off,on"/>
  <!--req,off-power off status,on-starting up status-->
  <brightNess min="0" max="15"/>
  <!--req,brightness value-->
</LEDStatus>
</LED>
<safetyCabinWorkMode opt="nobody-doorOpen,nobody-doorLocked"/>
<!--req, Protective tank working mode,nobody-doorOpen-door noramlly open with
no people,nobody-doorLocked-door normally lock when no people-->
<SafetyCabinPersonSignal>
  <!--people detection parameters of Protective tank-->
  <sensorType opt="infrared-shooting,curtain-sensor"/>
  <!--req,sensor type,infrared-shooting- infrared radiation,curtain-sensor-
```

```
Curtain sensor-->
  <sensorSensitivity min="0" max="100"/>
  <!--transducer sensitivity-->
  <devUseTimeOut min="5" max="30"/>
  <!--timeout of device using -->
  <curtainDelayTime min="0" max="10"/>
  <!--delay time of Curtain sensor detection-->
  <curtainResponseTime min="0" max="300"/>
  <!--response time of curtain human change control-->
</SafetyCabinPersonSignal>
<WhiteAlarm>
  <!--req, allowlist configuration-->
  <WhiteList>
    <listNum>6</listNum>
    <!--req, default number of allowlist is 6-->
    <enable>true</enable>
    <phoneNum>true</phoneNum>
    <!--whether to display cell phone number-->
    <alarmArmRight min="0" max="32"/>
    <!--req, sub system arming permission-->
    <alarmDisarmRight min="0" max="32"/>
    <!--req, sub system disarming permission-->
    <clearAlarmRight min="0" max="32"/>
    <!--Clear alarm permission-->
    <zoneReport>true</zoneReport>
    <!--req, arming region report, whether to upload-->
    <noneZoneReport
opt="softZoneReport,systemStateReport, cancelReport, testReport, armReport, disarmRe
port,

hijackReport, alarmRetoreReport, byPassReport, byPassRestoreReport, detectorConnectR
eport, detectorPowerReport, videoAlarmReport"/>
    <!--req, soft arming region report, system status arming region report,
cancel report. test report, arming report, disarming report, duress report,
alarm recovery report, bypass report. bypass recovery report, detector
connection status (online, offline), detector power status (normal,
undervoltage), video alarm report-->
  </WhiteList>
  <messageInterval opt="0,1,2,3,4,5,6"/>
  <!--req, 0s, 10s, 30s, 1min, 5min, 10min, custom (1 to 5999s) -->
  <defineInterval min="1" max="5999"/>
  <!--req, custom interval-->
</WhiteAlarm>
<SubSystemTimeAlarm>
  <NormalSchedule>
    <maxDay>7</maxDay>
    <!--req, max days-->
    <timesegment>8</timesegment>
    <!--req, time period-->
    <time>true</time>
    <!--req, normal schedule time-->
    <alarmType opt="outArm, stayArm, immediatelyArm, disArm"/>
```

```

    <!--req, away arming, stay arming, instant arming, disarming-->
</NormalSchedule>
<enableNormalSchedule>true</enableNormalSchedule>
<!--req, whether to enable normal schedule-->
<enableForceArm opt="forceArm,disForceArm"/>
<!--req, forced arming, non-forced arming-->
</SubSystemTimeAlarm>
<!--req, timed sub system disarming-->
<PriorSubSystemAlarm>
    <enableForceArm opt="forceArm,disForceArm"/>
    <!--req, forced arming, non-forced arming-->
    <PriorSchedule>
        <PriorSchedTimeNum>12</PriorSchedTimeNum>
        <!--req, time period-->
        <dataTime>true</dataTime>
        <!--req, normal schedule time-->
        <OneDayTime>
            <timesegment>8</timesegment>
            <!--req, time period-->
            <time>true</time>
            <!--req, priority schedule time-->
            <alarmType opt="outArm,stayArm,immediatelyArm,disArm"/>
            <!--req,away arming, stay arming, instant arming, disarming-->
        </OneDayTime>
    </PriorSchedule>
</PriorSubSystemAlarm>
<ModuleList>
    <enable>true</enable>
    <moduleType opt="keyBoard,trigger,zone, networkModule"/>
    <!--req,LED,LCD, trigger, zone, network module-->
    <keyBoardType opt="LCD,LED"/>
    <!--req, keyboard type, LCD,LED-->
    <TriggerType opt="localTrigger,4trigger,8trigger,singleZoneTrigger,
32trigger"/>
    <!--req, local trigger, 4-ch trigger, 8-ch trigger, single arming region
trigger, 32-ch trigger-->
    <ZoneType opt="localZone,singleZone,2Zone,8Zone,
8sensorZone,singleZoneTrigger"/>
    <!--req,local zone, single zone, dual-zone, 8-zone, 8-channel sensor zone,
single zone trigger-->
    <moduleAddress>true</moduleAddress>
    <moduleInfo>true</moduleInfo>
    <!--req, module information-->
    <versionInfo>true</versionInfo>
    <!--req, version information-->
</ModuleList>
<DeviceSelfCheckState>
    <rs485Chan>64</rs485Chan>
    <!--485 channel connection status-->
    <senorChan>true</senorChan>
    <!-- invalid analog channel No.-->
</DeviceSelfCheckState>

```

```
<AirCondition>
  <enable>true</enable>
  <!-- whether to shut down-->
  <mode opt="refrigeration,hearting,dehumidifier,auto"/>
  <!-- cooling, heating, dehumidifying, auto-->
  <temperature min="16" max="30"/>
  <!--temperature 16-30-->
</AirCondition>
<AuxPower>
  <enable>true</enable>
  <!-- whether to enable-->
  <auxType opt="DC12V,DC24V"/>
  <!-- output type DC12V, DC24V-->
</AuxPower>
<IsSupportAlarmChanAbility>
  <enable>true</enable>
  <!-- whether to support alarm input channel capability-->
</IsSupportAlarmChanAbility>

<AlarmCaptrue>
<!--req,Alarm triggered capture capability-->
  <AlarmCaptrueCfg>
    <channel>1</channel>
    <!--req, video channel No.-->
    <interval min="0" max="65535"/>
    <!--req, interval time-->
    <Resolution>
      <index>1</index>
      <!--req, index-->
      <name>CIF</name>
      <!--req, name-->
      <beforeAlarmPic min="0" max="40"/>
      <!--req, number of pre-alarm pictures-->
      <afterAlarmPic min="0" max="40"/>
      <!--req, number of post-alarm pictures-->
    </Resolution>
    <Resolution>
      <index>2</index>
      <!--req, index-->
      <name>QCIF</name>
      <!--req, name-->
      <beforeAlarmPic min="0" max="80"/>
      <!--req, number of pre-alarm pictures-->
      <afterAlarmPic min="0" max="80"/>
      <!--req, number of post-alarm pictures-->
    </Resolution>
    <Resolution>
      <index>3</index>
      <!--req, index-->
      <name>4CIF</name>
      <!--req, name-->
      <beforeAlarmPic min="0" max="10"/>
```

```
<!--req, number of pre-alarm pictures-->
<afterAlarmPic min="0" max="10"/>
<!--req, number of post-alarm pictures-->
</Resolution>
<Resolution>
  <index>4</index>
  <!--req, index-->
  <name>2CIF</name>
  <!--req, name-->
  <beforeAlarmPic min="0" max="20"/>
  <!--req, number of pre-alarm pictures-->
  <afterAlarmPic min="0" max="20"/>
  <!--req, number of post-alarm pictures-->
</Resolution>
<Resolution>
  <index>5</index>
  <!--req, index-->
  <name>WD1</name>
  <!--req, name-->
  <beforeAlarmPic min="" max=""/>
  <!--req, number of pre-alarm pictures-->
  <afterAlarmPic min="" max=""/>
  <!--req, number of post-alarm pictures-->
</Resolution>
<Resolution>
  <index>6</index>
  <!--req, index-->
  <name>VGA</name>
  <!--req, name-->
  <beforeAlarmPic min="" max=""/>
  <!--req, number of pre-alarm pictures-->
  <afterAlarmPic min="" max=""/>
  <!--req, number of post-alarm pictures-->
</Resolution>
<Resolution>
  <index>7</index>
  <!--req, index-->
  <name>XVGA</name>
  <!--req, name-->
  <beforeAlarmPic min="" max=""/>
  <!--req, number of pre-alarm pictures-->
  <afterAlarmPic min="" max=""/>
  <!--req, number of post-alarm pictures-->
</Resolution>
<Resolution>
  <index>8</index>
  <!--req, index-->
  <name>720P</name>
  <!--req, name-->
  <beforeAlarmPic min="" max=""/>
  <!--req, number of pre-alarm pictures-->
  <afterAlarmPic min="" max=""/>
```

```

        <!--req, number of post-alarm pictures-->
        <!--Picture number after alarm-->
    </Resolution>
</AlarmCaptrueCfg>
</AlarmCaptrue>

<KeyboardAlarm>
    <enableKeyboardAlarm>true</enableKeyboardAlarm>
    <enableAlarmSound>true</enableAlarmSound>
</KeyboardAlarm>
<WirelessBusiness>
    <enable>true</enable>
    <businessType opt="TelephoneCharges,DataFlow"/>
    <!--req, search service type-->
    <communicationOperatorNum opt="0,1,2,3,4,5,6,7,8,9"/>
    <!--req, characters supported by service provider-->
    <queryCode
opt="0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z,A,B
,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z"/>
    <!--req, characters supported by service search code-->
</WirelessBusiness>
<RemoteController>
<!--req, remote control-->
    <enable>true</enable>
    <deviceID min="1" max="32"/>
    <!--req, device (remote control) No.-->
    <IsSupportkeyboardAddr>true</IsSupportkeyboardAddr>
    <!--req, whether to receive keyboard address-->
    <keyboardAddr min="0" max="31"/>
    <!--req,dep IsSupportkeyboardAddr -->
    <!--req, keyboard address-->
    <subSystemID min="1" max="8"/>
    <!--req, sub system No. of device (remote control)-->
    <deviceSnLength min="0" max="9"/>
    <!--req, device (remote control) serial number length-->
    <deviceSnType opt="0,1,2,3,4,5,6,7,8,9"/>
    <!--req, device (remote control) serial number type-->
    <alarmRight
opt="armRight,disArmRight,armReportRight,disArmReportRight,clearAlarmRight"/>
    <!--req, supported permission type: arming permission, disarming
permission, arming report permission, disarming report permission, alarm clear
permission-->
    <enabledDel opt="true,false"/>
    <!--opt, supports deleting remote control user-->
    <alwaysOpenRight opt="Allow,NotAllow"></alwaysOpenRight>
    <!--whether the door remaining open is allowed: "Allow"-yes, "NotAllow"-no--
>
    <openingDirection opt="InDirection,OutDirection"></openingDirection>
    <!--door opening direction: "InDirection"-entrance, "OutDirection"-exit-->
</RemoteController>
<CenterServerConfig>
<!--req, alarm center server configuration-->

```



```

    <addressType opt="IP/IPV6, domain"/>
    <!--req, supported address type-->
    <time min="" max=""/>
    <!--req, alarm interval time-->
</CenterServerConfig>
<ZoneLinkageChannel>
<!--req, arming region; linked video channel-->
    <enable>true</enable>
    <!--req,support zone video channel linkage or not.-->
    <addressType opt="IP/IPV6, domain"/>
    <!--req, supported address type-->
    <ddnsType opt="IPServer,HiDDNS"/>
    <supportZoneNo min="" max=""/>
    <!--req, supported zones-->
</ZoneLinkageChannel>
<SafetyCabinCascadeCfg>
    <enable>true</enable>
    <!--req, whether to support safety cabin cascading-->
</SafetyCabinCascadeCfg>
<SafetyCabinWorkMode>
    <setConfig>true</setConfig>
    <!--req, whether to support safety cabin operating mode configuration-->
</SafetyCabinWorkMode>
<SafetyCabinStatus>
    <getConfig>true</getConfig>
    <!--req, whether to support getting safety cabin status by configuration-->
</SafetyCabinStatus>
<KeyboradUserLockCfg>
    <errorTimes min="" max=""/>
    <!--req,error times of inputting the keyboard password (lock keyboard)-->
    <lockTime min="" max=""/>
    <!--req,Keyboard locking period, second-->
</KeyboradUserLockCfg>
    <noZoneRelatedTrigger opt="true,false"/>
    <!--opt whether to support configuring zone trigger linkage-->
</AlarmHostAbility>

```

A.3.198 XML_AudioFileList

Message about the audio file list in XML format.

```

<AudioFileList version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
    <AudioFile>
        <type><!--required, xs:string, audio file type: "callWaitting",
"consultWaitting"--></type>
        <name><!--required, xs:string, file name--></name>
    </AudioFile>
</AudioFileList>

```

A.3.199 XML_AudioInOutCfg

Message about the audio input and output parameters in XML format.

```
<AudioInOutCfg version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <audioInManualCfgEnable><!--optional, xs:boolean, whether to enable
configuring audio input manually--></audioInManualCfgEnable>
  <audioOutManualCfgEnable><!--optional, xs:boolean, whether to enable
configuring audio output manually--></audioOutManualCfgEnable>
  <Intercom>
    <AudioInType><!--required, xs:string, audio input type: "micIn", "lineIn1",
"list"--></AudioInType>
    <AudioInList>
      <AudioIn>
        <id><!--required, xs:string, audio input ID: "micIn", "lineIn1"--></id>
        <isConfiged><!--optional, xs:boolean, whether the device type is
configured: true, false. If the device type of the device corresponding to the
ID is configured and applied, this node should be set to true--></isConfiged >
        <volume><!--required, xs:integer, volume level, value range: [0, 10], 0
means the device is muted, and the default volume is 6--></volume>
      </AudioIn>
    </AudioInList>
    <AudioOutType><!--required, xs:string, audio output type: "spkOut",
"spkOut1", "spkOut2", "lineOut1", "spkOut_lineOut1", "list"--></AudioOutType>
    <AudioOutList>
      <AudioOut>
        <id><!--required, xs:string, audio output ID: "spkOut", "spkOut1",
"spkOut2", "lineOut1", "spkOut_lineOut1"--></id>
        <isConfiged><!--optional, xs:boolean, whether the device type is
configured: true, false. If the device type of the device corresponding to the
ID is configured and applied, this node should be set to true--></isConfiged >
        <volume><!--required, xs:integer, volume level, value range: [0, 10], 0
means the device is muted, and the default volume is 6--></volume>
      </AudioOut>
    </AudioOutList>
  </Intercom>
  <Broadcast>
    <AudioOutType><!--required, xs:string, audio output type: "spkOut",
"spkOut1", "spkOut2", "lineOut1", "spkOut_lineOut1", "list"--></AudioOutType>
    <AudioOutList>
      <AudioOut>
        <id><!--required, xs:string, audio output ID: "spkOut", "spkOut1",
"spkOut2", "lineOut1", "spkOut_lineOut1"--></id>
        <isConfiged><!--optional, xs:boolean, whether the device type is
configured: true, false. If the device type of the device corresponding to the
ID is configured and applied, this node should be set to true--></isConfiged >
        <volume><!--required, xs:integer, volume level, value range: [0, 10], 0
means the device is muted, and the default volume is 6--></volume>
      </AudioOut>
    </AudioOutList>
  </Broadcast>
```

```
<VoicePrompt>
  <AudioOutType><!--required, xs:string, audio output type: "spkOut1",
"spkOut2", "lineOut1", "spkOut_lineOut1", "list"--></AudioOutType>
  <AudioOutList>
    <AudioOut>
      <id><!--required, xs:string, audio output ID: "spkOut1", "spkOut2",
"lineOut1", "spkOut_lineOut1"--></id>
      <isConfiged><!--optional, xs:boolean, whether the device type is
configured: true, false. If the device type of the device corresponding to the
ID is configured and applied, this node should be set to true--></isConfiged >
      <volume><!--required, xs:integer, volume level, value range: [0, 10], 0
means the device is muted, and the default volume is 6--></volume>
    </AudioOut>
  </AudioOutList>
</VoicePrompt>
</AudioInOutCfg>
```

A.3.200 XML_CAMERAPARA

CAMERAPARA message in XML format

```
<xml version="1.0" encoding="utf-8"?>
<!--req, camera parameter capability set description -->
<CAMERAPARA version="2.0">
  <ChannelList>
    <ChannelEntry>
      <ChannelNumber>1</ChannelNumber>
      <IPStartChanNoDefault><!--optional, xs:integer, start digital channel No.
The device's digital channel No. will start from the returned No.--></
IPStartChanNoDefault>
      <PowerLineFrequencyMode><!-- req, format -->
        <isNotSupportDigitalChanCfg opt="true,false"/><!--optional, whether
setting digital channels is not supported: true=yes (not supported), false=no
(supported). If this field is not returned, it indicates that setting digital
channels is supported-->
        <Range>0,1</Range><!-- req, 0-50HZ; 1-60HZ -->
        <Default>0</Default><!-- req, default value -->
      </PowerLineFrequencyMode>
      <CaptureMode>
        <!--req, correspond to byCaptureMode in NET_DVR_CAMERAPARAMCFG_EX-->
        <!--req, the device supports captureModePWithIndex and
captureModeNWithIndex, when returning captureModeP and captureModeN, the client
resolves the capability set, first with captureModePWithIndex and
captureModeNWithIndex, or captureModeP and captureModeN if the former nodes are
not supported-->
        <!--req, 0-close, 1-640*480@25fps, 2-640*480@30ps, 3-704*576@25fps,
4-704*480@30fps, 5-1280*720@25fps, 6-1280*720@30fps, 7-1280*720@50fps,
8-1280*720@60fps, 9-1280*960@15fps, 10-1280*960@25fps, 11-1280*960@30fps,
12-1280*1024@25fps, 13-1280*1024@30fps, 14-1600*900@15fps, 15-1600*1200@15fps,
16-1920*1080@15fps, 17-1920*1080@25fps, 18-1920*1080@30fps, 19-1920*1080@50fps,
20-1920*1080@60fps, 21-2048*1536@15fps, 22-2048*1536@20fps, 23-2048*1536@24fps,
```

```
24-2048*1536@25fps, 25-2048*1536@30fps, 26-2560*2048@25fps, 27-2560*2048@30fps,
28-2560*1920@7.5fps, 29-3072*2048@25fps, 30-3072*2048@30fps, 31-2048*1536@12.5,
32-2560*1920@6.25, 33-1600*1200@25, 34-1600*1200@30,
35-1600*1200@12.5, 36-1600*900@12.5, 37-1600@900@15, 38-800*600@25,
39-800*600@30fps, 136-640*960@25fps, 137-640*960@24fps, 142-2992*2192@25fps,
143-2992*2192@30fps, 158-384*288@8.3fps, 159-640*512@8.3fps, 160-160*120@8.3fps,
161-1024*768@8.3fps, 162-640*480@8.3fps-->
    <captureModeP opt="close, 640*480@25fps, 640*480@30ps, 704*576@25fps,
704*480@30fps, 1280*720@25fps, 1280*720@30fps, 1280*720@50fps, 1280*720@60fps,
1280*960@15fps, 1280*960@25fps, 1280*960@30fps, 1280*1024@25fps, 1280*1024@30fps,
1600*900@15fps, 1600*1200@15fps, 1920*1080@15fps, 1920*1080@25fps, 1920*1080@30fps,
1920*1080@50fps, 1920*1080@60fps, 2048*1536@15fps, 2048*1536@20fps,
2048*1536@24fps, 2048*1536@25fps, 2048*1536@30fps, 2560*2048@25fps,
2560*2048@30fps, 2560*1920@7.5fps, 3072*2048@25fps, 3072*2048@30fps,
2048*1536@12.5fps, 2560*1920@6.25fps, 1600*1200@25fps, 1600*1200@30fps,
1600*1200@12.5fps, 1600*900@12.5fps, 1600@900@15fps, 800*600@25fps, 800*600@30fps,
640*960@25fps, 640*960@24fps"/>
    <!--req, The value of captureMode in P standard-->
    <captureModeN opt="close, 640*480@25fps, 640*480@30ps, 704*576@25fps,
704*480@30fps, 1280*720@25fps, 1280*720@30fps, 1280*720@50fps, 1280*720@60fps,
1280*960@15fps, 1280*960@25fps, 1280*960@30fps, 1280*1024@25fps, 1280*1024@30fps,
1600*900@15fps, 1600*1200@15fps, 1920*1080@15fps, 1920*1080@25fps, 1920*1080@30fps,
1920*1080@50fps, 1920*1080@60fps, 2048*1536@15fps, 2048*1536@20fps,
2048*1536@24fps, 2048*1536@25fps, 2048*1536@30fps, 2560*2048@25fps,
2560*2048@30fps, 2560*1920@7.5fps, 3072*2048@25fps, 3072*2048@30fps,
2048*1536@12.5fps, 2560*1920@6.25fps, 1600*1200@25fps, 1600*1200@30fps,
1600*1200@12.5fps, 1600*900@12.5fps, 1600@900@15fps, 800*600@25fps, 800*600@30fps,
640*960@25fps, 640*960@24fps"/>
    <!--req, The value of captureMode in N standard-->
    <captureModePWithIndex opt="0-close, 1-640*480@25fps, 2-640*480@30ps,
3-704*576@25fps, 4-704*480@30fps, 5-1280*720@25fps, 6-1280*720@30fps,
7-1280*720@50fps, 8-1280*720@60fps, 9-1280*960@15fps, 10-1280*960@25fps,
11-1280*960@30fps, 12-1280*1024@25fps, 13-1280*1024@30fps, 14-1600*900@15fps,
15-1600*1200@15fps, 16-1920*1080@15fps, 17-1920*1080@25fps, 18-1920*1080@30fps,
19-1920*1080@50fps, 20-1920*1080@60fps, 21-2048*1536@15fps, 22-2048*1536@20fps,
23-2048*1536@24fps, 24-2048*1536@25fps, 25-2048*1536@30fps, 26-2560*2048@25fps,
27-2560*2048@30fps, 28-2560*1920@7.5fps, 29-3072*2048@25fps, 30-3072*2048@30fps,
31-2048*1536@12.5fps, 32-2560*1920@6.25fps, 33-1600*1200@25fps, 34-1600*1200@30fps,
35-1600*1200@12.5fps, 36-1600*900@12.5fps, 37-1280*960@12.5fps, 38-800*600@25fps,
39-800*600@30fps, 40-4000*3000@12.5fps, 41-4000*3000@15fps, 42-4096*2160@20fps,
43-3840*2160@20fps, 44-960*576@25fps, 45-960*480@30fps, 46-752*582@25fps,
47-768*494@30fps, 48-2560*1440@25fps, 49-2560*1440@30fps, 50-720P@100fps,
51-720P@120fps, 52-2048*1536@50fps, 53-2048*1536@60fps, 54-3840*2160@25fps,
55-3840*2160@30fps, 56-4096*2160@25fps, 57-4096*2160@30fps, 58-1280*1024@50fps,
59-1280*1024@60fps, 60-3072*2048@50fps, 61-3072*2048@60fps, 62-3072*1728@25fps,
63-3072*1728@30fps, 64-3072*1728@50fps, 65-3072*1728@60fps, 66-336*256@50fps,
67-336*256@60fps, 68-384*288@50fps, 69-384*288@60fps, 70-640*512@50fps,
71-640*512@60fps, 72-2592*1944@25fps, 73-2592*1944@30fps, 74-2688*1536@25fps,
75-2688*1536@30fps, 76-2592*1944@20fps, 77-2592*1944@15fps, 78-2688*1520@20fps,
79-2688*1520@15fps, 80-2688*1520@25fps, 81-2688*1520@30fps, 82-2720*2048@25fps,
83-2720*2048@30fps, 84- 336*256@25fps, 85-384*288@25fps, 86-640*512@25fps,
87-1280*960@50fps, 88-1280*960@60fps, 89-1280*960@100fps, 90-1280*960@120fps,
```

```
91-4000*3000@20fps,141-2688*1520@12.5fps"/>
    <!--req, captureMode value with index in P standard-->
    <captureModeNWithIndex opt="0-close,1-640*480@25fps,2-640*480@30ps,
3-704*576@25fps,4-704*480@30fps,5-1280*720@25fps, 6-1280*720@30fps,
7-1280*720@50fps,8-1280*720@60fps,9-1280*960@15fps,10-1280*960@25fps,
11-1280*960@30fps,12-1280*1024@25fps,13-1280*1024@30fps,14-1600*900@15fps,
15-1600*1200@15fps, 16-1920*1080@15fps,17-1920*1080@25fps,18-1920*1080@30fps,
19-1920*1080@50fps,20-1920*1080@60fps, 21-2048*1536@15fps,22-2048*1536@20fps,
23-2048*1536@24fps,24-2048*1536@25fps,25-2048*1536@30fps, 26-2560*2048@25fps,
27-2560*2048@30fps,28-2560*1920@7.5fps,29-3072*2048@25fps,30-3072*2048@30fps,
31-2048*1536@12.5fps,32-2560*1920@6.25fps,33-1600*1200@25fps,34-1600*1200@30fps,
35-1600*1200@12.5fps, 36-1600*900@12.5fps,37-1280*960@12.5fps,38-800*600@25fps,
39-800*600@30fps,40-4000*3000@12.5fps, 41-4000*3000@15fps,42-4096*2160@20fps,
43-3840*2160@20fps,44-960*576@25fps,45-960*480@30fps, 46-752*582@25fps,
47-768*494@30fps,48-2560*1440@25fps,49-2560*1440@30fps,50-720P@100fps,
51-720P@120fps,52-2048*1536@50fps,53-2048*1536@60fps,54-3840*2160@25fps,
55-3840*2160@30fps, 56-4096*2160@25fps,57-4096*2160@30fps,58-1280*1024@50fps,
59-1280*1024@60fps,60-3072*2048@50fps, 61-3072*2048@60fps,62-3072*1728@25fps,
63-3072*1728@30fps,64-3072*1728@50fps,65-3072*1728@60fps, 66-336*256@50fps,
67-336*256@60fps,68-384*288@50fps,69-384*288@60fps,70-640*512@50fps,
71-640*512@60fps,72-2592*1944@25fps,73-2592*1944@30fps,74-2688*1536@25fps,
75-2688*1536@30fps, 76-2592*1944@20fps,77-2592*1944@15fps,78-2688*1520@20fps,
79-2688*1520@15fps,80-2688*1520@25fps, 81-2688*1520@30fps,82-2720*2048@25fps,
83-2720*2048@30fps,84- 336*256@25fps,85-384*288@25fps, 86-640*512@25fps,
87-1280*960@50fps,88-1280*960@60fps,89-1280*960@100fps,90-1280*960@120fps,
91-4000*3000@20fps,141-2688*1520@12.5fps"/>
    <!--req, captureMode value with index in N standard-->

    <!--req, to enable 3D noise reduction, SMD, rotation or WDR in
1080p50/1080p60 mode, the prompt will show "please set capture mode with normal
frame rate"-->
    <CaptureModeIndex19>
        <!--1920*1080@50fps-->
        <!--req, mutually exclusive capability, difital noise reduction, line
crossing detection, rotation, WDR access capability-->
        <mutexAbility
opt="digitalNoiseReduction,traversingVirtualPlane,fieldDetection,corridorMode,WDR"/>
    </CaptureModeIndex19>

    <CaptureModeIndex20>
        <!--req, 1920*1080@60fps-->
        <!--req, mutually exclusive capability, difital noise reduction, line
crossing detection, rotation, WDR access capability-->
        <mutexAbility
opt="digitalNoiseReduction,traversingVirtualPlane,fieldDetection,corridorMode,WDR"/>
    </CaptureModeIndex20>

    <!--req, to enable rotation or WDR in 720p50/720p60 mode, the prompt
will show "please set capture mode with normal frame rate"-->
    <CaptureModeIndex7>
```

```
<!--1280*720@50fps-->
<!--req mutually exclusive capability rotation WDR access
capability-->
  <mutexAbility opt="corridorMode,WDR"/>
</CaptureModeIndex7>

<CaptureModeIndex8>
  <!--1280*720@60fps-->
  <!--req mutually exclusive capability rotation WDR access
capability-->
    <mutexAbility opt="corridorMode,WDR"/>
  </CaptureModeIndex8>

  <CaptureModeIndex52>
    <!--2048*1536@50fps-->
    <!--req mutually exclusive capability rotation WDR access
capability-->
      <mutexAbility opt="WDR"/>
    </CaptureModeIndex52>

    <CaptureModeIndex53>
      <!--2048*1536@60fps -->
      <!--req mutually exclusive capability rotation WDR access
capability-->
        <mutexAbility opt="WDR"/>
      </CaptureModeIndex53>

      <CaptureModeIndex87>
        <!--1280*960@50fps -->
        <!--req mutually exclusive capability WDR access capability-->
          <mutexAbility opt="WDR"/>
        </CaptureModeIndex87>

        <CaptureModeIndex88>
          <!--1280*960@60fps -->
          <!--req mutually exclusive capability WDR access capability-->
            <mutexAbility opt="WDR"/>
          </CaptureModeIndex88>

        <CaptureModeIndex89>
          <!--1280*960@100fps -->
          <!--req mutually exclusive capability WDR access capability-->
            <mutexAbility opt="WDR"/>
          </CaptureModeIndex89>

        <CaptureModeIndex90>
          <!--1280*960@120fps -->
          <!--req mutually exclusive capability WDR access capability-->
            <mutexAbility opt="WDR"/>
          </CaptureModeIndex90>
        <WhiteBalance><!-- req, white balance -->
          <WhiteBalanceMode><!-- req, white balance mode -->
```

```
<!-- req, 0-Manual, 1-AWB1, 2-AWB2, 3-Automatic
Control(4~9:reserved), 11-Auto Trace,12-One Push,13-Indoor, 14-Outdoor, 15-
Outdoor Auto, 16- SodiumLight Auto -->
<!-- req, 10~16: new options for speed domes -->
<Range>1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16</Range>
<Default>1</Default><!-- req, default value -->
</WhiteBalanceMode>
<WhiteBalanceModeRGain><!-- req, R gain of white balance -->
<Min>0</Min><!-- req, minimum value -->
<Max>255</Max><!-- req, maximum value -->
<Default>100</Default><!-- req, default value -->
</WhiteBalanceModeRGain>
<WhiteBalanceModeBGain><!-- req, B gain of white balance -->
<Min>0</Min><!-- req, minimum value -->
<Max>255</Max><!-- req, maximum value -->
<Default>100</Default><!-- req, default value -->
</WhiteBalanceModeBGain>
</WhiteBalance>

<!-- req, supported by IPC only -->
<Exposure> <!-- req, exposure -->
<ExposureMode> <!-- req, exposure mode, reserved currently -->
<Range>0,1</Range><!-- req, 0-manual exposure,1-auto exposure -->
<Default>0</Default><!-- req, default value -->
</ExposureMode>

<ExposureSet>
<!-- req, exposure time, 0(index): auto*8(display on the client),
40000*8 us(the actual value)-->
<!-- req,0-auto*8(40000*8us),1-auto*5(40000*5us),2-auto*4(40000*4us),
3-auto*2(40000*2us),-->
<!-- req,4-1/25(40000us),5-1/50(20000us),6-1/100(10000us),
7-1/250(4000us),8-1/500(2000us), -->
<!-- req,9-1/750(1333us),10-1/1000(1000us),11-1/2000(500us),
12-1/4000(250us),-->
<!-- req,13-1/10,000(100us),14-1/100,000(10us), 17-1/150, 18-1/200,
20-1-1000000us, 21-1/75, 22-1/125, 23-1/175, 24-1/225,25-1/300, 26-1/400 -->
<Range>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14</Range>
<Default>4</Default><!-- req, default value-->
<DynamicAbility>
<!--req, IPC 5.1.0 supports to get or set abilities dynamicly-->
<dynamicAbilityLinkTo opt="wdrEnable,irisType"/>
<!--req, Dynamic related items, WDR enable, the structure used
for defining the type of len is NET_DVR_CAMERAPARAMCFG_EX, the parameters is
struWdr.byWDREnabled and struWdr.byIrisMode -->
</DynamicAbility>
</ExposureSet>
<exposureUSERSET><!--req, customized exposure time-->
<Min>1</Min><!--req, minimum value-->
<Max>40000</Max><!--req, maximum value-->
<Default>20000</Default><!--req, default value-->
</exposureUSERSET>
```

```

    <ExposureTarget> <!--req, reserved-->
    <Min>0</Min><!-- req, minimum value -->
    <Max>2000000</Max><!-- req, maximum value -->
    <Default>1000000</Default><!-- req, default value -->
  </ExposureTarget>
</Exposure>
<IrisMode> <!--req, Lens mode-->
  <!--req, 0- auto iris, 1- manual iris, 2- Piri1"Tamron 2.8-8mm F1.2
  (M13VP288-IR) ", 3- Union 3-9mm F1.6-2.7 (T5280-PQ1), 4- Union 2.8-12mm
  F1.6-2.7(T5289-PQ1), 5- Private 3.8-16mm F1.5 (HV3816P-8MPIR), 6-Private
  11-40mm F1.7, 7- Private 2.7-12mm F1.2 (TV2712P-MPIR), 8- MZ5721D-12MPIR, 9-
  MZ1555D-12MPIR, 10- MZ5721D-12MPIR(RS485), 11- MZ1555D-12MPIR(RS485)-->
  <Range>0,1,2,3,4,5,6,7,8,9,10,11</Range>
  <Default>1</Default><!-- req, default value -->
  <Piris>
    <!--req valid when IrisMode>=2-->
    <Piris1><!--req, Tamron 2.8-8mm F1.2 (M13VP288-IR) -->
      <modeType opt="automatic, manual"/><!--req, 0-auto, 1-manual-->
      <PIrisAperture min="" max=""/><!--req, level range: 1 to 100 (can
be configured under the manual mode)-->
    </Piris1>
    <Piris2><!--req, Union 3-9mm F1.6-2.7 (T5280-PQ1)-->
      <modeType opt="automatic, manual"/><!--req, 0-auto, 1-manual-->
      <PIrisAperture min="" max=""/><!--req, level range: 1 to 100 (can
be configured under the manual mode)-->
    </Piris2>
    <Piris3><!--req, Union 2.8-12mm F1.6-2.7 (T5289-PQ1)-->
      <modeType opt="automatic, manual"/><!--req, 0-auto, 1-manual-->
      <PIrisAperture min="" max=""/><!--req, level range: 1 to 100 (can
be configured under the manual mode)-->
    </Piris3>
    <Piris4><!--req, Private 3.8-16mm F1.5 (HV3816P-8MPIR)-->
      <modeType opt="automatic, manual"/><!--req, 0-auto, 1-manual-->
      <PIrisAperture min="" max=""/><!--req, level range: 1 to 100 (can
be configured under the manual mode)-->
    </Piris4>
    <Piris6><!--req, Private 11-40mm F1.7 (HV1140P-8MPIR)-->
      <modeType opt="automatic, manual"/><!--req, 0-auto, 1-manual-->
      <PIrisAperture min="" max=""/><!--req, level range: 1 to 100 (can
be configured under the manual mode)-->
    </Piris6>
    <Piris7><!--req, Private 2.7-12mm F1.2 (TV2712P-MPIR) -->
      <modeType opt="automatic, manual"/><!--req, 0-auto, 1-manual-->
      <PIrisAperture min="" max=""/><!--req, level range: 1 to 100 (can
be configured under the manual mode)-->
    </Piris7>
  </Piris>
</IrisMode>

<AutoApertureLevel> <!-- req, auto aperture sensitivity -->
  <Min>0</Min><!-- req, minimum value -->
  <Max>15</Max><!-- req, maximum value -->

```



```

    <Default>7</Default><!-- req, default value -->
</AutoApertureLevel>

<FocusMode> <!--req, reserved-->
    <!--req, 0- manual focus; 1-auto focus; 2- auto back focus-->
    <Range>0,1,2</Range>
    <Default>0</Default><!--req, default value-->
    <GainLevel><!--req, gain, ranges from 0 to 100-->
        <Min>0</Min><!--req, minimum value-->
        <Max>100</Max><!--req, maximum value-->
        <Default>50</Default><!-- req, default value-->
    </GainLevel>
    <BrightnessLevel><!--req, brightness, ranges from 0 to 100 -->
        <Min>0</Min><!-- req, minimum value -->
        <Max>100</Max><!-- req, maximum value -->
        <Default>50</Default><!-- req, default value -->
    </BrightnessLevel>
    <ContrastLevel><!--req, contrast, ranges from 0 to 100-->
        <Min>0</Min><!--req, minimum value-->
        <Max>100</Max><!--req, maximum value-->
        <Default>50</Default><!--req, default value-->
    </ContrastLevel>
    <SharpnessType><!--sharpness type-->
        <Range><!--sharpness type range (for speed dome): 0-automatic, 1-
manual--></Range>
        <Default><!--default value--></Default>
    </SharpnessType>
    <SharpnessLevel><!--req, sharpness-->
        <!--req, IPC(min/max); speed dome/zoom camera(Range), from 0 to 100--
>
        <Min>0</Min><!-- req, minimum value -->
        <Max>100</Max><!-- req, maximum value -->
        <Range>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17</Range>
        <!-- req, speed dome: 0- auto, 1- manual, 2-1, 3-2, 4-3, 5-4, 6-5,
7-6, 8-7, 9-8, 10-9, 11-10, 12-11, 13-12, 14-13, 15-14, 16-15, 17-16 -->
        <Default>50</Default><!-- req, default value -->
    </SharpnessLevel>
    <HorizonAperture> <!-- req, horizontal sharpness -->
        <!-- req, special for speed dome: 1~64 -->
        <Min>1</Min><!-- req, minimum value -->
        <Max>64</Max><!-- req, maximum value -->
        <Default>10</Default><!-- req, default value -->
    </HorizonAperture>
    <VerticalAperture> <!-- req, vertical sharpness -->
        <!-- req, special for speed dome: 1~64 -->
        <Min>1</Min><!-- req, minimum value -->
        <Max>64</Max><!-- req, maximum value -->
        <Default>10</Default><!-- req, default value -->
    </VerticalAperture>

    <LaserConfig>
        <controlMode opt="auto,manual"/>

```

```
<autoMode><!--req,Control Mode-->
  <sensitivity min="0" max="100"/><!--req,laser light sensitivity-->
  <triggerMode opt=" cameraModuleTrigger, photoresistanceTrigger"/>
    <!--req laser light triggering mode-->
  <limitBrightness min="0" max="100"/>
    <!--req,laser light brightness limitation-->
  <angle min="1" max="36"/><!--req,laser light angle-->
  <enable opt="true,false" />
    <!-- dep, enable manual control laser: 0- No, 1- Yes -->
  <illumination min="0" max="100" /><!-- dep, laser light strength
limit-->
  <lightAngle min="0" max="100" /><!-- dep, light angle -->
</autoMode>

<manualMode>
  <sensitivity min="0" max="100"/><!--req,laser light sensitivity-->
  <triggerMode opt=" cameraModuleTrigger, photoresistanceTrigger"/>
    <!--req laser light triggering mode-->
  <brightness min="0" max="255"/><!--req,laser light brightness-->
  <angle min="1" max="36"/><!--req,laser light angle-->
</manualMode>
</LaserConfig>

<ChromaSuppress><!-- req, color suppression -->
  <!--req, special for speed dome: 0~100 -->
  <Min>0</Min><!--req, minimum value -->
  <Max>100</Max><!--req, maximum value -->
  <Default>50</Default><!--req, default value -->
</ChromaSuppress>
<SaturationLevel><!--req, saturation, from 0 to 100 -->
  <Min>0</Min><!--req, minimum value -->
  <Max>100</Max><!--req, maximum value -->
  <Default>50</Default><!--req, default value -->
</SaturationLevel>
<HueLevel><!--req, hue, from 0 to 100-->
  <Min>0</Min><!--req, minimum value -->
  <Max>100</Max><!--req, maximum value -->
  <Default>50</Default><!--req, default value -->
</HueLevel>
<GammaCorrection><!--req, gamma correction-->
  <GammaCorrectionEnabled><!--req, 0-disable, 1-enable-->
    <Range>0,1</Range>
    <Default>0</Default><!--req, default value-->
  </GammaCorrectionEnabled>
  <GammaCorrectionLevel><!--req, the level of Gamma correction-->
    <Min>0</Min>
    <Max>100</Max>
    <Default>50</Default><!--req, default value -->
  </GammaCorrectionLevel>
</GammaCorrection>
<WDR><!--req, wide dynamic range-->
  <WDREnabled><!--req, 0-disable, 1-enable, 2-auto-->
```

```

        <Range>0,1,2</Range>
        <Default>0</Default><!--req, default value-->
    </WDREnabled>
    <isNotSupportDigitalChanCfg opt="true,false"/><!--optional, whether
setting digital channels is not supported: true=yes (not supported), false=no
(supported). If this field is not returned, it indicates that setting digital
channels is supported-->
    <WDRLevel1><!--req, level 1 of wide dynamic range, from 0 to 15-->
        <Min>0</Min><!-- req, minimum value-->
        <Max>15</Max><!-- req, maximum value-->
        <Range>0,1,2</Range><!--req, speed dome: 0- low, 1- medium, 2-
high-->
        <Default>0</Default><!--req, default value-->
    </WDRLevel1>
    <WDRLevel2><!--req, level 2 of wide dynamic range, from 0 to 15-->
        <Min>0</Min><!--req, minimum value-->
        <Max>15</Max><!--req, maximum value-->
        <Default>0</Default><!--req, default value-->
    </WDRLevel2>
    <WDRContrastLevel><!--req, contrast of wide dynamic range, from 0 to
100 -->
        <Min>0</Min><!-- req, minimum value -->
        <Max>100</Max><!-- req, maximum value -->
        <Default>50</Default><!-- req, default value -->
    </WDRContrastLevel>
</WDR>
<DayNightFilter><!--req, day and night switch -->
    <DayNightFilterType><!--req, day and night switch mode -->
        <!--req, 0- day,1- night, 2- auto, 3- timing, 4- triggered by
alarm input , 5- Auto mode 2(no photosensitivity)-->
        <Range>0,1,2,3,4</Range>
        <Default>2</Default><!--req, default value -->
    </DayNightFilterType>
    <SwitchSchedule>
        <SwitchScheduleEnabled><!--req, reserved -->
            <Range>0,1</Range><!--req, 0- disable 1- enable -->
            <Default>1</Default><!--req, default value -->
        </SwitchScheduleEnabled>
        <DayToNightFilterLevel><!--req, sensitivity of switching day to
night -->
            <Range>0,1,2,3,4,5,6,7,8,9,10,11,12</Range>
            <!--req, 0, 1, 2, 3, 4, 5, 6, 7, 10-low, 11-medium, 12-high -->
            <!-- req, (10~12: new options for speed domes) -->
            <Default>3</Default><!-- req, default value -->
        </DayToNightFilterLevel>
        <NightToDayFilterLevel><!-- req, sensitivity of switching night to
day -->
            <Range>0,1,2,3,4,5,6,7,8,9,10,11,12</Range>
            <!-- req, 0, 1, 2, 3, 4, 5, 6, 7, 10-low, 11-medium, 12-high -->
            <!-- req, (10~12: new options for speed domes) -->
            <Default>3</Default><!-- req, default value -->
        </NightToDayFilterLevel>
    </SwitchSchedule>
</DayNightFilter>

```

```

        </NightToDayFilterLevel>
        <DayNightFilterTime><!-- req, filtering time of switching day to
night -->
            <!-- req, IPC(min/max); speed dome/zoom camera(Range) -->
            <Min>10</Min><!-- req, minimum value -->
            <Max>120</Max><!-- req, maximum value -->
            <Range>0,1,2,3,4,5,6,7</Range>
            <!-- req, zoom camera/speed dome: 0-2S, 1-3S, 2-5S, 3-10S,
4-15S, 5-30S, 6-45S, 7-60S -->
            <Default>55</Default><!-- req, default value -->
        </DayNightFilterTime>
        <NightDayFilterTime>
            <!-- req, IPC(min/max); speed dome/zoom camera(Range) -->
            <Min>0</Min><!-- req, minimum value -->
            <Max>120</Max><!-- req, maximum value -->
            <Range>0,1,2,3,4,5,6,7</Range>
            <!-- req, zoom camera/speed dome:0-2S,1-3S,2-5S,3-10S,4-15S,
5-30S,6-45S,7-60S -->
            <Default>55</Default><!-- req, default value -->
        </NightDayFilterTime>
        <TimeSchedule><!--2012-08-29-->
            <BeginTime>1</BeginTime><!--req, 1 means it supports the
beginning time-->
            <EndTime>1</EndTime><!--req, 1 means it supports the ending
time-->
        </TimeSchedule>
        </SwitchSchedule>
        <AlarmInTrigType><!--2012-08-29-->
            <Range>0,1</Range><!--req, triggered status of alarm input: 0-
day, 1- night-->
        </AlarmInTrigType>
        <DayNightFilterandGain>
            <!--opt, whether to support setting day/night auto-switch and gain
simultaneously-->
            <enabled><!--req, if this function is supported, this node must
exist and be set to "true"--></enabled>
        </DayNightFilterandGain>
        </DayNightFilter>
        <Backlight><!-- req, backlight compensation -->
            <BacklightMode><!-- req, option of backlight compensation -->
                <!-- req, 0-closed, 1-UP, 2-DOWN, 3-LEFT, 4-RIGHT, 5-MIDDLE, 6-
customized, 10-open, 11-auto, 12- multi-zone backlight compensation -->
                <!-- req, (10~12: new options for speed domes, when the value is
10(open), it supports to adjust the compensation level) -->
                <Range>0,1,2,3,4,5,6,7,8,9,10,11,12</Range>
                <Default>0</Default><!-- req, default value -->
            </BacklightMode>
            <BacklightLevel><!-- req, backlight compensation level -->
                <!-- req, IPC, 0-15 -->
                <Min>0</Min><!-- req, minimum value -->
                <Max>15</Max><!-- req, maximum value -->
                <Range>0,1,2</Range>

```

```

        <!-- req, speed dome/zoom camera: 0-low, 1-medium, 2-high -->
        <Default>0</Default><!-- req, default value -->
    </BacklightLevel>
</Backlight>
<LowLightLimit> <!--req, low illumination electronic shutter-->
    <LowLightLimitEnabled><!--req, enable: 0-closed, 1-open-->
        <Range>0,1</Range>
        <Default>0</Default><!-- req, default value -->
    </LowLightLimitEnabled>
    <LowLightLimitLevel> <!-- req, the level of low illumination
electronic shutter -->
        <!-- req, speed dome 0- slow shutter*2, 1-slow shutter*3, 2-slow
shutter*4, 3-slow shutter*8, 4-slow shutter*16, 5-slow shutter*32, 6-1, 7-2,
8-3, 9-4, 10-5, 11-6, 12-low, 13-medium, 14-high -->
        <Range>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14</Range>
        <Default>0</Default><!-- req, default value -->
    </LowLightLimitLevel>
</LowLightLimit>
<ImageStabilize>
    <ImageStabilizeLevel> <!--req, image stabilization level-->
        <Range>0,1,2</Range><!--req, speed dome: 0-low, 1-medium, 2-high--
>
        <Default>0</Default><!--req, default value -->
    </ImageStabilizeLevel>
</ImageStabilize>
<CameraIRCorrection> <!--req, the movement infrared correction
function-->
    <Range>0,1,2</Range><!--req, speed dome: 0-auto, 1-open, 2-closed-->
    <Default>0</Default><!--req, default value-->
</CameraIRCorrection>

<HighSensitivitySupport>
    <!--req, whether supports setting high sensitivity: 1- suport, no
this node if not support-->
    <Range>1</Range>
</HighSensitivitySupport>
<InitializeLensSupport>
    <!--req, whether supports initializing the Lens: 1-support, no this
node if not support-->
    <Range>1</Range>
</InitializeLensSupport>
<CameraResetSupport>
    <!--req, whether supports rebooting movement: 1-support, no this
node if not support-->
    <Range>1</Range>
</CameraResetSupport>
<CameraRestoreSupport>
    <!--req, whether supports resuming movement to the factory settings:
1-support, no this node if not support-->
    <Range>1</Range>
</CameraRestoreSupport>

```

```

    <Mirror> <!--req, mirror-->
    <Range>0,1,2,3</Range><!--req, 0-off, 1-leftright, 2-updown, 3-
center-->
    <Default>0</Default><!-- req, default value -->
    </Mirror>

    <EPTZ><!-- req, E-PTZ -->
    <!-- req, 1-support, and there is no this node if not support -->
    <Range>1</Range>
    </EPTZ>

    <LOCALOUTPUT><!--req, local output-->
    <!--req, 0-not support, 1-support-->
    <!--req, 365: mini-dome and cube camera don't support local output --
>
    <!--req, 6467:BNC-0,1, 10-closed, 11-scaling output, 12-cropping
output,-->
    <!--req, 13-cropping and scaling output (10~13: special for speed
dome);-->
    <!--req, HDMI@-0:not support,20:HDMI@(720P50),21:HDMI@(720P60),
22:HDMI@(1080I60)-->
    <!--req, 23 : HDMI@(1080I50), 24 : HDMI@(1080P24), 25 :
HDMI@(1080P25),-->
    <!--req, 26:HDMI@(1080P30), 27 : HDMI@(1080P50), 28 :
HDMI@(1080P60)-->
    <Range>0,1,10,11,12,13,20,21,22,23,24,25,26,27,28</Range>
    <Default>1</Default><!-- req, default value -->
    </LOCALOUTPUT>

    <DigitalNoiseReduction><!--req, noise reduction-->
    <DigitalNoiseReductionEnable>
    <!-- req, 0-closed,1-normal mode,2-expert mode,(3~9:reserved),10-
open -->
    <!-- req, (10: new options for speed domes, when the value is
10(open), it supports to adjust noise reduction level (that is,
DigitalNoiseReductionLevel is valid) ) -->
    <Range>0,1,2,3,4,5,6,7,8,9,10</Range>
    <Default>0</Default><!-- req, default value -->
    </DigitalNoiseReductionEnable>
    <DigitalNoiseReductionLevel>
    <!--req, digital noise reduction level in normal mode, from 0 to
100-->
    <Min>0</Min><!-- req, minimum value -->
    <Max>100</Max><!-- req, maximum value -->
    <!-- req, speed dome: 0-low,1-medium,2-high,
3-1,4-2,5-3,6-4,7-5,8-6,9-7,10-8,11-9,12-10,13-11,14-12,15-13,16-14,17-15 -->
    <Range>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17</Range>
    <Default>50</Default><!-- req, default value -->
    </DigitalNoiseReductionLevel>
    <DigitalNoiseSpectralLevel>
    <!-- req, spatial digital noise reduction level in expert mode,
from 0 to 100-->

```

```

        <Min>0</Min><!-- req, minimum value -->
        <Max>100</Max><!-- req, maximum value -->
        <Default>50</Default><!-- req, default value -->
    </DigitalNoiseSpectralLevel>
    <DigitalNoiseTemporalLevel>
        <!-- req, temporal digital noise reduction level in expert mode,
from 0 to 100-->
        <Min>0</Min><!-- req, minimum value -->
        <Max>100</Max><!-- req, maximum value -->
        <Default>50</Default><!-- req, default value -->
    </DigitalNoiseTemporalLevel>
    <DigitalNoiseRemove2DEnable><!--whether to enable 2D noise reduction
for captured frames: 0-disable, 1-enable-->
        <Range>0,1</Range>
        <Default>0</Default><!--default value-->
    </DigitalNoiseRemove2DEnable>
    <DigitalNoiseRemove2DLevel><!--2D noise reduction level for captured
frames, which is between 0 and 100-->
        <Min>0</Min><!--minimum value-->
        <Max>100</Max><!--maximum value-->
        <Default>50</Default><!--default value-->
    </DigitalNoiseRemove2DLevel>
    <mutexAbility opt="1920*1080@50fps,1920*1080@60fps"/>
    <!--req, if 1920*1080@50fps or 1920*1080@60fps needs to be enabled
after 3D noise reduction, SMD, rotation, or WDR (wide dynamic range) is
enabled, users will be prompted to disable 3D noise reduction, SMD, rotation,
and WDR in the self-adaptive mode and schedule mode first. This node is
mutually exclusive with CaptureMode-->
    </DigitalNoiseReduction>
    <SceneMode><!--req, scene mode: 0-outdoor, 1-indoor-->
        <Range>0,1</Range><!--req, 0-outdoor, 1-indoor, 2-default, 3-low
light-->
        <Default>0</Default><!--req, default value-->
    </SceneMode>
    <ColorRange><!--req, color scale range-->
        <Range>0,1</Range><!--req, 0:16-235, 1:0-255-->
        <Default>0</Default><!--req, default value-->
    </ColorRange>
    <DigitalZoom><!--req, digital zoom, special for thermal network
camera-->
        <Range>0,1,2,3,4,5</Range>
        <!-- req, digital zoom: 0-closed, 1-x2, 2-x4, 3-x8, 4-x16, 5-x32 --
>
        <Default>0</Default><!--req, default value-->
    </DigitalZoom>
    <DeadPixelDetect><!--req, dead pixel detection, 1-support, and there
is no this node if not support-->
        <Range>1</Range>-->
    </DeadPixelDetect>

    <LINEENCODE><!--req, whether it supports line coding capacity: 1-
support, and there is no this node if not support-->

```

```
<Range>1</Range>
</LINEENCODE>
<!--req, whether it supports one-key focus or not: 1- support, and
there is no this node if not support-->
<OnepushFocus>1</OnepushFocus>

<Dehaze><!--req, de-haze-->
  <DehazeEnable>0,1</DehazeEnable>
  <!--req, enable de-haze mode or not: 0-no, 1-adaptive mode-->
</Dehaze>

<!--req, the following from dimmer mode to auto shutter compensation
are special for thermal network camera-->
<DimmerMode>
  <!--req, dimmer mode: 0-semiautomatic, 1-automatic-->
  <Range>0,1</Range>
  <Default>0</Default><!--req, default value-->
</DimmerMode>

<PaletteMode>
  <!-- req, palette: 0- white heat, 1-black heat, 2-palette2, ..., 8-
palette8, 9-fusion 1, 10-rainbow, 11-fusion 2, 12-iron red 1, 13-iron red 2, 14-
sepia, 15-color 1, 16-color 2, 17-ice & fire, 18-rain, 19-red hot, 20-green
hot, 21-dark blue, 22-color 3-->
  <Range>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22</
Range>
  <Default>0</Default><!-- req, default value -->
</PaletteMode>

<EnhancedMode>
  <!-- req, enhanced mode(detection object surrounding): 0- not
enhanced, 1-1, 2-2, 3-3, 4-4 -->
  <Range>0,1,2,3,4</Range>
  <Default>0</Default><!-- req, default value -->
</EnhancedMode>

<FilterSwitch>
  <!-- req, filter switch: 1-support -->
  <Range>1</Range>
</FilterSwitch>

<FocusSpeed>
  <!-- req, focus speed: 0~10 -->
  <Min>0</Min><!-- req, minimum value -->
  <Max>10</Max><!-- req, maximum value -->
  <Default>5</Default><!-- req, default value -->
</FocusSpeed>

<AutoCompensationInterval><!--req, time interval of auto shutter
compensation-->
  <!-- req, timing auto shutter compensation: 1~120, unit: minute -->
  <Min>1</Min><!-- req, minimum value -->
```



```
<Max>120</Max><!-- req, maximum value -->
<Default>60</Default><!-- req, default value -->
</AutoCompensationInterval>

<SmartIR><!--2012-08-29-->
  <Range>0,1</Range><!--req,SMART IR: 0-closed, 1-open-->
  <modeType opt="automatic, manual"><!--req,0-auto 1-manual><-req valid
when switch is on>
  <IRDistance min="" max=""/><!--req,level 1-100(can be set in manaul
mode)-->
  <ShortIRDistance min="" max=""/>
  <!--req,the level of short light 1-100(can be set in manaul mode)-->
>
  <LongIRDistance min="" max=""/>
  <!--req,the level of long light 1-100(can be set in manaul mode)-->
</SmartIR>

<Illumination><!--2012-08-29-->
  <Range>0,1</Range><!--req,low light: 0-closed, 1-open-->
</Illumination>

<LightInhibit><!--2012-08-29-->
  <LightInhibitEnable opt="true,false"/>
  <!--req, enable high light compensation: 0-closed, 1-open-->
  <isNotSupportDigitalChanCfg opt="true,false"/><!--optional, whether
setting digital channels is not supported: true=yes (not supported), false=no
(supported). If this field is not returned, it indicates that setting digital
channels is supported-->
  <level min="0" max="100"/>
  <!--req,high light compensation level-->
</LightInhibit>

<GrayLevel><!--2012-08-29-->
  <Range>0,1</Range><!--req,grayscale value range,0- [0,255], 1-
[16,235]-->
</GrayLevel>

<AutoFocusMode><!--req, focus mode of zoom camera and speed dome-->
  <FocusModeSet>
    <!-- req, focus mode: 0-auto,1-manual,2-once,3-semiautomatic -->
    <Range>0,1,2,3</Range>
    <Default>0</Default><!-- req, default value -->
  </FocusModeSet>
  <AFModeChoose>
    <!-- req, auto focus mode: 0-closed, 1-mode A, 2-mode B, 3-mode AB,
4-mode C -->
    <Range>0,1,2,3,4</Range>
    <Default>0</Default><!-- req, default value -->
  </AFModeChoose>
  <MinFocusDistance>
    <!-- req, minimum focusing distance: 0- automatic, 0xffff-
unlimited -->
```

```

<Range>0,1,2,5,10,30,50,100,150,200,300,500,600,800,1000,2000,0xffff</Range>
  <Default>0</Default><!-- req, default value -->
</MinFocusDistance>
<ZoomSpeedLevel> <!-- req, zoom speed -->
  <!-- req, 0-0, 1-1, 2-2, 3-3, 4-4, 10-low, 11-medium, 12-high -->
  <!-- req, (0-4: special for zoom camera, 10-12: special for speed
dome) -->
  <Range>0,1,2,3,4,5,6,7,8,9,10,11,12</
Range>
  <Default>0</Default><!-- req, default value -->
</ZoomSpeedLevel>
<FocusSpeedLevel>
  <!-- req, focus speed: 0-low, 1-medium, 2-high -->
  <Range>0,1,2</Range>
  <Default>0</Default><!-- req, default value -->
</FocusSpeedLevel>
</AutoFocusMode>
<assistFocus opt="true"/><!--req, whether to enable assist zoom: 0-No,
1-Yes -->
  <focusSensitivity min="0" max="2" def="1"/>
  <!--opt, focus sensitivity, ranges from 0 to 2, it is valid when the
focus mode is auto or semi-auto-->
  <relativeFocusPos min="0" max="4000" def=""/>
  <!--opt, xs:integer, relative focus sensitivity, low 16 bytes
indicate focus value (ranges from 0 to 4000), and high 16 bytes indicate
temperature value under current focus, it is valid when the focus mode is
manual or semi-auto-->
  </FocusMode>

  <AutoExposureMode>
    <!-- req, exposure and gain control of zoom camera and speed dome-->
    <ExposureSet>
      <!-- req, exposure mode: 0-manual mode, 1-auto exposure, 2-aperture
priority, 3-shutter priority, 4-gain priority -->
      <Range>0,1,2,3,4</Range>
      <Default>0</Default><!-- req, default value -->
    </ExposureSet>
    <IrisSet> <!-- req, aperture -->
      <!-- req, 0-F1.2,1-F1.4,2-F1.6,3-F1.67,4-F1.8,5-F1.85,6-F1.96,7-
F2.0,8-F2.11 -->
      <!-- req, 9-F2.2,10-F2.4,11-F2.41,12-F2.64,13-F2.8,14-F2.86,15-
F3.13,16-F3.2 -->
      <!-- req, 17-F3.4,18-F3.53,19-F3.7,20-F3.95,21-F4.0,22-F4.4,23-
F4.49,24-F4.8 -->
      <!-- req, 25-F5.35,26-F5.6,27-F6.38,28-F6.4,29-F6.8,30-F7.90,31-
F8.0,32-F8.8 -->
      <!-- req, 33-F9.6,34-F11,35-F11.06,36-F12,37-F14,38-F16,39-F16.60,40-
F17 -->
      <!-- req, 41-F19,42-F22,43-F24,44-F33.19,45-F34 -->
    </IrisSet>
  </AutoExposureMode>
  <Range>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27

```

```
,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45</Range>
  <Default>0</Default><!-- req, default value -->
</IrisSet>
<ShutterSet> <!-- req, shutter -->
  <!-- req,0-closed, 1-auto x1,2-auto x2,3-auto x4,4-auto x8,5-auto
x16,6-auto x32,-->
  <!-- req,7-auto x64, 8-auto x128, 9-1/1, 10-1/2, 11-1/3, 12-1/4,
13-1/6, 14-1/8,-->
  <!-- req,15-1/12, 16-1/15, 17-1/25, 18-1/30, 19-1/50, 20-1/60,
21-1/75, 22-1/90,-->
  <!-- req,23-1/100, 24-1/120, 25-1/125, 26-1/150, 27-1/180, 28-1/200,
29-1/215,-->
  <!-- req,30-1/250, 31-1/300, 32-1/350, 33-1/425, 34-1/500, 35-1/600,
36-1/725,-->
  <!-- req,37-1/1000, 38-1/1250, 39-1500, 40-1/1750, 41-1/2000,
42-1/2500, 43-3000,-->
  <!-- req,44-1/3500, 45-1/4000, 46-1/6000, 47-1/10000, 48-1/30000,
49-1/100000 -->

<Range>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27
,28,2/9,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49</Range>
  <Default>0</Default><!-- req, default value -->
</ShutterSet>
<GainSet><!-- req, gain: 0~100 -->
  <Min>0</Min>
  <Max>100</Max>
  <!-- req, : 0-closed, 1-low, 2-medium, 3-high, 4-0, 5-1, 6-2, 7-3,
8-4, 9-5, 10-6, 11-7, 12-8, 13-9, 14-10, 15-11, 16-12, 17-13, 18-14, 19-15 -->
  <Range>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19</Range>
  <!-- req, (0~19: special for speed dome) -->
  <Default>50</Default><!-- req, default value -->
</GainSet>
<GainLimit> <!-- req, gain limit -->
  <!-- req, speed dome -->
  <Min>0</Min><!-- req, minimum value -->
  <Max>0x0f</Max><!-- req, maximum value -->
  <Default>0</Default><!-- req, default value -->
</GainLimit>
<ExposureComp>
  <!-- req, exposure compensation: 0~100 -->
  <Min>0</Min>
  <Max>100</Max>
  <Default>50</Default><!-- req, default value -->
  <!-- req, : 0-closed, 1-low, 2-medium, 3-high, 4-0, 5-1, 6-2, 7-3,
8-4, 9-5, 10-6, 11-7, 12-8, 13-9, 14-10, 15-11, 16-12, 17-13, 18-14 -->
  <!-- req, (0~18: special for speed dome) -->
  <Range>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19</Range>
</ExposureComp>
</AutoExposureMode>

<ZoomPara>
  <!-- req, zoom parameter of zoom camera/speed dome -->
```

```
<ZoomDisplay>
  <!-- req, zoom display: 0-closed, 1-open -->
  <Range>0,1</Range>
  <Default>0</Default><!-- req, default value -->
</ZoomDisplay>
<ZoomLimit>
  <!-- req, zoom camera(Range); speed dome(min,max)-->
  <!-- req, zoom limit of zoom camera:
0-10,1-18,2-20,3-22,4-23,5-30,6-36,7-37,8-38,9-39,10-40,11-unlimited -->
  <Range>0,1,2,3,4,5,6,7,8,9,10,11</Range>
  <!-- req, optical zoom of speed dome: 1~50 -->
  <Min>1</Min>
  <Max>50</Max>
  <Default>1</Default><!-- req, default value -->
</ZoomLimit>
<DigitalZoom>
  <!-- req, zoom camera(Range);speed dome(min,max) -->
  <!-- req, digital zoom of zoom camera: 0-closed, 1- *2, 2-*4, 3-*8, 4-
*10, 5-*12 -->
  <Range>0,1,2,3,4,5</Range>
  <!-- req, digital zoom of speed dome: 0~30 -->
  <Min>1</Min>
  <Max>30</Max>
  <Default>1</Default><!-- req, default value -->
</DigitalZoom>
</ZoomPara>

<SnapExposure>
  <!-- req, exposure control general triggered snapshot -->
  <SnapMode>
    <!-- req, snapshot mode: 0- snapshot mode 1, 1- snapshot mode 2, 2-
snapshot mode 3 -->
    <Range>0,1,2</Range>
    <Default>0</Default><!-- req, default value -->
  </SnapMode>
  <SnapGain1>
    <!-- req, snapshot gain 1: 0-100 -->
    <Min>0</Min>
    <Max>100</Max>
    <Default>50</Default><!-- req, default value -->
  </SnapGain1>
  <SnapGain2>
    <!-- req, snapshot gain 2: 0-100 -->
    <Min>0</Min>
    <Max>100</Max>
    <Default>50</Default><!-- req, default value -->
  </SnapGain2>
</SnapExposure>

<!--req, dynamic contrast ratio level of intelligent traffic camera-->
<DynamicContrast>
  <DynamicContrastLevel>
```

```

        <Min>0</Min>
        <Max>100</Max>
        <Default>50</Default><!--req, default value-->
    </DynamicContrastLevel>
</DynamicContrast>

<!--req, Rotation Mode-->
<CorridorMode>
    <corridorModeFunEnable opt="true,false"/>
    <!--req,Enable or not, true-Enable, false-Disable-->
    <!--req,If enable 1080p50/1080p60 after the 3D DNR, SMD, rotation or
WDR is enabled, the prompt will pop up "Close the 3D DNR, SMD, rotation and WDR
under the self-adaptive and continuous mode first."-->
    <!--req,If enable 720p50/720p60 after the rotation or WDR is enabled,
the prompt will pop up "Close rotation and WDR under the self-adaptive and
continuous mode first."-->
    <mutexAbility opt="19-1920*1080@50fps,20-1920*1080@60fps,
8-1280*720@60fps,7-1280*720@50fps"/>
    <!--req Mutex among the CaptureMode 1920*1080@50fps, 1920*1080@60fps,
1280*720@60fps and 1280*720@50fps, prompt will pop up when enabled any of the
modes.-->
</CorridorMode>

<ISPAAdvanceCfg><!--req,ISP supports return or not.-->
    <ISPSupportMode opt="dayMode,nightMode"/>
    <!--req Mode supported by ISP-->
    <workMode opt="auto,schedule"/>
    <!--req,0-Auto,1-Scheduled switch-->
    <TimeSchedule>
        <beginTime opt="hour,min,sec,millisecond"/>
        <!--req Type of start time period-->
        <endTime opt="hour,min,sec,millisecond"/>
        <!--req Type of end time period-->
    </TimeSchedule>
    <ISPCfgSupport
opt="whiteBalanceMode,whiteBalanceModeRGain,whiteBalanceModeBGain,exposureSet,
exposureUserSet,gainLevel,brightnessLevel,contrastLevel,sharpnessLevel,WDREnable
d,
WDRLevel1,WDRLevel2,WDRContrastLevel,backlightMode,backlightLevel,position1,
position2,imageStabilizeEnable,imageStabilizeLevel,digitalNoiseReductionEnable,
digitalNoiseReductionLevel,digitalNoiseSpectralLevel,digitalNoiseTemporalLevel,
dehazeEnable,dehazeLevel,lightInhibitEnable,lightInhibitLevel,grayLevel"/>
</ISPAAdvanceCfg>

    <supportCCDFunc
opt="whiteBalance,exposure,WDR,dayNightFilter,gammaCorrection,digitalNoiseReduct
ion,

```

```
backLight,lowLight,focus,infrared,domeAemode,dehaze,parkAction,elecStab,other,ISP,laser"/>
    <!--req supported front-end parameter capabilities.-->

    <!--req, Illumination Enhancement Capture Camera v3.5-->
    <BrightCompensate>
        <brightCompensate min="0" max="100"/>
    </BrightCompensate>

    <!--req, Exposure Control Capture Camera v3.5-->
    <ExposureSegment>
        <exposureSegmentEnable opt="true,false"/>
        <!--req,Enable or not, true-Enable, false-Disable-->
    </ExposureSegment>

    <LensDistortionCorrection>
        <enable opt="false,true" default="false"/>
        <!--req,Lens Distortion Correction (0-Disable/1-Enable)-->
        <mutexAbility opt="WDR"/>
        <!--req,mutex in WDR -->
        <level min="1" max="3">
            <!--opt, xs:integer, Distortion Correction Level, 1-3-->
        </level>
        <zoomedInDistantView>
            <!--dep, xs:integer, Remote zoom, takes effect when Distortion
Correction is enabled.-->
            <enabled>
                <!--req, xs:bool, "true,false"-->
            </enabled>
            <level min="1" max="3">
                <!--opt, xs:integer, Correction level of remote zoom, 1-3-->
            </level>
        </zoomedInDistantView>
        <horizontalFOV min="0" max="100">
            <!--opt, xs:integer, Horizontal FOV[0-100]-->
        </horizontalFOV>
        <verticalFOV min="0" max="100">
            <!--opt, xs:integer, Vertical FOV[0,100]-->
        </verticalFOV>
    </LensDistortionCorrection>
    <BrightnessSuddenChangeSuppressionCap>
        <enabled opt="true,false">
            <!--req, xs:boolean, brightness sudden change suppression-->
        </enabled>
    </BrightnessSuddenChangeSuppressionCap>

    <DPCParam>
        <!--req,DPC-->
        <ctrltype
opt="correct,cancelCorrect,crossDisplayOpen,crossDisplayClose,point,up,down,right,
left,allCorrect,save"/>
```

```
<!--req,correction, cancel correction, enable/disable DPC cross
display, DPC coordinate, up-forward offset of DPC coordinate,
down-forward offset of DPC coordinate, right-forward offset of DPC
coordinate, left-forward offset of DPC coordinate, DPC all, save defective
Pixel-->
    <dpcMode opt="manual,auto" def="auto"/>
    <!--req,xs:string,"manual-Manual Correction, auto-Auto Correction, if
device does not support this node, all will be handled manually"-->
</DPCParam>

<FFCParam>
    <mode opt="schedule,temperature"/>
    <!--req,1-Continuous mode, 2-Temperature difference mode, 3-Close-->
    <ScheduleMode>
        <compensateTime opt="10,20,30,40,50,60,120,180,240"/>
        <compensateTimeUnit opt="min"/>
        <!--req,min-->
    </ScheduleMode>
    <FFCManualCtrl opt="true"/>
    <!--req,FFC Manual Control-->
    <FFCBackCompCtrl opt="true"/>
    <!--req,FFC Background Compensation control-->
</FFCParam>

<DDEParam>
    <mode opt="off,normal,expert"/>
    <!--req,1-Close, 2-Normal Mode, 3-Expert Mode-->
    <normalLevel min="1" max="100"/>
    <!--req,Level settings under normal mode-->
    <expertLevel min="1" max="100"/>
    <!--req,Level settings under expert mode-->
</DDEParam>

<AGCParam>
    <scene opt="normal,highlight>manual"/>
    <!--req,1-Normal scene, 2-Highlight scene, 3- Manual scene-->
    <ManualMode>
        <lightLevel min="1" max="100"/>
        <!--req,Brightness level-->
        <gainLevel min="1" max="100"/>
        <!--req,Gain level-->
    </ManualMode>
</AGCParam>
    <fusionMode/> <!--opt, xs:string, visual and thermal image fusion mode:
"thermal"-thermal mode, "fusion"-fusion mode, "PIP"-picture in picture mode,
"Visible"-visible mode, "fusionB/W"-black and white fusion mode, "city",
"jungle", "desert", "sea", "snow"-->
    <ThermometryAGC>
        <mode opt = "close,auto>manual">
            <!--opt, xs:string-->
        </mode>
        <highTemperature min="-273" max="10000">
```

```
<!--dep, xs:integer-->
</highTemperature>
<lowTemperature min="-273" max="10000">
  <!--dep, xs:integer-->
  </lowTemperature>
</ThermometryAGC>

<isSupportGPSControl>
  <!--optional, boolean, whether the device supports GPS control
capability-->
  </isSupportGPSControl >

  <gearRange>
    <!--optional, xs:integer, the number of ranges supported by the device,
e.g., when the value is 3,it indicates supported three ranges-->
    <gearRange>

  </ChannelEntry>
</ChannelList>
</CAMERAPARA>
```

A.3.201 XML_Cap_AlarmLampConfig

Message about the alarm lamp configuration capability in XML format.

```
<AlarmLampConfig version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <enabled><!--required, xs:boolean, whether to enable schedule alarm lamp
flickering--></enabled>
  <flashDuration min="" max=""><!--required, xs:integer, scheduled flickering
duration of the alarm lamp, unit: second--></flashDuration>
  <flashIntervalTime min="" max=""><!--required, xs:integer, alarm lamp
flickering interval, unit: second--></flashIntervalTime>
</AlarmLampConfig>
```

A.3.202 XML_Cap_AudioFileList

Message about the audio file list.

```
<AudioFileList version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <AudioFile>
    <type opt="callWaitting,consultWaitting"><!--required, xs:string, audio
file type: "callWaitting", "consultWaitting"--></type>
    <name min="" max=""><!--required, xs:string, file name--></name>
  </AudioFile>
</AudioFileList>
```


A.3.203 XML_Cap_AudioInOutCfg

Message about the configuration capability of the audio input and output in XML format.

```
<AudioInOutCfg version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <audioInManualCfgEnable opt="true,false"><!--optional, xs:boolean, whether to
enable configuring audio input manually--></audioInManualCfgEnable>
  <audioOutManualCfgEnable opt="true,false"><!--optional, xs:boolean, whether
to enable configuring audio output manually--></audioOutManualCfgEnable>
  <Intercom>
    <AudioInType opt="micIn,lineIn1,list"><!--required, xs:string, audio input
type: "micIn", "lineIn1", "list". If this node is set to "list", the device
will parse the volume in AudioList--></AudioInType>
    <AudioInList>
      <AudioIn>
        <id opt="micIn,lineIn1"><!--required, xs:string, audio input ID:
"micIn", "lineIn1"--></id>
        <isConfiged><!--optional, xs:boolean, whether the device type is
configured: true, false. If the device type of the device corresponding to the
ID is configured and applied, this node should be set to true--></isConfiged >
        <volume min="0" max="10"><!--required, xs:integer, volume level, value
range: [0, 10], 0 means the device is muted, and the default volume is 6--></
volume>
      </AudioIn>
    </AudioInList>
    <AudioOutType
opt="spkOut,spkOut1,spkOut2,lineOut1,spkOut_lineOut1,list"><!--required,
xs:string, audio output type: "spkOut", "spkOut1", "spkOut2", "lineOut1",
"spkOut_lineOut1", "list"--></AudioOutType>
    <AudioOutList>
      <AudioOut>
        <id opt="spkOut,spkOut1,spkOut2,lineOut1,spkOut_lineOut1"><!--required,
xs:string, audio output ID: "spkOut", "spkOut1", "spkOut2", "lineOut1",
"spkOut_lineOut1"--></id>
        <isConfiged><!--optional, xs:boolean, whether the device type is
configured: true, false. If the device type of the device corresponding to the
ID is configured and applied, this node should be set to true--></isConfiged >
        <volume min="0" max="10"><!--required, xs:integer, volume level, value
range: [0, 10], 0 means the device is muted, and the default volume is 6--></
volume>
      </AudioOut>
    </AudioOutList>
  </Intercom>
  <Broadcast>
    <AudioOutType
opt="spkOut,spkOut1,spkOut2,lineOut1,spkOut_lineOut1,list"><!--required,
xs:string, audio output type: "spkOut", "spkOut1", "spkOut2", "lineOut1",
"spkOut_lineOut1", "list"--></AudioOutType>
    <AudioOutList>
      <AudioOut>
        <id opt="spkOut,spkOut1,spkOut2,lineOut1,spkOut_lineOut1"><!--required,
```

```
xs:string, audio output ID: "spkOut", "spkOut1", "spkOut2", "lineOut1",
"spkOut_lineOut1"--></id>
    <isConfiged><!--optional, xs:boolean, whether the device type is
configured: true, false. If the device type of the device corresponding to the
ID is configured and applied, this node should be set to true--></isConfiged >
    <volume min="0" max="10"><!--required, xs:integer, volume level, value
range: [0, 10], 0 means the device is muted, and the default volume is 6--></
volume>
    </AudioOut>
</AudioOutList>
</Broadcast>
<VoicePrompt>
    <AudioOutType opt="spkOut1,spkOut2,lineOut1,spkOut_lineOut1,list"><!--
required, xs:string, audio output type: "spkOut1", "spkOut2", "lineOut1",
"spkOut_lineOut1", "list"--></AudioOutType>
    <AudioOutList>
        <AudioOut>
            <id opt="spkOut1,spkOut2,lineOut1,spkOut_lineOut1"><!--required,
xs:string, audio output ID: "spkOut1", "spkOut2", "lineOut1",
"spkOut_lineOut1"--></id>
            <isConfiged><!--optional, xs:boolean, whether the device type is
configured: true, false. If the device type of the device corresponding to the
ID is configured and applied, this node should be set to true--></isConfiged >
            <volume min="0" max="10"><!--required, xs:integer, volume level, value
range: [0, 10], 0 means the device is muted, and the default volume is 6--></
volume>
            </AudioOut>
        </AudioOutList>
    </VoicePrompt>
</AudioInOutCfg>
```

A.3.204 XML_Cap_DetectorCfg

DetectorCfg capability message in XML format

```
<DetectorCfg version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
    <DetectorParamList>
        <DetectorParam>
            <enabled><!--req, xs: boolean, enable the zone linked detector or not--></
enabled>
            <detectorSerialNo min="" max=""
opt="0,1,2,3,4,5,6,7,8,9,Q,R,S,T,U,V,W,X,Y,Z">
                <!--req, xs:string, zone linked detector serial No.-->
            </detectorSerialNo>
            <detectorHeartBeatCycle opt="5min,10min,20min,30min,60min,2hour,4hour,
6hour,0-none">
                <!--req, xs:string, detector heartbeat period (5min,10min,20min,30min,
60min,2hour,4hour,6hour,no heartbeat time)-->
            </detectorHeartBeatCycle>
        </DetectorParam>
```

```
</DetectorParamList>
</DetectorCfg>
```

A.3.205 XML_Cap_LampSchedTimeList

Message about the configuration capability of the alarm lamp flickering schedule in XML format.

```
<LampSchedTimeList version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <enabled><!--required, xs:boolean, whether to enable the alarm lamp
flickering schedule--></enabled>
  <schedTimeMax min="" max=""><!--required, xs:integer, maximum number of time
periods that can be configured for each day--> </schedTimeMax>
  <index><!--required, xs:string, the day of the week that the schedule can be
configured for: "1", "2", "3", ...--> </index>
</LampSchedTimeList>
```

A.3.206 XML_Cap_ZoneAssociatedDetectorCfg

ZoneAssociatedDetectorCfg capability message in XML format

```
<ZoneAssociatedDetectorCfg version="2.0" xmlns="http://www.isapi.org/ver20/
XMLSchema">
  <DetectorCfgList><!--up to 64-->
    <DetectorCfg>
      <enabled>
        <!--req, xs: boolean, enable zone linked detector or not -->
      </enabled>
      <detectorSerialNo min="" max=""
opt="0,1,2,3,4,5,6,7,8,9,Q,R,S,T,U,V,W,X,Y,Z">
        <!--req, xs:string, the serial No. of zone linked detector -->
      </detectorSerialNo>
    </DetectorCfg>
  </DetectorCfgList>
</ZoneAssociatedDetectorCfg>
```

A.3.207 XML_Desc_AlarmHostAbility

Input description message for getting network security control panel capability.

```
<AlarmHostAbility version="2.0" >
  <!--optional, you can specify the node to return, and the specified node can
be the child node of the root node-->
  <ExtendAlarmbox/><!--optional, specify the node <ExtendAlarmbox> to return-->
</AlarmHostAbility>
```

A.3.208 XML_DeviceCap

XML message about device capability

```
<DeviceCap version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <SysCap><!--optional-->
    <isSupportDst><!--optional, xs: boolean, whether it supports daylight
saving time--></isSupportDst>
    <NetworkCap/><!--optional, xs: boolean, network capability-->
    <IOCap/><!--optional, IO capability-->
    <SerialCap/><!--optional, serial port capability-->
    <VideoCap/><!--optional, video capability, see details in the message of
XML_VideoCap-->
    <AudioCap/><!--optional, audio capability-->
    <isSupportHolidy><!--optional, xs:boolean--></isSupportHolidy>
    <RebootConfigurationCap>
      <Genetec><!--optional, xs:boolean--></Genetec>
      <ONVIF><!--optional, xs:boolean--></ONVIF>
      <RTSP><!--optional, xs:boolean--></RTSP>
      <HTTP><!--optional, xs:boolean--></HTTP>
      <SADP>
        <ISDiscoveryMode><!--optional, xs:boolean--></ISDiscoveryMode>
        <PcapMode><!--optional, xs:boolean--></PcapMode>
      </SADP>
      <IPCAddStatus><!--optional, xs:boolean--></IPCAddStatus>
    </RebootConfigurationCap>
    <isSupportExternalDevice><!--optional, xs:boolean--></
isSupportExternalDevice>
    <isSupportChangedUpload>
      <!--optional, xs: boolean, whether it supports uploading status changes-->
    </isSupportChangedUpload>
    <isSupportGettingWorkingStatus>
      <!--optional, xs:boolean, whether it supports getting device status-->
    </isSupportGettingWorkingStatus>
    <isSupportGettingChannelInfoByCondition>
      <!--optional, xs:boolean-->
    </isSupportGettingChannelInfoByCondition>
    <isSupportDiagnosedDataParameter>
      <!--optional, xs:boolean-->
    </isSupportDiagnosedDataParameter>
    <isSupportSimpleDevStatus>
      <!--optional, xs: boolean, whether it supports getting device working
status-->
    </isSupportSimpleDevStatus>
    <isSupportFlexible>
      <!--optional, xs: boolean, whether it supports getting channel status by
condition-->
    </isSupportFlexible>
    <isSupportPTZChannels>
      <!--optional, xs:boolean, whether it supports returning PTZ channel
(which is different from the video channel)-->
```

```
</isSupportPTZChannels>
<isSupportSubscribeEvent>
  <!--optional, xs:boolean, whether it supports alarm or event
subscription: "true,false"-->
</isSupportSubscribeEvent>
<isSupportDiagnosedData>
  <!--optional, xs:boolean, "true,false", whether it supports diagnosis
data-->
</isSupportDiagnosedData>
<isSupportTimeCap>
  <!--optional, xs:boolean, whether it supports time capability-->
</isSupportTimeCap>
<isSupportThermalStreamData>
  <!--optional, xs:boolean, whether it supports uploading thermal stream
data in real-time. If it is supported, the returned value is "true"; otherwise,
this node will not be returned-->
</isSupportThermalStreamData>
<isSupportPostUpdateFirmware>
  <!--optional,xs:boolean,"true,false", whether it supports upgrading the
firmware-->
</isSupportPostUpdateFirmware>
<isSupportPostConfigData>
  <!--optional, xs:boolean,"true,false", whether it supports importing or
exporting the configuration file-->
</isSupportPostConfigData>
<isSupportUserLock>
  <!--optional, xs:boolean,"true,false", whether it supports locking user-->
</isSupportUserLock>
<isSupportModuleLock><!--optional, xs:boolean, whether it supports locking
the module: "true,false"--></isSupportModuleLock>
<isSupportSoundCfg><!--optional, xs:boolean--></isSupportSoundCfg>
<isSupportMetadata>
  <!--optional, xs:boolean, if it is supported, return "true", otherwise,
this node will not be returned-->
</isSupportMetadata>
<isSupportShutdown><!--optional, xs:boolean, whether it supports shutdown
configuration--></isSupportShutdown>
<supportSmartOverlapChannles opt="1"/><!--optional, xs:boolean, whether it
supports stream configuration of smart events. If this function is supported,
this node and the corresponding channel ID will be returned; otherwise, this
node will not be returned-->
<isSupportConsumptionMode><!--optional, xs:boolean, whether it supports
switching power consumption mode:true (yes), this node is not returned (no).
Related URI: /ISAPI/System/consumptionMode/capabilities?format=json--></
isSupportConsumptionMode>
<isSupportManualPowerConsumption><!--optional, xs:boolean, whether it
supports control the power consumption mode manually: true (yes), this node is
not returned (no)--></isSupportManualPowerConsumption>
</SysCap>
<voicetalkNums><!--optional, xs:integer, the number of two-way audio
channels--></voicetalkNums>
<isSupportSnapshot><!--optional, xs:boolean, whether it supports capture:
```

```
"true, false"--></isSupportSnapshot>
  <SecurityCap/><!--optional, security capability-->
  <EventCap/><!--optional, event capability-->
  <ITCCap><!--optional--></ITCCap>
  <ImageCap/><!--optional, image capability-->
  <RacmCap/><!--optional, storage capability-->
  <PTZCtrlCap>
    <isSupportPatrols><!--optional, xs:boolean--></isSupportPatrols>
    <isSupportCombinedPath><!--optional, xs:boolean, whether the device
supports the PTZ combined path-->true</isSupportCombinedPath>
  </PTZCtrlCap>
  <SmartCap/><!--optional, intelligent capability-->
  <isSupportEhome><!--optional, xs:boolean--></isSupportEhome>
  <isSupportStreamingEncrypt><!--optional, xs:boolean--></
isSupportStreamingEncrypt>
  <TestCap>
    <isSupportEmailTest><!--optional, xs:boolean--></isSupportEmailTest>
  </TestCap>
  <ThermalCap/><!--optional, temperature measurement capability-->
  <WlAlarmCap/><!--optional, wireless alarm capability-->
  <SecurityCPCapabilities/><!--optional, security control panel capability-->
  <isSupportGIS>
    <!--optional, xs:boolean, whether it supports GIS capability-->
  </isSupportGIS>
  <isSupportCompass>
    <!--optional, xs:boolean-->
  </isSupportCompass>
  <isSupportRoadInfoOverlays>
    <!--optional, xs:boolean-->
  </isSupportRoadInfoOverlays>
  <isSupportFaceCaptureStatistics>
    <!--optional, xs:boolean-->
  </isSupportFaceCaptureStatistics>
  <isSupportExternalDevice>
    <!--optional, xs:boolean-->
  </isSupportExternalDevice>
  <isSupportElectronicsEnlarge>
    <!--optional, xs:boolean, whether it supports digital zoom-->
  </isSupportElectronicsEnlarge>
  <isSupportRemoveStorage>
    <!--optional, xs:boolean-->
  </isSupportRemoveStorage>
  <isSupportCloud>
    <!--optional, xs:boolean-->
  </isSupportCloud>
  <isSupportRecordHost>
    <!--optional, xs:boolean-->
  </isSupportRecordHost>
  <isSupportEagleEye>
    <!--optional, xs:boolean, whether it supports PanoVu series camera-->
  </isSupportEagleEye>
  <isSupportPanorama>
```

```
<!--optional, xs:boolean, whether it supports panorama-->
</isSupportPanorama>
<isSupportFirmwareVersionInfo>
  <!--optional, xs:boolean, whether it supports displaying firmware version
information-->
  </isSupportFirmwareVersionInfo>
<isSupportExternalWirelessServer>
  <!--optional, xs: boolean-->
</isSupportExternalWirelessServer>
<isSupportSetupCalibration>
  <!--optional, xs:boolean, whether it supports setting calibration-->
</isSupportSetupCalibration>
<isSupportGetmutexFuncErrMsg>
  <!--optional, xs:boolean, whether it supports getting mutex information-->
</isSupportGetmutexFuncErrMsg>
<isSupportTokenAuthenticate><!--optional, xs:boolean--></
isSupportTokenAuthenticate>
<isSupportStreamDualVCA><!--optional, xs:boolean--></isSupportStreamDualVCA>
<isSupportlaserSpotManual>
  <!--optional, boolean, whether it supports laser spot configuration-->
</isSupportlaserSpotManual>
<isSupportRTMP><!--optional, xs:boolean--></isSupportRTMP>
<isSupportTraffic><!--optional, xs:boolean--></isSupportTraffic>
<isSupportLaserSpotAdjustment>
  <!--optional, boolean, whether it supports adjusting laser spot size-->
</isSupportLaserSpotAdjustment>
<VideoIntercomCap/><!--optional, video intercom capability-->
<isSupportSafetyCabin>
  <!--optional, xs:boolean-->
</isSupportSafetyCabin>
<isSupportPEA>
  <!--optional, xs:boolean, whether it supports one-touch security control
panel capability-->
</isSupportPEA>
<isSupportCurrentLock>
  <!--optional, xs:boolean, whether it supports locking current
configuration-->
</isSupportCurrentLock>
<isSupportGuardAgainstTheft>
  <!--optional, xs:boolean, whether it supports device anti-theft
configuration-->
</isSupportGuardAgainstTheft>
<isSupportPicInfoOverlap>
  <!--optional, xs:boolean, whether it supports picture information overlay-->
</isSupportPicInfoOverlap>
<isSupportPlay>
  <!--optional, xs: boolean, whether it supports live view: "true,false"-->
</isSupportPlay>
<isSupportPlayback>
  <!--optional, xs: boolean, whether it supports playback: "true,false"-->
</isSupportPlayback>
<UHF RFIDReader>
```

```
<!--optional, supported capability of UHF RFID card reader-->
<isSupportBasicInformation>
  <!--optional, xs:boolean, whether it supports basic parameters of UHF
RFID card reader-->
</isSupportBasicInformation>
<isSupportHardDiskStorageTest>
  <!--optional, xs:boolean, whether it supports hard disk storage test of
UHF RFID card reader-->
</isSupportHardDiskStorageTest>
</UHFRFIDReader>
<isSupportIntelligentStructureAnalysis>
  <!--optional, xs:boolean, whether it supports structured VCA-->
</isSupportIntelligentStructureAnalysis>
<isSupportIntelligentAnalysisEngines>
  <!--optional, xs:boolean, whether it supports VCA engine configuration-->
</isSupportIntelligentAnalysisEngines>
<PreviewDisplayNum>
  <!--optional, xs:integer, the number of live view windows, which is the
number of simultaneous live view windows controlled by the device. Limited by
the performance of DeepinMind series network video recorder, currently only
live view of a network camera is supported, and playback is not supported-->
</PreviewDisplayNum>
<isSupportBoard opt="true,false">
  <!--optional, xs:boolean, whether it supports protocol related to sub-
board-->
</isSupportBoard>
<ResourceSwitch>
  <workMode opt="4KPreview,educationRecord">
    <!--req, xs:string, device working mode: "4KPreview"-4K live view mode,
"educationRecord"-education recording mode-->
  </workMode>
</ResourceSwitch>
<isSupportCustomStream><!--optional, xs:boolean--></isSupportCustomStream>
<isSupportTriggerCapCheck>
  <!--optional, xs:boolean, whether it supports verifying capability of alarm
linkage actions-->
</isSupportTriggerCapCheck>
<isSupportActiveMulticast>
  <!--optional, xs: boolean, whether it supports active multicast-->
</isSupportActiveMulticast>
<isSupportChannelEventCap>
  <!--optional, xs:boolean, whether it supports getting event capability by
channel-->
</isSupportChannelEventCap>
<isSupportPictureServer>
  <!-- opt, xs:boolean, whether it supports picture storage server-->
</isSupportPictureServer>
<isSupportVideoCompositeAlarm>
  <!--optional, xs:boolean, whether it supports video double check alarm-->
</isSupportVideoCompositeAlarm>
<isSupportSensorCalibrating>
  <!--optional, xs:boolean, whether it supports double sensor calibration-->
```



```
</isSupportSensorCalibrating>
<isSupportChannelEventListCap>
  <!--optional, xs:boolean, whether it supports getting event capability of
all channels-->
</isSupportChannelEventListCap>
<VCAResourceChannelsCap>
  <!--optional, whether it supports independently switching to another VCA
resource by channel-->
  <ChannelsList>
    <channelsID>
      <!--req, xs:integer, channel No. supported by the device-->
    </channelsID>
  </ChannelsList>
</VCAResourceChannelsCap>
<SensorCap/><!--optional, intelligent cabinet capability-->
<isSupportSecurityCP/>
  <!--optional, xs:boolean, whether it supports the applications of security
control panel: "true, false"-->
</isSupportSecurityCP>
<isSupportClientProxyWEB>
  <!--optional, xs:boolean, whether it supports the function that the client
proxy passes through the remote web configuration: "true"-->
</isSupportClientProxyWEB>
<WEBLocation>
  <!--optional, string type, web page location: "local"-local device,
"remote"-remote location. If this node is not returned, the web page will be in
the local device by default-->
</WEBLocation>
<isSupportTime/>
  <!--optional, xs:boolean, "true, false", whether it supports time
configuration-->
</isSupportTime>
<isSupportTimeZone/>
  <!--optional, xs:boolean, "true, false", whether it supports daylight
saving time (DST) configuration-->
</isSupportTimeZone>
<isSupportCityManagement>
  <!--optional, boolean, ro, whether it supports intelligent city management--
>true
</isSupportCityManagement>
<isSupportMixedTargetDetection>
  <!--optional, xs:boolean, "true, false", whether it supports multi-target-
type detection-->
</isSupportMixedTargetDetection>
<isSupportFaceContrastMode>
  <!--optional, xs:boolean, whether it supports face picture comparison mode--
>
</isSupportFaceContrastMode>
<isSupportPictureCaptureComparision>
  <!--optional, xs:boolean, whether it supports face picture N:1 comparison
between face pictures captured by the camera and imported face pictures-->
</isSupportPictureCaptureComparision>
```

```
<isSupportGPSCalibratation>
  <!--optional, xs:boolean, whether it supports GPS calibration capability-->
</isSupportGPSCalibratation>
<isSupportChannelFullEventListCap>
  <!--optional, xs:boolean, whether it supports getting event list capability
of all channels-->
</isSupportChannelFullEventListCap>
<isSupportAUXInfoCap>
  <!--optional, xs:boolean, whether it supports getting property capability
of all channels-->
</isSupportAUXInfoCap>
<isSupportCalibrationFile>
  <!--optional, xs:boolean, whether it supports importing calibration file-->
</isSupportCalibrationFile>
<isSupportDisplayTrajectory>
  <!--optional, xs:boolean, whether it supports displaying trajectory-->
</isSupportDisplayTrajectory>
<maximumSuperPositionTime opt="5,10,20,30">
  <!--dep,xs:integer, the maximum time of trajectory displaying, unit:
second, it is valid only when displaying trajectory is supported-->
</maximumSuperPositionTime>
<isSupportUnitConfig>
  <!--optional, xs:boolean, whether it supports unit configuration-->
</isSupportUnitConfig>
<isSupportAutoMaintenance>
  <!--optional, xs:boolean, whether it supports automatic maintenance. When
this node exists and values "true", it indicates support-->
</isSupportAutoMaintenance>
<isSupportGetLinkSocketIP>
  <!--optional, xs: boolean, "true,false", whether it supports getting the
SocketIP of current connection-->
</isSupportGetLinkSocketIP>
<isSupportIntelligentSearch>
  <!--optional, xs:boolean, whether it supports intelligent search-->
</isSupportIntelligentSearch>
<IOTCap><!--optional, xs:boolean, IoT device access capability-->
  <supportChannelNum>
    <!--req, xs:integer, number of supported channels of IoT device-->
  </supportChannelNum>
  <startChannelNo>
    <!--optional, xs:integer, initial channel ID, if this node is not
inputted, it indicates that the initial channel ID is 1-->
  </startChannelNo>
  <isSupportlinkageChannelsSearch>
    <!--optional, boolean, returns "true" if support, returns "false" if not
support-->
  </isSupportlinkageChannelsSearch>
</IOTCap>
<isSupportEncryption>
  <!--optional, xs: boolean, stream encryption capability-->
</isSupportEncryption>
<AIDEventSupport opt="abandonedObject, pedestrian, congestion, roadBlock,
```

```
construction, trafficAccident, fogDetection, wrongDirection, illegalParking,
SSharpDriving, lowSpeed, dragRacing">
  <!--optional, xs:string, supported traffic incident type: "abandonedObject"-
objects dropped down, "pedestrian"-pedestrian, "congestion"-congestion,
"roadBlock"-roadblock, "construction"-construction, "trafficAccident"-traffic
accident, "fogDetection"-fog, "wrongDirection"-wrong-way driving,
"illegalParking"-illegal parking, "SSharpDriving"-slalom driving, "lowSpeed"-
driving in low speed, "dragRacing"-street racing-->
  </AIDEventSupport>
  <TFSEventSupport
opt="illegalParking ,wrongDirection,crossLane,laneChange,vehicleExist,turnRound,
parallelParking,notKeepDistance,notSlowZebraCrossing,overtakeRightSide,lowSpeed,
dragRacing,changeLaneContinuously,SSharpDriving,largeVehicleOccupyLine,jamCrossL
ine">
  <!--optional, xs:string, supported enforcement event type: "illegalParking"-
illegal parking, "wrongDirection"-wrong-way driving, "crossLane"-driving on the
lane line, "laneChange"-illegal lane change, "vehicleExist"-motor vehicle on
non-motor vehicle lane, "turnRound"-illegal U-turn, "parallelParking"-parallel
parking, "notKeepDistance"-not keeping vehicle distance, "notSlowZebraCrossing"-
not slowing down at zebra corssing, "overtakeRightSide"-overtaking on the
right, "lowSpeed"-driving in low speed, "dragRacing"-street racing,
"changeLaneContinuously"-continuous lane change, "SSharpDriving"-slalom
driving, "largeVehicleOccupyLine"-lane occupation by large-sized vehicle,
"jamCrossLine"-queue jumping-->
  </TFSEventSupport>
  <isVehicleStatisticsSupport>
    <!--optional, xs: boolean, whether it supports setting parameters for
traffic data collection-->
  </isVehicleStatisticsSupport>
  <isSupportIntersectionAnalysis>
    <!--optional, xs: boolean, whether it supports intersection analysis-->
  </isSupportIntersectionAnalysis>
  <supportRemoteCtrl
opt="up,down,left,right,enter,menu,num,power,esc,edit,F1,.prev,rec,play,stop,not
Support"/><!--whether it supports remote control-->
  <isSptDiagnosis>
    <!--optional, xs:boolean, whether it supports device diagnosis: "true",
"false"-->
  </isSptDiagnosis>
  <isSptSerialLogCfg>
    <!--optional, xs:boolean, whether it supports configuring serial port log
redirection: "true", "false"-->
  </isSptSerialLogCfg>
  <isSptFileExport>
    <!--optional, xs:boolean, whether it supports exporting files from the
device: "true", "false"-->
  </isSptFileExport>
  <isSptCertificationStandard>
    <!--optional, xs:boolean, whether it supports configuring authentication
standard for security control panel: "true", "false"-->
  </isSptCertificationStandard>
  <isSptKeypadLock>
```

```
<!--optional, xs:boolean, whether it supports locking keypad: "true",
>false"-->
</isSptKeypadLock>
<MixedTargetDetection><!--optional, whether the device supports recognizing
specific target among mixed targets-->
  <isSupportFaceRecognition><!--optional, xs:boolean, whether it supports
face recognition--></isSupportFaceRecognition>
  <isSupportHumanRecognition><!--optional, xs:boolean, whether it supports
human body recognition--></isSupportHumanRecognition>
  <isSupportVehicleRecognition><!--optional, xs:boolean, whether it supports
vehicle recognition--></isSupportVehicleRecognition>
</MixedTargetDetection>
<isSupportDiscoveryMode><!--optional, xs:boolean--></isSupportDiscoveryMode>
<streamEncryptionType>
  <!--dep, xs:string, stream encryption type: "RTP/TLS", "SRTP/UDP", "SRTP/
MULTICAST". This node is valid when <b>isSupportEncryption</b> is "true", and the
device can support one or more stream encryption types-->
  </streamEncryptionType>
  <isSupportLms><!--optional, xs:boolean, whether it supports laser--></
isSupportLms>
  <isSupportLCDScreen><!--optional, xs:boolean, whether it supports LCD screen--
></isSupportLCDScreen>
  <isSupportBluetooth><!--optional, xs:boolean, whether it supports bluetooth--
></isSupportBluetooth>
  <isSupportAcsUpdate>
    <!--optional, whether it supports upgrading sub access control devices or
peripheral modules: "true"-yes, this node is not returned-no-->
    </isSupportAcsUpdate>
  <isSupportAccessControlCap>
    <!--optional, whether it supports access control capability: "true"-yes,
this node is not returned-no-->
    </isSupportAccessControlCap>
  <isSupportIDCardInfoEvent><!--optional, whether it supports ID card swiping
event: "true"-yes. This node will not be returned if this function is not
supported--></isSupportIDCardInfoEvent>
  <OpenPlatformCap><!--optional, embedded open platform capability, refer to
the message XML_OpenPlatformCap for details-->
  <isSupportInstallationAngleCalibration>
    <!--optional, xs:boolean, whether it supports installation angle
calibration-->
    </isSupportInstallationAngleCalibration>
  <isSupportZeroBiasCalibration>
    <!--optional, xs:boolean, whether it supports zero bias calibration-->
    </isSupportZeroBiasCalibration>
  <isSupportDevStatus><!--optional, xs:boolean, whether device supports getting
device status--></isSupportDevStatus>
  <isSupportRadar><!--optional, xs:boolean, whether it supports the security
radar--></isSupportRadar>
  <isSupportRadarChannels><!--optional, xs:boolean, whether it supports getting
radar channels--></isSupportRadarChannels>
  <radarIPDForm><!--optional, xs:string, radar form: "single"-single radar,
"double_diagonal"-two radars forming an 180° diagonal, "double_vertical"-two
```

```
radars forming a 90° vertical angle--></radarIPDForm>
  <isSupportRadarFieldDetection><!--optional, xs:boolean, whether it supports
intrusion detection (radar)--></isSupportRadarFieldDetection>
  <isSupportRadarLineDetection><!--optional, xs:boolean, whether it supports
line crossing detection (radar)--></isSupportRadarLineDetection>
  <mixedTargetDetectionWebNoDisplay><!--optional, xs:boolean, whether to enable
not displaying multi-target-type recognition--></
mixedTargetDetectionWebNoDisplay>
  <SHMCap><!--opt-->
    <isSupportHighHDDTemperature><!--optional, xs:boolean, whether it supports
HDD high temperature detection--></isSupportHighHDDTemperature>
    <isSupportLowHDDTemperature><!--optional, xs:boolean, whether it supports
HDD low temperature detection--></isSupportLowHDDTemperature>
    <isSupportHDImpact><!--optional, xs:boolean, whether it supports HDD impact
detection--></isSupportHDImpact>
    <isSupportHDBadBlock><!--optional, xs:boolean, whether it supports HDD bad
sector detection--></isSupportHDBadBlock>
    <isSupportSevereHDFailure><!--optional, xs:boolean, whether it supports HDD
severe fault detection--></isSupportSevereHDFailure>
  </SHMCap>
  <isSupportBVCorrect><!--optional, xs:boolean, whether it supports configuring
camera correction parameters--></isSupportBVCorrect>
  <guideEventSupport opt="linkageCapture">
    <!--optional,xs:string, events which support quick setup by instruction,
"linkageCapture"-capture by linkage-->
  </guideEventSupport>
  <isSupportAutoSwitch><!--optional, xs:boolean, whether it supports auto
switch--> true</isSupportAutoSwitch>
  <isSupportDataPrealarm><!--optional,xs:boolean, whether it supports traffic
pre-alarm event--></isSupportDataPrealarm>
  <supportGISEvent opt="AID,TPS,ANPR,mixedTargetDetection">
    <!--optional, xs:string, event types that support GIS information access:
AID (corresponding SDK event: COMM_ALARM_AID_V41), TPS (corresponding SDK
event: COMM_ALARM_TPS_REAL_TIME), ANPR (corresponding SDK event:
COMM_ITS_PLATE_RESULT), mixedTargetDetection-mixed targets detection-->
  </supportGISEvent>
  <isSupportIntelligentMode><!--optional, xs:boolean, whether it supports
intelligent scene switch (related URI:/ISAPI/System/IntelligentSceneSwitch?
format=json)--></isSupportIntelligentMode>
  <isSupportCertificateCaptureEvent><!--optional, xs:boolean, whether it
supports certificate capture and comparison events: true=yes. If this function
is not supported, this node will not be returned--></
isSupportCertificateCaptureEvent>
  <isSupportAlgorithmsInfo><!--optional, xs:boolean, whether it supports
getting the algorithm library version information: true=yes. If this function
is not supported, this node will not be returned--></isSupportAlgorithmsInfo>
  <isSupportVibrationDetection><!--optional, xs:boolean, whether it supports
vibration detection--></isSupportVibrationDetection>
  <isSupportFaceTemperatureMeasurementEvent><!--optional, xs:boolean, whether
it supports uploading face thermography events (eventType:
"FaceTemperatureMeasurementEvent")--></isSupportFaceTemperatureMeasurementEvent>
  <isSupportQRCodeEvent><!--optional, xs:boolean, whether it supports uploading
```

```
QR code events (eventType: "QRCodeEvent")--></isSupportQRCodeEvent>
  <isSupportPersonArmingTrack><!--optional, xs:boolean, whether device supports
person arming (related URI: /ISAPI/Intelligent/channels/<ID>/personArmingTrack/
capabilities?format=json)--></isSupportPersonArmingTrack>
  <isSupportManualPersonArmingTrack><!--optional, xs:boolean, whether device
supports manual person arming (related URI: /ISAPI/Intelligent/channels/<ID>/
manualPersonArmingTrack?format=json)--></isSupportManualPersonArmingTrack>
  <isSupportGPSCalibrationMode><!--optional, xs:boolean, whether device
supports GPS calibration (related URI: /ISAPI/System/GPSCalibration/channels/
<ID>/mode?format=json)--></isSupportGPSCalibrationMode>
  <isSupportGPSVerification><!--optional, xs:boolean, whether device supports
GPS verification (related URI: /ISAPI/System/GPSVerification/channels/<ID>/
points?format=json)--></isSupportGPSVerification>
  <isSupportHBDLib><!--optional, xs:boolean, whether device supports human body
picture library (related URI: /ISAPI/Intelligent/HBDLib/capabilities?
format=json)--></isSupportHBDLib>
  <isSupportFireEscapeDetection><!--optional, xs:boolean, whether the device
supports fire engine access detection (related URI: /ISAPI/Intelligent/channels/
<ID>/fireEscapeDetection/capabilities?format=json)--></
isSupportFireEscapeDetection>
  <isSupportTakingElevatorDetection><!--optional, xs:boolean, whether the
device supports elevator detection (related URI: /ISAPI/Intelligent/channels/
<ID>/takingElevatorDetection/capabilities?format=json)--></
isSupportTakingElevatorDetection>
  <isSupportSSDFileSystemUpgrade><!--optional, xs:boolean, whether the device
supports SSD file system upgrade (related URI: /ISAPI/System/SSDFileSystem/
upgrade?format=json)--></isSupportSSDFileSystemUpgrade>
  <isSupportSSDFileSystemFormat><!--optional, xs:boolean, whether the device
supports SSD file system formatting (related URI: /ISAPI/System/SSDFileSystem/
format?format=json)--></isSupportSSDFileSystemFormat>
  <isSupportSSDFileSystemCapacity><!--optional, xs:boolean, whether the device
supports getting space distribution information of SSD file system (related
URI: /ISAPI/System/SSDFileSystem/capacity?format=json)--></
isSupportSSDFileSystemCapacity>
  <isSupportAIOpenPlatform><!--optional, xs:boolean, whether the device
supports AI open platform capabilities; if supports, this node will be returned
and its value is true; if not, this node will not be returned--></
isSupportAIOpenPlatform>
  <isSupportPictureDownloadError><!--optional, xs:boolean, whether the device
supports reporting picture download failure--></isSupportPictureDownloadError>
  <characteristicCode min="1" max="128"><!--optional, xs:string, device
attribute code (related URI: /ISAPI/System/deviceInfo/characteristicCode?
format=json)--></characteristicCode>
  <isSupportContainerDetection><!--optional, xs:boolean, whether the device
supports container detection (if this node is not returned, refer to the value
returned by /ISAPI/Traffic/ContentMgmt/InputProxy/channels/<ID>/ocrScene/
capabilities to find whether the device supports container detection)--></
isSupportContainerDetection>
  <isSupportLensParamFile><!--optional, xs:boolean, whether the device supports
exporting and importing the lens parameters file--></isSupportLensParamFile>
  <isSupportCounting><!--optional, xs:boolean, ro, whether it supports people
counting--></isSupportCounting>
```

```
<isSupportFramesPeopleCounting><!--optional, xs:boolean, ro, whether it
supports regional people counting--></isSupportFramesPeopleCounting>
  <zoomFocusWebDisplay
opt="ROI,roadTrafficDetection,SMD,mixedTargetDetection,faceCapture"><!--
optional, string, zoom and focus page supported by the Web Client--></
zoomFocusWebDisplay>
  <isSupportDebugLogModuleType
opt="playService,communicationService,attendanceService,faceService"><!--
optional, xs:boolean, whether to export the debugging logs by module type; the
value of <moduleType> in the URI (/ISAPI/System/debugLog?
format=json&moduleType=<moduleType>) can be: "playService",
"communicationService", "attendanceService", "faceService"--></
isSupportDebugLogModuleType>
  </isSupportPlateQuaAlarm>
  <isSupportWiegand><!--optional, xs:boolean, ro, whether it supports the
Wiegand protocol (related URI: /ISAPI/System/Wiegand/<wiegandID>/capabilities?
format=json)-->true</isSupportWiegand>
  <isSupportChannelOccupy><!--optional, xs:boolean, whether it supports
detection of outdoor fire escape occupied by vehicle--></isSupportChannelOccupy>
  <isSupportOffDuty><!--optional, xs:boolean, whether it supports detection of
person absent in fire control room--></isSupportOffDuty>
  <isSupportNoCertificate><!--optional, xs:boolean, whether it supports
detection of authenticated staff not enough in fire control room--></
isSupportNoCertificate>
  <isSupportSmokeAlarm><!--optional, xs:boolean, whether it supports smoke
alarm--></isSupportSmokeAlarm>
  <isSupportBatteryCarDisobey><!--optional, xs:boolean, whether it supports
electric scooter parking violation detection--></isSupportBatteryCarDisobey>
  <isSupportNoFireExtinguisherRecog><!--optional, xs:boolean, whether it
supports fire extinguisher missing detection--></
isSupportNoFireExtinguisherRecog>
  <isSupportIndoorPasswayBlock><!--optional, xs:boolean, whether it supports
indoor channel blockage detection--></isSupportIndoorPasswayBlock>
  <isSupportFireSmartFireDetect><!--optional, xs:boolean, whether it supports
fire source detection--></isSupportFireSmartFireDetect>
  <isSupportDetectorRunningStatus><!--optional, xs:boolean, whether it supports
detector running status--></isSupportDetectorRunningStatus>
  <isSupportDetectorOperationStatus><!--optional, xs:boolean, whether it
supports detector operation status--></isSupportDetectorOperationStatus>
  <isSupportDetectorTemperatureAlarm
opt="highTemperature,raiseTemperature,flame"><!--optional, xs:boolean, whether
it supports temperature alarm: "highTemperature" (high temperature alarm),
"raiseTemperature" (temperature rising alarm), "flame" (flame alarm)--></
isSupportDetectorTemperatureAlarm>
  <isSupportDetectorShelterAlarm><!--optional, xs:boolean, whether it supports
detector video tampering alarm--></isSupportDetectorShelterAlarm>
  <isSupportDetectorMotionAlarm><!--optional, xs:boolean, whether it supports
detector movement alarm--></isSupportDetectorMotionAlarm>
  <isSupportDetectorTamperAlarm><!--optional, xs:boolean, whether it supports
detector tampering alarm--></isSupportDetectorTamperAlarm>
  <isSupportDetectorEmergencyAlarm><!--optional, xs:boolean, whether it
supports detector emergency alarm--></isSupportDetectorEmergencyAlarm>
```

```
<isSupportSmokingDetectAlarm><!--optional, xs:boolean, whether it supports
smoking alarm--></isSupportSmokingDetectAlarm>
<isSupportDetectorSmokeAlarm><!--optional, xs:boolean, whether it supports
smoke alarm--></isSupportDetectorSmokeAlarm>
<isSupportDetectorCombustibleGasAlarm><!--optional, xs:boolean, whether it
supports gas alarm--></isSupportDetectorCombustibleGasAlarm>
<isSupportFireControlData><!--optional, xs:boolean, whether it supports
uploading real-time fire protection data--></isSupportFireControlData>
<isSupportFireNoRegulation><!--optional, xs:boolean, whether it supports fire
no regulation alarm--></isSupportFireNoRegulation>
<isSupportSmokeFireRecognize><!--optional, xs:boolean, whether it supports
uploading the smoke and fire detection event--></isSupportSmokeFireRecognize>
</DeviceCap>
```

A.3.209 XML_DetectorCfg

DetectorCfg message in XML format

```
<DetectorCfg version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <ZoneCondList/>
  <!--opt, up to 64 conditions are allowed. When getting, this node will be
the input parameter, and no need for the device to return this part-->
  <DetectorParamList><!--up to 64-->
    <DetectorParam>
      <id><!--req, xs:string, zone No.--></id>
      <enabled>
        <!--opt, xs: boolean, enable the zone linked detector or not-->
      </enabled>
      <detectorSerialNo>
        <!--opt, xs:string, zone linked detector serial No.-->
      </detectorSerialNo>
      <detectorWorkMode>
        <!--req, xs:string, detector working mode, including normal mode,
sleeping mode, pacing mode. -->
      </detectorWorkMode>
      <detectorHeartBeatCycle>
        <!--req, xs:string, detector heartbeat period (5 minutes, 10 minutes,
20 minutes, 30 minutes, 60 minutes, no heartbeat time) -->
      </detectorHeartBeatCycle>
    </DetectorParam>
  </ DetectorParamList>
</DetectorCfg>
```

A.3.210 XML_EmergencyAlarmProductCap

Message about the capability of the one-touch panic alarm product in XML format.

```
<EmergencyAlarmProductCap version="2.0" xmlns="http://www.isapi.org/ver20/
XMLSchema">
```



```

    <callWaitingCfg><!--required, xs:boolean, whether it supports calling
waiting settings: true=yes. If this function is not supported, this node will
not be returned--></callWaitingCfg>
    <alarmLampCfg><!--required, xs:boolean, whether it supports calling alarm
lamp settings: true=yes. If this function is not supported, this node will not
be returned--></alarmLampCfg>
    <voicePromptionCfg><!--required, xs:boolean, whether it supports voice prompt
settings: true=yes. If this function is not supported, this node will not be
returned--></voicePromptionCfg>
    <emergencyAlarmResponse><!--required, xs:boolean, whether it supports
handling panic alarms: true=yes. If this function is not supported, this node
will not be returned--></voicePromptionCtrl>
    <audioInOutCfg><!--optional, xs:boolean, whether it supports audio input and
output settings: true=yes. If this function is not supported, this node will
not be returned--></audioInOutCfg>
    <nearInfraredInductionCfg><!--optional, xs:boolean, whether it supports close-
distance IR sensor settings: true=yes. If this function is not supported, this
node will not be returned--></nearInfraredInductionCfg>
    <getAudioFileListByType><!--optional, xs:boolean, whether it supports getting
the audio file list: true=yes. If this function is not supported, this node
will not be returned--></getAudioFileListByType>
    <deleteAudioFile><!--optional, xs:boolean, whether it supports deleting the
audio file: true=yes. If this function is not supported, this node will not be
returned--></deleteAudioFile>
</EmergencyAlarmProductCap>

```

A.3.211 XML_EventNotificationAlert_AlarmEventInfo

EventNotificationAlert message with alarm/event information in XML format.

```

<EventNotificationAlert version="2.0" xmlns="http://www.isapi.org/ver20/
XMLSchema">
    <ipAddress><!--dep, xs:string, device IPv4 address--></ipAddress>
    <ipv6Address><!--dep, xs:string, device IPv6 address--></ipv6Address>
    <portNo><!--opt, xs:integer, device port number--></portNo>
    <protocol><!--opt, xs:string, protocol type for uploading alarm/event
information, "HTTP,HTTPS"--></protocol>
    <macAddress><!--opt, xs:string, MAC address--></macAddress>
    <channelID><!--dep, xs:string, device channel No., starts from 1--></
channelID>
    <dateTime><!--req, alarm/event triggered or occurred time, format:
2017-07-19T10:06:41+08:00--></dateTime>
    <activePostCount><!--req, xs:integer, alarm/event frequency, starts from 1--
></activePostCount>
    <eventType><!--req, xs:string, alarm/event type, "peopleCounting, ANPR,..."--
></eventType>
    <eventState>
        <!--req, xs:string, durative alarm/event status: "active"-valid, "inactive"-
invalid, e.g., when a moving target is detected,
        the alarm/event information will be uploaded continuously unit the status
is set to "inactive"-->

```

```
</eventState>
<eventDescription><!--req, xs:string, alarm/event description--></
eventDescription>
<...><!--opt, for different alarm/event types, the nodes are different, see
the message examples in different applications--></...>
</EventNotificationAlert>
```

A.3.212 XML_IpViewDevAbility

XML message about the video intercom device capability

```
<IpViewDevAbility version="2.0">
  <!--video intercom device capability-->
  <SipServerLogin><!--required, registration capability of SIP server-->
    <AutoLogin><!--required, whether to support automatic registration-->
      <enable></enable>
    </AutoLogin>
    <loginStatus opt="registered,unregistered"/><!--required, registration
status-->
    <sipLoginNameLen min="1" max="32"/><!--required, length of registered user
name-->
    <sipLoginPasswordLen min="1" max="16"/><!--required, length of registered
password-->
    <displayNameLen min="1" max="128"/><!--required, length of displayed device
name-->
    <localNumber min="1" max="32"/><!--required, length of local station
number-->
    <loginCycle min="1" max="99"/><!--required, registration period, unit:
minute-->
    <serverSipPort min="1024" max="65535"/>
    <localPort min="1024" max="65535"><!--optional, local port No.-->
    <isNotSupportLocalPort></isNotSupportLocalPort><!--optional, whether the
local port is not supported: true=yes (the local port is not supported), this
field is not supported-no (the local port is supported)-->
    <domainNameLen min="", max=""><!--optional, the meaning of this field is
the same as that of the field <sipServerDomain>. For new devices, both fields
need to be returned-->
    <addressType opt="IP/IPV6, domain"/><!--supported address type-->
    <mutexAbility opt="gbt28181"/><!--required, mutex ability, gbt28181-->
    <notSupportCharacter opt=""><!--whether the domain name contains
unsupported characters: true=yes (the domain name contains unsupported
characters)-->
    <isNotSupportSipServerIP></isNotSupportSipServerIP><!--whether the IP
address of the SIP server is not supported: true=yes (the IP address of the SIP
server is not supported), this field is not supported-no (the IP address of the
SIP server is supported). For old devices, this field will not be returned;
this field is used to check whether new devices support the IP address of the
SIP server-->
    <sipServerDomain min="" max=""><!--optional, domain name of the SIP
server. The meaning of this field is the same as that of the field
<domainNameLen>. For new devices, both fields need to be returned-->
```

```
<stunServerIP min="" max=""/><!--optional, IP address of the STUN server-->
<stunServerDomain min="" max=""/><!--optional, domain name of the STUN
server-->
<stunServerPort min="" max=""/><!--optional, port No. of the STUN server-->
<proxyServerIP min="" max=""/><!--optional, IP address of the proxy server--
>
<proxyServerDomain min="" max=""/><!--optional, domain name of the proxy
server-->
<proxyServerPort min="" max=""/><!--optional, port No. of the proxy server--
>
<netWork opt="0,1,2,3"/><!--optional, network type: 0-invalid, 1-wired
network 1, 2-wired network 2, 3-wireless network-->
<CalledTargetName min="" max=""/><!--optional, user name length of the
called person-->
</SipServerLogin>
<LocalAbility><!--required, basic capability of video intercom extension-->
<defaultRing min="1" max="6"/><!--required, options of the default local
ringtone-->
<ringVolume min="1" max="9"/><!--required, range of the local ringtone
volume-->
<inputVolume min="0" max="6"/><!--required, input volume options-->
<outputVolume min="0" max="9"/><!--required, output volume options-->
<audioEncPriNum><!--required, supported number of audio encoding levels--></
audioEncPriNum>
<delayPreview min="0" max="30"/><!--required, range of live view delay,
unit: second-->
<AudioEncEntry><!--required, supported audio encoding type, multiple types
can be set for the same node-->
<index></index>
<name></name>
<packetLen opt="160,320"/>
</AudioEncEntry>
<CallAbility><!--required, calling capability-->
<AutoResponse><!--required, support automatic response-->
<autoResponse min="0" max="30"/><!--required, range of automatic
response duration, unit: second-->
</AutoResponse>
<callNumber><!--required, supported number of calling numbers--></
callNumber>
<callNumberLen min="0" max="32"/>
</CallAbility>
</LocalAbIility>
<VideoIntercom>
<!--video intercom, indoor station, door station, main station, door
station (V series), doorphone-->
<monitorChannelNo min="1" max="100"/><!--required, camera channel of main
station or indoor station-->
<DeviceID><!--device No.-->
<enabled>true</enabled><!--supported device No. configuration-->
<supportUnitType opt=""/><!--required, supported video intercom type-->
<InDoorDevice><!--required, indoor station-->
<floorNum min="1" max="16"/><!--required, floor No.-->
```

```
<roomNum min="1" max="16"/><!--required, room No.-->
<devIndex min="0" max="10"/><!--required, indoor station No.-->
</InDoorDevice>
<OutDoorDevice><!--required, door station/intelligent access control
device-->
    <period min="1" max="16"/><!--required, community No.-->
    <buildingNum min="1" max="16"/><!--required, building No.-->
    <unitNum min="0" max="10"/><!--required, unit No.-->
    <floorNum min="1" max="16"/><!--required, floor No.-->
    <devIndex min="0" max="10"/><!--required, door station No.-->
</OutDoorDevice>
<ManageUnitDevice><!--required, main station-->
    <period min="1" max="16"/><!--required, community No.-->
    <devIndex min="0" max="10"/><!--required, main station No.-->
</ManageUnitDevice>
<OutDoorFenceDevice><!--required, outer door station-->
    <period min="1" max="16"/><!--required, community No.-->
    <devIndex min="0" max="10"/><!--required, outer door station No.-->
</OutDoorFenceDevice>
<VillaOutDoorDevice><!--required, door station (V series)-->
    <period min="1" max="16"/><!--required, community No.-->
    <buildingNum min="1" max="16"/><!--required, building No.-->
    <unitNum min="0" max="10"/><!--required, unit No.-->
    <floorNum min="1" max="16"/><!--required, floor No.-->
    <devIndex min="0" max="10"/><!--required, door station No.-->
</VillaOutDoorDevice>
<AgainDevice><!--required, doorphone--></AgainDevice>
</DeviceID>
<PrivilegePwd><!--permission password-->
    <pwdType opt="engineering,setupAlarm,householderUnlock,antiHijacking"/
><!--password type: "engineering"-project password, "setupAlarm"-arming and
disarming password, "householderUnlock"-resident unlocking password,
"antiHijacking"-duress password-->
    <pwdLen min="6" max="16"/><!--password length-->
</PrivilegePwd>
<OperationTime><!--operation time-->
    <monitoringTime min="10" max="60"/><!--maximum live view duration, unit:
second-->
    <ringTime min="15" max="60"/><!--maximum ringing duration, unit: second-->
    <messageTime min="30" max="60"/><!--maximum messaging duration, unit:
second-->
    <talkTime min="90" max="120"/><!--maximum calling duration, unit: second--
>
    <callForwardingTime min="0" max="20"/><!--calling forwarding timeout,
unit: second-->
    <dwRingDurationTime min="30" max="60"/><!--ringing time duration, unit:
second-->
</OperationTime>
<RelateDevice>
    <outdoorUnitIP opt="ipv4,ipv6"/><!--IP address of main door station-->
    <manageUnitIP opt="ipv4,ipv6"/><!--IP address of main station-->
    <sipServerIP opt="ipv4,ipv6"/><!--IP address of SIP server-->
```

```
<centerIP opt="ipv4,ipv6"/><!--center IP address-->
<centerPort min="" max=""/><!--center port No.-->
<indoorUnitIP opt="ipv4,ipv6"/><!--IP address of indoor station-->
<notSupportAgainUnitIP opt="true,false"/><!--required, whether
configuring doorphone IP address is not supported: "true"-yes, "false"-no-->
<againUnitIP opt="ipv4,ipv6"/><!--doorphone IP address-->
<notSupportOutDoorType opt="true,false"/><!--required, whether
configuring main door station type is not supported: "true"-yes, "false"-no-->
<outDoorType opt="unitOutdoor,villaOutDoor"/><!--main door station type:
"unitOutdoor"-main door station (D series), "villaOutDoor"-main door station (V
series)-->
<outInConnectMode opt="sameLan,diffLan"/><!--networ mode of door station
and sub indoor station: "sameLan"-in the same LAN, "diffLan"-in different LANs--
>
<indoorConnectMode opt="wireless,wired"/><!--network mode of main indoor
station and sub indoor station: "wireless"-wireless NIC, "wired"-wired NIC-->
</RelateDevice>
<NoticeData>
<noticeTime><!--notice time--></noticeTime>
<noticeNumberLen min="0" max="32"/><!--notice No.-->
<noticeThemeLen min="0" max="64"/><!--notice theme-->
<noticeDetailLen min="0" max="1024"/><!--notice details-->
<noticeLevel opt="advertisement,propertyMgmt,alarm,notification"/><!--
notice level: "advertisement"-advertisement information, "propertyMgmt"-
property information, "alarm"-alarm information, "notification"-notice
information-->
<maxPicNum><!--number of pictures--></maxPicNum>
<maxSinglePicSize><!--maximum size of a picture, unit: KB--></
maxSinglePicSize>
</NoticeData>
<ControlGateway><!--unlock remotely-->
<gatewayIndex min="1"/><!--access control No.-->
<command opt="close,open"/><!--control command-->
<controlSrc>true</controlSrc><!--operation source information-->
<controlType opt="monitor,calling"/><!--unlocking scene type: "monitor",
"calling"-->
<lockType opt="normal,smartLock"/><!--lock type-->
<lockID min="" max=""/><!--lock ID-->
<password min="" max=""/><!--password length of the smart lock-->
</ControlGateway>
<Zone><!--zone configuration-->
<ZoneConfig>
<enabled><!--whether to support configuring zone parameters (alarm
input parameters)--></enabled>
<delayInParam><!--method of setting delay: "true"-the client sets the
delay time by dwParam in the structure NET_DVR_ALARMIN_PARAM (this method is
used by power and environment monitoring system and ATM security control
panel), "false"-the client sets the delay time by wEnterDelay and wExitDelay in
the structure NET_DVR_ALARMSUBSYSTEMPARAM--></delayInParam>
<detectorType
opt="panicButton,magneticContact,smokeDetector,activeInfraredDetector,passiveInf
raredDetector,glassBreakDetector,vibrationDetector,dualTechnologyPirDetector,tri
```

```

pleTechnologyPirDetector, humidityDetector, temperatureDetector, combustibleGasDetector, dynamicSwitch, controlSwitch, smartLock, waterDetector, otherDetector"/><!--
detector type supported by the device: "panicButton"-panic switch,
"magneticContact"-magnetic contact, "smokeDetector"-smoke detector,
"activeInfraredDetector"-active infrared detector, "passiveInfraredDetector"-
passive infrared detector, "glassBreakDetector"-glass break detector,
"vibrationDetector"-vibration detector, "dualTechnologyPirDetector"-dual
technology motion detector, "tripleTechnologyPirDetector"-triple technology
detector, "humidityDetector"-humidity detector, "temperatureDetector"-
temperature detector, "combustibleGasDetector"-gas detector, "dynamicSwitch"-
follow-up switch, "controlSwitch"-control switch, "smartLock"-smart lock,
"waterDetector"-water detector, "otherDetector"-other detector type-->
    <zoneType opt="instantZone,
24hourAudibleZone, delayZone, interiorWithDelayZone, keyswitchZone, supervisedFireZone,
perimeterZone, 24hourSilentZone, disable"/><!--zone type supported by the
device-->
    <LimitedDetectorType>
        <Detector>
            <name>smokeDetector</name>
            <zoneType opt="24hourAudibleZone"/>
        </Detector>
        <Detector>
            <name>glassBreakDetector</name>
            <zoneType opt="24hourAudibleZone"/>
        </Detector>
    </LimitedDetectorType>
    <uploadAlarmRecoveryReport>true</uploadAlarmRecoveryReport><!--whether
to support report configuration of uploading alarm recovery-->
    <zoneDelayTime min="" max=""/><!--delayed zone delay-->
    <sensitivity opt="10ms, 250ms, 500ms, 750ms"/><!--sensitivity-->
    <arrayBypass>true</arrayBypass><!--whether to support zone group bypass
configuration-->
    <moduleStatus attri="readonly" opt="offline, online"/><!--module status-->
    <moduleAddress min="" max=""/><!--module address-->
    <moduleChannel>true</moduleChannel><!--module channel-->
    <moduleType opt="localZone, 1zoneExpander, 2zoneExpander, 8ZoneExpander,
8sensorZoneExpander, 1Zone&Trigger"/><!--supported zone type-->
    <zoneNo attri="readonly" min="" max=""/><!--zone No. which can only be
obtained-->
    <subsystemNo attri="readonly"><!--No. of the partition that the zone
belongs to, it can only be obtained--></subsystemNo>
    <alarmType opt="open, close" default="open"/><!--required, alarm device
type: "open"-remain open, "close"-remain closed-->
    <InDelayTime min="0" max="255"/><!--required, entrance delay, unit:
second-->
    <OutDelayTime min="0" max="255"/><!--required, exiting delay, unit:
second-->
</ZoneConfig>
<GetZoneList>
    <enabled><!--whether to support getting zone list--></enabled>
</GetZoneList>

```

```
<ZoneArmDisarm>
  <enabled><!--whether to support arming and disarming the zone--></
enabled>
</ZoneArmDisarm>
<ZoneGroupBypass>
  <enabled><!--whethe to support zone group bypass--></enabled>
</ZoneGroupBypass>
</Zone>
<IOIn>
  <IOInNo attri="readonly" min="" max=""/><!--IP input No. which can only
be obtained-->
  <useType opt="disabled,openDoorBtn,doorStatus,custom"/><!--purpose:
"disabled", "openDoorBtn"-door exit button, "doorStatus"-door status, "custom"--
>
</IOIN>
<IOOut>
  <IOOutNo attri="readonly" min="" max=""/><!--IO output No. which can only
be obtained-->
  <useType opt="disabled,electricLock,custom"/><!--purpose: "disabled",
"electricLock"-electric lock, "custom"-->
</IOOut>
<ElevatorControl>
  <elevatorNo attri="readonly" min="" max=""/> <!--ro, elevator No.-->
  <interfaceType opt="RS485,network"/><!--interface type: "RS485"-RS-485,
"network"-->
  <RS485Protocol opt="private,custom"/><!--RS-485 protocol type: "private",
"custom"-->
  <networkProtocol opt="private,custom"/><!--NIC protocol type: "private",
"custom"-->
  <serverIP opt="ipv4,ipv6"/><!--optional, IP address of the elevator
control server-->
  <serverPort min="" max=""/><!--optional, port No. of the elevator control
server-->
</ElevatorControl>
<RS485Config>
  <!--RS-485 configuration, this node will not be returned if RS-485
configuration is not supported-->
  <deviceNameLength min="0" max="32"/><!--RS-485 name-->
  <deviceType>true</DeviceType><!--whether to support configuring device
type-->
  <deviceProtocol>true</deviceProtocol><!--whether to support configuring
device protocol-->
</RS485Config>
  <supportDevInfo opt="true,false"/><!--required, whether to support getting
complete indoor station No. This node will not be returned if this function is
not supported, and this node is valid only when the device is an indoor
station-->
  <supportRegisterInfo opt="true,false"/><!--required, whether the door
station supports getting registration information. This node will not be
returned if this function is not supported, and this node is valid only when
the device is an door station-->
  <CallRoomConfig><!--configuration of calling resident by pressing button
```

```

for door station (V series)-->
    <keyNo min="" max=""/><!--villa button No.-->
    <floorNo min="" max=""/><!--villa floor No.-->
    <roomNo min="" max=""/><!--villa room No.-->
    <callManageCenter opt="true,false"/><!--required, whether to set it to
the calling management center-->
    <calledName min="" max=""/><!--optional, user name to be called, it
supports letters, digits, @, and dot. This node is valid in standard SIP mode-->
</CallRoomConfig>
    <VideoCall><!--optional, video intercom capability-->
        <enabled opt="true,false"/><!--required, whether supports starting video
intercom-->
        <supportCmd
opt="callRequest,Cancel,Answer,Decline,Timeout,Bye,deviceCalling,clientCalling"/
>
            <!--required, command type supported by the device: "callRequest"-request
for call, "Cancel"-cancel the call, "Answer"-answer, "Decline"-decline,
"Timeout"-time out, "Bye"-end the call, "deviceCalling"-device is in call,
"clientCalling"-client is in call-->
        </VideoCall>
        <CallerDevice><!--required, calling device information-->
            <period min="1" max="16"/><!--required, community No.-->
            <buildingNum min="1" max="16"/><!--required, building No.-->
            <unitNum min="0" max="10"/><!--required, unit No.-->
            <floorNum min="1" max="16"/><!--required, floor No.-->
            <devIndex min="0" max="10"/><!--required, device No.-->
            <devType min="1" max="9"/><!--required, device type-->
        </CallerDevice>
        <CallStatus><!--required, calling status-->
            <callingStatus min="1" max="3"/><!--required, calling status-->
        </CallStatus>
        <EzvizDeviceInfo><!--required, EZVIZ device information-->
            <deviceNum min="1" max="32"/><!--required, number of devices-->
            <DeviceCfg size="16"/>
            <deviceNameLen min="1" max="32"/><!--required, device name-->
            <deviceType min="1" max="3"/><!--required, device type-->
            <deviceID min="0" max="10"/><!--required, device No.-->
        </EzvizDeviceInfo>
        <VideoCallParam><!--required, signal interaction command (non-persistent
connection)-->
            <cmdType min="0" max="4"/><!--required, command-->
        </VideoCallParam>
        <VideoIntercomStream><!--optional, video source-->
            <sourceType opt="IPC,DVR/DVS/
NVR,OutDoorDevice,OutDoorFenceDevice,AgainDevice"/><!--optional, video source
type-->
            <againDeviceNumber min="" max=""/><!--optional, number of doorphones-->
        </VideoIntercomStream>
        <indoorDevChangeEnabled opt="true,false"/><!--required, whether to support
switching between main indoor station and sub indoor station-->
        <indoorDevChangeReboot opt="true,false"/><!--required, whether the device
will reboot after switching devices of main indoor station and sub indoor

```



```

station-->
  </VideoIntercom>
  <UploadAlarmCfg><!--configuration of video intercom alarm, it corresponds to
the structure NET_DVR_VIDEO_INTERCOM_ALARM_CFG-->
    <UploadDoorNotCloseAlarm opt="true,false"/><!--required, whether to upload
alarms of unlocking the door-->
  </UploadAlarmCfg>
  <ZoneList size="64"><!--special zone configuration, which is the capability
of distinguishing different zones according to the zone ID-->
    <ZoneConfig>
      <zoneID min="" max=""><!--zone ID--></zoneID>
      <enabled><!--whether to support configuring zone parameters (alarm input
parameters)--></enabled>
      <delayInParam><!--method of setting delay: "true"-the client sets the
delay time by dwParam in the structure NET_DVR_ALARMIN_PARAM (this method is
used by power and environment monitoring system and ATM security control
panel), "false"-the client sets the delay time by wEnterDelay and wExitDelay in
the structure NET_DVR_ALARMSUBSYSTEMPARAM--></delayInParam>
      <detectorType
opt="panicButton,magneticContact,smokeDetector,activeInfraredDetector,passiveInf
raredDetector,glassBreakDetector,vibrationDetector,dualTechnologyPirDetector,tri
pleTechnologyPirDetector,humidityDetector,temperatureDetector,combustibleGasDete
cto,dynamicSwitch,controlSwitch,smartLock,waterDetector,otherDetector"/><!--
detector type supported by the device: "panicButton"-panic switch,
"magneticContact"-magnetic contact, "smokeDetector"-smoke detector,
"activeInfraredDetector"-active infrared detector, "passiveInfraredDetector"-
passive infrared detector, "glassBreakDetector"-glass break detector,
"vibrationDetector"-vibration detector, "dualTechnologyPirDetector"-dual
technology motion detector, "tripleTechnologyPirDetector"-triple technology
detector, "humidityDetector"-humidity detector, "temperatureDetector"-
temperature detector, "combustibleGasDetecto"-gas detector, "dynamicSwitch"-
follow-up switch, "controlSwitch"-control switch, "smartLock"-smart lock,
"waterDetector"-water detector, "otherDetector"-other detector type-->
      <zoneType opt="instantZone,
24hourAudibleZone,delayZone,interiorWithDelayZone,keyswitchZone,supervisedFireZo
ne,perimeterZone,24hourSlientZone,disable"/><!--zone type supported by the
device-->
      <uploadAlarmRecoveryReport><!--whether to support report configuration of
uploading alarm recovery--></uploadAlarmRecoveryReport>
      <zoneDelayTime min="" max=""><!--delayed zone delay-->
      <sensitivity opt="10ms,250ms,500ms,750ms"/><!--sensitivity-->
      <arrayBypass><!--whether to support zone group bypass configuration--></
arrayBypass>
      <moduleStatus attri="readonly" opt="offline,online"/><!--module status-->
      <moduleAddress min="" max=""><!--module address-->
      <moduleChannel><!--module channel--></moduleChannel>
      <moduleType opt="localZone, 1zoneExpander,2zoneExpander,8ZoneExpander,
8sensorZoneExpander,1ZoneAndTrigger"/><!--supported zone type-->
      <zoneNo attri="readonly" min="" max=""><!--zone No. which can only be
obtained-->
      <subsystemNo attri="readonly"><!--No. of the partition that the zone
belongs to, it can only be obtained--></subsystemNo>

```

```
<alarmType opt="open,close" default="open"/><!--required, alarm device
type: "open"-remain open, "close"-remain closed-->
<InDelayTime min="0" max="255"/><!--required, entrance delay, unit:
second-->
<OutDelayTime min="0" max="255"/><!--required, exiting delay, unit:
second-->
</ZoneConfig>
<GetZoneList>
  <enabled><!--whether the device supports getting zone list--></enabled>
</GetZoneList>
<ZoneArmDisarm>
  <enabled><!--whether the device supports arming and disarming the zone--
></enabled>
</ZoneArmDisarm>
<ZoneGroupBypass>
  <enabled><!--whether the device supports zone group bypass--></enabled>
</ZoneGroupBypass>
</ZoneList>
<NoNeedReboot>
  <videoResolutionChange opt="true,false"/><!--optional, whether the device
does not reboot after changing video resolution-->
  <videoFrameRateChange opt="true,false"/><!--optional, whether the device
does not reboot after changing video frame rate-->
</NoNeedReboot>
</IpViewDevAbility >
```

A.3.213 XML_LampSchedTimeList

Message about the parameters of the alarm lamp flickering schedule in XML format.

```
<LampSchedTimeList version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <enabled><!--required, xs:boolean, whether to enable the alarm lamp
flickering schedule--></enabled>
  <weekday><!--multiple <weekday> nodes can be returned according to the actual
settings-->
    <index><!--required, xs:string, the day of the week: "1", "2", "3", ...--> </
index>
    <schedTime><!--required, xs:string, time periods in the schedule. For
example, if the start time is 09:22:41 and the end time is 10:01:56, this node
will be "092241-100156". Multiple <schedTime> nodes can be returned according
to the actual settings--></schedTime>
  </weekday>
</LampSchedTimeList>
```

A.3.214 XML_ResponseStatus

XML message about response status

```
<?xml version="1.0" encoding="utf-8"?>
<ResponseStatus version="2.0" xmlns="http://www.std-cgi.org/ver20/XMLSchema">
  <requestURL>
    <!--required, read-only, xs:string, request URL-->
  </requestURL>
  <statusCode>
    <!--required, read-only, xs:integer, status code: 0,1-OK, 2-Device Busy, 3-
Device Error, 4-Invalid Operation, 5-Invalid XML Format, 6-Invalid XML Content,
7-Reboot Required, 9-Additional Error-->
  </statusCode>
  <statusString>
    <!--required, read-only, xs:string, status description: OK, Device Busy,
Device Error, Invalid Operation, Invalid XML Format, Invalid XML Content,
Reboot, Additional Error-->
  </statusString>
  <subStatusCode>
    <!--required, read-only, xs:string, describe the error reason in detail-->
  </subStatusCode>
  <MErrCode>
    <!--optional, xs:string, error code categorized by functional modules,
e.g., 0x12345678-->
  </MErrCode>
  <MErrDevSelfEx>
    <!--optional, xs:string, extension field of MErrCode. It is used to define
the custom error code, which is categorized by functional modules-->
  </MErrDevSelfEx>
</ResponseStatus>
```

A.3.215 XML_ZoneAssociatedDetectorCfg

ZoneAssociatedDetectorCfg message in XML format

```
<ZoneAssociatedDetectorCfg version="2.0" xmlns="http://www.isapi.org/ver20/
XMLSchema">
  <ZoneCondList/>
  <!--opt, up to 64 conditions are supported. When getting the configuration,
this node will be the input parameter, no need to return this part-->
  <DetectorCfgList><!--up to 64-->
    <DetectorCfg>
      <id><!--req, xs:string, zone No.--></id>
      <enabled>
        <!--opt, xs: boolean, enable zone linked detector or not -->
      </enabled>
      <detectorSerialNo>
        <!--req, xs:string, the serial No. of zone linked detector -->
      </detectorSerialNo>
    </DetectorCfg>
  </DetectorCfgList>
</ZoneAssociatedDetectorCfg>
```

A.3.216 XML_ZoneCondList

ZoneCondList message in XML format

```
<ZoneCondList version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <ZoneCond><!--up to 64 conditions are allowed-->
    <id><!--req, xs:string, zone No.--></id>
  </ZoneCond>
</ZoneCondList>
```

A.4 Device Network SDK Errors

The errors that may occur during the device network SDK integration are listed here for reference. You can search for the error descriptions according to the error codes or names returned by a specific API (NET_DVR_GetLastError or NET_DVR_GetErrorMsg).

General Errors

Error Name	Error Code	Error Description
NET_DVR_NOERROR	0	No error.
NET_DVR_PASSWORD_ERROR	1	Incorrect user name or password.
NET_DVR_NOENOUGHPRI	2	No permission.
NET_DVR_NOINIT	3	Uninitialized.
NET_DVR_CHANNEL_ERROR	4	Incorrect channel No.
NET_DVR_OVER_MAXLINK	5	No more device can be connected.
NET_DVR_VERSIONNOMATCH	6	Version mismatches.
NET_DVR_NETWORK_FAIL_CONNECT	7	Connecting to device failed. The device is offline or network connection timed out.
NET_DVR_NETWORK_SEND_ERROR	8	Sending data to device failed.
NET_DVR_NETWORK_RECV_ERROR	9	Receiving data from device failed.
NET_DVR_NETWORK_RECV_TIMEOUT	10	Receiving data from device timed out.
NET_DVR_NETWORK_ERRORDATA	11	The data sent to the device is illegal, or the data received from the device error. E.g. The input data is not supported by the device for remote configuration.

Error Name	Error Code	Error Description
NET_DVR_ORDER_ERROR	12	API calling order error.
NET_DVR_OPERNOPERMIT	13	No permission for this operation.
NET_DVR_COMMANDTIMEOUT	14	Executing device command timed out.
NET_DVR_ERRORSERIALPORT	15	Incorrect serial port No. The specified serial port does not exist.
NET_DVR_ERRORALARMPORT	16	Alarm port No. error. The alarm input or output port of the specified device does not exist.
NET_DVR_PARAMETER_ERROR	17	Incorrect parameter. The input or output parameters of the SDK API is empty, or the parameter value or format is invalid.
NET_DVR_CHAN_EXCEPTION	18	Device channel is in exception status.
NET_DVR_NODISK	19	No HDD in the device.
NET_DVR_ERRORDISKNUM	20	Incorrect HDD No.
NET_DVR_DISK_FULL	21	HDD full.
NET_DVR_DISK_ERROR	22	HDD error.
NET_DVR_NOSUPPORT	23	Device does not support this function.
NET_DVR_BUSY	24	Device is busy.
NET_DVR_MODIFY_FAIL	25	Failed to edit device parameters.
NET_DVR_PASSWORD_FORMAT_ERROR	26	Invalid password format.
NET_DVR_DISK_FORMATING	27	HDD is formatting. Failed to startup.
NET_DVR_DVRNORESOURCE	28	Insufficient device resources.
NET_DVR_DVROPRATEFAILED	29	Device operation failed.
NET_DVR_OPENHOSTSOUND_FAIL	30	Failed to collect local audio data or open audio output during two-way audio and broadcast.
NET_DVR_DVRVOICEOPENED	31	Two-way audio channel is occupied.
NET_DVR_TIMEINPUTERROR	32	Incorrect time input.
NET_DVR_NOSPECFILE	33	No video file for playback.

Error Name	Error Code	Error Description
NET_DVR_CREATEFILE_ERROR	34	Failed to create a file during local recording, saving picture, getting configuration file or downloading video file remotely.
NET_DVR_FILEOPENFAIL	35	Failed to open a file. The file does not exist or directory error.
NET_DVR_OPERNOTFINISH	36	Operation conflicted.
NET_DVR_GETPLAYTIMEFAIL	37	Failed to get the current played time.
NET_DVR_PLAYFAIL	38	Failed to play.
NET_DVR_FILEFORMAT_ERROR	39	Invalid file format.
NET_DVR_DIR_ERROR	40	File directory error.
NET_DVR_ALLOC_RESOURCE_ERROR	41	Allocating resources failed.
NET_DVR_AUDIO_MODE_ERROR	42	Invalid sound card mode error. The opened sound play mode and configured mode mismatched.
NET_DVR_NOENOUGH_BUF	43	Insufficient buffer for receiving data or saving picture.
NET_DVR_CREATESOCKET_ERROR	44	Failed to create SOCKET.
NET_DVR_SETSOCKET_ERROR	45	Failed to set SOCKET.
NET_DVR_MAX_NUM	46	No more registrations and live views can be connected.
NET_DVR_USERNOTEXIST	47	The user does not exist. The user ID is logged out or unavailable.
NET_DVR_WRITEFLASHERROR	48	Writing FLASH error during device upgrade.
NET_DVR_UPGRADEFAIL	49	Failed to upgrade device. Network problem or language mismatches.
NET_DVR_CARDHAVEINIT	50	The decoding card is already initialized.
NET_DVR_PLAYERFAILED	51	Failed to call the function of player SDK.
NET_DVR_MAX_USERNUM	52	No more users can log in to.

Error Name	Error Code	Error Description
NET_DVR_GETLOCALIPANDMACFAIL	53	Failed to get the IP address or physical address of local PC.
NET_DVR_NOENCODEING	54	The decoding function of this channel is not enabled.
NET_DVR_IPMISMATCH	55	IP address mismatches.
NET_DVR_MACMISMATCH	56	MAC address mismatches.
NET_DVR_UPGRADELANGMISMATCH	57	The language of upgrade file mismatches.
NET_DVR_MAX_PLAYERPORT	58	No more channels can be started to play.
NET_DVR_NOSPACEBACKUP	59	Insufficient space to back up file.
NET_DVR_NODEVICEBACKUP	60	No backup device found.
NET_DVR_PICTURE_BITS_ERROR	61	Picture pixel bit mismatches. Only 24 bits are allowed.
NET_DVR_PICTURE_DIMENSION_ERROR	62	Too large picture. The height*width should be less than 128x256.
NET_DVR_PICTURE_SIZ_ERROR	63	Too large picture. The picture size should be smaller than 100K.
NET_DVR_LOADPLAYERSDKFAILED	64	Failed to load the player(PlayCtrl.dll, SuperRender.dll, AudioRender.dll) to the current directory.
NET_DVR_LOADPLAYERSDKPROC_ERROR	65	Failed to find the function in player SDK.
NET_DVR_LOADDSSDKFAILED	66	Failed to load the DS SDK to the current directory.
NET_DVR_LOADDSSDKPROC_ERROR	67	Failed to find the function in the DS SDK.
NET_DVR_DSSDK_ERROR	68	Failed to call the API in the hardware decoding library.
NET_DVR_VOICEMONOPOLIZE	69	The sound card is exclusive.
NET_DVR_JOINMULTICASTFAILED	70	Failed to join to multicast group.
NET_DVR_CREATEDIR_ERROR	71	Failed to create log file directory.

Error Name	Error Code	Error Description
NET_DVR_BINDSOCKET_ERROR	72	Failed to bind socket.
NET_DVR_SOCKETCLOSE_ERROR	73	Socket disconnected. Network disconnected or the destination is unreachable.
NET_DVR_USERID_ISUSING	74	Operation is executing. Failed to log out.
NET_DVR_SOCKETLISTEN_ERROR	75	Failed to listen.
NET_DVR_PROGRAM_EXCEPTION	76	Program exception.
NET_DVR_WRITEFILE_FAILED	77	Failed to write file during local recording, downloading file remotely or saving picture.
NET_DVR_FORMAT_READONLY	78	The HDD is read-only. Formatting is forbidden.
NET_DVR_WITHSAMEUSERNAME	79	The user name already exists.
NET_DVR_DEVICETYPE_ERROR	80	Device model mismatches when importing parameters.
NET_DVR_LANGUAGE_ERROR	81	Language mismatches when importing parameters.
NET_DVR_PARAVERSION_ERROR	82	Software version mismatches when importing parameters.
NET_DVR_IPCHAN_NOTALIVE	83	The external IP channel is offline live view.
NET_DVR_RTSP_SDK_ERROR	84	Failed to load StreamTransClient.dll.
NET_DVR_CONVERT_SDK_ERROR	85	Failed to load SystemTransform.dll.
NET_DVR_IPC_COUNT_OVERFLOW	86	No more IP channels can access to.
NET_DVR_MAX_ADD_NUM	87	No more video tags can be added.
NET_DVR_PARAMMODE_ERROR	88	Invalid parameter mode of image enhancement.
NET_DVR_CODESPITTER_OFFLINE	89	Code distributer is offline.
NET_DVR_BACKUP_COPYING	90	Device is backing up.
NET_DVR_CHAN_NOTSUPPORT	91	This operation is not supported by the channel.

Error Name	Error Code	Error Description
NET_DVR_CALLINEINVALID	92	The height line is too concentrated, or the length line is not inclined enough.
NET_DVR_CALCANCELCONFLICT	93	Cancel calibration conflict, if the rule and global actual size filter are configured.
NET_DVR_CALPOINTOUTRANGE	94	The calibration point is out of limitation.
NET_DVR_FILTERRECTINVALID	95	The size filter does not meet the requirement.
NET_DVR_DDNS_DEVOFFLINE	96	Device has not registered to DDNS.
NET_DVR_DDNS_INTER_ERROR	97	DDNS internal error.
NET_DVR_FUNCTION_NOT_SUPPORT_OS	98	This function is not supported by this Operating system.
NET_DVR_DEC_CHAN_REBIND	99	Decoding channel binding display output is limited.
NET_DVR_INTERCOM_SDK_ERROR	100	Failed to load the two-way audio SDK of the current directory.
NET_DVR_NO_CURRENT_UPDATEFILE	101	No correct upgrade packet.
NET_DVR_USER_NOT_SUCC_LOGIN	102	Login failed.
NET_DVR_USE_LOG_SWITCH_FILE	103	The log switch file is under using.
NET_DVR_POOL_PORT_EXHAUST	104	No port can be bound in the port pool.
NET_DVR_PACKET_TYPE_NOT_SUPPORT	105	Incorrect stream packaging format.
NET_DVR_IPPARA_IPID_ERROR	106	Incorrect IPID for IP access configuration.
NET_DVR_LOAD_HCPREVIEW_SDK_ERROR	107	Failed to load the live view component.
NET_DVR_LOAD_HCVOICETALK_SDK_ERROR	108	Failed to load the audio component.
NET_DVR_LOAD_HCALARM_SDK_ERROR	109	Failed to load the alarm component.

Error Name	Error Code	Error Description
NET_DVR_LOAD_HCPLAYBACK_SDK_ERROR	110	Failed to load the playback component.
NET_DVR_LOAD_HCDISPLAY_SDK_ERROR	111	Failed to load the display component.
NET_DVR_LOAD_HCINDUSTRY_SDK_ERROR	112	Failed to load application component.
NET_DVR_LOAD_HCGENERALCFGMGR_SDK_ERROR	113	Failed to load the general configuration management component.
NET_DVR_CORE_VER_MISMATCH	121	Component version and core version mismatched when loading the component singly.
NET_DVR_CORE_VER_MISMATCH_HCPREVIEW	122	Live view component version and core version mismatched.
NET_DVR_CORE_VER_MISMATCH_HCVOICETALK	123	Audio component version and the core version mismatched.
NET_DVR_CORE_VER_MISMATCH_HCALARM	124	Alarm component version and the core version mismatched.
NET_DVR_CORE_VER_MISMATCH_HCPLAYBACK	125	Playback component version and the core version mismatched.
NET_DVR_CORE_VER_MISMATCH_HCDISPLAY	126	Display component version and the core version mismatched.
NET_DVR_CORE_VER_MISMATCH_HCINDUSTRY	127	Application component version and the core version mismatched.
NET_DVR_CORE_VER_MISMATCH_HCGENERALCFGMGR	128	General configuration management component version and the core version mismatched.
NET_DVR_COM_VER_MISMATCH_HCPREVIEW	136	Live view component version and SDK version mismatched.
NET_DVR_COM_VER_MISMATCH_HCVOICETALKy	137	Audio component version and SDK version mismatched.
NET_DVR_COM_VER_MISMATCH_HCALARM	138	Alarm component version and SDK version mismatched.

Error Name	Error Code	Error Description
NET_DVR_COM_VER_MISMATCH_HCPPLAYBACK	139	Playback component version and SDK version mismatched.
NET_DVR_COM_VER_MISMATCH_HCDISPLAY	140	Display component version and SDK version mismatched.
NET_DVR_COM_VER_MISMATCH_HCINDUSTRY	141	Application component version and SDK version mismatched.
NET_DVR_COM_VER_MISMATCH_HCGENERALCFGMGR	142	General configuration management component version and SDK version mismatched.
NET_DVR_ALIAS_DUPLICATE	150	Duplicated alias(for HiDDNS configuration).
NET_DVR_USERNAME_NOT_EXIST	152	User name does not exist (error code of network camera and network speed dome with version from 5.1.7 to 5.3.1).
NET_ERR_USERNAME_LOCKED	153	The user name is locked.
NET_DVR_INVALID_USERID	154	Invalid user ID.
NET_DVR_LOW_LOGIN_VERSION	155	The version is too low.
NET_DVR_LOAD_LIBEAY32_DLL_ERROR	156	Failed to load libeay32.dll.
NET_DVR_LOAD_SSLEAY32_DLL_ERROR	157	Failed to load ssleay32.dll.
NET_ERR_LOAD_LIBICONV	158	Failed to load libiconv.dll.
NET_ERR_SSL_CONNECT_FAILED	159	Connecting to SSL failed.
NET_DVR_TEST_SERVER_FAIL_CONNECT	165	Failed to connect to test server.
NET_DVR_NAS_SERVER_INVALID_DIR	166	Failed to load NAS server to the directory, Invalid directory, or incorrect user name and password.
NET_DVR_NAS_SERVER_NOENOUGH_PRI	167	Failed to load NAS server th the directory. No permission.

Error Name	Error Code	Error Description
NET_DVR_EMAIL_SERVER_NOT_CONFIG_DNS	168	The server uses domain name without configuring DNS, the domain name may be invalid.
NET_DVR_EMAIL_SERVER_NOT_CONFIG_GATEWAY	169	No gateway configured. Sending email may be failed.
NET_DVR_TEST_SERVER_PASSWORD_ERROR	170	Incorrect user name or password of test server.
NET_DVR_EMAIL_SERVER_CONNECT_EXCEPTION_WITH_SMTP	171	Interaction exception between device and SMTP server.
NET_DVR_FTP_SERVER_FAIL_CREATE_DIR	172	FTP server creating directory failed.
NET_DVR_FTP_SERVER_NO_WRITE_PIR	173	FTP server has no writing permission.
NET_DVR_IP_CONFLICT	174	IP conflicted.
NET_DVR_INSUFFICIENT_STORAGEPOOL_SPACE	175	Storage pool space is full.
NET_DVR_STORAGEPOOL_INVALID	176	Invalid cloud storage pool. No storage pool configured or incorrect storage pool ID.
NET_DVR_EFFECTIVENESS_REBOOT	177	Restart to take effect.
NET_ERR_ANR_ARMING_EXIST	178	The ANR arming connection already exists(the error will be returned when arming with ANR function if the private SDK protocol arming connection is established).
NET_ERR_UPLOADLINK_EXIST	179	The ANR uploading connection already exists(the error will be returned when EHome protocol and private SDK protocol do not support ANR at the same time).
NET_ERR_INCORRECT_FILE_FORMAT	180	The imported file format is incorrect.
NET_ERR_INCORRECT_FILE_CONTENT	181	The imported file content is incorrect.
NET_ERR_MAX_HRUDP_LINK	182	No more HRUDP can be connected to device.

Error Name	Error Code	Error Description
NET_ERR_MAX_PORT_MULTIPLEX	183	Maximum number of multiplexed ports reaches.
NET_ERR_CREATE_PORT_MULTIPLEX	184	Creating port multiplier failed.
NET_DVR_NONBLOCKING_CAPTURE_NOTSUPPORT	185	Non-blocking picture capture is not supported.
NET_SDK_ERR_FUNCTION_INVALID	186	Invalid function. The asynchronous mode is enabled.
NET_SDK_ERR_MAX_PORT_MULTIPLEX	187	Maximum number of multiplex ports reached.
NET_DVR_INVALID_LINK	188	Link has not been created or the link is invalid.
NET_DVR_NAME_NOT_ONLY	200	This name already exists.
NET_DVR_OVER_MAX_ARRAY	201	The number of RAID reaches the upper-limit.
NET_DVR_OVER_MAX_VD	202	The number of virtual disk reaches the upper-limit.
NET_DVR_VD_SLOT_EXCEED	203	The virtual disk slots are full.
NET_DVR_PD_STATUS_INVALID	204	The physical disk for rebuilding RAID is error.
NET_DVR_PD_BE_DEDicate_SPARE	205	The physical disk for rebuilding RAID is specified as hot spare.
NET_DVR_PD_NOT_FREE	206	The physical disk for rebuilding RAID is busy.
NET_DVR_CANNOT_MIG2NEWMODE	207	Failed to migrate the current RAID type to the new type.
NET_DVR_MIG_PAUSE	208	Migration is paused.
NET_DVR_MIG_ABOUTED	209	Migration is cancelled.
NET_DVR_EXIST_VD	210	Failed to delete RAID. Virtual disk exists in the RAID.
NET_DVR_TARGET_IN_LD_FUNCTIONAL	211	Target physical disk is a part of the virtual disk and it is working normally.

Error Name	Error Code	Error Description
NET_DVR_HD_IS_ASSIGNED_ALREADY	212	The specified physical disk is allocated as virtual disk.
NET_DVR_INVALID_HD_COUNT	213	The number of physical disks and specified RAID level mismatched.
NET_DVR_LD_IS_FUNCTIONAL	214	The RAID is normal. Failed to rebuild.
NET_DVR_BGA_RUNNING	215	Background task is executing.
NET_DVR_LD_NO_ATAPI	216	Failed to create virtual disk by ATAPI disk.
NET_DVR_MIGRATION_NOT_NEED	217	There is no need to migrate the RAID.
NET_DVR_HD_TYPE_MISMATCH	218	The physical disk type is not allowed.
NET_DVR_NO_LD_IN_DG	219	No virtual disk. Operation failed.
NET_DVR_NO_ROOM_FOR_SPARE	220	Insufficient disk space. Failed to allocate the disk as hot spare.
NET_DVR_SPARE_IS_IN_MULTI_DG	221	The disk is already allocated as the hot spare of one RAID.
NET_DVR_DG_HAS_MISSING_PD	222	No disk in the RAID.
NET_DVR_NAME_EMPTY	223	The name is empty.
NET_DVR_INPUT_PARAM	224	Incorrect input parameters.
NET_DVR_PD_NOT_AVAILABLE	225	The physical disk is not available.
NET_DVR_ARRAY_NOT_AVAILABLE	226	The RAID is not available.
NET_DVR_PD_COUNT	227	Incorrect number of physical disks.
NET_DVR_VD_SMALL	228	Insufficient virtual disk space.
NET_DVR_NO_EXIST	229	Not exist.
NET_DVR_NOT_SUPPORT	230	This operation is not supported.
NET_DVR_NOT_FUNCTIONAL	231	The RAID status is exception.
NET_DVR_DEV_NODE_NOT_FOUND	232	The device node of virtual disk does not exist.
NET_DVR_SLOT_EXCEED	233	No more slots are allowed.
NET_DVR_NO_VD_IN_ARRAY	234	No virtual disk exists in the RAID.
NET_DVR_VD_SLOT_INVALID	235	Invalid virtual disk slot.

Error Name	Error Code	Error Description
NET_DVR_PD_NO_ENOUGH_SPACE	236	Insufficient physical disk space.
NET_DVR_ARRAY_NONFUNCTION	237	Only the RAID in normal status supports to be migrated.
NET_DVR_ARRAY_NO_ENOUGH_SPACE	238	Insufficient RAID space.
NET_DVR_STOPPING_SCANNING_ARRAY	239	Pulling disk out safely or rescanning.
NET_DVR_NOT_SUPPORT_16T	240	Creating RAID with size larger than 16T is not supported.
NET_DVR_ERROR_DEVICE_NOT_ACTIVATED	250	The device is not activated (login failed.)
NET_DVR_ERROR_RISK_PASSWORD	251	Risky password.
NET_DVR_ERROR_DEVICE_HAS_ACTIVATED	252	The device is already activated.
NET_DVR_ID_ERROR	300	The configured ID is invalid.
NET_DVR_POLYGON_ERROR	301	Invalid polygon shape.
NET_DVR_RULE_PARAM_ERROR	302	Invalid rule parameters.
NET_DVR_RULE_CFG_CONFLICT	303	Configured information conflicted.
NET_DVR_CALIBRATE_NOT_READY	304	No calibration information.
NET_DVR_CAMERA_DATA_ERROR	305	Invalid camera parameters.
NET_DVR_CALIBRATE_DATA_UNFIT	306	Invalid inclination angle for calibration.
NET_DVR_CALIBRATE_DATA_CONFLICT	307	Calibration error.
NET_DVR_CALIBRATE_CALC_FAIL	308	Failed to calculate calibration parameter values of camera.
NET_DVR_CALIBRATE_LINE_OUT_RECT	309	The inputted calibration line exceeds the external sample rectangle.
NET_DVR_ENTER_RULE_NOT_READY	310	No region entrance is configured.
NET_DVR_AID_RULE_NO_INCLUDE_LANE	311	No lane configured in the traffic event rule (especially for traffic jam or driving against the traffic).

Error Name	Error Code	Error Description
NET_DVR_LANE_NOT_READY	312	Lane not configured.
NET_DVR_RULE_INCLUDE_TWO_WAY	313	Two different directions are contained in event rule.
NET_DVR_LANE_TPS_RULE_CONFLICT	314	Lane and data rule conflicted.
NET_DVR_NOT_SUPPORT_EVENT_TYPE	315	This event type is not supported.
NET_DVR_LANE_NO_WAY	316	The lane has no direction.
NET_DVR_SIZE_FILTER_ERROR	317	Invalid size of filter frame.
NET_DVR_LIB_FFL_NO_FACE	318	No face picture exists in the image inputted when positioning feature point.
NET_DVR_LIB_FFL_IMG_TOO_SMALL	319	The inputted image is too small when positioning feature point.
NET_DVR_LIB_FD_IMG_NO_FACE	320	No face picture exists in the image inputted when detecting single face picture.
NET_DVR_LIB_FACE_TOO_SMALL	321	Face picture is too small when building model.
NET_DVR_LIB_FACE_QUALITY_TOO_BAD	322	The face picture quality is too poor when building model.
NET_DVR_KEY_PARAM_ERR	323	The configured advanced parameter is incorrect.
NET_DVR_CALIBRATE_DATA_ERR	324	Calibration sample number error, or data value error, or the sample points are beyond the horizontal line.
NET_DVR_CALIBRATE_DISABLE_FAIL	325	Canceling calibration is not allowed for configured rules.
NET_DVR_VCA_LIB_FD_SCALE_OUTRANGE	326	The minimum width and height of maximum filter frame are twice or more larger than the maximum width and height of minimum filter frame.
NET_DVR_LIB_FD_REGION_TOO_LARGE	327	Too large detection region. The maximum region should be 2/3 of the image.

Error Name	Error Code	Error Description
NET_DVR_TRIAL_OVERDUE	328	Trial period is ended.
NET_DVR_CONFIG_FILE_CONFLICT	329	Device type and configuration file conflicted.
NET_DVR_FR_FPL_FAIL	330	Failed to positioning face feature points.
NET_DVR_FR_IQA_FAIL	331	Failed to test face picture quality.
NET_DVR_FR_FEM_FAIL	332	Failed to extract the face feature points.
NET_DVR_FPL_DT_CONF_TOO_LOW	333	The face detection validity is too low when positioning face feature points.
NET_DVR_FPL_CONF_TOO_LOW	334	The validity of feature points positionong is too low.
NET_DVR_E_DATA_SIZE	335	Data size mismatches.
NET_DVR_FR_MODEL_VERSION_ERR	336	Incorrect model version in face model library.
NET_DVR_FR_FD_FAIL	337	Failed to detect face in the face recognition library.
NET_DVR_FA_NORMALIZE_ERR	338	Failed to normalize face attribute.
NET_DVR_DOG_PUSTREAM_NOT_MATCH	339	Dongle type and camera type mismatched.
NET_DVR_DEV_PUSTREAM_NOT_MATCH	340	Camera version mismatches.
NET_DVR_PUSTREAM_ALREADY_EXISTS	341	This camera is already added to other channels of devices.
NET_DVR_SEARCH_CONNECT_FAILED	342	Failed to connect to face retrieval server.
NET_DVR_INSUFFICIENT_DISK_SPACE	343	Insufficient storage space.
NET_DVR_DATABASE_CONNECTION_FAILED	344	Failed to connect to database.
NET_DVR_DATABASE_ADM_PW_ERROR	345	Incorrect database user name and password.
NET_DVR_DECODE_YUV	346	Decoding failed.

Error Name	Error Code	Error Description
NET_DVR_IMAGE_RESOLUTION_ERROR	347	Invalid picture resolution
NET_DVR_CHAN_WORKMODE_ERROR	348	Invalid channel working mode.
NET_ERROR_TRUNK_LINE	711	Sub system is configured as the trunk line.
NET_ERROR_MIXED_JOINT	712	Mixed joint is not supported.
NET_ERROR_DISPLAY_SWITCH	713	Switch of display channel is not supported.
NET_ERROR_USED_BY_BIG_SCREEN	714	Decoded resource is occupied by the big screen.
NET_ERROR_USE_OTHER_DEC_RESOURCE	715	Using resources of other sub system is not allowed.
NET_ERROR_SCENE_USING	717	The scene is being used.
NET_ERR_NO_ENOUGH_DEC_RESOURCE	718	Insufficient resources for decoding.
NET_ERR_NO_ENOUGH_FREE_SHOW_RESOURCE	719	Insufficient resources for display.
NET_ERR_NO_ENOUGH_VIDEO_MEMORY	720	Insufficient video storage resources.
NET_ERR_MAX_VIDEO_NUM	721	Insufficient resources for multiple channels.
NET_ERR_WINDOW_COVER_FREE_SHOW_AND_NORMAL	722	Windows cover free display output channel and normal output channel.
NET_ERR_FREE_SHOW_WINDOW_SPLIT	723	Window division is not supported for free display windows.
NET_ERR_INAPPROPRIATE_WINDOW_FREE_SHOW	724	For the windows whose number is not integral multiple of the number of output channels, free display is not supported.
NET_DVR_TRANSPARENT_WINDOW_NOT_SUPPORT_SPLIT	725	For windows whose transparency configuration is enabled, window division is not supported.

Error Name	Error Code	Error Description
NET_DVR_SPLIT_WINDOW_NOT_SUPPORT_TRANSPARENT	726	For windows whose window division is enabled, transparency configuration is not supported.
NET_ERR_TERMINAL_BUSY	780	The terminal busy.
NET_DVR_FUNCTION_RESOURCE_USAGE_ERROR	791	Failed to enable this function. The resources is occupied by other functions.
NET_DVR_DEV_NET_OVERFLOW	800	Network traffic is out of the limitation.
NET_DVR_STATUS_RECORDFILE_WRITING_NOT_LOCK	801	Failed to lock. The video file is recording.
NET_DVR_STATUS_CANT_FORMAT_LITTLE_DISK	802	Failed to format HDD. The HDD space is too small.
NET_SDK_ERR_REMOTE_DISCONNECT	803	Failed to connect to the remote terminal.
NET_SDK_ERR_RD_ADD_RD	804	Spare server cannot be added to spare server.
NET_SDK_ERR_BACKUP_DISK_EXCEPT	805	Backup disk exception.
NET_SDK_ERR_RD_LIMIT	806	No more spare server can be added.
NET_SDK_ERR_ADDED_RD_IS_WD	807	The added spare server is a working server.
NET_SDK_ERR_ADD_ORDER_WRONG	808	Adding flow error.
NET_SDK_ERR_WD_ADD_WD	809	Working server cannot be added to working server.
NET_SDK_ERR_WD_SERVICE_EXCETP	810	CVR service exception (For N+1 mode, it refers to CVR working server exception).
NET_SDK_ERR_RD_SERVICE_EXCETP	811	Spare CVR server exception.
NET_SDK_ERR_ADDED_WD_IS_RD	812	The added working server is spare server.
NET_SDK_ERR_PERFORMANCE_LIMIT	813	The performance reaches the upper-limit.
NET_SDK_ERR_ADDED_DEVICE_EXIST	814	This device already exists.

Error Name	Error Code	Error Description
NET_SDK_ERR_INQUEST_RESUMING	815	Inquest resuming.
NET_SDK_ERR_RECORD_BACKUPING	816	Inquest video backing up.
NET_SDK_ERR_DISK_PLAYING	817	Playing.
NET_SDK_ERR_INQUEST_STARTED	818	Inquest started.
NET_SDK_ERR_LOCAL_OPERATING	819	Locally operating.
NET_SDK_ERR_INQUEST_NOT_START	820	Inquest is not started.
NET_SDK_ERR_CHAN_AUDIO_BIND	821	The channel is not bound or binding two-way audio failed.
NET_DVR_N_PLUS_ONE_MODE	822	Ddevice is in N+1 mode. Cloud storage is not supported.
NET_DVR_CLOUD_STORAGE_OPENED	823	Cloud storage mode is enbaled.
NET_DVR_ERR_OPER_NOT_ALLOWED	824	Operation failed. The device is in N+0 taken over status.
NET_DVR_ERR_NEED_RELOCATE	825	The device is in N+0 taken over status. Get re-positioning information and try again.
NET_SDK_ERR_IR_PORT_ERROR	830	IR output error.
NET_SDK_ERR_IR_CMD_ERROR	831	IR output port command number error
NET_SDK_ERR_NOT_INQUESTING	832	Device is not in inquest status.
NET_SDK_ERR_INQUEST_NOT_PAUSED	833	Device is not in paused status.
NET_DVR_CHECK_PASSWORD_MISTAKE_ERROR	834	Incorrect verification code.
NET_DVR_CHECK_PASSWORD_NULL_ERROR	835	Verification code is required.
NET_DVR_UNABLE_CALIB_ERROR	836	Failed to calibrate.
NET_DVR_PLEASE_CALIB_ERROR	837	Calibration first.
NET_DVR_ERR_PANORAMIC_CAL_EMPTY	838	Panoramic calibration is empty in Flash.

Error Name	Error Code	Error Description
NET_DVR_ERR_CALIB_FAIL_PLEASEAGAIN	839	Calibration failed, please try again.
NET_DVR_ERR_DETECTION_LINE	840	Rule line configuration error. Please try again and make sure the line is within the red region.
NET_DVR_EXCEED_FACE_IMAGES_ERROR	843	No more face pictures can be added.
NET_DVR_ANALYSIS_FACE_IMAGES_ERROR	844	Picture recognition failed.
NET_ERR_ALARM_INPUT_OCCUPIED	845	A<-1 alarm number is used for triggering vehicle capture.
NET_DVR_FACELIB_DATABASE_ERROR	846	Database version in face picture library mismatched.
NET_DVR_FACELIB_DATA_ERROR	847	Face picture library data error.
NET_DVR_FACE_DATA_ID_ERROR	848	Invalid face data PID.
NET_DVR_FACELIB_ID_ERROR	849	Invalid face picture library ID.
NET_DVR_EXCEED_FACE_LIBRARY_ERROR	850	No more face picture libraries can be established..
NET_DVR_PIC_ANALYSIS_NO_TARGET_ERROR	851	No target recognized in the picture.
NET_DVR_SUBPIC_ANALYSIS_MODELING_ERROR	852	Sub picture modeling failed.
NET_DVR_PIC_ANALYSIS_NO_RESOURCE_ERROR	853	No VCA engine supports picture secondary recognition.
NET_DVR_ANALYSIS_ENGINES_NO_RESOURCE_ERROR	854	No VCA engine.
NET_DVR_ANALYSIS_ENGINES_USAGE_EXCEED_ERROR	855	Overload. The engine CPU reached 100%.
NET_DVR_EXCEED_HUMANMISINFO_FILTER_ENABLED_ERROR	856	No more false alarm channel can be enabled.
NET_DVR_NAME_ERROR	857	Name error.
NET_DVR_NAME_EXIST_ERROR	858	The name already exists.

Error Name	Error Code	Error Description
NET_DVR_FACELIB_PIC_IMPORTING_ERROR	859	The pictures is importing to face picture library.
NET_DVR_PIC_FORMAT_ERROR	864	Invalid picture format.
NET_DVR_PIC_RESOLUTION_INVALID_ERROR	865	Invalid picture resolution.
NET_DVR_PIC_SIZE_EXCEED_ERROR	866	The picture size is too large.
NET_DVR_PIC_ANALYSIS_TARGRT_NUM_EXCEED_ERROR	867	Too many targets in the picture.
NET_DVR_ANALYSIS_ENGINES_LOADING_ERROR	868	Initializing analysis engine.
NET_DVR_ANALYSIS_ENGINES_ABNORMA_ERROR	869	Analysis engine exception.
NET_DVR_ANALYSIS_ENGINES_FACELIB_IMPORTING	870	Analysis engine is importing pictures to face picture library.
NET_DVR_NO_DATA_FOR_MODELING_ERROR	871	No data for modeling.
NET_DVR_FACE_DATA_MODELING_ERROR	872	Device is modeling picture. Concurrent processing is not supported.
NET_ERR_FACELIBDATA_OVERLIMIT	873	No more face picture can be added to the device (the data of imported face picture library)
NET_DVR_ANALYSIS_ENGINES_ASSOCIATED_CHANNEL	874	Channel is linked to the analysis engine.
NET_DVR_ERR_CUSTOMID_LEN	875	The minimum length of upper layer custom ID is 32 bytes.
NET_DVR_ERR_CUSTOMFACELIBID_REPEAT	876	The applied custom face picture library ID is duplicated
NET_DVR_ERR_CUSTOMHUMANID_REPEAT	877	The applied custom person ID is duplicated.
NET_DVR_ERR_URL_DOWNLOAD_FAIL	878	URL download failed.

Error Name	Error Code	Error Description
NET_DVR_ERR_URL_DOWNLOAD_NOTSTART	879	URL download has not started.
NET_DVR_CFG_FILE_SECRETKEY_ERROR	880	The security verification key of configuration file is error.
NET_DVR_THERMOMETRY_REGION_OVERSTEP_ERROR	883	Invalid thermometry region
NET_DVR_ERR_TOO_SHORT_CALIBRATING_TIME	894	Too short time for calibration.
NET_DVR_ERR_AUTO_CALIBRATE_FAILED	895	Auto calibration failed.
NET_DVR_ERR_VERIFICATION_FAILED	896	Verification failed.
NET_DVR_NO_TEMP_SENSOR_ERROR	897	No temperature sensor.
NET_DVR_PUPIL_DISTANCE_OVERSIZE_ERROR	898	The pupil distance is too large.
NET_ERR_WINCHAN_IDX	901	Window channel index error.
NET_ERR_WIN_LAYER	902	Window layer number error(the count of window layers on a single screen exceeds the max number).
NET_ERR_WIN_BLK_NUM	903	Window block number error(the count of screens that single window overlays exceeds the max number).
NET_ERR_OUTPUT_RESOLUTION	904	The output resolution error.
NET_ERR_LAYOUT	905	Layout index error.
NET_ERR_INPUT_RESOLUTION	906	The input resolution is not supported.
NET_ERR_SUBDEVICE_OFFLINE	907	The sub-device is off-line.
NET_ERR_NO_DECODE_CHAN	908	There is no free decoding channel.
NET_ERR_MAX_WINDOW_ABILITY	909	The upper limit of window number.
NET_ERR_ORDER_ERROR	910	Calling order error.
NET_ERR_PLAYING_PLAN	911	Be playing plan.
NET_ERR_DECODER_USED	912	Decoder board is being used.

Error Name	Error Code	Error Description
NET_ERR_OUTPUT_BOARD_DATA_OVERFLOW	913	Output board data overflow
NET_ERR_SAME_USER_NAME	914	Duplicate user name
NET_ERR_INVALID_USER_NAME	915	Invalid user name
NET_ERR_MATRIX_USING	916	Input matrix is in use.
NET_ERR_DIFFERENT_CHAN_TYPE	917	Different channel type (the type of matrix output channel mismatches that of the controller input channel)
NET_ERR_INPUT_CHAN_BINDED	918	Input channel has been bound by other matrix
NET_ERR_BINDED_OUTPUT_CHAN_OVERFLOW	919	The matrix output channels in use exceeded the number bound by matrix and controller
NET_ERR_MAX_SIGNAL_NUM	920	Number of input signals reached upper limit
NET_ERR_INPUT_CHAN_USING	921	Input channel is in use
NET_ERR_MANAGER_LOGON	922	Administrator has logged in, operation failed
NET_ERR_USERALREADY_LOGON	923	The user has logged in, operation failed
NET_ERR_LAYOUT_INIT	924	Scene is initializing, operation failed
NET_ERR_BASEMAP_SIZE_NOT_MATCH	925	Base image size does not match
NET_ERR_WINDOW_OPERATING	926	Window is in other operation, operation failed
NET_ERR_SIGNAL_UPLIMIT	927	Number of signal source window reached upper limit
NET_ERR_WINDOW_SIZE_OVERLIMIT	943	The window size exceeds the limit.
NET_ERR_MAX_WIN_OVERLAP	951	The number of windows overlap has reached the maximum limit.
NET_ERR_STREAMID_CHAN_BOTH_VALID	952	stream ID and channel number are both valid.

Error Name	Error Code	Error Description
NET_ERR_NO_ZERO_CHAN	953	The device has no zero channel.
NEED_RECONNECT	955	Need redirection (for transcoding system)
NET_ERR_NO_STREAM_ID	956	The stream ID does not exist.
NET_DVR_TRANS_NOT_START	957	The transcoding has not been started.
NET_ERR_MAXNUM_STREAM_ID	958	The number of stream ID has reached the maximum limit.
NET_ERR_WORKMODE_MISMATCH	959	The work mode does not match with the requirement.
NET_ERR_MODE_IS_USING	960	It Has been working in current mode.
NET_ERR_DEV_PROGRESSING	961	The device is in processing
NET_ERR_PASSIVE_TRANSCODING	962	It is in transcoding.
NET_DVR_ERR_WINDOW_SIZE_PLACE	975	Wrong window position.
NET_DVR_ERR_RGIONAL_RESTRICTIONS	976	Screen distance exceeds the limit.
NET_DVR_ERR_CLOSE_WINDOWS	984	Operation failed. Close the window first.
NET_DVR_ERR_MATRIX_LOOP_ABILITY	985	Beyond the cycle decoding capacity.
NET_DVR_ERR_MATRIX_LOOP_TIME	986	Invalid cycle decoding time.
NET_DVR_ERR_LINKED_OUT_ABILITY	987	No more linked camera can be added.
NET_ERR_RESOLUTION_NOT_SUPPORT_ODD_VOUT	990	The resolution is not supported (odd No.).
NET_ERR_RESOLUTION_NOT_SUPPORT_EVEN_VOUT	991	The resolution is not supported (even No.).
NET_ERR_UnitConfig_Failed	998	Unit configuration failed.
XML_ABILITY_NOTSUPPORT	1000	Getting capability node is not supported
XML_ANALYZE_NOENOUGH_BUF	1001	Not enough output memory

Error Name	Error Code	Error Description
XML_ANALYZE_FIND_LOCALXML_ERROR	1002	Failed to find related local xml
XML_ANALYZE_LOAD_LOCALXML_ERROR	1003	Loading local xml error
XML_NANLYZE_DVR_DATA_FORMAT_ERROR	1004	Device capability data format error
XML_ANALYZE_TYPE_ERROR	1005	Capability set type error
XML_ANALYZE_XML_NODE_ERROR	1006	XML capability node format error
XML_INPUT_PARAM_ERROR	1007	Input capability XML node value error
XML_VERSION_MISMATCH	1008	XML version does not match
NET_ERR_TRANS_CHAN_START	1101	Transparent channel has been open, operation failed
NET_ERR_DEV_UPGRADING	1102	Device is upgrading
NET_ERR_MISMATCH_UPGRADE_PACK_TYPE	1103	Upgrade pack type does not match
NET_ERR_DEV_FORMATTING	1104	Device is formatting
NET_ERR_MISMATCH_UPGRADE_PACK_VERSION	1105	Upgrade pack version does not match
NET_ERR_PT_LOCKED	1106	PT is locked.
NET_DVR_ERR_ILLEGAL_VERIFICATION_CODE	1111	Illegal verification code. Change the verification code.
NET_DVR_ERR_LACK_VERIFICATION_CODE	1112	No verification code. Enter the verification code.
NET_DVR_ERR_FORBIDDEN_IP	1113	The IP address cannot be configured.
NET_DVR_ERR_HTTP_BKN_EXCEED_ONE	1125	Up to one channel's ANR function can be enabled.
NET_DVR_ERR_FORMATTING_FAILED	1131	Formatting HDD failed.
NET_DVR_ERR_ENCRYPTED_FORMATTING_FAILED	1132	Formatting encrypted HDD failed.
NET_DVR_ERR_WRONG_PASSWORD	1133	Verifying password of SD card failed. Incorrect password.

Error Name	Error Code	Error Description
NET_ERR_SEARCHING_MODULE	1201	Searching peripherals.
NET_ERR_REGISTERING_MODULE	1202	Registering external module
NET_ERR_GETTING_ZONES	1203	Getting arming region parameter
NET_ERR_GETTING_TRIGGERS	1204	Getting trigger
NET_ERR_ARMED_STATUS	1205	System is in arming status
NET_ERR_PROGRAM_MODE_STATUS	1206	System is in programming mode
NET_ERR_WALK_TEST_MODE_STATUS	1207	System is in pacing measuring mode
NET_ERR_BYPASS_STATUS	1208	Bypass status
NET_ERR_DISABLED_MODULE_STATUS	1209	Function not enabled
NET_ERR_NOT_SUPPORT_OPERATE_ZONE	1210	Operation is not supported by arming region
NET_ERR_NOT_SUPPORT_MOD_MODULE_ADDR	1211	Module address cannot be modified
NET_ERR_UNREGISTERED_MODULE	1212	Module is not registered
NET_ERR_PUBLIC_SUBSYSTEM_ASSOCIATE_SELF	1213	Public sub system associate with its self
NET_ERR_EXCEEDS_ASSOCIATE_SUBSYSTEM_NUM	1214	Number of associated public sub system reached upper limit
NET_ERR_BE_ASSOCIATED_BY_PUBLIC_SUBSYSTEM	1215	Sub system is associated by other public sub system
NET_ERR_ZONE_FAULT_STATUS	1216	Arming region is in failure status
NET_ERR_SAME_EVENT_TYPE	1217	Same event type exists in enable event trigger alarm output and disable event trigger alarm output
NET_ERR_ZONE_ALARM_STATUS	1218	Arming region is in alarm status
NET_ERR_EXPANSION_BUS_SHORT_CIRCUIT	1219	Extension bus short-circuit
NET_ERR_PWD_CONFLICT	1220	Password conflict, e.g., lock password is identical with duress password

Error Name	Error Code	Error Description
NET_ERR_DETECTOR_GISTERED_BY_OTHER_ZONE	1221	Detector has been registered by other arming regions
NET_ERR_DETECTOR_GISTERED_BY_OTHER_PU	1222	Detector has been registered by other hosts
NET_ERR_DETECTOR_DISCONNECT	1223	Detector offline
NET_ERR_CALL_BUSY	1224	Device in call
NET_ERR_FILE_NAME	1357	File name error, empty or invalid
NET_ERR_BROADCAST_BUSY	1358	Device in broadcast
NET_DVR_ERR_LANENUM_EXCEED	1400	Over the number of lanes.
NET_DVR_ERR_PRAREA_EXCEED	1401	Recognition area is too large.
NET_DVR_ERR_LIGHT_PARAM	1402	Signal lamp access parameters error.
NET_DVR_ERR_LANE_LINE_INVALID	1403	Lane configuration error.
NET_DVR_ERR_STOP_LINE_INVALID	1404	Stop line configuration error.
NET_DVR_ERR_LEFTORRIGHT_LINE_INVALID	1405	Turn left / right boundary configuration error.
NET_DVR_ERR_LANE_NO_REPEAT	1406	Overlay lane number repetition.
NET_DVR_ERR_PRAREA_INVALID	1407	The polygon does not meet the requirements.
NET_DVR_ERR_LIGHT_NUM_EXCEED	1408	Video detection of traffic light signal exceeds the maximum number of.
NET_DVR_ERR_SUBLIGHT_NUM_INVALID	1409	Video detection of traffic signal lamp lights are not legitimate
NET_DVR_ERR_LIGHT_AREASIZE_INVALID	1410	The size of the video detection of traffic light input signal lamp is not valid.
NET_DVR_ERR_LIGHT_COLOR_INVALID	1411	The color of the video detection of traffic light input signal lamp color is not legitimate.
NET_DVR_ERR_LIGHT_DIRECTION_INVALID	1412	The direction property of the video detection of traffic light input light is not valid.
NET_DVR_ERR_LACK_IOABLITY	1413	Lack of IO ablity.

Error Name	Error Code	Error Description
NET_DVR_ERR_FTP_PORT	1414	FTP port error.
NET_DVR_ERR_FTP_CATALOGUE	1415	FTP catalogue error.
NET_DVR_ERR_FTP_UPLOAD_TYPE	1416	FTP upload type error.
NET_DVR_ERR_FLASH_PARAM_WRITE	1417	Setting param flash write error.
NET_DVR_ERR_FLASH_PARAM_READ	1418	Getting param flash read error.
NET_DVR_ERR_PICNAME_DELIMITER	1419	Pic name delimiter error.
NET_DVR_ERR_PICNAME_ITEM	1420	Pic name item error.
NET_DVR_ERR_PLATE_RECOGNIZE_TYPE	1421	Plate recognize type error.
NET_DVR_ERR_CAPTURE_TIMES	1422	Capture times error.
NET_DVR_ERR_LOOP_DISTANCE	1423	Loop distance error.
NET_DVR_ERR_LOOP_INPUT_STATUS	1424	Loop input status error.
NET_DVR_ERR_RELATE_IO_CONFLICT	1425	Related IO conflict.
NET_DVR_ERR_INTERVAL_TIME	1426	Interval time error.
NET_DVR_ERR_SIGN_SPEED	1427	Sign speed error.
NET_DVR_ERR_PIC_FLIP	1428	Flip is used.
NET_DVR_ERR_RELATE_LANE_NUMBER	1429	Related lane number error.
NET_DVR_ERR_TRIGGER_MODE	1430	Trigger mode error.
NET_DVR_ERR_DELAY_TIME	1431	Delay time error.
NET_DVR_ERR_EXCEED_RS485_COUNT	1432	Exceed RS485 count.
NET_DVR_ERR_RADAR_TYPE	1433	Radar type error.
NET_DVR_ERR_RADAR_ANGLE	1434	Radar angle error.
NET_DVR_ERR_RADAR_SPEED_VALID_TIME	1435	Radar speed valid time error.
NET_DVR_ERR_RADAR_LINE_CORRECT	1436	Radar line correct error.

Error Name	Error Code	Error Description
NET_DVR_ERR_RADAR_CONST_CORRECT	1437	Radar const correct error.
NET_DVR_ERR_RECORD_PARAM	1438	Record param error.
NET_DVR_ERR_LIGHT_WITHOUT_COLOR_AND_DIRECTION	1439	Light number and other param error.
NET_DVR_ERR_LIGHT_WITHOUT_DETECTION_REGION	1440	Light number and detection region error.
NET_DVR_ERR_RECOGNIZE_PROVINCE_PARAM	1441	Plate recognize Province param error.
NET_DVR_ERR_SPEED_TIMEOUT	1442	IO Speed TimeOut Param error.
NET_DVR_ERR_NTP_TIMEZONE	1443	NTP TimeZone Param error.
NET_DVR_ERR_NTP_INTERVAL_TIME	1444	NTP Interval Time error.
NET_DVR_ERR_NETWORK_CARD_NUM	1445	Network Card Num error.
NET_DVR_ERR_DEFAULT_ROUTE	1446	Default Route error.
NET_DVR_ERR_BONDING_WORK_MODE	1447	Banding Work Mode error.
NET_DVR_ERR_SLAVE_CARD	1448	Sub-Card error.
NET_DVR_ERR_PRIMARY_CARD	1449	Primary Card error.
NET_DVR_ERR_DHCP_PPOE_WORK	1450	DHCP and PPOE not Meanwhile start.
NET_DVR_ERR_NET_INTERFACE	1451	Net Interface invalid.
NET_DVR_ERR_MTU	1452	Invalid MTU parameters.
NET_DVR_ERR_NETMASK	1453	Netmask address invalid.
NET_DVR_ERR_IP_INVALID	1454	IP address invalid.
NET_DVR_ERR_MULTICAST_IP_INVALID	1455	Multicast IP address invalid.
NET_DVR_ERR_GATEWAY_INVALID	1456	Gateway address invalid.
NET_DVR_ERR_DNS_INVALID	1457	DNS Param invalid.
NET_DVR_ERR_ALARMHOST_IP_INVALID	1458	AlarmHost IP invalid.

Error Name	Error Code	Error Description
NET_DVR_ERR_IP_CONFLICT	1459	IP address Conflict.
NET_DVR_ERR_NETWORK_SEGMENT	1460	IP not support Multi Network segment.
NET_DVR_ERR_NETPORT	1461	NetPort error.
NET_DVR_ERR_PPPOE_NOSUPPORT	1462	PPPoE is not supported.
NET_DVR_ERR_DOMAINNAME_NOSUPPORT	1463	Not Support Domain Name.
NET_DVR_ERR_NO_SPEED	1464	Speed Not Enabled.
NET_DVR_ERR_IOSTATUS_INVALID	1465	IO Status invalid.
NET_DVR_ERR_BURST_INTERVAL_INVALID	1466	Burst Interval invalid.
NET_DVR_ERR_RESERVE_MODE	1467	Reserve Mode invalid.
NET_DVR_ERR_LANE_NO	1468	Lane No error.
NET_DVR_ERR_COIL_AREA_TYPE	1469	Coil Area Type error.
NET_DVR_ERR_TRIGGER_AREA_PARAM	1470	Trigger Area Param error.
NET_DVR_ERR_SPEED_LIMIT_PARAM	1471	Speed Limit Param error.
NET_DVR_ERR_LANE_PROTOCOL_TYPE	1472	Lane Protocol Type error.
NET_DVR_ERR_INTERVAL_TYPE	1473	Capture Interval Type error.
NET_DVR_ERR_INTERVAL_DISTANCE	1474	Capture Interval Distance error.
NET_DVR_ERR_RS485_ASSOCIATE_DEVTYPE	1475	Rs485 Associate DevType error.
NET_DVR_ERR_RS485_ASSOCIATE_LANENO	1476	Rs485 Associate LaneNo error.
NET_DVR_ERR_LANENO_ASSOCIATE_MULTIRS485	1477	LaneNo Associate MulitRs485 error.
NET_DVR_ERR_LIGHT_DETECTION_REGION	1478	Light Detection Region error.
NET_DVR_ERR_DN2D_NOSUPPORT	1479	UnSupport Capture Frame 2D Noise Reduction.

Error Name	Error Code	Error Description
NET_DVR_ERR_IRISMODE_NOSUPPORT	1480	UnSupport scene Mode.
NET_DVR_ERR_WB_NOSUPPORT	1481	UnSupport White Balance Mode.
NET_DVR_ERR_IO_EFFECTIVENESS	1482	IO Effectiveness invalid.
NET_DVR_ERR_LIGHTNO_MAX	1483	Access Detector Lights Red / Yellow Overrun.
NET_DVR_ERR_LIGHTNO_CONFLICT	1484	Access Detector Lights Red / Yellow Conflict.
NET_DVR_ERR_CANCEL_LINE	1485	Trigger straight line error.
NET_DVR_ERR_STOP_LINE	1486	Subject line area stop line error.
NET_DVR_ERR_RUSH_REDLIGHT_LINE	1487	Red light trigger lines error.
NET_DVR_ERR_IOOUTNO_MAX	1488	IO out port error.
NET_DVR_ERR_IOOUTNO_AHEADTIME_MAX	1489	IO out ahead time error.
NET_DVR_ERR_IOOUTNO_IOWORKTIME	1490	IO out inwork time error.
NET_DVR_ERR_IOOUTNO_FREQMULTI	1491	IO out frequency multiplication error.
NET_DVR_ERR_IOOUTNO_DUTYRATE	1492	IO out duty rate error.
NET_DVR_ERR_VIDEO_WITH_EXPOSURE	1493	IO out work mode error.
NET_DVR_ERR_PLATE_BRIGHTNESS_WITHOUT_FLASHDET	1494	Plate enable in plate compensate mode on.
NET_DVR_ERR_RECOGNIZE_TYPE_PARAM	1495	Recognize Type error.
NET_DVR_ERR_PALTE_RECOGNIZE_AREA_PARAM	1496	Plate Recognize Area Param error.
NET_DVR_ERR_PORT_CONFLICT	1497	Port Conflict.
NET_DVR_ERR_LOOP_IP	1498	IP cannot be the loopback address.
NET_DVR_ERR_DRIVELINE_SENSITIVE	1499	Driveline sensitivity error.
NET_ERR_VQD_TIME_CONFLICT	1500	The time period conflict.

Error Name	Error Code	Error Description
NET_ERR_VQD_PLAN_NO_EXIST	1501	The diagnostic plan of VQD dese not exist.
NET_ERR_VQD_CHAN_NO_EXIST	1502	The channel dese not exist.
NET_ERR_VQD_CHAN_MAX	1503	The total number of VQD plans exceeds the max limit.
NET_ERR_VQD_TASK_MAX	1504	The total number of VQD tasks exceeds the max limit.
NET_DVR_ERR_EXCEED_MAX_CAPTURE_TIMES	1600	Capture times exceed 2 in flash mode.
NET_DVR_ERR_REDAR_TYPE_CONFLICT	1601	Radar type conflict.
NET_DVR_ERR_LICENSE_PLATE_NULL	1602	The license plate is null.
NET_DVR_ERR_WRITE_DATABASE	1603	Failed to write data into the database.
NET_DVR_ERR_LICENSE_EFFECTIVE_TIME	1604	The effective time of license plate error.
NET_DVR_ERR_PRERECORDED_STARTTIME_LONG	1605	The pre recorded start time is greater than the number of illegal capture.
NET_DVR_ERR_TRIGGER_RULE_LINE	1606	Trigger rule line error.
NET_DVR_ERR_LEFTRIGHT_TRIGGERLINE_NOTVERTICAL	1607	Left and right trigger line is not vertical.
NET_DVR_ERR_FLASH_LAMP_MODE	1608	Flash lamp mode error.
NET_DVR_ERR_ILLEGAL_SNAPSHOT_NUM	1609	Illegal capture number error.
NET_DVR_ERR_ILLEGAL_DETECTION_TYPE	1610	Illegal detection type error.
NET_DVR_ERR_POSITIVEBACK_TRIGGERLINE_HIGH	1611	Positive back to trigger line height error.
NET_DVR_ERR_MIXEDMODE_CAPTYPE_ALLTARGETS	1612	Mixed mode only supports capture type all targets.
NET_DVR_ERR_CARSIGNSPEED_GREATERTHAN_LIMITSPEED	1613	Car sign speed greater than speed limit value.

Error Name	Error Code	Error Description
NET_DVR_ERR_BIGCARSIGNSPEED_GREATER_THAN_LIMITSPEED	1614	Big car sign speed limit greater than speed limit value.
NET_DVR_ERR_BIGCARSIGNSPEED_GREATER_THAN_CARSIGNSPEED	1615	Big car sign speed limit is greater than the car sign speed limit value.
NET_DVR_ERR_BIGCARLIMITSPEED_GREATER_THAN_CARLIMITSPEED	1616	Big car speed limit value is greater than the car speed limit value.
NET_DVR_ERR_BIGCARLOWSPEEDLIMIT_GREATER_THAN_CARLOWSPEEDLIMIT	1617	Big car low speed limit value is greater than the car low speed limit value.
NET_DVR_ERR_CARLIMITSPEED_GREATER_THAN_EXCEPT_HIGHSPEED	1618	Car speed limit greater than exception high speed value.
NET_DVR_ERR_BIGCARLIMITSPEED_GREATER_THAN_EXCEPT_HIGHSPEED	1619	Big car speed limit greater than exception high speed value.
NET_DVR_ERR_STOPLINE_MORE_THAN_TRIGGERLINE	1620	Stopping more than straight lines trigger lines.
NET_ERR_TIME_OVERLAP	1900	Time periods overlap
NET_ERR_HOLIDAY_PLAN_OVERLAP	1901	Holiday plan overlap
NET_ERR_CARDNO_NOT_SORT	1902	Card number is not sorted
NET_ERR_CARDNO_NOT_EXIST	1903	Card number does not exist
NET_ERR_ILLEGAL_CARDNO	1904	Card number error
NET_ERR_ZONE_ALARM	1905	Arming region is in arming status (parameter cannot be modified)
NET_ERR_ZONE_OPERATION_NOT_SUPPORT	1906	Arming region does not support the operation
NET_ERR_INTERLOCK_ANTI_CONFLICT	1907	Interlock and anti-passback configuration conflict
NET_ERR_DEVICE_CARD_FULL	1908	Card full (return after card reached 10,000)
NET_ERR_HOLIDAY_GROUP_DOWNLOAD	1909	Failed to download holiday group
NET_ERR_LOCAL_CONTROL_OFF	1910	Distributed access controller offline

Error Name	Error Code	Error Description
NET_ERR_LOCAL_CONTROL_DISADD	1911	Distributed access controller is not added
NET_ERR_LOCAL_CONTROL_HASADD	1912	Distributed access controller is added
NET_ERR_LOCAL_CONTROL_DOORNO_CONFLICT	1913	Conflict with added distributed access controller
NET_ERR_LOCAL_CONTROL_COMMUNICATION_FAIL	1914	Distributed access controller communication failed
NET_ERR_OPERAND_INEXISTENCE	1915	Operation object does not exist (operation to door, alarm output, alarm input, return when the object is not added)
NET_ERR_LOCAL_CONTROL_OVER_LIMIT	1916	Distributed access controller exceeded device capability upper limit
NET_ERR_DOOR_OVER_LIMIT	1917	Door exceeded device capability upper limit
NET_ERR_ALARM_OVER_LIMIT	1918	Alarm input and output exceeded device capability upper limit
NET_ERR_LOCAL_CONTROL_ADDRESS_INCONFORMITY_TYPE	1919	Distributed access controller address does not match with type
NET_ERR_NOT_SUPPORT_ONE_MORE_CARD	1920	not support one person multi-card
NET_ERR_DELETE_NO_EXISTENCE_FACE	1921	The face picture does not exist.
NET_ERR_DOOR_SPECIAL_PASSWORD_REPEAT	1922	Repeated door door duress code, the super password, or the dismiss code.
NET_ERR_AUTH_CODE_REPEAT	1923	Repeated device authentication code
NET_ERR_DEPLOY_EXCEED_MAX	1924	No more devices can be armed.
NET_ERR_NOT_SUPPORT_DEL_FP_BY_ID	1925	The fingerprint module does not support deleting fingerprint by finger ID.
NET_ERR_TIME_RANGE	1926	Invalid range of the effective period.
NET_ERR_CAPTURE_TIMEOUT	1927	Collection timed out.
NET_ERR_LOW_SCORE	1928	Low quality of collected data.

Error Name	Error Code	Error Description
NET_ERR_OFFLINE_CAPTURING	1929	The device is collecting data offline and cannot respond.
NET_DVR_ERR_OUTDOOR_COMMUNICATION	1950	Communication exception with outdoor terminal
NET_DVR_ERR_ROOMNO_UNDEFINED	1951	Room number is not set
NET_DVR_ERR_NO_CALLING	1952	No call
NET_DVR_ERR_RINGING	1953	Ringling
NET_DVR_ERR_IS_CALLING_NOW	1954	Call in progress
NET_DVR_ERR_LOCK_PASSWORD_WRONG	1955	Incorrect smart lock password
NET_DVR_ERR_CONTROL_LOCK_FAILURE	1956	Lock control failure
NET_DVR_ERR_CONTROL_LOCK_OVERTIME	1957	Lock control timed out
NET_DVR_ERR_LOCK_DEVICE_BUSY	1958	Smart lock device busy
NET_DVR_ERR_UNOPEN_REMOTE_LOCK_FUNCTION	1959	Remote lock control not enabled
NET_DVR_ERR_FILE_NOT_COMPLETE	2100	Downloaded file is incomplete
NET_DVR_ERR_IPC_EXIST	2101	The camera already exists
NET_DVR_ERR_ADD_IPC	2102	Camera has been added to the channel
NET_DVR_ERR_OUT_OF_RES	2103	Not enough network bandwidth
NET_DVR_ERR_CONFLICT_TO_LOCALIP	2104	IP address of camera conflicts with that of DVR
NET_DVR_ERR_IP_SET	2105	Invalid IP address
NET_DVR_ERR_PORT_SET	2106	Invalid port number
NET_ERR_WAN_NOTSUPPORT	2107	Not in the same LAN, cannot set security question or export GUID file
NET_ERR_MUTEX_FUNCTION	2108	Mutually exclusive function

Error Name	Error Code	Error Description
NET_ERR_QUESTION_CONFIGNUM	2109	Error in number of security question configurations
NET_ERR_FACECHAN_NORESOURCE	2110	All the face VCA channels are occupied.
NET_ERR_DATA_CALLBACK	2111	Data is calling back.
NET_ERR_ATM_VCA_CHAN_IS_RELATED	2112	The VCA channel is already linked.
NET_ERR_ATM_VCA_CHAN_IS_OVERLAPED	2113	The VCA channel is already overlaid.
NET_ERR_FACE_CHAN_UNOVERLAP_EACH_OTHER	2114	The face channels cannot be overlaid.
NET_DVR_SMD_ENCODING_NORESOURCE	2116	Insufficient SMD encoding resource
NET_DVR_SMD_DECODING_NORESOURCE	2117	Insufficient SMD decoding resource
NET_DVR_FACELIB_DATA_PROCESSING	2118	Face picture library data is in processing
NET_DVR_ERR_LARGE_TIME_DIFFERENCE	2119	There is a great time difference between device and server.
NET_DVR_NO_SUPPORT_WITH_PLAYBACK	2120	It is not supported. Playback is enabled.
NET_DVR_CHANNEL_NO_SUPPORT_WITH_SMD	2121	It is not supported. SMD of channel is enabled.
NET_DVR_CHANNEL_NO_SUPPORT_WITH_FD	2122	It is not supported. Face capture of channel is enabled.
NET_DVR_ILLEGAL_PHONE_NUMBER	2123	Invalid telephone number
NET_DVR_ILLEGAL_CERTIFICATE_NUMBER	2124	Invalid ID No.
NET_DVR_ERR_CHANNEL_RESOLUTION_NO_SUPPORT	2125	The channel resolution is not supported
NET_DVR_ERR_CHANNEL_COMPRESSION_NO_SUPPORT	2126	The channel encoding format is not supported

Error Name	Error Code	Error Description
NET_DVR_ERR_CLUSTER_DEVICE_TOO_LESS	2127	Deleting is not allowed. The number of devices is not enough
NET_DVR_ERR_CLUSTER_DEL_DEVICE_CM_PAYLOAD	2128	Deleting is not allowed. The device is cluster host.
NET_DVR_ERR_CLUSTER_DEVNUM_OVER_UPPER_LIMIT	2129	No more devices can be added.
NET_DVR_ERR_CLUSTER_DEVICE_TYPE_INCONFORMITY	2130	Device type mismatched.
NET_DVR_ERR_CLUSTER_DEVICE_VERSION_INCONFORMITY	2131	Device version mismatched.
NET_DVR_ERR_CLUSTER_IP_CONFLICT	2132	Cluster system IP address conflict: ipv4 address conflict, invalid ipv6.
NET_DVR_ERR_CLUSTER_IP_INVALID	2133	Invalid cluster system IP address: invalid ipv4, invalid ipv6.
NET_DVR_ERR_CLUSTER_PORT_CONFLICT	2134	Cluster system port conflict
NET_DVR_ERR_CLUSTER_PORT_INVALID	2135	Invalid cluster system port
NET_DVR_ERR_CLUSTER_USERNAEM_OR_PASSWORD_INVALID	2136	Invalid user name or password
NET_DVR_ERR_CLUSTER_DEVICE_ALREADY_EXIST	2137	The device already exists.
NET_DVR_ERR_CLUSTER_DEVICE_NOT_EXIST	2138	The device does not exist.
NET_DVR_ERR_CLUSTER_NON_CLUSTER_MODE	2139	The device working mode is not the cluster mode .
NET_DVR_ERR_CLUSTER_IP_NOT_SAME_LAN	2140	IP addresses are in different LAN. Building cluster or extending capacity for NVRs in different LAN is not allowed.
NET_DVR_ERR_IDENTITY_KEY	2147	Incorrect interaction password
NET_DVR_MISSING_IDENTITY_KEY	2148	Interaction password is missing

Error Name	Error Code	Error Description
NET_DVR_ERR_CAPTURE_PACKAGE_FAILED	2141	Capturing packets failed.
NET_DVR_ERR_CAPTURE_PACKAGE_PROCESSING	2142	Capturing packet.
NET_DVR_ERR_SAFETY_HELMET_NO_RESOURCE	2143	No enough hard hat detection resource.
NET_DVR_NO_SUPPORT_WITH_ABSTRACT	2144	This function is not supported. Video synopsis is already enabled.
NET_DVR_INSUFFICIENT_DEEP_LEARNING_RESOURCES	2146	No more deep learning resources can be added.
NET_DVR_NO_SUPPORT_WITH_PERSON_DENSITY_DETECT	2149	People gathering density is enabled, it is not supported
NET_DVR_IPC_RESOLUTION_OVERFLOW	2150	The network camera resolution is too large
NET_DVR_IPC_BITRATE_OVERFLOW	2151	The network camera bitrate is too large
NET_DVR_ERR_INVALID_TASKID	2152	Invalid taskID
NET_DVR_PANEL_MODE_NOT_CONFIG	2153	The ATM panel mode is not configured.
NET_DVR_NO_HUMAN_ENGINES_RESOURCE	2154	No enough engine resource
NET_DVR_ERR_TASK_NUMBER_OVERFLOW	2155	No more task data is allowed
NET_DVR_ERR_COLLISION_TIME_OVERFLOW	2156	Collision time is over the limit
NET_DVR_ERR_EVENT_NOTSUPPORT	2159	Subscribing alarm/event is not supported.
NET_DVR_IPC_NUM_REACHES_LIMIT	2184	The max. number of network camera channels reached.
NET_DVR_IOT_NUM_REACHES_LIMIT	2185	The max. number of IoT channels reached
NET_DVR_IOT_CHANNEL_DEVICE_EXIST	2186	Device of the IoT channel already exists.

Error Name	Error Code	Error Description
NET_DVR_IOT_CHANNEL_DEVICE_NOT_EXIST	2187	Device of the IoT channel does not exist.
NET_DVR_INVALID_IOT_PROTOCOL_TYPE	2188	Invalid IoT protocol type
NET_DVR_INVALID_EZVIZ_SECRET_KEY	2189	Invalid verification code
NET_DVR_DUPLICATE_IOT_DEVICE	2190	Duplicated IoT device
NET_DVR_ERROR_NEED_DOUBLE_VERIFICATION	2206	Double verification is required
NET_DVR_NO_DOUBLE_VERIFICATION_USER	2207	No double verification user
NET_DVR_TIMESPAN_NUM_OVER_LIMIT	2209	Max. number of time buckets reached
NET_DVR_CHANNEL_NUM_OVER_LIMIT	2210	Max. number of channels reached
NET_DVR_NO_SEARCH_ID_RESOURCE	2211	Insufficient searchID resources
NET_DVR_SWITCH_TIMEDIFF_LESS_LIMIT	2249	Time difference between power on and off should be less than 10 minutes.
NET_DVR_NO_SUPPORT_DELETE_STRANGER_LIB	2262	Deleting stranger library is not supported
NET_DVR_NO_SUPPORT_CREATE_STRANGER_LIB	2263	Creating stranger library is not supported
NET_DVR_SSD_FILE_SYSTEM_ERROR	2266	SSD file system error
NET_DVR_INSUFFICIENT_SSD__FOR_FPD	2267	Insufficient SSD space for person frequency detection
NET_DVR_SMRDISK_NOT_SUPPORT_RAID	2269	SMR disk does not support RAID.
NET_DVR_ERR_NOTSUPPORT_DEICING	3001	Device does not support deicing function under current status.(Deicing function is only supported under the power status of POE+, AC24V, and DC12V).

Error Name	Error Code	Error Description
NET_DVR_ERR_THERMENABLE_CLOSE	3002	Temperature measurement function is not enabled. (The enable function in NET_DVR_THERMOMETRY_BASICPARAM is not turned on)
NET_DVR_ERR_PANORAMIC_LIMIT_OPERATED	3004	Panoramic map and limit cannot be operated at same time
NET_DVR_ERR_SMARTH264_ROI_OPERATED	3005	SmartH264 and ROI cannot be enabled at the same time.
NET_DVR_ERR_RULENUM_LIMIT	3006	No more rules can be added.
NET_DVR_ERR_LASER_DEICING_OPERATED	3007	Laser and deicing function cannot be enabled at the same time.
NET_DVR_ERR_OFFDIGITALZOOM_OR_MINZOOMLIMIT	3008	Please disable the digital zoom function or set the zoom limit to the minimum value. Otherwise, when enabling smoke and fire detection, behavior analysis, ship detection, defective point correction, temperature measurement, smoke and fire shielding function, this error code will be prompted.
NET_DVR_SYNCHRONIZEFOV_ERROR	3010	Field of view synchronization failed.
NET_DVR_RULE_SHIELDMASK_CONFLICT_ERROR	3013	The rule region conflicts with the shielded area.
NET_DVR_ERR_NO_SAFETY_HELMET_REGION	3501	The hard hat detection area is not configured.
NET_DVR_ERR_UNCLOSED_SAFETY_HELMET	3502	The hard hat detection is enabled.
NET_DVR_UPLOAD_HBDLIBID_ERROR	3504	Incorrect ID of human body picture library (incorrect HBDID or customHBDID)

RTSP Communication Library Related Errors

Error Name	Error Code	Error Description
NET_DVR_RTSP_ERROR_NOENOUGHPRI	401	Authentication failed: if server returns 401, it will change to this error code
NET_DVR_RTSP_ERROR_ALLOC_RESOURCE	402	Failed to allocate the resource
NET_DVR_RTSP_ERROR_PARAMETER	403	Parameter error
NET_DVR_RTSP_ERROR_NO_URL	404	The assigned URL does not exist: when the server returns 404, SDK turns to this error code. E.g. the channel is not available, or the channel does not support sub stream
NET_DVR_RTSP_ERROR_FORCE_STOP	406	The user forces to exit midway
NET_DVR_RTSP_GETPORTFAILED	407	RTSP port getting error.
NET_DVR_RTSP_DESCRIBERROR	410	RTSP DESCRIBE communicate error
NET_DVR_RTSP_DESCRIBESENDTIMEOUT	411	Sending "RTSP DESCRIBE" is timeout.
NET_DVR_RTSP_DESCRIBESENDERERROR	412	Failed to send "RTSP DESCRIBE".
NET_DVR_RTSP_DESCRIBERECVTIMEOUT	413	Receiving "RTSP DESCRIBE" is timeout.
NET_DVR_RTSP_DESCRIBERECVDATALOST	414	Receiving data of "RTSP DESCRIBE" error.
NET_DVR_RTSP_DESCRIBERECVERROR	415	Failed to receive "RTSP DESCRIBE".
NET_DVR_RTSP_DESCRIBESERVERERR	416	"RTSP DESCRIBE, the device returns the error code: 501 (failed to allocate the resource in the device)
NET_DVR_RTSP_SETUPERROR	420	(or 419), RTSP SETUP interaction error. Generally, it is that the address(URL) returned by the device is not accessible, or it is rejected by the server
NET_DVR_RTSP_SETUPSENDTIMEOUT	421	Sending "RTSP SETUP" is timeout.

Error Name	Error Code	Error Description
NET_DVR_RTSP_SETUPSENDERERROR	422	Sending "RTSP SETUP" error.
NET_DVR_RTSP_SETUPRECVTIMEOUT	423	Receiving "RTSP SETUP" is timeout.
NET_DVR_RTSP_SETUPRECVDATALOST	424	Receiving data of "RTSP SETUP" error.
NET_DVR_RTSP_SETUPRECVERERROR	425	Failed to receive "RTSP SETUP".
NET_DVR_RTSP_OVER_MAX_CHAN	426	"RTSP SETUP" device returns the error that values 401 or 501. It exceeds the max connection number.
NET_DVR_RTSP_PLAYERERROR	430	RTSP PLAY interaction error.
NET_DVR_RTSP_PLAYSENDTIMEOUT	431	Sending "RTSP PLAY" is timeout.
NET_DVR_RTSP_PLAYSENDERERROR	432	Sending "RTSP PLAY" error.
NET_DVR_RTSP_PLAYRECVTIMEOUT	433	Receiving "RTSP PLAY" is timeout.
NET_DVR_RTSP_PLAYRECVDATALOST	434	Receiving data of "RTSP PLAY" error.
NET_DVR_RTSP_PLAYRECVERERROR	435	Failed to receive "RTSP PLAY".
NET_DVR_RTSP_PLAYSERVERERR	436	"RTSP PLAY" device returns the error that values 401 or 501.
NET_DVR_RTSP_TEARDOWNERROR	440	RTSP TEARDOWN interaction error.
NET_DVR_RTSP_TEARDOWNSENDTIMEOUT	441	Sending "RTSP TEARDOWN" is timeout.
NET_DVR_RTSP_TEARDOWNSENDERERROR	442	Sending "RTSP TEARDOWN" error.
NET_DVR_RTSP_TEARDOWNRECVTIMEOUT	443	Receiving "RTSP TEARDOWN" is timeout.
NET_DVR_RTSP_TEARDOWNRECVDATALOST	444	Receiving data of "RTSP TEARDOWN" error.
NET_DVR_RTSP_TEARDOWNRECVERERROR	445	Failed to receive "RTSP TEARDOWN".
NET_DVR_RTSP_TEARDOWNSERVERERR	446	"RTSP TEARDOWN" device returns the error that values 401 or 501.

Software Decoding Library Related Errors

Error Name	Error Code	Error Description
NET_PLAYM4_NOERROR	500	No error.
NET_PLAYM4_PARA_OVER	501	Input parameter is invalid.
NET_PLAYM4_ORDER_ERROR	502	API calling order error.
NET_PLAYM4_TIMER_ERROR	503	Failed to create multimedia clock.
NET_PLAYM4_DEC_VIDEO_ERROR	504	Failed to decode video data.
NET_PLAYM4_DEC_AUDIO_ERROR	505	Failed to decode audio data.
NET_PLAYM4_ALLOC_MEMORY_ERROR	506	Failed to allocate memory.
NET_PLAYM4_OPEN_FILE_ERROR	507	Failed to open the file.
NET_PLAYM4_CREATE_OBJ_ERROR	508	Failed to create thread event.
NET_PLAYM4_CREATE_DDRAW_ERROR	509	Failed to create DirectDraw object.
NET_PLAYM4_CREATE_OFFSCREEN_ERROR	510	Failed to create backstage cache for OFFSCREEN mode.
NET_PLAYM4_BUF_OVER	511	Buffer overflow, failed to input stream.
NET_PLAYM4_CREATE_SOUND_ERROR	512	Failed to create audio equipment.
NET_PLAYM4_SET_VOLUME_ERROR	513	Failed to set the volume.
NET_PLAYM4_SUPPORT_FILE_ONLY	514	This API can be called only for file playback mode.
NET_PLAYM4_SUPPORT_STREAM_ONLY	515	This API can be called only when playing stream.
NET_PLAYM4_SYS_NOT_SUPPORT	516	Not support by the system. Decoder can only work on the system above Pentium 3.
NET_PLAYM4_FILEHEADER_UNKNOWN	517	There is no file header.
NET_PLAYM4_VERSION_INCORRECT	518	The version mismatch between decoder and encoder.

Error Name	Error Code	Error Description
NET_PLAYM4_INIT_DECODER_ERROR	519	Failed to initialize the decoder.
NET_PLAYM4_CHECK_FILE_ERROR	520	The file is too short, or the stream data is unknown.
NET_PLAYM4_INIT_TIMER_ERROR	521	Failed to initialize multimedia clock.
NET_PLAYM4_BLT_ERROR	522	BLT failure.
NET_PLAYM4_UPDATE_ERROR	523	Failed to update overlay surface
NET_PLAYM4_OPEN_FILE_ERROR_MULTI	524	Failed to open video & audio stream file.
NET_PLAYM4_OPEN_FILE_ERROR_VIDEO	525	Failed to open video stream file.
NET_PLAYM4_JPEG_COMPRESS_ERROR	526	JPEG compression error.
NET_PLAYM4_EXTRACT_NOT_SUPPORT	527	Don't support the version of this file.
NET_PLAYM4_EXTRACT_DATA_ERROR	528	Extract video data failed.

Container Format Conversion Library Related Errors

Error Name	Error Code	Error Description
NET_CONVERT_ERROR_NOT_SUPPORT	581	This container format is not supported.

Two Way Audio Library Related Errors

Error Name	Error Code	Error Description
NET_AUDIOINTERCOM_OK	600	No error.
NET_AUDIOINTECOM_ERR_NOTSUPPORT	601	Not support.
NET_AUDIOINTECOM_ERR_ALLOC_MEMORY	602	Memory allocation error.
NET_AUDIOINTECOM_ERR_PARAMETER	603	Parameter error.
NET_AUDIOINTECOM_ERR_CALL_ORDER	604	API calling order error.

Error Name	Error Code	Error Description
NET_AUDIOINTECOM_ERR_FIND_DEVICE	605	No audio device
NET_AUDIOINTECOM_ERR_OPEN_DEVICE	606	Failed to open the audio device
NET_AUDIOINTECOM_ERR_NO_CONTEXT	607	Context error.
NET_AUDIOINTECOM_ERR_NO_WAVFILE	608	WAV file error.
NET_AUDIOINTECOM_ERR_INVALID_TYPE	609	The type of WAV parameter is invalid
NET_AUDIOINTECOM_ERR_ENCODE_FAIL	610	Failed to encode data
NET_AUDIOINTECOM_ERR_DECODE_FAIL	611	Failed to decode data
NET_AUDIOINTECOM_ERR_NO_PLAYBACK	612	Failed to play audio
NET_AUDIOINTECOM_ERR_DENOISE_FAIL	613	Failed to denoise
NET_AUDIOINTECOM_ERR_UNKOWN	619	Unknown

QoS Stream Control Library Related Errors

Error Name	Error Code	Error Description
NET_QOS_ERR_SCHEDPARAMS_BAD_MINIMUM_INTERVAL	678	Incorrect predefined minimum interval.
NET_QOS_ERR_SCHEDPARAMS_BAD_FRACTION	679	Incorrect predefined score.
NET_QOS_ERR_SCHEDPARAMS_INVALID_BANDWIDTH	680	Invalid predefined bandwidth.
NET_QOS_ERR_PACKET_TOO_BIG	687	The packet size is too large.
NET_QOS_ERR_PACKET_LENGTH	688	Invalid packet size.
NET_QOS_ERR_PACKET_VERSION	689	Incorrect packet versio information.
NET_QOS_ERR_PACKET_UNKNOW	690	Unknown packet.
NET_QOS_ERR_OUTOFMEM	695	Out of memory.
NET_QOS_ERR_LIB_NOT_INITIALIZED	696	The library is not initialized.
NET_QOS_ERR_SESSION_NOT_FOUND	697	No session found.
NET_QOS_ERR_INVALID_ARGUMENTS	698	Invalid parameters.

Error Name	Error Code	Error Description
NET_QOS_ERROR	699	QoS Stream Control Library error.
NET_QOS_OK	700	No error.

NPQ (Network Protocol Quality) Related Error

Error Name	Error Code	Error Description
NET_ERR_NPQ_PARAM	8001	NPQ library: Incorrect parameter.
NET_ERR_NPQ_SYSTEM	8002	NPQ library: Operating system error.
NET_ERR_NPQ_GENRAL	8003	NPQ library: Internal error.
NET_ERR_NPQ_PRECONDITION	8004	NPQ library: Calling sequence error.
NET_ERR_NPQ_NOTSUPPORT	8005	NPQ library: This function is not supported.
NET_ERR_NPQ_NOTCALLBACK	8100	No data is called back.
NET_ERR_NPQ_LOADLIB	8101	Loading NPQ library failed.
NET_ERR_NPQ_STREAM_CLOSE	8104	The NPQ function of this stream is not enabled.
NET_ERR_NPQ_MAX_LINK	8110	No more streaming channel's NPQ function can be enabled.
NET_ERR_NPQ_STREAM_CFG_CONFLICT	8111	The configured encoding parameters conflicted.

A.5 Response Codes of Text Protocol

The response codes returned during the text protocol integration is based on the status codes of HTTP. 7 kinds of status codes are predefined, including 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid Message Format), 6 (Invalid Message Content), and 7 (Reboot Required). Each kind of status code contains multiple sub status codes, and the response codes are in a one-to-one correspondence with the sub status codes.

StatusCode=1

SubStatusCode	Error Code	Description
ok	0x1	Operation completed.
riskPassword	0x10000002	Risky password.
armProcess	0x10000005	Arming process.

StatusCode=2

Sub Status Code	Error Code	Description
noMemory	0x20000001	Insufficient memory.
serviceUnavailable	0x20000002	The service is not available.
upgrading	0x20000003	Upgrading.
deviceBusy	0x20000004	The device is busy or no response.
reConnectIpc	0x20000005	The video server is reconnected.
transferUpgradePackageFailed	0x20000006	Transmitting device upgrade data failed.
startUpgradeFailed	0x20000007	Starting upgrading device failed.
getUpgradeProcessfailed.	0x20000008	Getting upgrade status failed.
certificateExist	0x2000000B	The Authentication certificate already exists.

StatusCode=3

Sub Status Code	Error Code	Description
deviceError	0x30000001	Hardware error.
badFlash	0x30000002	Flash operation error.
28181Uninitialized	0x30000003	The 28181 configuration is not initialized.
socketConnectError	0x30000005	Connecting to socket failed.

Sub Status Code	Error Code	Description
receiveError	0x30000007	Receive response message failed.
deletePictureError	0x3000000A	Deleting picture failed.
pictureSizeExceedLimit	0x3000000C	Too large picture size.
clearCacheError	0x3000000D	Clearing cache failed.
updateDatabasError	0x3000000F	Updating database failed.
searchDatabaseError	0x30000010	Searching in the database failed.
writeDatabaseError	0x30000011	Writing to database failed.
deleteDatabaseError	0x30000012	Deleting database element failed.
searchDatabaseElementError	0x30000013	Getting number of database elements failed.
cloudAutoUpgradeException	0x30000016	Downloading upgrade packet from cloud and upgrading failed.
HBPException	0x30001000	HBP exception.
UDEPException	0x30001001	UDEP exception
elasticSearchException	0x30001002	Elastic exception.
kafkaException	0x30001003	Kafka exception.
HBaseException	0x30001004	Hbase exception.
sparkException	0x30001005	Spark exception.
yarnException	0x30001006	Yarn exception.
cacheException	0x30001007	Cache exception.
trafficException	0x30001008	Monitoring point big data server exception.
faceException	0x30001009	Human face big data server exception.
SSDFileSystemsIsError	0x30001013	SSD file system error (Error occurs when it is non-Ext4 file system)

Sub Status Code	Error Code	Description
insufficientSSDCapacityForFPD	0x30001014	Insufficient SSD space for person frequency detection.
wifiException	0x3000100A	Wi-Fi big data server exception
structException	0x3000100D	Video parameters structure server exception.
noLinkageResource	0x30001015	Insufficient linkage resources.
engineAbnormal	0x30002015	Engine exception.
engineInitialization	0x30002016	Initializing the engine.
algorithmLoadingFailed	0x30002017	Loading the model failed.
algorithmDownloadFailed	0x30002018	Downloading the model failed.
algorithmDecryptionFailed	0x30002019	Decrypting the model failed.
unboundChannel	0x30002020	Delete the linked channel to load the new model.
unsupportedResolution	0x30002021	Invalid resolution.
unsupportedStreamType	0x30002022	Invalid stream type.
insufficientDecRes	0x30002023	Insufficient decoding resources.
insufficientEnginePerformance	0x30002024	Insufficient engine performance (The number of channels to be analyzed exceeds the engine's capability).
improperResolution	0x30002025	Improper resolution (The maximum resolution allowed is 4096×4096).
improperPicSize	0x30002026	Improper picture size (The maximum size allowed is 5MB).
URLDownloadFailed	0x30002027	Downloading the picture via the URI failed.
unsupportedImageFormat	0x30002028	Invalid picture format (Only JPG is supported currently).

Sub Status Code	Error Code	Description
unsupportedPollingIntervalTime	0x30002029	Invalid polling interval (The interval should be more than 10s).
exceedImagesNumber	0x30002030	The number of pictures exceeds the limit (The platform can apply 1 to 100 picture URIs per time, the maximum number allowed is 100).
unsupportedMPID	0x30002031	The applied MPID does not exist in the device, so updating this MPID is not supported.
modelPackageNotMatchLabel	0x30002032	The model and the description file mismatch.
modelPackageNotMatchTask	0x30002033	The task and the model type mismatch.
insufficientSpace	0x30002034	Insufficient space (When the number of model packages does not reach the maximum number allowed but their size together exceeds the free space, the model packages cannot be added).
engineUnLoadingModelPackage	0x30002035	Applying the task failed. This engine is not linked to a model package (Canceling the linkage failed, this engine is not linked to a model package).
engineWithModelPackage	0x30002036	Linking the engine to this model package failed. The engine has been linked to another model package. Please cancel their linkage first.
modelPackageDelete	0x30002037	Linking the model package failed. The model package has been deleted.

Sub Status Code	Error Code	Description
deleteTaskFailed	0x30002038	Deleting the task failed (It is returned when the user fails to end a task).
modelPackageNumberslimited	0x30002039	Adding the model package failed. The number of model package has reached the maximum number allowed.
modelPackageDeleteFailed	0x30002040	Deleting the model package failed.
noArmingResource	0x30001016	Insufficient arming resources.
calibrationTimeout	0x30002051	Calibration timed out.
captureTimeout	0x30006000	Data collection timed out.
lowScore	0x30006001	Low quality of collected data.
uploadingFailed	0x30007004	Uploading failed.

StatusCode=4

Sub Status Code	Error Code	Description
notSupport	0x40000001	Not supported.
lowPrivilege	0x40000002	No permission.
badAuthorization	0x40000003	Authentication failed.
methodNotAllowed	0x40000004	Invalid HTTP method.
notSetHdiskRedund	0x40000005	Setting spare HDD failed.
invalidOperation	0x40000006	Invalid operation.
notActivated	0x40000007	Inactivated.
hasActivated	0x40000008	Activated.
certificateAlreadyExist	0x40000009	The certificate already exists.
operateFailed	0x4000000F	Operation failed.
USBNotExist	0x40000010	USB device is not connected.
upgradePackageMorethan2GB	0x40001000	Up to 2GB upgrade package is allowed to be uploaded.

Sub Status Code	Error Code	Description
IDNotExist	0x40001001	The ID does not exist.
interfaceOperationError	0x40001002	API operation failed.
synchronizationError	0x40001003	Synchronization failed.
synchronizing	0x40001004	Synchronizing.
importError	0x40001005	Importing failed.
importing	0x40001006	Importing.
fileAlreadyExists	0x40001007	The file already exists.
invalidID	0x40001008	Invalid ID.
backupnodeNotAllowedLog	0x40001009	Accessing to backup node is not allowed.
exportingError	0x4000100A	Exporting failed.
exporting	0x4000100B	Exporting.
exportEnded	0x4000100C	Exporting stopped.
exported	0x4000100D	Exported.
IPOccupied	0x4000100E	The IP address is already occupied.
IDAlreadyExists	0x4000100F	The ID already exists.
exportItemsExceedLimit	0x40001010	No more items can be exported.
noFiles	0x40001011	The file does not exist.
beingExportedByAnotherUser	0x40001012	Being exported by others.
needReAuthentication	0x40001013	Authentication is needed after upgrade.
unitAddNotOnline	0x40001015	The added data analysis server is offline.
unitControl	0x40001016	The data analysis server is already added.
analysis unitFull	0x40001017	No more data analysis server can be added.
unitIDError	0x40001018	The data analysis server ID does not exist.
unitExit	0x40001019	The data analysis server already exists in the list.

Sub Status Code	Error Code	Description
unitSearch	0x4000101A	Searching data analysis server in the list failed.
unitNotOnline	0x4000101B	The data analysis server is offline.
unitInfoError	0x4000101C	Getting data analysis server information failed.
unitGetNodeInfoError	0x4000101D	Getting node information failed.
unitGetNetworkInfoError	0x4000101E	Getting the network information of data analysis server failed
unitSetNetworkInfoError	0x4000101F	Setting the network information of data analysis server failed
setSmartNodeInfoError	0x40001020	Setting node information failed.
setUnitNetworkInfoError	0x40001021	Setting data analysis server network information failed.
unitRestartCloseError	0x40001022	Rebooting or shutting down data analysis server failed.
virtualIPnotAllowed	0x40001023	Adding virtual IP address is not allowed.
unitInstalled	0x40001024	The data analysis server is already installed.
badSubnetMask	0x40001025	Invalid subnet mask.
uintVersionMismatched	0x40001026	Data analysis server version mismatches.
deviceModelMismatched	0x40001027	Adding failed. Device model mismatches.
unitAddNotSelf	0x40001028	Adding peripherals is not allowed.
noValidUnit	0x40001029	No valid data analysis server.
unitNameDuplicate	0x4000102A	Duplicated data analysis server name.
deleteUnitFirst	0x4000102B	Delete the added data analysis server of the node first.
getLocalInfoFailed	0x4000102C	Getting the server information failed.
getClientAddedNodeFailed	0x4000102D	Getting the added node information of data analysis server failed.
taskExit	0x4000102E	The task already exists.
taskInitError	0x4000102F	Initializing task failed.

Sub Status Code	Error Code	Description
taskSubmitError	0x40001030	Submitting task failed.
taskDelError	0x40001031	Deleting task failed.
taskPauseError	0x40001032	Pausing task failed.
taskContinueError	0x40001033	Starting task failed.
taskSeverNoCfg	0x40001035	Full-text search server is not configured.
taskPicSeverNoCfg	0x40001036	The picture server is not configured.
taskStreamError	0x40001037	Streaming information exception.
taskRecSDK	0x40001038	History recording is not supported.
taskCasaError	0x4000103A	Cascading is not supported.
taskVCARuleError	0x4000103B	Invalid VCA rule.
taskNoRun	0x4000103C	The task is not executed.
unitLinksNoStorageNode	0x4000103D	No node is linked with the data analysis server. Configure the node first.
searchFailed	0x4000103E	Searching video files failed.
searchNull	0x4000103F	No video clip.
userScheOffline	0x40001040	The task scheduler service is offline.
updateTypeUnmatched	0x40001041	The upgrade package type mismatches.
userExist	0x40001043	The user already exists.
userCannotDelAdmin	0x40001044	The administrator cannot be deleted.
userInexistence	0x40001045	The user name does not exist.
userCannotCreateAdmin	0x40001046	The administrator cannot be created.
monitorCamExceed	0x40001048	Up to 3000 cameras can be added.
monitorCunitOverLimit	0x40001049	Adding failed. Up to 5 lower-levels are supported by the control center.
monitorReginOverLimit	0x4000104A	Adding failed. Up to 5 lower-levels are supported by the area.
monitorArming	0x4000104B	The camera is already armed. Disarm the camera and try again.

Sub Status Code	Error Code	Description
monitorSyncCfgNotSet	0x4000104C	The system parameters are not configured.
monitorFdSyncing	0x4000104E	Synchronizing. Try again after completing the synchronization.
monitorParseFailed	0x4000104F	Parsing camera information failed.
monitorCreatRootFailed	0x40001050	Creating resource node failed.
deleteArmingInfo	0x40001051	The camera is already . Disarm the camera and try again.
cannotModify	0x40001052	Editing is not allowed. Select again.
cannotDel	0x40001053	Deletion is not allowed. Select again.
deviceExist	0x40001054	The device already exists.
IPErrorConnectFailed	0x40001056	Connection failed. Check the network port.
cannotAdd	0x40001057	Only the capture cameras can be added.
serverExist	0x40001058	The server already exists.
fullTextParamError	0x40001059	Incorrect full-text search parameters.
storParamError	0x4000105A	Incorrect storage server parameters.
picServerFull	0x4000105B	The storage space of picture storage server is full.
NTPUnconnect	0x4000105C	Connecting to NTP server failed. Check the parameters.
storSerConnectFailed	0x4000105D	Connecting to storage server failed. Check the network port.
storSerLoginFailed	0x4000105E	Logging in to storage server failed. Check the user name and password.
searchSerConnectFailed	0x4000105F	Connecting to full-text search server failed. Check the network port.
searchSerLoginFailed	0x40001060	Logging in to full-text search server failed. Check the user name and password.
kafkaConnectFailed	0x40001061	Connecting to Kafka failed. Check the network port.

Sub Status Code	Error Code	Description
mgmtConnectFailed	0x40001062	Connecting to system failed. Check the network port.
mgmtLoginFailed	0x40001063	Logging in to system failed. Check the user name and password.
TDAConnectFailed	0x40001064	Connecting to traffic data access server failed. Checking the server status.
86sdkConnectFailed	0x40001065	Connecting to listening port of iVMS-8600 System failed. Check the parameters.
nameExist	0x40001066	Duplicated server name.
batchProcessFailed	0x40001067	Processing in batch failed.
IDNotExist	0x40001068	The server ID does not exist.
serviceNumberReachesLimit	0x40001069	No more service can be added.
invalidServiceType.	0x4000106A	Invalid service type.
clusterGetInfo	0x4000106B	Getting cluster group information failed.
clusterDelNode	0x4000106C	Deletion node failed.
clusterAddNode	0x4000106D	Adding node failed.
clusterInstalling	0x4000106E	Creating cluster...Do not operate.
clusterUninstall	0x4000106F	Reseting cluster...Do not operate.
clusterInstall	0x40001070	Creating cluster failed.
clusterIpError	0x40001071	Invalid IP address of task scheduler server.
clusterNotSameSeg	0x40001072	The main node and sub node must be in the same network segment.
clusterVirIpError	0x40001073	Automatically getting virtual IP address failed. Enter manually.
clusterNodeUnadd	0x40001074	The specified main (sub) node is not added.
clusterNodeOffline	0x40001075	The task scheduler server is offline.
nodeNotCurrentIP	0x40001076	The analysis node of the current IP address is required when adding main and sub nodes.
addNodeNetFailed	0x40001077	Adding node failed. The network disconnected.

Sub Status Code	Error Code	Description
needTwoMgmtNode	0x40001078	Two management nodes are required when adding main and sub nodes.
ipConflict	0x40001079	The virtual IP address and data analysis server's IP address conflicted.
ipUsed	0x4000107A	The virtual IP address has been occupied.
cloudAnalyseOnline	0x4000107B	The cloud analytic server is online.
virIP&mainIPnotSame NetSegment	0x4000107C	The virtual IP address is not in the same network segment with the IP address of main/sub node.
getNodeDispatchInfoFailed	0x4000107D	Getting node scheduler information failed.
unableModifyManagementNetworkIP	0x4000107E	Editing management network interface failed. The analysis board is in the cluster.
notSpecifyVirtualIP	0x4000107F	Virtual IP address should be specified for main and sub cluster.
armingFull	0x40001080	No more device can be armed.
armingNoFind	0x40001081	The arming information does not exist.
disArming	0x40001082	Disarming failed.
getArmingError	0x40001084	Getting arming information failed.
refreshArmingError	0x40001085	Refreshing arming information failed.
ArmingPlateSame	0x40001086	The license plate number is repeatedly armed.
ArmingParseXLSError	0x40001087	Parsing arming information file failed.
ArmingTimeError	0x40001088	Invalid arming time period.
ArmingSearchTimeError	0x40001089	Invalid search time period.
armingRelationshipReachesLimit	0x4000108A	No more relation can be created.
duplicateArmingName	0x4000108B	The relation name already exists.
noMoreArmingListAdded	0x4000108C	No more blocklist library can be armed.

Sub Status Code	Error Code	Description
noMoreCamerasAdded	0x4000108D	No more camera can be armed.
noMoreArmingListAddedWithCamera	0x4000108E	No more library can be linked to the camera.
noMoreArmingPeriodAdded	0x4000108F	No more time period can be added to the arming schedule.
armingPeriodsOverlapped	0x40001090	The time periods in the arming schedule are overlapped.
noArmingAlarmInfo	0x40001091	The alarm information does not exist.
armingAlarmUnRead	0x40001092	Getting number of unread alarms failed.
getArmingAlarmError	0x40001093	Getting alarm information failed.
searchByPictureTimeout	0x40001094	Searching picture by picture timeout. Search again.
comparisonTimeRangeError	0x40001095	Comparison time period error.
selectMonitorNumberUpperLimit	0x40001096	No more monitoring point ID can be filtered.
noMoreComparisonTasksAdded	0x40001097	No more comparison task can be executed at the same time.
GetComparisonResultFailed	0x40001098	Getting comparison result failed.
comparisonTypeError	0x40001099	Comparison type error.
comparisonUnfinished	0x4000109A	The comparison is not completed.
facePictureModelInvalid	0x4000109B	Invalid face model.
duplicateLibraryName.	0x4000109C	The library name already exists.
noRecord	0x4000109D	No record found.
countingRecordsFailed.	0x4000109E	Calculate the number of records failed.
getHumanFaceFrameFailed	0x4000109F	Getting face thumbnail from the picture failed.
modelingFailed.	0x400010A0	Modeling face according to picture URL failed.

Sub Status Code	Error Code	Description
1V1FacePictureComparisonFailed	0x400010A1	Comparison 1 VS 1 face picture failed.
libraryArmed	0x400010A2	The blocklist library is armed.
licenseExceedLimit	0x400010A3	Dongle limited.
licenseExpired	0x400010A4	Dongle expired.
licenseDisabled	0x400010A5	Unavailable dongle.
licenseNotExist	0x400010A6	The dongle does not exist.
SessionExpired	0x400010A7	Session expired .
beyondConcurrentLimit	0x400010A8	Out of concurrent limit.
stopSync	0x400010A9	Synchronization stopped.
getProgressFailed	0x400010AA	Getting progress failed.
uploadExtraCaps	0x400010AB	No more files can be uploaded.
timeRangeError	0x400010AC	Time period error.
dataPortNotConnected	0x400010AD	The data port is not connected.
addClusterNodeFailed	0x400010AE	Adding to the cluster failed. The device is already added to other cluster.
taskNotExist	0x400010AF	The task does not exist.
taskQueryFailed	0x400010B0	Searching task failed.
modifyTimeRuleFailed	0x400010B2	The task already exists. Editing time rule is not allowed.
modifySmartRuleFailed	0x400010B3	The task already exists. Editing VAC rule is not allowed.
queryHistoryVideoFailed	0x400010B4	Searching history video failed.
addDeviceFailed	0x400010B5	Adding device failed.
addVideoFailed	0x400010B6	Adding video files failed.
deleteAllVideoFailed	0x400010B7	Deleting all video files failed.
createVideoIndexFailed	0x400010B8	Indexing video files failed.
videoCheckTypeFailed	0x400010B9	Verifying video files types failed.

Sub Status Code	Error Code	Description
configStructuredAddressFailed	0x400010BA	Configuring IP address of structured server failed.
configPictureServerAddressFailed	0x400010BB	Configuring IP address of picture stored server failed.
storageServiceIPNotExist	0x400010BD	The storage server IP address does not exist.
syncBackupDatabaseFailed	0x400010BE	Synchronizing sub database failed. Try again.
syncBackupNTPTimeFailed	0x400010BF	Synchronizing NTP time of sub server failed.
clusterNotSelectLoopbackAddress	0x400010C0	Loopback address is not supported by the main or sub cluster.
addFaceRecordFailed	0x400010C1	Adding face record failed.
deleteFaceRecordFailed	0x400010C2	Deleting face record failed.
modifyFaceRecordFailed	0x400010C3	Editing face record failed.
queryFaceRecordFailed	0x400010C4	Searching face record failed.
faceDetectFailed	0x400010C5	Detecting face failed.
libraryNotExist	0x400010C6	The library does not exist.
blackListQueryExporting	0x400010C7	Exporting matched blocklists.
blackListQueryExported	0x400010C8	The matched blocklists are exported.
blackListQueryStopExporting	0x400010C9	Exporting matched blocklists is stopped.
blackListAlarmQueryExporting	0x400010CA	Exporting matched blocklist alarms.
blackListAlarmQueryExported	0x400010CB	The matched blocklists alarms are exported.
blackListAlarmQueryStopExporting	0x400010CC	Exporting matched blocklist alarms is stopped.

Sub Status Code	Error Code	Description
getBigDataCloudAnalyisFailed	0x400010CD	Getting big data cloud analytic information failed.
setBigDataCloudAnalyisFailed	0x400010CE	Configuring big data cloud analytic failed.
submitMapSearchFailed	0x400010CF	Submitting search by picture task failed.
controlRelationshipNotExist	0x400010D0	The relation does not exist.
getHistoryAlarmInfoFailed	0x400010D1	Getting history alarm information failed.
getFlowReportFailed	0x400010D2	Getting people counting report failed.
addGuardFailed	0x400010D3	Adding arming configuration failed.
deleteGuardFailed	0x400010D4	Deleting arming configuration failed.
modifyGuardFailed	0x400010D5	Editing arming configuration failed.
queryGuardFailed	0x400010D6	Searching arming configurations failed.
uploadUserSuperCaps	0x400010D7	No more user information can be uploaded.
bigDataServerConnectFailed	0x400010D8	Connecting to big data server failed.
microVideoCloudRequestInfoBuildFailed	0x400010D9	Adding response information of micro video cloud failed.
microVideoCloudResponseInfoBuildFailed	0x400010DA	Parsing response information of micro video cloud failed.
transcodingServerRequestInfoBuildFailed	0x400010DB	Adding response information of transcoding server failed.
transcodingServerResponseInfoParseFailed	0x400010DC	Parsing response information of transcoding server failed.
transcodingServerOffline	0x400010DD	Transcoding server is offline.
microVideoCloudOffline	0x400010DE	Micro video cloud is offline.
UPSServerOffline	0x400010DF	UPS monitor server is offline.

Sub Status Code	Error Code	Description
statisticReportRequestInfoBuildFailed	0x400010E0	Adding response information of statistics report failed.
statisticReportResponseInfoParseFailed	0x400010E1	Parsing response information of statistics report failed.
DisplayConfigInfoBuildFailed	0x400010E2	Adding display configuration information failed.
DisplayConfigInfoParseFailed	0x400010E3	Parsing display configuration information failed.
DisplayConfigInfoSaveFailed	0x400010E4	Saving display configuration information failed.
notSupportDisplayConfigType	0x400010E5	The display configuration type is not supported.
passError	0x400010E7	Incorrect password.
upgradePackageLarge	0x400010EB	Too large upgrade package.
sessionUserReachesLimit	0x400010EC	No more user can log in via session.
ISO8601TimeFormatError	0x400010ED	Invalid ISO8601 time format.
clusterDissolutionFailed	0x400010EE	Deleting cluster failed.
getServiceNodeInfoFailed	0x400010EF	Getting service node information failed.
getUPSInfoFailed	0x400010F0	Getting UPS configuration information failed.
getDataStatisticsReportFailed	0x400010F1	Getting data statistic report failed.
getDisplayConfigInfoFailed	0x400010F2	Getting display configuration failed.
namingAnalysisBoardNotAllowed	0x400010F3	Renaming analysis board is not allowed.
onlyDrawRegionsOfConvexPolygon	0x400010F4	Only drawing convex polygon area is supported.
bigDataServerResponseInfoParseFailed	0x400010F5	Parsing response message of big data service failed.

Sub Status Code	Error Code	Description
bigDataServerReturnFailed	0x400010F6	No response is returned by big data service.
microVideoReturnFailed	0x400010F7	No response is returned by micro video cloud service.
transcodingServerReturnFailed	0x400010F8	No response is returned by transcoding service.
UPSServerReturnFailed	0x400010F9	No response is returned by UPS monitoring service.
forwardingServerReturnFailed	0x400010FA	No response is returned by forwarding service.
storageServerReturnFailed	0x400010FB	No response is returned by storage service.
cloudAnalysisServerReturnFailed	0x400010FC	No response is returned by cloud analytic service.
modelEmpty	0x400010FD	No model is obtained.
mainAndBackupNodeCannotModifyManagementNetworkInterfaceIP	0x400010FE	Editing the management interface IP address of main node and backup node is not allowed.
IDTooLong	0x400010FF	The ID is too long.
pictureCheckFailed	0x40001100	Detecting picture failed.
pictureModelingFailed	0x40001101	Modeling picture failed.
setCloudAnalysisDefaultProvinceFailed	0x40001102	Setting default province of cloud analytic service failed.
InspectionAreasNumberExceedLimit	0x40001103	No more detection regions can be added.
picturePixelsTooLarge	0x40001105	The picture resolution is too high.
picturePixelsTooSmall	0x40001106	The picture resolution is too low.
storageServiceIPEmpty	0x40001107	The storage server IP address is required.
bigDataServerRequestInfoBuildFail	0x40001108	Creating request message of big data service failed.
analysisTimedOut	0x40001109	Analysis time out.

Sub Status Code	Error Code	Description
high-performanceModeDisabled.	0x4000110A	Please enable high-performance mode.
configuringUPSMonitoringServerTimedOut	0x4000110B	Configuring the UPS monitoring server time out. Check IP address.
cloudAnalysisRequestInformationBuildFailed	0x4000110C	Creating request message of cloud analytic service failed.
cloudAnalysisResponseInformationParseFailed	0x4000110D	Parsing response message of cloud analytic service failed.
allCloudAnalysisInterfaceFailed	0x4000110E	Calling API for cloud analytic service failed.
cloudAnalysisModelCompareFailed	0x4000110F	Model comparison of cloud analytic service failed.
cloudAnalysisFacePictureQualityRatingFailed	0x40001110	Getting face quality grading of cloud analytic service failed.
cloudAnalysisExtractFeaturePointsFailed	0x40001111	Extracting feature of cloud analytic service failed.
cloudAnalysisExtractPropertyFailed	0x40001112	Extracting property of cloud analytic service failed.
getAddedNodeInformationFailed	0x40001113	Getting the added nodes information of data analysis server failed.
noMoreAnalysisUnitsAdded	0x40001114	No more data analysis servers can be added.
detectionAreaInvalid	0x40001115	Invalid detection region.
shieldAreaInvalid	0x40001116	Invalid shield region.
noMoreShieldAreasAdded	0x40001117	No more shield region can be drawn.
onlyAreaOfRectangleShapeAllowed	0x40001118	Only drawing rectangle is allowed in detection area.
numberReachedLimit	0x40001119	Number reached the limit.
wait1~3MinutesGetIPAfterSetupDHCP	0x4000111A	Wait 1 to 3 minutes to get IP address after configuring DHCP.

Sub Status Code	Error Code	Description
plannedTimeMustbeHalfAnHour	0x4000111B	Schedule must be half an hour.
oneDeviceCannotBuildCluster	0x4000111C	Creating main and backup cluster requires at least two devices.
updatePackageFileNotUploaded	0x4000111E	Upgrade package is not uploaded.
highPerformanceTasksNotSupportDrawingDetectionRegions	0x4000111F	Drawing detection area is not allowed under high-performance mode.
controlCenterIDDoesNotExist	0x40001120	The control center ID does not exist.
regionIDDoesNotExist	0x40001121	The area ID does not exist.
licensePlateFormatError	0x40001122	Invalid license plate format.
managementNodesNotSupportThisOperation	0x40001123	The operation is not supported.
searchByPictureResourceNotConfiged	0x40001124	The conditions for searching picture by picture are not configured.
videoFileEncapsulationFormatNotSupported	0x40001125	The video container format is not supported.
videoPackageFailure	0x40001126	Converting video container format failed.
videoCodingFormatNotSupported	0x40001127	Video coding format is not supported.
monitorOfDeviceArmingdeleteArmingInfo	0x40001129	The camera is armed. Disarm it and try again.
getVideoSourceTypeFailed	0x4000112A	Getting video source type failed.
smartRulesBuildFailed	0x4000112B	Creating VAC rule failed.
smartRulesParseFailed	0x4000112C	Parsing VAC rule failed.
timeRulesBuildFailed	0x4000112D	Creating time rule failed.
timeRulesParseFailed	0x4000112E	Parsing time rule failed.

Sub Status Code	Error Code	Description
monitoInfoInvalid	0x4000112F	Invalid camera information.
addingFailedVersionMismatches	0x40001130	Adding failed. The device version mismatches.
theInformationReturnedAfterCloudAnalysisIsEmpty	0x40001131	No response is returned by the cloud analytic service.
selectingIpAddressOfHostAndSpareNodeFailedCheckTheStatus	0x40001132	Setting IP address for main node and backup node failed. Check the node status.
theSearchIdDoesNotExist	0x40001133	The search ID does not exist.
theSynchronizationIdDoesNotExist	0x40001134	The synchronization ID does not exist.
theUserIdDoesNotExist	0x40001136	The user ID does not exist.
theIndexCodeDoesNotExist	0x40001138	The index code does not exist.
theControlCenterIdDoesNotExist	0x40001139	The control center ID does not exist.
theAreaIdDoesNotExist	0x4000113A	The area ID does not exist.
theArmingLinkageIdDoesNotExist	0x4000113C	The arming relationship ID does not exist.
theListLibraryIdDoesNotExist	0x4000113D	The list library ID does not exist.
invalidCityCode	0x4000113E	Invalid city code.
synchronizingThePasswordOfSpareServerFailed	0x4000113F	Synchronizing backup system password failed.
editingStreamingTypesNotSupported	0x40001140	Editing streaming type is not supported.
switchingScheduledTaskToTemporaryTaskIsNotSupported	0x40001141	Switching scheduled task to temporary task is not supported.

Sub Status Code	Error Code	Description
switchingTemporaryTaskToScheduledTasksNotSupported	0x40001142	Switching temporary task to scheduled task is not supported.
theTaskIsNotDispatchedOrItIsUpdating	0x40001143	The task is not dispatched or is updating.
thisTaskDoesNotExist	0x40001144	This task does not exist in the cloud analytic service.
duplicatedSchedule	0x40001145	Schedule period cannot be overlapped.
continuousScheduleWithSameAlgorithmTypeShouldBeMerged	0x40001146	The continuous schedule periods with same algorithm type should be merged.
invalidStreamingTimeRange	0x40001147	Invalid streaming time period.
invalidListLibraryType	0x40001148	Invalid list library type.
theNumberOfMatchedResultsShouldBeLargerThan0	0x40001149	The number of search results should be larger than 0.
invalidValueRangeOfSimilarity	0x4000114A	Invalid similarity range.
invalidSortingType	0x4000114B	Invalid sorting type.
noMoreListLibraryCanBeLinkedToTheDevice	0x4000114C	No more lists can be added to one device.
InvalidRecipientAddressFormat	0x4000114D	Invalid address format of result receiver.
creatingClusterFailedTheDongleIsNotPluggedIn	0x4000114E	Insert the dongle before creating cluster.
theURLIsTooLong	0x4000114F	No schedule configured for the task.
noScheduleIsConfiguredForTheTask	0x40001150	No schedule configured for the task.
theDongleIsExpired	0x40001151	Dongle has expired.
dongleException	0x40001152	Dongle exception.
invalidKey	0x40001153	Invalid authorization service key.

Sub Status Code	Error Code	Description
decryptionFailed	0x40001154	Decrypting authorization service failed.
encryptionFailed	0x40001155	Encrypting authorization service failed.
AuthorizeServiceResponseError	0x40001156	Authorization service response exception.
incorrectParameter	0x40001157	Authorization service parameters error.
operationFailed	0x40001158	Operating authorization service error.
noAnalysisResourceOrNoDataInTheListLibrary	0x40001159	No cloud analytic resources or no data in the list library.
calculationException	0x4000115A	Calculation exception.
allocatingList	0x4000115B	Allocating list.
thisOperationIsNotSupportedByTheCloudAnalytics	0x4000115C	This operation is not supported by the cloud analytic service.
theCloudAnalyticsIsInterrupted	0x4000115D	The operation of cloud analytic service is interrupted.
theServiceIsNotReady	0x4000115E	The service is not ready.
searchingForExternalApiFailed	0x4000115F	Searching external interfaces failed.
noOnlineNode	0x40001160	No node is online.
noNodeAllocated	0x40001161	No allocated node.
noMatchedList	0x40001162	No matched list.
allocatingFailedTooManyFacePictureLists	0x40001163	Allocation failed. Too many lists of big data service.
searchIsNotCompletedSearchAgain	0x40001164	Current searching is not completed. Search again.
allocatingListIsNotCompleted	0x40001165	Allocating list is not completed.
searchingForCloudAnalyticsResultsFailed	0x40001166	Searching cloud analytic service overtime.
noDataOfTheCurrentLibraryFound	0x40001167	No data in the current library. Make sure there is data in the Hbase.

Sub Status Code	Error Code	Description
noFacePictureLibraryIsArmed	0x40001168	No face picture library is armed for big data service.
noAvailableDataSlicingVersionInformationArmedFirstAndSliceTheData	0x40001169	Invalid standard version information.
duplicatedOperationDataSlicingIsExecuting	0x4000116A	Slicing failed. Duplicated operation.
slicingDataFailedNoArmedFacePictureLibrary	0x4000116B	Slicing failed. No arming information in the face big data.
GenerateBenchmarkFileFailedSlicingAgain	0x4000116C	Generating sliced file failed. Slice again.
NonprimaryNodesProhibitedFromSlicingData	0x4000116D	Slicing is not allowed by the backup node.
NoReadyNodeToClusterServers	0x4000116E	Creating the cluster failed. No ready node.
NodeManagementServicesOffline	0x4000116F	The node management server is offline.
theCamera(s)OfTheControlCenterAreAlreadyArmed.DisarmThemFirst	0x40001170	Some cameras in control center are already armed. Disarm them and try again.
theCamera(s)OfTheAreaAreAlreadyArmed.DisarmThemFirst	0x40001171	Some cameras in this area are already armed. Disarm them and try again.
configuringHigh-frequencyPeopleDetectionFailed	0x40001172	Configuring high frequency people detection failed.
searchingForHigh-frequencyPeopleDetectionLogsFailed.	0x40001173	Searching detection event logs of high-frequency people detection failed.
gettingDetailsOfSearchesHigh-frequencyPeopleDetectionLogsFailed.	0x40001174	Getting the search result details of frequently appeared person alarms failed.

Sub Status Code	Error Code	Description
theArmedCamerasAlreadyExistInTheControlCenter	0x40001175	Some cameras in control center are already armed.
disarmingFailedTheCamerasIsNotArmed	0x40001177	Disarming failed. The camera is not armed.
noDataReturned	0x40001178	No response is returned by the big data service.
preallocFailure	0x40001179	Pre-allocating algorithm resource failed.
overDogLimit	0x4000117A	Configuration failed. No more resources can be pre-allocated.
analysisServicesDoNotSupport	0x4000117B	Not supported.
commandAndDispatchServiceError	0x4000117C	Scheduling service of cloud analytic service error.
engineModuleError	0x4000117D	Engine module of cloud analytic service error.
streamingServiceError	0x4000117E	Streaming component of cloud analytic service error.
faceAnalysisModuleError	0x4000117F	Face analysis module of cloud analytic service error.
vehicleAnalysisModuleError	0x40001180	Vehicle pictures analytic module of cloud analytic service error.
videoStructuralAnalysisModuleError	0x40001181	Video structuring module of cloud analytic service error.
postprocessingModuleError	0x40001182	Post-processing module of cloud analytic service error.
frequentlyAppearedPersonAlarmsAlreadyConfiguredForListLibrary	0x40001183	Frequently appeared person alarm is already armed for blocklist library.
creatingListLibraryFailed	0x40001184	Creating list library failed.
invalidIdentityKeyOfListLibrary	0x40001185	Invalid identity key of list library.
noMoreDevicesCanBeArmed	0x40001186	No more camera can be added.

Sub Status Code	Error Code	Description
settingAlgorithmTypeForDeviceFailed	0x40001187	Allocating task resource failed.
gettingHighFrequencyPersonDetectionAlarmInformationFailed	0x40001188	Setting frequently appeared person alarm failed.
invalidSearchConfiton	0x40001189	Invalid result.
theTaskIsNotCompleted	0x4000118B	The task is not completed.
resourceOverRemainLimit	0x4000118C	No more resource can be pre-allocated.
frequentlyAppearedPersonAlarmsAlreadyConfiguredForTheCameraDisarmFirstAndTryAgain	0x4000118D	The frequently appeared person alarm of this camera is configured. Delete the arming information and try again.
switchtimedifflesslimit	0x4000123b	Time difference between power on and off should be less than 10 minutes.
associatedFaceLibNumOverLimit	0x40001279	Maximum number of linked face picture libraries reached.
noMorePeopleNumChangeRulesAdded	0x4000128A	Maximum number of people number changing rules reached.
noMoreViolentMotionRulesAdded	0x4000128D	Maximum number of violent motion rules reached.
noMoreLeavePositionRulesAdded	0x4000128E	Maximum number of leaving position rules reached.
SMRDiskNotSupportRaid	0x40001291	SMR disk does not support RAID.
OnlySupportHikAndCustomProtocol	0x400012A3	IPv6 camera can only be added via Device Network SDK or custom protocols.
vehicleEnginesNoResource	0x400012A6	Insufficient vehicle engine resources.
noMoreRunningRulesAdded	0x400012A9	Maximum number of running rules reached.

Sub Status Code	Error Code	Description
noMoreGroupRulesAdded	0x400012AA	Maximum number of people gathering rules reached.
noMoreFailDownRulesAdded	0x400012AB	Maximum number of people falling down rules reached.
noMorePlayCellphoneRulesAdded	0x400012AC	Maximum number of playing cellphone rules reached.
ruleEventTypeDuplicate	0x400012C8	Event type duplicated.
noMoreRetentionRulesAdded	0x400015AD	Maximum number of people retention rules reached.
noMoreSleepOnDutyRulesAdded	0x400015AE	Maximum number of sleeping on duty rules reached.
polygonNotAllowedCrossing	0x400015C2	Polygons are not allowed to cross.
configureRuleBeforeAdvanceParam	0x400015F8	Advanced parameters fail to be configured as no rule is configured, please configure rule information first.
behaviorCanNotPackToPic	0x40001603	The behavior model cannot be packaged as a picture algorithm.
noCluster	0x40001608	No cluster created.
NotAssociatedWithOwnChannel	0x400019C1	Current channel is not linked.
AITargetBPCaptureFail	0x400019C5	Capturing reference picture for AI target comparison failed.
AITargetBPToDSPFail	0x400019C6	Sending reference picture to DSP for AI target comparison failed.
AITargetBPDuplicateName	0x400019C7	Duplicated name of reference picture for AI target comparison.
audioFileNameWrong	0x400019D0	Incorrect audio file name.
audioFileImportFail	0x400019D1	Importing audio file failed.
NonOperationalStandbyMachine	0x400019F0	Non-operational hot spare.

Sub Status Code	Error Code	Description
MaximumNumberOfDevices	0x400019F1	The maximum number of devices reached.
StandbyMmachineCannotBeDeleted	0x400019F2	The hot spare cannot be deleted.
alreadyRunning	0x40002026	The application program is running.
notRunning	0x40002027	The application program is stopped.
packNotFound	0x40002028	The software packet does not exist.
alreadyExist	0x40002029	The application program already exists.
noMemory	0x4000202A	Insufficient memory.
invalidLicense	0x4000202B	Invalid License.
noClientCertificate	0x40002036	The client certificate is not installed.
noCACertificate	0x40002037	The CA certificate is not installed.
authenticationFailed	0x40002038	Authenticating certificate failed. Check the certificate.
clientCertificateExpired	0x40002039	The client certificate is expired.
clientCertificateRevocation	0x4000203A	The client certificate is revoked.
CACertificateExpired	0x4000203B	The CA certificate is expired.
CACertificateRevocation	0x4000203C	The CA certificate is revoked.
connectFail	0x4000203D	Connection failed.
loginNumExceedLimit	0x4000203F	No more user can log in.
HDMIResolutionIllegal	0x40002040	The HDMI video resolution cannot be larger than that of main and sub stream.
hdFormatFail	0x40002049	Formatting HDD failed.
formattingFailed	0x40002056	Formatting HDD failed.
encryptedFormattingFailed	0x40002057	Formatting encrypted HDD failed.
wrongPassword	0x40002058	Verifying password of SD card failed. Incorrect password.

Sub Status Code	Error Code	Description
audiolsPlayingPleaseWait	0x40002067	Audio is playing. Please wait.
twoWayAudioInProgressPleaseWait	0x40002068	Two-way audio in progress. Please wait.
calibrationPointNumFull	0x40002069	The maximum number of calibration points reached.
completeTheLevelCalibrationFirst	0x4000206A	The level calibration is not set.
completeTheRadarCameraCalibrationFirst	0x4000206B	The radar-camera calibration is not set.
pointsOnStraightLine	0x4000209C	Calibrating failed. The calibration points cannot be one the same line.
TValueLessThanOrEqualZero	0x4000209D	Calibration failed. The T value of the calibration points should be larger than 0.
HBDLibNumOverLimit	0x40002092	The number of human body picture libraries reaches the upper limit
theShieldRegionError	0x40002093	Saving failed. The shielded area should be the ground area where the shielded object is located.
theDetectionAreaError	0x40002094	Saving failed. The detection area should only cover the ground area.
invalidLaneLine	0x40002096	Saving failed. Invalid lane line.
enableITSFunctionOfThisChannelFirst	0x400020A2	Enable ITS function of this channel first.
noCloudStorageServer	0x400020C5	No cloud storage server
NotSupportWithVideoTask	0x400020F3	This function is not supported.
noDetectionArea	0x400050df	No detection area
armingFailed	0x40008000	Arming failed.
disarmingFailed	0x40008001	Disarming failed.
clearAlarmFailed	0x40008002	Clearing alarm failed.
bypassFailed	0x40008003	Bypass failed.

Sub Status Code	Error Code	Description
bypassRecoverFailed	0x40008004	Bypass recovery failed.
outputsOpenFailed	0x40008005	Opening relay failed.
outputsCloseFailed	0x40008006	Closing relay failed.
registerTimeOut	0x40008007	Registering timed out.
registerFailed	0x40008008	Registering failed.
addedByOtherHost	0x40008009	The peripheral is already added by other security control panel.
alreadyAdded	0x4000800A	The peripheral is already added.
armedStatus	0x4000800B	The partition is armed.
bypassStatus	0x4000800C	Bypassed.
zoneNotSupport	0x4000800D	This operation is not supported by the zone.
zoneFault	0x4000800E	The zone is in fault status.
pwdConflict	0x4000800F	Password conflicted.
audioTestEntryFailed	0x40008010	Enabling audio test mode failed.
audioTestRecoveryFailed	0x40008011	Disabling audio test mode failed.
addCardMode	0x40008012	Adding card mode.
searchMode	0x40008013	Search mode.
addRemoterMode	0x40008014	Adding keyfob mode.
registerMode	0x40008015	Registration mode.
exDevNotExist	0x40008016	The peripheral does not exist.
theNumberOfExDevLimited	0x40008017	No peripheral can be added.
sirenConfigFailed	0x40008018	Setting siren failed.
chanCannotRepeatedBinded	0x40008019	This channel is already linked by the zone.
inProgramMode	0x4000801B	The keypad is in programming mode.
inPaceTest	0x4000801C	In pacing mode.
arming	0x4000801D	Arming.

Sub Status Code	Error Code	Description
masterSlaveIsEnable	0x4000802c	The main-sub relationship has taken effect, the sub radar does not support this operation.
forceTrackNotEnabled	0x4000802d	Mandatory tracking is disabled.
isNotSupportZoneConfigByLocalArea	0x4000802e	This area does not support the zone type.
alarmLineCross	0x4000802f	Trigger lines are overlapped.
zoneDrawingOutOfRange	0x40008030	The drawn zone is out of detection range.
alarmLineDrawingOutOfRange	0x40008031	The drawn alarm trigger line is out of detection range.
hasTargetInWarningArea	0x40008032	The warning zone already contains targets. Whether to enable mandatory arming?
radarModuleConnectFail	0x40008033	Radar module communication failed.
importCfgFilePasswordErr	0x40008034	Incorrect password for importing configuration files.
overAudioFileNumLimit	0x40008038	The number of audio files exceeds the limit.
audioFileNameIsLong	0x40008039	The audio file name is too long.
audioFormatIsWrong	0x4000803a	The audio file format is invalid.
audioFileIsLarge	0x4000803b	The size of the audio file exceeds the limit.
pircamCapTimeOut	0x4000803c	Capturing of pircam timed out.
pircamCapFail	0x4000803d	Capturing of pircam failed.
pircamIsCaping	0x4000803e	The pircam is capturing.
audioFileHasExisted	0x4000803f	The audio file already exists.
subscribeTypeErr	0x4000a016	This metadata type is not supported to be subscribed.
EISError	0x4000A01C	Electronic image stabilization failed. The smart event function is enabled.
jpegPicWithAppendDataError	0x4000A01D	Capturing the thermal graphic failed. Check if the temperature measurement parameters

Sub Status Code	Error Code	Description
		(emissivity, distance, reflective temperature) are configured correctly.
startAppFail	/	Starting running application program failed.
yuvconflict	/	The raw video stream conflicted.
overMaxAppNum	/	No more application program can be uploaded.
noFlash	/	Insufficient flash.
platMismatch	/	The platform mismatches.
emptyEventName	0x400015E0	Event name is empty.
sameEventName	0x400015E1	A same event name already exists.
emptyEventType	0x400015E2	Event type is required.
sameEventType	0x400015E3	A same event type already exists.
maxEventNameReached	0x400015E4	Maximum of events reached.
hotSpareNotAllowedExternalStorage	0x400015FC	External storage is not allowed when hot spare is enabled.
sameCustomProtocolName	0x400015FD	A same protocol name already exists.
maxPTZTriggerChannelReached	0x400015FE	Maximum of channels linked with PTZ reached.
POSCannotAddHolidayPlan	0x400015FF	No POS events during holidays.
eventTypesTooLong	0x40001600	Event type is too long.
eventNamesTooLong	0x40001601	Event name is too long.
PerimeterEnginesNoResource	0x40001602	No more perimeter engines.
invalidProvinceCode	0x40001607	Invalid province code.

StatusCode=5

Sub Status Code	Error Code	Description
badXmlFormat	0x50000001	Invalid XML format.

StatusCode=6

Sub Status Code	Error Code	Description
badParameters	0x60000001	Invalid parameter.
badHostAddress	0x60000002	Invalid host IP address.
badXmlContent	0x60000003	Invalid XML content.
badIPv4Address	0x60000004	Invalid IPv4 address.
badIPv6Address	0x60000005	Invalid IPv6 address.
conflictIPv4Address	0x60000006	IPv4 address conflicted.
conflictIPv6Address	0x60000007	IPv6 address conflicted.
badDomainName	0x60000008	Invalid domain name.
connectSreverFail	0x60000009	Connecting to server failed.
conflictDomainName	0x6000000A	Domain name conflicted.
badPort	0x6000000B	Port number conflicted.
portError	0x6000000C	Port error.
exportErrorData	0x6000000D	Importing data failed.
badNetMask	0x6000000E	Invalid sub-net mask.
badVersion	0x6000000F	Version mismatches.
badDevType	0x60000010	Device type mismatches.
badLanguage	0x60000011	Language mismatches.
incorrentUserNameOrPasswor d	0x60000012	Incorrect user name or password.
invalidStoragePoolOfCloudServ er	0x60000013	Invalid storage pool. The storage pool is not configured or incorrect ID.
noFreeSpaceOfStoragePool	0x60000014	Storage pool is full.
riskPassword	0x60000015	Risky password.
UnSupportCapture	0x60000016	Capturing in 4096*2160 or 3072*2048 resolution is not supported when H.264+ is enabled.

Sub Status Code	Error Code	Description
userPwdLenUnder8	0x60000023	At least two kinds of characters, including digits, letters, and symbols, should be contained in the password.
userPwdNameSame	0x60000025	Duplicated password.
userPwdNameMirror	0x60000026	The password cannot be the reverse order of user name.
beyondARGSRangeLimit	0x60000027	The parameter value is out of limit.
DetectionLineOutOfDetectionRegion	0x60000085	The rule line is out of region.
DetectionRegionError	0x60000086	Rule region error. Make sure the rule region is convex polygon.
DetectionRegionOutOfCountingRegion	0x60000087	The rule region must be marked as red frame.
PedalAreaError	0x60000088	The pedal area must be in the rule region.
DetectionAreaABError	0x60000089	The detection region A and B must be in the a rule frame.
ABRegionCannotIntersect	0x6000008a	Region A and B cannot be overlapped.
customHBPIDError	0x6000008b	Incorrect ID of custom human body picture library
customHBPIDRepeat	0x6000008c	Duplicated ID of custom human body picture library
dataVersionsInHBDLibMismatches	0x6000008d	Database versions mismatches of human body picture library
invalidHBPID	0x6000008e	Invalid human body picture PID
invalidHBDID	0x6000008f	Invalid ID of human body picture library
humanLibraryError	0x60000090	Error of human body picture library

Sub Status Code	Error Code	Description
humanLibraryNumError	0x60000091	No more human body picture library can be added
humanImagesNumError	0x60000092	No more human body picture can be added
noHumanInThePicture	0x60000093	Modeling failed, no human body in the picture
analysisEnginesNoResourceError	0x60001000	No analysis engine.
analysisEnginesUsageExcced	0x60001001	The engine usage is overloaded.
PicAnalysisNoResourceError	0x60001002	No analysis engine provided for picture secondary recognition.
analysisEnginesLoadingError	0x60001003	Initializing analysis engine.
analysisEnginesAbnormaError	0x60001004	Analysis engine exception.
analysisEnginesFacelibImporting	0x60001005	Importing pictures to face picture library. Failed to edit analysis engine parameters.
analysisEnginesAssociatedChannel	0x60001006	The analysis engine is linked to channel.
smdEncodingNoResource	0x60001007	Insufficient motion detection encoding resources.
smdDecodingNoResource	0x60001008	Insufficient motion detection decoding resources.
diskError	0x60001009	HDD error.
diskFull	0x6000100a	HDD full.
facelibDataProcessing	0x6000100b	Handling face picture library data.
capturePackageFailed	0x6000100c	Capturing packet failed.
capturePackageProcessing	0x6000100d	Capturing packet.
noSupportWithPlaybackAbstract	0x6000100e	This function is not supported. Playback by video synopsis is enabled.

Sub Status Code	Error Code	Description
insufficientNetworkBandwidth	0x6000100f	Insufficient network bandwidth.
tapeLibNeedStopArchive	0x60001010	Stop the filing operation of tape library first.
identityKeyError	0x60001011	Incorrect interaction command.
identityKeyMissing	0x60001012	The interaction command is lost.
noSupportWithPersonDensityDetect	0x60001013	This function is not supported. The people density detection is enabled.
ipcResolutionOverflow	0x60001014	The configured resolution of network camera is invalid.
ipcBitrateOverflow	0x60001015	The configured bit rate of network camera is invalid.
tooGreatTimeDifference	0x60001016	Too large time difference between device and server.
noSupportWithPlayback	0x60001017	This function is not supported. Playback is enabled.
channelNoSupportWithSMD	0x60001018	This function is not supported. Motion detection is enabled.
channelNoSupportWithFD	0x60001019	This function is not supported. Face capture is enabled.
illegalPhoneNumber	0x6000101a	Invalid phone number.
illegalCertificateNumber	0x6000101b	Invalid certificate No.
linkedCameraOutLimit	0x6000101c	Connecting camera timed out.
achieveMaxChannelLimit	0x6000101e	No more channels are allowed.
humanMisInfoFilterEnabledChannelNumError	0x6000101f	No more channels are allowed to enable preventing false alarm.
humanEnginesNoResource	0x60001020	Insufficient human body analysis engine resources.
taskNumberOverflow	0x60001021	No more tasks can be added.

Sub Status Code	Error Code	Description
collisionTimeOverflow	0x60001022	No more comparison duration can be configured.
invalidTaskID	0x60001023	Invalid task ID.
eventNotSupport	0x60001024	Event subscription is not supported.
invalidEZVIZSecretKey	0x60001034	Invalid verification code for Hik-Connect.
needDoubleVerification	0x60001042	Double verification required
noDoubleVerificationUser	0x60001043	No double verification user
timeSpanNumOverLimit	0x60001044	Max. number of time buckets reached
channelNumOverLimit	0x60001045	Max. number of channels reached
noSearchIDResource	0x60001046	Insufficient searchID resources
noSupportDeleteStrangerLib	0x60001051	Deleting stranger library is not supported
noSupportCreateStrangerLib	0x60001052	Creating stranger library is not supported
behaviorAnalysisRuleInfoError	0x60001053	Behavior analysis rule parameters error.
safetyHelmetParamError	0x60001054	Hard hat parameters error.
OneChannelOnlyCanBindOneEngine	0x60001077	No more engines can be bound.
engineTypeMismatch	0x60001079	Engine type mismatched.
badUpgradePackage	0x6000107A	Invalid upgrade package.
AudioFileNameDuplicate	0x60001135	Duplicated audio file name.
CurrentAudioFileAIRuleInUseAlreadyDelete	0x60001136	The AI rule linkage related to current audio file has been deleted.
TransitionUseEmmc	0x60002000	Starting device failed. The EMMC is overused.

Sub Status Code	Error Code	Description
AdaptiveStreamNotEnabled	0x60002001	The stream self-adaptive function is not enabled.
AdaptiveStreamAndVariableBit rateEnabled	0x60002002	Stream self-adaptive and variable bitrate function cannot be enabled at the same time.
noSafetyHelmetRegion	0x60002023	The hard hat detection area is not configured (if users save their settings without configuring the arming area, they should be prompted to configure one).
unclosedSafetyHelmet	0x60002024	The hard hat detection is enabled (If users save their settings after deleting the arming area, they should be prompted to disable hard hat detection first and then delete the arming area).
width/heightRatioOfPictureError	0x6000202C	The width/height ratio of the uploaded picture should be in the range from 1:2 to 2:1.
PTZNotInitialized	0x6000202E	PTZ is not initialized.
PTZSelfChecking	0x6000202F	PTZ is self-checking.
PTZLocked	0x60002030	PTZ is locked.
advancedParametersError	0x60002031	Auto-switch interval in advanced parameters cannot be shorter than parking tolerance for illegal parking detection in speed dome rule settings.
resolutionError	0x60005003	Invalid resolution
deployExceedMax	0x60006018	The arming connections exceed the maximum number.
detectorTypeMismatch	0x60008000	The detector type mismatched.
nameExist	0x60008001	The name already exists.

Sub Status Code	Error Code	Description
uploadImageSizeError	0x60008016	The size of the uploaded picture is larger than 5 MB.
laneAndRegionOverlap	/	The lanes are overlapped.
unitConfigurationNotInEffect	/	Invalid unit parameter.
ruleAndShieldingMaskConflict	/	The line-rule region overlaps with the shielded area.
wholeRuleInShieldingMask	/	There are complete temperature measurement rules in the shielded area.
LogDiskNotSetReadOnlyInGroupMode	0x60001100	The log HDD in the HDD group cannot be set to read-only.
LogDiskNotSetRedundancyInGroupMode	0x60001101	The log HDD in the HDD group cannot be set to redundancy.
holidayNameContainChineseOrSpecialChar	0x60001080	No Chinese and special characters allowed in holiday name.
genderValueError	0x60001081	Invalid gender.
certificateTypeValueError	0x60001082	Invalid identification type.
personInfoExtendValuesTooLong	0x60001083	The length of customized tags exceeds limit.
personInfoExtendValueContainsInvalidChar	0x60001084	Invalid characters are not allowed in customized tags of the face picture library.
excelHeaderError	0x60001085	Excel header error.
intelligentTrafficMutexWithHighFrames	0x60008014	Please disable all functions of traffic incident detection, violation enforcement, and traffic data collection, or adjust the video frame rate to that lower than 50 fps.
intelligentTrafficMutexWithHighFramesEx	0x60008018	Please disable all functions of traffic incident detection, violation enforcement, traffic data collection, and vehicle

Sub Status Code	Error Code	Description
		detection, or adjust the video frame rate to that lower than 50 fps.

StatusCode=7

SubStatusCode	Error Code	Description
rebootRequired	0x70000001	Reboot to take effect.

A.6 Error Codes Categorized by Functional Modules

The error codes returned during the text protocol integration is categorized by different functional modules. See the error codes, error descriptions, and debugging suggestions in the table below.

Public Function Module (Error Codes Range: 0x00000000, from 0x00100001 to 0x001fffff)

Error String	Error Code	Description	Debugging Suggestion
success	0x00000000	Succeeded.	
deviceNotActivated	0x00100001	The device is not activated.	Activate the device.
deviceNoPermission	0x00100002	Device operation failed. No permission.	Update user's permission.
deviceNotSupport	0x00100003	This function is not supported.	Check the device capability set and call the API corresponding to supported function.
deviceResourceNotEnough	0x00100004	Insufficient resources.	Release resources.
dataFormatError	0x00100005	Invalid message format.	
resetError	0x00100006	Restoring to factory settings failed. Reactivating device is required after the device is	

Error String	Error Code	Description	Debugging Suggestion
		reboot as the Reset button may be stuck.	
parameterError	0x00100007	Incorrect parameter	
	0x00100100	Invalid channel	Check if the channel is valid.
	0x00100101	NPQ live view is not supported for stream encryption.	Replace streaming mode for stream encryption.
	0x00100102	No more channels are allowed for NPQ streaming.	Reduce NPQ streaming channels and try again.
	0x00100103	The stream type is not supported.	Check the requested stream type.
	0x00100104	The number of connections exceeded limit.	Reduce the number of streaming clients and try again.
	0x00100105	Not enough bandwidth.	Reduce the number of remote streaming channels.

User Function Module (Error Codes Range: from 0x00200001 to 0x002fffff)

Error String	Error Code	Description	Debugging Suggestion
passwordError	0x00200001	Incorrect user name or password.	Check if the password is correct.
userNameNotExist	0x00200002	The account does not exist.	Check if the account exists, or add the account.
userNameLocked	0x00200003	The account is locked.	Wait for the device to unlock.
userNumLimited	0x00200004	The number of users allowed to log in exceeded the upper limit.	Log out.
lowPrivilege	0x00200005	No permissions for this operation	For users operations, check the following situations: <ul style="list-style-type: none"> Deleting your own account is not allowed. Editing your own level or permission is not allowed.

Error String	Error Code	Description	Debugging Suggestion
			<ul style="list-style-type: none"> Getting information about users with higher permission is not allowed. Elevating the user's level or permission is not allowed. <p>For other operations, check according to the following measures: If operations unrelated to user's permission configuration failed, you can check the user type and permission, if not solved, contact the developers.</p>
incorrectUserNameOrPassword	0x00200006	Incorrect user name or password	Check if the configured user name and password are matched. If not, contact the administrator to configure again. If the administrator forgets the password, reset the password of the device.
riskPassword	0x00200007	Risk password	Low password strength. Change password again.
passwordMustContainMorethan8Characters	0x00200008	The password length must be greater than or equal to 8.	Check if the password length is greater than or equal to 8. If not, change password again.
passwordLenNoMoreThan16	0x00200009	The password length cannot be greater than 16.	Check if the password length is greater than 16. If yes, change password again.
adminUserNotAllowedModify	0x0020000a	Editing admin information is not allowed.	Check if the edited account is admin.
confirmPasswordError	0x0020000b	Incorrect confirm password.	Check the confirm password.
passwordMustContainMorethan2Types	0x0020000c	The password must contain at least two or more of followings: numbers, lowercase,	Check if the configured password conforms the requirements.

Error String	Error Code	Description	Debugging Suggestion
		uppercase, and special characters.	
passwordContainUserName	0x0020000d	The password cannot contain the user name.	Check if the password contains the user name.
userPwdNameMirror	0x0020000e	The password cannot be reversed user name.	Check if the password is reversed user name.

Time Function Module (Error Codes Range: from 0x00300001 to 0x003fffff)

Error String	Error Code	Description	Debugging Suggestion
manualAdjustmentFailed	0x00300001	Time synchronization failed.	
NTPError	0x00300002	Invalid NTP server address.	Check if the NTP server address is valid.
timeFormatError	0x00300003	Incorrect time format during time calibration. For example, the time in ISO 8601 format should be "2018-02-01T19:54:04", but the applied time is "2018-02-01 19:54:04".	Incorrect message format or incorrect time format.
beyondTimeRangeLimit	0x00300004	The calibration time is not within the time range supported by the device.	Get the device capability and check if the configured time is within the time range supported by the device.
endtimeEarlierThanBeginTime	0x00300005	The start time of the validity period cannot be later than the end time.	Check if the start time and end time are valid.

Network Function Module (Error Codes Range: from 0x00400001 to 0x004fffff)

Error String	Error Code	Description	Debugging Suggestion
domainNameParseFailed	0x00400001	Parsing domain name failed.	
PPPOEConnectedFailed	0x00400002	Connecting PPPOE to the network failed.	
FTPConnectedFailed	0x00400003	The FTP server is disconnected.	
deviceIPConflicted	0x00400004	IP addresses of devices conflicted.	
libraryConnectedFailed	0x00400005	The image and video library is disconnected.	
fileUploadFailed	0x00400006	Uploading failed.	Check if the network connection is normal. If yes, contact after-sales.
storSerDownloadFileFailed	0x00400007	Downloading failed.	Check if the network connection is normal. If yes, contact after-sales.
storSerDownloadFileSizeZero	0x00400008	The size of file downloaded from the storage service is 0.	Check if the network connection is normal. If yes, contact after-sales.
storSerNotConfig	0x00400009	Storage service is not configured.	Check if the configuration is correct.
badHostAddress	0x0040000a	Host address error	Check if the configuration is correct.
badIPv4Address	0x0040000b	Incorrect IPv4 address.	Check if the configuration is correct.
badIPv6Address	0x0040000c	Incorrect IPv6 address.	Check if the configuration is correct.
conflictIPv4Address	0x0040000d	IPv4 address conflict.	Check the configuration status of IPV4 in the network.
conflictIPv6Address	0x0040000e	IPv6 address conflict	Check the configuration status of IPV6 in the network.

Error String	Error Code	Description	Debugging Suggestion
badDomainName	0x0040000f	Incorrect domain name.	Check if the configuration is correct.
connectSreverFail	0x00400010	Connecting to server failed.	Check if the network is normal and check if the configuration is correct.
conflictDomainName	0x00400011	Domain name conflict.	Check if the configuration is correct.
badPort	0x00400012	Port conflict.	Check if the configuration is correct.
portError	0x00400013	Port error	Check if the configuration is correct.
badNetMask	0x00400014	Subnet mask error	Check if the configuration is correct.
badVersion	0x00400015	Version mismatch	Check if the version is correct.
badDns	0x00400016	DNS error	Check if the configuration is correct.
badMTU	0x00400017	MTU error	Check if the configuration is correct.
badGateway	0x00400018	Wrong gateway	Check if the configuration is correct.
urlDownloadFail	0x00400019	Downloading via URL failed.	Check if the network is normal and check if the URL is correct.
deployExceedMax	0x0040001a	The number of armed channels exceeds the maximum number of connections.	Get the supported maximum number of arming and the number of armed channels.

Maintenance Function Module (Error Codes Range: from 0x00500001 to 0x005fffff)

Error String	Error Code	Description	Debugging Suggestion
upgradeXMLFormatError	0x00500001	Incorrect XML upgrading request.	Check if the upgrade file is correct. If the file is correct, try the local upgrade.
upgradeContentError	0x00500002	Incorrect upgrading request content.	Check if the upgrade file is correct. If the file is correct, try the local upgrade.
noUpgradePermission	0x00500003	No upgrade permission.	Switch to admin account or ask admin for advanced operation permission.
upgrading	0x00500004	Upgrading...	Wait for the upgrade to complete.
receiveUpgradePackageError	0x00500005	Receiving upgrade package failed.	Check if the network is normal.
upgradePackageLanguageMismatch	0x00500006	Upgrade package language mismatch.	Check the language type of upgrade package and the device.
upgradePackageMismatch	0x00500007	Upgrade file does not match with the device type.	Check the type of upgrade package and device.
OEMCodeMismatch	0x00500008	Upgrade package error. The OEM code mismatch.	Contact after-sales to get the correct upgrade package.
versionMismatch	0x00500009	Upgrade file version mismatch.	Contact after-sales to get the correct upgrade package.
upgradeHalfFailed	0x0050000c	Error occurred in the halfway of device upgrading. Flash error or cache error.	
deviceParameterImportFailed	0x0050000d	Importing device parameters failed. Device model, version, or platform mismatches.	

Error String	Error Code	Description	Debugging Suggestion
deviceEncryptionError	0x0050000e	Upgrade package mismatches. Device encryption error.	
SDCardFormatError	0x00500025	Formatting SD card failed.	
SDCardLoadFailed	0x00500026	Loading page failed after the SD card is inserted.	
NASFailed	0x00500027	Mounting NAS failed.	
hardDiskError	0x00500028	HDD exception (possible reasons: HDD does not exist, incompatible, encrypted, insufficient capacity, formatting exception, array exception, array incompatible, etc.)	
upgradeError	0x00500030	Upgrade error	
upgradePackageSizeMismatch	0x00500032	Mismatch between the actual size of the downloaded upgrade package and the size in the upgrading request.	
upgradePackageSizeExceeded	0x00500033	The size of the package exceeded that of the partition.	
domainNameParseFailedForDownload	0x00500034	Parsing the domain name of the address for downloading failed.	
netWorkUnstable	0x00500035	Unstable network. Downloading timed out or the maximum number of attempts reached.	
digestValueMismatch	0x00500036	Mismatched digest value.	
signatureVerifyFailed	0x00500037	Verifying the signature failed.	

Error String	Error Code	Description	Debugging Suggestion
innerFormatError	0x00500038	Incorrect inner format of the upgrade package.	
memoryNotEnough	0x00500039	Insufficient memory.	
burnFailed	0x0050003a	Burning firmware failed.	
unknownError	0x0050003b	Unknown error occurred in the underlying APIs.	
userCancel	0x0050003c	User requested cancel of current operation.	
systemResume	0x0050003d	Upgrading failed. You can resume via the backup system or minimum system.	
	0x00500080	Upgrade file is not found.	Check if the upgrade package path is too long or if there is a correct upgrade package under the upgrade package path.
	0x00500081	Upgrade file does not match with the engine type.	Select the upgrade package matched with the device engine type.
	0x00500082	Parsing camera domain name failed.	Confirm if the device is correctly configured DNS service and if the camera domain is valid.
	0x00500083	Camera network is unreachable.	Confirm if the local network can access the network where the added channel located.

Live View Module (Error Codes Range: from 0x00600001 to 0x006fffff)

Error String	Error Code	Description	Debugging Suggestion
liveViewFailed	0x00600001	Live view failed. The number of streaming channels exceeded limit.	
	0x00600002	Request packaging format exception.	Check the packaging format of requested live view.
	0x00600003	NPQ will be unavailable after enabling EHome 2.x.	When EHome 2.x is enable, use other live view mode.
	0x00600005	NPQ live view is not supported for channel-zero.	User other live view mode for channel-zero.
	0x00600007	Only virtual stream supports NPQ live view.	Switch to virtual stream.
	0x0060000A	The IP channel is offline.	Check if the IP channel is online and try again.
	0x0060000B	Live view transcoding is not supported by the device.	Use other stream type for live view.
	0x0060000C	Channel-zero is not enabled.	Enable channel-zero before starting live view of channel-zero.
	0x0060000D	Transcoding capability exceeded limit.	Reduce camera resolution or the number of transcoding channels.
	0x00600010	The channel does not have sub-stream.	Use main stream mode for live view.
	0x00600011	NPQ live view is not supported by the device.	Switch to other live view mode.
	0x00600012	NPQ function is disabled.	Enable NPQ function or switch to other live view mode.

Playback Module (Error Codes Range: from 0x00700001 to 0x007fffff)

Error String	Error Code	Description	Debugging Suggestion
	0x00700001	Playback failed. Up to one channel's playback is supported.	
	0x00700002	The speed of playback displayed on video wall is not supported.	Reduce the playback speed.
	0x00700003	The transmission rate of playback stream is too high.	Reduce the transmission rate of playback stream.
	0x00700004	The encoding type of playback stream is not supported.	Provide the stream with encoding type supported by device.
	0x00700005	The container format of playback stream is not supported.	Provide the stream with container format supported by device.
	0x00700007	Exception occurred when decoding playback stream Possible reasons: displaying on video wall exception, image exception, display exception, decoding exception, image is stuck, black screen, invalid stream type, live view is stuck, audio decoding exception, and blurred screen.	
	0x00700008	Playback video does not exit, or searching failed.	Search again or check if HDD is normal.
	0x00700009	Playback time parameter error.	Check if the time period of searched video is correct and try again.
	0x0070000A	Invalid video type.	Select the correct video type to search.
	0x0070000B	Invalid time type.	Select the correct time type to search.

Error String	Error Code	Description	Debugging Suggestion
	0x0070000C	Invalid event parameter.	Select the correct event parameter to search.
	0x0070000D	Invalid event type.	Select the correct event type to search.
	0x0070000E	The device does not support smart search.	Select the non smart search mode to search.
	0x0070000F	Invalid smart event type.	Select the correct smart event type to search.
	0x00700010	Invalid dynamic analysis sensitivity.	Select the correct sensitivity to search video.
	0x00700011	Reverse playback is not supported.	Select the correct playback mode.
	0x00700012	Invalid file status.	Select the correct file status to search.
	0x00700013	Invalid searching start position.	Use the correct searching start position to search.
	0x00700014	Invalid maximum number of searching.	Use the correct maximum number of searching to search.

Capture Module (Error Codes Range: from 0x00800001 to 0x008fffff)

Error String	Error Code	Description	Debugging Suggestion
	0x00800001	Manual capture failed.	

Two-Way Audio Module (Error Codes Range: from 0x00900001 to 0x009fffff)

Error String	Error Code	Description	Debugging Suggestion
startFailed	0x00900001	Starting two-way audio failed. Audio loss or driver error.	
codingFormatNot Match	0x00900002	The encoding format of the intercom is inconsistent, and the negotiation fails	Check or capture the packets on the platform, then analyze if the audio

Error String	Error Code	Description	Debugging Suggestion
			encoding formats negotiated by both sides are consistent.
dialedIsBusy	0x00900003	The intercom party is already in the intercom and can no longer respond to the intercom	Check if the intercom party is already in the intercom, if not, get the protocol message and analyze the response message.
destinationLongNumberError	0x00900004	The requested destination long number is wrong	Check or capture the packets on the platform, then analyze the long number.

Video Storage Module (Error Codes Range: from 0x00a00001 to 0x00afffff)

Error String	Error Code	Description	Debugging Suggestion
videoSearchFailed	0x00a00001	Searching videos failed.	No resource stored in the device.
notFindStorageMedium	0x00a00002	No storage medium found.	
videoDownloadFailed	0x00a00003	Downloading videos failed.	

Picture Storage Module (Error Codes Range: from 0x00b00001 to 0x00bfffff)

Error String	Error Code	Description	Debugging Suggestion
	0x00b00001	Searching pictures failed.	No picture resource.

IO Function Module (Error Codes Range: from 0x00c00001 to 0x00cfffff)

Error String	Error Code	Description	Debugging Suggestion
	0x00c00001	Invalid alarm input No.	
	0x00c00002	Invalid alarm output No.	

Event Function Module (Error Codes Range: from 0x00d00001 to 0x00dfffff)

Error String	Error Code	Description	Debugging Suggestion
	0x00d00001	Incorrect event rule.	Refer to the manual for correct configuration.

Parking Service Module (Error Codes Range: from 0x00e00001 to 0x00efffff)

Error String	Error Code	Description	Debugging Suggestion
	0x00e00001	The vehicle with parking pass already exists.	Parking pass is created by license plate, you need to check if the parking pass for this license plate already created.
	0x00e00002	The license plate number is required.	

General Function Module (Error Codes Range: from 0x00f00001 to 0x00ffffff)

Error String	Error Code	Description	Debugging Suggestion
noMemory	0x00f00001	Insufficient device memory (heap space allocation failed).	Check the free memory and send logs to the developer for analysis.
deviceBusy	0x00f00002	The device is busy or the device is not responding.	Send logs to the developers for analysis. For fingerprint collection, face collection, file application, and file uploading services, check if the last operation is completed.
notSupport	0x00f00003	The URL is not supported by the device.	Capture the packets, check if the applied URL exists in the PMP platform. If yes, send the URL to the developer for analysis.

Error String	Error Code	Description	Debugging Suggestion
methodNotAllowed	0x00f00004	HTTP method is not allowed.	Capture the packets, check the method corresponding to the URL in the PMP platform.
invalidOperation	0x00f00005	Invalid operation of API command.	
IDNotExist	0x00f00006	The ID does not exist (the URL should contain ID, but the actual URL does not contain the ID).	Capture the packets and check if the ID included in the URL is correct.
invalidID	0x00f00007	Invalid ID (the ID in the URL exceeds the capability set or the ID format is invalid).	Capture the packets and check if the ID included in the URL is correct. Get the capabilities of URL and check the ID range.
invalidIURL	0x00f00008	The content after the "?" in the URL is wrong.	Capture the packets and check if the URL is correct.
deviceAckTimeout	0x00f00009	Device response timed out.	If the communication with the external module timed out, check if the external module is offline. When the above situation is eliminated, send logs to the developer for analysis.
badXmlFormat	0x00f0000a	XML format error	
badJsonFormat	0x00f0000b	JSON format error	
badURLFormat	0x00f0000c	URL format error	Get the URL and check if it is correct.
badXmlContent	0x00f0000d	XML message error:	

Error String	Error Code	Description	Debugging Suggestion
		<ul style="list-style-type: none"> The message contains only URL but no message body The required node is not configured. Node value exceeds the range limit (incorrect node value). 	
badJsonContent	0x00f0000e	JSON message error: <ul style="list-style-type: none"> The message contains only URL but no message body The required node is not configured. Node value exceeds the range limit (incorrect node value). 	
messageParametersLack	0x00f0000f	The required node does not exist.	
invalidSearchConditions	0x00f00010	Invalid search condition, search again.	Check if searchID is correct.
operObjectNotExist	0x00f00011	The object does not exist (for the operations about door, alarm IO, the object is not added).	Check if door lock is connected.

Door Control Module (Error Codes Range: from 0x01000001 to 0x010fffff)

Error String	Error Code	Description	Debugging Suggestion
multiAuthenticationFailed	0x01000001	Multi-factor authentication status operation failed.	
securityModuleOffline	0x01000002	The safety door control module is offline and fails to open the door.	Check if the safety door control is offline.

Schedule Template Module (Error Codes Range: from 0x01100001 to 0x011fffff)

Error String	Error Code	Description	Debugging Suggestion
planNumberConflict	0x01100001	Plan number conflict.	
timeOverlap	0x01100002	Time period conflict.	Check the message to find out if there is a time overlap of different time periods in one day.

Person Information Module (Error Codes Range: from 0x01200001 to 0x012fffff)

Error String	Error Code	Description	Debugging Suggestion

Certificate Module (Error Codes Range: from 0x01300001 to 0x013fffff)

Error String	Error Code	Description	Debugging Suggestion

Security Function Module (Error Codes Range: from 0x01400001 to 0x014fffff)

Error String	Error Code	Description	Debugging Suggestion
decryptFailed	0x01400001	Decryption failed, when decrypting sensitive information fields or importing data files.	The import secret key should be consistent with the export.
certificateNotmatch	0x01400003	Certificates mismatched, SSL/TLS public and private keys need to be matched in pairs.	The public and private keys need to be generated at the same time.
notActivated	0x01400004	Device is not activated.	Activate the device by tools such as SADP before use.
hasActivated	0x01400005	Device has been activated.	
forbiddenIP	0x01400006	IP address is banned	IP address is banned when illegal login attempts exceed the upper limit.

Error String	Error Code	Description	Debugging Suggestion
bondMacAddressNotMatch	0x01400007	The MAC address does not match the user.	Check if the specific MAC address has linked to the user.
bondIpAddressNotMatch	0x01400008	IP address does not match the user.	Check if the specific IP address has linked to the user.
badAuthorization	0x01400009	Triggered by illegal login	Incorrect password triggered the illegal login.

Advertising Function Module (Error Codes Range: from 0x01500001 to 0x015fffff)

Error String	Error Code	Description	Debugging Suggestion
materialDownloadFailed	0x01500001	Material download failed.	<ul style="list-style-type: none"> • Check if the network connection is normal. • Check if the device is running normally. • Check the log print.
materialNumberIsOver	0x01500002	The number of materials in the program list reached the upper limit.	Check if the number of materials in applied program list exceeded the limit.

A.7 Log Types for ISAPI

There are four major log types, i.e., alarm log, exception log, operation log, event log, and information log. And each major type contains multiple minor types, see details in the following contents.

Alarm Logs

Log Type	Description
shortCircuit	Short Circuit Alarm
dataPrealarm	Data warning
vehicleMonitor	Vehicle arming alarm
peopleCounting	People counting alarm

Log Type	Description
framesPeopleCounting	Regional people counting alarm
brokenCircuit	Open Circuit Alarm
alarmReset	Alarm Reset
alarmNormal	Return to Normal
passwordError	Incorrect Password (3 Tmes in a Row)
idCardIllegally	Invalid Card ID
keyPADRemove	Keypad Tampered
keyPADRemoveRestore	Keypad Restored
devRemove	Device Tampered
devRemoveRestore	Device Restored
belowAlarmLimit1	Sensor Value is Lower than Alarm Limit Value 1
belowAlarmLimit2	Sensor Value is Lower than Alarm Limit Value 2
belowAlarmLimit3	Sensor Value is Lower than Alarm Limit Value 3
belowAlarmLimit4	Sensor Value is Lower than Alarm Limit Value 4
aboveAlarmLimit1	Sensor Value is Higher than Alarm Limit Value 1
aboveAlarmLimit2	Sensor Value is Higher than Alarm Limit Value 2
aboveAlarmLimit3	Sensor Value is Higher than Alarm Limit Value 3
aboveAlarmLimit4	Sensor Value is Higher than Alarm Limit Value 4
UrgencyBtnON	Panic Button Triggered
UrgencyBtnOFF	Panic Button Restored
virtualDefenceBandit	Virtual Zone Burglary Alarm
virtualDefenceFire	Virtual Zone Fire Alarm
virtualDefenceUrgent	Virtual Zone Panic Alarm
motDetStart	Motion Detection Alarm Started
motDetStop	Motion Detection Alarm Stopped
hideAlarmStart	Device Blocked
hideAlarmStop	Device Blocking Alarm Restored
UPSAlarm	UPS Alarm

Log Type	Description
electricityMeterAlarm	Coulombmeter Alarm
switchPowerAlarm	Switch Power Supply Alarm
GasDetectSys	Gas Detection Alarm
transformerTempAlarm	Transformer Temperature Alarm
tempHumiAlarm	Temperature and Humidity Sensor Alarm
UPSAlarmRestore	UPS Alarm Restored
electricityMeterAlarmRestore	Coulombmeter Alarm Restored
switchPowerAlarmRestore	Switch Power Supply Alarm Restored
GasDetectSysRestore	Gas Detection Alarm Restored
transformerTempAlarmRestore	Transformer Temperature Alarm Restored
tempHumiAlarmRestore	Temperature-Humidity Sensor Alarm Restored
waterLevelSensorAlarm	Temperature-Humidity Sensor Alarm Restored
waterLevelSensorAlarmRestore	Flood Sensor Restored
dustNoiseAlarm	Dust and Noise Sensor Alarm
dustNoiseAlarmRestore	Dust and Noise Sensor Alarm Restored
environmentalLogger	Environmental Data Collector Alarm
environmentalLoggerAlarm	Environmental Data Collector Restored
triggerTemper	Detector Tampered
triggerTemperRestore	Detector Restored
emergencyCallHelp	Panic Alarm
emergencyCallHelpRestore	Panic Alarm Restored
consult	Consultation Alarm
consultRestore	Consultation Alarm Restored
deviceMoveAlarm	Device Motion Alarm
deviceMoveAlarmRestore	Device Motion Alarm Restored
earlyWarningAlarm	Early Warning Zone Alarm
earlyWarningAlarmRestore	Early Warning Zone Restored
warningAlarm	Warning Zone Alarm

Log Type	Description
warningAlarmRestore	Warning Zone Restored
wirelessOutputModTamperEvident	Wireless Output Expander Tampered
wirelessOutputModTamperEvidentReset	Wireless Output Expander tamper Restored
wirelessRepeaterTamperEvident	Wireless Repeater Tampered
wirelessRepeaterTamperEvidentReset	Wireless Repeater tamper Restored
wirelessSirenTamperEvident	Wireless Siren Tampered
wirelessSirenTamperEvidentReset	Wireless Siren Tamper Restored
wirelessKeypadTamperEvident	Wireless Keypad Tampered
wirelessKeypadTamperEvidentReset	Wireless Keypad Tamper Restored
wirelessCardReaderTamperEvident	Wireless Card Reader Tampered
wirelessCardReaderTamperEvidentReset	Wireless Card Reader Tamper Restored
softZoneMedicalAlarm	Virtual Zone Medical Alarm
accessControllerEvent	Access Controller Event
videoIntercomEvent	Video Intercom Event
GJDEvent	GJD Security Control Panel Event
LuminateEvent	LUMINITE Security Control Panel Event
OPTEXEvent	OPTEX Security Control Panel Event
cameraDetectorEvent	Detector Event
securityControlPanelEvent	Security Control Panel Event
RS-485AlarmInputModuleEvident	RS-485 Zone Module Tampered
RS-485AlarmInputModuleTamperReset	RS-485 Zone Module Tampering Reset
RS-485WirelessReceiverTamperEvident	RS-485 Wireless Receiver Module Tampered
RS-485WirelessReceiverTamperEvidentReset	RS-485 Wireless Receiver Module Tampering Reset
dredgerDetectionAlarm	Dredger Detection Alarm
crossLineAlarm	Line Crossing Alarm
crossLineAlarmRestore	Line Crossing Alarm Restored
HFPDAlarmStart	High Frequently Appeared Person Alarm Started

Log Type	Description
HFPDAlarmStop	High Frequently Appeared Person Alarm Stopped
LFPDAlarmStart	Low Frequency Person Alarm Started
LFPDAlarmStop	Low Frequency Person Alarm Stopped
safetyHelmetAlarmStart	Hard Hat Detection Alarm Started
safetyHelmetAlarmStop	Hard Hat Detection Alarm Stopped
dataPrealarm	Traffic Pre-alarm
playCellphoneStart	Playing Cellphone Alarm Started
playCellphoneStop	Playing Cellphone Alarm Stopped
sleepOnDutyStart	Sleeping on Duty Alarm Started
sleepOnDutyStop	Sleeping on Duty Alarm Stopped
vibrationDetectionStart	Vibration Detection Alarm Started
vibrationDetectionStop	Vibration Detection Alarm Stopped
fireEscapeDetectionStart	Fire Engine Access Detection Started
fireEscapeDetectionStop	Fire Engine Access Detection Ended
takingElevatorDetectionStart	Elevator Detection Started
takingElevatorDetectionStop	Elevator Detection Ended
unregisteredStreetVendorStart	Unlicensed Business Vendor Detection Started
unregisteredStreetVendorStop	Unlicensed Business Vendor Detection Ended
stallOutsideShopStart	Business Outside Store Detection Started
stallOutsideShopStop	Business Outside Store Detection Ended
stallOccupyingRoadStart	Business on Sidewalk Detection Started
stallOccupyingRoadStop	Business on Sidewalk Detection Ended
illegalHeapStart	Pile Goods or Materials in Chaos Detection Started
illegalHeapStop	Pile Goods or Materials in Chaos Detection Ended
illParkofNonMotorVehicleStart	Non-Motor Vehicles Parking in Chaos Detection Started
illParkofNonMotorVehicleStop	Non-Motor Vehicles Parking in Chaos Detection Ended
illegalOutdoorAdvertisementStart	Unauthorized Outdoor Advertisement Detection Started

Log Type	Description
illegalOutdoorAdvertisementStop	Unauthorized Outdoor Advertisement Detection Ended
packGarbageStart	Packaged Garbage Detection Started
packGarbageStop	Packaged Garbage Detection Ended
stallUnderUmbrellaStart	Running Business with Patio Umbrella Detection Started
stallUnderUmbrellaStop	Running Business with Patio Umbrella Detection Ended
dustbinOverflowStart	Overflowing Dustbin Detection Started
dustbinOverflowStop	Overflowing Dustbin Detection Ended
exposeGarbageStart	Exposed Garbage Detection Started
exposeGarbageStop	Exposed Garbage Detection Ended
hangClothingAlongStreetStart	Hanging Clothes On Street Detection Started
hangClothingAlongStreetStop	Hanging Clothes On Street Detection Ended
ATMPanelStart	ATM Panel Alarm Started
ATMPanelStop	ATM Panel Alarm Ended
ATMSurroundStart	ATM Surround Alarm Started
ATMSurroundStop	ATM Surround Alarm Ended
ATMFaceStart	ATM Face Alarm Started
ATMFaceStop	ATM Face Alarm Ended
ATMSafetyCabinStart	ATM Safety Cabin Alarm Started
ATMSafetyCabinStop	ATM Safety Cabin Alarm Ended
soundIntensityMutation	Sudden Increase of Sound Intensity Detection
soundIntensityMutationStop	Sudden Increase of Sound Intensity Detection Ended
soundIntensitySteepFall	Sudden Decrease of Sound Intensity Detection
soundIntensitySteepFallStop	Sudden Decrease of Sound Intensity Detection Ended
moveAlarm	Motion Alarm
moveAlarmRestored	Motion Alarm Restored
lowTemperatureAlarm	Low Temperature Alarm

Log Type	Description
lowTemperatureAlarmRestored	Low Temperature Alarm Restored
highTemperatureAlarm	High Temperature Alarm
highTemperatureAlarmRestored	High Temperature Alarm Restored
TemperatureIntervalMeasurementStart	Interval Temperature Measurement Started
TemperatureIntervalMeasurementStop	Interval Temperature Measurement Stopped
laneOccupationbyMotorVehicleStart	Lane Occupation by Motor Vehicle Detection Started
humanBodyStart	Mass Incident Detection Started
hangingOutdoorBannerStart	Hanging Banner Outdoors Detection Started
illegalAdvertisementStart	Illegal Advertisement Detection Started
illegalAdvertisementBannerStart	Illegal Advertisement Banner Detection Started
dirtyWaterbodyStart	Dirty Waters Detection Started
puttingTemporaryShelterIllegallyStart	Illegally Putting Temporary Shelter Detection Started
dirtyRoadStart	Dirty Road Detection Started
heapingWasteIllegallyStart	Illegally Heaping Waste Detection Started
stagnantRoadStart	Stagnant Road Detection Started
burningRubbishandLeavesStart	Burning Garbage and Leaves Detection Started
destroyingandOccupyingGreenBeltStart	Destroying and Occupying the Green Belt Detection Started
laneOccupationbyConstructionStart	Lane Occupation by Construction Detection Started
improperRubbishBinStart	Improper Dustbin Detection Started
abandonedFurnitureStart	Abandoned Furniture Detection Started
outsideAirConditionerHangingLowStart	Hanging the Air Conditioner Lowly Outside Detection Started
puttingInflatableArchIllegallyStart	Illegally Putting Inflatable Arch Detection Started
roadDamagedStart	Road Damaged Detection Started
constructionMaterialsMisplacedStart	Construction Materials Misplaced Detection Started
settingUpSlopeIllegallyStart	Illegally Setting Up Slope Detection Started
laneOccupationbyWasteRecyclingStart	Lane Occupation by Waste Recycling Detection Started

Log Type	Description
wasteWaterDischargedonRoadStart	Waste Water Discharged on Road Detection Started
barbecueStallinPublicSitesStart	Barbecue Stall in Public Sites Detection Started
litteringWasteonRoadStart	Littering Waste on Road Detection Started
distributingFlyersinPublicSitesStart	Distributing Flyers in Public Sites Detection Started
roadCollapsedStart	Road Collapsed Detection Started
constructionWasteMisplacedStart	Construction Water Misplaced Detection Started
greeningWasteStart	Greening Waste Detection Started
nonDecorativeHangingonTreeStart	Non-Decorative Tree Hangings Detection Started
installingWiresandPipesIllegallyStart	Illegally Installing Wires and Pipes Detection Started
raisingDomesticAnimalsStart	Raising Domestic Animals Detection Started
carcassNotClearedStart	Carcass Not Cleared Detection Started
streetSlaughterStart	Slaughtering Animals Along the Street Detection Started
vagrantsBegginginPublicSitesStart	Vagrant and Beggar in Public Sites Detection Started
dustbinExceptionStart	Dustbin Exception Detection Started
shortcutIsolationPileExceptionStart	Shortcut Isolation Pile Exception Detection Started
laneSeparatorDamagedStart	Lane Separator Damaged Detection Started
antiCollisionBarrelDamagedStart	Anti-Collision Barrel Damaged Detection Started
fireFightingFacilityDamagedStart	Fire Facility Damaged Detection Started
electricPowerFacilityExceptionStart	Power Facility Damaged Detection Started
transformerTankExceptionStart	Transformer Box Exception Detection Started
streetTreeExceptionStart	Street Tree Exception Detection Started
treeProtectionFacilityDamagedStart	Tree-Protection Facility Damaged Detection Started
manholeCoverExceptionStart	Manhole Cover Exception Detection Started
drainGratingDamagedStart	Drain Grating Damaged Detection Started
billboardDamagedStart	Billboard Damaged Detection Started
advertisingSignDamagedStart	Advertising Sign Damaged Detection Started
laneOccupationbyMotorVehicleStop	Lane Occupation by Motor Vehicle Detection Ended

Log Type	Description
humanBodyStop	Mass Incident Detection Ended
hangingOutdoorBannerStop	Hanging Banner Outdoors Detection Ended
illegalAdvertisementStop	Illegal Advertisement Detection Ended
illegalAdvertisementBannerStop	Illegal Advertisement Banner Detection Ended
dirtyWaterbodyStop	Dirty Waters Detection Ended
puttingTemporaryShelterIllegallyStop	Illegally Putting Temporary Shelter Detection Ended
dirtyRoadStop	Dirty Road Detection Ended
heapingWasteIllegallyStop	Illegally Heaping Waste Detection Ended
stagnantRoadStop	Stagnant Road Detection Ended
burningRubbishandLeavesStop	Burning Garbage and Leaves Detection Ended
destroyingandOccupyingGreenBeltStop	Destroying and Occupying the Green Belt Detection Ended
laneOccupationbyConstructionStop	Lane Occupation by Construction Detection Ended
improperRubbishBinStop	Improper Dustbin Detection Ended
abandonedFurnitureStop	Abandoned Furniture Detection Ended
outsideAirConditionerHangingLowStop	Hanging the Air Conditioner Lowly Outside Detection Ended
puttingInflatableArchIllegallyStop	Illegally Putting Inflatable Arch Detection Ended
roadDamagedStop	Road Damaged Detection Ended
constructionMaterialsMisplacedStop	Construction Materials Misplaced Detection Ended
settingUpSlopeIllegallyStop	Illegally Setting Up Slope Detection Ended
laneOccupationbyWasteRecyclingStop	Lane Occupation by Waste Recycling Detection Ended
wasteWaterDischargedonRoadStop	Waste Water Discharged on Road Detection Ended
barbecueStallinPublicSitesStop	Barbecue Stall in Public Sites Detection Ended
litteringWasteonRoadStop	Littering Waste on Road Detection Ended
distributingFlyersinPublicSitesStop	Distributing Flyers in Public Sites Detection Ended
roadCollapsedStop	Road Collapsed Detection Ended
constructionWasteMisplacedStop	Construction Waste Misplaced Detection Ended
greeningWasteStop	Greening Waste Detection Ended

Log Type	Description
nonDecorativeHangingonTreeStop	Non-Decorative Tree Hangings Detection Ended
installingWiresandPipesIllegallyStop	Illegally Installing Wires and Pipes Detection Ended
raisingDomesticAnimalsStop	Raising Domestic Animals Detection Ended
carcassNotClearedStop	Carcass Not Cleared Detection Ended
streetSlaughterStop	Slaughtering Animals Along the Street Detection Ended
vagrantsBegginginPublicSitesStop	Vagrant and Beggar in Public Sites Detection Ended
dustbinExceptionStop	Dustbin Exception Detection Ended
shortcutIsolationPileExceptionStop	Shortcut Isolation Pile Exception Detection Ended
laneSeparatorDamagedStop	Lane Separator Damaged Detection Ended
antiCollisionBarrelDamagedStop	Anti-Collision Barrel Damaged Detection Ended
fireFightingFacilityDamagedStop	Fire Facility Damaged Detection Ended
electricPowerFacilityExceptionStop	Power Facility Damaged Detection Ended
transformerTankExceptionStop	Transformer Box Exception Detection Ended
streetTreeExceptionStop	Street Tree Exception Detection Ended
treeProtectionFacilityDamagedStop	Tree-Protection Facility Damaged Detection Ended
manholeCoverExceptionStop	Manhole Cover Exception Detection Ended
drainGratingDamagedStop	Drain Grating Damaged Detection Ended
billboardDamagedStop	Billboard Damaged Detection Ended
advertisingSignDamagedStop	Advertising Sign Damaged Detection Ended
thermalCalibrationFileExceptionStart	Calibration file exception alarm started
thermalCalibrationFileExceptionStop	Calibration file exception alarm ended

Exception Logs

Log Type	Description
powerOn	Power on
powerOff	Power off

Log Type	Description
AIOPDetResolutionOverflow	Resolution of the detection stream in AI open platform exceeds the limit
HEOPDetResolutionOverflow	Resolution of the detection stream in Hikvision Embedded Open Platform exceeds the limit
WDTRreset	WDT Reset
lowBatteryVoltage	Low Battery Voltage
ACLoss	AC Power Disconnected
ACRestore	AC Power Restored
RTCException	RTC Real-time Clock Exception
netFailure	Network Disconnected
netRestore	Network Connected
telLineBroken	Telephone Line Disconnected
telLineRestore	Telephone Line Connected
expanderBusLoss	Bus Expander Disconnected
expanderBusRestore	Bus Expander Connected
keypadBusLoss	Keypad Expander Disconnected
keypadBusRestore	Keypad Expander Connected
sensorFailure	Analog Sensor Fault
sensorRestore	Analog Sensor Restored
RS485DisConnect	RS-485 Channel Disconnected
RS485Connect	RS-486 Channel Connected
batteryVoltageRestore	Battery Voltage Restored
wiredNetAbnormal	Wired Network Exception
wiredNetRestore	Wired Network Restored
GPRSAbnormal	GPRS Exception
GPRSRestore	GPRS Restored
3GAbnormal	3G Network Exception
3GRestore	3G Network Restored
SIMCardAbnormal	SIM Card Exception

Log Type	Description
SIMCardRestore	SIM Card Restored
VILost	Video Loss
illegalAccess	Illegal Login
HDFull	HDD Full
HDError	HDD Error
DCDLost	MODEM Disconnected
IPConflict	IP Address Conflicted
netbroken	Network Disconnected
recError	Recording Error
VIError	Video Input Exception(Only for Analog Channel)
formatHDDError	Remote HDD Formatting Failed
USBError	USB Communication Error
USBRestore	USB Communication Error Restored
printError	Printer Error
printRestore	Printer Error Restored
subsystemCommunicationError	Sub-board Communication Error
IPCIPconflict	Network Camera IP Address Conflicted
VIMisMatch	Video Standard Mismatches
MCURestart	MCU Restarted
GprsMouleFault	GPRS Module Fault
telephoneFault	Telephone Module Fault
wifiAbnormal	Wi-Fi Exception
wifiRestore	Wi-Fi Restored
RFAbornal	RF Exception
RFRestore	RF Restored
detectorOnline	Detector Connected
detectorOffline	Detector Disconnected
detectorBatteryNormal	Detector Battery Restored

Log Type	Description
detectorBatteryLow	Detector Battery Low
dataTrafficOverflow	Cellular Network Data Exceeded
radarSignalFault	Radar Transmitter Fault
radarSignalFaultRestore	Radar Transmitter Restored
wirelessOutputModOffline	Wireless Output Expander Disconnected
wirelessOutputModOnline	Wireless Output Expander Connected
wirelessRepeaterOffline	Wireless Repeater Disconnected
wirelessRepeaterOnline	Wireless Repeater Connected
triggerOffline	Trigger Disconnected
triggerOnline	Trigger Connected
wirelessSirenOffline	Wireless Siren Disconnected
wirelessSirenOnline	Wireless Siren Connected
sirenLowPower	Siren Battery Low
sirenPowerRecovery	Siren Battery Restored
ipcDisconnect	Network Camera Disconnected
ipcConnectRecovery	Network Camera Connected
sendMailFailed	Sending Email Failed
eventUploadException	Uploading Event Failed or Uploaded Event Lost
keyfobLowPower	Low Keyfob Battery
keyfobPowerRecovery	Normal Keyfob Battery
detectorOvertime	Detector Heartbeat Timed Out
detectorOvertimeRecovery	Detector Heartbeat Timeout Restored
wSirenOvertime	Wireless Siren Heartbeat Timed Out
wSirenOvertimeRecovery	Wireless Siren Heartbeat Timeout Restored
wOutputOvertime	Wireless Output Module Heartbeat Timed Out
wOutputOvertimeRecovery	Wireless Output Module Heartbeat Timeout Restored
wRepeaterOvertime	Wireless Repeater Heartbeat Timed Out

Log Type	Description
wRepeaterOvertimeRecovery	Wireless Repeater Heartbeat Timeout Restored
rfJamming	RF Wireless Communication Blocked
rfJammingRecovery	RF Wireless Communication Blocking Restored
batteryMiss	Storage Battery Loss
batteryMissRecovery	Storage Battery Restored
ARCUploadFailed	Uploading to ARC Failed
ARCUploadSucceeded	Uploaded to ARC
ARCUploadRecovery	Uploading to ARC Restored
wirelessKeypadOffline	Wireless Keypad Disconnected
wirelessKeypadOnline	Wireless Keypad Connected
wirelessCardReaderOffline	Wireless Card Reader Disconnected
wirelessCardReaderOnline	Wireless Card Reader Connected
keypadLowPower	Low Keypad Battery
keypadLowPowerRecovery	Low Keypad Battery Recovered
cardReaderLowPower	Low Card Reader Battery
cardReaderLowPowerRecovery	Low Card Reader Battery Recovered
wKeypadOvertime	Wireless Keypad Heartbeat Timed Out
wKeypadOvertimeRecovery	Wireless Keypad Heartbeat Timeout Recovered
wCardReaderOvertime	Wireless Card Reader Heartbeat Timed Out
wCardReaderOvertimeRecovery	Wireless Card Reader Heartbeat Timeout Recovered
ATSFailed	ATS Failed
ATSRecovery	ATS Recovered
LANPathFailed	Wired or Wireless Connection Failed
LANPathRecovery	Wired or Wireless Connection Recovered
mobileNetPathFailed	Mobile Network Connection Failed
mobileNetPathRecovery	Mobile Network Connection Recovered
RS-485AlarmInputModuleDisconnected	RS-485 Zone Module Offline
RS-485AlarmInputModuleConnected	RS-485 Zone Module Online

Log Type	Description
RS-485WirelessReceiverDisconnected	RS-485 Wireless Receiver Module Offline
RS-485WirelessReceiverConnected	RS-485 Wireless Receiver Module Online
keypadDisconnected	Keypad Offline
keypadConnected	Keypad Online
overvoltage	High Supply Voltage
undervoltage	Low Supply Voltage
highHDDTemperature	HDD High Temperature
lowHDDTemperature	HDD Low Temperature
hdImpact	HDD Impact
hdBadBlock	HDD Bad Sector
severeHDFailure	HDD Severe Fault
safetyHelmetException	Hard Hat Detection Exception
ezvizUpgradeException	Hik-Connect Upgrade Exception

Operation Logs

Log Type	Description
guard	Normal Arming
unguard	Normal Disarming
bypass	Bypass
duressAccess	Duress
localReboot	Local Reboot
remoteReboot	Remote Reboot
localUpgrade	Local Upgrade
remoteUpgrade	Remote Upgrade
recoveryDefaultParam	Restore Default Settings
outputAlarm	Remote Alarm Output Control
accessOpen	Access Control: Open
accessClose	Access Control : Closed

Log Type	Description
sirenOpen	Siren: On
sirenClose	Siren: Off
modZoneConfig	Zone Settings
modAlarmoutConfig	Alarm Output Settings
modAnalogConfig	Sensor Settings
RS485Config	RS-485 Channel Settings
phoneConfig	Dialing Settings
addAdmin	Added Administrator
modAdminParam	Edited Administrator
delAdmin	Deleted Administrator
addNetUser	Added DVR/NVR Operator
modNetUserParam	Edited DVR/NVR Operator
delNetUser	Deleted DVR/NVR Operator
addOperator	Added Camera Operator
modOperatorPw	Edited Camera Operator Password
delOperator	Deleted Camera Operator Password
addKeyPadUser	Added Keypad/Card Reader User
delKeyPadUser	Deleted Keyboard/Card Reader User
remoteUserLogin	Remote Login
remoteUserLogout	Remote Logout
remoteGuard	Remote Arming
remoteUnguard	Remote Disarming
modHostConfig	Edited Control Panel Settings
restoreBypass	Bypass Restored
alarmOutOpen	Turned on Output
alarmOutClose	Turned off Output
modSubsystemParam	Edited Subsystem Parameters
groupBypass	Group Bypass

Log Type	Description
groupBypassRestore	Group Bypass Restored
modGprsParam	Edited GPRS Parameters
modNetReportParam	Edited Network Report Settings
modReportMode	Edited Uploading Mode
modGatewayParam	Edited Access Control Settings
remoteStartRec	Remote: Started Recording
remoteStopRec	Remote: Stopped Recording
transChanStart	Transparent Transmission Started
transChanStop	Transparent Transmission Stopped
startVoiceTalk	Two-way Audio Started
stopVoiceTalk	Two-way Audio Terminated
remotePlayByFile	Remote: Playback or Downloaded by File
remotePlayByTime	Remote: Playback by Time
remotePTZCtrl	Remote: PTZ Control
remoteLockFile	Remote: Locked File
remoteUnlockFile	Remote: Unlocked File
remoteFormatHd	Remote: Formatted HDD
remoteDownloadCfgFile	Remote: Exported Configuration Files
remoteUploadCfgFile	Remote: Imported Configuration Files
remoteDownloadRecFile	Remote: Exported File
stayArm	Stay Arming
quickArm	Instant Arming
keyswitchArm	Key Zone Arming
keyswitckDisarm	Key Zone Disarming
clearAlarm	Alarm Cleared
modFaultConfig	Edited System Fault Settings
modAlarmOutConfig	Edited Event Alarm Output Settings
searchExternalModule	Searched for External Module

Log Type	Description
registerExternalModule	Re-registered External Module
closeKeypadAlarm	Disabled Keypad Beep
mod3GConfig	Edited Mobile Parameters
modPrintConfig	Edited Printer Parameters
SDCardFormat	Formatted SD Card
upgradeSubsystem	Upgraded Sub-board
planArmConfig	Arming/Disarming Schedule Configuration
phoneArm	SMS Arming
phoneStayArm	SMS Stay Arming
phoneQuickArm	SMS Instant Arming
phoneDisarm	SMS Disarming
phoneClearAlarm	SMS Alarm Cleared
whiteConfig	Allowlist Settings
timeTriggerConfig	Enabled/Disabled Trigger Configuration by Schedule
pictureConfig	Capture Settings
tamperConfig	Zone Tamper-Proof Settings
remoteKeypadUpgrade	Remote: Upgraded Keypad
singlePartionArmORDisarm	Single-Zone Arming/Disarming
cardConfiguration	Card Settings
cardAramORDisarm	Arming/Disarming by Card
expndNetCenterConfig	Extension Network Center Settings
netCardConfig	NIC Settings
DDNSConfig	DDNS Settings
RS485BusConfig	RS-485 Bus Settings
RS485BusReRegistration	RS-485 Bus Re-registration
remoteOpenElectricLock	Remote: Unlocked
remoteCloseElectricLock	Remote: Locked
localOpenElectricLock	Local: Unlocked

Log Type	Description
localCloseElectricLock	Local: Locked
openAlarmLamp	Remote: Turned On Alarm Lamp
closeAlarmLamp	Remote: Turned Off Strobe
temporaryPassword	Operation Record of Temporary Password
oneKeyAwayArm	One-Push Away Arming
oneKeyStayArm	One-Push Stay Arming
remoteDelAllVideoAnalysisTask	Empty All Video Analysis Tasks Remotely
singleZoneArm	Single-Zone Arming
singleZoneDisarm	Single-Zone Disarming
HiDDNSConfig	HiDDNS Settings
remoteKeypadUpdata	Remote: Upgraded Keypad
zoneAddDetector	Added Detector
zoneDelDetector	Deleted Detector
queryDetectorSignal	Checked Detector Signal Strength on Security Control Panel
queryDetectorBattery	Checked Detector Remaining Battery on Security Control Panel
setDetectorGuard	Detector Arming
setDetectorUnguard	Detector Disarming
setWifiParm	Wi-Fi Settings
voiceOpen	Audio On
voiceClose	Mute
functionKeyEnable	Enabled Function Key
functionKeyDisable	Disabled Panel Function Button
readCard	Swiped Patrol Card
localDeviceActive	Activated Device Remotely
localFactoryDefault	Restored Factory Settings Locally
remoteFactoryDefault	Restored Factory Settings Remotely
addWirelessOutputMod	Added Wireless Output Module

Log Type	Description
delWirelessOutputMod	Deleted Wireless Output Module
addWirelessRepeater	Added Wireless Repeater
delWirelessRepeater	Deleted Wireless Repeater
telListConfig	Mobile Phone Number Settings
searchRFSignal	Checked RF Signal
addWirelessSiren	Added Wireless Siren
delWirelessSiren	Deleted Wireless Siren
flowConfig	Cellular Data Limit Settings
addRemoter	Added Keyfob
delRemoter	Deleted Keyfob
addCard	Added Card
delCard	Deleted Card
remoteAddIpc	Added Network Camera
remoteDelIpc	Deleted Network Camera
remoteSetIpc	Edited Network Camera
localAddressFilterConfig/ remoteAddressFilterConfig	Local/Remote Address Filter Configuration
enterProgramMode	Programming Mode Enabled for Keypad
existProgramMode	Programming Mode Disabled for Keypad
localIOTCfgFileInput	Local operation: import IoT configuration file
localIOTCfgFileOutput	Local operation: export IoT configuration file
remoteIOTCfgFileInput	Remote operation: import IoT configuration file
remoteIOTCfgFileOutput	Remote operation: export IoT configuration file
localIOTAdd	Local operation: add IoT channel
remoteIOTAdd	Remote operation: add IoT channel
localIOTDelete	Local operation: delete IoT channel
remoteIOTDelete	Remote operation: delete IoT channel
localIOTSet	Local operation: configure IoT channel

Log Type	Description
remoteIoTSet	Remote operation: configure IoT channel
armWithFault	Armed with Fault
entryDelay	Entering and Exiting Delay
modArmConfig	Edit Arming Parameters
modCertificateStandard	Edit Authentication Standard
entryPaceTest	Pacing Mode Entered
exitPaceTest	Pacing Mode Exited
addNetOperator	Add Operator
modNetOperator	Edit Operator Information
delNetOperator	Delete Operator
addNetInstaller	Add Installer
modNetInstaller	Edit Installer Information
delNetInstaller	Delete Installer
addManufacturer	Add Manufacturer
modManufacturer	Edit Manufacturer Information
delManufacturer	Delete Manufacturer
upgradeSucceeded	Upgraded
upgradeFailed	Upgrading Failed
zoneDisabled	Zone Shielded
localCfgSecurity	Security Parameter Configured Locally
remoteCfgSecurity	Security Parameter Configured Remotely
remoteGetParaSecurity	Security Parameters Obtained Remotely
delRS-485InputModule	RS-485 Zone Module Deleted
delRS-485OutputModule	RS-485 Output Module Deleted
delRS-485WirelessReceiver	RS-485 Wireless Receiver Module Deleted
enrollRS-485InputModule	RS-485 Zone Module Registered
enrollRS-485OutputModule	RS-485 Output Module Registered
delRS-485OutputModule	RS-485 Output Module Deleted

Log Type	Description
enrollRS-485WirelessReceiver	RS-485 Wireless Receiver Module Registered
enrollKeypad	Keypad Registered
delKeypad	Keypad Deleted
scheduledAngleCalibration	Scheduled Angle Calibration
addZone	Added Zone
modZone	Edited Zone
delZone	Deleted Zone
addAlarmLine	Added Trigger Line
modAlarmLine	Edited Trigger Line
delAlarmLine	Deleted Trigger Line
remoteHFPDconfig/localHFPDconfig	Remote/Local Configuration of Frequently Appeared Person Detection
remoteLFPDconfig	Remote Configuration of Low Frequency Person Detection
modifyUserPassword	Password Changed
logOut	Logged Out
indicatorStatusSettings	Indicator Switch Settings
wiredNetworkSettings	Wired Network Settings
notificationSettings	Message Notification Settings
alarmCenterSettings	Alarm Receiving Center Settings
videoRecordStrategySettings	Video Recording Strategy Settings
cameraLinkageSettings	Camera Linkage Settings
armingDisarmingScheduleSettings	Arming and Disarming Schedule Settings
alarmSpeedSettings	Speed Threshold Settings for Triggering Alarms
videoTrackingSwitchSettings	Video Tracking Switching Settings
frequencySettings	Frequency Band Settings
masterSlaveTrackingSettings	Smart Linkage Settings
parkingPointSettings	Parking Point Settings
administratorEdited	Administrator Parameters Edited

Log Type	Description
securityConfigured	Security Settings
relayParametersEdited	Relay Parameters Edited
radarSensitivitySettings	Radar Sensitivity Settings
detectAngandDistanceSettings	Detector Angle and Range Settings
masterSlaveRadarSettings	Main Radar and Sub Radar Settings
remoteCheckTime	Remote: Manual Time Synchronization
remoteParamSimpleDefault	Remote: Partly Restore to Default Settings
remoteParamFactoryDefault	Remote: Restore All to Default Settings
localAutoSwitchConfig	Configure Auto Power On or Off Locally
remoteAutoSwitchConfig	Configure Auto Power On or Off Remotely
remoteCfgWirelessDialParam	Configure wireless dial-up parameters remotely
localCfgWirelessDialParam	Configure wireless dial-up parameters locally
remoteCfgWirelessSmsParam	Configure wireless message parameters remotely
localCfgWirelessSmsParam	Configure wireless message parameters locally
remoteCfgWirelessSmsSelfHelp	Configure SMS self-service parameters remotely
localCfgWirelessSmsSelfHelp	Configure SMS self-service parameters locally
remoteCfgWirelessNetFlowParam	Configure wireless traffic parameters remotely
localCfgWirelessNetFlowParam	Configure wireless traffic parameters locally
scaleCfg	Scale Settings
radarTrailCfg	Radar Pattern Settings
MapImportCfg	Map Importing Settings
radarCalibrationCfg	Radar Calibration Settings
LocalEzvizOperation	Local EZVIZ Operations
RemoteEzvizOperation	Remote EZVIZ Operations
SSHEnabled	SSH Enabled
SSHDisabled	SSH Disabled
installationModeEntered	Installation Mode Enabled
installationModeExited	Installation Mode Disabled

Log Type	Description
diagnosisModeConfigured	Diagnosis Mode Configured
fileExported	File Exported
audioFileUploaded	Audio File Uploaded
audioFileDeleted	Audio File Deleted
PIRCAMCapture	PIRCAM Captured
SIPIntercomStarted	SIP Intercom Started
SIPIntercomEnded	SIP Intercom Ended
enrollmentModeEntered	Registration Mode Enabled
enrollmentModeExited	Registration Mode Disabled
videoAudioSelfCheckStarted	Self-Test of Audio and Video Started
videoAudioSelfCheckStopped	Self-Test of Audio and Video Stopped
cardReaderunlocked	Card Reader Unlocked
cardReaderlocked	Card Reader Locked
videoAudioSelfCheckEnded	Self-Test of Audio and Video Ended
previewStart	Live View Started
previewStop	Live View Stopped
remoteDelAllVideoAnalysisTask	All Video Analysis Tasks Cleared (One-Touch)
localSSDOperateStart	Local SSD operation (firmware operations) started
localSSDOperateStop	Local SSD operation (firmware operations) ended
remoteSSDOperateStart	Remote SSD operation (firmware operations) started
remoteSSDOperateStop	Remote SSD operation (firmware operations) ended
AITargetBPAdd	Reference picture for AI target comparison added
AITargetBPDelete	Reference picture for AI target comparison deleted
AITargetBPSearch	Reference picture for AI target comparison AI searched for
AITargetBPUpdate	Reference picture for AI target comparison updated
AIRuleConfigTrigger	AI rule linkage configured
AudioFileImport	Audio file imported

Log Type	Description
AudioFileDownload	Audio file downloaded
cardNoNotRegistered	Card No. not registered
LocalBackupConfig	Local hot spare device configuration
RemoteBackupConfig	Remote hot spare device configuration
remoteAIModelAdd	Model package added
remoteAIModelQuery	Model package searched
remoteAIModelDelete	Model package deleted
remoteAIModelUpdate	Model package updated
remoteAIPicturePollingTaskAdd	Picture polling analysis task added
remoteAIPicturePollingTaskQuery	Picture polling analysis task searched
remoteAIPicturePollingTaskDelete	Picture polling analysis task deleted
remoteAIPicturePollingTaskModify	Picture polling analysis task edited
remoteAIVideoTaskAdd	Video analysis task added
remoteAIVideoTaskQuery	Video analysis task searched
remoteAIVideoTaskDelete	Video analysis task deleted
remoteAIVideoTaskModify	Video analysis task edited
remoteAIPictureTaskAdd	Picture analysis task added
remoteAIPictureTaskQuery	Picture analysis task searched
remoteAIPictureTaskDelete	Picture analysis task deleted
remoteAIPictureTaskModify	Picture analysis task edited
remoteAIVideoPollingTaskAdd	Video polling analysis task added
remoteAIVideoPollingTaskQuery	Video polling analysis task searched
remoteAIVideoPollingTaskDelete	Video polling analysis task deleted
remoteAIVideoPollingTaskModify	Video polling analysis task edited
AIRuleConfig	AI rule configuration
localParamFactoryDefault	Restore to default settings locally
remoteParamFactoryDefault	Restore to default settings remotely

Log Type	Description
remoteDeleteAllVerifyOrCapPics	Delete all authenticated or captured face pictures remotely
localDeleteAllVerifyOrCapPics	Delete all authenticated or captured face pictures locally
remoteDeleteEventsAtSpecTime	Delete events by specified time remotely
localDeleteEventsAtSpecTime	Delete events by specified time locally
remoteOpenSummerTime	Enable DST remotely
localOpenSummerTime	Enable DST locally
remoteCloseSummerTime	Disable DST remotely
localCloseSummerTime	Disable DST locally
remoteEZVIZUnbind	Unbind from EZVIZ cloud remotely
localEZVIZUnbind	Unbind from EZVIZ cloud locally
enterLocalUIBackground	Enter UI background
remoteDeleteFaceBaseMap	Delete registered face pictures remotely
localDeleteFaceBaseMap	Delete registered face pictures locally
SVCEnhanced	Enhance SVC
remoteImportEventTableFile	Remotely import the customized list of events
remoteExportEventTableFile	Remotely export the customized list of events
remoteEventTypeCfg	Remotely configure event type

Event Logs

Log Type	Description
SDKSchool	SDK Synchronization
presetsSatatusChange	Status of preset changed
selfTimeSchool	Time Synchronization by Schedule
insertSubsystem	Plugged in Sub-board
pullOutSubsystem	Pulled out Sub-board
autoArm	Auto Arming

Log Type	Description
autoDisarm	Auto Disarming
triggerOn	Activated Trigger by Schedule
triggerOff	Deactivated Trigger by Schedule
autoArmFailed	Auto Arming Failed
autoDisarmFailed	Auto Disarming Failed
triggerOnFailed	Activating Trigger Failed
triggerOffFailed	Deactivating Trigger Failed
mandatoryAlarm	Forced Arming
keyPADlocked	Keypad Locked
keyPADunlocked	Keypad Unlocked
insetUSB	Plugged in USB Flash Drive
pulloutUSB	Removed USB Flash Drive
lateRemind	Late to Disarm
keypadUnlocked	Unlocked Keypad
timeSynchronization	Time Synchronization
armFailed	Arming Failed
ARCStart	ARC Connected
locked	Locked

Information Logs

Log Type	Description
doubleVerificationPass	Double Verification Completed
hdFormatStart	Formatting HDD Started
hdFormatStop	Formatting HDD Stopped
wirelessRunningStatus	Wireless Network Running Status
BackupInfo	Hot Spare Information
addUserInfo	Added person information (access control permission)

Log Type	Description
modifyUserInfo	Edit person information (access control permission)
clearUserInfo	Delete person information by employee No. (access control permission)
clearAllUser	Delete all person information (access control permission)
ezvizLinkageInfo	EZVIZ linkage information

