



# **Device Network SDK (Behavior Analysis)**

**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

<b>Chapter 1 Overview .....</b>	<b>1</b>
1.1 Introduction .....	1
1.2 Update History .....	1
<b>Chapter 2 Alarm and Event Receiving .....</b>	<b>5</b>
2.1 Configure Specific Behavior Analysis Alarms for Intelligent Device .....	5
2.2 Intrusion Alarm .....	9
2.2.1 Configure Intrusion Alarm for Intelligent Device .....	9
2.2.2 Configure Intrusion Alarm for Smart Device .....	13
2.2.3 Configure Radar Intrusion Alarm .....	15
2.3 Line Crossing Alarm .....	18
2.3.1 Configure Line Crossing Alarm for Intelligent Device .....	18
2.3.2 Configure Line Crossing Alarm for Smart Device .....	22
2.3.3 Configure Radar Line Crossing Alarm .....	24
2.4 Region Entrance Alarm .....	27
2.4.1 Configure Region Entrance Alarm for Intelligent Device .....	27
2.4.2 Configure Region Entrance Alarm for Smart Device .....	30
2.5 Region Exiting Alarm .....	33
2.5.1 Configure Region Exiting Alarm for Intelligent Device .....	34
2.5.2 Configure Region Exiting Alarm for Smart Device .....	38
2.6 Configure Hard Hat Detection Alarm .....	42
2.7 Receive Alarm/Event in Arming Mode .....	46
2.8 Receive Alarm/Event in Listening Mode .....	49
<b>Chapter 3 Other Settings .....</b>	<b>53</b>
<b>Chapter 4 API Reference .....</b>	<b>54</b>
4.1 NET_DVR_Cleanup .....	54
4.2 NET_DVR_GetDeviceAbility .....	54

4.3 NET_DVR_GetDeviceConfig .....	55
4.4 NET_DVR_GetErrorMsg .....	56
4.5 NET_DVR_GetLastError .....	57
4.6 NET_DVR_GetSTDAbility .....	57
4.7 NET_DVR_GetSTDConfig .....	58
4.8 NET_DVR_Init .....	59
4.9 NET_DVR_Login_V40 .....	59
4.9.1 fLoginResultCallBack .....	60
4.10 NET_DVR_Logout .....	60
4.11 NET_DVR_SetDeviceConfig .....	61
4.12 NET_DVR_SetSDKInitCfg .....	62
4.13 NET_DVR_SetSTDConfig .....	64
4.14 NET_DVR_STDXMLConfig .....	64
4.15 NET_DVR_CloseAlarmChan_V30 .....	66
4.16 NET_DVR_GetDVRConfig .....	66
4.17 NET_DVR_SetDVRConfig .....	67
4.18 NET_DVR_SetDVRMessageCallBack_V50 .....	68
4.18.1 MSGCallBack .....	76
4.19 NET_DVR_SetupAlarmChan_V50 .....	77
4.20 NET_DVR_StartListen_V30 .....	77
4.21 NET_DVR_StopListen_V30 .....	78
<b>Appendix A. Data Structure .....</b>	<b>80</b>
A.1 NET_DVR_DEVICEINFO_V30 .....	80
A.2 NET_DVR_DEVICEINFO_V40 .....	84
A.3 NET_DVR_INIT_CFG_ABILITY .....	86
A.4 NET_DVR_LOCAL_SDK_PATH .....	87
A.5 NET_DVR_MIME_UNIT .....	88
A.6 NET_DVR_STD_CONFIG .....	88

A.7 NET_DVR_STD_ABILITY .....	89
A.8 NET_DVR_USER_LOGIN_INFO .....	90
A.9 NET_DVR_XML_CONFIG_INPUT .....	91
A.10 NET_DVR_XML_CONFIG_OUTPUT .....	92
A.11 NET_DVR_CHANNEL_GROUP .....	93
A.12 NET_DVR_CRUISECHAN_INFO .....	94
A.13 NET_DVR_EVENT_SCHEDULE .....	94
A.14 NET_DVR_EVENT_TRIGGER .....	95
A.15 NET_DVR_HANDLEEXCEPTION_V41 .....	95
A.16 NET_DVR_JPEGPARA .....	96
A.17 NET_DVR_PRESETCHAN_INFO .....	97
A.18 NET_DVR_PTZTRACKCHAN_INFO .....	97
A.19 NET_DVR_PTZ_POSITION .....	98
A.20 NET_DVR_PTZPOS .....	98
A.21 NET_DVR_REGION_ENTRANCE_DETECTION .....	99
A.22 NET_DVR_REGION_EXITING_DETECTION .....	100
A.23 NET_DVR_REGIONENTRANCE_REGION .....	101
A.24 NET_DVR_REGIONEXITING_REGION .....	102
A.25 NET_DVR_SCHEDTIME .....	103
A.26 NET_DVR_SMART_REGION_COND .....	103
A.27 NET_VCA_ADV_REACH_HEIGHT .....	104
A.28 NET_VCA_ADV_TRAVERSE_PLANE .....	105
A.29 NET_VCA_ANSWER .....	106
A.30 NET_VCA_APPEND_INFO .....	106
A.31 NET_VCA_AREA .....	107
A.32 NET_VCA_AUDIO_ABNORMAL .....	108
A.33 NET_VCA_BEHAVIOR_ABILITY .....	108
A.34 NET_VCA_BLACKBOARD_WRITE .....	109

A.35 NET_VCA_CHAN_IN_PARAM .....	110
A.36 NET_VCA_COMBINED_RULE .....	110
A.37 NET_VCA_DEV_ABILITY .....	111
A.38 NET_VCA_DEV_INFO .....	115
A.39 NET_VCA_DURATION .....	116
A.40 NET_VCA_EVENT_UNION .....	116
A.41 NET_VCA_FAKECARD .....	120
A.42 NET_VCA_FALL_DOWN .....	120
A.43 NET_VCA_FIELDDETECCION .....	121
A.44 NET_VCA_FILTER_STRATEGY .....	123
A.45 NET_VCA_GET_UP .....	123
A.46 NET_VCA_HIGH_DENSITY .....	124
A.47 NET_VCA_HIGH_DENSITY_STATUS .....	125
A.48 NET_VCA_HUMAN_ENTER .....	125
A.49 NET_VCA_INTRUSION .....	126
A.50 NET_VCA_LEAVE_POSITION .....	127
A.51 NET_VCA_LLECTURE .....	128
A.52 NET_VCA_LEFT .....	129
A.53 NET_VCA_LINE .....	129
A.54 NET_VCA_LOITER .....	130
A.55 NET_VCA_ONE_RULE_V42 .....	130
A.56 NET_VCA_OVER_TIME .....	132
A.57 NET_VCA_PARKING .....	133
A.58 NET_VCA_PEOPLENUM_CHANGE .....	133
A.59 NET_VCA_PLAY_CELLPHONE .....	134
A.60 NET_VCA_POINT .....	135
A.61 NET_VCA_POLYGON .....	135
A.62 NET_VCA_REACH_HIGHT .....	135

A.63 NET_VCA_RECT .....	136
A.64 NET_VCA_RELATE_RULE_PARAM .....	136
A.65 NET_VCA_RETENTION .....	137
A.66 NET_VCA_RULECFG_V42 .....	137
A.67 NET_VCA_RULE_ALARM .....	139
A.68 NET_VCA_RULE_INFO .....	141
A.69 NET_VCA_RULE_TRIGGER_PARAM .....	141
A.70 NET_VCA_RUN .....	142
A.71 NET_VCA_RUNNING .....	143
A.72 NET_VCA_SCANNER .....	144
A.73 NET_VCA_SITUATION_ANALYSIS .....	144
A.74 NET_VCA_SIT_QUIETLY .....	145
A.75 NET_VCA_SIZE_FILTER .....	145
A.76 NET_VCA_SPACING_CHANGE .....	146
A.77 NET_VCA_STANDUP .....	147
A.78 NET_VCA_STICK_UP .....	148
A.79 NET_VCA_TAKE .....	148
A.80 NET_VCA_TAKE_LEFT .....	149
A.81 NET_VCA_TARGET_INFO .....	150
A.82 NET_VCA_TOILET_TARRY .....	150
A.83 NET_VCA_TRAIL .....	151
A.84 NET_VCA_TRAVERSE_PLANE .....	151
A.85 NET_VCA_TRAVERSE_PLANE_DETECTION .....	152
A.86 NET_VCA_VIOLENT_MOTION .....	154
A.87 NET_VCA_YARD_TARRY .....	154
A.88 TRAFFIC_SCENE_MODE .....	155
A.89 TFS_CHAN_MODE_TYPE .....	155
A.90 VCA_ABILITY_TYPE .....	156

A.91 VCA_ABILITY_TYPE_EX .....	157
A.92 VCA_CHAN_ABILITY_TYPE .....	158
A.93 VCA_CHAN_MODE_TYPE .....	160
A.94 VCA_CROSS_DIRECTION .....	160
A.95 VCA_EVENT_TYPE .....	161
A.96 VCA_RULE_EVENT_TYPE_EX .....	163
A.97 BEHAVIOR_SCENE_MODE_TYPE .....	167
A.98 NET_ALARM_CVR_SUBINFO_UNION .....	167
A.99 NET_ALARM_RECORD_EXCEPTION .....	168
A.100 NET_ALARM_RECORDFILE_LOSS .....	168
A.101 NET_ALARM_RESOURCE_USAGE .....	169
A.102 NET_ALARM_STREAM_EXCEPTION .....	169
A.103 NET_DVR_ALARMER .....	170
A.104 NET_DVR_ALARMINFO_DEV .....	171
A.105 NET_DVR_ALARMINFO_DEV_V40 .....	171
A.106 NET_DVR_ALARMINFO_V30 .....	172
A.107 NET_DVR_ALARMINFO_V40 .....	173
A.108 NET_DVR_ALRAM_FIXED_HEADER .....	174
A.109 NET_DVR_ALARM_ISAPI_INFO .....	178
A.110 NET_DVR_ALARM_ISAPI_PICDATA .....	179
A.111 NET_DVR_ETHERNET_V30 .....	179
A.112 NET_DVR_IPADDR_UNION .....	180
A.113 NET_DVR_NETCFG_V50 .....	180
A.114 NET_DVR_PPPOECFG .....	182
A.115 NET_DVR_SETUPALARM_PARAM_V50 .....	182
A.116 NET_DVR_TIME .....	185
A.117 NET_DVR_TIME_EX .....	186
<b>Appendix B. Request URIs .....</b>	<b>187</b>



B.1 /ISAPI/Event/channels/<ID>/capabilities .....	189
B.2 /ISAPI/Event/channels/capabilities .....	189
B.3 /ISAPI/Event/schedules/radarFieldDetections .....	190
B.4 /ISAPI/Event/schedules/radarFieldDetections/<ID> .....	191
B.5 /ISAPI/Event/schedules/radarLineDetections .....	191
B.6 /ISAPI/Event/schedules/radarLineDetections/<ID> .....	192
B.7 /ISAPI/Event/triggers/<eventType>-<channelID> .....	193
B.8 /ISAPI/Event/triggersCap .....	194
B.9 /ISAPI/Intelligent/analysisEngines/<ID> .....	194
B.10 /ISAPI/Intelligent/analysisEngines/capabilities .....	195
B.11 /ISAPI/Intelligent/capabilities .....	195
B.12 /ISAPI/Intelligent/channels/<ID>/capabilities .....	196
B.13 /ISAPI/Intelligent/channels/<ID>/behaviorRule/<SID>/notifications .....	196
B.14 /ISAPI/Intelligent/channels/<ID>/behaviorRule/<SID>/rule/<ID> .....	197
B.15 /ISAPI/Intelligent/channels/<ID>/behaviorRule/<SID>/rule/capabilities .....	198
B.16 /ISAPI/Intelligent/channels/<ID>/behaviorRule/<SID>/schedules .....	199
B.17 /ISAPI/Intelligent/channels/<ID>/intelliTrace .....	200
B.18 /ISAPI/Intelligent/channels/<ID>/intelliTrace/<SID> .....	200
B.19 /ISAPI/Intelligent/channels/<ID>/intelliTrace/capabilities .....	201
B.20 /ISAPI/Intelligent/channels/<ID>/safetyHelmetDetection .....	202
B.21 /ISAPI/Intelligent/channels/<ID>/safetyHelmetDetection/advanceConfiguration .....	203
B.22 /ISAPI/Intelligent/channels/<ID>/safetyHelmetDetection/capabilities .....	203
B.23 /ISAPI/Intelligent/channels/<ID>/safetyHelmetDetection/search?format=json .....	204
B.24 /ISAPI/Intelligent/channels/<ID>/zoomRatio/<SID>/goto?format=json .....	205
B.25 /ISAPI/Intelligent/channels/<ID>/zoomRatio/capabilities .....	205
B.26 /ISAPI/Smart/capabilities .....	206
B.27 /ISAPI/Smart/RadarChannels/<ID>/radarFieldDetection/capabilities?format=json .....	206

B.28 /ISAPI/Smart/RadarChannels/<ID>/radarFieldDetection/regions/<regionID>?format=json .....	207
B.29 /ISAPI/Smart/RadarChannels/<ID>/radarFieldDetection/regions?format=json .....	207
B.30 /ISAPI/Smart/RadarChannels/<ID>/radarFieldDetection?format=json .....	208
B.31 /ISAPI/Smart/RadarChannels/<ID>/radarLineDetection/capabilities?format=json .....	209
B.32 /ISAPI/Smart/RadarChannels/<ID>/radarLineDetection/lineItems/<lineID>?format=json .....	209
B.33 /ISAPI/Smart/RadarChannels/<ID>/radarLineDetection/lineItems?format=json .....	210
B.34 /ISAPI/Smart/RadarChannels/<ID>/radarLineDetection?format=json .....	211
B.35 /ISAPI/System/capabilities .....	212
<b>Appendix C. Request and Response Messages .....</b>	<b>213</b>
C.1 JSON_EventNotificationAlert_Alarm/EventInfo .....	213
C.2 JSON_EventNotificationAlert_safetyHelmetDetection .....	213
C.3 JSON_LineItemInfo .....	218
C.4 JSON_LineItemList .....	218
C.5 JSON_RadarFieldDetection .....	219
C.6 JSON_RadarFieldDetectionCap .....	220
C.7 JSON_RadarLineDetection .....	221
C.8 JSON_RadarLineDetectionCap .....	222
C.9 JSON_RegionInfo .....	224
C.10 JSON_RegionList .....	225
C.11 JSON_ResponseStatus .....	225
C.12 JSON_SearchDescription .....	226
C.13 JSON_SearchResult .....	226
C.14 XML_AdvanceConfiguration .....	228
C.15 XML_AnalysisEngine .....	229
C.16 XML_AnalysisEnginesCap .....	230
C.17 XML_AttendedBaggageParam .....	233
C.18 XML_Cap_RegionEntrance .....	233

C.19 XML_Cap_RegionExiting .....	234
C.20 XML_Cap_RuleInfo .....	236
C.21 XML_Cap_Schedule .....	240
C.22 XML_ChannelEventCap .....	241
C.23 XML_ChannelEventCapList .....	246
C.24 XML_Desc_EventAbility .....	247
C.25 XML_Desc_VcaChanAbility .....	247
C.26 XML_DeviceCap .....	247
C.27 XML_EventAbility .....	259
C.28 XML_EventNotificationAlert_AlarmEventInfo .....	281
C.29 XML_EventTrigger .....	281
C.30 XML_EventTriggerNotification .....	282
C.31 XML_EventTriggerNotificationList .....	283
C.32 XML_EventTriggersCap .....	284
C.33 XML_EventTriggerCapType .....	287
C.34 XML_FieldDetectionParam .....	289
C.35 XML_GroupParam .....	290
C.36 XML_IntelliCap .....	290
C.37 XML_Intelli_IntelliTraceCap .....	294
C.38 XML_IntelliTraceBlock .....	295
C.39 XML_IntelliTraceBlockList .....	296
C.40 XML_LineDetectionParam .....	296
C.41 XML_LoiteringParam .....	296
C.42 XML_ParkingParam .....	297
C.43 XML_RadarFieldDetectionScheduleList .....	297
C.44 XML_RadarLineDetectionScheduleList .....	297
C.45 XML_RapidMoveParam .....	297
C.46 XML_RegionEntrance .....	298

C.47 XML_RegionEntranceRegion .....	298
C.48 XML_RegionEntranceRegionList .....	299
C.49 XML_RegionEntranceParam .....	299
C.50 XML_RegionExitingParam .....	300
C.51 XML_ResponseStatus .....	300
C.52 XML_RuleInfo .....	301
C.53 XML_RuleNotification .....	303
C.54 XML_RuleSchedule .....	304
C.55 XML_RunningParam .....	305
C.56 XML_SafetyHelmetDetection .....	305
C.57 XML_SafetyHelmetDetectionCap .....	307
C.58 XML_Schedule .....	309
C.59 XML_SmartCap .....	311
C.60 XML_SubscribeEvent .....	312
C.61 XML_TeacherParam .....	313
C.62 XML_UnattendedBaggageParam .....	313
C.63 XML_VcaChanAbility .....	313
C.64 XML_VideoCap .....	341
C.65 XML_ZoomRatioCap .....	342
<b>Appendix D. Appendixes .....</b>	<b>344</b>
D.1 Device Network SDK Errors .....	344
D.2 Response Codes of Text Protocol .....	387

# Chapter 1 Overview

This manual provides the integration methods and flows based on Device Network SDK (here is referred to as "HCNetSDK") for behavior analysis applications.

## 1.1 Introduction

Behavior analysis functions identify the objects and confirm if the objects behavior is qualified based on the rules. It can trigger alarm and provide useful information in the quickest and best way once detecting unqualified objects. For different behavior types, the detection rules are different, this manual mainly introduces the programming process of some typical types, such as intrusion, line crossing, region entrance, region exiting, and so on.

## 1.2 Update History

### Summary of Changes in Version 6.1.4.45\_July, 2020

1. Extended the intrusion detection parameter structure **NET\_VCA\_INTRUSION** :  
added two members, i.e., **byAlarmConfidence** (confidence of alarm notification) and **byRecordConfidence** (confidence of recording video) via two reserved bytes.
2. Extended the structure about line crossing detection parameters **NET\_VCA\_TRAVERSE\_PLANE** :  
added two members, i.e., **byAlarmConfidence** (confidence of alarm notification) and **byRecordConfidence** (confidence of recording video) via two reserved bytes.
3. Extended the structure about the parameters of region entrance detection **NET\_DVR\_REGIONENTRANCE\_REGION** :  
added two members, i.e., **byAlarmConfidence** (confidence of alarm notification) and **byRecordConfidence** (confidence of recording video) via two reserved bytes.
4. Extended the structure about the parameters of region exiting detection **NET\_DVR\_REGIONEXITING\_REGION** :  
added two members, i.e., **byAlarmConfidence** (confidence of alarm notification) and **byRecordConfidence** (confidence of recording video) via two reserved bytes.

### Summary of Changes in Version 6.1.4.40\_June, 2020

1. Extended the intelligent channel capability **XML\_VcaChanAbility** (related API: **NET\_DVR\_GetDeviceAbility** , capability type: DEVICE\_ABILITY\_INFO):  
added one sub node <FakeCard> (fake card detection) to <EventType> of <Behavior>;  
added one event type "fakeCard" (fake card detection) to <eventType> of <RuleEntry>;

added two sub nodes **<fallDownMaxRuleNum>** (maximum number of rules for people falling down detection) and **<violentMotionMaxRuleNum>** (maximum number of rules for violent motion) to **<RuleEntry>** of **<BehaviorRule>**.

2. Extended the alarm rule parameters structure **NET\_VCA\_EVENT\_UNION** :  
added one member **struFakeCard** (fake card detection parameters: **NET\_VCA\_FAKECARD** ).
3. Extended the enumeration about behavior analysis types **VCA\_ABILITY\_TYPE\_EX** :  
added one behavior analysis type **EVENT\_FAKECARD** (fake card detection).
4. Extended the enumeration about behavior analysis alarm/event types **VCA\_RULE\_EVENT\_TYPE\_EX** :  
added one behavior analysis alarm/event type **ENUM\_VCA\_EVENT\_FAKECARD** (fake card detection).

### Summary of Changes in Version 6.1.4.10\_Feb., 2020

Extended the message about auto-tracking capability **XML\_Intelli\_IntelliTraceCap** (related API: **NET\_DVR\_STDXMLConfig** , related URI: **/ISAPI/Intelligent/channels/<ID>/intelliTrace/capabilities** ) and parameters **XML\_IntelliTraceBlock** (related API: **NET\_DVR\_STDXMLConfig** , related URI: **/ISAPI/Intelligent/channels/<ID>/intelliTrace** ):  
added one sub node **<sceneRatio>** (scene ratio) to node **<IntelliTraceBlock>**.

### Summary of Changes in Version 6.1.3.45\_Jan., 2020

1. Added an API calling flow of configuring intrusion alarm for radar PTZ camera, refer to **Configure Radar Intrusion Alarm** for details.
2. Added an API calling flow of configuring line crossing alarm for radar PTZ camera, refer to **Configure Radar Line Crossing Alarm**
3. Extended the additional information structure of behavior analysis **NET\_VCA\_APPEND\_INFO** :  
added four members, i.e., **dwTargetSpeed** (Target speed detected by radar), **dwTargetDistance** (Target distance detected by radar), **byAlarmType** (alarm trigger source type), and **byRadarChannel** (Radar channel No.), via 74 reserved bytes.
4. Extended the device capability **XML\_DeviceCap** (related API: **NET\_DVR\_STDXMLConfig** ; related URI: **/ISAPI/System/capabilities** ):  
added two nodes, i.e., **<isSupportRadarFieldDetection>** (whether it supports radar intrusion detection) and **<isSupportRadarLineDetection>** (whether it supports radar line crossing detection).
5. Extended the channel event capability **XML\_ChannelEventCap** (related API: **NET\_DVR\_STDXMLConfig** ; related URI: **/ISAPI/Event/channels/<ID>/capabilities** ):  
added two values to the node **<eventType>**, i.e., "radarFieldDetection" (radar intrusion detection) and "radarLineDetection" (radar line crossing detection).
6. Extended the linkage capabilities of different alarm categories **XML\_EventTriggersCap** (related API: **NET\_DVR\_STDXMLConfig** ; related URI: **/ISAPI/Event/triggersCap** ):  
added two nodes, i.e., **<RadarLineDetectionTriggerCap>** (alarm linkage capability of radar line crossing detection) and **<RadarFieldDetectionTriggerCap>** (alarm linkage capability of radar intrusion detection).

### Summary of Changes in Version 6.1.3.30\_Jan., 2020

1. Extended capability message of alarm linkage action types **XML\_EventTriggerCapType** (related API: **NET\_DVR\_GetSTDAbility**):  
added a node <isSupportFaceContrast> (whether it supports face picture comparison linkage).
2. Extended capability message of hard hat detection **XML\_SafetyHelmetDetectionCap** (related API: **NET\_DVR\_STDXMLConfig**, related URI: **/ISAPI/Intelligent/channels/<ID>/safetyHelmetDetection/capabilities**) and message about advanced parameters of hard hat detection **XML\_AdvanceConfiguration** (related API: **NET\_DVR\_STDXMLConfig**, related URI: **/ISAPI/Intelligent/channels/<ID>/safetyHelmetDetection/advanceConfiguration**):  
added two sub nodes <FDLibList> (channel's linked face picture library list) and <FaceScore> (face picture scoring configuration) to the node <AdvanceConfiguration>.
3. Extended message about alarm details of hard hat detection **JSON\_EventNotificationAlert\_safetyHelmetDetection**:  
added two sub nodes "Face" (target information) and "faceContrastResult" (face picture comparison result) to the node "Target".
4. Extended message about hard hat detection search result **JSON\_SearchResult** (related API: **NET\_DVR\_STDXMLConfig**, related URI: **/ISAPI/Intelligent/channels/<ID>/safetyHelmetDetection/search?format=json**):  
added one sub node "face" (face picture information) to the node "Target" of "targets".
5. Extended the structure about alarm rule parameters **NET\_VCA\_EVENT\_UNION** (related API: **NET\_DVR\_SetupAlarmChan\_V50**):  
added one member **struPlayCellphone** (playing mobile phone detection parameters).

### Summary of Changes in Version 6.1.0.25\_July, 2019

1. Extended the configuration capability of zooming ratio in behavior analysis scene **XML\_ZoomRatioCap** (related API: **NET\_DVR\_STDXMLConfig**; related URL: **/ISAPI/Intelligent/channels/<ID>/zoomRatio/capabilities**):  
added one node: <isSupportZoomRatioGoto> (whether supports optimizing zooming ratio in behavior analysis scene).
2. Added a URL for optimizing zooming ratio in behavior analysis scene (related API: **NET\_DVR\_STDXMLConfig**): PUT **/ISAPI/Intelligent/channels/<ID>/zoomRatio/<SID>/goto?format=json**.

### Summary of Changes in Version 6.1.0.15\_Aug., 2019

1. Added the function of configuring hard hat detection, refer to **Configure Hard Hat Detection Alarm**.
2. Extended error codes in **Device Network SDK Errors**:  
added two error codes: 3501-"NET\_DVR\_ERR\_NO\_SAFETY\_HELMET\_REGION" (The hard hat detection area is not configured) and 3502-"NET\_DVR\_ERR\_UNCLOSED\_SAFETY\_HELMET" (The hard hat detection is enabled).
3. Added two sub status codes to status code 6 (Invalid Message Content) in **Response Codes of Text Protocol**:

0x60002023-"noSafetyHelmetRegion" (The hard hat detection area is not configured) and  
0x60002024-"unclosedSafetyHelmet" (The hard hat detection is enabled).

### Summary of Changes in Version 6.0.2.40\_May, 2019

1. Extended the alarm linkage parameters **XML\_RuleNotification** (related API: **NET\_DVR\_STDXMLConfig** ; related URL: **/ISAPI/Intelligent/channels/<ID>/behaviorRule/<SID>/notifications** ):  
added one linkage action "audio" to node <notificationMethod>.
2. Extended the capability of setting behavior analysis rules by detection scene ID  
**XML\_Cap\_RuleInfo** (related API **NET\_DVR\_STDXMLConfig** ; request URL: **/ISAPI/Intelligent/channels/<ID>/behaviorRule/<SID>/rule/capabilities** ):  
added one sub node <ObjectSizeList> (the default values of width and height for size filter) to node <SizeFilter>.

### Summary of Changes in Version 6.0.2.30\_03/2019

1. Extended the structure of smart alarm/event rule information **NET\_VCA\_RULE\_INFO** via one reserved type:  
added one parameter **bySceneID** (scene ID).
2. Extended the intelligent channel capability **XML\_VcaChanAbility** (related API: **NET\_DVR\_GetDeviceAbility** , capability type: DEVICE\_ABILITY\_INFO):  
added one node <uploadSceneID> (whether to uploads scene ID of smart event/alarm).

### Summary of Changes in Version 6.0.2.5\_01/2019

1. Extended the configuration structures of line crossing, intrusion, region entrance, and region exiting detection, i.e., **NET\_VCA\_TRAVERSE\_PLANE** , **NET\_VCA\_INTRUSION** , **NET\_DVR\_REGIONEXITING\_REGION** , and **NET\_DVR\_REGIONENTRANCE\_REGION** :  
added a value to the detection target type (**byDetectionTarget**): 0x04 (other).
2. Extended alarm details structures of line crossing, intrusion, region entrance, and region exiting detection, i.e., **NET\_VCA\_TRAVERSE\_PLANE** , **NET\_VCA\_INTRUSION** , and **NET\_VCA\_AREA** :  
added a value to the detection target type (**byDetectionTarget**): 0x04 (other).
3. Extended device event capability ( **XML\_EventAbility** , related API: **NET\_DVR\_GetDeviceAbility** , capability type: DEVICE\_ABILITY\_INFO):  
added a value to the detection target type (**detectionTarget**): "others".

### Summary of Changes in Version 5.3.6.151\_08/2018

Related Product Model: DS-2TD21xx/V1 series, DS-2TD21xx/VP series, DS-2TD26xx series, DS-2TD26xx/V1 series, DS-2TD28xx series, DS-2TD28xx/V1 series, DS-2TD12xx/V1 series

1. Extended the behavior analysis rule structure **NET\_VCA\_ONE\_RULE\_V42** (related API: **NET\_DVR\_GetDeviceConfig** and **NET\_DVR\_SetDeviceConfig** ):  
added one parameter **byBackgroundSuppression** (background suppression).
2. New this document.



## Chapter 2 Alarm and Event Receiving

The alarm/event information from the device can be received in third-party platform or system when the alarms are triggered or event occurred. Two modes are available for receiving alarms, including arming mode and listening mode.

### Arming Mode

The third-party platform connects to device automatically, when the alarm is triggered, the platform sends alarm uploading command to the device, and then the device will upload the alarm to the platform.

### Listening Mode

When alarm is triggered, the device automatically uploads the alarm, and then the third-party platform receives the uploaded alarm via the configured listening host (listening address and port should be configured). This mode is applicable for multiple devices uploading alarm/event information to one third-party platform without logging in to devices, and the restart of devices will not affect the alarm/event uploading. But a device can only support the configuration of one or two listening addresses and ports.

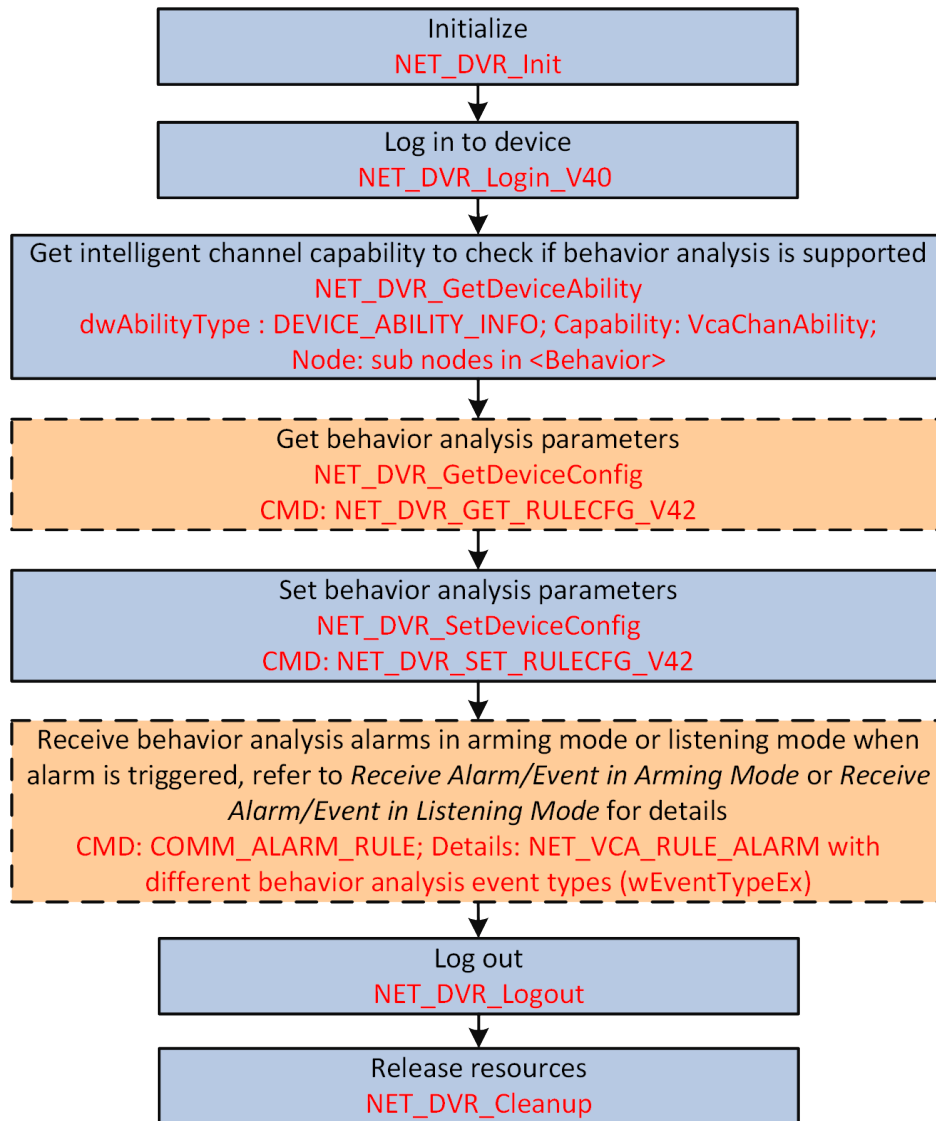
## 2.1 Configure Specific Behavior Analysis Alarms for Intelligent Device

This chapter provides a common configuration method for some specific behavior analysis alarms, such as intrusion detection, line crossing detection, region entrance detection, region exiting detection, loitering detection, and people gathering detection, of intelligent device.

### Before You Start

- Make sure you have called **NET DVR Init** to initialize the development environment.
- Make sure you have called **NET DVR Login\_V40** to log in to the device.

## Steps



**Figure 2-1 Programming Flow of Configuring Specific Behavior Analysis Alarms for Intelligent Device**

1. Call **NET\_DVR\_GetDeviceAbility**, set the capability type (**dwAbilityType**) to "DEVICE\_ABILITY\_INFO" (value: 0x011), and set the input buffer (**pInBuf**) to the message **XML\_Desc\_VcaChanAbility** for getting the intelligent device capability to check if some specific types of behavior analysis are supported.

The intelligent device capability is returned in the message **XML\_VcaChanAbility** by the output parameter **pOutBuf**.

If the nodes of different behavior analysis types, e.g., **<Intrusion>**, **<TraversePlane>**, **<EnterArea>**, **<ExitArea>**, etc., exists in the message, it indicates that the

corresponding behavior analysis type is supported by device, and then you can continue to perform the following steps.

Otherwise, the behavior analysis type is not supported, please end the task.

2. **Optional:** Call **NET\_DVR\_GetDeviceAbility**, set the capability type (**dwAbilityType**) to "VCA\_CHAN\_ABILITY" (value: 0x110), and set the input buffer (**pInBuf**) to the structure **NET\_VCA\_CHAN\_IN\_PARAM** for getting the intelligent channel capability.

The intelligent channel capability is returned in the structure **NET\_VCA\_BEHAVIOR\_ABILITY** by the output parameter **pOutBuf**.

3. **Optional:** Call **NET\_DVR\_GetDeviceConfig** with **NET\_DVR\_GET\_RULECFG\_V42**

(command No.: 5049), and set the input parameter pointer **lpInBuffer** to

**NET\_DVR\_CHANNEL\_GROUP** for getting default or configured behavior analysis alarm parameters (e.g., rules, sensitivity, threshold, arming schedule, linkage actions, etc.) for reference.

The behavior analysis alarm parameters are returned in the structure **NET\_VCA\_RULECFG\_V42** by the output parameter pointer **lpOutBuffer**.

4. Call **NET\_DVR\_SetDeviceConfig** with command 5050-"NET\_DVR\_SET\_RULECFG\_V42", set the input parameter pointer **lpInBuffer** to **NET\_DVR\_CHANNEL\_GROUP**, and set the input parameter **lpInParamBuffer** to **NET\_VCA\_RULECFG\_V42** for setting behavior analysis alarm parameters.



### Note

- To receive alarm in the platform, the linkage action must be set to "center" (upload to center).
- The above behavior analysis alarm parameters can also be configured by logging in to device via web browser.

5. **Optional:** Receive behavior analysis alarms in arming mode (see **Receive Alarm/Event in Arming Mode**) or listening mode (see **Receive Alarm/Event in Listening Mode**) when alarm is triggered.



### Note

- The command (**lCommand**) to receive behavior analysis alarms should be set to **COMM\_ALARM\_RULE** (command No.: 0x1102) in the APIs of **NET\_DVR\_SetDVRMessageCallBack\_V50** and **NET\_DVR\_StartListen\_V30**.
- For the alarm details, refer to the structure **NET\_VCA\_RULE\_ALARM**.

---

## Example

### Sample Code of Receiving Behavior Analysis Alarm in Arming Mode

```
#include <stdio.h>
#include <iostream>
#include "Windows.h"
#include "HCNetSDK.h"
using namespace std;
```

```
//Macro definition of time resolution
#define GET_YEAR(_time_)      (((_time_)>>26) + 2000)
#define GET_MONTH(_time_)    (((_time_)>>22) & 15)
#define GET_DAY(_time_)      (((_time_)>>17) & 31)
#define GET_HOUR(_time_)     (((_time_)>>12) & 31)
#define GET_MINUTE(_time_)   (((_time_)>>6) & 63)
#define GET_SECOND(_time_)   (((_time_)>>0) & 63)

BOOL CALLBACK MessageCallback(LONG lCommand, NET_DVR_ALARMER *pAlarmer, char
*pAlarmInfo, DWORD dwBufLen, void* pUser)
{
    switch(lCommand)
    {
        case COMM_ALARM_RULE: //Behavior analysis alarm information
        {
            NET_VCA_RULE_ALARM struVcaAlarm = {0};
            memcpy(&struVcaAlarm, pAlarmInfo, sizeof(NET_VCA_RULE_ALARM));

            NET_DVR_TIME struAbsTime = {0};
            struAbsTime.dwYear = GET_YEAR(struVcaAlarm.dwAbsTime);
            struAbsTime.dwMonth = GET_MONTH(struVcaAlarm.dwAbsTime);
            struAbsTime.dwDay = GET_DAY(struVcaAlarm.dwAbsTime);
            struAbsTime.dwHour = GET_HOUR(struVcaAlarm.dwAbsTime);
            struAbsTime.dwMinute = GET_MINUTE(struVcaAlarm.dwAbsTime);
            struAbsTime.dwSecond = GET_SECOND(struVcaAlarm.dwAbsTime);

            //Save captured scene picture
            if (struVcaAlarm.dwPicDataLen > 0 && struVcaAlarm.pImage != NULL)
            {
                char cFilename[256] = {0};
                HANDLE hFile;
                DWORD dwReturn;

                char chTime[128];
                sprintf(chTime,"%4.4d%2.2d%2.2d%2.2d%2.2d
%2.2d",struAbsTime.dwYear, struAbsTime.dwMonth, struAbsTime.dwDay,
                struAbsTime.dwHour, struAbsTime.dwMinute, struAbsTime.dwSecond);

                sprintf(cFilename, "VcaAlarmPic[%s]
[%s].jpg",struVcaAlarm.struDevInfo.struDevIP.sIPv4, chTime);

                hFile = CreateFile(cFilename, GENERIC_WRITE, FILE_SHARE_READ,
NULL, CREATE_ALWAYS, FILE_ATTRIBUTE_NORMAL, NULL);
                if (hFile == INVALID_HANDLE_VALUE)
                {
                    break;
                }
                WriteFile(hFile, struVcaAlarm.pImage,
struVcaAlarm.dwPicDataLen, &dwReturn, NULL);
                CloseHandle(hFile);
                hFile = INVALID_HANDLE_VALUE;
            }
        }
    }
}
```

```
    }

    WORD wEventType = struVcaAlarm.struRuleInfo.wEventTypeEx;

    printf("\n\n");
    printf("Behavior Analysis Alarm [0x%x]: Abs[%4.4d%2.2d%2.2d%2.2d
%2.2d%2.2d] Dev[ip:%s,port:%d,ivmsChan:%d] Smart[%d] EventTypeEx[%d]\n", \
        lCommand, struAbsTime.dwYear, struAbsTime.dwMonth,
struAbsTime.dwDay, struAbsTime.dwHour, struAbsTime.dwMinute, \
        struAbsTime.dwSecond, struVcaAlarm.struDevInfo.struDevIP.sIPv4,
struVcaAlarm.struDevInfo.wPort, \
        struVcaAlarm.struDevInfo.byIvmsChannel, struVcaAlarm.bySmart,
wEventType);

    NET_VCA_TARGET_INFO tmpTargetInfo;
    memcpy(&tmpTargetInfo, &struVcaAlarm.struTargetInfo,
sizeof(NET_VCA_TARGET_INFO));
    printf("Target Information: ID[%d]RECT[%f][%f][%f][%f]
\n",
        tmpTargetInfo.dwID, tmpTargetInfo.struRect.fX,
tmpTargetInfo.struRect.fY,
        tmpTargetInfo.struRect.fWidth, tmpTargetInfo.struRect.fHeight);
    break;
}
default:
{
    printf("Other alarms, alarm information type: 0x%x\n", lCommand);
    break;
}
}
return TRUE;
}
```

### What to do next

Call [NET\\_DVR\\_Logout](#) and [NET\\_DVR\\_Cleanup](#) to log out and release resources.

## 2.2 Intrusion Alarm

Intrusion detection function detects people, vehicle, or other objects which enter and loiter in a predefined virtual region, and some certain actions can be taken when the alarm is triggered. And for intelligent device and smart device, the APIs or commands called to configure intrusion alarm rule, arming schedule, alarm linkage, and so on, are different.

### 2.2.1 Configure Intrusion Alarm for Intelligent Device

For intelligent device, you should configure the analysis engine of device first, and then set rule, arming schedule, alarm linkage for a specific detection scene that may trigger intrusion alarm.

### Before You Start

- Make sure you have called **NET DVR Init** to initialize the development environment.
- Make sure you have called **NET DVR Login V40** to log in to the device.

## Steps

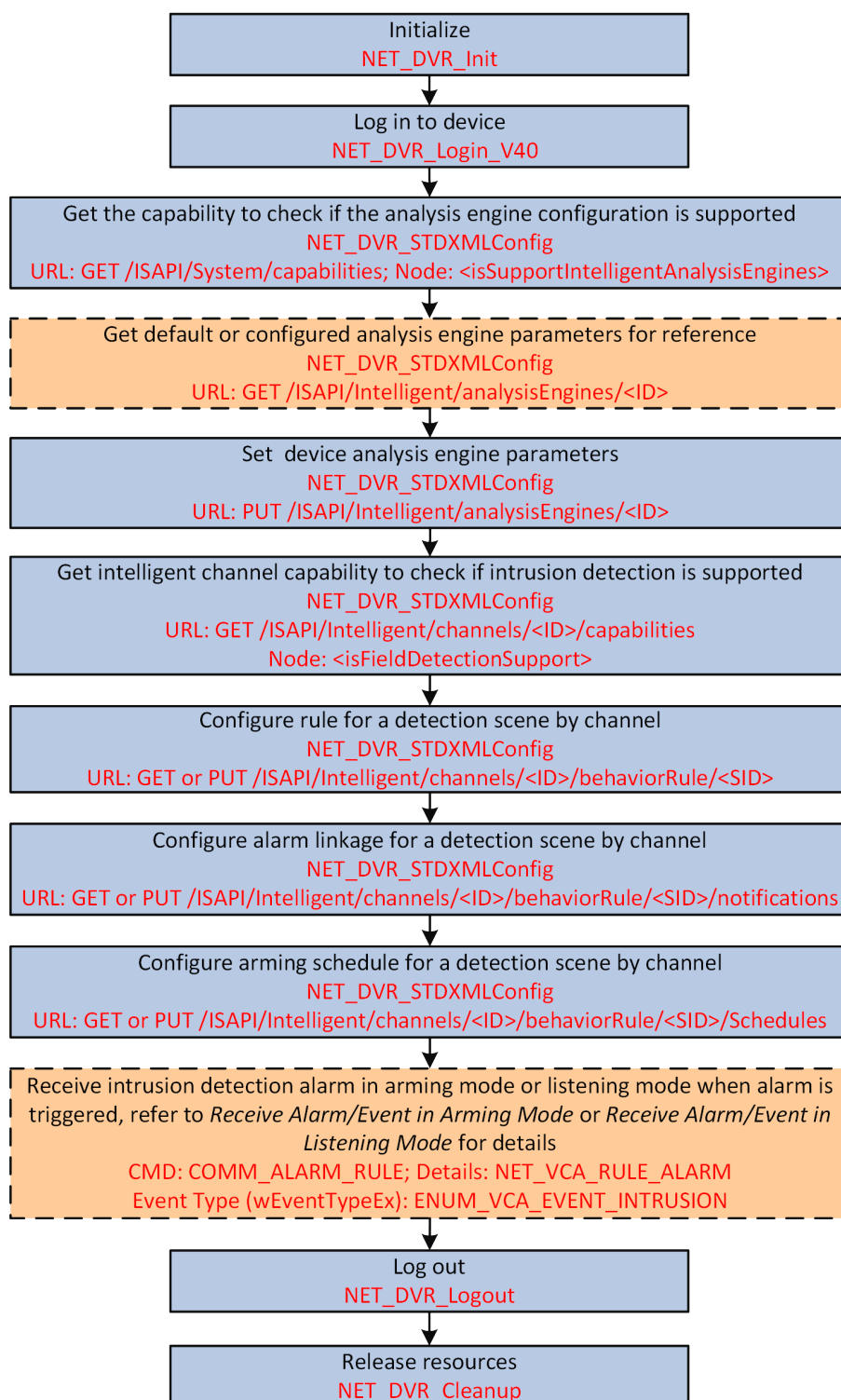


Figure 2-2 Programming Flow of Configuring Intrusion Alarm for Intelligent Device

1. Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: GET **/ISAPI/System/capabilities** for getting the device capability to check if the analysis engine configuration is supported.

The device capability is returned in the message **XML\_DeviceCap** by output parameter pointer (**lpOutputParam**).

If the node **<isSupportIntelligentAnalysisEngines>** is returned in the message and its value is "true", it indicates that analysis engine configuration is supported, you can continue to perform this step.

Otherwise, the analysis engine configuration is not supported by device, please end this task.

2. **Optional:** Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: GET **/ISAPI/Intelligent/analysisEngines/<ID>** for getting default or configured analysis engine parameters for reference.

The analysis engine parameters is returned in the message **XML\_AnalysisEngine** by output parameter pointer (**lpOutputParam**).

3. Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: PUT **/ISAPI/Intelligent/analysisEngines/<ID>**, and set input parameter pointer (**lpInputParam**) to **XML\_AnalysisEngine** for setting the analysis engine parameters.

4. Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: GET **/ISAPI/Intelligent/channels/<ID>/capabilities** for getting intelligent channel capability to check if intrusion detection is supported.

The intelligent channel capability is returned in the message **XML\_IntelliCap** by output parameter pointer (**lpOutputParam**).

If the node **<isFieldDetectionSupport>** is returned in the message and its value is "true", it indicates that intrusion detection is supported, you can continue to perform this step.

Otherwise, the intrusion detection is not supported by device, please end this task.

5. Configure intrusion detection rule for a detection scene by channel.

- 1) **Optional:** Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: GET for getting the intrusion detection rule capability.

The intrusion detection rule capability is returned in the message **XML\_Cap\_RuleInfo** by output parameter pointer (**lpOutputParam**).

- 2) **Optional:** Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: GET **/ISAPI/Intelligent/channels/<ID>/behaviorRule/<SID>/rule/<ID>** for getting default or configured intrusion detection rule for reference.

The intrusion detection rule parameters are returned in the message **XML\_RuleInfo** by output parameter pointer (**lpOutputParam**).

- 3) Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: PUT **/ISAPI/Intelligent/channels/<ID>/behaviorRule/<SID>/rule/<ID>**, and set input parameter pointer (**lpInputParam**) to **XML\_RuleInfo** for setting the intrusion detection rule of a specific detection scene.

6. Configure alarm linkage for the intrusion detection of a detection scene by channel.

- 1) **Optional:** Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: GET **/ISAPI/Intelligent/channels/<ID>/behaviorRule/<SID>/notifications** for getting default or configured intrusion alarm linkage parameters for reference.



The intrusion alarm linkage parameters are returned in the message [XML\\_RuleNotification](#) by output parameter pointer (**IpOutputParam**).

- 2) Call [NET\\_DVR\\_STDXMLConfig](#) to pass through the request URL: PUT [/ISAPI/Intelligent/channels/<ID>/behaviorRule/<SID>/notifications](#) , and set input parameter pointer (**IpInputParam**) to [XML\\_RuleNotification](#) for setting the intrusion alarm linkage of a specific detection scene.



### Note

To receive the alarm in the platform, the linkage action must be set to "center" (upload to center).

- 
7. Configure arming schedule for the intrusion detection of a detection scene by channel.

- 1) **Optional:** Call [NET\\_DVR\\_STDXMLConfig](#) to pass through the request URL: GET [/ISAPI/Intelligent/channels/<ID>/behaviorRule/<SID>/schedules](#) for getting default or configured intrusion arming schedule for reference.

The arming schedule are returned in the message [XML\\_RuleSchedule](#) by output parameter pointer (**IpOutputParam**).

- 2) Call [NET\\_DVR\\_STDXMLConfig](#) to pass through the request URL: PUT [/ISAPI/Intelligent/channels/<ID>/behaviorRule/<SID>/schedules](#) , and set input parameter pointer (**IpInputParam**) to [XML\\_RuleSchedule](#) for setting arming schedule of a specific detection scene.



### Note

The above rules, arming schedule, and alarm linkage can also be configured by logging in to device via web browser.

- 
1. **Optional:** Receive intrusion alarm in arming mode (see [Receive Alarm/Event in Arming Mode](#) ) or listening mode (see [Receive Alarm/Event in Listening Mode](#) ) when alarm is triggered.



### Note

- The command (**ICommand**) to receive intrusion alarms should be set to "COMM\_ALARM\_RULE" (command No.: 0x1102) and the alarm/event type (**wEventTypeEx**) to "ENUM\_VCA\_EVENT\_INTRUSION" in the APIs [NET\\_DVR\\_SetDVRMessageCallBack\\_V50](#) and [NET\\_DVR\\_StartListen\\_V30](#) .
- For the alarm details, refer to the structure [NET\\_VCA\\_RULE\\_ALARM](#) .

---

### What to do next

Call [NET\\_DVR\\_Logout](#) and [NET\\_DVR\\_Cleanup](#) to log out and release resources.

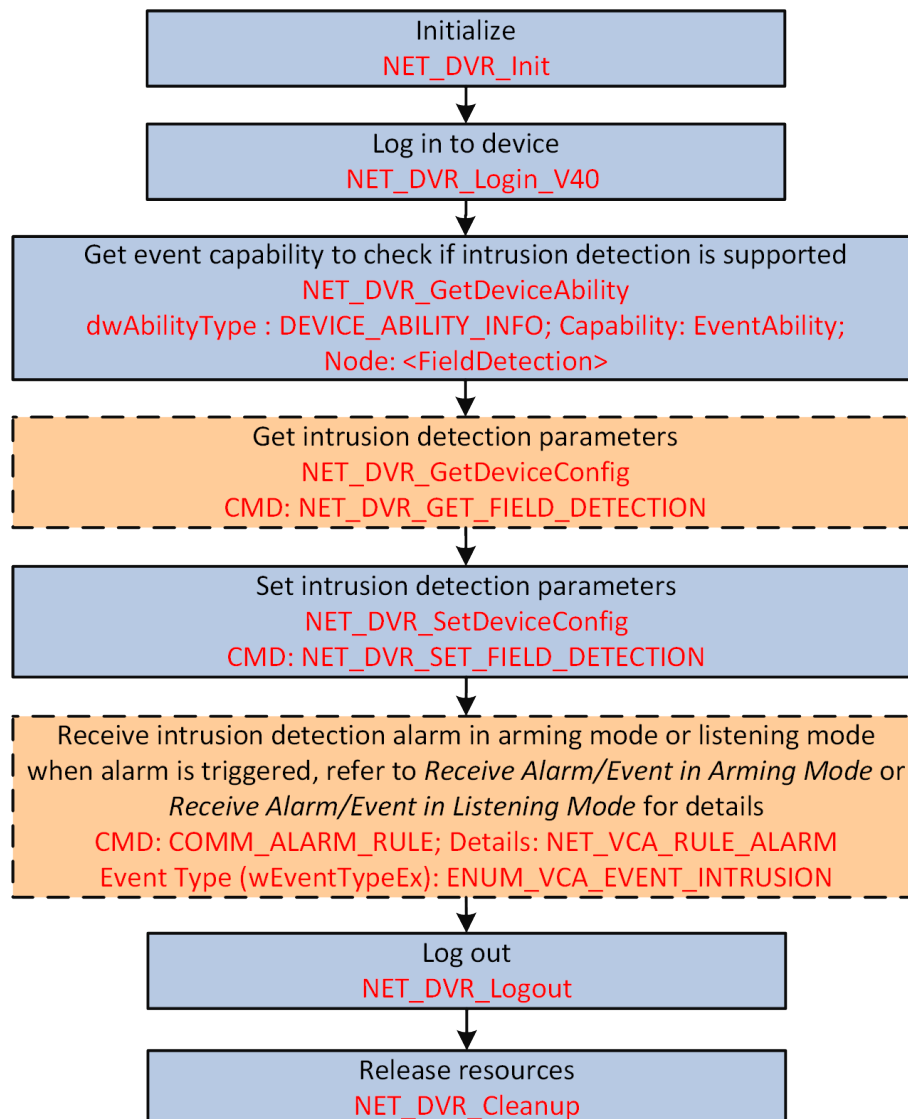
## 2.2.2 Configure Intrusion Alarm for Smart Device

For smart device, you can call HCNetSDK APIs to set intrusion detection and alarm parameters without passing through the request URLs.

### Before You Start

- Make sure you have called **NET\_DVR\_Init** to initialize the development environment.
- Make sure you have called **NET\_DVR\_Login\_V40** to log in to the device.

### Steps



**Figure 2-3 Programming Flow of Configuring Intrusion Alarm for Smart Device**

1. Call **NET\_DVR\_GetDeviceAbility**, set the capability type (**dwAbilityType**) to "DEVICE\_ABILITY\_INFO", and set the input buffer (**pInBuf**) to **XML\_Desc\_EventAbility** for getting the event capability to check if intrusion detection is supported.  
The event capability is returned in the message **XML\_EventAbility** by output buffer (**pOutBuf**).  
If the node **<FieldDetection>** is returned in the message, it indicates that intrusion detection is supported, you can continue to perform this step.

Otherwise, the intrusion detection is not supported by device, please end this task.

2. **Optional:** Call **NET\_DVR\_GetDeviceConfig** with "NET\_DVR\_GET\_FIELD\_DETECTION" (command No.: 3362), and set the input parameter pointer (**pInBuffer**) to multiple (i.e., the value of **dwCount**) structures **NET\_DVR\_CHANNEL\_GROUP** for getting the intrusion detection parameters, including detection rule, arming schedule, alarm linkage, and so on, for reference. The intrusion detection parameters is returned in the structure **NET\_VCA\_FIELDDETECTION** by output parameter **lpOutBuffer**.
3. Call **NET\_DVR\_SetDeviceConfig** with "NET\_DVR\_SET\_FIELD\_DETECTION" (command No.: 3363), set the input parameter pointer (**pInBuffer**) to multiple (i.e., the value of **dwCount**) structures **NET\_DVR\_CHANNEL\_GROUP**, and set the input parameter (**lpInParamBuffer**) to the structure **NET\_VCA\_FIELDDETECTION** for setting the intrusion detection parameters.



### Note

- To receive the alarm in the platform, the linkage action must be set to "center" (upload to center).
  - The above intrusion detection parameters can also be configured by logging in to device via web browser.
- 
4. Receive intrusion alarm in arming mode (see **Receive Alarm/Event in Arming Mode**) or listening mode (see **Receive Alarm/Event in Listening Mode**) when alarm is triggered.



### Note

- The command (**lCommand**) to receive intrusion alarms should be set to "COMM\_ALARM\_RULE" (command No.: 0x1102) and the alarm/event type (**wEventTypeEx**) to "ENUM\_VCA\_EVENT\_INTRUSION" in the APIs **NET\_DVR\_SetDVRMessageCallBack\_V50** and **NET\_DVR\_StartListen\_V30**.
  - For the alarm details, refer to the structure **NET\_VCA\_RULE\_ALARM**.
- 

### What to do next

Call **NET\_DVR\_Logout** and **NET\_DVR\_Cleanup** to log out and release resources.

## 2.2.3 Configure Radar Intrusion Alarm

The radar PTZ camera detects people, vehicle, or other objects which enter and loiter in a predefined virtual region to trigger intrusion alarm, and some certain actions can be taken when the alarm is triggered.

### Before You Start

- Make sure you have called **NET\_DVR\_Init** to initialize the development environment.
- Make sure you have called **NET\_DVR\_Login\_V40** to log in to the device.

## Steps

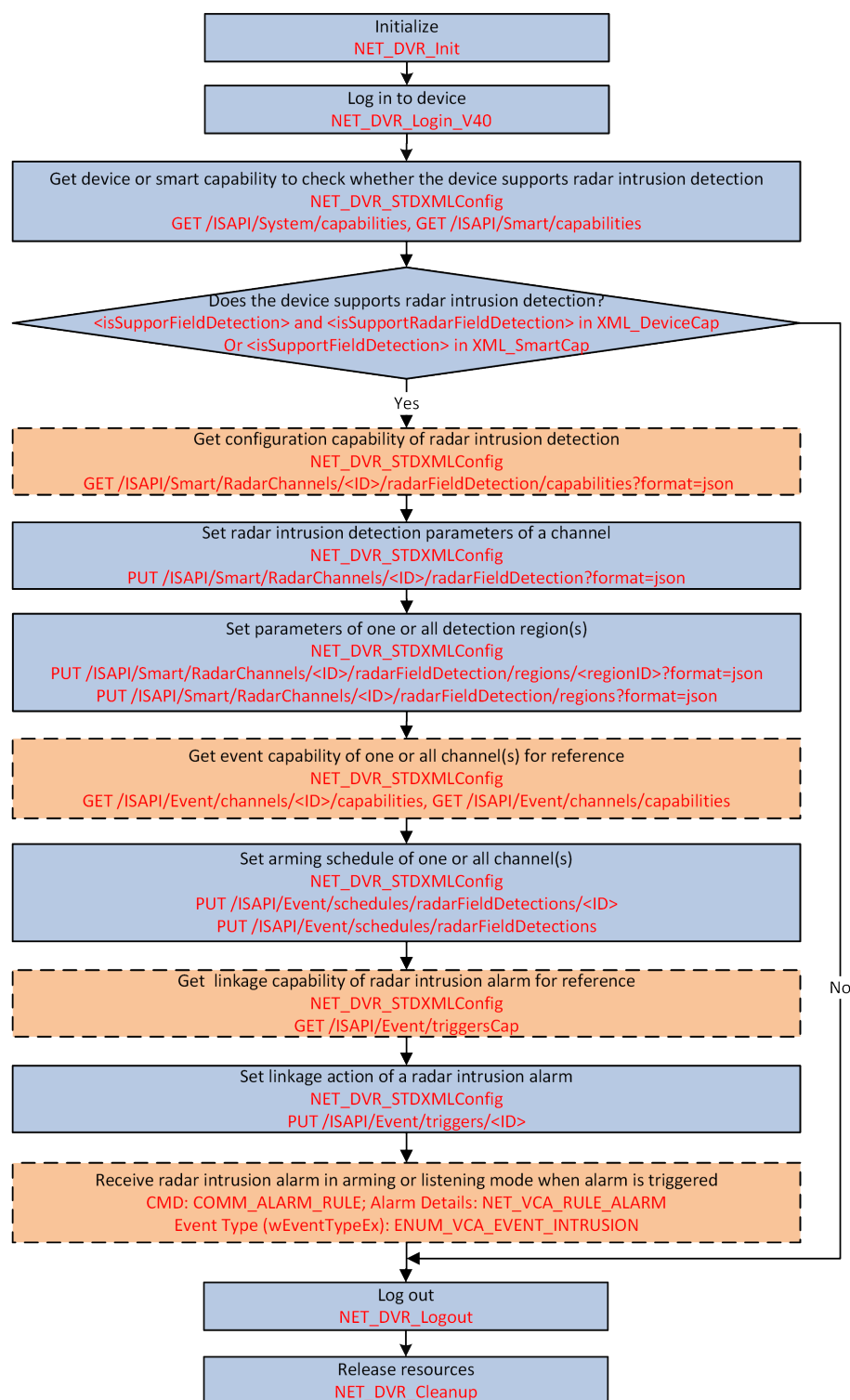


Figure 2-4 API Calling Flow of Configuring Radar Intrusion Alarm

1. Call **NET\_DVR\_STDXMLConfig** to transmit **/ISAPI/System/capabilities** or **/ISAPI/Smart/capabilities** by GET method for getting device or smart capability to check.  
The device capability ( **XML\_DeviceCap** ) or smart capability ( **XML\_SmartCap** ) is returned by **IpOutputParam**.
2. Check whether the device supports radar intrusion detection.
  - It supports and continue the following steps when the nodes **<isSupportFieldDetection>** and **<isSupportRadarFieldDetection>** is returned in the **XML\_DeviceCap** and their values are "true".  
\
    - It supports and continue the following steps when the node **<isSupportFieldDetection>** is returned in the **XML\_SmartCap** and its value is "true".
    - It does not support and end this task when neither the **XML\_DeviceCap** nor the **XML\_SmartCap** contains node **<isSupportFieldDetection>**.
3. Optional: Call **NET\_DVR\_STDXMLConfig** to transmit **/ISAPI/Smart/RadarChannels/<ID>/radarFieldDetection/capabilities?format=json** by GET method for getting configuration capability of radar intrusion detection.  
The configuration capability is returned in the message **JSON\_RadarFieldDetectionCap** by **IpOutputParam**.
4. Call **NET\_DVR\_STDXMLConfig** to transmit **/ISAPI/Smart/RadarChannels/<ID>/radarFieldDetection?format=json** by PUT method and set **IpInputParam** to **JSON\_RadarFieldDetection** for setting radar intrusion detection parameters of a channel.
5. Call **NET\_DVR\_STDXMLConfig** to transmit **/ISAPI/Smart/RadarChannels/<ID>/radarFieldDetection/regions/<regionID>?format=json** or **/ISAPI/Smart/RadarChannels/<ID>/radarFieldDetection/regions?format=json** by PUT method and set **IpInputParam** to **JSON\_RegionInfo** or **JSON\_RegionList** respectively, for setting parameters of one or all detection region(s).
6. Optional: Call **NET\_DVR\_STDXMLConfig** to transmit **/ISAPI/Event/channels/<ID>/capabilities** or **/ISAPI/Event/channels/capabilities** by GET method for getting event capability of one or all channel(s).  
The event capability is returned in the message **XML\_ChannelEventCap** by **IpOutputParam**.
7. Call **NET\_DVR\_STDXMLConfig** to transmit **/ISAPI/Event/schedules/radarFieldDetections/<ID>** or **/ISAPI/Event/schedules/radarFieldDetections** by PUT method, set **<ID>** in the single configuration URI to "radarFieldDetection-<channelID>" (e.g., /ISAPI/Event/schedules/radarFieldDetections/radarFieldDetection-101), and set **IpInputParam** to **XML\_RadarFieldDetectionScheduleList** or **XML\_Schedule** respectively, for setting arming schedules of one or all channel(s).
8. Optional: Call **NET\_DVR\_STDXMLConfig** to transmit **/ISAPI/Event/triggersCap** by GET method for getting linkage capability of radar intrusion alarm.  
The linkage capability is returned in the message **XML\_EventTriggersCap** by **IpOutputParam**.
9. Call **NET\_DVR\_STDXMLConfig** to transmit **/ISAPI/Event/triggers/<eventType>-<channelID>** by PUT method, set **<ID>** in the URI to "radarFieldDetection-<channelID>" (e.g., /ISAPI/Event/

triggers/radarFieldDetection-101), and set **IpInputParam** to ***XML\_EventTrigger*** for setting linkage action of a radar intrusion alarm.



### Note

To receive alarm in the platform, the linkage action must be set to "center" (upload to center).

- 
- 10. Optional:** Set **ICommand** to "COMM\_ALARM\_RULE" (command No.: 0x1102) and set **wEventTypeEx** to "ENUM\_VCA\_EVENT\_INTRUSION" for receiving radar intrusion alarm in arming mode (see ***Receive Alarm/Event in Arming Mode*** ) or listening mode (see ***Receive Alarm/Event in Listening Mode*** ) when alarm is triggered.

The radar intrusion alarm details are returned in the structure ***NET\_VCA\_RULE\_ALARM*** .

### What to do next

Call ***NET\_DVR\_Logout*** and ***NET\_DVR\_Cleanup*** to log out and release resources.

## 2.3 Line Crossing Alarm

Line crossing detection function is used to detect people, vehicles, or objects crossing a predefined virtual line, and some certain actions can be taken when the alarm is triggered. The crossing direction can be set to bidirectional, from left to right, or from right to left. And for intelligent device and smart device, the APIs or commands called to configure line crossing alarm rule, arming schedule, alarm linkage, and so on, are different.

### 2.3.1 Configure Line Crossing Alarm for Intelligent Device

For intelligent device, you should configure the analysis engine of device first, and then set rule, arming schedule, alarm linkage for a specific detection scene that may trigger line crossing alarm.

#### Before You Start

- Make sure you have called ***NET\_DVR\_Init*** to initialize the development environment.
- Make sure you have called ***NET\_DVR\_Login\_V40*** to log in to the device.

## Steps

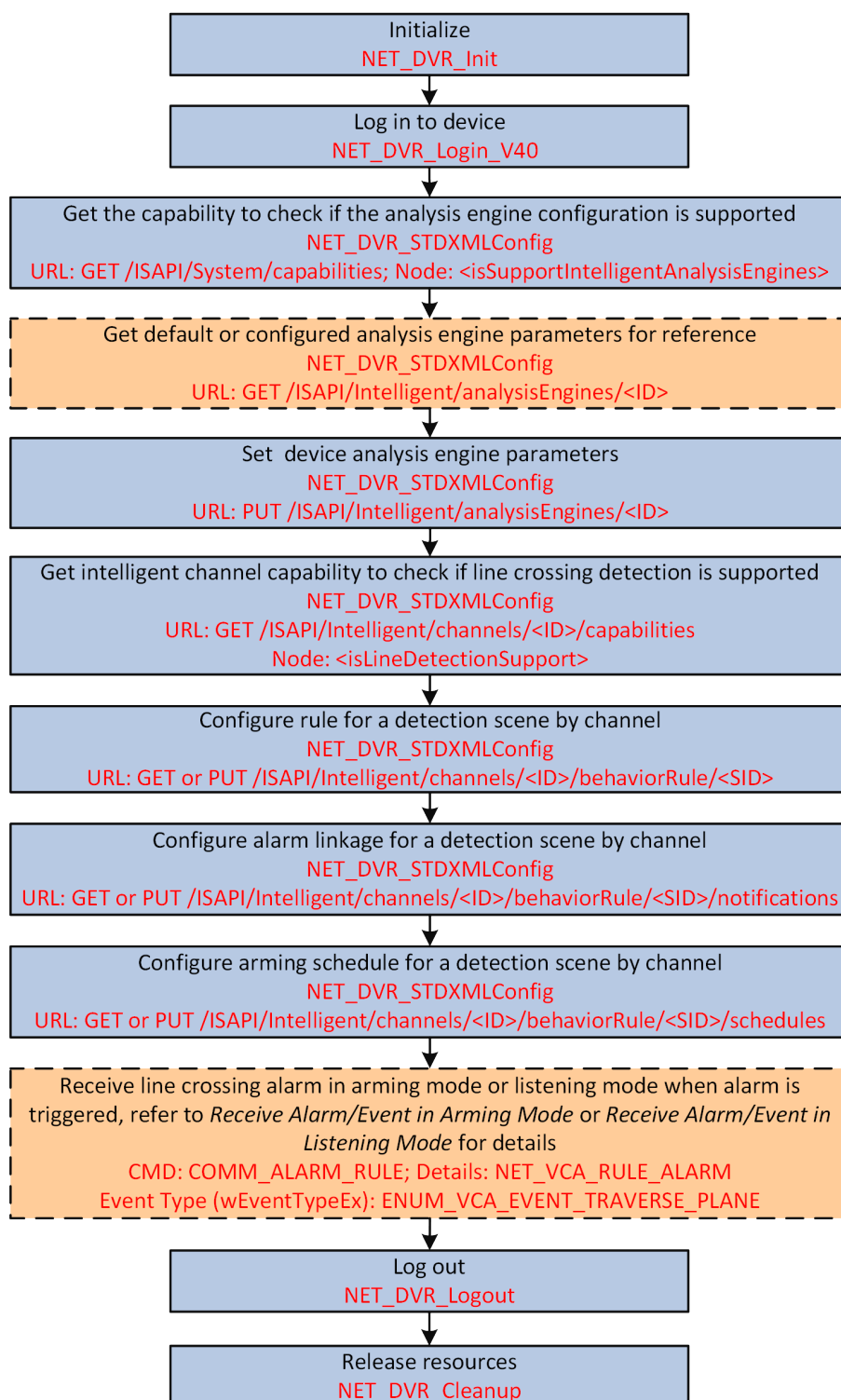


Figure 2-5 Programming Flow of Configuring Line Crossing Alarm for Intelligent Device

1. Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: GET **/ISAPI/System/capabilities** for getting the device capability to check if the analysis engine configuration is supported.

The device capability is returned in the message **XML\_DeviceCap** by output parameter pointer (**lpOutputParam**).

If the node **<isSupportIntelligentAnalysisEngines>** is returned in the message and its value is "true", it indicates that analysis engine configuration is supported, you can continue to perform this step.

Otherwise, the analysis engine configuration is not supported by device, please end this task.

2. **Optional:** Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: GET **/ISAPI/Intelligent/analysisEngines/<ID>** for getting default or configured analysis engine parameters for reference.

The analysis engine parameters is returned in the message **XML\_AnalysisEngine** by output parameter pointer (**lpOutputParam**).

3. Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: PUT **/ISAPI/Intelligent/analysisEngines/<ID>**, and set input parameter pointer (**lpInputParam**) to **XML\_AnalysisEngine** for setting the analysis engine parameters.

4. Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: GET **/ISAPI/Intelligent/channels/<ID>/capabilities** for getting intelligent channel capability to check if line crossing detection is supported.

The intelligent channel capability is returned in the message **XML\_IntelliCap** by output parameter pointer (**lpOutputParam**).

If the node **<isLineDetectionSupport>** is returned in the message and its value is "true", it indicates that line crossing detection is supported, you can continue to perform this step.

Otherwise, the line crossing detection is not supported by device, please end this task.

5. Configure line crossing detection rule for a detection scene by channel.

- 1) **Optional:** Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: GET for getting the line crossing detection rule capability.

The line crossing detection rule capability is returned in the message **XML\_Cap\_RuleInfo** by output parameter pointer (**lpOutputParam**).

- 2) **Optional:** Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: GET for getting default or configured line crossing detection rule for reference.

The line crossing detection rule parameters are returned in the message **XML\_RuleInfo** by output parameter pointer (**lpOutputParam**).

- 3) Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: PUT, and set input parameter pointer (**lpInputParam**) to **XML\_RuleInfo** for setting the line crossing detection rule of a specific detection scene.

6. Configure alarm linkage for the line crossing detection of a detection scene by channel.

- 1) **Optional:** Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: GET **/ISAPI/Intelligent/channels/<ID>/behaviorRule/<SID>/notifications** for getting default or configured line crossing alarm linkage parameters for reference.



The line crossing alarm linkage parameters are returned in the message

**XML\_RuleNotification** by output parameter pointer (**lpOutputParam**).

- 2) Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: PUT **/ISAPI/Intelligent/channels/<ID>/behaviorRule/<SID>/notifications** , and set input parameter pointer (**lpInputParam**) to **XML\_RuleNotification** for setting the line crossing alarm linkage of a specific detection scene.



### Note

To receive the alarm in the platform, the linkage action must be set to "center" (upload to center).

- 
7. Configure arming schedule for the line crossing detection of a detection scene by channel.

- 1) **Optional:** Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: GET **/ISAPI/Intelligent/channels/<ID>/behaviorRule/<SID>/schedules** for getting default or configured line crossing arming schedule for reference.

The arming schedule are returned in the message **XML\_RuleSchedule** by output parameter pointer (**lpOutputParam**).

- 2) Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: PUT **/ISAPI/Intelligent/channels/<ID>/behaviorRule/<SID>/schedules** , and set input parameter pointer (**lpInputParam**) to **XML\_RuleSchedule** for setting arming schedule of a specific detection scene.



### Note

The above rules, arming schedule, and alarm linkage can also be configured by logging in to device via web browser.

- 
1. **Optional:** Receive line crossing alarm in arming mode (see **Receive Alarm/Event in Arming Mode** ) or listening mode (see **Receive Alarm/Event in Listening Mode** ) when alarm is triggered.



### Note

- The command (**lCommand**) to receive line crossing alarms should be set to **COMM\_ALARM\_RULE** (command No.: 0x1102) and the alarm/event type (**wEventTypeEx**) to "ENUM\_VCA\_EVENT\_TRAVERSE\_PLANE" in the APIs **NET\_DVR\_SetDVRMessageCallBack\_V50** and **NET\_DVR\_StartListen\_V30** .
- For the alarm details, refer to the structure .

---

### What to do next

Call **NET\_DVR\_Logout** and **NET\_DVR\_Cleanup** to log out and release resources.

### 2.3.2 Configure Line Crossing Alarm for Smart Device

For smart device, you can call HCNetSDK APIs to set line crossing detection and alarm parameters without passing through the request URLs.

#### Before You Start

- Make sure you have called ***NET\_DVR\_Init*** to initialize the development environment.
- Make sure you have called ***NET\_DVR\_Login\_V40*** to log in to the device.

#### Steps

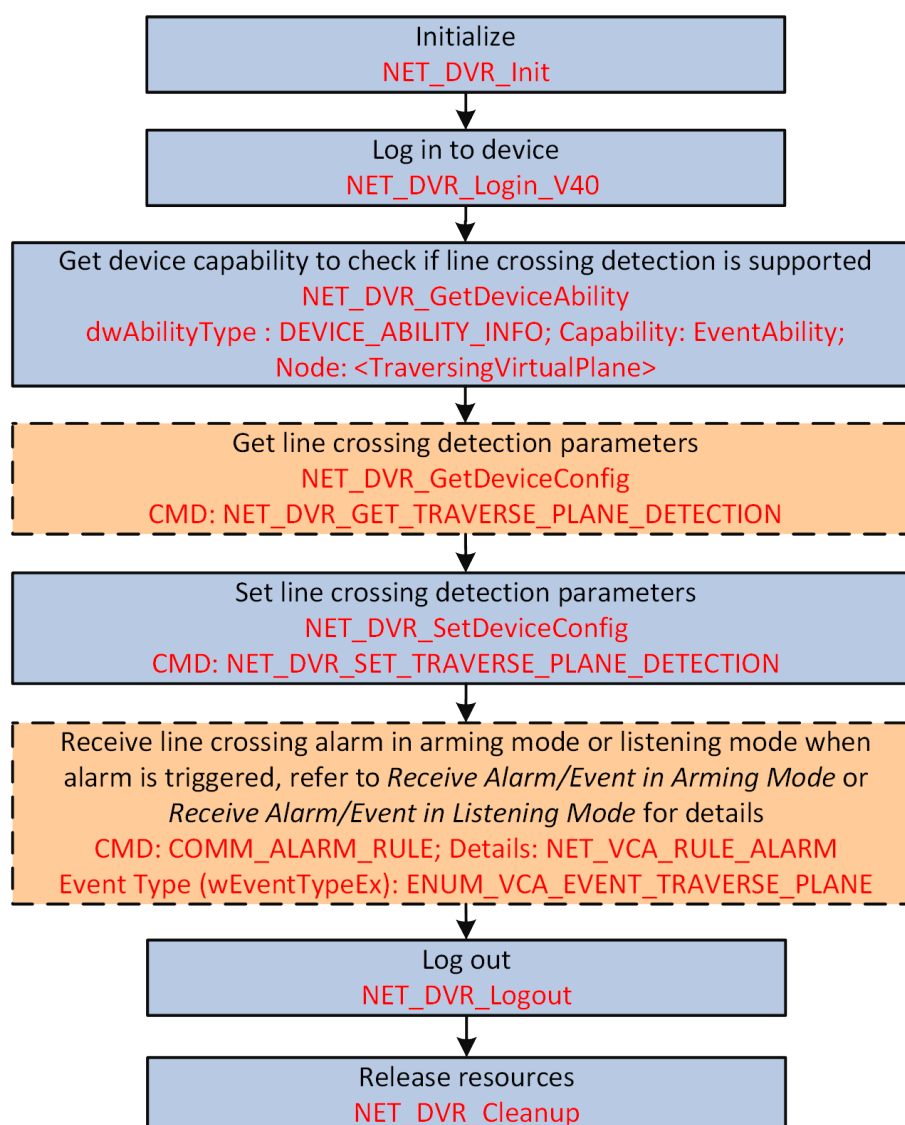


Figure 2-6 Programming Flow of Configuring Line Crossing Alarm for Smart Device

1. Call **NET\_DVR\_GetDeviceAbility**, set the capability type (**dwAbilityType**) to "DEVICE\_ABILITY\_INFO", and set the input buffer (**pInBuf**) to **XML\_Desc\_EventAbility** for getting the event capability to check if line crossing detection is supported.  
The event capability is returned in the message **XML\_EventAbility** by output buffer (**pOutBuf**).  
If the node <TraversingVirtualPlane> is returned in the message, it indicates that line crossing detection is supported, you can continue to perform this step.  
Otherwise, the line crossing detection is not supported by device, please end this task.
2. **Optional:** Call **NET\_DVR\_GetDeviceConfig** with "NET\_DVR\_GET\_TRAVERSE\_PLANE\_DETECTION" (command No.: 3360), and set the input parameter pointer (**pInBuffer**) to multiple (i.e., the value of **dwCount**) structures **NET\_DVR\_CHANNEL\_GROUP** for getting the line crossing detection parameters, including detection rule, arming schedule, alarm linkage, and so on, for reference.  
The line crossing detection parameters is returned in the structure **NET\_VCA\_TRAVERSE\_PLANE\_DETECTION** by output parameter **lpOutBuffer**.
3. Call **NET\_DVR\_SetDeviceConfig** with "NET\_DVR\_SET\_TRAVERSE\_PLANE\_DETECTION" (command No.: 3361), set the input parameter pointer (**pInBuffer**) to multiple (i.e., the value of **dwCount**) structures **NET\_DVR\_CHANNEL\_GROUP**, and set the input parameter (**lpInParamBuffer**) to the structure **NET\_VCA\_TRAVERSE\_PLANE\_DETECTION** for setting the line crossing detection parameters.



### Note

- To receive the alarm in the platform, the linkage action must be set to "center" (upload to center).
- The above line crossing detection parameters can also be configured by logging in to device via web browser.

- 
4. Receive line crossing alarm in arming mode (see **Receive Alarm/Event in Arming Mode**) or listening mode (see **Receive Alarm/Event in Listening Mode**) when alarm is triggered.



### Note

- The command (**lCommand**) to receive line crossing alarms should be set to **COMM\_ALARM\_RULE** (command No.: 0x1102) and the alarm/event type (**wEventTypeEx**) to "ENUM\_VCA\_EVENT\_TRAVERSE\_PLANE" in the APIs **NET\_DVR\_SetDVRMessageCallback\_V50** and **NET\_DVR\_StartListen\_V30**.
- For the alarm details, refer to the structure **NET\_VCA\_RULE\_ALARM**.

---

### What to do next

Call **NET\_DVR\_Logout** and **NET\_DVR\_Cleanup** to log out and release resources.

### 2.3.3 Configure Radar Line Crossing Alarm

The radar PTZ camera detects people, vehicles, or objects crossing a predefined virtual line to trigger line crossing alarm, and some certain actions can be taken when the alarm is triggered. The crossing direction can be set to bidirectional, from left to right, or from right to left.

#### Before You Start

- Make sure you have called **NET\_DVR\_Init** to initialize the development environment.
- Make sure you have called **NET\_DVR\_Login\_V40** to log in to the device.

## Steps

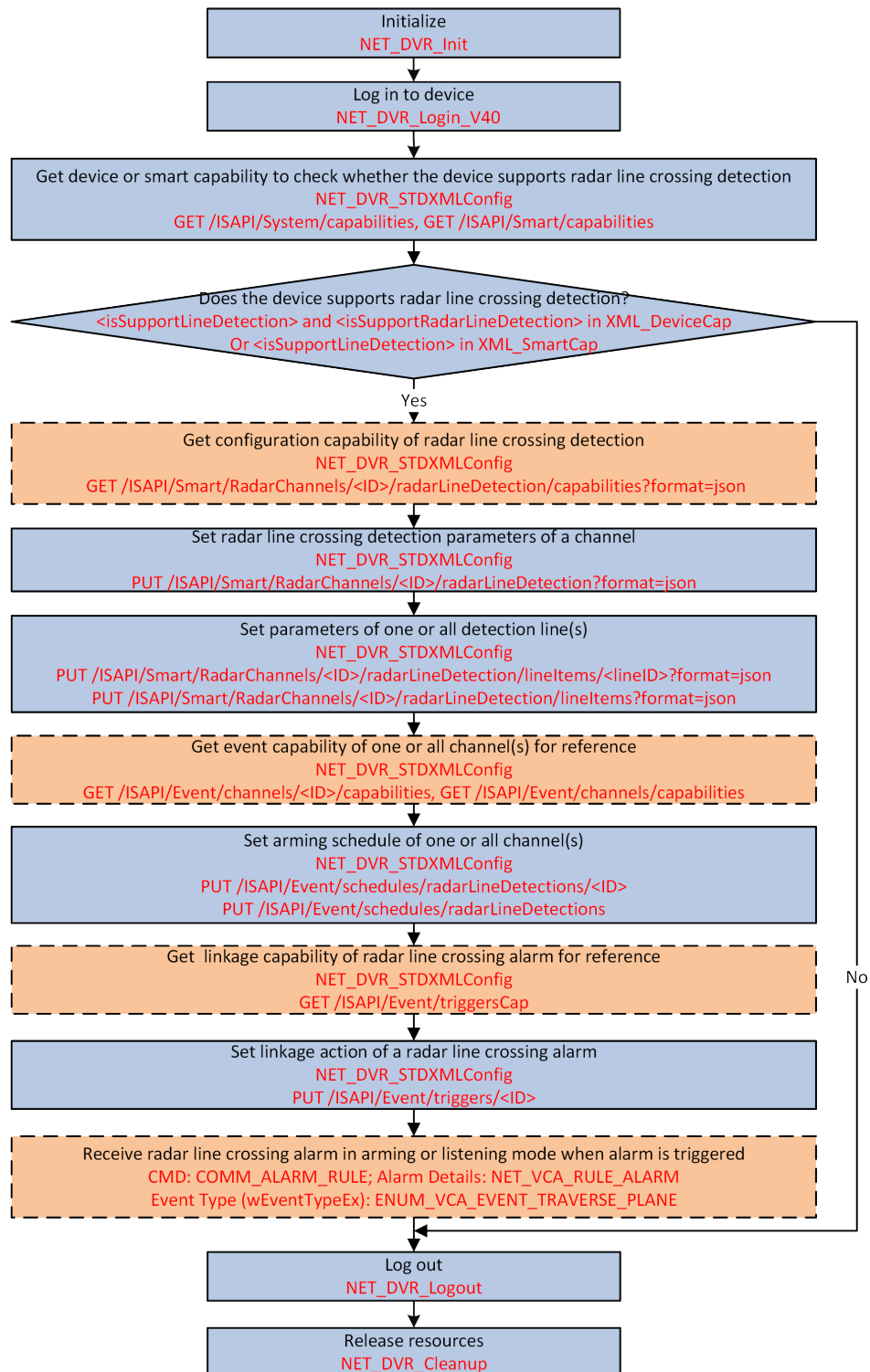


Figure 2-7 API Calling Flow of Configuring Radar Line Crossing Alarm

1. Call **NET\_DVR\_STDXMLConfig** to transmit **/ISAPI/System/capabilities** or **/ISAPI/Smart/capabilities** by GET method for getting device or smart capability to check.

The device capability ( **XML\_DeviceCap** ) or smart capability ( **XML\_SmartCap** ) is returned by **IpOutputParam**.

2. Check whether the device supports radar line crossing detection.
  - It supports and continue the following steps when the nodes **<isSupportLineDetection>** and **<isSupportRadarLineDetection>** is returned in the **XML\_DeviceCap** and their values are "true".  
\
    - It supports and continue the following steps when the node **<isSupportLineDetection>** is returned in the **XML\_SmartCap** and its value is "true".
    - It does not support and end this task when neither the **XML\_DeviceCap** nor the **XML\_SmartCap** contains node **<isSupportLineDetection>**.

3. Optional: Call **NET\_DVR\_STDXMLConfig** to transmit **/ISAPI/Smart/RadarChannels/<ID>/radarLineDetection/capabilities?format=json** by GET method for getting configuration capability of radar line crossing detection.

The configuration capability is returned in the message **JSON\_RadarLineDetectionCap** by **IpOutputParam**.

4. Call **NET\_DVR\_STDXMLConfig** to transmit **/ISAPI/Smart/RadarChannels/<ID>/radarLineDetection?format=json** by PUT method and set **IpInputParam** to **JSON\_RadarLineDetection** for setting radar line crossing detection parameters of a channel.

5. Call **NET\_DVR\_STDXMLConfig** to transmit **/ISAPI/Smart/RadarChannels/<ID>/radarLineDetection/lineltems/<lineID>?format=json** or **/ISAPI/Smart/RadarChannels/<ID>/radarLineDetection/lineltems?format=json** by PUT method and set **IpInputParam** to **JSON\_LineltemInfo** or **JSON\_LineltemList** respectively, for setting parameters of one or all detection line(s).

6. Optional: Call **NET\_DVR\_STDXMLConfig** to transmit **/ISAPI/Event/channels/<ID>/capabilities** or **/ISAPI/Event/channels/capabilities** by GET method for getting event capability of one or all channel(s).

The event capability is returned in the message **XML\_ChannelEventCap** by **IpOutputParam**.

7. Call **NET\_DVR\_STDXMLConfig** to transmit **/ISAPI/Event/schedules/radarLineDetections/<ID>** or **/ISAPI/Event/schedules/radarLineDetections** by PUT method, set **<ID>** in the single configuration URI to "radarLineDetection-<channelID>" (e.g., /ISAPI/Event/schedules/radarLineDetections/radarLineDetection-101), and set **IpInputParam** to **XML\_RadarLineDetectionScheduleList** or **XML\_Schedule** respectively, for setting arming schedules of one or all channel(s).

8. Optional: Call **NET\_DVR\_STDXMLConfig** to transmit **/ISAPI/Event/triggersCap** by GET method for getting linkage capability of radar line crossing alarm.

The linkage capability is returned in the message **XML\_EventTriggersCap** by **IpOutputParam**.

9. Call **NET\_DVR\_STDXMLConfig** to transmit **/ISAPI/Event/triggers/<eventType>-<channelID>** by PUT method, set **<ID>** in the URI to "radarLineDetection-<channelID>" (e.g., /ISAPI/Event/

triggers/radarLineDetection-101), and set **lpInputParam** to ***XML\_EventTrigger*** for setting linkage action of a radar line crossing alarm.



### Note

To receive alarm in the platform, the linkage action must be set to "center" (upload to center).

- 
- 10. Optional:** Set **ICommand** to "COMM\_ALARM\_RULE" (command No.: 0x1102) and set **wEventTypeEx** to "ENUM\_VCA\_EVENT\_TRAVERSE\_PLANE" for receiving radar line crossing alarm in arming mode (see ***Receive Alarm/Event in Arming Mode*** ) or listening mode (see ***Receive Alarm/Event in Listening Mode*** ) when alarm is triggered.

The radar line crossing alarm details are returned in the structure ***NET\_VCA\_RULE\_ALARM*** .

### What to do next

Call ***NET\_DVR\_Logout*** and ***NET\_DVR\_Cleanup*** to log out and release resources.

## 2.4 Region Entrance Alarm

Region entrance detection function detects objects that enter a predefined virtual region from the outside place, and some certain actions can be taken when the alarm is triggered. And for intelligent device and smart device, the APIs or commands called to configure region entrance alarm rule, arming schedule, alarm linkage, and so on, are different.

### 2.4.1 Configure Region Entrance Alarm for Intelligent Device

For intelligent device, you should configure the analysis engine of device first, and then set rule, arming schedule, alarm linkage for a specific detection scene that may trigger region entrance alarm.

#### Before You Start

- Make sure you have called ***NET\_DVR\_Init*** to initialize the development environment.
- Make sure you have called ***NET\_DVR\_Login\_V40*** to log in to the device.

### Steps

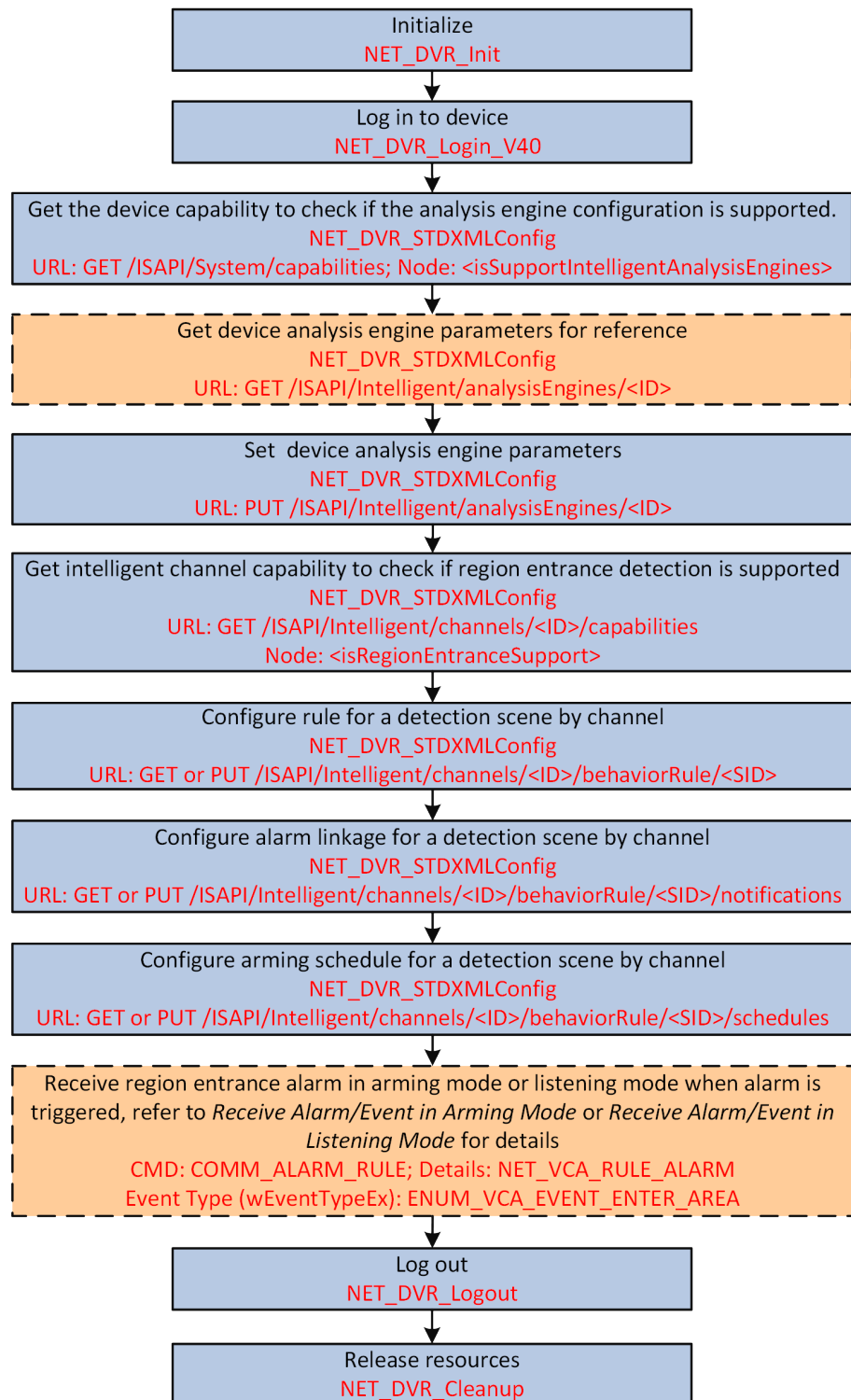


Figure 2-8 API Calling Flow of Configuring Region Entrance Alarm for Intelligent Device



1. Call **NET\_DVR\_STDXMLConfig** to transmit **/ISAPI/System/capabilities** by GET method for getting the device capability to check if the analysis engine configuration is supported.

The device capability is returned in the message **XML\_DeviceCap** by **IpOutputParam**.

If the node **<isSupportIntelligentAnalysisEngines>** is returned in the message and its value is "true", it indicates that analysis engine configuration is supported, you can continue to perform this step.

Otherwise, the analysis engine configuration is not supported by device, please end this task.

2. **Optional:** Call **NET\_DVR\_STDXMLConfig** to transmit **/ISAPI/Intelligent/analysisEngines/<ID>** by GET method for getting default or configured analysis engine parameters for reference.

The analysis engine parameters is returned in the message **XML\_AnalysisEngine** by **IpOutputParam**.

3. Call **NET\_DVR\_STDXMLConfig** to transmit **/ISAPI/Intelligent/analysisEngines/<ID>** by PUT method and set **IpInputParam** to **XML\_AnalysisEngine** for setting the analysis engine parameters.

4. Call **NET\_DVR\_STDXMLConfig** to transmit **/ISAPI/Intelligent/channels/<ID>/capabilities** by GET method for getting intelligent channel capability to check if region entrance detection is supported.

The intelligent channel capability is returned in the message **XML\_IntelliCap** by **IpOutputParam**.

If the node **<isRegionEntranceSupport>** is returned in the message and its value is "true", it indicates that region entrance detection is supported, you can continue to perform this step.

Otherwise, the region entrance detection is not supported by device, please end this task.

5. Configure region entrance detection rule for a detection scene by channel.

- 1) **Optional:** Call **NET\_DVR\_STDXMLConfig** to transmit **/ISAPI/Intelligent/channels/<ID>/behaviorRule/<SID>/rule/capabilities** by GET method for getting the region entrance detection rule capability.

The region entrance detection rule capability is returned in the message **XML\_Cap\_RuleInfo** by **IpOutputParam**.

- 2) **Optional:** Call **NET\_DVR\_STDXMLConfig** to transmit **/ISAPI/Intelligent/channels/<ID>/behaviorRule/<SID>/rule/<ID>** by GET method for getting default or configured region entrance detection rule for reference.

The region entrance detection rule parameters are returned in the message **XML\_RuleInfo** by **IpOutputParam**.

- 3) Call **NET\_DVR\_STDXMLConfig** to transmit **/ISAPI/Intelligent/channels/<ID>/behaviorRule/<SID>/rule/<ID>** by PUT method and set **IpInputParam** to **XML\_RuleInfo** for setting the region entrance detection rule of a specific detection scene.

6. Configure alarm linkage for the region entrance detection of a detection scene by channel.

- 1) **Optional:** Call **NET\_DVR\_STDXMLConfig** to transmit **/ISAPI/Intelligent/channels/<ID>/behaviorRule/<SID>/notifications** by GET method for getting default or configured region entrance alarm linkage parameters for reference.

The region entrance alarm linkage parameters are returned in the message **XML\_RuleNotification** by **IpOutputParam**.

- 2) Call **NET\_DVR\_STDXMLConfig** to transmit **/ISAPI/Intelligent/channels/<ID>/behaviorRule/<SID>/notifications** by PUT method and set **IpInputParam** to **XML\_RuleNotification** for setting the region entrance alarm linkage of a specific detection scene.



### Note

To receive the alarm in the platform, the linkage action must be set to "center" (upload to center).

7. Configure arming schedule for the region entrance detection of a detection scene by channel.

- 1) **Optional:** Call **NET\_DVR\_STDXMLConfig** to transmit **/ISAPI/Intelligent/channels/<ID>/behaviorRule/<SID>/schedules** by GET method for getting default or configured region entrance arming schedule for reference.

The arming schedule are returned in the message **XML\_RuleSchedule** by **IpOutputParam**.

- 2) Call **NET\_DVR\_STDXMLConfig** to transmit **/ISAPI/Intelligent/channels/<ID>/behaviorRule/<SID>/schedules** by PUT method and set **IpInputParam** to **XML\_RuleSchedule** for setting arming schedule of a specific detection scene.



### Note

The above rules, arming schedule, and alarm linkage can also be configured by logging in to device via web browser.

8. **Optional:** Set **ICommand** to "COMM\_ALARM\_RULE" (command No.: 0x1102) and set **wEventTypeEx** to "ENUM\_VCA\_EVENT\_ENTER\_AREA" in the alarm callback function to receive region entrance alarm in arming mode (see **Receive Alarm/Event in Arming Mode** ) or listening mode (see **Receive Alarm/Event in Listening Mode** ) when alarm is triggered.

The received alarm details in the structure **NET\_VCA\_RULE\_ALARM** .

### What to do next

Call **NET\_DVR\_Logout** and **NET\_DVR\_Cleanup** to log out and release resources.

## 2.4.2 Configure Region Entrance Alarm for Smart Device

For smart device, you can call HCNetsdk APIs to set region entrance detection and alarm parameters without passing through the request URLs.

### Before You Start

- Make sure you have called **NET\_DVR\_Init** to initialize the development environment.
- Make sure you have called **NET\_DVR\_Login\_V40** to log in to the device.

## Steps

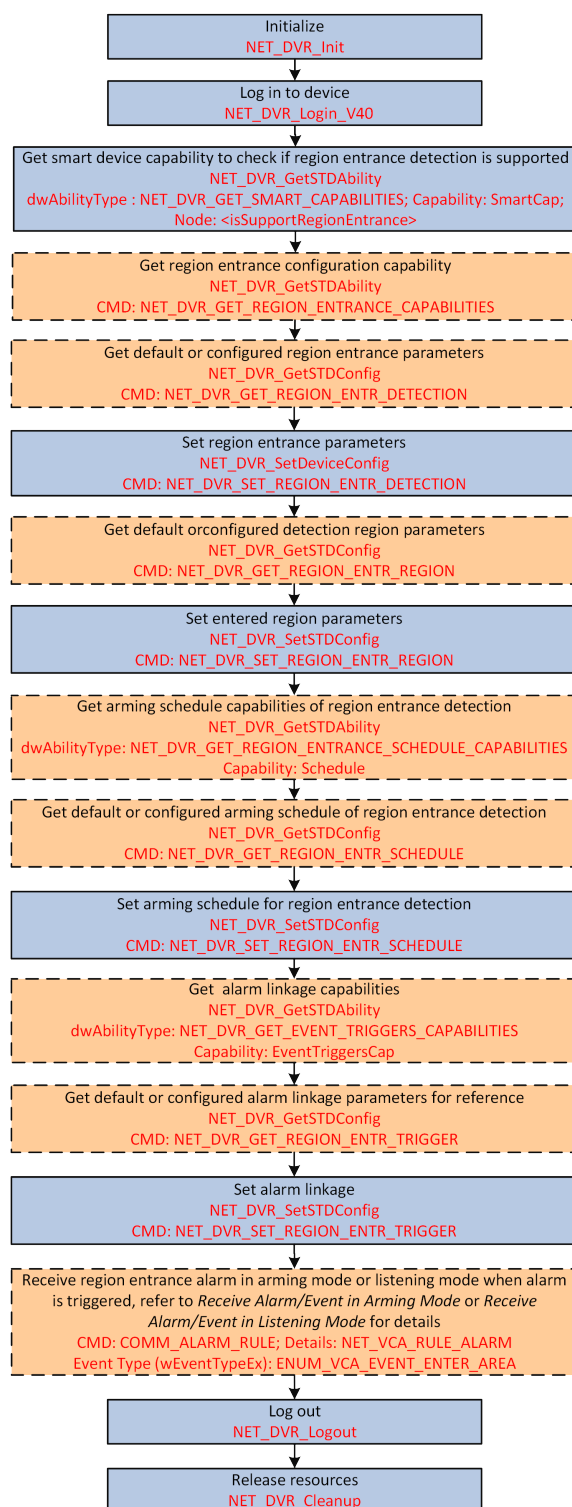


Figure 2-9 API Calling Flow of Configuring Region Entrance Alarm for Smart Device

1. Call **NET\_DVR\_GetSTDAbility**, and set **dwAbilityType** to "NET\_DVR\_GET\_SMART\_CAPABILITIES" (value: 3500) for getting the smart device capability to check whether the region entrance detection is supported.

The smart device capability is returned in the message **XML\_SmartCap** by **lpOutBuffer**.

If the node **<isSupportRegionEntrance>** is returned in the message and its value is "true", it indicates that region entrance detection is supported, you can continue to perform this step.

Otherwise, the region entrance detection is not supported by device, please end this task.

2. Configure region entrance detection rules (including all detection regions).
  - 1) **Optional:** Call **NET\_DVR\_GetSTDAbility**, set **dwAbilityType** to "NET\_DVR\_GET\_REGION\_ENTRANCE\_CAPABILITIES" (value: 3502), and set **lpCondBuffer** to a 4-byte channel No. for getting configuration capability of region entrance detection rules.

The configuration capability is returned in the message **XML\_Cap\_RegionEntrance** by **lpOutBuffer**.
  - 2) **Optional:** Call **NET\_DVR\_GetSTDConfig** with "NET\_DVR\_GET\_REGION\_ENTR\_DETECTION" (command No.: 3503) and set **lpCondBuffer** to a 4-byte channel No. for getting default or configured region entrance detection rules for reference.

The region entrance detection parameters are returned in the structure **NET\_DVR\_REGION\_ENTRANCE\_DETECTION** by **lpOutBuffer**.
  - 3) Call **NET\_DVR\_SetSTDConfig** with "NET\_DVR\_SET\_REGION\_ENTR\_DETECTION" (command No.: 3504), set **lpCondBuffer** to a 4-byte channel No., and set **lpInBuffer** to **NET\_DVR\_REGION\_ENTRANCE\_DETECTION** for setting region entrance detection rules.
3. **Optional:** Configure parameters of a detection region.
  - 1) **Optional:** Call **NET\_DVR\_GetSTDConfig** with "NET\_DVR\_GET\_REGION\_ENTR\_REGION" (command No.: 3505), and set **lpCondBuffer** to **NET\_DVR\_SMART\_REGION\_COND** for getting default or configured parameters of a detection region for reference.

The parameters of a detection are returned in the structure **NET\_DVR\_REGION\_ENTRANCE\_DETECTION** by **lpOutBuffer**.
  - 2) Call **NET\_DVR\_SetSTDConfig** with "NET\_DVR\_SET\_REGION\_ENTR\_REGION" (command No.: 3506), set **lpCondBuffer** to **NET\_DVR\_SMART\_REGION\_COND**, and set **lpInBuffer** to **NET\_DVR\_REGION\_ENTRANCE\_DETECTION** for setting parameters of a detection region.
4. Configure arming schedule of region entrance detection.
  - 1) **Optional:** Call **NET\_DVR\_GetSTDAbility**, set **dwAbilityType** to "NET\_DVR\_GET\_REGION\_ENTRANCE\_SCHEDULE\_CAPABILITIES" (value: 3584), and set **lpCondBuffer** to a 4-byte channel No. for getting arming schedule capability of region entrance detection.

The arming schedule capability of region entrance detection is returned in the message **XML\_Cap\_Schedule** by **lpOutBuffer**.
  - 2) **Optional:** Call **NET\_DVR\_GetSTDConfig** with "NET\_DVR\_GET\_REGION\_ENTR\_SCHEDULE" (command No.: 3509), and set condition buffer (**lpCondBuffer**) of structure **NET\_DVR\_STD\_ABILITY** to 4-byte channel No. for getting default or configured arming schedule parameters of region entrance detection for reference.

The arming schedule parameters of region entrance detection are returned in the structure **NET\_DVR\_EVENT\_SCHEDULE** by **IpOutBuffer**.

- 3) Call **NET\_DVR\_SetSTDConfig** with "NET\_DVR\_SET\_REGION\_ENTR\_SCHEDULE" (command No.: 3510), set **IpCondBuffer** to a 4-byte channel No., and set **IpInBuffer** to **NET\_DVR\_EVENT\_SCHEDULE** for setting arming schedule of region entrance detection.

5. Configure alarm linkage for region entrance detection.

- 1) **Optional:** Call **NET\_DVR\_GetSTDAbility**, and set **dwAbilityType** to "NET\_DVR\_GET\_EVENT\_TRIGGERS\_CAPABILITIES" (value: 3501) for getting alarm linkage capability.

The alarm linkage capability is returned in the message **XML\_EventTriggersCap** by **IpOutBuffer**.

- 2) **Optional:** Call **NET\_DVR\_GetSTDConfig** with "NET\_DVR\_GET\_REGION\_ENTR\_TRIGGER" (command No.: 3507) and set **IpCondBuffer** to 4-byte channel No. for getting default or configured alarm linkage parameters for reference.

The alarm linkage parameters are returned in the structure **NET\_DVR\_EVENT\_TRIGGER** by **IpOutBuffer**.

- 3) Call **NET\_DVR\_SetSTDConfig** with "NET\_DVR\_SET\_REGION\_ENTR\_TRIGGER" (command No.: 3508), set **IpCondBuffer** to a 4-byte channel No., and set **IpInBuffer** to **NET\_DVR\_EVENT\_TRIGGER** for setting alarm linkage.



### Note

- To receive the alarm in the platform, the linkage action must be set to "center" (upload to center).
- The above region entrance detection parameters can also be configured by logging in to device via web browser.

- 
6. **Optional:** Set **ICommand** to "COMM\_ALARM\_RULE" (command No.: 0x1102) and set **wEventTypeEx** to "ENUM\_VCA\_EVENT\_ENTER\_AREA" in the alarm callback function to receive region entrance alarm in arming mode (see **Receive Alarm/Event in Arming Mode**) or listening mode (see **Receive Alarm/Event in Listening Mode**) when alarm is triggered.

The received alarm details in the structure **NET\_VCA\_RULE\_ALARM**.

### What to do next

Call **NET\_DVR\_Logout** and **NET\_DVR\_Cleanup** to log out and release resources.

## 2.5 Region Exiting Alarm

Region exiting detection function detects objects that exit from a predefined virtual region to the outside place, and some certain actions can be taken when the alarm is triggered. And for intelligent device and smart device, the APIs or commands called to configure region exiting alarm rule, arming schedule, alarm linkage, and so on, are different.

### 2.5.1 Configure Region Exiting Alarm for Intelligent Device

For intelligent device, you should configure the analysis engine of device first, and then set rule, arming schedule, alarm linkage for a specific detection scene that may trigger region exiting alarm.

#### Before You Start

- Make sure you have called **NET\_DVR\_Init** to initialize the development environment.
- Make sure you have called **NET\_DVR\_Login\_V40** to log in to the device.

## Steps

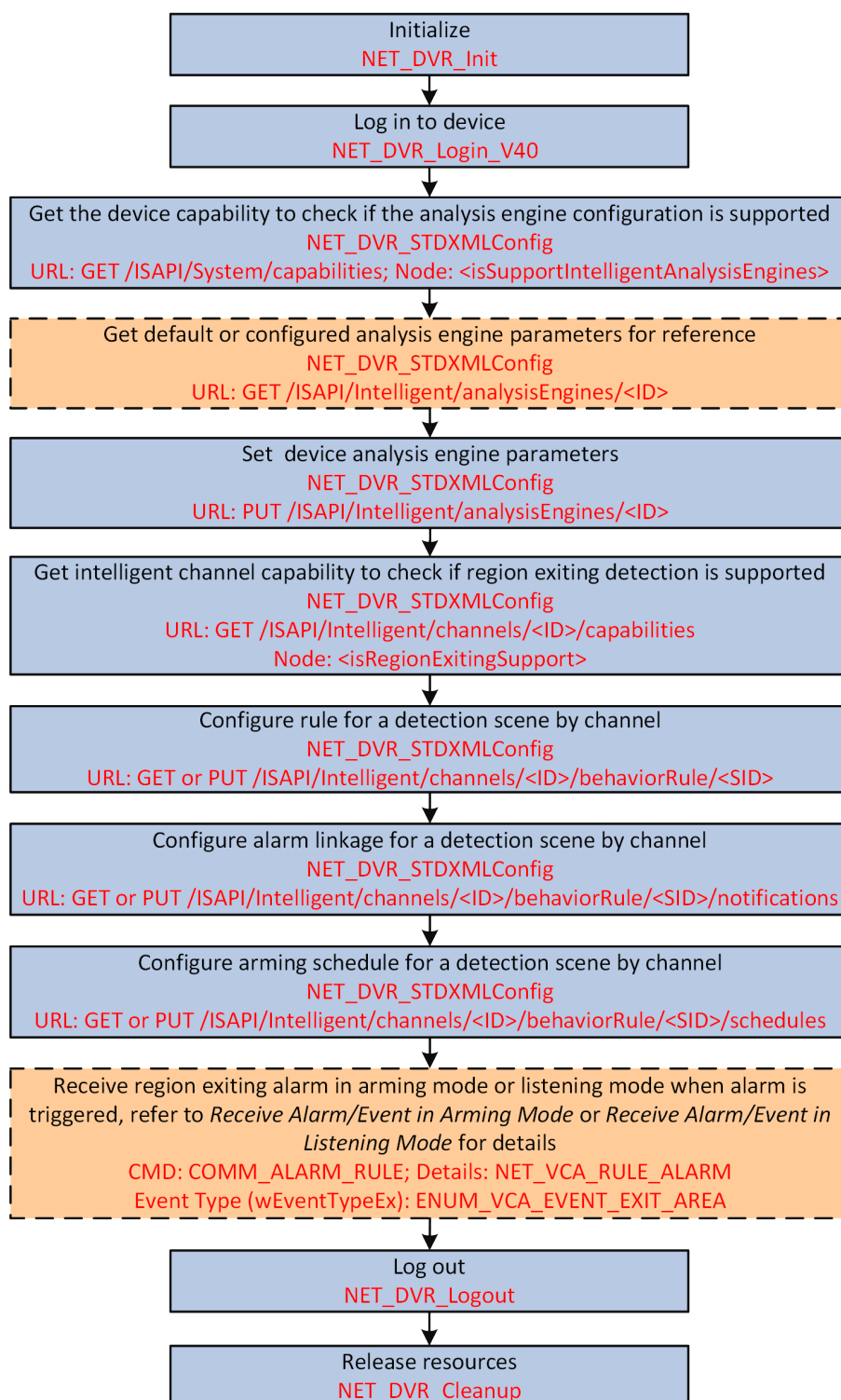


Figure 2-10 Programming Flow of Configuring Region Exiting Alarm for Intelligent Device

1. Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: GET **/ISAPI/System/capabilities** for getting the device capability to check if the analysis engine configuration is supported.

The device capability is returned in the message **XML\_DeviceCap** by output parameter pointer (**lpOutputParam**).

If the node **<isSupportIntelligentAnalysisEngines>** is returned in the message and its value is "true", it indicates that analysis engine configuration is supported, you can continue to perform this step.

Otherwise, the analysis engine configuration is not supported by device, please end this task.

2. **Optional:** Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: GET **/ISAPI/Intelligent/analysisEngines/<ID>** for getting default or configured analysis engine parameters for reference.

The analysis engine parameters is returned in the message **XML\_AnalysisEngine** by output parameter pointer (**lpOutputParam**).

3. Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: PUT **/ISAPI/Intelligent/analysisEngines/<ID>**, and set input parameter pointer (**lpInputParam**) to **XML\_AnalysisEngine** for setting the analysis engine parameters.

4. Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: GET **/ISAPI/Intelligent/channels/<ID>/capabilities** for getting intelligent channel capability to check if region exiting detection is supported.

The intelligent channel capability is returned in the message **XML\_IntelliCap** by output parameter pointer (**lpOutputParam**).

If the node **<isRegionExitingSupport>** is returned in the message and its value is "true", it indicates that region exiting detection is supported, you can continue to perform this step.

Otherwise, the region exiting detection is not supported by device, please end this task.

5. Configure region exiting detection rule for a detection scene by channel.

- 1) **Optional:** Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: GET for getting the region exiting detection rule capability.

The region exiting detection rule capability is returned in the message **XML\_Cap\_RuleInfo** by output parameter pointer (**lpOutputParam**).

- 2) **Optional:** Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: GET for getting default or configured region exiting detection rule for reference.

The region exiting detection rule parameters are returned in the message **XML\_RuleInfo** by output parameter pointer (**lpOutputParam**).

- 3) Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: PUT, and set input parameter pointer (**lpInputParam**) to **XML\_RuleInfo** for setting the region exiting detection rule of a specific detection scene.

6. Configure alarm linkage for the region exiting detection of a detection scene by channel.

- 1) **Optional:** Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: GET **/ISAPI/Intelligent/channels/<ID>/behaviorRule/<SID>/notifications** for getting default or configured region exiting alarm linkage parameters for reference.



The region exiting alarm linkage parameters are returned in the message

**XML\_RuleNotification** by output parameter pointer (**IpOutputParam**).

- 2) Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: PUT **/ISAPI/Intelligent/channels/<ID>/behaviorRule/<SID>/notifications** , and set input parameter pointer (**IpInputParam**) to **XML\_RuleNotification** for setting the region exiting alarm linkage of a specific detection scene.



### Note

To receive the alarm in the platform, the linkage action must be set to "center" (upload to center).

- 
7. Configure arming schedule for the region exiting detection of a detection scene by channel.

- 1) **Optional:** Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: GET **/ISAPI/Intelligent/channels/<ID>/behaviorRule/<SID>/schedules** for getting default or configured region exiting arming schedule for reference.

The arming schedule are returned in the message **XML\_RuleSchedule** by output parameter pointer (**IpOutputParam**).

- 2) Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: PUT **/ISAPI/Intelligent/channels/<ID>/behaviorRule/<SID>/schedules** , and set input parameter pointer (**IpInputParam**) to **XML\_RuleSchedule** for setting arming schedule of a specific detection scene.



### Note

The above rules, arming schedule, and alarm linkage can also be configured by logging in to device via web browser.

- 
1. **Optional:** Receive region exiting alarm in arming mode (see **Receive Alarm/Event in Arming Mode** ) or listening mode (see **Receive Alarm/Event in Listening Mode** ) when alarm is triggered.



### Note

- The command (**lCommand**) to receive region exiting alarms should be set to **COMM\_ALARM\_RULE** (command No.: 0x1102) and the alarm/event type (**wEventTypeEx**) to "ENUM\_VCA\_EVENT\_EXIT\_AREA" in the APIs **NET\_DVR\_SetDVRMessageCallBack\_V50** and **NET\_DVR\_StartListen\_V30** .
- For the alarm details, refer to the structure .

---

### What to do next

Call **NET\_DVR\_Logout** and **NET\_DVR\_Cleanup** to log out and release resources.

### 2.5.2 Configure Region Exiting Alarm for Smart Device

For smart device, you can call HCNetSDK APIs to set region exiting detection and alarm parameters without passing through the request URLs.

#### Before You Start

- Make sure you have called **NET\_DVR\_Init** to initialize the development environment.
- Make sure you have called **NET\_DVR\_Login\_V40** to log in to the device.

## Steps

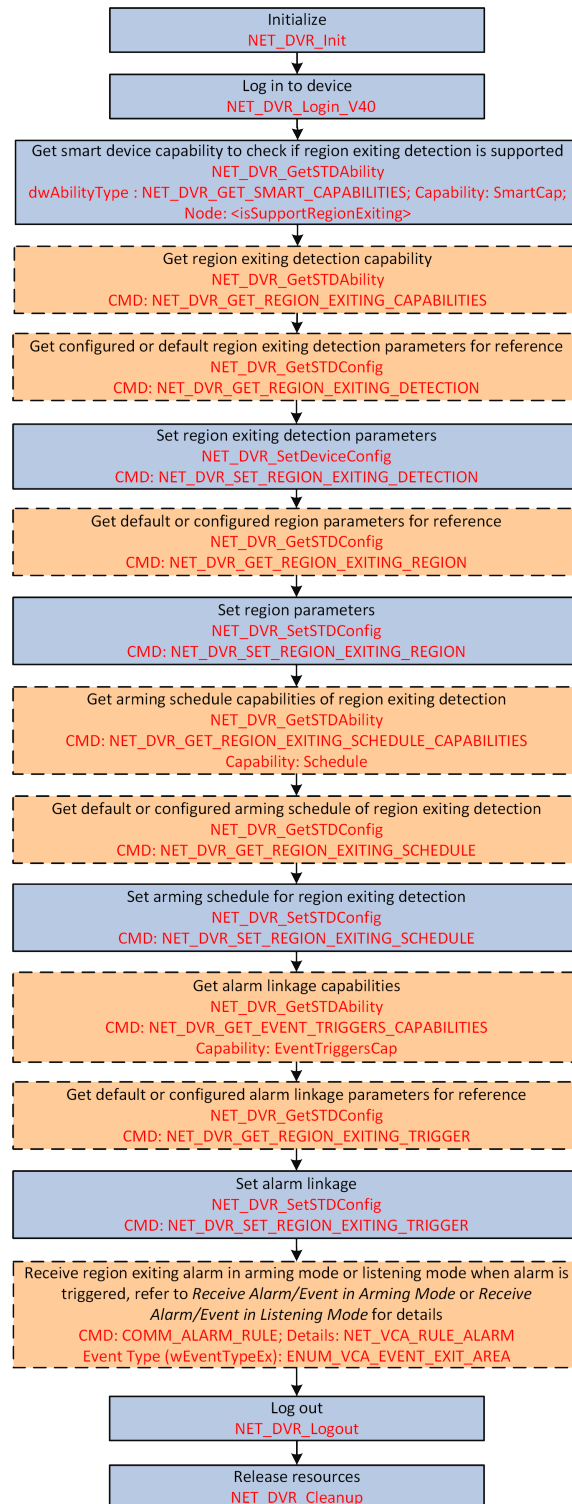


Figure 2-11 Programming Flow of Configuring Region Exiting Alarm for Smart Device

1. Call **NET\_DVR\_GetSTDAbility** , and set the capability type (**dwAbilityType**) to "NET\_DVR\_GET\_SMART\_CAPABILITIES" (value: 3500) for getting the smart device capability to check if region exiting detection is supported.  
The smart device capability is returned in the message **XML\_SmartCap** by output buffer (**lpOutBuffer**) of structure **NET\_DVR\_STD\_ABILITY** .  
If the node **<isSupportRegionExiting>** is returned in the message and its value is "true", it indicates that region exiting detection is supported, you can continue to perform this step.  
Otherwise, the region exiting detection is not supported by device, please end this task.
2. **Optional:** Call **NET\_DVR\_GetSTDAbility** , set the capability type (**dwAbilityType**) to "NET\_DVR\_GET\_REGION\_EXITING\_CAPABILITIES" (value: 3511), and set the condition buffer (**lpCondBuffer**) of structure **NET\_DVR\_STD\_ABILITY** to 4-byte channel No. for getting region exiting detection capability.  
The region exiting detection capability is returned in the message **XML\_Cap\_RegionExiting** by output buffer (**lpOutBuffer**) of structure **NET\_DVR\_STD\_ABILITY** .
3. **Optional:** Call **NET\_DVR\_GetSTDConfig** with "NET\_DVR\_GET\_REGION\_EXITING\_DETECTION" (command No.: 3512) and set condition buffer (**lpCondBuffer**) of structure **NET\_DVR\_STD\_CONFIG** to 4-byte channel No. for getting default or configured region exiting detection parameters for reference.  
The region exiting detection parameters are returned in the structure **NET\_DVR\_REGION\_EXITING\_DETECTION** by output buffer (**lpOutBuffer**) of structure **NET\_DVR\_STD\_CONFIG** .
4. Call **NET\_DVR\_SetSTDConfig** with "NET\_DVR\_SET\_REGION\_EXITING\_DETECTION" (command No.: 3513), and set condition buffer (**lpCondBuffer**) and input buffer (**lpInBuffer**) of structure **NET\_DVR\_STD\_CONFIG** to 4-byte channel No. and **NET\_DVR\_REGION\_EXITING\_DETECTION** for setting region exiting detection parameters.
5. **Optional:** Call **NET\_DVR\_GetSTDConfig** with "NET\_DVR\_GET\_REGION\_EXITING\_REGION" (command No.: 3514), and set condition buffer (**lpCondBuffer**) of structure **NET\_DVR\_STD\_ABILITY** to **NET\_DVR\_SMART\_REGION\_COND** for getting default or configured detection region parameters for reference.  
The detection region parameters are returned in the structure **NET\_DVR\_REGIONEXITING\_REGION** by output buffer (**lpOutBuffer**) of structure **NET\_DVR\_STD\_ABILITY** .
6. Call **NET\_DVR\_SetSTDConfig** with "NET\_DVR\_SET\_REGION\_EXITING\_REGION" (command No.: 3515), and set condition buffer (**lpCondBuffer**) and input buffer (**lpInBuffer**) of structure **NET\_DVR\_STD\_ABILITY** to **NET\_DVR\_SMART\_REGION\_COND** and **NET\_DVR\_REGIONEXITING\_REGION** for setting detection region parameters.
7. **Optional:** Call **NET\_DVR\_GetSTDAbility** , set the capability type (**dwAbilityType**) to "NET\_DVR\_GET\_REGION\_EXITING\_SCHEDULE\_CAPABILITIES" (value: 3585), and set the condition buffer (**lpCondBuffer**) of structure **NET\_DVR\_STD\_ABILITY** to 4-byte channel No. for getting arming schedule capability of region exiting detection.  
The arming schedule capability of region exiting detection is returned in the message **XML\_Cap\_Schedule** by output buffer (**lpOutBuffer**) of structure **NET\_DVR\_STD\_ABILITY** .

8. **Optional:** Call **NET\_DVR\_GetSTDConfig** with "NET\_DVR\_GET\_REGION\_EXIT\_SCHEDULE" (command No.: 3518), and set condition buffer (**IpCondBuffer**) of structure **NET\_DVR\_STD\_ABILITY** to 4-byte channel No. for getting default or configured arming schedule parameters of region exiting detection for reference.  
The arming schedule parameters of region exiting detection are returned in the structure **NET\_DVR\_EVENT\_SCHEDULE** by output buffer (**IpOutBuffer**) of structure **NET\_DVR\_STD\_ABILITY**.
9. Call **NET\_DVR\_SetSTDConfig** with "NET\_DVR\_SET\_REGION\_EXIT\_SCHEDULE" (command No.: 3519), and set condition buffer (**IpCondBuffer**) and input buffer (**IpInBuffer**) of structure **NET\_DVR\_STD\_ABILITY** to 4-byte channel No. and **NET\_DVR\_EVENT\_SCHEDULE** for setting arming schedule of region exiting detection.
10. **Optional:** Call **NET\_DVR\_GetSTDAbility**, and set the capability type (**dwAbilityType**) to "NET\_DVR\_GET\_EVENT\_TRIGGERS\_CAPABILITIES" (value: 3501) for getting alarm linkage capability.  
The alarm linkage capability is returned in the message **XML\_EventTriggersCap** by output buffer (**IpOutBuffer**) of structure **NET\_DVR\_STD\_ABILITY**.
11. **Optional:** Call **NET\_DVR\_GetSTDConfig** with "NET\_DVR\_GET\_REGION\_EXIT\_TRIGGER" (command No.: 3516), and set condition buffer (**IpCondBuffer**) of structure **NET\_DVR\_STD\_ABILITY** to 4-byte channel No. for getting default or configured alarm linkage parameters for reference.  
The alarm linkage parameters are returned in the structure **NET\_DVR\_EVENT\_TRIGGER** by output buffer (**IpOutBuffer**) of structure **NET\_DVR\_STD\_ABILITY**.
12. Call **NET\_DVR\_SetSTDConfig** with "NET\_DVR\_SET\_REGION\_EXIT\_TRIGGER" (command No.: 3517), and set condition buffer (**IpCondBuffer**) and input buffer (**IpInBuffer**) of structure **NET\_DVR\_STD\_ABILITY** to 4-byte channel No. and **NET\_DVR\_EVENT\_TRIGGER**, for setting alarm linkage.



### Note

- To receive the alarm in the platform, the linkage action must be set to "center" (upload to center).
- The above region exiting detection parameters can also be configured by logging in to device via web browser.

- 
13. **Optional:** Receive region exiting alarm in arming mode (see **Receive Alarm/Event in Arming Mode**) or listening mode (see **Receive Alarm/Event in Listening Mode**) when alarm is triggered.



### Note

- The command (**lCommand**) to receive line crossing alarms should be set to "COMM\_ALARM\_RULE" (command No.: 0x1102) and the alarm/event type (**wEventTypeEx**) to "ENUM\_VCA\_EVENT\_EXIT\_AREA" in the APIs **NET\_DVR\_SetDVRMessageCallBack\_V50** and **NET\_DVR\_StartListen\_V30**.
  - For the alarm details, refer to the structure **NET\_VCA\_RULE\_ALARM**.
-

### What to do next

Call **NET\_DVR\_Logout** and **NET\_DVR\_Cleanup** to log out and release resources.

## 2.6 Configure Hard Hat Detection Alarm

Hard hat detection function detects people who are not wearing hard hats in a specific region. You can configure the arming schedule and linkage actions to be taken when the hard hat detection alarms are triggered. For people who are not wearing hard hat, you can also link face picture libraries with the hard hat detection to recognize these people and know their names.

### Before You Start

- Make sure you have called **NET\_DVR\_Init** to initialize the development environment.
- Make sure you have called **NET\_DVR\_Login\_V40** to log in to the device.
- Make sure you have set the analysis engine mode to hard hat detection.

## Steps

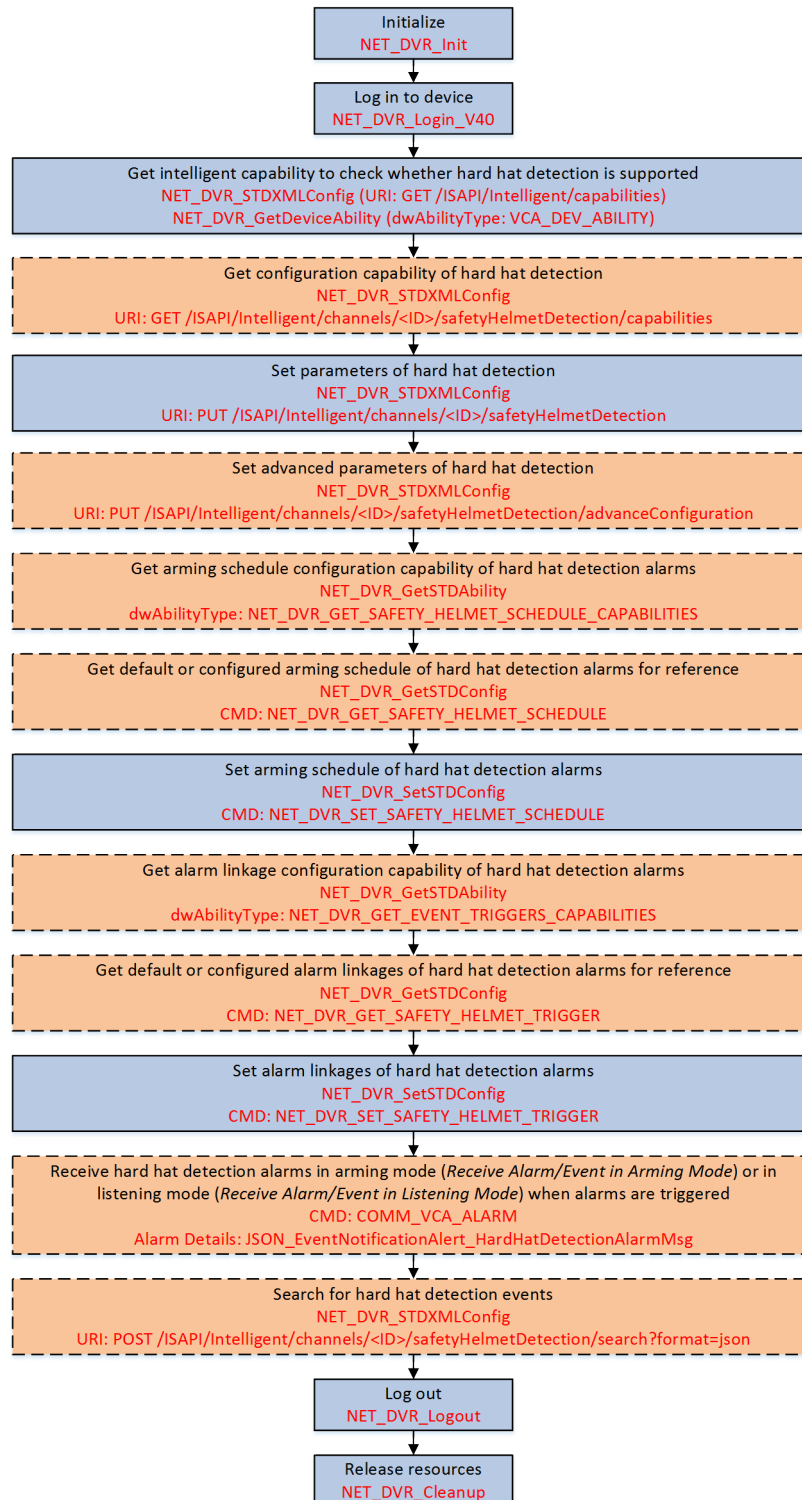


Figure 2-12 Programming Flow of Configuring Hard Hat Detection Alarm

1. Perform one of the following operations to get the intelligent capability of the device and check whether the hard hat detection is supported by the device.

- Call **NET\_DVR\_STDXMLConfig** to transmit the request URI: GET /ISAPI/Intelligent/capabilities . The intelligent capability is returned in the message XML\_IntelliCap by **lpOutputParam**.

If the node **<isSupportSafetyHelmetDetection>** is returned and its value is "true", it indicates that the hard hat detection is supported, and then you can continue to perform the following steps. Otherwise, the hard hat detection is not supported by the device, please end this task.

- Call **NET\_DVR\_GetDeviceAbility** , set the capability type (**dwAbilityType**) to "VCA\_DEV\_ABILITY" (macro definition value: 0x100) to get the intelligent device capability.

The intelligent device capability is returned in the structure **NET\_VCA\_DEV\_ABILITY** by **pOutBuf**, and the related member is **bySafetyHelmetDetection**.

2. Optional: Call **NET\_DVR\_STDXMLConfig** to transmit the request URI: GET /ISAPI/Intelligent/channels/<ID>/safetyHelmetDetection/capabilities for getting the configuration capability of hard hat detection and knowing the configuration details or notices.

The configuration capability is returned in the message XML\_SafetyHelmetDetectionCap by **lpOutputParam**.

3. Call **NET\_DVR\_STDXMLConfig** to transmit the request URI: PUT /ISAPI/Intelligent/channels/<ID>/safetyHelmetDetection and set **lpInputParam** to XML\_SafetyHelmetDetection for setting the parameters of hard hat detection.

---

### Note

Before setting parameters, you can get the default or configured parameters for reference by calling **NET\_DVR\_STDXMLConfig** to transmit the request URI: GET /ISAPI/Intelligent/channels/<ID>/safetyHelmetDetection . And the parameters are returned in the message XML\_SafetyHelmetDetection by **lpOutputParam**.

4. Optional: Call **NET\_DVR\_STDXMLConfig** to transmit the request URI: PUT /ISAPI/Intelligent/channels/<ID>/safetyHelmetDetection/advanceConfiguration and set **lpInputParam** to XML\_AdvanceConfiguration for setting advanced parameters of hard hat detection.

---

### Note

- To check whether setting advanced parameters is supported, you can check whether the node **<AdvanceConfiguration>** exists in the configuration capability message of hard hat detection XML\_SafetyHelmetDetectionCap .
- Before setting parameters, you can get the default or configured parameters for reference by calling **NET\_DVR\_STDXMLConfig** to transmit the request URI: GET /ISAPI/Intelligent/channels/<ID>/safetyHelmetDetection/advanceConfiguration . And the parameters are returned in the message XML\_AdvanceConfiguration by **lpOutputParam**.

5. Optional: Call **NET\_DVR\_GetSTDAbility** , set the **dwAbilityType** to "NET\_DVR\_GET\_SAFETY\_HELMET\_SCHEDULE\_CAPABILITIES" (value: 3988), and set condition parameter **lpCondBuffer** in the structure **NET\_DVR\_STD\_ABILITY** to 4-byte channel No. for getting the arming schedule configuration capability of hard hat detection alarms.



The arming schedule configuration capability ( *XML Cap Schedule* ) is returned by the output parameter **IpOutBuffer** in the structure *NET\_DVR\_STD\_ABILITY* .

6. **Optional:** Call *NET\_DVR\_GetSTDConfig* with "NET\_DVR\_GET\_SAFETY\_HELMET\_SCHEDULE" (command No.: 3989) and set the condition parameter **IpCondBuffer** in the structure *NET\_DVR\_STD\_CONFIG* to 4-byte channel No. for getting the default or configured arming schedule parameters for reference.

The arming schedule parameters ( *NET\_DVR\_EVENT\_SCHEDULE* ) are returned by the output parameter **IpOutBuffer** in the structure *NET\_DVR\_STD\_CONFIG* .

7. Call *NET\_DVR\_SetSTDConfig* with "NET\_DVR\_SET\_SAFETY\_HELMET\_SCHEDULE" (command No.: 3990), set the condition parameter **IpCondBuffer** and input parameter **IpInBuffer** in the structure *NET\_DVR\_STD\_CONFIG* to 4-byte channel No. and *NET\_DVR\_EVENT\_SCHEDULE* respectively for setting the arming schedule.

8. **Optional:** Call *NET\_DVR\_GetSTDAbility* , set the **dwAbilityType** to "NET\_DVR\_GET\_EVENT\_TRIGGERS\_CAPABILITIES" (value: 3501), and set condition parameter **IpCondBuffer** in the structure *NET\_DVR\_STD\_ABILITY* to NULL for getting the alarm linkage configuration capability of hard hat detection alarms.

The alarm linkage configuration capability ( *XML EventTriggersCap* ) is returned by the output parameter **IpOutBuffer** in the structure *NET\_DVR\_STD\_ABILITY* .

9. **Optional:** Call *NET\_DVR\_GetSTDConfig* with "NET\_DVR\_GET\_SAFETY\_HELMET\_TRIGGER" (command No.: 3986) and set the condition parameter **IpCondBuffer** in the structure *NET\_DVR\_STD\_CONFIG* to the channel No. for getting the default or configured alarm linkage parameters for reference.

The alarm linkage parameters ( *NET\_DVR\_EVENT\_TRIGGER* ) are returned by the output parameter **IpOutBuffer** in the structure *NET\_DVR\_STD\_CONFIG* .

10. Call *NET\_DVR\_SetSTDConfig* with "NET\_DVR\_SET\_SAFETY\_HELMET\_TRIGGER" (command No.: 3987), set the condition parameter **IpCondBuffer** and input parameter **IpInBuffer** in the structure *NET\_DVR\_STD\_CONFIG* to the channel No. and *NET\_DVR\_EVENT\_TRIGGER* respectively for setting alarm linkages.



### Note

The above detection parameters, arming schedule, and alarm linkages can be configured by logging in to the device via web browser.

11. **Optional:** Set the uploading message type (**ICommand**) in the alarm callback function ( *MSGCallback* ) to "COMM\_VCA\_ALARM" (macro definition value: 0x4993) for receiving hard hat detection alarms in arming mode (see *Receive Alarm/Event in Arming Mode* for details) or listening mode (see *Receive Alarm/Event in Listening Mode* for details).

The hard hat detection alarms are uploaded in the message

*JSON\_EventNotificationAlert\_safetyHelmetDetection* .

12. **Optional:** Call *NET\_DVR\_STDXMLConfig* to transmit the request URI: POST */ISAPI/Intelligent/channels/<ID>/safetyHelmetDetection/search?format=json* and set **IpInputParam** to *JSON\_SearchDescription* for searching for hard hat detection events.



### Note

To check whether searching for hard hat detection events is supported, you can check whether the node **<SearchDescriptionCap>** exists in the configuration capability message of hard hat detection **XML\_SafetyHelmetDetectionCap**.

---

The search results are returned in the message **JSON\_SearchResult** by **IpOutputParam**.

### What to do next

Call **NET\_DVR\_Logout** and **NET\_DVR\_Cleanup** to log out of the device and release resources.

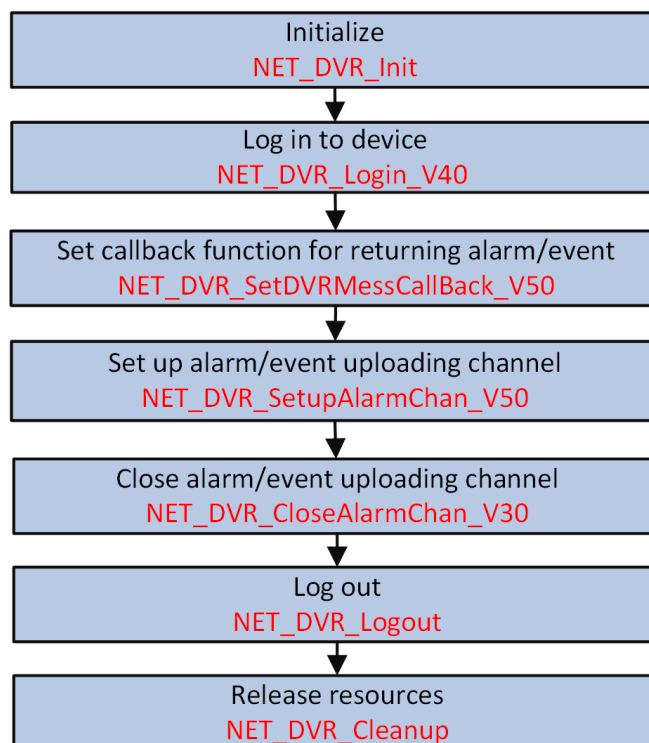
## 2.7 Receive Alarm/Event in Arming Mode

When the alarm is triggered or the event occurred, the secondarily developed third-party platform can automatically connect and send alarm/event uploading command to the device, and then the device uploads the alarm/event information to the platform for receiving.

### Before You Start

- Make sure you have called **NET\_DVR\_Init** to initialize the development environment.
- Make sure you have called **NET\_DVR\_Login\_V40** to log in to the device.
- Make sure you have configured the alarm/event parameters, refer to the typical alarm/event configurations for details.

## Steps



**Figure 2-13 Programming Flow of Receiving Alarm/Event in Arming Mode**

1. Call **NET\_DVR\_SetDVRMessageCallBack\_V50** to set callback function for returning alarm/event information.



### Note

- If the configured alarm is triggered or event occurred, the alarm/event information will be uploaded by device and returned in the callback function. You can view the alarm/event and do some processing operations.
- For the integration via device network SDK (HCNetSDK), to receive different types of alarm/event information, the parameter **lCommand** (data type to be uploaded) in the configured callback function should be different (refer to the typical alarm/event configurations). For the integration via text protocol, the **lCommand** should be set to "COMM\_ISAPI\_ALARM" (command No.: 0x6009) and the input parameter **pAlarmInfo** in the callback function **MSGCallBack** should be set to **NET\_DVR\_ALARM\_ISAPI\_INFO**.

2. Call **NET\_DVR\_SetupAlarmChan\_V50** to set up uploading channel.
3. Call **NET\_DVR\_CloseAlarmChan\_V30** to close uploading channel and stop receiving alarm or event information.

## Example

Sample Code of Receiving Alarm or Event in Arming Mode

```
#include <stdio.h>
#include <iostream>
#include "Windows.h"
#include "HCNetSDK.h"
using namespace std;

void main() {
    //-----
    // Initialize
    NET_DVR_Init();
    //Set connection time and reconnection time
    NET_DVR_SetConnectTime(2000, 1);
    NET_DVR_SetReconnect(10000, true);
    //-----
    // Log in to device
    LONG lUserID;
    //Login parameters, including device IP address, user name, password, and so
    on.
    NET_DVR_USER_LOGIN_INFO struLoginInfo = {0};
    struLoginInfo.bUseAsynLogin = 0; //Synchronous login mode
    strcpy(struLoginInfo.sDeviceAddress, "192.0.0.64"); //Device IP address
    struLoginInfo.wPort = 8000; //Service port No.
    strcpy(struLoginInfo.sUserName, "admin"); //User name
    strcpy(struLoginInfo.sPassword, "abcd1234"); //Password
    //Device information, output parameter
    NET_DVR_DEVICEINFO_V40 struDeviceInfoV40 = {0};
    lUserID = NET_DVR_Login_V40(&struLoginInfo, &struDeviceInfoV40);
    if (lUserID < 0)
    {
        printf("Login failed, error code: %d\n", NET_DVR_GetLastError());
        NET_DVR_Cleanup();
        return;
    }

    //Set alarm callback function
    NET_DVR_SetDVRMessageCallBack_V50(0, MessageCallbackNo1, NULL);
    NET_DVR_SetDVRMessageCallBack_V50(1, MessageCallbackNo2, NULL);

    //Enable arming
    NET_DVR_SETUPALARM_PARAM_V50 struSetupParamV50={0};
    struSetupParamV50.dwSize=sizeof(NET_DVR_SETUPALARM_PARAM_V50);
    //Alarm category to be uploaded
    struSetupParamV50.byAlarmInfoType=1;
    //Arming level
    struSetupParamV50.byLevel=1;

    char szSubscribe[1024] = {0};
    //The following code is for alarm subscription (subscribe all)
    memcpy(szSubscribe, "<SubscribeEvent version=\"2.0\" xmlns=\"http://
www.isapi.org/ver20/XMLSchema\">\r\n<eventMode>all</eventMode>\r\n", 1024);
    LONG lHandle = -1;
    if (0 == strlen(szSubscribe))
```

```
{
    //Arm
    lHandle = NET_DVR_SetupAlarmChan_V50(lUserID, &struSetupParamV50, NULL,
strlen(szSubscribe));
}
else
{
    //Subscribe
    LlHandle = NET_DVR_SetupAlarmChan_V50(lUserID, &struSetupParamV50,
szSubscribe, strlen(szSubscribe));
}

if (lHandle < 0)
{
    printf("NET_DVR_SetupAlarmChan_V50 error, %d\n", NET_DVR_GetLastError());
    NET_DVR_Logout(lUserID);
    NET_DVR_Cleanup();
    return;
}

Sleep(20000);
//Disarm the uploading channel
if (!NET_DVR_CloseAlarmChan_V30(lHandle))
{
    printf("NET_DVR_CloseAlarmChan_V30 error, %d\n", NET_DVR_GetLastError());
    NET_DVR_Logout(lUserID);
    NET_DVR_Cleanup();
    return;
}

//Log out
NET_DVR_Logout(lUserID);
//Release resources
NET_DVR_Cleanup();
return;
}
```

### What to do next

Call **NET\_DVR\_Logout** and **NET\_DVR\_Cleanup** to log out and release resources.

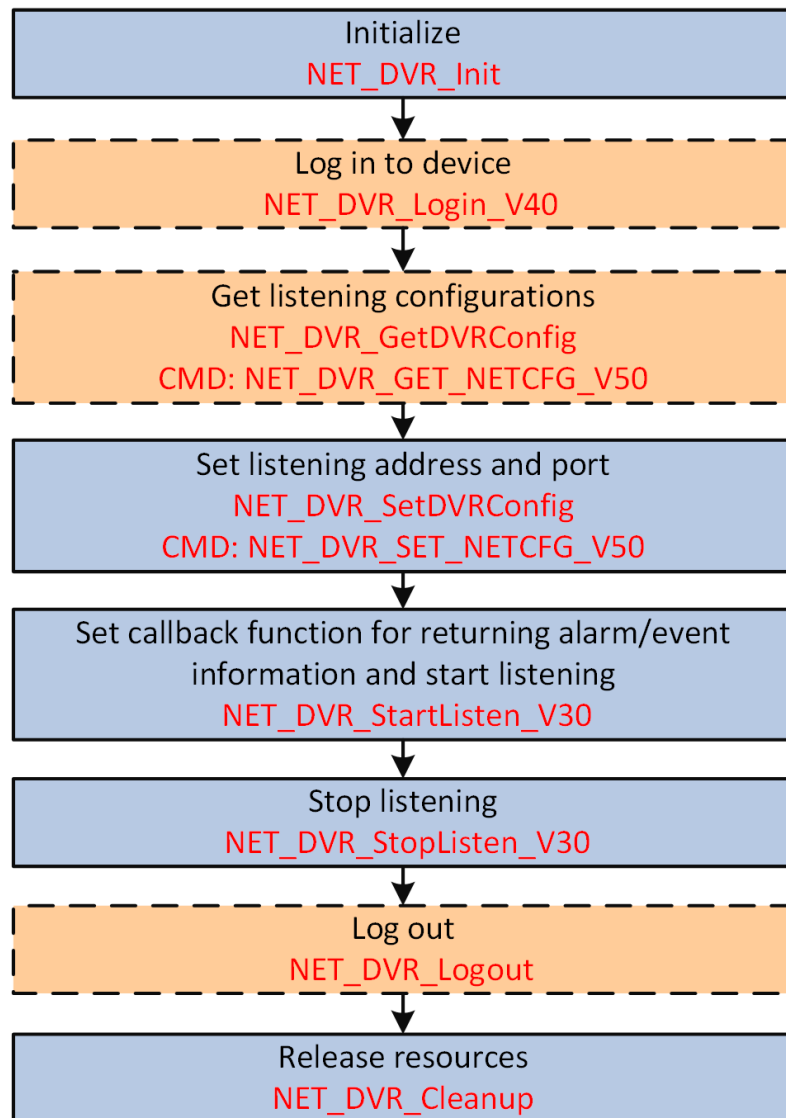
## 2.8 Receive Alarm/Event in Listening Mode

When alarm is triggered or event occurred, the device uploads the alarm/event information automatically, so you can configure the listening address and port for listening and receiving the alarm/event in the secondarily developed third-part platform.

### Before You Start

- Make sure you have called **NET\_DVR\_Init** to initialize the development environment.
- Make sure you have configured the alarm/event parameters, refer to the typical alarm/event configurations for details.

### Steps



**Figure 2-14 Programming Flow of Receiving Alarm/Event in Listening Mode**

1. **Optional:** Call **NET\_DVR\_Login\_V40** to log in to device.
2. **Optional:** Call **NET\_DVR\_GetDVRConfig** with "NET\_DVR\_GET\_NETCFG\_V50" (command No.: 1015) to get the existing listening configurations (i.e., listening address and port) for reference. The listening parameters are retruned in the structure **NET\_DVR\_NETCFG\_V50** by the output parameter pointer **lpOutBuffer**.

3. Call **NET\_DVR\_SetDVRConfig** with "NET\_DVR\_SET\_NETCFG\_V50" (command No.: 1016) and specify the input parameter pointer **lpInBuffer** to the structure **NET\_DVR\_NETCFG\_V50** for setting the listening address and port.
4. Call **NET\_DVR\_StartListen\_V30** to set callback function for returning alarm/event information and start the listening.

---

### Note

For the integration via device network SDK (HCNetSDK), to receive different types of alarm/event information, the parameter **lCommand** (data type to be uploaded) in the configured callback function should be different (refer to the typical alarm/event configurations). For the integration via text protocol, the **lCommand** should be set to "COMM\_ISAPI\_ALARM" and the input parameter **pAlarmInfo** in the callback function **MSGCallback** should be set to **NET\_DVR\_ALARM\_ISAPI\_INFO**.

---

The alarm/event information is automatically uploaded by the device when the configured alarm is triggered or event occurred, and the third-party platform or system gets the alarm/event information from the configured callback function.

5. Call **NET\_DVR\_StopListen\_V30** to stop listening and receiving alarm or event information.

### Example

#### Sample Code of Receiving Alarm/Event in Listening Mode

```
#include <stdio.h>
#include <iostream>
#include "Windows.h"
#include "HCNetSDK.h"
using namespace std;
void main() {
    //-----
    // Initialize
    NET_DVR_Init();
    //Set connection time and reconnection time
    NET_DVR_SetConnectTime(2000, 1);
    NET_DVR_SetReconnect(10000, true);
    //-----
    // Log in to device
    LONG lUserID;
    NET_DVR_DEVICEINFO_V30 struDeviceInfo;
    lUserID = NET_DVR_Login_V30("172.0.0.100", 8000, "admin", "12345",
    &struDeviceInfo);
    if (lUserID < 0)
    {
        printf("Login error, %d\n", NET_DVR_GetLastError());
        NET_DVR_Cleanup();
        return;
    }
    //Enable listening
    LONG lHandle;
    lHandle = NET_DVR_StartListen_V30(NULL, 7200, MessageCallback, NULL);
    if (lHandle < 0)
```

```
{
    printf("NET_DVR_StartListen_V30 error, %d\n", NET_DVR_GetLastError());
    NET_DVR_Logout(lUserID);
    NET_DVR_Cleanup();
    return;
}
Sleep(5000);
//Disable listening
if (!NET_DVR_StopListen_V30(lHandle))
{
    printf("NET_DVR_StopListen_V30 error, %d\n", NET_DVR_GetLastError());
    NET_DVR_Logout(lUserID);
    NET_DVR_Cleanup();
    return;
}
//Log out
NET_DVR_Logout(lUserID);
//Release SDK resource
NET_DVR_Cleanup();
return;
}
```

### What to do next

Call **NET\_DVR\_Logout** (if logged in) and **NET\_DVR\_Cleanup** to log out and release resources.



## Chapter 3 Other Settings

Table 3-1 Auto-tracking Settings

Get auto-tracking configuration capability by channel	Call <b><u>NET_DVR_STDXMLConfig</u></b> to transmit <b><u>/ISAPI/Intelligent/channels/&lt;ID&gt;/intelliTrace/capabilities</u></b> by GET method. The capability is returned in the message <b><u>XML_Intelli_IntelliTraceCap</u></b> by <b>lpOutputParam</b> .
Get auto-tracking parameters of all scenes	Call <b><u>NET_DVR_STDXMLConfig</u></b> to transmit <b><u>/ISAPI/Intelligent/channels/&lt;ID&gt;/intelliTrace</u></b> by GET method. The parameters are returned in the message <b><u>XML_IntelliTraceBlockList</u></b> by <b>lpOutputParam</b>
Set auto-tracking parameters of all scenes	Call <b><u>NET_DVR_STDXMLConfig</u></b> to transmit <b><u>/ISAPI/Intelligent/channels/&lt;ID&gt;/intelliTrace</u></b> by PUT method and set <b>lpInputParam</b> to <b><u>XML_IntelliTraceBlockList</u></b>
Get auto-tracking parameters of a single scene	Call <b><u>NET_DVR_STDXMLConfig</u></b> to transmit <b><u>/ISAPI/Intelligent/channels/&lt;ID&gt;/intelliTrace/&lt;SID&gt;</u></b> by GET method. The parameters are returned in the message <b><u>XML_IntelliTraceBlock</u></b> by <b>lpOutputParam</b>
Set auto-tracking parameters of a single scene	Call <b><u>NET_DVR_STDXMLConfig</u></b> to transmit <b><u>/ISAPI/Intelligent/channels/&lt;ID&gt;/intelliTrace/&lt;SID&gt;</u></b> by PUT method and set <b>lpInputParam</b> to <b><u>XML_IntelliTraceBlock</u></b>
Delete auto-tracking parameters of a single scene	Call <b><u>NET_DVR_STDXMLConfig</u></b> to transmit <b><u>/ISAPI/Intelligent/channels/&lt;ID&gt;/intelliTrace/&lt;SID&gt;</u></b> by DELETE method

## Chapter 4 API Reference

### 4.1 NET\_DVR\_Cleanup

Release the resources after the program is ended.

#### API Definition

```
BOOL NET_DVR_Cleanup(  
);
```

#### 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 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.

### 4.2 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.

## **4.3 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**

## **4.4 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.

## 4.5 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.

## 4.6 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.

## 4.7 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\*\*](#)

## 4.8 NET\_DVR\_Init

Initialize the programming environment before calling other APIs.

### API Definition

```
BOOL NET_DVR_Init(  
);
```

### 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 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\*\*](#)

## 4.9 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**

#### 4.9.1 fLoginResultCallBack

#### Login Status Callback Function

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

#### 4.10 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](#).

## 4.11 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**

## **4.12 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 4-1 NET\_SDK\_INIT\_CFG\_TYPE**

enumType	Value	Description	IpInBuff
NET_SDK_INIT_CFG_ABILITY	1	Capability supported by SDK.	<b><u>NET_DVR_INIT_CFG_ABILITY</u></b>
NET_SDK_INIT_CFG_SDK_PATH	2	Set loading path for component libraries (supported by both Linux and Windows system).	<b><u>NET_DVR_LOCAL_SDK_PATH</u></b>
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., <b><i>C:\libeay32.dll</i></b> .
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., <b><i>C:\ssleay32.dll</i></b> .

#### IpInBuff

[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.

## 4.13 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](#)

## 4.14 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 **lpInputParam** and output parameter **lpOutputParam** 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
<b>lpInputParam</b>	<b>lpRequestUrl</b> (see in structure <u><a href="#">NET_DVR_XML_CONFIG_INPUT</a></u> )	Method+URL E.g., GET /ISAPI/System/capabilities
	<b>lpInBuffer</b> (see in structure <u><a href="#">NET_DVR_XML_CONFIG_INPUT</a></u> )	Request Message
<b>lpOutputParam</b>	<b>lpOutBuffer</b> (see in structure <u><a href="#">NET_DVR_XML_CONFIG_OUTPUT</a></u> )	Response Message
	<b>lpStatusBuffer</b> (see in structure <u><a href="#">NET_DVR_XML_CONFIG_OUTPUT</a></u> )	Response Message

## 4.15 NET\_DVR\_CloseAlarmChan\_V30

Close alarm uploading channel.

### API Definition

```
BOOL NET_DVR_CloseAlarmChan_V30(  
    LONG    lAlarmHandle  
);
```

### Parameters

#### lAlarmHandle

Value returned by [NET\\_DVR\\_SetupAlarmChan\\_V50](#) .

### Return Values

Return *TURE* for success, and return *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, 3, 6, 12, 17, 41, and 47. See details in the [Device Network SDK Errors](#) .

## 4.16 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***

## **4.17 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.

### IChannel

[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**

## 4.18 NET\_DVR\_SetDVRMessageCallback\_V50

Set callback functions for getting the video data.

### API Definition

```
BOOL NET_DVR_SetDVRMessageCallback_V50(  
    int             iIndex,  
    MSGCallback     fMessageCallback,  
    void            *pUser  
);
```

### Parameters

#### iIndex

[IN] Callback function index No., which ranges from 0 to 15.

#### fMessageCallback

[IN] Callback function, see details in **MSGCallback**.

#### pUser

[IN] User data.

### Return Values

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

If *FALSE* returned, call **NET\_DVR\_GetLastError** to get the error code.



### Remarks

- This API supports setting multiple callback functions for different channels (up to 16 channels are supported) at same time, and the configured callback functions are distinguished by the index No.
- All alarm/event information will be returned in each configured callback function, and you can distinguish the devices via the **pAlarmInfo** in the callback function ( [\*MSGCallback\*](#) ).

### Example

Sample Code of Setting Multiple Callback Functions to Receive Different Alarms/Events in Arming Mode

```
#include <stdio.h>
#include <iostream>
#include "Windows.h"
#include "HCNetSDK.h"
using namespace std;

int iNum=0;
void CALLBACK MessageCallbackNo1(LONG lCommand, NET_DVR_ALARMER *pAlarmer, char
*pAlarmInfo, DWORD dwBufLen, void* pUser)
{
    int i=0;
    char filename[100];
    FILE *fSnapPic=NULL;
    FILE *fSnapPicPlate=NULL;

    //This sample code is for reference only. Actually, it is not recommended
    to process the data and save file in the callback function directly.
    //You'd better process the data in the message response function via message
    mode (PostMessage).

    switch(lCommand)
    {
        case COMM_ALARM:
        {
            NET_DVR_ALARMINFO struAlarmInfo;
            memcpy(&struAlarmInfo, pAlarmInfo, sizeof(NET_DVR_ALARMINFO));
            switch (struAlarmInfo.dwAlarmType)
            {
                case 3: //Motion detection alarm
                    for (i=0; i<16; i++)    //define MAX_CHANNUM    16    //The
maximum number of channels
                    {
                        if (struAlarmInfo.dwChannel[i] == 1)
                        {
                            printf("Channel Number with Motion Detection Alarm
%d\n", i+1);
                        }
                    }
                    break;
            }
        }
    }
}
```

```
        default:
            break;
    }
    break;
}
case COMM_UPLOAD_PLATE_RESULT:
{
    NET_DVR_PLATE_RESULT struPlateResult={0};
    memcpy(&struPlateResult, pAlarmInfo, sizeof(struPlateResult));
    printf("License Plate Number: %s\n",
struPlateResult.struPlateInfo.sLicense); //License plate number

    switch(struPlateResult.struPlateInfo.byColor) //License plate color
    {
    case VCA_BLUE_PLATE:
        printf("Vehicle Color: Blue\n");
        break;
    case VCA_YELLOW_PLATE:
        printf("Vehicle Color: Yellow\n");
        break;
    case VCA_WHITE_PLATE:
        printf("Vehicle Color: White\n");
        break;
    case VCA_BLACK_PLATE:
        printf("Vehicle Color: Black\n");
        break;
    default:
        break;
    }
    //Scene picture
    if (struPlateResult.dwPicLen != 0 && struPlateResult.byResultType
== 1 )
    {
        sprintf(filename, "testpic_%d.jpg", iNum);
        fSnapPic=fopen(filename, "wb");
        fwrite(struPlateResult.pBuffer1, struPlateResult.dwPicLen,
1, fSnapPic);
        iNum++;
        fclose(fSnapPic);
    }
    //License plate picture
    if (struPlateResult.dwPicPlateLen != 0 &&
struPlateResult.byResultType == 1)
    {
        sprintf(filename, "testPicPlate_%d.jpg", iNum);
        fSnapPicPlate=fopen(filename, "wb");
        fwrite(struPlateResult.pBuffer1, struPlateResult.dwPicLen,
1, fSnapPicPlate);
        iNum++;
        fclose(fSnapPicPlate);
    }
    //Processing other data...
```

```
        break;
    }
    case COMM_ITS_PLATE_RESULT:
    {
        NET_ITS_PLATE_RESULT struITSPlateResult={0};
        memcpy(&struITSPlateResult, pAlarmInfo, sizeof(struITSPlateResult));

        for (i=0;i<struITSPlateResult.dwPicNum;i++)
        {
            printf("License Plate Number: %s\n",
struITSPlateResult.struPlateInfo.sLicense);//License plate number
            switch(struITSPlateResult.struPlateInfo.byColor)//License plate
color
            {
                case VCA_BLUE_PLATE:
                    printf("Vehicle Color: Blue\n");
                    break;
                case VCA_YELLOW_PLATE:
                    printf("Vehicle Color: Yellow\n");
                    break;
                case VCA_WHITE_PLATE:
                    printf("Vehicle Color: White\n");
                    break;
                case VCA_BLACK_PLATE:
                    printf("Vehicle Color: Black\n");
                    break;
                default:
                    break;
            }
            //Save scene picture
            if ((struITSPlateResult.struPicInfo[i].dwDataLen !=
0)&&(struITSPlateResult.struPicInfo[i].byType== 1)||
(struITSPlateResult.struPicInfo[i].byType == 2))
            {
                sprintf(filename,"testITSpic%d_%d.jpg",iNum,i);
                fSnapPic=fopen(filename,"wb");
                fwrite(struITSPlateResult.struPicInfo[i].pBuffer,
struITSPlateResult.struPicInfo[i].dwDataLen,1,fSnapPic);
                iNum++;
                fclose(fSnapPic);
            }
            //License plate thumbnails
            if ((struITSPlateResult.struPicInfo[i].dwDataLen !=
0)&&(struITSPlateResult.struPicInfo[i].byType == 0))
            {
                sprintf(filename,"testPicPlate%d_%d.jpg",iNum,i);
                fSnapPicPlate=fopen(filename,"wb");
                fwrite(struITSPlateResult.struPicInfo[i].pBuffer,
struITSPlateResult.struPicInfo[i].dwDataLen, 1, \ fSnapPicPlate);
                iNum++;
                fclose(fSnapPicPlate);
            }
        }
    }
}
```

```
        //Processing other data...
    }
    break;
}
default:
    break;
}
}

void CALLBACK MessageCallbackNo2(LONG lCommand, NET_DVR_ALARMER *pAlarmer, char
*pAlarmInfo, DWORD dwBufLen, void* pUser)
{
    int i=0;
    char filename[100];
    FILE *fSnapPic=NULL;
    FILE *fSnapPicPlate=NULL;

    //This sample code is for reference only. Actually, it is not recommended
    to process the data and save file in the callback function directly.
    //You'd better process the data in the message response function via message
    mode (PostMessage).

    switch(lCommand)
    {
        case COMM_ALARM:
        {
            NET_DVR_ALARMINFO struAlarmInfo;
            memcpy(&struAlarmInfo, pAlarmInfo, sizeof(NET_DVR_ALARMINFO));
            switch (struAlarmInfo.dwAlarmType)
            {
                case 3: //Motion detection alarm
                    for (i=0; i<16; i++)    //define MAX_CHANNUM    16    //The
maximum number of channel
                    {
                        if (struAlarmInfo.dwChannel[i] == 1)
                        {
                            printf("Channel No. with Motion Detection Alarm %d
\n", i+1);
                        }
                    }
                    break;
                default:
                    break;
            }
            break;
        }
        case COMM_UPLOAD_PLATE_RESULT:
        {
            NET_DVR_PLATE_RESULT struPlateResult={0};
            memcpy(&struPlateResult, pAlarmInfo, sizeof(struPlateResult));
            printf("License Plate Number: %s\n",
struPlateResult.struPlateInfo.sLicense);//License plate number
```

```
switch(struPlateResult.struPlateInfo.byColor)//License plate color
{
case VCA_BLUE_PLATE:
    printf("Vehicle Color: Blue\n");
    break;
case VCA_YELLOW_PLATE:
    printf("Vehicle Color: Yellow\n");
    break;
case VCA_WHITE_PLATE:
    printf("Vehicle color: White\n");
    break;
case VCA_BLACK_PLATE:
    printf("Vehicle Color: Black\n");
    break;
default:
    break;
}
//Scene picture
if (struPlateResult.dwPicLen != 0 && struPlateResult.byResultType
== 1 )
{
    sprintf(filename,"testpic_%d.jpg",iNum);
    fSnapPic=fopen(filename,"wb");
    fwrite(struPlateResult.pBuffer1,struPlateResult.dwPicLen,
1,fSnapPic);
    iNum++;
    fclose(fSnapPic);
}
//License plate picture
if (struPlateResult.dwPicPlateLen != 0 &&
struPlateResult.byResultType == 1)
{
    sprintf(filename,"testPicPlate_%d.jpg",iNum);
    fSnapPicPlate=fopen(filename,"wb");
    fwrite(struPlateResult.pBuffer1,struPlateResult.dwPicLen,
1,fSnapPicPlate);
    iNum++;
    fclose(fSnapPicPlate);
}
//Processing other data...
break;
}
case COMM_ITS_PLATE_RESULT:
{
    NET_ITS_PLATE_RESULT struITSPlateResult={0};
    memcpy(&struITSPlateResult, pAlarmInfo, sizeof(struITSPlateResult));

    for (i=0;i<struITSPlateResult.dwPicNum;i++)
    {
        printf("License Plate Number: %s\n",
struITSPlateResult.struPlateInfo.sLicense);//License plate number
```

```
switch(struITSPlateResult.struPlateInfo.byColor)//License plate
color
{
case VCA_BLUE_PLATE:
    printf("Vehicle Color: Blue\n");
    break;
case VCA_YELLOW_PLATE:
    printf("Vehicle Color: Yellow\n");
    break;
case VCA_WHITE_PLATE:
    printf("Vehicle Color: White\n");
    break;
case VCA_BLACK_PLATE:
    printf("Vehicle Color: Black\n");
    break;
default:
    break;
}
//Save scene picture
if ((struITSPlateResult.struPicInfo[i].dwDataLen !=
0)&&(struITSPlateResult.struPicInfo[i].byType== 1)||
(struITSPlateResult.struPicInfo[i].byType == 2))
{
    sprintf(filename,"testITSpic%d_%d.jpg",iNum,i);
    fSnapPic=fopen(filename,"wb");
    fwrite(struITSPlateResult.struPicInfo[i].pBuffer,
struITSPlateResult.struPicInfo[i].dwDataLen,1,fSnapPic);
    iNum++;
    fclose(fSnapPic);
}
//License plate thumbnails
if ((struITSPlateResult.struPicInfo[i].dwDataLen !=
0)&&(struITSPlateResult.struPicInfo[i].byType == 0))
{
    sprintf(filename,"testPicPlate%d_%d.jpg",iNum,i);
    fSnapPicPlate=fopen(filename,"wb");
    fwrite(struITSPlateResult.struPicInfo[i].pBuffer,
struITSPlateResult.struPicInfo[i].dwDataLen, 1, \ fSnapPicPlate);
    iNum++;
    fclose(fSnapPicPlate);
}
//Processing other data...
}
break;
}
default:
    break;
}
}

void main() {
```

```
//-----
//Initialize
NET_DVR_Init();
//Set the connection time and reconnection time
NET_DVR_SetConnectTime(2000, 1);
NET_DVR_SetReconnect(10000, true);

//-----
//Log in to device
LONG lUserID;
NET_DVR_DEVICEINFO_V30 struDeviceInfo;
lUserID = NET_DVR_Login_V30("172.0.0.100", 8000, "admin", "12345",
&struDeviceInfo);
if (lUserID < 0)
{
    printf("Login error, %d\n", NET_DVR_GetLastError());
    NET_DVR_Cleanup();
    return;
}

//Set alarm callback function
NET_DVR_SetDVRMessageCallBack_V50(0, MessageCallbackNo1, NULL);
NET_DVR_SetDVRMessageCallBack_V50(1, MessageCallbackNo2, NULL);

//Enable arming
NET_DVR_SETUPALARM_PARAM struSetupParam={0};
struSetupParam.dwSize=sizeof(NET_DVR_SETUPALARM_PARAM);

//Alarm information type to upload: 0-History Alarm (NET_DVR_PLATE_RESULT), 1-
Real-Time Alarm (NET_DVR_PLATE_RESULT)
struSetupParam.byAlarmInfoType=1;
//Arming Level: Level-2 arming (for traffic device)
struSetupParam.byLevel=1;

LONG lHandle = NET_DVR_SetupAlarmChan_V41(lUserID,&struSetupParam);
if (lHandle < 0)
{
    printf("NET_DVR_SetupAlarmChan_V41 error, %d\n", NET_DVR_GetLastError());
    NET_DVR_Logout(lUserID);
    NET_DVR_Cleanup();
    return;
}

Sleep(20000);
//Disarm uploading channel
if (!NET_DVR_CloseAlarmChan_V30(lHandle))
{
    printf("NET_DVR_CloseAlarmChan_V30 error, %d\n", NET_DVR_GetLastError());
    NET_DVR_Logout(lUserID);
    NET_DVR_Cleanup();
    return;
}
```

```
//User logout
NET_DVR_Logout(lUserID);
//Release SDK resource
NET_DVR_Cleanup();
return;
}
```

### See Also

**NET\_DVR\_SetupAlarmChan\_V50**

### 4.18.1 MSGCallback

Alarm/event information callback function.

#### Callback Function Definition

```
typedef void(CALLBACK *MSGCallback) (
    LONG                lCommand,
    NET_DVR_ALARMER     *pAlarmer,
    char                *pAlarmInfo,
    DWORD               dwBufLen,
    void                *pUser
);
```

#### Parameters

##### **lCommand**

[OUT] Uploaded message type. You can distinguish the alarm/event information via the type.

##### **pAlarmer**

[OUT] Alarm device information, including serial No., IP address, login handle, and so on, see details in **NET\_DVR\_ALARMER**.

##### **pAlarmInfo**

[OUT] Alarm/event information, the details are returned in different structures according to **lCommand**.

##### **dwBufLen**

[OUT] Size of alarm/event information buffer.

##### **pUser**

[OUT] User data.



## 4.19 NET\_DVR\_SetupAlarmChan\_V50

Set up persistent connection to receive alarm/event information (supports alarm/event subscription).

### API Definition

```
LONG NET_DVR_SetupAlarmChan_V50(  
    LONG                lUserID,  
    NET_DVR_SETUPALARM_PARAM_V50 lpSetupParam,  
    char                *pData,  
    DWORD               dwDataLen,  
);
```

### Parameters

#### lUserID

[IN] Value returned by NET\_DVR\_Login\_V40.

#### lpSetupParam

[IN] Arming parameters, refer to the structure NET\_DVR\_SETUPALARM\_PARAM\_V50 for details.

#### pData

[IN] Alarm/event subscription conditions.

#### dwDataLen

[IN] Length of alarm/event subscription conditions.

### Return Values

Return -1 for failure, and return other values as the handles of NET\_DVR\_CloseAlarmChan\_V30. If -1 is returned, you can call NET\_DVR\_GetLastError to get the error code.

### Remarks

This API supports alarm/event subscription, you can specify the types of alarm or event to be uploaded by device by setting **pData** and **dwDataLen**.

## 4.20 NET\_DVR\_StartListen\_V30

Register callback function for receiving alarm/event information and start listening (supports multiple threads).

### API Definition

```
LONG NET_DVR_StartListen_V30(  
    char                *sLocalIP,
```

```
WORD          wLocalPort,  
MSGCallback   DataCallback,  
void          *pUserData  
);
```

### Parameters

#### sLocalIP

[IN] IP address of local PC. It can be set to null.

#### wLocalPort

[IN] Listening port No. of local PC. It is configured by user, and it should be the same with that of device.

#### DataCallback

[IN] Alarm/event information callback function, see details in [\*\*\*MSGCallback\*\*\*](#) .

#### pUserData

[IN] User data.

### Return Values

Return -1 for failure, and return other values for the handle parameters of

[\*\*\*NET\\_DVR\\_StopListen\\_V30\*\*\*](#) .

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

The available error codes of this API are 0, 3, 6, 12, 17, 41, 44, 47, 72, and 75. See details in the [\*\*\*Device Network SDK Errors\*\*\*](#) .

### Remarks

- To receive the alarm/event information sent by device, you should set the management host server address or listening host server address of device to the IP address of PC (which is same with the **sLocalIP**), or set the management host server port or listening host server port to the listening port No. of PC (which is same with the **wLocalPort**).
- The callback function in this API is prior to other callback functions, that is, if the callback function is configured in this API, other callback functions will not receive the alarm information. All the device alarm information is returned in same callback function, and you can distinguish the devices via the alarm device information (**pAlarmInfo**).

## 4.21 NET\_DVR\_StopListen\_V30

Stop listening (supports multiple threads).

### API Definition

```
BOOL NET_DVR_StopListen_V30(  
    LONG      lListenHandle  
);
```

### Parameters

#### **IListenHandle**

Listening handle, which is returned by **NET\_DVR\_StartListen\_V30** .

### 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.

The available error codes of this API are 0, 3, 12, and 17. See details in the **Device Network SDK Errors** .

## Appendix A. Data Structure

### A.1 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
bySubProto	BYTE	Transmission protocol type of sub-stream: 0-private protocol (default), 1-RTSP, 2-private protocol+RTSP
bySupport	BYTE	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. <ul style="list-style-type: none"><li>bySupport&amp;0x1: whether supports VCA search.</li><li>bySupport&amp;0x2: whether supports backup.</li><li>bySupport&amp;0x4: whether supports getting encoding parameters.</li></ul>

Member	Data Type	Description
		<ul style="list-style-type: none"> <li>bySupport&amp;0x8: whether supports dual-NIC.</li> <li>bySupport&amp;0x10: whether supports remote SADP.</li> <li>bySupport&amp;0x20: whether supports RAID card.</li> <li>bySupport&amp;0x40: whether supports searching in IPSAN directory.</li> <li>bySupport&amp;0x80: whether supports RTP over RTSP.</li> </ul>
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"> <li>bySupport1&amp;0x1: whether supports SNMP with version 30.</li> <li>bySupport1&amp;0x2: whether supports playback and downloading video files.</li> <li>bySupport1&amp;0x4: whether supports setting the arming priority.</li> <li>bySupport1&amp;0x8: whether supports extending the arming time period.</li> <li>bySupport1&amp;0x10: whether supports multiple HDDs (more than 33).</li> <li>bySupport1&amp;0x20: whether supports RTP over RTSP.</li> <li>bySupport1&amp;0x80: whether supports license plate recognition alarm.</li> </ul>
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"> <li>bySupport2&amp;0x1: whether supports getting stream via URL.</li> <li>bySupport2&amp;0x2: whether supports FTP with version 40.</li> <li>bySupport2&amp;0x4: whether supports ANR.</li> </ul>

Member	Data Type	Description
		<ul style="list-style-type: none"> <li>bySupport2&amp;0x20: whether supports getting device status.</li> <li>bySupport2&amp;0x40: whether supports encrypting stream.</li> </ul>
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"> <li>bySupport3&amp;0x1: whether supports multi-stream.</li> <li>bySupport3&amp;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).</li> <li>bySupport3&amp;0x20: whether supports getting stream via DDNS.</li> </ul>
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"> <li>byMultiStreamProto&amp;0x1: whether supports third-stream.</li> <li>byMultiStreamProto&amp;0x2: whether supports fourth-stream.</li> <li>byMultiStreamProto&amp;0x40: whether supports main stream.</li> <li>byMultiStreamProto&amp;0x80: whether supports sub-stream.</li> </ul>
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	Extended capabilities, if the result of bitwise operation is 0, it refers that the capability is not

Member	Data Type	Description
		supported, if the result is 1, it indicates that the capability is supported. <ul style="list-style-type: none"> <li>bySupport4&amp;0x01: whether all stream types support RTSP and private protocol.</li> <li>bySupport4&amp;0x02: whether the device supports transmitting form format data via API (NET_DVR_STDXMLConfig).</li> <li>bySupport4&amp;0x10: whether supports loading network disk by domain name.</li> </ul>
byLanguageType	BYTE	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. <ul style="list-style-type: none"> <li>byLanguageType==0: this field is not supported by device.</li> <li>byLanguageType&amp;0x1: whether supports Chinese.</li> <li>byLanguageType&amp;0x2: whether supports English.</li> </ul>
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  $\text{byIPChanNum} + \text{byHighDChanNum} * 256$ .
- 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.2 NET\_DVR\_DEVICEINFO\_V40

### Device Parameter Structure (V40)

Member	Data Type	Description
struDeviceV30	<b><u>NET_DVR_DEVICEINFO_V30</u></b>	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 <b>bySupportLock</b> 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 <b>bySupportLock</b> is 1. During the 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 <b>bySupport&amp;0x1</b> indicates that this member is reserved; the result of <b>bySupport&amp;0x2</b> indicates



Member	Data Type	Description
		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 <b>iResidualValidity</b> is valid: 0-invalid, 1-valid.
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).

Member	Data Type	Description
bySupportStreamEncrypt	BYTE	Whether it supports stream encryption, it depends on the result of bitwise AND (&) operation: 0-no, 1-yes. The result of <b>bySupportStreamEncrypt&amp;0x1</b> indicates whether to support RTP/TLS streaming, the result of <b>bySupportStreamEncrypt&amp;0x2</b> indicates whether to support SRTP/UDP streaming, and the result of <b>bySupportStreamEncrypt&amp;0x4</b> 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.3 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.4 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.5 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 <b>Content-Type</b> 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 <b>name</b> field in the message), e.g., name="upload".
szFilename	Array of char	Content file name (corresponds to <b>filename</b> field in the message), e.g., filename="C:\Users\test\Desktop\11.txt".
dwContentLen	DWORD	Content size
pContent	char*	Data point
bySelfRead	BYTE	0-External file, 1-Internal data, whose address is specified by <b>szFilename</b> .
byRes	Array of BYTE	Reserved. Set to 0. Maximum: 15 bytes.

## See Also

**NET\_DVR\_XML\_CONFIG\_INPUT**

## A.6 NET\_DVR\_STD\_CONFIG

## Structure About Configuring Input and Output Parameters

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

## A.7 NET\_DVR\_STD\_ABILITY

### Input and Output Parameter Structure for Getting Capabilities

Member	Data Type	Description
<b>lpCondBuffer</b>	LPVOID	Condition parameters (ASCII character format), e.g., the channel No., it can be set to "null".
<b>dwCondSize</b>	DWORD	Buffer size of condition parameters.

Member	Data Type	Description
<b>IpOutBuffer</b>	LPVOID	Output parameters buffer (the parameter is returned in the message with XML format), it cannot be set to "null".
<b>dwOutSize</b>	DWORD	Output buffer size.
<b>IpStatusBuffer</b>	LPVOID	Get the returned status parameters ( <i><b><u>XML_ResponseStatus</u></b></i> ) when getting capabilities failed. It can be set to null.
<b>dwStatusSize</b>	DWORD	Status buffer size.
<b>dwRetSize</b>	DWORD	Obtained data size (if the capability is obtained, the value refers to the size of <b>IpOutBuffer</b> ; if getting failed, the value refers to the size of <b>IpStatusBuffer</b> ).
<b>byRes</b>	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 **IpCondBuffer** and output parameter **IpOutBuffer** are different. For details, refer to the typical applications.

## A.8 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.
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.

Member	Data Type	Description
cbLoginResult	<u><b>fLoginResultCallback</b></u>	Callback function used to return login status, it is valid only when <b>bUseAsynLogin</b> 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 <b>byLoginMode</b> is 0 and <b>byHttps</b> is 1); when <b>byVerifyMode</b> is 0, CA certificate is not required, when <b>byVerifyMode</b> 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.

## A.9 NET\_DVR\_XML\_CONFIG\_INPUT

**Input Parameter Structure of Message Transmission API (NET\_DVR\_STDXMLConfig)**

Member	Data Type	Description
dwSize	DWORD	Structure size.
IpRequestUrl	void*	Request URL (command) for implement different functions, and it is in string format.
dwRequestUrlLen	DWORD	Request URL size.
IpInBuffer	void*	Buffer for storing input parameters (request messages), see the input content details structure in <b><u>NET_DVR_MIME_UNIT</u></b> .
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 transmitted by the parameter <b>IpInBuffer</b> in the structure <b><u>NET_DVR_MIME_UNIT</u></b> .
byRes	Array of BYTE	Reserved, set to 0.

**Related API****NET\_DVR\_STDXMLConfig****A.10 NET\_DVR\_XML\_CONFIG\_OUTPUT****Output Parameter Structure of Message Transmission API (NET\_DVR\_STDXMLConfig)**

Member	Data Type	Description
dwSize	DWORD	Structure size.
IpOutBuffer	void*	Buffer for storing output parameters (response messages), which is allocated when passing through URL by GET method.



Member	Data Type	Description
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 <b>byNumOfMultiPart</b> 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.11 NET\_DVR\_CHANNEL\_GROUP

Channel No. and group No. information structure.

### Structure Definition

```
struct{
    DWORD    dwSize;
    DWORD    dwChannel;
    DWORD    dwGroup;
    BYTE     byID;
    BYTE     byRes1[3];
    DWORD    dwPositionNo;
    BYTE     byRes[56];
}NET_DVR_CHANNEL_GROUP, *LPNET_DVR_CHANNEL_GROUP;
```

## Members

### dwSize

Structure size

**dwChannel**

Channel No.

**dwGroup**

Group No., which starts from 0, 0-group No.1, 1-group No.2, ...

**byID**

Device region configuration ID

**byRes1**

Reserved, set to 0.

**dwPositionNo**

Scene No., for network camera, it is 0; for speed dome, it starts from 1.

**byRes**

Reserved, set to 0.

## A.12 NET\_DVR\_CRUISECHAN\_INFO

### Structure about The Information of Channel That Called Patrol

Member	Data Type	Description
dwEnableCruiseChan	DWORD	Channel that called patrol.
dwCruiseNo	DWORD	Patrol No., 0xffffffff-invalid.

## A.13 NET\_DVR\_EVENT\_SCHEDULE

### Arming Schedule Parameter Structure

Member	Data Type	Description
dwSize	DWORD	Structure size
struAlarmTime	Array of <b><u>NET_DVR_SCHEDTIME</u></b>	Arming schedule, 7 days per week, 8 time periods per day
struHolidayAlarmTime	Array of <b><u>NET_DVR_SCHEDTIME</u></b>	Holiday arming schedule, see details in the structure .
byRes	BYTE	Reserved.

## A.14 NET\_DVR\_EVENT\_TRIGGER

### Structure About Event Linkage Configuration

Member	Data Type	Description
dwSize	DWORD	Structure size.
struHandleException	Array of <b><u>NET_DVR_HANDLEEXCEPTION_V41</u></b>	Exception handling mode
dwRelRecordChan	Array of DWORD	Actually triggered video channel, represented by value, read starts from 0, and it is invalid after the value of 0xffffffff being read.
struPresetChanInfo	Array of <b><u>NET_DVR_PRESETCHANNEL_INFO</u></b>	Information of channel that called preset
struCruiseChanInfo	Array of <b><u>NET_DVR_CRUISECHANNEL_INFO</u></b>	Information of channel that called patrol
struPtzTrackInfo	Array of <b><u>NET_DVR_PTZTRACKCHANNEL_INFO</u></b>	Information of channel that called pattern
byDirection	Array of BYTE	Triggering direction: 0-reserved, 1-all, 2-forward, 3-backward
szFDID	Char	Face picture library ID
byRes2	Array of BYTE	Reserved

## A.15 NET\_DVR\_HANDLEEXCEPTION\_V41

### Exception Information Structure

Member	Data Type	Description
dwHandleType	DWORD	Handling types, see details below: <ul style="list-style-type: none"><li>• 0x00: no response</li><li>• 0x01: display alarm on monitor screen</li></ul>

Member	Data Type	Description
		<ul style="list-style-type: none"> <li>• 0x02: audio warning</li> <li>• 0x04: upload to center</li> <li>• 0x08: trigger alarm output</li> <li>• 0x10: send picture with JPEG format by e-mail</li> <li>• 0x20: trigger wireless sound and light alarm</li> <li>• 0x40: trigger e-map (supported by PCNVR only)</li> <li>• 0x200: capture picture and upload to FTP</li> <li>• 0x400: focus mode linkage (for defocus detection)</li> <li>• 0x800: PTZ linkage (speed dome tracks the target)</li> <li>• 0x1000: capture picture and upload to cloud storage.</li> <li>• 0x10000: message alarm</li> </ul> <p>E.g., if <b>dwHandleType</b> is 0x01 0x04, it indicates that the alarm information will be displayed on monitor screen and uploaded to alarm center when the alarm is triggered.</p>
dwMaxAlarmOutChannelNum	DWORD	Manixmum number of alarm outputs (read only) supported by the device.
dwRelAlarmOut	Array of DWORD	Alarm output No. triggered by alarm, which starts from 0, 0xffffffff-invalid. E.g. <b>byRelAlarmOut[i]==3</b> indicates that the alarm output No.4 is triggered.
byRes	Array of BYTE	Reserved, set to 0.

## A.16 NET\_DVR\_JPEGPARA

Structure about information of picture in JPEG format.

### Structure Definition

```
struct{
    WORD        wPicSize;
    WORD        wPicQuality;
}NET_DVR_JPEGPARA, *LPNET_DVR_JPEGPARA;
```

## Members

### wPicSize

Picture size: 0-CIF(352 × 288/352 × 240), 1-QCIF(176 × 144/176 × 120), 2-4CIF(704 × 576/704 × 480) or D1(720 × 576/720 × 486), 3-UXGA(1600 × 1200), 4-SVGA(800 × 600), 5-HD720P(1280 × 720), 6-VGA(640 × 480), 7-XVGA(1280 × 960), 8-HD900P(1600 × 900), 9-HD1080P(1920 × 1080), 10-2560 × 1920, 11-1600 × 304, 12-2048 × 1536, 13-2448 × 2048, 14-2448 × 1200, 15-2448 × 800, 16-XGA(1024 × 768), 17-SXGA(1280 × 1024), 18-WD1(960 × 576/960 × 480), 19-1080I(1920 × 1080), 20-576 × 576, 21-1536 × 1536, 22-1920 × 1920, 23-320 × 240, 24-720 × 720, 25-1024 × 768, 26-1280 × 1280, 27-1600 × 600, 28-2048 × 768, 29-160 × 120, 75-336 × 256, 78-384 × 256, 79-384 × 216, 80-320 × 256, 82-320 × 192, 83-512 × 384, 127-480 × 272, 128-512 × 272, 161-288 × 320, 162-144 × 176, 163-480 × 640, 164-240 × 320, 165-120 × 160, 166-576 × 720, 167-720 × 1280, 168-576 × 960, 180-180 × 240, 181-360 × 480, 182-540 × 720, 183-720 × 960, 184-960 × 1280, 185-1080 × 1440, 215-1080 × 720 (reserved), 216-360 × 640 (reserved), 218-1440 × 1440, 500-384 × 288, 0xff-Auto.

### wPicQuality

Picture quality: 0-high, 1-medium, 2-low

## A.17 NET\_DVR\_PRESETCHAN\_INFO

### Preset Information Structure

Member	Data Type	Description
dwEnablePresetChan	DWORD	Channel that called preset.
dwPresetPointNo	DWORD	Called preset No., 0xffffffff-not call preset.

## A.18 NET\_DVR\_PTZTRACKCHAN\_INFO

### Pattern Information Structure

Member	Data Type	Description
dwEnablePtzTrackChan	DWORD	Channel that called the pattern.
dwPtzTrackNo	DWORD	Called pattern No., 0xffffffff-invalid.

## A.19 NET\_DVR\_PTZ\_POSITION

Structure about scene configuration.

### Structure Definition

```
struct{
    BYTE            byEnable;
    BYTE            byRes1[3];
    BYTE            byPtzPositionName[NAME_LEN/*32*/];

                                NET_DVR_PTZPOS
                                struPtzPos;
    BYTE            byRes2[40];
}NET_DVR_PTZ_POSITION, *LPNET_DVR_PTZ_POSITION;
```

### Members

#### byEnable

Whether to enable scene: 0-no, 1-yes

#### byRes1

Reserved, set to 0

#### byPtzPositionName

Scene name

#### struPtzPos

PTZ coordinates

#### byRes2

Reserved, set to 0

### Remarks

The **byEnable** is invalid when setting behavior analysis rule, but it is valid when setting local scene parameters.

## A.20 NET\_DVR\_PTZPOS

PTZ position parameter structure.

### Structure Definition

```
struct{
    WORD            wAction;
    WORD            wPanPos;
    WORD            wTiltPos;
```

```
WORD        wZoomPos;  
}NET_DVR_PTZPOS, *LPNET_DVR_PTZPOS;
```

### Members

#### wAction

Operation type, it is valid only when setting parameters. 1-locate PTZ parameter, 2-locate Pan parameter, 3-locate Tilt parameter, 4-locate Zoom parameter, 5-locate Pan and Tilt parameter.

#### wPanPos

Pan parameter (horizontal)

#### wTiltPos

Tilt parameter (vertical)

#### wZoomPos

Zoom parameter (zoom in or out)

### Remarks

The actual PTZ position value is the one-tenth of the received hexadecimal value. If the obtained Pan value is 0x1750, the actual Pan value is 175 degree; if the obtained Tilt value is 0x0789, the actual Tilt value is 78.9 degree; if the obtained Zoom value is 0x1100, the actual Zoom value is 110.

## A.21 NET\_DVR\_REGION\_ENTRANCE\_DETECTION

Region entrance detection parameter structure.

### Structure Definition

```
struct{  
    DWORD        dwSize;  
    BYTE          byEnabled;  
    BYTE          byEnableHumanMisinfoFilter;  
    BYTE          byEnableVehicleMisinfoFilter;  
    BYTE          byRes1;  
    NET_DVR_REGIONENTRANCE_REGION  struRegion[MAX_REGION_NUM/*8*/];  
    BYTE          byRes2[128];  
}NET_DVR_REGION_ENTRANCE_DETECTION, *LPNET_DVR_REGION_ENTRANCE_DETECTION;
```

### Members

#### dwSize

Structure size.

#### byEnabled

Whether to enable region entrance detection: 0-no, 1-yes

### **byEnableHumanMisinfoFilter**

Whether to enable false human body alarm prevention: 0-no, 1-yes

### **byEnableVehicleMisinfoFilter**

Whether to enable false vehicle alarm prevention: 0-no, 1-yes

### **byRes1**

Reserved, set to 0.

### **struRegion**

Coordinates of detection region, refer to the structure **NET\_DVR\_REGIONENTRANCE\_REGION** for details.

### **byRes2**

Reserved, set to 0.

### **Remarks**

- If the values of all members in the structure **NET\_DVR\_REGIONENTRANCE\_REGION** are set to 0, it indicates clearing this region. If the value of any members in the structure is not set to 0, the parameter **struRegion** is invalid (that is the settings will not take effect).
- The detection region should be configured one by one.
- If the false alarm prevention is enabled, the false alarm will be filtered without uploading, but the false alarm will still be locally stored.

## **A.22 NET\_DVR\_REGION\_EXITING\_DETECTION**

Region exiting detection parameter structure.

### **Structure Definition**

```
struct{
    DWORD          dwSize;
    BYTE           byEnabled;
    BYTE           byEnableHumanMisinfoFilter;
    BYTE           byEnableVehicleMisinfoFilter;
    BYTE           byRes1;
    NET_DVR_REGIONEXITING_REGION    struRegion[MAX_REGION_NUM/*8*/];
    BYTE           byRes[128];
}NET_DVR_REGION_EXITING_DETECTION,*LPNET_DVR_REGION_EXITING_DETECTION;
```

### **Members**

#### **dwSize**

Structure size.

#### **byEnabled**



Whether to enable region exiting detection: 0-no, 1-yes

**byEnableHumanMisinfoFilter**

Whether to enable false human body alarm prevention: 0-no, 1-yes

**byEnableVehicleMisinfoFilter**

Whether to enable false vehicle alarm prevention: 0-no, 1-yes

**byRes1**

Reserved, set to 0.

**struRegion**

Coordinates of detection region, refer to the structure **NET\_DVR\_REGIONEXITING\_REGION** for details.

**byRes2**

Reserved, set to 0.

**Remarks**

- If the values of all members in the structure **NET\_DVR\_REGIONEXITING\_REGION** are set to 0, it indicates clearing this region. If the value of any members in the structure is not set to 0, the parameter **struRegion** is invalid (that is the settings will not take effect).
- If the false alarm prevention is enabled, the false alarm will be filtered without uploading, but the false alarm will still be locally stored.

## A.23 NET\_DVR\_REGIONENTRANCE\_REGION

### Structure about the parameters of region entrance detection

Member	Data Type	Description
<b>struRegion</b>	<b><u>NET_VCA_POLYGON</u></b>	Coordinates of detection region.
<b>bySensitivity</b>	BYTE	Sensitivity, value range: [1,100].
<b>byDetectionTarget</b>	BYTE	Detection target: 0-all (all targets will be detected), 0x01-human, 0x02-vehicle, 0x04-other. Multiple types of targets can be selected, e.g., 0x3-detect vehicle and human.

Member	Data Type	Description
<b>byAlarmConfidence</b>	BYTE	Confidence of alarm notification: 0-low, 1-medium low, 2-medium high, 3-high.
<b>byRecordConfidence</b>	BYTE	Confidence of video recording: 0-low, 1-medium low, 2-medium high, 3-high.
<b>byRes</b>	BYTE[]	Reserved, the maximum size is 60 bytes.

### Remarks

If the detection region coordinates (**struRegion**) is set to 0, it indicates clearing region. And for speed dome, when getting each detection region, the lens will positioning to the scene position of the detection region.

## A.24 NET\_DVR\_REGIONEXITING\_REGION

### Structure about the parameters of region exiting detection

Member	Data Type	Description
<b>struRegion</b>	<b><u>NET_VCA_POLYGON</u></b>	Coordinates of detection region.
<b>bySensitivity</b>	BYTE	Sensitivity, value range: [1,100].
<b>byDetectionTarget</b>	BYTE	Detection target: 0-all (all targets will be detected), 0x01-human, 0x02-vehicle, 0x04-other. Multiple types of targets can be selected, e.g., 0x3-detect vehicle and human.
<b>byAlarmConfidence</b>	BYTE	Confidence of alarm notification: 0-low, 1-medium low, 2-medium high, 3-high.

Member	Data Type	Description
<b>byRecordConfidence</b>	BYTE	Confidence of video recording: 0-low, 1-medium low, 2-medium high, 3-high.
<b>byRes</b>	BYTE[]	Reserved, the maximum size is 60 bytes.

### Remarks

If the detection region coordinates (**struRegion**) is set to 0, it indicates clearing region. And for speed dome, when getting each detection region, the lens will positioning to the scene position of the detection region.

## A.25 NET\_DVR\_SCHEDTIME

### Structure About Start and End Time Parameters

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

## A.26 NET\_DVR\_SMART\_REGION\_COND

Structure about the configuration conditions of smart detection region.

### Structure Definition

```
struct{
    DWORD    dwSize;
    DWORD    dwChannel;
    DWORD    dwRegion;
}NET_DVR_SMART_REGION_COND, *LPNET_DVR_SMART_REGION_COND;
```

### Members

#### **dwSize**

Structure size

### **dwChannel**

Channel No.

### **dwRegion**

Region ID, start from 1.

## **A.27 NET\_VCA\_ADV\_REACH\_HEIGHT**

Structure about advanced climbing detection parameters.

### **Structure Definition**

```
struct{
    NET_VCA_POLYGON      struRegion;
    DWORD                dwCrossDirection;
    BYTE                 byRes[4];
}NET_VCA_ADV_REACH_HEIGHT, *LPNET_VCA_ADV_REACH_HEIGHT;
```

### **Members**

#### **struRegion**

Coordinates of detection region, refer to the structure for details.

#### **dwCrossDirection**

Crossing direction, see details in the enumeration definition below:

```
enum _VCA_CROSS_DIRECTION_{
    VCA_BOTH_DIRECTION    = 0,
    VCA_LEFT_GO_RIGHT     = 1,
    VCA_RIGHT_GO_LEFT     = 2
}_VCA_CROSS_DIRECTION
```

#### **VCA\_BOTH\_DIRECTION**

Bidirectional

#### **VCA\_LEFT\_GO\_RIGHT**

From left to right

#### **VCA\_RIGHT\_GO\_LEFT**

From right to left

#### **byRes**

Reserved, set to 0.

### **See Also**

## A.28 NET\_VCA\_ADV\_TRAVERSE\_PLANE

Structure about advanced line crossing detection parameters.

### Structure Definition

```
struct{
    NET_VCA_POLYGON      struRegion;
    DWORD                dwCrossDirection;
    BYTE                 bySensitivity;
    BYTE                 byRes[3];
}NET_VCA_ADV_TRAVERSE_PLANE, *LPNET_VCA_ADV_TRAVERSE_PLANE;
```

### Members

#### **struRegion**

Coordinates of detection region, refer to the structure for details.

#### **dwCrossDirection**

Crossing direction, see details in the enumeration definition below:

```
enum _VCA_CROSS_DIRECTION_{
    VCA_BOTH_DIRECTION    = 0,
    VCA_LEFT_GO_RIGHT     = 1,
    VCA_RIGHT_GO_LEFT     = 2
}_VCA_CROSS_DIRECTION
```

#### **VCA\_BOTH\_DIRECTION**

Bidirectional

#### **VCA\_LEFT\_GO\_RIGHT**

From left to right

#### **VCA\_RIGHT\_GO\_LEFT**

From right to left

#### **bySensitivity**

Sensitivity, range: [1,5]

#### **byRes**

Reserved, set to 0.

### See Also

## A.29 NET\_VCA\_ANSWER

Structure about question answer detection parameters.

### Structure Definition

```
struct{
    NET_VCA_POLYGON          struRegion;
    BYTE                     bySensitivity;
    BYTE                     byAlarmState;
    BYTE                     byZoomOver;
    BYTE                     byAnswerStudent;
    BYTE                     byRes[4];
}NET_VCA_ANSWER, *LPNET_VCA_ANSWER;
```

### Members

#### **struRegion**

Coordinates of detection region, refer to the structure for details.

#### **bySensitivity**

Sensitivity, range: [1,100]

#### **byAlarmState**

Read-only field will not be displayed on the interface: 0- Reserved, 1- Alarm starts, 2- Alarm ends

#### **byZoomOver**

0-reserved, 1-zoomed (3-close-up) (C-always set alarm to 0) (alarm is uploaded to education sharing system, which is for image switch).

#### **byAnswerStudent**

Question answer status: 0-reserved, 1-no student answered, 2-a student answered, 3-multiple student answered (C-panorama).

#### **byRes**

Reserved, set to 0.

### See Also

## A.30 NET\_VCA\_APPEND\_INFO

## Structure about Additional Information of Behavior Analysis

Member	Data Type	Description
<b>dwSize</b>	DWORD	Structure size.
<b>dwAppendPicLen</b>	DWORD	Attached picture size.
<b>pAppendPicBuff</b>	BYTE	Buffer pointer for saving attached picture information.
<b>byAppendPicType</b>	BYTE	Attached picture type.
<b>byUID</b>	Array of BYTE	Device alarm UID, the maximum size is 64 bytes.
<b>byRes1</b>	Array of BYTE	Reserved. The maximum size is 3 bytes.
<b>dwTargetSpeed</b>	DWORD	Target speed detected by radar, unit: km/h.
<b>dwTargetDistance</b>	DWORD	Target distance detected by radar, unit: m.
<b>byAlarmType</b>	BYTE	Alarm trigger source type: 0-triggered by video, 1-triggered by radar.
<b>byRadarChannel</b>	BYTE	Radar channel No., which starts from 1.
<b>byRes</b>	Array of BYTE	Reserved. The maximum size is 50 bytes.

### See Also

**NET\_VCA\_RULE\_ALARM**

## A.31 NET\_VCA\_AREA

Structure about entering/exiting region detection parameters

### Structure Definition

```
struct{
    NET_VCA_POLYGON    struRegion;
    BYTE               bySensitivity;
    BYTE               byDetectionTarget;
    BYTE               byPriority;
    BYTE               byRes[5];
}NET_VCA_AREA, *LPNET_VCA_AREA;
```

### Members

**struRegion**

Coordinates of detection region, refer to the structure for details.

**bySensitivity**

Sensitivity, range: [1,5]

**byDetectionTarget**

Detection target: 0-all (all targets will be detected), 0x01-human, 0x02-vehicle, 0x04-other.  
Multiple types of targets can be selected, e.g., 0x3-detect vehicle and human.

**byPriority**

Alarm priority: 0-low, 1-medium, 2-high

**byRes**

Reserved, set to 0.

**See Also**

### A.32 NET\_VCA\_AUDIO\_ABNORMAL

**Table A-1 Structure about Detection Parameters for Sudden Increase of Sound Intensity**

Member	Data Type	Description
<b>wDecibel</b>	WORD	Sound intensity.
<b>bySensitivity</b>	BYTE	Sensitivity, value range: [1,100].
<b>byAudioMode</b>	BYTE	Sound detection mode: 0-enable sensitivity detection, 1-enable decibel threshold detection, 2-enable both.
<b>byEnable</b>	BYTE	Whether to enable sudden increase of sound intensity detection: 0-no, 1-yes.
<b>byThreshold</b>	BYTE	Sound intensity threshold, value range: [0,100].
<b>byRes</b>	Array [BYTE]	Reserved, set to 0. The maximum size is 54 bytes.

### A.33 NET\_VCA\_BEHAVIOR\_ABILITY



## Structure about Behavior Analysis Capability

Member	Data Type	Description
<b>dwSize</b>	DWORD	Structure size
<b>dwAbilityType</b>	DWORD	Types of supported behavior analysis, see <a href="#"><b><u>VCA_ABILITY_TYPE</u></b></a> for detailed definitions
<b>byMaxRuleNum</b>	BYTE	The maximum number of rules
<b>byMaxTargetNum</b>	BYTE	The maximum number of targets
<b>bySupport</b>	BYTE	The supported function, "bySupport & 0x01" indicates that device supports calibration
<b>byRes</b>	Array of BYTE	Reserved, set to 0. The maximum size is 5 bytes.
<b>dwAbilityTypeEx</b>	DWORD	Types of supported behavior analysis, see <a href="#"><b><u>VCA_ABILITY_TYPE_EX</u></b></a> for details.

### Remarks

When getting the supported behavior analysis types, you should configure both **dwAbilityType** and **dwAbilityTypeEx**.

## A.34 NET\_VCA\_BLACKBOARD\_WRITE

Structure about blackboard writing detection parameters.

### Structure Definition

```
struct{
    NET_VCA_POLYGON    struRegion;
    BYTE               byTeacherState;
    BYTE               byWritingState;
    BYTE               byWritingArea;
    BYTE               byRes[5];
}NET_VCA_BLACKBOARD_WRITE, *LPNET_VCA_BLACKBOARD_WRITE;
```

### Members

#### **struRegion**

Coordinates of detection area, refer to the structure [\*\*NET\\_VCA\\_POLYGON\*\*](#) for details.

#### **byTeacherState**

Teacher status (it is valid when uploading alarm): 0-close-up, 1-panorama, 2-no teacher

#### **byWritingState**

Blackboard writing status (it is valid when uploading alarm): 0-started, 1-ended

**byWritingArea**

Blackboard writing region (it is valid when uploading alarm): 0-whole blackboard, 1-left part of blackboard, 2-right part of blackboard.

**byRes**

Reserved, set to 0.

**See Also**

## A.35 NET\_VCA\_CHAN\_IN\_PARAM

### Structure about Input Parameters of Intelligent Channel Capability

Member	Data Type	Description
<b>byVCAType</b>	BYTE	Intelligent device types, see <u><b>VCA_CHAN_ABILITY_TYPE</b></u> for details.
<b>byMode</b>	BYTE	Mode, its value depends on the <b>byVCAType</b> , see the table below in remarks for details.
<b>byRes</b>	Array of BYTE	Reserved, set to 0. The maximum size is 2 bytes.

**Remarks**

<b>byVCAType</b>	<b>Intelligent Function</b>	<b>byMode</b>
<b>VCA_BEHAVIOR_FULL</b>	Complete behavior analysis	<u><b>BEHAVIOR_SCENE_MODE_TYPE</b></u>
<b>VCA_ATM</b>	ATM capability	<u><b>VCA_CHAN_MODE_TYPE</b></u>
<b>VCA_ITS</b>	ITS (intelligent traffic) capability	<u><b>TRAFFIC_SCENE_MODE</b></u>
<b>VCA_TFS</b>	TFS (traffic violation enforcement) capability	<u><b>TFS_CHAN_MODE_TYPE</b></u>

## A.36 NET\_VCA\_COMBINED\_RULE

Structure about rule combination parameters.

### Structure Definition

```
struct{  
    BYTE                byRuleSequence;  
    BYTE                byRes[7];  
    DWORD               dwMinInterval;  
    DWORD               dwMaxInterval;  
    NET_VCA_RELATE_RULE_PARAM struRule1Raram;  
    NET_VCA_RELATE_RULE_PARAM struRule2Raram;  
    BYTE                byRes1[36];  
}NET_VCA_COMBINED_RULE,*LPNET_VCA_COMBINED_RULE;
```

### Members

#### byRuleSequence

Rule trigger sequence: 0-forward or backward sequence, 1-backward sequence.

#### byRes

Reserved, set to 0.

#### dwMinInterval

Minimum time interval, unit: second

#### dwMaxInterval

Maximum time interval, unit: second

#### struRule1Raram

Rule 1, refer to the structure for details.

#### struRule2Raram

Rule 2, refer to the structure for details.

#### byRes1

Reserved, set to 0.

### See Also

## A.37 NET\_VCA\_DEV\_ABILITY

Intelligent device capability structure

### Structure Definition

```
struct{  
    DWORD    dwSize;  
    BYTE     byVCAChanNum;  
    BYTE     byPlateChanNum;
```

```

BYTE    byBBaseChanNum;
BYTE    byBAdvanceChanNum;
BYTE    byBFullChanNum;
BYTE    byATMChanNum;
BYTE    byPDCChanNum;
BYTE    byITSChanNum;
BYTE    byBPrisonChanNum;
BYTE    byFSnapChanNum;
BYTE    byFSnapRecogChanNum;
BYTE    byFRetrievalChanNum;
BYTE    bySupport;
BYTE    byFRecogChanNum;
BYTE    byBPPerimeterChanNum;
BYTE    byTPSChanNum;
BYTE    byTFSCChanNum;
BYTE    byFSnapBFullChanNum;
BYTE    byHeatMapChanNum;
BYTE    bySmartVehicleNum;
BYTE    bySmartHVTNum;
BYTE    bySmartNum;
BYTE    byVehicleNum;
BYTE    bySmartRoadDetectionNum;
BYTE    bySmartFaceDetectionNum;
BYTE    bySmartHeatMapNum;
BYTE    byHumanRecognitionNum;
BYTE    byEdcationStudentNum;
BYTE    byRoadDetectionNum;
BYTE    byPersonDensityDetection;
BYTE    bySafetyHelmetDetection;
BYTE    byPerimeterCapture;
BYTE    byHeelPDC;
BYTE    byl2MPLiveView;
BYTE    byTeacherBehaviorDetectNum;
BYTE    byMixedTargetDetection;
BYTE    byRes[4];
}NET_VCA_DEV_ABILITY, *LPNET_VCA_DEV_ABILITY;

```

### Members

#### **dwSize**

Structure size.

#### **byVCAChanNum**

Number of intelligent channels.

#### **byPlateChanNum**

Number of ANPR channels.

#### **byBBaseChanNum**

Number of behavior analysis channels (basic).

#### **byBAdvanceChanNum**

Number of behavior analysis channels (advanced)

**byBFullChanNum**

Number of behavior analysis channels (complete)

**byATMChanNum**

Number of ATM channels

**byPDCChanNum**

Number of people counting channels

**byITSCChanNum**

Number of traffic incident channels.

**byBPrisonChanNum**

Number of behavior analysis channels (cell).

**byFSnapChanNum**

Number of face capture channels

**byFSnapRecogChanNum**

Number of face capture and recognition channels.

**byFRetrievalChanNum**

Number of face picture dual-VCA channels.

**bySupport**

Capability, bit operation result: 0: not support, 1: support.

bySupport&0x1: whether supports smart tracking.

bySupport&0x2: whether supports extending to 128-channel streaming.

**byFRecogChanNum**

Number of face recognition channels.

**byBPPerimeterChanNum**

Number of behavior analysis channels (perimeter).

**byTPSChanNum**

Number of traffic guidance channels.

**byTFSCChanNum**

Number of traffic violation enforcement channels.

**byFSnapBFullChanNum**

Number of face capture and behavior analysis channels.

**byHeatMapChanNum**

Number of heat map statistics channels

**bySmartVehicleNum**

Number of SMART event and vehicle detection channels.

**bySmartHVTNum**

Number of SMART event and multi-target-type detection channels.

**bySmartNum**

Number of SMART event channels.

**byVehicleNum**

Number of vehicle detection channels.

**bySmartRoadDetectionNum**

Number of SMART event and traffic monitoring channels.

**bySmartFaceDetectionNum**

Number of SMART event and face detection channels.

**bySmartHeatMapNum**

Number of SMART event and heat map statistics channels.

**byHumanRecognitionNum**

Number of human body recognition channels.

**byEdcationStudentNum**

Number of classroom behavior tracking (student getting up detection and student counting) channels.

**byRoadDetectionNum**

Number of traffic monitoring channels.

**byPersonDensityDetection**

Number of people density detection channels.

**bySafetyHelmetDetection**

Number of hard hat detection channels.

**byPerimeterCapture**

Number of perimeter capture channels.

**byHeelPDC**

Number of heel people counting channels.

**by12MPLiveView**

Main stream 1200W@20fps live view (4000\*3000)

**byTeacherBehaviorDetectNum**

Number of teacher behavior analysis channels.

**byMixedTargetDetection**

Number of multi-target-type detection channels.

**byRes**

Reserved, set to 0.

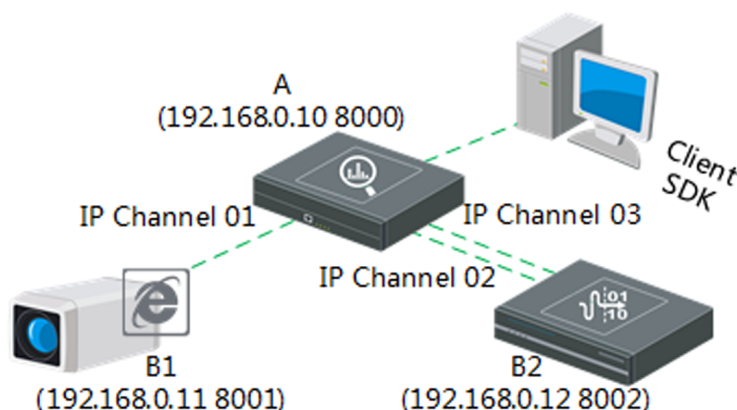
## A.38 NET\_VCA\_DEV\_INFO

### Structure About Camera Information

Member	Data Type	Description
struDevIP	<b><u>NET_DVR_IPADDR_UNION</u></b>	Device IP address
wPort	WORD	Device port No.
byChannel	BYTE	Device channel No.
byIvmsChannel	BYTE	Device channel No. for the HCNetSDK to access the device.

### Remarks

When accessing to device A, see the following figure for details.



- When the analog channel alarm of connected device (IPC, DVR, DVS, iVMS) is triggered, **struDevIP** and **wPort** are the IP address and port No. of connected device; **byChannel** and **byIvmsChannel** are both the alarm analog channel No.
- When the digital channel (IP channel) alarm of connected device (HDVR, NVR, iVMS) is triggered, **struDevIP**, **wPort** and **byChannel** are the IP address, port No., and channel No. of connected device, respectively; **byIvmsChannel** is the digital channel. In the above figure, the channel No.1 of device B1 and the channel No.1, 2 of device B2 are used as channel No.1, 2, 3 of the connected device A; **struDevIP**, **wPort**, and **byChannel** are the IP address, port No. and channel No. of B1 or B2; **byIvmsChannel** is the digital channel No. of device A. E.g., if the behavior analysis alarm of channel No.2 of device B2 is triggered, **struDevIP** is 192.168.0.12, **wPort** is 8002, **byChannel** is 2 and **byIvmsChannel** is 3 in the received alarm message.

## A.39 NET\_VCA\_DURATION

Structure about alarm duration parameters.

### Structure Definition

```
struct{
    WORD    wRelationEventType;
    BYTE    byRes[90];
}NET_VCA_DURATION, *LPNET_VCA_DURATION;
```

### Members

#### wRelationEventType

Behavior analysis alarm/event types, refer to VCA\_RULE\_EVENT\_TYPE\_EX.

#### byRes

Reserved.

## A.40 NET\_VCA\_EVENT\_UNION

Structure about alarm rule parameters.

### Structure Definition

```
union{
    DWORD
    NET_VCA_TRAVERSE_PLANE
    NET_VCA_AREA
    NET_VCA_INTRUSION
    NET_VCA_LOITER
    NET_VCA_TAKE_LEFT
    NET_VCA_PARKING
    NET_VCA_RUN
    NET_VCA_HIGH_DENSITY
    NET_VCA_VIOLENT_MOTION
    NET_VCA_REACH_HIGHT
    NET_VCA_GET_UP
    NET_VCA_LEFT
    NET_VCA_TAKE
    NET_VCA_HUMAN_ENTER
    NET_VCA_OVER_TIME
    NET_VCA_STICK_UP
    NET_VCA_SCANNER
    NET_VCA_LEAVE_POSITION
    NET_VCA_TRAIL
    NET_VCA_FALL_DOWN
    NET_VCA_AUDIO_ABNORMAL

    uLen[23];
    struTraversePlane;
    struArea;
    struIntrusion;
    struLoiter;
    struTakeTeft;
    struParking;
    struRun;
    struHighDensity;
    struViolentMotion;
    struReachHight;
    struGetUp;
    struLeft;
    struTake;
    struHumanEnter;
    struOvertime;
    struStickUp;
    struScanner;
    struLeavePos;
    struTrail;
    struFallDown;
    struAudioAbnormal;
```



```
NET_VCA_ADV_REACH_HEIGHT      struAdvReachHeight;
NET_VCA_TOILET_TARRY          struToiletTarry;
NET_VCA_YARD_TARRY            struYardTarry;
NET_VCA_ADV_TRAVERSE_PLANE    struAdvTraversePlane;
NET_VCA_LECTURE                struLecture;
NET_VCA_ANSWER                 struAnswer;
NET_VCA_STANDUP                struStandUp;
NET_VCA_PEOPLENUM_CHANGE      struPeopleNumChange;
NET_VCA_SPACING_CHANGE        struSpacingChange;
NET_VCA_COMBINED_RULE          struCombinedRule;
NET_VCA_SIT_QUIETLY           struSitQuietly;
NET_VCA_HIGH_DENSITY_STATUS    struHighDensityStatus;
NET_VCA_RUNNING                struRunning;
NET_VCA_RETENTION              struRetention;
NET_VCA_BLACKBOARD_WRITE      struBlackboardWrite;
NET_VCA_SITUATION_ANALYSIS     struSituationAnalysis;
NET_VCA_PLAY_CELLPHONE        struPlayCellphone;
NET_VCA_DURATION              struDruation;
NET_VCA_FAKECARD               struFakeCard
}NET_VCA_EVENT_UNION, *LPNET_VCA_EVENT_UNION;
```

### Members

#### uLen

Union size, 4\*23 (92 bytes)

#### struTraversePlane

Line crossing deteciton parameters, refer to the structure [\*\*NET\\_VCA\\_TRAVERSE\\_PLANE\*\*](#) for details.

#### struArea

Region entrance/exiting detection parameters, refer to the structure [\*\*NET\\_VCA\\_AREA\*\*](#) for details.

#### struIntrusion

Intrusion detection parameters, refer to the structure [\*\*NET\\_VCA\\_INTRUSION\*\*](#) for details.

#### struLoiter

Loitering detection parameters, refer to the structure [\*\*NET\\_VCA\\_LOITER\*\*](#) for details.

#### struTakeTeft

Unattended baggage/object removal detection parameters, refer to the structure [\*\*NET\\_VCA\\_TAKE\\_LEFT\*\*](#) for details.

#### struParking

Parking detection parameters, refer to the structure [\*\*NET\\_VCA\\_PARKING\*\*](#) for details.

#### struRun

Fast moving detection parameters, refer to the structure [\*\*NET\\_VCA\\_RUN\*\*](#) for details.

#### struHighDensity

People gathering detection parameters, alarm will be triggered and uploaded when the people gathering density exceeds the configured threshold. Refer to the structure **NET\_VCA\_HIGH\_DENSITY** for details.

### **struViolentMotion**

Violent motion detection parameters, refer to the structure **NET\_VCA\_VIOLENT\_MOTION** for details.

### **struReachHight**

Climbing detection parameters, refer to the structure **NET\_VCA\_REACH\_HIGHT** for details.

### **struGetUp**

Getting up detection parameters, refer to the structure **NET\_VCA\_GET\_UP** for details.

### **struLeft**

Unattended baggage detection parameters, refer to the structure **NET\_VCA\_LEFT** for details.

### **struTake**

Object removal detection parameters, refer to the structure **NET\_VCA\_TAKE** for details.

### **struHumanEnter**

Region entrance detection parameters, refer to the structure **NET\_VCA\_HUMAN\_ENTER** for details.

### **struOvertime**

Operation overtime detection parameters, refer to the structure **NET\_VCA\_OVER\_TIME** for details.

### **struStickUp**

Sticking-up detection parameters, refer to the structure **NET\_VCA\_STICK\_UP** for details.

### **struScanner**

Card reader parameters, refer to the structure **NET\_VCA\_SCANNER** for details.

### **struLeavePos**

Absence detection parameters, refer to the structure **NET\_VCA\_LEAVE\_POSITION** for details.

### **struTrail**

Trailing detection parameters, refer to the structure **NET\_VCA\_TRAIL** for details.

### **struFallDown**

Falling down detection parameters, refer to the structure **NET\_VCA\_FALL\_DOWN** for details.

### **struAudioAbnormal**

Sudden change of sound intensity detection parameters, refer to the structure **NET\_VCA\_AUDIO\_ABNORMAL** for details.

### **struAdvReachHeight**

Climbing detection parameters, refer to the structure **NET\_VCA\_ADV\_REACH\_HEIGHT** for details.

### **struToiletTarry**

In-toilet overtime detection parameters, refer to the structure [NET\\_VCA\\_TOILET\\_TARRY](#) for details.

### **struYardTarry**

Playground overstay detection parameters, refer to the structure [NET\\_VCA\\_YARD\\_TARRY](#) for details.

### **struAdvTraversePlane**

Line crossing detection parameters, refer to the structure [NET\\_VCA\\_ADV\\_TRAVERSE\\_PLANE](#) for details.

### **struLecture**

Teaching event parameters, refer to the structure [NET\\_VCA\\_LECTURE](#) for details.

### **struAnswer**

Question answer detection parameters, refer to the structure [NET\\_VCA\\_ANSWER](#) for details.

### **struStandUp**

Standing up detection parameters, refer to the structure [NET\\_VCA\\_STANDUP](#) for details.

### **struPeopleNumChange**

People amount changed detection parameters, refer to the structure [NET\\_VCA\\_PEOPLENUM\\_CHANGE](#) for details.

### **struSpacingChange**

Distance changed detection parameters, refer to the structure [NET\\_VCA\\_SPACING\\_CHANGE](#) for details.

### **struCombinedRule**

Combined rule parameters, refer to the structure [NET\\_VCA\\_COMBINED\\_RULE](#) for details.

### **struSitQuietly**

Sedentary detection parameters, refer to the structure [NET\\_VCA\\_SIT\\_QUIETLY](#) for details.

### **struHighDensityStatus**

People gathering status, refer to the structure [NET\\_VCA\\_HIGH\\_DENSITY\\_STATUS](#) for details.  
Device uploads people gathering status according to the configured time interval. This time interval cannot be configured via HCNetsDK, and can only be configured via the server. The default time interval is 10s.

### **struRunning**

Running detection parameters, refer to the structure [NET\\_VCA\\_RUNNING](#) for details.

### **struRetention**

Overstay detection parameters, refer to the structure [NET\\_VCA\\_RETENTION](#) for details.

### **struBlackboardWrite**

Blackboard writing detection parameters, refer to the structure **NET\_VCA\_BLACKBOARD\_WRITE** for details.

### **struSituationAnalysis**

Situation analysis detection parameters, refer to the structure **NET\_VCA\_SITUATION\_ANALYSIS** for details.

### **struPlayCellphone**

Playing mobile phone detection parameters, refer to the structure **NET\_VCA\_PLAY\_CELLPHONE** for details.

### **struDruation**

Alarm duration parameters, refer to the structure **NET\_VCA\_DURATION** for details.

### **struFakeCard**

Fake card detection parameters, refer to the structure **NET\_VCA\_FAKECARD**

## **A.41 NET\_VCA\_FAKECARD**

### **Structure about Fake Card Detection Parameters**

Member	Data Type	Description
<b>struRegion</b>	<b><u>NET_VCA_POLYGON</u></b>	Coordinates of detection area
<b>bySensitivity</b>	BYTE	Sensitivity, value range: [1,5], default value: 3
<b>byRes</b>	BYTE	Reserved, set to 0. The maximum length is 7 bytes.

## **A.42 NET\_VCA\_FALL\_DOWN**

Structure about falling down detection parameters.

### **Structure Definition**

```
struct{
    NET_VCA_POLYGON    struRegion;
    WORD               wDuration;
    BYTE               bySensitivity;
    BYTE               byHeightThreshold;
    BYTE               byRes[4];
}NET_VCA_FALL_DOWN, *LPNET_VCA_FALL_DOWN;
```

### Members

#### struRegion

Coordinates of detection region, refer to the structure [NET\\_VCA\\_POLYGON](#) for details.

#### wDuration

Alarm time duration, which ranges from 1 to 60, unit: second.

#### bySensitivity

Sensitivity, range: [1,5]

#### byHeightThreshold

Height threshold, range: [0,250], default: 90, unit: cm

#### byRes

Reserved, set to 0.

### See Also

[NET\\_VCA\\_EVENT\\_UNION](#)

## A.43 NET\_VCA\_FIELDDETECCION

Intrusion detection parameter structure

### Structure Definition

```
struct{
    DWORD                dwSize;
    BYTE                 byEnable;
    BYTE                 byEnableDualVca;
    BYTE                 byEnableHumanMisinfoFilter;
    BYTE                 byEnableVehicleMisinfoFilter;
    NET_VCA_INTRUSION    struIntrusion[MAX_INTRUSIONREGION_NUM/*8*/];
    NET_DVR_SCHEDTIME    struAlarmSched[MAX_DAYS/*7*/];
    [MAX_TIMESEGMENT_V30/*8*/];
    NET_DVR_HANDLEEXCEPTION_V41 struHandleException;
    DWORD                dwMaxRelRecordChanNum;
    DWORD                dwRelRecordChanNum;
    DWORD                byRelRecordChan[MAX_CHANNUM_V30/*64*/];
    NET_DVR_SCHEDTIME    struHolidayTime[MAX_TIMESEGMENT_V30/*8*/];
    BYTE                 byRes2[100];
}NET_VCA_FIELDDETECCION, *LPNET_VCA_FIELDDETECCION;
```

### Members

#### dwSize

Structure size

**byEnable**

Whether to enable intrusion detection: 0-no, 1-yes

**byEnableDualVca**

Whether to enable dual-VCA: 0-no, 1-yes

**byEnableHumanMisinfoFilter**

Whether to enable preventing false human body alarm: 0-no, 1-yes

**byEnableVehicleMisinfoFilter**

Whether to enable preventing false vehicle alarm filter: 0-no, 1-yes

**struIntrusion**

Region parameters, refer to the structure for details.

**struAlarmSched**

Arming schedule, refer to the structure for details.

**struHandleException**

Alarm linkage types, refer to the structure for details.

**dwMaxRelRecordChanNum**

Maximum number of alarm triggered video channels (read-only).

**dwRelRecordChanNum**

Actual number of alarm triggered video channels.

**byRelRecordChan**

Alarm triggered channel No., which is between 0 and the value of **dwRelRecordChanNum**. For example, if the **dwRelRecordChan** is 5, the available channel No. that can trigger alarm is between **dwRelRecordChan[0]** and **dwRelRecordChan[4]**.

**byRes2**

Reserved, set to 0.

**Remarks**

- When the value of **dwMaxRelRecordChanNum** is larger than 64, the channels will be divided into groups and each group will be numbered (**dwGroup**) in the structure, and up to 64 channels can be contained in a group.
- The video channel No. links with the group No., e.g., when the group No. is 0, the No. of channels in the group is between 1 and 64; when the group No. is 1, the No. of channels in the group is between 65 and 128, and so on. If the group No. is 0xffffffff, it indicates that the No. of channels in current group and the following channels is invalid.
- If the false alarm prevention is enabled, the false alarm will be filtered without uploading, but the false alarm will still be locally stored.

## A.44 NET\_VCA\_FILTER\_STRATEGY

Structure about size filter strategy.

### Structure Definition

```
struct{
    BYTE    byStrategy;
    BYTE    byRes[11];
}NET_VCA_FILTER_STRATEGY, *LPNET_VCA_FILTER_STRATEGY;
```

### Members

#### **byStrategy**

Size filter strategy: 0-disabled, 1-filter by height and width, 2- filter by area

#### **byRes**

Reserved, set to 0

### See Also

## A.45 NET\_VCA\_GET\_UP

Structure about getting up detection parameters.

### Structure Definition

```
struct{
    NET_VCA_POLYGON    struRegion;
    WORD               wDuration;
    BYTE               byMode;
    BYTE               bySensitivity;
    BYTE               byRes[4];
}NET_VCA_GET_UP, *LPNET_VCA_GET_UP;
```

### Members

#### **struRegion**

Coordinates of detection region, refer to the structure for details.

#### **wDuration**

Alarm time duration, which ranges from 1 to 100 s.

#### **byMode**

Standing up detection mode: 0-wide bed mode, 1-bunk bed mode, 2-getting up on wide bed mode

### **bySensitivity**

Sensitivity, range: [1,100]

### **byRes**

Reserved, set to 0

### **See Also**

## **A.46 NET\_VCA\_HIGH\_DENSITY**

Structure about people gathering detection parameters.

### **Structure Definition**

```
struct{
    NET_VCA_POLYGON    struRegion;
    float              fDensity;
    BYTE               bySensitivity;
    BYTE               byRes;
    WORD               wDuration;
}NET_VCA_HIGH_DENSITY, *LPNET_VCA_HIGH_DENSITY;
```

### **Members**

#### **struRegion**

Coordinates of detection region, refer to the structure for details.

#### **fDensity**

Density ratio, range: [0.1, 1.0]

#### **bySensitivity**

Sensitivity, range: [1,5]

#### **byRes**

Reserved, set to 0.

#### **wDuration**

Alarm time duration, which ranges from 20s to 360s.

### **See Also**



## A.47 NET\_VCA\_HIGH\_DENSITY\_STATUS

Structure about people gathering detection parameters.

### Structure Definition

```
struct{
    NET_VCA_POLYGON    struRegion;
    float              fDensity;
    BYTE               bySensitivity;
    BYTE               byRes[3];
}NET_VCA_HIGH_DENSITY_STATUS, *LPNET_VCA_HIGH_DENSITY_STATUS;
```

### Members

#### **struRegion**

Coordinates of detection region, refer to the structure for details.

#### **fDensity**

Density ratio, range: [0.1, 1.0]

#### **bySensitivity**

Sensitivity, range: [0, 100]

#### **byRes**

Reserved, set to 0.

### See Also

## A.48 NET\_VCA\_HUMAN\_ENTER

Structure about region entrance detection parameters.

### Structure Definition

```
struct{
    DWORD             dwRes[23];
}NET_VCA_HUMAN_ENTER, *LPNET_VCA_HUMAN_ENTER;
```

### Members

#### **dwRes**

Reserved, set to 0.

## See Also

## A.49 NET\_VCA\_INTRUSION

Intrusion detection parameter structure

### Structure Definition

```
struct{
    NET_VCA_POLYGON    struRegion;
    WORD               wDuration;
    BYTE               bySensitivity;
    BYTE               byRate;
    BYTE               byDetectionTarget;
    BYTE               byPriority;
    BYTE               byAlarmConfidence;
    BYTE               byRecordConfidence;
}NET_VCA_INTRUSION, *LPNET_VCA_INTRUSION;
```

### Members

#### **struRegion**

Coordinates of region, refer to the structure for details.

#### **wDuration**

Alarm duration, for intelligent server, it is between 1 and 120s (default: 5s); for ATM DVR, it is between 1 and 1200s; for smart network camera, it is between 1 and 100s.

#### **bySensitivity**

Sensitivity, the value is between 1 and 100.

#### **byRate**

Proportion, which equals to region size of all targets without triggering alarm comparing to whole detection region size, the value is normalized from 1 to 100

#### **byDetectionTarget**

Detection target: 0-all (all targets will be detected), 0x01-human, 0x02-vehicle, 0x04-other. Multiple types of targets can be selected, e.g., 0x3-detect vehicle and human.

#### **byPriority**

Alarm priority: 0-low, 1-medium, 2-high.

#### **byAlarmConfidence**

Confidence of alarm notification: 0-low, 1-medium low, 2-medium high, 3-high.

#### **byRecordConfidence**

Confidence of video recording: 0-low, 1-medium low, 2-medium high, 3-high.

### A.50 NET\_VCA\_LEAVE\_POSITION

Structure about absence detection parameters.

#### Structure Definition

```
struct{
    NET_VCA_POLYGON    struRegion;
    WORD               wLeaveDelay;
    WORD               wStaticDelay;
    BYTE               byMode;
    BYTE               byPersonType;
    BYTE               byOnPosition;
    BYTE               byRes;
}NET_VCA_LEAVE_POSITION, *LPNET_VCA_LEAVE_POSITION;
```

#### Members

##### struRegion

Coordinates of detection region, refer to the structure [NET\\_VCA\\_POLYGON](#) for details.

##### wLeaveDelay

Absence alarm time, which ranges from 1 to 1800, unit: second

##### wStaticDelay

Sleeping alarm time, which ranges from 1 to 1800, unit: second

##### byMode

Mode: 0-absence, 1-sleeping, 2-sleeping and absence

##### byPersonType

Number of people on duty: 0-one, 1-two

##### byOnPosition

Number of persons on duty, it is between 1 and 10, the default value is 1.

##### byRes

Reserved, set to 0.

#### See Also

[NET\\_VCA\\_EVENT\\_UNION](#)

## A.51 NET\_VCA\_LECTURE

Structure about teaching detection parameters.

### Structure Definition

```
struct{
    NET_VCA_POLYGON          struRegion;
    WORD                     wDuration;
    BYTE                     bySensitivity;
    BYTE                     byAlarmState;
    BYTE                     byTrackingMode;
    BYTE                     byZoomMode;
    BYTE                     byZoomOver;
    BYTE                     byTrackStatus;
}NET_VCA_LECTURE, *LPNET_VCA_LECTURE;
```

### Members

#### struRegion

Coordinates of detection region, refer to the structure for details.

#### wDuration

Alarm time duration, which ranges from 1 to 10s, and 1s is recommended. It is used to check if the alarm is valid.

#### bySensitivity

Sensitivity, range: [1,100]

#### byAlarmState

Read-only: 0-reserved, 1-alarm started, 2-alarm ended

#### byTrackingMode

Tracking mode, 0-auto (default), 1-horizontal, 2-vertical

#### byZoomMode

Zoom mode: 0-fixed (default), 1-auto

#### byZoomOver

0-reserved, 1-zoomed (alarm is uploaded to education sharing system, which is for image switch).

#### byTrackStatus

Tracking status: 0-reserved, 1-start teaching (A-panorama) 2-tracking (2-close-up), 3-tracking failed (2-panorama).

### See Also

## A.52 NET\_VCA\_LEFT

Structure about unattended baggage detection parameters.

### Structure Definition

```
struct{
    NET_VCA_POLYGON    struRegion;
    WORD               wDuration;
    BYTE               bySensitivity;
    BYTE               byRes[5];
}NET_VCA_LEFT, *LPNET_VCA_LEFT;
```

### Members

#### **struRegion**

Coordinates of detection region, refer to the structure for details.

#### **wDuration**

Alarm time duration, which ranges from 1 to 100 s.

#### **bySensitivity**

Sensitivity, range: [1,5]

#### **byRes**

Reserved, set to 0.

### See Also

## A.53 NET\_VCA\_LINE

Structure about line parameters.

### Structure Definition

```
struct{
    NET_VCA_POINT    struStart;
    NET_VCA_POINT    struEnd;
}NET_VCA_LINE, *LPNET_VCA_LINE;
```

### Members

#### **struStart**

Start point, see details in the structure **NET\_VCA\_POINT**

#### **struEnd**

End point, see details in the structure [\*\*NET\\_VCA\\_POINT\*\*](#) .

### A.54 NET\_VCA\_LOITER

Structure about loitering detection parameters .

#### Structure Definition

```
struct{
    NET_VCA_POLYGON    struRegion;
    WORD               wDuration;
    BYTE               bySensitivity;
    BYTE               byRes[1];
    DWORD              dwLoiterDistance;
}NET_VCA_LOITER, *LPNET_VCA_LOITER;
```

#### Members

##### **struRegion**

Coordinates of detection region, refer to the structure [\*\*NET\\_VCA\\_POLYGON\*\*](#) for details.

##### **wDuration**

Alarm time duration, which ranges from 1s to 120s, and 10s is recommended.

##### **bySensitivity**

Sensitivity, it is between 1 and 5.

##### **byRes**

Reserved, set to 0.

##### **dwLoiterDistance**

Total distance threshold, it is between 100 and 5000, the default value is 1000, unit: cm.

#### See Also

[\*\*NET\\_VCA\\_EVENT\\_UNION\*\*](#)

### A.55 NET\_VCA\_ONE\_RULE\_V42

Structure about alarm rule parameters.

#### Structure Definition

```
struct{
    BYTE               byActive;
    BYTE               byEventPriority;
    BYTE               byBackgroundSuppression;
    BYTE               byRes1[3];
```

```

WORD           wEventType;
BYTE           byRuleName [NAME_LEN/*32*/];
NET_VCA_EVENT_UNION uEventParam;
NET_VCA_SIZE_FILTER struSizeFilter;
NET_DVR_SCHEDTIME struAlarmTime [MAX_DAYS/*7*/];
[MAX_TIMESEGMENT_V30/*8*/];
NET_DVR_HANDLEEXCEPTION_V41 struAlarmHandleType;
DWORD          dwRelRecordChan [MAX_CHANNUM_V30/*64*/];
WORD           wAlarmDelay;
BYTE           byRes2 [2];
NET_VCA_FILTER_STRATEGY struFilterStrategy;
NET_VCA_RULE_TRIGGER_PARAM struTriggerParam;
BYTE           byRes [32];
}NET_VCA_ONE_RULE_V42, *LPNET_VCA_ONE_RULE_V42;

```

## Members

### byActive

Whether to enable alarm rule: 0-no, other value-yes

### byEventPriority

Event priority: 0-low, 1-medium, 2-high

### byBackgroundSuppression

Background suppression: 0-disable, 1-enable, 2-"self-adaption"

### byRes1

Reserved, set to 0.

### wEventType

Event type, see the structure [\*\*VCA\\_RULE\\_EVENT\\_TYPE\\_EX\*\*](#) for details.

### byRuleName

Rule name

### uEventParam

Behavior analysis event parameter, see the structure [\*\*NET\\_VCA\\_EVENT\\_UNION\*\*](#) for details.

### struSizeFilter

Size filter, see the structure [\*\*NET\\_VCA\\_SIZE\\_FILTER\*\*](#) for details.

### struAlarmTime

Arming time, see the structure [\*\*NET\\_DVR\\_SCHEDTIME\*\*](#) for details.

### struAlarmHandleType

Alarm/event linkage actions, see the structure [\*\*NET\\_DVR\\_HANDLEEXCEPTION\\_V41\*\*](#) for details.

### dwRelRecordChan

Alarm triggered recording channel. The channel number is bound with group No., i.e., when the group No. is 0, the related channel No. ranges from 1 to 64; when the group No. is 1, the related

channel No. ranges from 65 to 128. 0xffffffff indicates the current and following related channel No. is invalid in the group.

### **wAlarmDelay**

Alarm delayed time: "0"-5s, "1"-10s, "2"-30s, "3"-60s, "4"-120s, "5"-300s, "6"-600s

### **byRes2**

Reserved, set to 0

### **struFilterStrategy**

Size filter strategy, see the structure **NET\_VCA\_FILTER\_STRATEGY** for details.

### **struTriggerParam**

Rule triggering parameter, see the structure **NET\_VCA\_RULE\_TRIGGER\_PARAM** for details.

### **byRes**

Reserved, set to 0

## **Remarks**

The video channel No. links with the group No., e.g., when the group No. is 0, the No. of channels in the group is between 1 and 64; when the group No. is 1, the No. of channels in the group is between 65 and 128, and so on. If the group No. is 0xffffffff, it indicates that the No. of channels in current group and the following channels is invalid.

## **See Also**

**NET\_VCA\_RULECFG\_V42**

## **A.56 NET\_VCA\_OVER\_TIME**

Structure about operation overtime alarm parameters.

### **Structure Definition**

```
struct{
    NET_VCA_POLYGON      struRegion;
    WORD                 wDuration;
    BYTE                 byRes[6];
}NET_VCA_OVER_TIME, *LPNET_VCA_OVER_TIME;
```

## **Members**

### **struRegion**

Coordinates of detection region, refer to the structure for details.

### **wDuration**

Alarm time duration



### **byRes**

Reserved, set to 0.

### **See Also**

## **A.57 NET\_VCA\_PARKING**

Structure about parking detection parameters.

### **Structure Definition**

```
struct{
    NET_VCA_POLYGON    struRegion;
    WORD               wDuration;
    BYTE               byRes[6];
}NET_VCA_PARKING, *LPNET_VCA_PARKING;
```

### **Members**

#### **struRegion**

Coordinates of detection region, refer to the structure for details.

#### **wDuration**

Alarm time duration, which ranges from 1s to 120s, and 10s is recommended.

#### **byRes**

Reserved, set to 0.

### **See Also**

## **A.58 NET\_VCA\_PEOPLENUM\_CHANGE**

Structure about people amount changed alarm parameters.

### **Structure Definition**

```
struct{
    NET_VCA_POLYGON    struRegion;
    BYTE               bySensitivity;
    BYTE               byPeopleNumThreshold;
    BYTE               byDetectMode;
    BYTE               byNoneStateEffective;
    WORD               wDuration;
```

```
    BYTE                byRes[2];  
}NET_VCA_PEOPLENUM_CHANGE, *LPNET_VCA_PEOPLENUM_CHANGE;
```

### Members

#### struRegion

Coordinates of detection region, refer to the structure for details.

#### bySensitivity

Sensitivity, range: [1,100]

#### byPeopleNumThreshold

Number threshold of people, range: [0,5], default: 1

#### byDetectMode

Detection mode, compared with number threshold of people: 1-more than, 2-less than, 3-equal to, 4-not equal to

#### byNoneStateEffective

Whether the no person status is valid: 0-no, 1-yes

#### wDuration

Event time duration, unit: second, which ranges from 1 to 3600, default: 2

#### byRes

Reserved, set to 0.

### See Also

## A.59 NET\_VCA\_PLAY\_CELLPHONE

Structure about playing mobile phone detection parameters.

### Structure Definition

```
struct{  
    NET_VCA_POLYGON  struRegion;  
    WORD             wDuration;  
    BYTE             byRes[6];  
}NET_VCA_PLAY_CELLPHONE, *LPNET_VCA_PLAY_CELLPHONE;
```

### Members

#### struRegion

Coordinates of detection area, refer to the structure [NET\\_VCA\\_POLYGON](#) for details.

#### wDuration

Duration time of playing mobile phone, range: [1,600], default value: 20, unit: second.

### byRes

Reserved.

## A.60 NET\_VCA\_POINT

### Structure About Point Coordinates Parameters

Member	Data Type	Description
fX	float	X-coordinate, it is a normalized value ranging from 0.000 to 1. The floating-point number is the percentage of the current image size and is accurate to three decimal places.
fY	float	Y-coordinate, it is a normalized value ranging from 0.000 to 1. The floating-point number is the percentage of the current image size and is accurate to three decimal places.

## A.61 NET\_VCA\_POLYGON

### Polygon Coordinate Parameter Structure

Member	Data Type	Description
dwPointNum	DWORD	Valid point (larger than or equal 3), if 3 points are in the same line, as well as line-cross, region is invalid.
struPos	Array of <u>NET_VCA_POINT</u>	Boundary point of polygon, up to 10 points

## A.62 NET\_VCA\_REACH\_HIGHT

Structure about climbing detection parameters.

### Structure Definition

```
struct{  
    NET_VCA_LINE    struVcaLine;
```

```
WORD          wDuration;  
BYTE          byRes[6];  
}NET_VCA_REACH_HIGHT, *LPNET_VCA_REACH_HIGHT;
```

## Members

### struVcaLine

Coordinates of detection line, refer to the structure **NET\_VCA\_LINE** for details.

### wDuration

Alarm time duration, which ranges from 1 to 120, unit: second.

### byRes

Reserved, set to 0.

## See Also

## A.63 NET\_VCA\_RECT

### Structure About Rectangle Region Coordinate Parameters

Member	Data Type	Description
fX	float	X-coordinate of frame's upper-left corner, it ranges from 0.000 to 1.
fY	float	Y-coordinate of frame' upper-left corner, it ranges from 0.000 to 1.
fWidth	float	Frame width, it ranges from 0.000 to 1.
fHeight	float	Frame height, it ranges from 0.000 to 1.

## A.64 NET\_VCA\_RELATE\_RULE\_PARAM

Rule parameter structure

### Structure Definition

```
struct{  
    BYTE      byRuleID;  
    BYTE      byRes;  
    WORD      wEventType;  
}NET_VCA_RELATE_RULE_PARAM, *LPNET_VCA_RELATE_RULE_PARAM;
```

### Members

#### byRuleID

Rule No., 0-none

#### byRes

Reserved, set to 0.

#### wEventType

Behavior analysis event types, refer to the enumeration definition in [VCA\\_RULE\\_EVENT\\_TYPE\\_EX](#) for details.

### See Also

## A.65 NET\_VCA\_RETENTION

Structure about the overstay detection parameters.

```
struct{
    NET_VCA_POLYGON    struRegion;
    WORD               wDuration;
    BYTE               byRes[6];
}NET_VCA_RETENTION, *LPNET_VCA_RETENTION;
```

### Members

#### struRegion

Area range, refer to the structure [NET\\_VCA\\_POLYGON](#) for details.

#### wDuration

Time threshold for triggering overstay alarms, it is between 60 and 3600, the default value is 1800, unit: second.

#### byRes

Reserved.

## A.66 NET\_VCA\_RULECFG\_V42

Behavior analysis parameter structure.

### Structure Definition

```
struct{
    DWORD              dwSize;
    BYTE              byPicProType;
```

```
BYTE          byUpLastAlarm;
BYTE          byPicRecordEnable;
BYTE          byRes1;
NET_DVR_JPEGPARAM struPicParam;
NET_VCA_ONE_RULE_V42 struRule[MAX_RULE_NUM_V42/*16*/];
WORD          wRelSnapChan[MAX_REL_SNAPCHAN_NUM/*3*/];
BYTE          byTrackEnable;
BYTE          byRes2;
NET_DVR_PTZ_POSITION struPTZPosition;
WORD          wTrackDuration;
BYTE          byRes[62];
}NET_VCA_RULECFG_V42,*LPNET_VCA_RULECFG_V42;
```

### Members

#### dwSize

Structure size

#### byPicProType

Whether to process the alarm picture: 0-no, other values-yes

#### byUpLastAlarm

Whether to upload the latest alarm: 0-no, 1-yes

#### byPicRecordEnable

Whether to save picture: 0-no, 1-yes

#### byRes1

Reserved, set to 0.

#### struPictureParam

Picture size and quality, see details in the structure [NET\\_DVR\\_JPEGPARAM](#).

#### struRule

Rule array, each array refers to the rule configuration of one event type, see details in the structure [NET\\_VCA\\_ONE\\_RULE\\_V42](#).

#### wRelSnapChan

Whether to link channel for capturing when alarm is triggered: 0-no, other values-yes.

#### byTrackEnable

Whether to enable tracking: 0-no, 1-yes

#### byRes2

Reserved, set to 0.

#### struPTZPosition

Scene information, see details in the structure .

#### wTrackDuration

Tracking duration, unit: s

**byRes**

Reserved, set to 0.

## A.67 NET\_VCA\_RULE\_ALARM

Behavior analysis alarm parameter structure.

### Structure Definition

```
struct{
    DWORD                dwSize;
    DWORD                dwRelativeTime;
    DWORD                dwAbsTime;
    NET_VCA_RULE_INFO    struRuleInfo;
    NET_VCA_TARGET_INFO  struTargetInfo;
    NET_VCA_DEV_INFO     struDevInfo;
    DWORD                dwPicDataLen;
    BYTE                 byPicType;
    BYTE                 byRelAlarmPicNum;
    BYTE                 bySmart;
    BYTE                 byPicTransType;
    DWORD                dwAlarmID;
    WORD                 wDevInfoIvmsChannelEx;
    BYTE                 byRelativeTimeFlag;
    BYTE                 byAppendInfoUploadEnabled;
    BYTE                 *pAppendInfo;
    BYTE                 *pImage;
}NET_VCA_RULE_ALARM, *LPNET_VCA_RULE_ALARM;
```

### Members

**dwSize**

Structure size

**dwRelativeTime**

Relative time duration, which is between behavior analysis enabled time and alarm triggered time. It is the UTC time when **byRelativeTimeFlag** is "1".

**dwAbsTime**

Absolute time duration, it is the OSD time.

**struRuleInfo**

Event rule information, see details in the structure [NET\\_VCA\\_RULE\\_INFO](#) .

**struTargetInfo**

Alarm target information, see details in the structure [NET\\_VCA\\_TARGET\\_INFO](#) .

**struDevInfo**

Front-end device information, see details in the structure **NET\_VCA\_DEV\_INFO**.

**dwPicDataLen**

Returned picture size: 0-no picture, other values-with picture.

**byPicType**

0-captured picture, 1-picture for comparison

**byRelAlarmPicNum**

The number of alarm pictures of linked channel, the actual total number equal to **byRelAlarmPicNum+1**.

**bySmart**

0-returned by intelligent devices, 1-returned by Smart devices.

**byPicTransType**

Picture data transmission mode: 0-binary, 1-URL

**dwAlarmID**

Alarm ID, 0-invalid. It is used to distinguish the alarms when the alarm type is same.

**wDevInfolvmsChannelEx**

Channel No. of platform or system that connecting with device.

**byRelativeTimeFlag**

Whether the time difference parameter is valid: 0-invalid, 1-valid.

**byAppendInfoUploadEnabled**

Whether to upload additional information: 0-no, 1-yes.

**pAppendInfo**

Additional information pointer, see details in the structure **NET\_VCA\_APPEND\_INFO**, it is valid only when **byAppendInfoUploadEnabled** is 1.

**pImage**

Picture pointer

### Remarks

The time is parsed from the absolute time according to the following algorithm:

```
#define GET_YEAR(_time_)      (((_time_)>>26) + 2000)
#define GET_MONTH(_time_)    (((_time_)>>22) & 15)
#define GET_DAY(_time_)      (((_time_)>>17) & 31)
#define GET_HOUR(_time_)     (((_time_)>>12) & 31)
#define GET_MINUTE(_time_)   (((_time_)>>6) & 63)
#define GET_SECOND(_time_)   (((_time_)>>0) & 63)
```



## A.68 NET\_VCA\_RULE\_INFO

Structure about alarm/event rule information.

### Structure Definition

```
struct{
    BYTE                byRuleID;
    BYTE                bySceneID;
    WORD                wEventTypeEx;
    BYTE                byRuleName [NAME_LEN/*32*/];
    DWORD               dwEventType;
    NET_VCA_EVENT_UNION uEventParam;
}NET_VCA_RULE_INFO, *LPNET_VCA_RULE_INFO;
```

### Members

#### byRuleID

Rule No., which ranges from 0 to 7.

#### bySceneID

Scene ID, 0-invalid

#### wEventTypeEx

Extension types of behavior analysis, refer to the enumeration definition **VCA\_RULE\_EVENT\_TYPE\_EX** for details.

#### byRuleName

Rule name

#### dwEventType

Behavior analysis event types, refer to the enumeration definition **VCA\_EVENT\_TYPE** for details.

#### uEventParam

Event parameters, see the structure **NET\_VCA\_EVENT\_UNION** for details.

### Remarks

The member **wEventTypeEx** compile with the member **dwEventType**.

### See Also

**NET\_VCA\_RULE\_ALARM**

## A.69 NET\_VCA\_RULE\_TRIGGER\_PARAM

Structure about rule trigger parameters.

### Structure Definition

```
struct{
    BYTE        byTriggerMode;
    BYTE        byTriggerPoint;
    BYTE        byRes1[2];
    float       fTriggerArea;
    BYTE        byRes2[4];
}NET_VCA_RULE_TRIGGER_PARAM, *LPNET_VCA_RULE_TRIGGER_PARAM;
```

### Members

#### **byTriggerMode**

Trigger mode: 0-disable, 1-trigger by preset, 2-trigger by target region size

#### **byTriggerPoint**

Trigger point: 0-medium, 1-up, 2-down. It is valid when **byTriggerMode** is 1.

#### **byRes1**

Reserved, set to 0

#### **fTriggerArea**

Percentage of the triggered target region, it ranges from 0 to 100. It is valid when **byTriggerMode** is 2

#### **byRes2**

Reserved, set to 0

### See Also

## A.70 NET\_VCA\_RUN

Structure about fast moving detection parameters.

### Structure Definition

```
struct{
    NET_VCA_POLYGON    struRegion;
    float              fRunDistance;
    BYTE               bySensitivity;
    BYTE               byMode;
    BYTE               byDetectionTarget;
    BYTE               byRes;
}NET_VCA_RUN, *LPNET_VCA_RUN;
```

### Members

#### **struRegion**

Coordinates of detection region, refer to the structure for details.

#### **fRunDistance**

Maximum running speed, the value ranges from 0.1 m/s to 100 m/s for pixel mode, and the value ranges from 1 m/s to 20 m/s for actual mode.

#### **bySensitivity**

Sensitivity, range: [1,5]

#### **byMode**

Running mode: 0-pixel mode, 1-actual mode

#### **byDetectionTarget**

Detection target: 0-all (all targets will be detected), 0x01-human, 0x02-vehicle. Multiple types of targets can be selected, e.g., 0x3-detect vehicle and human.

#### **byRes**

Reserved, set to 0.

### See Also

## A.71 NET\_VCA\_RUNNING

Structure about the running detection parameters.

```
struct{
    NET_VCA_POLYGON    struRegion;
    DWORD              dwSpeed;
    BYTE               byRes[4];
}NET_VCA_RUNNING, *LPNET_VCA_RUNNING;
```

### Members

#### **struRegion**

Area range, refer to the structure **NET\_VCA\_POLYGON** for details.

#### **dwSpeed**

Running speed, it is between 1 and 10.

#### **byRes**

Reserved.

## A.72 NET\_VCA\_SCANNER

Structure about card reader parameters.

### Structure Definition

```
struct{
    NET_VCA_POLYGON    struRegion;
    WORD               wDuration;
    BYTE               bySensitivity;
    BYTE               byRes[5];
}NET_VCA_SCANNER, *LPNET_VCA_SCANNER;
```

### Members

#### **struRegion**

Coordinates of detection region, refer to the structure for details.

#### **wDuration**

Card reading duration, which ranges from 4 s to 60 s, and 10s is recommended.

#### **bySensitivity**

Sensitivity, range: [1,5]

#### **byRes**

Reserved, set to 0.

### See Also

## A.73 NET\_VCA\_SITUATION\_ANALYSIS

Structure about situation analysis detection parameters.

### Structure Definition

```
struct{
    NET_VCA_POLYGON    struRegion;
    WORD               wPeopleNum;
    BYTE               byRes[6];
}NET_VCA_SITUATION_ANALYSIS, *LPNET_VCA_SITUATION_ANALYSIS;
```

### Members

#### **struRegion**

Coordinates of detection area, refer to the structure **NET\_VCA\_POLYGON** for details.

### wPeopleNum

Number of people in the detection area.

### byRes

Reserved.

## A.74 NET\_VCA\_SIT\_QUIETLY

Structure about sedentary detection parameters.

### Structure Definition

```
struct{
    NET_VCA_POLYGON    struRegion;
    DWORD              dwDuration;
    BYTE               byRes[4];
}NET_VCA_SIT_QUIETLY, *LPNET_VCA_SIT_QUIETLY;
```

### Members

#### struRegion

Coordinates of detection region, refer to the structure for details.

#### dwDuration

Event time duration, unit: second, which ranges from 1 to 3600

#### byRes

Reserved, set to 0.

### See Also

## A.75 NET\_VCA\_SIZE\_FILTER

Structure about size filter parameters.

### Structure Definition

```
struct{
    BYTE                byActive;
    BYTE                byMode;
    BYTE                byRes[3];

    NET_VCA_RECT
    struMiniRect;
```

```
                                NET_VCA_RECT
                                struMaxRect;
}NET_VCA_SIZE_FILTER, *LPNET_VCA_SIZE_FILTER;
```

### Members

#### byActive

Whether to enable size filter: 0-no, other value-yes

#### byMode

Filter mode, see the definitions below:

```
enum _VCA_SIZE_FILTER_MODE_{
    IMAGE_PIX_MODE,
    REAL_WORLD_MODE,
    DEFAULT_MODE
}SIZE_FILTER_MODE
```

#### IMAGE\_PIX\_MODE

Set by pixel

#### REAL\_WORLD\_MODE

Set by actual size

#### DEFAULT\_MODE

Default mode, it is only supported by ATM

#### byRes

Reserved, set to 0

#### struMiniRect

Minimum target frame, it ranges from 0 to 50 in REAL\_WORLD\_MODE mode, 0-not set, unit: m.

#### struMaxRect

Maximum target frame, it ranges from 0 to 50 in REAL\_WORLD\_MODE mode, 0-not set, unit: m.

## A.76 NET\_VCA\_SPACING\_CHANGE

Structure about distance changed detection parameters.

### Structure Definition

```
struct{
    NET_VCA_POLYGON      struRegion;
    float                fSpacingThreshold;
    BYTE                 bySensitivity;
    BYTE                 byDetectMode;
    WORD                 wDuration;
}NET_VCA_SPACING_CHANGE, *LPNET_VCA_SPACING_CHANGE;
```

### Members

#### **struRegion**

Coordinates of detection region, refer to the structure for details.

#### **fSpacingThreshold**

Distance threshold, range: [0,10.0], default: 1.0, unit: m

#### **bySensitivity**

Sensitivity, range: [1,100]

#### **byDetectMode**

Detection mode, compared with distance threshold: 1-more than, 2-less than

#### **wDuration**

Event time duration, unit: second, which ranges from 1 to 3600, default: 2

### See Also

## A.77 NET\_VCA\_STANDUP

Structure about standing up detection parameters.

### Structure Definition

```
struct{
    NET_VCA_POLYGON      struRegion;
    BYTE                 bySensitivity;
    BYTE                 byHeightThreshold;
    WORD                 wDuration;
    BYTE                 byRes[4];
}NET_VCA_STANDUP, *LPNET_VCA_STANDUP;
```

### Members

#### **struRegion**

Coordinates of detection region, refer to the structure for details.

#### **bySensitivity**

Sensitivity, range: [1,100]

#### **byHeightThreshold**

Height threshold, range: [0,250], default: 130, unit: cm

#### **wDuration**

Event time duration, unit: second, which ranges from 1 to 3600, default: 2

**byRes**

Reserved, set to 0.

**See Also**

## A.78 NET\_VCA\_STICK\_UP

Structure about sticking up detection parameters.

**Structure Definition**

```
struct{
    NET_VCA_POLYGON    struRegion;
    WORD               wDuration;
    BYTE               bySensitivity;
    BYTE               byRes[5];
}NET_VCA_STICK_UP, *LPNET_VCA_STICK_UP;
```

**Members****struRegion**

Coordinates of detection region, refer to the structure for details.

**wDuration**

Alarm time duration, which ranges from 4 s to 60 s, and 10s is recommended.

**bySensitivity**

Sensitivity, range: [1,5]

**byRes**

Reserved, set to 0.

**See Also**

## A.79 NET\_VCA\_TAKE

Structure about object removal detection parameters.

**Structure Definition**

```
struct{
    NET_VCA_POLYGON    struRegion;
    WORD               wDuration;
    BYTE               bySensitivity;
```



```
    BYTE                byRes[5];  
}NET_VCA_TAKE, *LPNET_VCA_TAKE;
```

### Members

#### **struRegion**

Coordinates of detection region, refer to the structure for details.

#### **wDuration**

Alarm time duration, which ranges from 10 s to 100 s.

#### **bySensitivity**

Sensitivity, range: [1,5]

#### **byRes**

Reserved, set to 0.

### See Also

## A.80 NET\_VCA\_TAKE\_LEFT

Structure about unattended baggage/object removal detection parameters.

### Structure Definition

```
struct{  
    NET_VCA_POLYGON    struRegion;  
    WORD               wDuration;  
    BYTE               byRes[6];  
}NET_VCA_TAKE_LEFT, *LPNET_VCA_TAKE_LEFT;
```

### Members

#### **struRegion**

Coordinates of detection region, refer to the structure for details.

#### **wDuration**

Alarm time duration, which ranges from 1s to 120s, and 10s is recommended.

#### **byRes**

Reserved, set to 0.

### See Also

## A.81 NET\_VCA\_TARGET\_INFO

Structure about alarm target information.

### Structure Definition

```
struct{
    DWORD          dwID;
    NET_VCA_RECT    struRect;
    BYTE           byRes[4];
}NET_VCA_TARGET_INFO, *LPNET_VCA_TARGET_INFO;
```

### Members

#### dwID

Target ID, it is 0 when the people gathering density is too high and the alarm is triggered

#### struRect

Target frame.

#### byRes

Reserved, set to 0.

## A.82 NET\_VCA\_TOILET\_TARRY

Structure about in-toilet overtime detection parameters.

### Structure Definition

```
struct{
    NET_VCA_POLYGON    struRegion;
    WORD               wDelay;
    BYTE               byRes[6];
}NET_VCA_TOILET_TARRY, *LPNET_VCA_TOILET_TARRY;
```

### Members

#### struRegion

Coordinates of detection region, refer to the structure for details.

#### wDelay

In-toilet overtime, unit: second, range: [1,120]

#### byRes

Reserved, set to 0.

## See Also

### A.83 NET\_VCA\_TRAIL

Structure about trailing detection parameters.

#### Structure Definition

```
struct{
    NET_VCA_POLYGON    struRegion;
    WORD               wRes;
    BYTE               bySensitivity;
    BYTE               byRes[5];
}NET_VCA_TRAIL, *LPNET_VCA_TRAIL;
```

#### Members

##### **struRegion**

Coordinates of detection region, refer to the structure for details.

##### **wRes**

Reserved.

##### **bySensitivity**

Sensitivity, range: [1,5]

##### **byRes**

Reserved, set to 0.

## See Also

### A.84 NET\_VCA\_TRAVERSE\_PLANE

#### Structure About Line Crossing Detection Parameters

Member	Data Type	Description
<b>struPlaneBottom</b>	<u><i>NET_VCA_LINE</i></u>	Detection 2D line bottom.
<b>dwCrossDirection</b>	<u><i>VCA_CROSS_DIRECTION</i></u>	Line crossing direction: 0, 1, 2.

Member	Data Type	Description
<b>bySensitivity</b>	BYTE	Sensitivity, range: [1, 5] (for smart network camera, the range is [1,100]).
<b>byPlaneHeight</b>	BYTE	Detection 2D line height (this member is not supported by network camera).
<b>byDetectionTarget</b>	BYTE	Objects to be detected: 0-all targets, 0x01-human, 0x02-vehicle.
<b>byPriority</b>	BYTE	Priority: 0-low, 1-medium, 2-high.
<b>byAlarmConfidence</b>	BYTE	Confidence of alarm notification: 0-low, 1-medium low, 2-medium high, 3-high.
<b>byRecordConfidence</b>	BYTE	Confidence of video recording: 0-low, 1-medium low, 2-medium high, 3-high.
<b>byRes2</b>	Array of BYTE	Reserved, set to 0. The maximum size is 34 bytes.

## A.85 NET\_VCA\_TRAVERSE\_PLANE\_DETECTION

Line crossing detection parameter structure

### Structure Definition

```

struct{
    DWORD                dwSize;
    BYTE                byEnable;
    BYTE                byEnableDualVca;
    BYTE                byEnableHumanMisinfoFilter;
    BYTE                byEnableVehicleMisinfoFilter;
    NET_VCA_TRAVERSE_PLANE    struAlertParam[MAX_ALERTLINE_NUM/*8*/];
    NET_DVR_SCHEDTIME    struAlarmSched[MAX_DAYS][MAX_TIMESEGMENT_V30/*8*/];
    NET_DVR_HANDLEEXCEPTION_V41    struHandleException;
    DWORD                dwMaxRelRecordChanNum;
    DWORD                dwRelRecordChanNum;
    DWORD                byRelRecordChan[MAX_CHANNUM_V30/*64*/];
    NET_DVR_SCHEDTIME    struHolidayTime[MAX_TIMESEGMENT_V30/*8*/];
    BYTE                byRes2[100];
}NET_VCA_TRAVERSE_PLANE_DETECTION,*LPNET_VCA_TRAVERSE_PLANE_DETECTION;

```

### Members

**dwSize**

Structure size.

### **byEnable**

Whether to enable line crossing detection: 0-no, 1-yes

### **byEnableDualVca**

Whether to enable dual-VCA: 0-no, 1-yes

### **byEnableHumanMisinfoFilter**

Whether to enable false human body alarm prevention: 0-no, 1-yes

### **byEnableVehicleMisinfoFilter**

Whether to enable false vehicle alarm prevention: 0-no, 1-yes

### **struAlertParam**

Detection line parameters, refer to the structure **NET\_VCA\_TRAVERSE\_PLANE** for details.

### **struAlarmSched**

Arming schedule, refer to the structure **NET\_DVR\_SCHEDTIME** for details.

### **struHandleException**

Alarm linkage types, refer to the structure **NET\_DVR\_HANDLEEXCEPTION\_V41** for details.

### **dwMaxRelRecordChanNum**

Maximum number of alarm triggered video channels (read-only).

### **dwRelRecordChanNum**

Actual number of alarm triggered video channels.

### **byRelRecordChan**

Alarm triggered channel No., which is between 0 and the value of **dwRelRecordChanNum**. For example, if the **dwRelRecordChan** is 5, the available channel No. that can trigger alarm is between **dwRelRecordChan[0]** and **dwRelRecordChan[4]**.

### **struHolidayTime**

Holiday schedule, up to 8 time periods can be configured, refer to the structure **NET\_DVR\_SCHEDTIME** for details.

### **byRes2**

Reserved, set to 0.

## **Remarks**

- When the value of **dwMaxRelRecordChanNum** is larger than 64, the channels will be divided in to groups and each group will be numbered (**dwGroup**) in the structure **NET\_DVR\_CHANNEL\_GROUP**, and up to 64 channels can be contained in a group.
- The video channel No. links with the group No., e.g., when the group No. is 0, the No. of channels in the group is between 1 and 64; when the group No. is 1, the No. of channels in the

group is between 65 and 128, and so on. If the group No. is 0xffffffff, it indicates that the No. of channels in current group and the following channels is invalid.

- If the false alarm prevention is enabled, the false alarm will be filtered without uploading, but the false alarm will still be locally stored.

### A.86 NET\_VCA\_VIOLENT\_MOTION

Structure about violent motion detection parameters.

#### Structure Definition

```
struct{
    NET_VCA_POLYGON      struRegion;
    WORD                 wDuration;
    BYTE                 bySensitivity;
    BYTE                 byMode;
    BYTE                 byRes[4];
}NET_VCA_VIOLENT_MOTION, *LPNET_VCA_VIOLENT_MOTION;
```

#### Members

##### **struRegion**

Coordinates of detection region, refer to the structure **NET\_VCA\_POLYGON** for details.

##### **wDuration**

Alarm time duration, which ranges from 1 to 50, unit: second.

##### **bySensitivity**

Sensitivity, range: [1,5]

##### **byMode**

Mode: 0-video, 1-video and audio, 2-audio

##### **byRes**

Reserved, set to 0.

#### See Also

**NET\_VCA\_EVENT\_UNION**

### A.87 NET\_VCA\_YARD\_TARRY

Structure about playground overstay detection parameters.

## Structure Definition

```
struct{  
    NET_VCA_POLYGON      struRegion;  
    WORD                 wDelay;  
    BYTE                 byRes[6];  
}NET_VCA_YARD_TARRY, *LPNET_VCA_YARD_TARRY;
```

## Members

### **struRegion**

Coordinates of detection region, refer to the structure for details.

### **wDelay**

Playground overstay time, unit: second, range: [1,120]

### **byRes**

Reserved, set to 0.

## See Also

## A.88 TRAFFIC\_SCENE\_MODE

### Enumeration about Intelligent Traffic Scene

Enumeration Type	Macro Definition Value	Description
<b>FREEWAY</b>	0	Highway outdoor
<b>TUNNEL</b>		Highway tunnel
<b>BRIDGE</b>		Highway bridge

## A.89 TFS\_CHAN\_MODE\_TYPE

### Enumeration about Traffic Violation Enforcement Mode

Enumeration Type	Macro Definition Value	Description
<b>TFS_CITYROAD</b>	0	Cityroad
<b>TFS_FREEWAY</b>	1	Highway

## A.90 VCA\_ABILITY\_TYPE

### Enumeration About Behavior Analysis Types

Enumeration Type	Macro Definition Value	Description
TRAVERSE_PLANE_ABILITY	0x01	Line crossing
ENTER_AREA_ABILITY	0x02	Enter region
EXIT_AREA_ABILITY	0x04	Exit region
INTRUSION_ABILITY	0x08	Intrusion
LOITER_ABILITY	0x10	Loitering
LEFT_TAKE_ABILITY	0x20	Unattended baggage or object removal
PARKING_ABILITY	0x40	Parking
RUN_ABILITY	0x80	Running
HIGH_DENSITY_ABILITY	0x100	People gathering density
LF_TRACK_ABILITY	0x200	Speed dome tracking
VIOLENT_MOTION_ABILITY	0x400	Violent motion detection
REACH_HIGHT_ABILITY	0x800	Climbing detection
GET_UP_ABILITY	0x1000	Getting up detection
LEFT_ABILITY	0x2000	Unattended baggage detection
TAKE_ABILITY	0x4000	Object removal detection
LEAVE_POSITION	0x8000	Absence
TRAIL_ABILITY	0x10000	Tailing
KEY_PERSON_GET_UP_ABILITY	0x20000	Key person getting up detection
STANDUP_ABILITY	0x40000	Standing up
FALL_DOWN_ABILITY	0x80000	Falling down



Enumeration Type	Macro Definition Value	Description
AUDIO_ABNORMAL_ABILITY	0x100000	Sudden change of sound density
ADV_REACH_HEIGHT_ABILITY	0x200000	Climbing
TOILET_TARRY_ABILITY	0x400000	Toilet overtime
YARD_TARRY_ABILITY	0x800000	Playground overstay
ADV_TRAVERSE_PLANE_ABILITY	0x1000000	Virtual line
HUMAN_ENTER_ABILITY	0x10000000	People approaching ATM, supported only in ATM_PANEL mode
OVER_TIME_ABILITY	0x20000000	Operation timed out, supported only in ATM_PANEL mode
STICK_UP_ABILITY	0x40000000	Sticking, support area rule
INSTALL_SCANNER_ABILITY	0x80000000	Installing card reader, support area rule

## A.91 VCA\_ABILITY\_TYPE\_EX

### Enumeration about Behavior Analysis Types

Enumeration Type	Macro Definition Value	Description
PEOPLENUM_CHANGE_ABILITY	0x00000002	Detection of people quantity changing
SPACING_CHANGE_ABILITY	0x00000004	Distance changing detection
EVENT_COMBINED_ABILITY	0x00000008	Combined rule event
EVENT_SIT_QUIETLY	0x00000010	Sitting quietly event
EVENT_HIGH_DENSITY_STATUS_ABILITY	0x00000020	People gathering status
EVENT_RUNNING	0x00000040	Running

Enumeration Type	Macro Definition Value	Description
<b>EVENT_RETENTION</b>	0x00000080	Overstay detection
<b>EVENT_TEACHER_WRITING</b>	0x00000100	Blackboard writing
<b>EVENT_FAKECARD</b>	0x00000200	Fake card detection

## A.92 VCA\_CHAN\_ABILITY\_TYPE

Enumerate the intelligent channel types.

```
enum _VCA_CHAN_ABILITY_TYPE_{
    VCA_BEHAVIOR_BASE                = 1,
    VCA_BEHAVIOR_ADVANCE              = 2,
    VCA_BEHAVIOR_FULL                 = 3,
    VCA_PLATE                        = 4,
    VCA_ATM                          = 5,
    VCA_PDC                          = 6,
    VCA_ITS                          = 7,
    VCA_BEHAVIOR_PRISON               = 8,
    VCA_FACE_SNAP                    = 9,
    VCA_FACE_SNAPRECOG               = 10,
    VCA_FACE_RETRIEVAL               = 11,
    VCA_FACE_RECOG                   = 12,
    VCA_BEHAVIOR_PRISON_PERIMETER     = 13,
    VCA_TPS                          = 14,
    VCA_TFS                          = 15,
    VCA_BEHAVIOR_FACESNAP             = 16,
    VCA_HEATMAP                      = 17,
    VCA_SMART_VEHICLE_DETECTION       = 18,
    VCA_SMART_HVT_DETECTION           = 19,
    VCA_SMART_EVENT                   = 20,
    VCA_VEHICLE_DETECTION             = 21,
    VCA_SMART_ROAD_DETECTION          = 22,
    VCA_SMART_FACE_DETECTION          = 23,
    VCA_SMART_HEATMAP                 = 24,
    VCA_HUMAN_RECOGNITION             = 25,
    VCA_EDUCATION_STUDENT_DETECTION   = 26,
    VCA_ROAD_DETECTION                = 27,
    VCA_PERSON_DENSITY_DETECTION      = 28,
    VCA_SAFETY_HELMET_DETECTION       = 30
}VCA_CHAN_ABILITY_TYPE
```

### VCA\_BEHAVIOR\_BASE

Behavior analysis (basic).

### VCA\_BEHAVIOR\_ADVANCE

Behavior analysis (advanced)

### **VCA\_BEHAVIOR\_FULL**

Behavior analysis (complete)

### **VCA\_PLATE**

ANPR

### **VCA\_ATM**

ATM

### **VCA\_PDC**

People counting

### **VCA\_ITS**

Intelligent traffic incident

### **VCA\_BEHAVIOR\_PRISON**

Behavior analysis (cell)

### **VCA\_FACE\_SNAP**

Face capture

### **VCA\_FACE\_SNAPRECOG**

Face capture and recognition

### **VCA\_FACE\_RETRIEVAL**

Face picture dual-VCA

### **VCA\_FACE\_RECOG**

Face recognition

### **VCA\_BEHAVIOR\_PRISON\_PERIMETER**

Behavior analysis (perimeter)

### **VCA\_TPS**

Traffic guidance

### **VCA\_TFS**

Traffic violation enforcement

### **VCA\_BEHAVIOR\_FACESNAP**

Face capture and behavior analysis

### **VCA\_HEATMAP**

Heat map

### **VCA\_SMART\_VEHICLE\_DETECTION**

SMART event and vehicle detection

### **VCA\_SMART\_HVT\_DETECTION**

SMART event and multi-target-type detection

### **VCA\_SMART\_EVENT**

SMART event

### **VCA\_VEHICLE\_DETECTION**

Vehicle detection

### **VCA\_SMART\_ROAD\_DETECTION**

SMART event and traffic monitoring

### **VCA\_SMART\_FACE\_DETECTION**

SMART event and face detection

### **VCA\_SMART\_HEATMAP**

SMART event and heat map

### **VCA\_HUMAN\_RECOGNITION**

Human body recognition

### **VCA\_EDUCATION\_STUDENT\_DETECTION**

Classroom behavior tracking

### **VCA\_ROAD\_DETECTION**

Traffic monitoring

### **VCA\_PERSON\_DENSITY\_DETECTION**

People density detection

### **VCA\_SAFETY\_HELMET\_DETECTION**

Hard hat detection

## **A.93 VCA\_CHAN\_MODE\_TYPE**

### **Enumeration about ATM Mode**

Enumeration Type	Macro Definition Value	Description
<b>VCA_ATM_PANEL</b>	0	ATM panel
<b>VCA_ATM_SURROUND</b>	1	ATM surrounding
<b>VCA_ATM_FACE</b>	2	ATM human face
<b>VCA_ATM_SAFETYCABIN</b>	3	ATM safety cabin

## **A.94 VCA\_CROSS\_DIRECTION**

## Enumerate Line Crossing Direction

Member	Macro Definition Value	Description
VCA_BOTH_DIRECTION	0	Bidirectional.
VCA_LEFT_GO_RIGHT	1	From left to right.
VCA_RIGHT_GO_LEFT	2	From right to left.

## A.95 VCA\_EVENT\_TYPE

Enumerate the behavior analysis alarm/event types.

### Enumeration Definition

```
enum _VCA_EVENT_TYPE_{
    VCA_TRAVERSE_PLANE      = 0x1,
    VCA_ENTER_AREA          = 0x2,
    VCA_EXIT_AREA           = 0x4,
    VCA_INTRUSION           = 0x8,
    VCA_LOITER              = 0x10,
    VCA_LEFT_TAKE           = 0x20,
    VCA_PARKING             = 0x40,
    VCA_RUN                 = 0x80,
    VCA_HIGH_DENSITY        = 0x100,
    VCA_VIOLENT_MOTION      = 0x200,
    VCA_REACH_HIGHT         = 0x400,
    VCA_GET_UP              = 0x800,
    VCA_LEFT                = 0x1000,
    VCA_TAKE                 = 0x2000,
    VCA_LEAVE_POSITION      = 0x4000,
    VCA_TRAIL               = 0x8000,
    VCA_KEY_PERSON_GET_UP   = 0x10000,
    VCA_FALL_DOWN           = 0x80000,
    VCA_AUDIO_ABNORMAL      = 0x100000,
    VCA_ADV_REACH_HEIGHT    = 0x200000,
    VCA_TOILET_TARRY        = 0x400000,
    VCA_YARD_TARRY          = 0x800000,
    VCA_ADV_TRAVERSE_PLANE  = 0x1000000,
    VCA_HUMAN_ENTER         = 0x10000000,
    VCA_OVER_TIME           = 0x20000000,
    VCA_STICK_UP            = 0x40000000,
    VCA_INSTALL_SCANNER     = 0x80000000
}VCA_EVENT_TYPE
```

## Members

### **VCA\_TRAVERSE\_PLANE**

Line crossing

### **VCA\_ENTER\_AREA**

Region entrance, supports region rule.

### **VCA\_EXIT\_AREA**

Region exiting, supports region rule

### **VCA\_INTRUSION**

Intrusion, supports region rule

### **VCA\_LOITER**

Loitering, supports region rule

### **VCA\_LEFT\_TAKE**

Unattended baggage/object removal, supports region rule

### **VCA\_PARKING**

Parking

### **VCA\_RUN**

Fast moving

### **VCA\_HIGH\_DENSITY**

People gathering, supports region rule

### **VCA\_VIOLENT\_MOTION**

Violent motion detection

### **VCA\_REACH\_HIGHT**

Climbing detection

### **VCA\_GET\_UP**

Getting up detection

### **VCA\_LEFT**

Unattended baggage

### **VCA\_TAKE**

Object removal

### **VCA\_LEAVE\_POSITION**

Absence detection

### **VCA\_TRAIL**

Trailing

### **VCA\_KEY\_PERSON\_GET\_UP**

Key person getting up detection

### **VCA\_STANDUP**

Standing up detection

### **VCA\_FALL\_DOWN**

Falling down detection

### **VCA\_AUDIO\_ABNORMAL**

Sudden change of sound density detection

### **VCA\_ADV\_REACH\_HEIGHT**

Climbing detection

### **VCA\_TOILET\_TARRY**

In-toilet overtime

### **VCA\_YARD\_TARRY**

Playground overstay

### **VCA\_ADV\_TRAVERSE\_PLANE**

Line crossing

### **VCA\_HUMAN\_ENTER**

People closing to ATM, it is supported only in ATM\_PANEL mode

### **VCA\_OVER\_TIME**

Operation overtime, it is supported only in ATM\_PANEL mode

### **VCA\_STICK\_UP**

Sticking up detection, supports region rule

### **VCA\_INSTALL\_SCANNER**

Card reader installing, supports region rule

## **A.96 VCA\_RULE\_EVENT\_TYPE\_EX**

Enumerate the behavior analysis alarm/event types.

### **Enumeration Definition**

```
enum _VCA_RULE_EVENT_TYPE_EX {  
    ENUM_VCA_EVENT_TRAVERSE_PLANE    = 1,  
    ENUM_VCA_EVENT_ENTER_AREA         = 2,  
    ENUM_VCA_EVENT_EXIT_AREA          = 3,  
    ENUM_VCA_EVENT_INTRUSION          = 4,  
    ENUM_VCA_EVENT_LOITER              = 5,  
    ENUM_VCA_EVENT_LEFT_TAKE          = 6,  
    ENUM_VCA_EVENT_PARKING            = 7,  
    ENUM_VCA_EVENT_RUN                 = 8,  
}
```

```
ENUM_VCA_EVENT_HIGH_DENSITY      = 9,
ENUM_VCA_EVENT_VIOLENT_MOTION    = 10,
ENUM_VCA_EVENT_REACH_HIGHT       = 11,
ENUM_VCA_EVENT_GET_UP            = 12,
ENUM_VCA_EVENT_LEFT              = 13,
ENUM_VCA_EVENT_TAKE              = 14,
ENUM_VCA_EVENT_LEAVE_POSITION    = 15,
ENUM_VCA_EVENT_TRAIL             = 16,
ENUM_VCA_EVENT_KEY_PERSON_GET_UP = 17,
ENUM_VCA_EVENT_STANDUP           = 18,
ENUM_VCA_EVENT_FALL_DOWN         = 20,
ENUM_VCA_EVENT_AUDIO_ABNORMAL    = 21,
ENUM_VCA_EVENT_ADV_REACH_HEIGHT  = 22,
ENUM_VCA_EVENT_TOILET_TARRY      = 23,
ENUM_VCA_EVENT_YARD_TARRY        = 24,
ENUM_VCA_EVENT_ADV_TRAVERSE_PLANE = 25,
ENUM_VCA_EVENT_LECTURE           = 26,
ENUM_VCA_EVENT_ANSWER            = 27,
ENUM_VCA_EVENT_HUMAN_ENTER       = 29,
ENUM_VCA_EVENT_OVER_TIME         = 30,
ENUM_VCA_EVENT_STICK_UP          = 31,
ENUM_VCA_EVENT_INSTALL_SCANNER   = 32,
ENUM_VCA_EVENT_PEOPLENUM_CHANGE  = 35,
ENUM_VCA_EVENT_SPACING_CHANGE    = 36,
ENUM_VCA_EVENT_COMBINED_RULE     = 37,
ENUM_VCA_EVENT_SIT_QUIETLY       = 38,
ENUM_VCA_EVENT_HIGH_DENSITY_STATUS = 39,
ENUM_VCA_EVENT_RUNNING           = 40,
ENUM_VCA_EVENT_RETENTION         = 41,
ENUM_VCA_EVENT_BLACKBOARD_WRITE  = 42,
ENUM_VCA_EVENT_SITUATION_ANALYSIS = 43,
ENUM_VCA_EVENT_PLAY_CELLPHONE    = 44,
ENUM_VCA_EVENT_DURATION          = 45,
ENUM_VCA_EVENT_FAKECARD          = 46
}VCA_RULE_EVENT_TYPE_EX
```

### Members

#### **ENUM\_VCA\_EVENT\_TRAVERSE\_PLANE**

Line crossing

#### **ENUM\_VCA\_EVENT\_ENTER\_AREA**

Region entrance, supports setting rule by region.

#### **ENUM\_VCA\_EVENT\_EXIT\_AREA**

Region exiting, supports setting rule by region.

#### **ENUM\_VCA\_EVENT\_INTRUSION**

Intrusion, supports setting rule by region.

#### **ENUM\_VCA\_EVENT\_LOITER**



Loitering, supports setting rule by region.

### **ENUM\_VCA\_EVENT\_LEFT\_TAKE**

Unattended baggage/object removal, supports setting rule by region.

### **ENUM\_VCA\_EVENT\_PARKING**

Parking, supports setting rule by region.

### **ENUM\_VCA\_EVENT\_RUN**

Fast moving, supports setting rule by region.

### **ENUM\_VCA\_EVENT\_HIGH\_DENSITY**

People gathering, supports setting rule by region.

### **ENUM\_VCA\_EVENT\_VIOLENT\_MOTION**

Violent motion detection

### **ENUM\_VCA\_EVENT\_REACH\_HIGHT**

Climbing detection

### **ENUM\_VCA\_EVENT\_GET\_UP**

Getting up detection

### **ENUM\_VCA\_EVENT\_LEFT**

Unattended baggage

### **ENUM\_VCA\_EVENT\_TAKE**

Object removal

### **ENUM\_VCA\_EVENT\_LEAVE\_POSITION**

Absence detection

### **ENUM\_VCA\_EVENT\_TRAIL**

Trailing

### **ENUM\_VCA\_EVENT\_KEY\_PERSON\_GET\_UP**

Key person getting up detection

### **ENUM\_VCA\_EVENT\_STANDUP**

Person standing up

### **ENUM\_VCA\_EVENT\_FALL\_DOWN**

Falling down detection

### **ENUM\_VCA\_EVENT\_AUDIO\_ABNORMAL**

Sudden change of sound density detection

### **ENUM\_VCA\_EVENT\_ADV\_REACH\_HEIGHT**

Climbing detection

### **ENUM\_VCA\_EVENT\_TOILET\_TARRY**

Toilet timeout

### **ENUM\_VCA\_EVENT\_YARD\_TARRY**

In-toilet overtime

### **ENUM\_VCA\_EVENT\_YARD\_TARRY**

Playground overstay

### **ENUM\_VCA\_EVENT\_ADV\_TRAVERSE\_PLANE**

Line crossing

### **ENUM\_VCA\_EVENT\_LECTURE**

Giving lectures (education)

### **ENUM\_VCA\_EVENT\_ANSWER**

Answering questions (education)

### **ENUM\_VCA\_EVENT\_HUMAN\_ENTER**

People closing to ATM, it is supported only in ATM\_PANEL mode

### **ENUM\_VCA\_EVENT\_OVER\_TIME**

Operation overtime, it is supported only in ATM\_PANEL mode

### **ENUM\_VCA\_EVENT\_STICK\_UP**

Sticking up detection, supports setting rule by region.

### **ENUM\_VCA\_EVENT\_INSTALL\_SCANNER**

Card reader installation, supports setting rule by region.

### **ENUM\_VCA\_EVENT\_PEOPLENUM\_CHANGE**

People number changed event

### **ENUM\_VCA\_EVENT\_SPACING\_CHANGE**

Distance changed event

### **ENUM\_VCA\_EVENT\_COMBINED\_RULE**

Combined rule event

### **ENUM\_VCA\_EVENT\_SIT\_QUIETLY**

Sedentary detection

### **ENUM\_VCA\_EVENT\_HIGH\_DENSITY\_STATUS**

People gathering status in area, the device uploads the status according to the configured time interval. This time interval cannot be configured via HCNetSDK, and it can only be configured via the server, the default time interval is 10s.

### **ENUM\_VCA\_EVENT\_RUNNING**

Running detection

### **ENUM\_VCA\_EVENT\_RETENTION**

Overstay detection

**ENUM\_VCA\_EVENT\_BLACKBOARD\_WRITE**

Blackboard writing

**ENUM\_VCA\_EVENT\_SITUATION\_ANALYSIS**

People density analysis

**ENUM\_VCA\_EVENT\_PLAY\_CELLPHONE**

Playing mobile phone detection

**ENUM\_VCA\_EVENT\_DURATION**

Continuous alarm

**ENUM\_VCA\_EVENT\_FAKECARD**

Fake card detection

## A.97 BEHAVIOR\_SCENE\_MODE\_TYPE

### Enumeration about Behavior Analysis Scene Mode

Enumeration Type	Macro Definition Value	Description
BEHAVIOR_SCENE_DEFAULT	0	Default mode
BEHAVIOR_SCENE_WALL	1	Wall
BEHAVIOR_SCENE_INDOOR	2	Indoor

## A.98 NET\_ALARM\_CVR\_SUBINFO\_UNION

### Union about CVR Alarm Information

Member	Data Type	Description
byLen	BYTE[]	Union size, the maximum array length is 492 bytes.
struRecordLost	<b><u>NET_ALARM_RECORD_FILE_LOSS</u></b>	Video loss alarm information, the value of <b>dwAlarmType</b> in <b><u>NET_DVR_ALARMINFO_DEV_V40</u></b> is 8.
struStreamException	<b><u>NET_ALARM_STREAM_EXCEPTION</u></b>	Streaming exception alarm information, the value of <b>dwAlarmType</b> in <b><u>NET_DVR_ALARMINFO_DEV_V40</u></b> is 9.

Member	Data Type	Description
<b>struResourceUsage</b>	<b><u>NET_ALARM_RESOURCE_USAGE</u></b>	Resource usage alarm information, the value of <b>dwAlarmType</b> in <b><u>NET_DVR_ALARMINFO_DEV_V40</u></b> is 10.
<b>struRecordException</b>	<b><u>NET_ALARM_RECORD_EXCEPTION</u></b>	Recording exception alarm information, the value of <b>dwAlarmType</b> in <b><u>NET_DVR_ALARMINFO_DEV_V40</u></b> is 12.

## A.99 NET\_ALARM\_RECORD\_EXCEPTION

### Structure about Recording Exception Alarm Information

Member	Data Type	Description
<b>byReason</b>	BYTE	Exception reason: 0-video volume full, 1-video volume exception, 2-no available video volume.
<b>byRes1</b>	BYTE[]	Reserved, set to 0. The maximum array length is 3 bytes.
<b>sVolumeName</b>	BYTE[]	Video volume name, the maximum array length is "MAX_VOLUMENAME_LEN" (32 bytes).
<b>dwVolumeID</b>	DWORD	Video volume ID, or HDD No.
<b>byRes</b>	BYTE[]	Reserved, set to 0. The maximum array length is 452 bytes.

## A.100 NET\_ALARM\_RECORDFILE\_LOSS

### Structure about Video Loss Alarm Information

Member	Data Type	Description
<b>struInspectStart</b>	<b><u>NET_DVR_TIME_EX</u></b>	Start time of video loss check.
<b>struInspectEnd</b>	<b><u>NET_DVR_TIME_EX</u></b>	End time of video loss check.
<b>struIP</b>	<b><u>NET_DVR_IPADDR_UNION</u></b>	IP address of video loss channel.
<b>dwChanNo</b>	DWORD	Channel No.

Member	Data Type	Description
<b>dwIDIndex</b>	DWORD	Encoder ID.
<b>sName</b>	BYTE[]	Encoder name, the maximum array length is "STREAM_ID_LEN" (32 bytes).
<b>struLossStartTime</b>	<u>NET_DVR_TIME_EX</u>	Start time of video loss.
<b>struLossEndTime</b>	<u>NET_DVR_TIME_EX</u>	End time of video loss.
<b>dwLostNum</b>	DWORD	Number of lost video files, 0xffffffff-all video files are lost.
<b>byRes</b>	BYTE[]	Reserved, set to 0. The maximum array length is 240 bytes.

## A.101 NET\_ALARM\_RESOURCE\_USAGE

### Structure about Resource Usage Alarm Information

Member	Data Type	Description
<b>byLevel</b>	BYTE	Usage alarm level: 0-normal, 1-alarm level 1, 2-alarm level 2, 3-alarm level 3.
<b>byRes</b>	BYTE[]	Reserved, set to 0. The maximum array length is 491 bytes.

## A.102 NET\_ALARM\_STREAM\_EXCEPTION

### Structure about Video Exception Alarm Information

Member	Data Type	Description
<b>struIP</b>	<u>NET_DVR_IPADDR_UNION</u>	IP address of video exception channel.
<b>dwChanNo</b>	DWORD	Channel No.
<b>dwIDIndex</b>	DWORD	Encoder ID.
<b>sName</b>	BYTE[]	Encoder name, the maximum array length is "STREAM_ID_LEN" (32 bytes).

Member	Data Type	Description
<b>byExceptionCase</b>	BYTE	Exception reason: 0-data writing exception, 1-network exception.
<b>byRes</b>	BYTE[]	Reserved, set to 0. The maximum array length is 307 bytes.

## A.103 NET\_DVR\_ALARMER

### Alarm Device Information Structure

Member	Data Type	Description
byUserIDValid	BYTE	Whether the user ID is valid: 0-no, 1-yes
bySerialValid	BYTE	Whether the serial No. is valid: 0-no, 1-yes
byVersionValid	BYTE	Whether the version No. is valid: 0-no, 1-yes
byDeviceNameValid	BYTE	Whether the device name is valid: 0-no, 1-yes
byMacAddrValid	BYTE	Whether the MAC address is valid: 0-no, 1-yes
byLinkPortValid	BYTE	Whether the login port No. is valid: 0-no, 1-yes
byDeviceIPValid	BYTE	Whether the device IP address is valid: 0-no, 1-yes
bySocketIPValid	BYTE	Whether the Socket IP address is valid: 0-no, 1-yes
lUserID	LONG	Value returned by <b><i>NET DVR Login V40</i></b> , it is valid when arming.
sSerialNumber	Array of BYTE	Serial No.
dwDeviceVersion	DWORD	Version information
sDeviceName	Array of char	Device name
byMacAddr	Array of BYTE	MAC address
wLinkPort	WORD	Device communication port No.
sDeviceIP	Array of char	Device IP address
sSocketIP	Array of char	Socket IP address when actively uploading alarm.

Member	Data Type	Description
byIpProtocol	BYTE	Network protocol: 0-IPv4, 1-IPv6
byRes2	Array of BYTE	Reserved, set to 0.

## A.104 NET\_DVR\_ALARMINFO\_DEV

### Device Alarm Information Structure

Memeber	Data Type	Description
<b>dwAlarmType</b>	DWORD	Alarm types: 0-alarm input alarm of encoder, 1-second private volume damaged, 2-NVR disconnected, 3-encoder exception, 4-system clock exception, 5-the remaining capacity of the recording volume is too low, 6-motion detection alarm of encoder or encoding channel, 7-video tampering alarm of encoder or encoding channel.
<b>struTime</b>		Alarm time
<b>byRes</b>	Array of BYTE	Reserved, set to 0.
<b>dwNumber</b>	DWORD	Number of alarm triggered channels.
<b>pNO</b>	WORD*	Channel No. or disk No., which ranges from 0 to 65535.

### Remarks

For **pNO**: if **dwAlarmType** is 0, 3, 6, or 7, it may be channel No.; if **dwAlarmType** is 5, it may be disk No.

## A.105 NET\_DVR\_ALARMINFO\_DEV\_V40

### Structure about CVR Alarm Information

Member	Data Type	Description
<b>dwAlarmType</b>	DWORD	Alarm categories: 0-alarm input alarm of encoder, 1-second private volume damaged, 2-

Member	Data Type	Description
		NVR disconnected, 3-encoder exception, 4-system clock exception, 5-the remaining capacity of the recording volume is too low, 6-motion detection alarm of encoder or encoding channel, 7-video tampering alarm of encoder or encoding channel, 8-video loss alarm, 9-real-time health monitoring alarm, 10-usage alarm, 11-CVR exception recovered, 12-recording exception.
<b>struTime</b>	<b><u>NET_DVR_TIME</u></b>	Alarm time
<b>uSubAlarmInfo</b>	<b><u>NET_ALARM_CVR_SUBINFO_UNION</u></b>	CVR alarm information structure, and it is valid when the alarm type is 8, 9, 10, and 12.
<b>byRes</b>	Array of BYTE	Reserved, set to 0. The maximum size is 256 bytes.
<b>dwNumber</b>	DWORD	Number of alarm triggered channels.
<b>pNO</b>	WORD*	Channel No. or disk No., which ranges from 0 to 65535.

**Remarks**

For **pNO**: if **dwAlarmType** is 0, 3, 6, or 7, it may be channel No.; if **dwAlarmType** is 5, it may be disk No.

**A.106 NET\_DVR\_ALARMINFO\_V30****Structure About Uploaded Alarm Information**

Member	Data Type	Description
<b>dwAlarmType</b>	DWORD	Alarm types: 0-alarm input alarm of encoder, 1-second private volume damaged, 2-NVR disconnected, 3-encoder exception, 4-system clock exception, 5-the remaining capacity of the recording volume is too low, 6-motion detection alarm of encoder or encoding channel, 7-video tampering alarm of encoder or encoding channel, 8-video loss alarm, 9-real-time health



Member	Data Type	Description
		monitoring alarm, 10-usage alarm, 11-CVR exception recovered, 12-recording exception.
dwAlarmInputNumber	DWORD	Alarm input No., it is valid when alarm type is 0 or 23
byAlarmOutputNumber	Array of BYTE	The triggered alarm output No. E.g. dwAlarmOutputNumber[0]==1 indicates that alarm output No.1 is triggered; dwAlarmOutputNumber[1]==1 indicates that alarm output No.2 is triggered.
byAlarmRelateChannel	Array of BYTE	The triggered recording channel No.: 0-not triggered, 1-triggered. E.g. dwAlarmRelateChannel[0]==1 indicates that the channel No.1 is triggered to record.
byChannel	Array of BYTE	Alarm channel, it is valid when alarm type is 2, 3, 6, 9, 10 or 11. E.g. dwChannel[0]==1 indicates that the channel No. is in alarm.
byDiskNumber	Array of BYTE	Alarm HDD, it is valid when alarm type is 1, 4, or 5. E.g. dwDiskNumber [0]==1 indicates that the HDD No.1 is abnormal.

## Remarks

The time interval to upload the alarm of face picture library changed is 1 hour; for other alarm type, the alarm information is uploaded in real-time, and the time interval is 1s. Currently, editing the time interval is not supported.

## A.107 NET\_DVR\_ALARMINFO\_V40

### Structure About Uploaded Alarm Information

Member	Data Type	Description
struAlarmFixedHeader	<b><u>NET_DVR_ALRAM_FIXED_HEADER</u></b>	Constant content in alarm information, see details in the structure .
pAlarmData	DWORD*	Variable content in alarm information

## Remarks

- The time interval to upload the alarm of face picture library changed is 1 hour; for other alarm type, the alarm information is uploaded in real-time, and the time interval is 1s. Currently, editing the time interval is not supported.
- The content of **pAlarmData** varies with the value of **dwAlarmType** in the structure **NET\_DVR\_ALARM\_FIXED\_HEADER**, see details in the table below:

**Table A-2 Relations Between pAlarmData and dwAlarmType**

dwAlarmType	Description	pAlarmData
0, 23	Alarm input alarm, pulse alarm	dwTrigerAlarmOutNum*(DWORD) Alarm output No., +dwTrigerRecordChanNum*(DWORD) Channel No.
2, 3, 6, 9, 10, 11, 13, 15, 16, 19	Video loss, motion detection, video tampering alarm, video exception, recording exception, scene change, resolution mismatched, VCA detection, PoE power supply exception, audio loss	dwAlarmChanNum*(DWORD) channel No.
1, 4, 5	HDD full, HDD uninitialized, writing to HDD failed	dwAlarmHardDiskNum*(DWORD) HDD No.
7, 8, 12, 17, 18, 24, 25, 26	Standard mismatches, invalid login, array exception, education sharing system alarm, two-way audio request alarm, face library HDD exception, face library changed, picture changed in face picture library	None

## A.108 NET\_DVR\_ALARM\_FIXED\_HEADER

### Structure About Constant Alarm Information

Member	Data Type	Description
dwAlarmType	DWORD	Alarm information type: 0-alarm input alarm, 1-HDD full, 2-video loss, 3-motion detection, 4-

Member	Data Type	Description
		HDD unformatted, 5-writing to HDD failed, 6-video tampering alarm, 7-standard mismatched, 8-invalid login, 9-video exception, 10-recording exception, 11-scene change, 12-RAID exception, 13-resolution mismatched, 15-VCA detection, 16- PoE power supply exception, 17-education sharing system alarm, 18-two-way audio request alarm, 23-pulse alarm, 24-face picture library HDD exception, 25-face picture library changed, 26-picture of face picture library changed, 27-POC exception, 28-camera FOV exception, 30-no SD card, 31-supply voltage exception, 32-PTZ locked
struAlarmTime	<b><u>NET_DVR_TIME_EX</u></b>	Alarm time
uStruAlarm	Union ( <b><u>Table 5-3</u></b> )	Alarm information union
pRes	DWORD*	Reserved.
byTimeDiffFlag	BYTE	Whether the time difference parameter is valid: 0-invalid, 1-valid.
cTimeDifferenceH	char	Time difference between time and UTC time, unit: hour, the value is between -12 and +14 ("+" indicates the east time zone), it is valid when <b>byISO8601</b> is "1".
cTimeDifferenceM	char	Time difference between time and UTC time, unit: minute, the value is -30, +30, or +45 ("+" indicates the east time zone), it is valid when <b>byISO8601</b> is "1".
byRes	Array of BYTE	Reserved, set to 0. The maximum size is 5 bytes.

Table A-3 Union about Alarm Information Structures (uStruAlarm)

Member	Data Type	Description
byUnionLen	Array of BYTE	Union size, which is 116 bytes.
struIOAlarm	Struct ( <b><u>Table 5-4</u></b> )	Structure about alarm input parameters
struAlarmChannel	Struct ( <b><u>Table 5-5</u></b> )	Structure about alarm channel parameters
struAlarmHardDisk	Struct ( <b><u>Table 5-6</u></b> )	Structure about HDD alarm parameters

Member	Data Type	Description
struRecordingHost	Struct ( <a href="#">Table 5-7</a> )	Structure about alarm parameters of education sharing system
struVoltageInstable	Struct ( <a href="#">Table 5-8</a> )	Structure about alarm parameters of supply voltage exception
struPTLocking	Struct ( <a href="#">Table 5-9</a> )	Structure about parameters of PTZ locked alarm

**Table A-4 Structure about Alarm Input Parameters (struIOAlarm)**

Member	Data Type	Description
dwAlarmInputNo	DWORD	Alarm input No.
dwTrigerAlarmOutNum	DWORD	The number of triggered alarm outputs. It is used for calculating the number of all triggered alarm outputs by <b>pAlarmData</b> in <b>NET_DVR_ALARMINFO_V40</b> , each alarm output is represented by 4 bytes.
dwTrigerRecordChanNum	DWORD	The number of triggered recording channels. It is used for calculating the number of all triggered recording channels by <b>pAlarmData</b> of <b>NET_DVR_ALARMINFO_V40</b> , each channel is represented by 4 bytes.

**Table A-5 Structure about Alarm Channel Parameters (struAlarmChannel)**

Member	Data Type	Description
dwAlarmChanNum	DWORD	The number of alarm channels. It is used for calculating the number of all alarm channels by <b>pAlarmData</b> of <b>NET_DVR_ALARMINFO_V40</b> , each alarm channel is represented by 4 bytes.
dwPicLen	DWORD	Size of JPEG picture.
byPicURL	BYTE	Picture data format: 0-binary data, 1-URL.
byTarget	BYTE	Detection target type: 0-not supported, 1-person, 2-vehicle.
byRes1	Array of BYTE	Reserved, the maximum size is 2 bytes.

Member	Data Type	Description
pDataBuff	char*	Alarm picture data or URL. The pointer size is 8 bytes.
byRes3	Array of BYTE	Reserved, the maximum size is 4 bytes. This member is only available for 64-bit Window operating system and 64-bit Linux operating system.

**Table A-6 Structure about HDD Alarm Parameters (struAlarmHardDisk)**

Member	Data Type	Description
dwAlarmHardDiskNum	DWORD	The number of alarm HDD. It is used for calculating the number of all alarm HDDs by <b>pAlarmData</b> of <u><b>NET DVR ALARMINFO V40</b></u> , each alarm HDD is represented by 4 bytes.

**Table A-7 Structure about Alarm Parameters of Education Sharing System (struRecordingHost)**

Member	Data Type	Description
bySubAlarmType	BYTE	Alarm minor type: 1-one-touch post-record
byRes1	Array of BYTE	Reserved, set to 0. The maximum size is 3 bytes.
struRecordEndTime	<u><b>NET DVR TIME EX</b></u>	Recording end time.

**Table A-8 Structure about Alarm Parameters of Supply Voltage Exception (struVoltageInstable)**

Member	Data Type	Description
fVoltageValue	float	Supply voltage, unit: V, corrects to one decimal place.
byVoltageAlarmType	BYTE	Supply voltage exception type: 0-high supply voltage, 1-low supply voltage
byRes1	Array of BYTE	Reserved, set to 0. The maximum size is 3 bytes.

**Table A-9 Structure about Parameters of PTZ Locked Alarm (struPTLocking)**

Member	Data Type	Description
fTemperature	float	Sensor temperature, which is accurate to one decimal place.
dwCustomInfoLength	DWORD	Custom information length.

Member	Data Type	Description
pCustomInfo	BYTE*	Custom information.
byType	BYTE	PTZ locked direction: 1-panning is locked, 2-tilting is locked.
byDeicingEnabled	BYTE	Whether to enable heat for PTZ: 0-no, 1-yes.

## Remarks

**dwAlarmType**==0, 23 corresponds to the structure struIOAlarm; **dwAlarmType**==2/3/6/9/10/11/13/15/16/28 corresponds to the structure struAlarmChannel; **dwAlarmType**==1/4/5 corresponds to the structure struAlarmHardDisk; **dwAlarmType**== 17 corresponds to the structure struRecordingHost; **dwAlarmType**== 31 corresponds to the structure struVoltageInstable; for other value, the union is not available.

## A.109 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.
byDataType	BYTE	Alarm data type: 0-invalid, 1-XML, 2-JSON.
byPicturesNumber	BYTE	The number of pictures (number of <b>pPicPackData</b> returned). When this member is 1, only one structure of <b><u>NET_DVR_ALARM_ISAPI_PICDATA</u></b> will be returned by <b>pPicPackData</b> . When this member is larger than 1, multiple structures of <b><u>NET_DVR_ALARM_ISAPI_PICDATA</u></b> will be returned by <b>pPicPackData</b> .

Member	Data Type	Description
<b>byRes</b>	Array of BYTE	Reserved, set to 0. The maximum size is 2 bytes.
<b>pPicPackData</b>	void*	Alarm picture structure, see <a href="#"><b><u>NET_DVR_ALARM_ISAPI_PICDATA</u></b></a> for details.
<b>byRes</b>	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.110 NET\_DVR\_ALARM\_ISAPI\_PICDATA

### Structure about Alarm Picture Data Transmitted Based on Text Protocol

Member	Data Type	Description
<b>dwPicLen</b>	DWORD	Alarm picture data length.
<b>byRes</b>	Array of BYTE	Reserved, set to 0. The maximum size is 4 bytes.
<b>szFilename</b>	Array of char	Picture file saving path, including file name. The maximum size is 256 bytes.
<b>pPicData</b>	BYTE*	Pointer that pointing to the uploaded image data.

## A.111 NET\_DVR\_ETHERNET\_V30

## Ethernet Configuration Structure

Member	Data Type	Description
struDVRIP	<u><b>NET_DVR_IPADDR_UNION</b></u>	Device IP address
struDVRIPMask	<u><b>NET_DVR_IPADDR_UNION</b></u>	Mask of device IP address
dwNetInterface	DWORD	Network interface type: 1-10MBase-T; 2-10MBase-T (full duplex); 3-100MBase-TX; 4-100M (full duplex); 5-10M/100M/1000M (self-adaptive); 6-1000M (full duplex)
wDVRPort	WORD	Device port No.
wMTU	WORD	MTU settings, the default is 1500.
byMACAddr	Array of BYTE	Device physical address.
byEthernetPortNo	BYTE	Network interface No.: 0-invalid, 1-interface 0, 2-interface 1, and so on. This parameter is read-only.
byRes	Array of BYTE	Reserved.

### A.112 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.113 NET\_DVR\_NETCFG\_V50



## Network Configuration Structure

Member	Data Type	Description
dwSize	DWORD	Structure size.
struEtherNet	Array of <u><b>NET_DVR_ETHERNET_V30</b></u>	Ethernet interface
struRes1	Array of	Reserved, set to 0.
struAlarmHostIpAddr	<u><b>NET_DVR_IPADDR_UNION</b></u>	Listening service IP address
byRes2	Array of BYTE	Reserved, set as 0
wAlarmHostIpPort	WORD	Listening service port No.
byUseDhcp	BYTE	Whether to enable DHCP: 0xff- invalid; 0-disable, 1-enable
byIPv6Mode	BYTE	Allocation mode of IPv6 address: 0-by router advertisement, 1-by manual setting, 2-by enabling DHCP allocation.
struDnsServer1IpAddr	<u><b>NET_DVR_IPADDR_UNION</b></u>	IP address of domain name server 1
struDnsServer2IpAddr	<u><b>NET_DVR_IPADDR_UNION</b></u>	IP address of domain name server 2
byIpResolver	Array of BYTE	IP resolver domain name or IP address (if the port No. of device is 8000, the domain name is not supported).
wIpResolverPort	WORD	IP resolver port No.
wHttpPortNo	WORD	HTTP port No.
struMulticastIpAddr	<u><b>NET_DVR_IPADDR_UNION</b></u>	Multicast group address
struGatewayIpAddr	<u><b>NET_DVR_IPADDR_UNION</b></u>	Gateway address
struPPPoE	<u><b>NET_DVR_PPPOECFG</b></u>	PPPoE parameters
byEnablePrivateMulticastDiscovery	BYTE	Private multicast search (SADP): 0-default, 1-enable, 2-disable

Member	Data Type	Description
byEnableOnvifMulticastDiscovery	BYTE	Onvif multicast search (SADP): 0-default, 1-enable, 2-disable
wAlarmHost2IpPort	WORD	Port No. of listening host 2.
struAlarmHost2IpAddr	<b><u>NET_DVR_IPADDR_UNION</u></b>	IP address of listening host 2
byEnableDNS	BYTE	DNS address setting mode: 0-automatically get, 1-manually set.
byRes	Array of BYTE	Reserved, set to 0

### Remarks

- For device only supports the private protocol with version 3.0 or lower, when the parameter **byUseDhcp**="0xff", you should set the device IP address to null, and then the device will automatically get the DHCP information.
- When the parameter **byIPv6Mode** is set to 0 or 2, setting IPv6 address in the parameter **struEtherNet** is not required, it will be obtained automatically by the device; when **byIPv6Mode** is set to 1, you should set IPv6 address. As there are multiple IPv6 addresses, the IPv6 address of current logged-in device may be different with that in **struEtherNet**.

## A.114 NET\_DVR\_PPPOECFG

### PPPoE Configuration Structure

Member	Data Type	Description
dwPPPOE	DWORD	Whether to enable PPPoE: 0-no, 1-yes.
sPPPoEUser	Array of BYTE	PPPoE user name.
sPPPoEPassword	Array of char	PPPoE password.
struPPPoEIP	<b><u>NET_DVR_IPADDR_UNION</u></b>	PPPoE IP address

## A.115 NET\_DVR\_SETUPALARM\_PARAM\_V50

## Arming Parameter Structure

Member	Data Type	Description
<b>dwSize</b>	DWORD	Structure size.
<b>byLevel</b>	BYTE	Arming priority: 0-high, 1-medium, 2-low.
<b>byAlarmInfoType</b>	BYTE	Intelligent traffic alarm information type: 0-old (NET_DVR_PLATE_RESULT), 1-new (NET_ITS_PLATE_RESULT).
<b>byRetAlarmTypeV40</b>	BYTE	0-the motion detection, video loss, video tampering, and alarm input alarm information is uploaded in normal mode (alarm type: COMM_ALARM_V30, alarm information structure: <b>NET_DVR_ALARMINFO_V30</b> ); 1-alarm information is uploaded in variable size (alarm type: COMM_ALARM_V40, alarm information structure: <b>NET_DVR_ALARMINFO_V40</b> ).
<b>byRetDevInfoVersion</b>	BYTE	Alarm types of CVR: 0-COMM_ALARM_DEVICE (alarm information structure: <b>NET_DVR_ALARMINFO_DEV</b> ), 1-COMM_ALARM_DEVICE_V40 (alarm information structure: <b>NET_DVR_ALARMINFO_DEV_V40</b> ).
<b>byRetVQDAlarmType</b>	BYTE	VQD alarm types: 0-COMM_ALARM_VQD (alarm information structure: NET_DVR_VQD_DIAGNOSE_INFO), 1-COMM_ALARM_VQD_EX (alarm information structure: NET_DVR_VQD_ALARM, including camera information and captured pictures)
<b>byFaceAlarmDetection</b>	BYTE	Face detection alarm types: 1-face detection alarm (alarm type: COMM_ALARM_FACE_DETECTION, alarm information structure: NET_DVR_FACE_DETECTION), 0-face capture alarm (alarm type: COMM_UPLOAD_FACESNAP_RESULT, alarm information structure: NET_VCA_FACESNAP_RESULT).
<b>bySupport</b>	BYTE	Capabilities, which is represented by bit: <ul style="list-style-type: none"> <li>• bit0-whether to upload picture: 0-yes, 1-no</li> <li>• bit1-whether to enable ANR: 0-no, 1-yes</li> </ul>

Member	Data Type	Description
		<ul style="list-style-type: none"> <li>bit4-whether to upload behavior analysis events of all detection targets: 0-no, 1-yes. It is used to enable the NVR to get events of all targets detected by network cameras.</li> <li>bit5-whether to enable all-day event or alarm uploading: 0-no, 1-yes. It is used to enable the NVR to receive all alarms from network cameras.</li> </ul>
<b>byBrokenNetHttp</b>	BYTE	ANR type, which is represented by bit and should be supported by device: <ul style="list-style-type: none"> <li>bit0-whether to enable ANR for ANPR: 0-no, 1-yes.</li> <li>bit1-whether to enable ANR for people counting: 0-no, 1-yes.</li> <li>bit2-whether to enable ANR for heat map: 0-no, 1-yes.</li> <li>bit3-whether to enable ANR for face capture: 0-no, 1-yes.</li> <li>bit4-whether to enable ANR for face picture comparison: 0-no, 1-yes.</li> <li>bit5-whether to enable ANR for JSON message transmission: 0-no, 1-yes.</li> <li>bit6: whether to enable ANR for uploading heat map data by dwell time duration and by people quantity: 0-no, 1-yes.</li> <li>bit7: whether to enable ANR for uploading intersection analysis result: 0-no, 1-yes.</li> </ul>
<b>wTaskNo</b>	BYTE	Task No.
<b>byDeployType</b>	BYTE	Arming type: 0-arm via client software, 1-real-time arming.
<b>bySubScriptio</b>	BYTE	Subscription parameters, which is represent by bit. Bit7-whether to upload picture after subscribing motion detection alarm by person or vehicle: 0-no, 1-yes.
<b>byRes1</b>	Array [BYTE]	Reserved, set to 0. The maximum size is 2 bytes.

Member	Data Type	Description
<b>byAlarmTypeURL</b>	BYTE	Alarm picture data type, which is represented by bit, if the device supports uploading alarm pictures in binary format and URL format, you can specify the data type to be uploading via this parameter, if the device only supports URL format, this parameter is invalid. If the URL format is selected, you should set the device and enable the cloud storage, otherwise, the picture will still be transmitted in binary format. <ul style="list-style-type: none"><li>• bit0-type of captured face pictures: 0-binary data, 1-URL</li><li>• bit1-type of picture uploaded in message: 0-binary, 1-URL</li><li>• bit2-type of picture uploaded for face picture comparison: 0-binary, 1-URL</li></ul>
<b>byCustomCtrl</b>	BYTE	Custom control type, which is represented by bit, bit0-whether to upload the face thumbnail of the front passenger: 0-no, 1-yes
<b>byRes4</b>	Array [BYTE]	Reserved, set to 0. The maximum size is 128 bytes.

### Remarks

- The parameters **byLevel** and **byAlarmInfoType** are available for traffic cameras. Up to 1 cameras can be armed in the priority of level 0, up to 3 cameras can be armed in the priority of level 1, and up to 5 cameras can be armed in the priority of level 3, the alarm/event information from the camera in highest priority will be uploaded first.
- For arming via client software, only supports arming one channel, and supports uploading the alarm/event when device is offline; for real-time arming, up to four channels can be armed at same time, but uploading alarm/event when device is offline is not supported.
- The parameter **wTaskNo** is used to distinguish different arming connections. If the value of this parameter in different arming connections is same, error will be returned.

## A.116 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.117 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.

## Appendix B. 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

## B.1 /ISAPI/Event/channels/<ID>/capabilities

Get event capabilities supported by the channel.

### Request URI Definition

**Table B-1 GET /ISAPI/Event/channels/<ID>/capabilities**

<b>Method</b>	GET
<b>Description</b>	Get event capabilities supported by the channel.
<b>Query</b>	None.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u>XML_ChannelEventCap</u> Failed: <u>XML_ResponseStatus</u>

### Remarks

The <ID> in the URI refers to the channel ID.

## B.2 /ISAPI/Event/channels/capabilities

Get the event capability of all channels.

## Request URL Definition

**Table B-2 GET /ISAPI/Event/channels/capabilities**

<b>Method</b>	GET
<b>Description</b>	Get the event capability of all channels.
<b>Query</b>	None.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><i>XML_ChannelEventCapList</i></u> Failed: <u><i>XML_ResponseStatus</i></u>

## B.3 /ISAPI/Event/schedules/radarFieldDetections

Get or set arming schedules of all radar channels for intrusion detection.

### Request URI Definition

**Table B-3 GET /ISAPI/Event/schedules/radarFieldDetections**

<b>Method</b>	GET
<b>Description</b>	Get arming schedules of all radar channels for intrusion detection.
<b>Query</b>	None
<b>Request</b>	None
<b>Response</b>	Succeeded: <u><i>XML_RadarFieldDetectionScheduleList</i></u> Failed: <u><i>XML_ResponseStatus</i></u>

**Table B-4 PUT /ISAPI/Event/schedules/radarFieldDetections**

<b>Method</b>	PUT
<b>Description</b>	Set arming schedules of all radar channels for intrusion detection.
<b>Query</b>	None
<b>Request</b>	<u><i>XML_RadarFieldDetectionScheduleList</i></u>
<b>Response</b>	<u><i>XML_ResponseStatus</i></u>

## B.4 /ISAPI/Event/schedules/radarFieldDetections/<ID>

Get or set the arming schedule of a radar channel for intrusion detection.

### Request URI Definition

**Table B-5 GET /ISAPI/Event/schedules/radarFieldDetections/<ID>**

<b>Method</b>	GET
<b>Description</b>	Get the arming schedule of a radar channel for intrusion detection.
<b>Query</b>	None
<b>Request</b>	None
<b>Response</b>	Succeeded: <u><i>XML_Schedule</i></u> Failed: <u><i>XML_ResponseStatus</i></u>

**Table B-6 PUT /ISAPI/Event/schedules/radarFieldDetections/<ID>**

<b>Method</b>	PUT
<b>Description</b>	Set the arming schedule of a radar channel for intrusion detection.
<b>Query</b>	None
<b>Request</b>	<u><i>XML_Schedule</i></u>
<b>Response</b>	<u><i>XML_ResponseStatus</i></u>

### Remarks

The <ID> in the request URI refers to the ID of radar channel for intrusion detection, and it consists of the detection type name (here, it is "radarFieldChannel") and channel ID, e.g., "radarFieldChannel-101".

## B.5 /ISAPI/Event/schedules/radarLineDetections

Get or set arming schedules of all radar channels for line crossing detection.

### Request URI Definition

**Table B-7 GET /ISAPI/Event/schedules/radarLineDetections**

<b>Method</b>	GET
<b>Description</b>	Get arming schedules of all radar channels for line crossing detection.

Query	None
Request	None
Response	Succeeded: <u><i>XML_RadarLineDetectionScheduleList</i></u> Failed: <u><i>XML_ResponseStatus</i></u>

**Table B-8 PUT /ISAPI/Event/schedules/radarLineDetections**

Method	GET
Description	Set arming schedules of all radar channels for line crossing detection.
Query	None
Request	<u><i>XML_RadarLineDetectionScheduleList</i></u>
Response	<u><i>XML_ResponseStatus</i></u>

## B.6 /ISAPI/Event/schedules/radarLineDetections/<ID>

Get or set the arming schedule of a radar channel for line crossing detection.

### Request URI Definition

**Table B-9 GET /ISAPI/Event/schedules/radarLineDetections/<ID>**

Method	GET
Description	Get the arming schedule of a radar channel for line crossing detection.
Query	None
Request	None
Response	Succeeded: <u><i>XML_Schedule</i></u> Failed: <u><i>XML_ResponseStatus</i></u>

**Table B-10 PUT /ISAPI/Event/schedules/radarLineDetections/<ID>**

Method	PUT
Description	Set the arming schedule of a radar channel for line crossing detection.
Query	None

Request	<u><i>XML_Schedule</i></u>
Response	<u><i>XML_ResponseStatus</i></u>

### Remarks

The <ID> in the request URI refers to the ID of radar channel for line crossing detection, and it consists of the detection type name (here, it is "radarLineChannel") and channel ID, e.g., "radarLineChannel-101".

## B.7 /ISAPI/Event/triggers/<eventType>--<channelID>

Get, set, or delete the alarm linkage action by channel.

### Request URI Definition

Table B-11 GET /ISAPI/Event/triggers/<eventType>--<channelID>

Method	GET
Description	Get the alarm linkage action by channel.
Query	None
Request	None
Response	Succeeded: <u><i>XML_EventTrigger</i></u> Failed: <u><i>XML_ResponseStatus</i></u>

Table B-12 PUT /ISAPI/Event/triggers/<eventType>--<channelID>

Method	PUT
Description	Set the alarm linkage action by channel.
Query	None
Request	<u><i>XML_EventTrigger</i></u>
Response	<u><i>XML_ResponseStatus</i></u>

Table B-13 DELETE /ISAPI/Event/triggers/<eventType>--<channelID>

Method	DELETE
Description	Delete the alarm linkage action by channel.
Query	None

Request	None
Response	<u><i>XML_ResponseStatus</i></u>

### Remarks

The **<eventType>** in the request URI refers to the predefined event or alarm type name, and the **<channelID>** is the No. of the event detection channel. For example, if the No. of the face capture channel is 101, the "**<eventType>-<channelID>**" is "faceSnap-101".

## B.8 /ISAPI/Event/triggersCap

Get alarm linkage capability.

### Request URI Definition

**Table B-14 GET /ISAPI/Event/triggersCap**

Method	GET
Description	Get alarm linkage capability.
Query	None
Request	None
Response	Succeeded: <u><i>XML_EventTriggersCap</i></u> Failed: <u><i>XML_ResponseStatus</i></u>

## B.9 /ISAPI/Intelligent/analysisEngines/<ID>

Get or set parameters of an analysis engine.

### Request URI Definition

**Table B-15 GET /ISAPI/Intelligent/analysisEngines/<ID>**

Method	GET
Description	Get the parameters of an analysis engine.
Query	None.
Request	None.
Response	Succeeded: <u><i>XML_AnalysisEngine</i></u>

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

**Table B-16 PUT /ISAPI/Intelligent/analysisEngines/<ID>**

Method	PUT
Description	Set the parameters of an analysis engine.
Query	None.
Request	<u><i>XML_AnalysisEngine</i></u>
Response	<u><i>XML_ResponseStatus</i></u>

### Remarks

The <ID> in the request URI refers to the analysis engine No.

## B.10 /ISAPI/Intelligent/analysisEngines/capabilities

Get configuration capability of analysis engine.

### Request URI Definition

**Table B-17 GET /ISAPI/Intelligent/analysisEngines/capabilities**

Method	GET
Description	Get configuration capability of analysis engine.
Query	None.
Request	None.
Response	Succeeded: <u><i>XML_AnalysisEnginesCap</i></u> Failed: <u><i>XML_ResponseStatus</i></u>

## B.11 /ISAPI/Intelligent/capabilities

Get the intelligent capability set.

## Request URI Definition

**Table B-18 GET /ISAPI/Intelligent/capabilities**

<b>Method</b>	GET
<b>Description</b>	Get the intelligent capability.
<b>Query</b>	None.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><i>XML_IntelliCap</i></u> Failed: <u><i>XML_ResponseStatus</i></u>

## B.12 /ISAPI/Intelligent/channels/<ID>/capabilities

Get the intelligent capability of device.

## Request URI Definition

**Table B-19 GET /ISAPI/Intelligent/channels/<ID>/capabilities**

<b>Method</b>	GET
<b>Description</b>	Get the intelligent capability of device.
<b>Query</b>	None
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><i>XML_IntelliCap</i></u> Failed: <u><i>XML_ResponseStatus</i></u>

## Remarks

The <ID> in the URI refers to the intelligent channel ID.

## B.13 /ISAPI/Intelligent/channels/<ID>/behaviorRule/<SID>/notifications

Operations about alarm linkage configuration of behavior analysis of a specific detection scene.



## Request URI Definition

**Table B-20 GET /ISAPI/Intelligent/channels/<ID>/behaviorRule/<SID>/notifications**

<b>Method</b>	GET
<b>Description</b>	Get alarm linkage parameters of behavior analysis of a specific detection scene.
<b>Query</b>	None.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><i>XML_RuleNotification</i></u> Failed: <u><i>XML_ResponseStatus</i></u>

**Table B-21 PUT /ISAPI/Intelligent/channels/<ID>/behaviorRule/<SID>/notifications**

<b>Method</b>	PUT
<b>Description</b>	Set alarm linkage of behavior analysis of a specific detection scene.
<b>Query</b>	None.
<b>Request</b>	<u><i>XML_RuleNotification</i></u>
<b>Response</b>	<u><i>XML_ResponseStatus</i></u>

## Remarks

The <ID> in the URI refers to the video channel No., and the <SID> is the detection scene No.

## B.14 /ISAPI/Intelligent/channels/<ID>/behaviorRule/<SID>/rule/<ID>

Operations about configuring a behavior analysis rule by detection scene ID.

## Request URL Definition

**Table B-22 GET /ISAPI/Intelligent/channels/<ID>/behaviorRule/<SID>/rule/<ID>**

<b>Method</b>	GET
<b>Description</b>	Get the parameters of a behavior analysis rule by detection scene ID.
<b>Query</b>	None.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><i>XML_RuleInfo</i></u>

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

**Table B-23 PUT /ISAPI/Intelligent/channels/<ID>/behaviorRule/<SID>/rule/<ID>**

Method	PUT
Description	Set a behavior analysis rule by detection scene ID.
Query	None.
Request	<u><i>XML_RuleInfo</i></u>
Response	<u><i>XML_ResponseStatus</i></u>

**Table B-24 DELETE /ISAPI/Intelligent/channels/<ID>/behaviorRule/<SID>/rule/<ID>**

Method	DELETE
Description	Delete a behavior analysis rule by detection scene ID.
Query	None.
Request	None.
Response	<u><i>XML_ResponseStatus</i></u>

### Remarks

The first <ID> in the request URL refers to the channel No., and the second <ID> refers to the rule ID. The <SID> in the URL is the detection scene ID,

## B.15 /ISAPI/Intelligent/channels/<ID>/behaviorRule/<SID>/rule/capabilities

Get capability of setting behavior analysis rules by detection scene ID.

### Request URI Definition

**Table B-25 GET /ISAPI/Intelligent/channels/<ID>/behaviorRule/<SID>/rule/capabilities**

Method	GET
Description	Get capability of setting behavior analysis rules by detection scene ID.
Query	None.

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

### Remarks

The <ID> in the request URI refers to the channel No., and the <SID> is the detection scene ID.

## B.16 /ISAPI/Intelligent/channels/<ID>/behaviorRule/<SID>/schedules

Operations about arming schedule configuration of behavior analysis of a specific detection scene.

### Request URL Definition

Table B-26 GET /ISAPI/Intelligent/channels/<ID>/behaviorRule/<SID>/schedules

Method	GET
Description	Get arming schedule parameters of behavior analysis of a specific detection scene.
Query	None.
Request	None.
Response	Succeeded: <u><i>XML_RuleSchedule</i></u> Failed: <u><i>XML_ResponseStatus</i></u>

Table B-27 PUT /ISAPI/Intelligent/channels/<ID>/behaviorRule/<SID>/schedules

Method	PUT
Description	Set arming schedule of behavior analysis of a specific detection scene.
Query	None.
Request	<u><i>XML_RuleSchedule</i></u>
Response	<u><i>XML_ResponseStatus</i></u>

### Remarks

The <ID> in the URL refers to the video channel No., and the <SID> is the detection scene No.

## B.17 /ISAPI/Intelligent/channels/<ID>/intelliTrace

Operations about the auto-tracking configurations by channel.

### Request URI Definition

**Table B-28 GET /ISAPI/Intelligent/channels/<ID>/intelliTrace**

<b>Method</b>	GET
<b>Description</b>	Get the auto-tracking configurations by channel.
<b>Query</b>	None.
<b>Request</b>	None.
<b>Response</b>	<u><i>XML_IntelliTraceBlockList</i></u>

**Table B-29 PUT /ISAPI/Intelligent/channels/<ID>/intelliTrace**

<b>Method</b>	PUT
<b>Description</b>	Set the auto-tracking configurations by channel.
<b>Query</b>	None.
<b>Request</b>	<u><i>XML_IntelliTraceBlockList</i></u>
<b>Response</b>	<u><i>XML_ResponseStatus</i></u>

### Remarks

- NVR supports the GET method only.
- The <ID> in the request URI refers to the channel No.

## B.18 /ISAPI/Intelligent/channels/<ID>/intelliTrace/<SID>

Operations about configurations of auto-tracking scene by channel and scene ID.

### Request URI Definition

**Table B-30 GET /ISAPI/Intelligent/channels/<ID>/intelliTrace/<SID>**

<b>Method</b>	GET
<b>Description</b>	Get the configurations of auto-tracking scene by channel and scene ID.

Query	None.
Request	None.
Response	<u><i>XML_IntelliTraceBlock</i></u>

**Table B-31 PUT /ISAPI/Intelligent/channels/<ID>/intelliTrace/<SID>**

Method	PUT
Description	Set the configurations of auto-tracking scene by channel and scene ID.
Query	None.
Request	<u><i>XML_IntelliTraceBlock</i></u>
Response	<u><i>XML_ResponseStatus</i></u>

**Table B-32 DELETE /ISAPI/Intelligent/channels/<ID>/intelliTrace/<SID>**

Method	DELETE
Description	Delete auto-tracking scene by channel and scene ID.
Query	None.
Request	None.
Response	<u><i>XML_ResponseStatus</i></u>

### Remarks

- NVR supports the GET method only.
- The <ID> in the request URI refers to the channel No.
- The <SID> in the URL refers to the scene ID.

## B.19 /ISAPI/Intelligent/channels/<ID>/intelliTrace/capabilities

Get the auto-tracking configuration capability.

### Request URI Definition

**Table B-33 GET /ISAPI/Intelligent/channels/<ID>/intelliTrace/capabilities**

Method	GET
Description	Get the auto-tracking configuration capability.

Query	None.
Request	None.
Response	<u><i>XML_Intelli_IntelliTraceCap</i></u>

**Remarks**

The <ID> in the request URI refers to the channel No.

**B.20 /ISAPI/Intelligent/channels/<ID>/safetyHelmetDetection**

Get or set the parameters of hard hat detection.

**Request URL Definition****Table B-34 GET /ISAPI/Intelligent/channels/<ID>/safetyHelmetDetection**

Method	GET
Description	Get the parameters of hard hat detection.
Query	None.
Request	None.
Response	Succeeded: <u><i>XML_SafetyHelmetDetection</i></u> Failed: <u><i>XML_ResponseStatus</i></u>

**Table B-35 PUT /ISAPI/Intelligent/channels/<ID>/safetyHelmetDetection**

Method	PUT
Description	Set the parameters of hard hat detection.
Query	None.
Request	<u><i>XML_SafetyHelmetDetection</i></u>
Response	<u><i>XML_ResponseStatus</i></u>

**Remarks**

The <ID> in the request URL refers to the channel No.

## B.21 /ISAPI/Intelligent/channels/<ID>/safetyHelmetDetection/advanceConfiguration

Get or set advanced parameters of hard hat detection.

### Request URL Definition

**Table B-36 GET /ISAPI/Intelligent/channels/<ID>/safetyHelmetDetection/advanceConfiguration**

<b>Method</b>	GET
<b>Description</b>	Get advanced parameters of hard hat detection.
<b>Query</b>	None.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><i>XML_AdvanceConfiguration</i></u> Failed: <u><i>XML_ResponseStatus</i></u>

**Table B-37 PUT /ISAPI/Intelligent/channels/<ID>/safetyHelmetDetection/advanceConfiguration**

<b>Method</b>	PUT
<b>Description</b>	Set advanced parameters of hard hat detection.
<b>Query</b>	None.
<b>Request</b>	<u><i>XML_AdvanceConfiguration</i></u>
<b>Response</b>	<u><i>XML_ResponseStatus</i></u>

### Remarks

The <ID> in the request URL refers to the channel No.

## B.22 /ISAPI/Intelligent/channels/<ID>/safetyHelmetDetection/capabilities

Get the configuration capability of hard hat detection.

## Request URL Definition

Table B-38 GET /ISAPI/Intelligent/channels/<ID>/safetyHelmetDetection/capabilities

Method	GET
Description	Get the configuration capability of hard hat detection.
Query	None.
Request	None.
Response	Succeeded: <u><i>XML_SafetyHelmetDetectionCap</i></u> Failed: <u><i>XML_ResponseStatus</i></u>

### Remarks

The <ID> in the request URL refers to the channel No.

## B.23 /ISAPI/Intelligent/channels/<ID>/safetyHelmetDetection/search?format=json

Search for hard hat detection events.

## Request URI Definition

Table B-39 POST /ISAPI/Intelligent/channels/<ID>/safetyHelmetDetection/search?format=json

Method	POST
Description	Search for hard hat detection events.
Query	<b>format:</b> determine the format of request or response message.
Request	<u><i>JSON_SearchDescription</i></u>
Response	Succeeded: <u><i>JSON_SearchResult</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

### Remarks

The <ID> in the URI refers to channel ID.



## B.24 /ISAPI/Intelligent/channels/<ID>/zoomRatio/<SID>/goto?format=json

Optimize the zooming ratio in behavior analysis scene.

### Request URI Definition

Table B-40 PUT /ISAPI/Intelligent/channels/<ID>/zoomRatio/<SID>/goto?format=json

Method	PUT
Description	Optimize the zooming ratio in behavior analysis scene.
Query	<b>format</b> : determine the format of request or response message.
Request	None.
Response	<u><i>JSON_ResponseStatus</i></u>

### Remarks

- The <ID> in the URI refers to the channel No.
- The <SID> in the URI refers to the scene ID.

## B.25 /ISAPI/Intelligent/channels/<ID>/zoomRatio/capabilities

Get the configuration capability of zooming ratio in behavior analysis scene.

### Request URI Definition

Table B-41 GET /ISAPI/Intelligent/channels/<ID>/zoomRatio/capabilities

Method	GET
Description	Get the configuration capability of zooming ratio in behavior analysis scene.
Query	None.
Request	None.
Response	Succeeded: <u><i>XML_ZoomRatioCap</i></u> Failed: <u><i>XML_ResponseStatus</i></u>

### Remarks

The <ID> in the URI refers to the channel No.

## B.26 /ISAPI/Smart/capabilities

Get the capability of smart devices.

### Request URI Definition

**Table B-42 GET /ISAPI/Smart/capabilities**

<b>Method</b>	GET
<b>Description</b>	Get the capability of smart devices.
<b>Query</b>	None
<b>Request</b>	None
<b>Response</b>	Succeeded: <u><i>XML_SmartCap</i></u> Failed: <u><i>XML_ResponseStatus</i></u>

## B.27 /ISAPI/Smart/RadarChannels/<ID>/radarFieldDetection/capabilities?format=json

Get radar intrusion detection capability.

### Request URI Definition

**Table B-43 GET /ISAPI/Smart/RadarChannels/<ID>/radarFieldDetection/capabilities?format=json**

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

### Remarks

The <ID> in the request URI refers to the ID of radar channel for intrusion detection

## B.28 /ISAPI/Smart/RadarChannels/<ID>/radarFieldDetection/regions/<regionID>?format=json

Get or set parameters of a radar detection region.

### Request URI Definition

**Table B-44 GET /ISAPI/Smart/RadarChannels/<ID>/radarFieldDetection/regions/<regionID>?format=json**

<b>Method</b>	GET
<b>Description</b>	Get parameters of a radar detection region.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None
<b>Response</b>	Succeeded: <u><i>JSON_RegionInfo</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

**Table B-45 PUT /ISAPI/Smart/RadarChannels/<ID>/radarFieldDetection/regions/<regionID>?format=json**

<b>Method</b>	PUT
<b>Description</b>	Set parameters of a radar detection region.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<u><i>JSON_RegionInfo</i></u>
<b>Response</b>	<u><i>JSON_ResponseStatus</i></u>

### Remarks

The <ID> in the request URI refers to the ID of radar channel for intrusion detection. And the <regionID> in the URI is the detection region No.

## B.29 /ISAPI/Smart/RadarChannels/<ID>/radarFieldDetection/regions?format=json

Get or set parameters of all radar detection regions.

## Request URI Definition

Table B-46 GET /ISAPI/Smart/RadarChannels/<ID>/radarFieldDetection/regions?format=json

Method	GET
Description	Get parameters of all radar detection regions.
Query	<b>format</b> : determine the format of request or response message.
Request	None
Response	Succeeded: <i><u>JSON RegionList</u></i> Failed: <i><u>JSON ResponseStatus</u></i>

Table B-47 PUT /ISAPI/Smart/RadarChannels/<ID>/radarFieldDetection/regions?format=json

Method	PUT
Description	Set parameters of all radar detection regions.
Query	<b>format</b> : determine the format of request or response message.
Request	<i><u>JSON RegionList</u></i>
Response	<i><u>JSON ResponseStatus</u></i>

## Remarks

The <ID> in the request URI refers to the ID of radar channel for intrusion detection.

## B.30 /ISAPI/Smart/RadarChannels/<ID>/radarFieldDetection?format=json

Get or set parameters of radar intrusion detection.

## Request URI Definition

Table B-48 GET /ISAPI/Smart/RadarChannels/<ID>/radarFieldDetection?format=json

Method	GET
Description	Get parameters of radar intrusion detection.
Query	<b>format</b> : determine the format of request or response message.
Request	None
Response	Succeeded: <i><u>JSON RadarFieldDetection</u></i>

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

**Table B-49 PUT /ISAPI/Smart/RadarChannels/<ID>/radarFieldDetection?format=json**

Method	PUT
Description	Set parameters of radar intrusion detection.
Query	<b>format:</b> determine the format of request or response message.
Request	<u><i>JSON_RadarFieldDetection</i></u>
Response	<u><i>JSON_ResponseStatus</i></u>

**Remarks**

The <ID> in the request URI refers to the ID of radar channel for intrusion detection.

**B.31 /ISAPI/Smart/RadarChannels/<ID>/radarLineDetection/capabilities?format=json**

Get radar line crossing detection capability.

**Request URI Definition****Table B-50 GET /ISAPI/Smart/RadarChannels/<ID>/radarLineDetection/capabilities?format=json**

Method	GET
Description	Get radar line crossing detection capability.
Query	<b>format:</b> determine the format of request or response message.
Request	None
Response	Succeeded: <u><i>JSON_RadarLineDetectionCap</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

**Remarks**

The <ID> in the request URI refers to the ID of radar channel for line crossing detection.

**B.32 /ISAPI/Smart/RadarChannels/<ID>/radarLineDetection/lineItems/<lineID>?format=json**

Get or set parameters of a radar detection line.

## Request URI Definition

**Table B-51 GET /ISAPI/Smart/RadarChannels/<ID>/radarLineDetection/lineItems/<lineID>?format=json**

<b>Method</b>	GET
<b>Description</b>	Get parameters of a radar detection line.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None
<b>Response</b>	Succeeded: <u><i>JSON_LineItemInfo</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

**Table B-52 PUT /ISAPI/Smart/RadarChannels/<ID>/radarLineDetection/lineItems/<lineID>?format=json**

<b>Method</b>	PUT
<b>Description</b>	Set parameters of a radar detection line.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<u><i>JSON_LineItemInfo</i></u>
<b>Response</b>	<u><i>JSON_ResponseStatus</i></u>

## Remarks

The <ID> in the request URI refers to the ID of radar channel for line crossing detection. And the<lineID> in the URI is the detection line No.

## B.33 /ISAPI/Smart/RadarChannels/<ID>/radarLineDetection/lineItems?format=json

Get or set parameters of all radar detection lines.

## Request URI Definition

**Table B-53 GET /ISAPI/Smart/RadarChannels/<ID>/radarLineDetection/lineItems?format=json**

<b>Method</b>	GET
<b>Description</b>	Get parameters of all radar detection lines.
<b>Query</b>	<b>format:</b> determine the format of request or response message.

<b>Request</b>	None
<b>Response</b>	Succeeded: <u><i>JSON_LineItem</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

**Table B-54 PUT /ISAPI/Smart/RadarChannels/<ID>/radarLineDetection/lineItems?format=json**

<b>Method</b>	PUT
<b>Description</b>	Set parameters of all radar detection lines.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<u><i>JSON_LineItem</i></u>
<b>Response</b>	<u><i>JSON_ResponseStatus</i></u>

### Remarks

The <ID> in the request URI refers to the ID of radar channel for line crossing detection.

## B.34 /ISAPI/Smart/RadarChannels/<ID>/radarLineDetection?format=json

Get or set radar line crossing detection parameters.

### Request URI Definition

**Table B-55 GET /ISAPI/Smart/RadarChannels/<ID>/radarLineDetection?format=json**

<b>Method</b>	GET
<b>Description</b>	Get radar line crossing detection parameters.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None
<b>Response</b>	Succeeded: <u><i>JSON_RadarLineDetection</i></u> Failed: <u><i>JSON_ResponseStatus</i></u>

**Table B-56 PUT /ISAPI/Smart/RadarChannels/<ID>/radarLineDetection?format=json**

<b>Method</b>	PUT
<b>Description</b>	Set radar line crossing detection parameters.
<b>Query</b>	<b>format:</b> determine the format of request or response message.

<b>Request</b>	<u><i>JSON_RadarLineDetection</i></u>
<b>Response</b>	<u><i>JSON_ResponseStatus</i></u>

### Remarks

The <ID> in the request URI refers to the ID of radar channel for line crossing detection.

## B.35 /ISAPI/System/capabilities

Get device capability.

### Request URI Definition

**Table B-57 GET /ISAPI/System/capabilities**

<b>Method</b>	GET
<b>Description</b>	Get device capability.
<b>Query</b>	None
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><i>XML_DeviceCap</i></u> Failed: <u><i>XML_ResponseStatus</i></u>



## Appendix C. Request and Response Messages

### C.1 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*/
}
```

### C.2 JSON\_EventNotificationAlert\_safetyHelmetDetection

The details about hard hat detection alarm are uploaded in JSON format of EventNotificationAlert message, here shows an example.

## Hard Hat Detection Alarm with Binary Picture Data

```
Content-Type: multipart/form-data; boundary=MIME_boundary
--MIME_boundary
Content-Type: application/json
Content-Length: 480

{
  "ipAddress":,
  /*required, string, IPv4 address of alarm device, the maximum size is 32 bytes*/
  "ipv6Address": "",
  /*optional, string, IPv6 address of alarm device, the maximum size is 128
bytes*/
  "portNo":,
  /*optional, integer32, port No. of alarm device*/
  "protocolType": "",
  /*optional, string, protocol type: "HTTP", "HTTPS", "EHome"; the maximum size
is 32 bytes*/
  "macAddress": "",
  /*optional, string, MAC address, the maximum size is 32 bytes*/
  "channelID":,
  /*optional, integer32, device channel No. triggering alarm*/
  "relatedChannelList": [1,2,3],
  /*optional, array of integers, list of alarm related channels, which are of the
same camera with channelID; this parameter is used for live view or playback on
the platform*/
  "dateTime": "",
  /*required, string, alarm triggered time (ISO 8601 time format), the maximum
size is 32 bytes*/
  "activePostCount":,
  /*required, integer32, uploaded times of one alarm*/
  "eventType": "",
  /*required, string, triggered event type, here it should be set to
"safetyHelmetDetection" (hard hat detection alarm); the maximum size is 128
bytes*/
  "eventState": "",
  /*required, string, event triggering status: "active"-triggered, "inactive"-not
triggered (heartbeat data); the maximum size is 32 bytes*/
  "eventDescription": "",
  /*required, string, event description*/
  "channelName": "",
  /*required, string, channel name (camera name)*/
  "deviceID": "",
  /*optional, device ID which is also the PUID. This field must be returned when
passing through ISAPI event information by ISUP*/
  "Target": [{
  /*array, target, multiple targets can be output after being recognized once*/
    "recognitionType": "",
    /*required, string, recognition type, here it should be set to "safetyHelmet"
(hard hat detection)*/
    "TargetInfo": {
```

```
/*target information*/
    "recognition": "",
/*required, string, recognition type, here it should be set to "safetyHelmet"
(hard hat detection)*/
    "dataSource": ""
/*required, detection data source: "timing"-scheduled, "trigger" (when
exceeding the configured level limit)*/
    },
    "rect": [{
/*optional, rectangle frame of hard hat*/
        "height":,
/*optional, float, height*/
        "width":,
/*optional, float, width*/
        "x":,
/*optional, float, X-coordinate*/
        "y":
/*optional, float, Y-coordinate*/
    }],
    "contentID": "",
/*optional, string, picture content ID, the maximum size is 256 bytes*/
    "pId": ""
/*string, thumbnail ID of no hard hat detection alarm, the maximum length is 64
characters*/
    "faceContrastResult": "",
/*optional, string, face picture comparison result, this node is returned only
when the hard hat detection is linked to face picture comparison*/
    "Face": {
/*optional, face information, this node is returned when faceContrastResult is
"success"*/
        "FDID": "",
/*required, string, face picture library ID, the maximum size is 63 bytes*/
        "FPID": "",
/*optional, string, face record ID, if it is inputted, it should be unique ID
consisting of letters and digits, and the maximum size is 63 bytes*/
        "contentID": ""
/*optional, string, picture content ID, the maximum size is 256 bytes*/
        "pId": "",
/*optional, string, ID of the matched picture in the library, the maximum size
is 64 bytes*/
    }
    },
    "contentID": "",
    "pId": ""
/*string, background picture ID, the maximum length is 64 characters*/
}
--MIME_boundary
Content-Disposition: form-data; name="F4F665D6A18E41308CE9934DCDDD1111";
filename=" background_image.jpg";
Content-Type: image/jpeg
Content-Length: 516876
Content-ID: background_image
```

```
fefefwageegfqaeg...//picture data (the background picture is prior to the
thumbnail of no hard hat detection alarm when both of them need to be uploaded)
--MIME_boundary
Content-Disposition: form-data; name="F4F665D6A18E41308CE9934DCDDD2222";
filename="safetyHelmet1_image.jpg";
Content-Type: image/jpeg
Content-Length: 516876
Content-ID: safetyHelmet1_image

fefefwageegfqaeg...//picture data
--MIME_boundary--
```

### Hard Hat Detection Alarm with Picture URL

```
Content-Type: multipart/form-data; boundary=MIME_boundary
--MIME_boundary
Content-Type: application/json
Content-Length: 480

{
  "ipAddress":,
  /*required, string, IPv4 address of alarm device, the maximum size is 32 bytes*/
  "ipv6Address": "",
  /*optional, string, IPv6 address of alarm device, the maximum size is 128
  bytes*/
  "portNo":,
  /*optional, integer32, port No. of alarm device*/
  "protocolType": "",
  /*optional, string, protocol type: "HTTP", "HTTPS", the maximum size is 32
  bytes*/
  "macAddress": "",
  /*optional, string, MAC address, the maximum size is 32 bytes*/
  "channelID":,
  /*optional, integer32, device channel No. triggering alarm*/
  "dateTime": "",
  /*required, string, alarm triggered time (ISO 8601 time format), the maximum
  size is 32 bytes*/
  "activePostCount":,
  /*required, integer32, uploaded times of one alarm*/
  "eventType": "",
  /*required, string, triggered event type, here it should be set to
  "safetyHelmetDetection" (hard hat detection alarm); the maximum size is 128
  bytes*/
  "eventState": "",
  /*required, string, event triggering status: "active"-triggered, "inactive"-not
  triggered (heartbeat data); the maximum size is 32 bytes*/
  "eventDescription": "",
  /*required, event description*/
  "channelName": "",
  /*required, channel name (camera name)*/
  "deviceID": "",
```

```
/*optional, device ID which is also the PUID. This field must be returned when
passing through ISAPI event information by ISUP*/
    "Target":[{
/*array, target, multiple targets can be output after being recognized once*/
        "recognitionType": "",
/*required, string, recognition type, here it should be set to "safetyHelmet"
(hard hat detection)*/
        "TargetInfo":{
/*target information*/
            "recognition": "",
/*required, string, recognition type, here it should be set to "safetyHelmet"
(hard hat detection)*/
            "dataSource": ""
/*required, detection data source: "timing"-scheduled, "trigger" (when
exceeding the configured level limit)*/
        },
        "rect":[{
/*optional, rectangle frame of hard hat*/
            "height":,
/*optional, float, height*/
            "width":,
/*optional, float, width*/
            "x":,
/*optional, float, X-coordinate*/
            "y":
/*optional, float, Y-coordinate*/
        }],
        "safetyHelmetURL": ""
/*string, thumbnail URL of no hard hat detection alarm*/
        "faceContrastResult": "",
/*optional, string, face picture comparison result, "success,failed"*/
        "Face":{
/*optional, face picture information, this node is returned when
faceContrastResult is "success"*/
            "FDID": "",
/*required, string, face picture library ID, the maximum size is 63 bytes*/
            "FPID": "",
/*optional, string, face picture record ID, if it is inputted, it should be the
unique ID consisting of letters and digits, and the maximum length is 63 bytes*/
            "libFacePicURL": "",
/*optional, string, URL of the matched picture in the library, the maximum size
is 256 bytes*/
        }
    }],
    "imageUrl": "",
/*optional, string, background picture URL, the maximum size is 256 bytes*/
    "URLCertificationType": ""
/*optional, string, picture URL authentication mode: "no"-no authentication,
"digest"-digest authentication (it is used for devices (such as DVR, NVR, etc.)
to store returned URLs locally); the maximum size is 32 bytes*/
}
--MIME_boundary--
```

## C.3 JSON\_LineItemInfo

JSON message about detection line parameters

```
{
  "LineItemInfo": {
    "lineID": ,
    /*required, int, detection line ID*/
    "enabled": ,
    /*required, boolean, whether to enable detection*/
    "sensitivityLevel": ,
    /*required, int, detection sensitivity*/
    "directionSensitivity": "",
    /*required, string, detection direction: "left-right" (from left to right),
    "right-left" (from right to left), "any" (all directions)*/
    "LineItem": [{
      /*required, parameters of a detection line*/
      "x": ,
      /*required, float, x-coordinate, its value is between 0.001 and 1*/
      "y":
      /*required, float, y-coordinate, its value is between 0.001 and 1*/
    }]
  }
}
```

## C.4 JSON\_LineItemList

JSON message about detection line list

```
{
  "LineItemList": [{
    "LineItemInfo": {
      "lineID": ,
      /*required, int, detection line ID*/
      "enabled": ,
      /*required, boolean, whether to enable detection*/
      "sensitivityLevel": ,
      /*required, int, detection sensitivity*/
      "directionSensitivity": "",
      /*required, string, detection direction: "left-right" (from left to right),
      "right-left" (from right to left), "any" (all directions)*/
      "LineItem": [{
        /*required, parameters of a detection line*/
        "x": ,
        /*required, float, x-coordinate, its value is between 0.001 and 1*/
        "y":
        /*required, float, y-coordinate, its value is between 0.001 and 1*/
      }]
    }
  ]
}
```

```
    ]]  
  }
```

## C.5 JSON\_RadarFieldDetection

JSON message about radar intrusion detection parameters

```
{  
  "RadarFieldDetection": {  
    /*required, radar intrusion detection parameters*/  
    "enabled": ,  
    /*required, boolean, whether to enable detection*/  
    "intelliBackSearch": ,  
    /*optional, boolean, whether it supports dual-VCA search*/  
    "startTriggerTime": ,  
    /*required, int, triggering start time, unit: millisecond*/  
    "endTriggerTime": ,  
    /*required, int, triggering end time, unit: millisecond*/  
    "minObjectSize": ,  
    /*optional, int, minimum number of pixels*/  
    "maxObjectSize": ,  
    /*optional, int, maximum number of pixels*/  
    "RegionList": [{  
    /*required, detection region list, the region can contains 3 to 10 edges*/  
      "RegionInfo": {  
        "regionID": ,  
        "enabled": ,  
        "sensitivityLevel": ,  
        /*required, int, detection sensitivity*/  
        "objectOccupation": ,  
        /*required, int*/  
        "Region": [{  
          "x": ,  
          /*required, float, x-coordinate, its value is between 0.001 and 1*/  
          "y": ,  
          /*required, float, y-coordinate, its value is between 0.001 and 1*/  
        }]  
      }  
    }],  
    "humanMisinfoFilter": ,  
    /*optional, boolean, whether it supports preventing false human body detection  
    alarm; if it does not support, this node will not be returned*/  
    "vehicleMisinfoFilter": ,  
    /*optional, boolean, whether it supports preventing false vehicle detection  
    alarm; if it does not support, this node will not be returned*/  
  }  
}
```

## C.6 JSON\_RadarFieldDetectionCap

JSON message about capability of radar intrusion detection

```
{
  "RadarFieldDetectionCap": {
    /*required, radar intrusion detection capability*/
    "enabled": [true, false],
    /*required, boolean, whether to enable detection*/
    "intelliBackSearch": ,
    /*optional, boolean, whether it supports dual-VCA search; if it does not
    support, this node will not be returned*/
    "startTriggerTime": {
      /*required, int, triggering start time, unit: millisecond*/
      "@min": 1,
      "@max": 100
    },
    "endTriggerTime": {
      /*required, int, triggering end time, unit: millisecond*/
      "@min": 0,
      "@max": 0
    },
    "minObjectSize": {
      /*optional, int, minimum number of pixels*/
      "@min": 0,
      "@max": 0
    },
    "maxObjectSize": {
      /*optional, int, maximum number of pixels*/
      "@min": 0,
      "@max": 0
    },
    "RegionListCap": {
      /*required, detection region parameters, the region can contains 3 to 10 edges*/
      "maxSize": 4,
      /*required, int, maximum number of detection regions*/
      "enabled": [true, false],
      /*required, boolean, whether to enable the detection of this region*/
      "sensitivityLevel": {
        /*required, int, detection sensitivity*/
        "@min": 1,
        "@max": 100
      },
      "objectOccupation": {
        /*required, int*/
        "@min": 1,
        "@max": 1
      },
      "RegionCap": {
        "minSize": 3,
        /*required, int, minimum number of detection region edges*/
```



```
    "maxSize": 10,
    /*required, int, maximum number of detection region edges*/
    "x": {
    /*required, float, x-coordinate, its value is between 0.001 and 1*/
        "@min": 0.000,
        "@max": 1.000
    },
    "y": {
    /*required, float, y-coordinate, its value is between 0.001 and 1*/
        "@min": 0.000,
        "@max": 1.000
    }
    },
    "isSupportMultiScene": ,
    /*optional, boolean, whether it supports multi-scene mode; if it does not
    support, this node will not be returned*/
    "humanMisinfoFilter": ,
    /*optional, boolean, whether it supports preventing false human body detection
    alarm; if it does not support, this node will not be returned*/
    "vehicleMisinfoFilter": ,
    /*optional, boolean, whether it supports preventing false vehicle detection
    alarm; if it does not support, this node will not be returned*/
    "isSupportTargetMultiSelect": ,
    /*optional, boolean, whether it supports multi-target detection; if it does not
    support, this node will not be returned*/
    "isSupportAllDayUpload":
    /*optional, boolean, whether it supports uploading all-day events; if it does
    not support, this node will not be returned*/
    }
}
```

## C.7 JSON\_RadarLineDetection

JSON message about radar line crossing detection parameters

```
{
  "RadarLineDetection": {
    /*required, radar line crossing detection parameters*/
    "enabled": ,
    /*required, boolean, whether to enable detection*/
    "intelliBackSearch": ,
    /*optional, boolean, whether it supports dual-VCA search*/
    "duration": ,
    /*required, int, time duration*/
    "startTriggerTime": ,
    /*required, int, triggering start time, unit: millisecond*/
    "endTriggerTime": ,
    /*required, int, triggering end time, unit: millisecond*/
    "minObjectSize": ,
  }
}
```

```
/*optional, int, minimum number of pixels*/
  "maxObjectSize": ,
/*optional, int, maximum number of pixels*/
  "LineItemList": [{
/*required, detection line list*/
    "LineItemInfo": {
      "lineID": ,
/*required, int, detection line ID*/
      "enabled": ,
/*required, boolean, whether to enable detection of the line*/
      "sensitivityLevel": ,
/*required, int, detection sensitivity*/
      "directionSensitivity": "",
/*required, string, detection direction: "left-right" (from left to right),
"right-left" (from right to left), "any" (all directions)*/
      "LineItem": [{
/*required, parameters of a detection line*/
        "x": ,
/*required, float, x-coordinate, its value is between 0.001 and 1*/
        "y": ,
/*required, float, y-coordinate, its value is between 0.001 and 1*/
      }]
    }
  ]},
  "recogRuleType": "",
/*optional, string, direction recognition mode: "vectorMode" (vector mode),
"slopeMode" (slope mode)*/
  "humanMisinfoFilter": ,
/*optional, boolean, whether it supports preventing false human body detection
alarm; if it does not support, this node will not be returned*/
  "vehicleMisinfoFilter": ,
/*optional, boolean, whether it supports preventing false vehicle detection
alarm; if it does not support, this node will not be returned*/
}
}
```

## C.8 JSON\_RadarLineDetectionCap

JSON message about capability of radar line crossing detection

```
{
  "RadarLineDetectionCap": {
/*required, radar line crossing detection capability*/
    "enabled": [true, false],
/*required, boolean, whether to enable detection*/
    "intelliBackSearch": ,
/*optional, boolean, whether it supports dual-VCA search; if it does not
support, this node will not be returned*/
    "duration": {
/*required, int, time duration*/
```

```
        "@min": 0,
        "@max": 0
    },
    "startTriggerTime": {
/*required, int, triggering start time, unit: millisecond*/
        "@min": 0,
        "@max": 0
    },
    "endTriggerTime": {
/*required, int, triggering end time, unit: millisecond*/
        "@min": 0,
        "@max": 0
    },
    "minObjectSize": {
/*optional, int, minimum number of pixels*/
        "@min": 0,
        "@max": 0
    },
    "maxObjectSize": {
/*optional, int, maximum number of pixels*/
        "@min": 0,
        "@max": 0
    },
    "LineItemListCap": {
/*required, detection line list*/
        "maxSize": 4,
/*required, int, maximum number of detection lines*/
        "enabled": [true, false],
/*required, boolean, whether to enable detection of the line*/
        "sensitivityLevel": {
/*required, int, detection sensitivity*/
            "@min": 1,
            "@max": 100
        },
        "directionSensitivity": {
/*required, string, detection direction: "left-right" (from left to right),
"right-left" (from right to left), "any" (all directions)*/
            "@opt": ["left-right", "right-left", "any"],
            "@def": "any"
        },
        "LineItemCap": {
/*required, parameters of a detection line*/
            "maxSize": 2,
/*required, int, maximum number of points on the detection line*/
            "x": {
/*required, float, x-coordinate, its value is between 0.001 and 1*/
                "@min": 0.000,
                "@max": 1.000,
            },
            "y": {
/*required, float, y-coordinate, its value is between 0.001 and 1*/
                "@min": 0.000,
```

```
        "@max": 1.000,
    },
    },
    },
    "isSupportMultiScene": ,
    /*optional, boolean, whether it supports multi-scene mode; if it does not
    support, this node will not be returned*/
    "recogRuleType": {
    /*optional, string, direction recognition mode: "vectorMode" (vector mode),
    "slopeMode" (slope mode)*/
        "@opt": "vectorMode,slopeMode"
    },
    "humanMisinfoFilter": ,
    /*optional, boolean, whether it supports preventing false human body detection
    alarm; if it does not support, this node will not be returned*/
    "vehicleMisinfoFilter": ,
    /*optional, boolean, whether it supports preventing false vehicle detection
    alarm; if it does not support, this node will not be returned*/
    "isSupportTargetMultiSelect": ,
    /*optional, boolean, whether it supports multi-target detection; if it does not
    support, this node will not be returned*/
    "isSupportAllDayUpload":
    /*optional, boolean, whether it supports uploading all-day events; if it does
    not support, this node will not be returned*/
    }
}
```

### C.9 JSON\_RegionInfo

JSON message about detection region parameters

```
{
  "RegionInfo": {
    "regionID": ,
    "enabled": ,
    "sensitivityLevel": ,
    /*required, int, region detection sensitivity*/
    "objectOccupation": ,
    /*required, int*/
    "Region": [{
      "x": ,
      /*required, float, x-coordinate, its value is between 0.001 and 1*/
      "y":
      /*required, float, Y-coordinate, its value is between 0.001 and 1*/
    }]
  }
}
```

## C.10 JSON\_RegionList

JSON message about detection region list

```
{
  "RegionList": [{
    "RegionInfo": {
      "regionID": ,
      "enabled": ,
      "sensitivityLevel": ,
      /*required, int, region detection sensitivity*/
      "objectOccupation": ,
      /*required, int*/
      "Region": [{
        "x": ,
        /*required, float, x-coordinate, its value is between 0.001 and 1*/
        "y":
        /*required, float, y-coordinate, its value is between 0.001 and 1*/
      }]
    }
  }]
}
```

## C.11 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*/
}
```

### C.12 JSON\_SearchDescription

JSON message about hard hat detection search conditions

```
{
  "SearchDescription":{
    "searchID": "",
    /*required, string, search record ID*/
    "searchResultPosition": ,
    /*required, int, 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": ,
    /*required, int, the maximum number of records matched in a single search*/
    "timeSpanList": [{
      "startTime": "",
      /*required, string, start time in ISO8601 format, e.g.,
      2004-05-03T17:30:08+08:00*/
      "endTime": ""
      /*required, string, end time in ISO8601 format, e.g.,
      2004-05-03T17:30:08+08:00*/
    }]
    "faceContrastResult":"","
    /*required, comparison result: "all"-all are succeeded or failed, "success",
    "failed"*/
    "FDLibList": [{
      /*list library list, this node is valid when the value of "faceContrastResult"
      is "success"*/
      "FDLib":{
        "FDID":""
        /*optional, string, list library ID*/
      }
    }],
    "name":""
    /*optional, xs:string, person name, this node is valid when the "FDLibList"
    does exist*/
  }
}
```

### C.13 JSON\_SearchResult

JSON message about hard hat detection search result

```
{
  "SearchResult":{
```

```
"responseStatusStrg": "",
/*required, string, search status description: "OK"-search ended, "MORE"-
searching, "NO_MATCHES"-no data found, the maximum size is 32 bytes, {dep if
errcode == 1 && errMsg == ok}*/
"numOfMatches": ,
/*required, integer32, number of records matched in this search, {dep if
errcode == 1 && errMsg == ok}*/
"totalMatches": ,
/*required, integer32, total number of records matched, {dep if errcode == 1 &&
errMsg == ok}*/
"targets": [{
/*optional, search result*/
"time": "",
/*required, statistics time in ISO8601 format, e.g., 2017-08-25T11:34:59+08:00*/
"channel": ,
/*required, int, channel No.*/
"Target":{
/*required, target information*/
"Rect": [{
/*optional, target frame*/
"targetID": ,
/*optional, target ID*/
"height": ,
/*required, float, height*/
"width": ,
/*required, float, width*/
"x": ,
/*required, float, X-coordinate (the upper-left corner is the origin)*/
"y":
/*required, float, Y-coordinate (the upper-left corner is the origin)*/
}],
"face":[{
/*face picture information, this node is returned only when face picture
comparison is linked to hard hat detection*/
"FDID": "",
/*optional, string, face picture library ID, the maximum size is 63 bytes*/
"FPID": "",
/*optional, string, face record ID, if it is inputted, it should be the unique
ID consisting of letters and digits, and the maximum length is 63 bytes*/
"name": "",
/*optional, string, name*/
"captureFacePicURL": "",
/*required, string, captured picture URL, the maximum size is 256 bytes, the
captured picture will be returned no matter the face picture comparison is
successful or not*/
"libFacePicURL": "",
/*optional, string, URL of the matched picture in the library, the maximum size
is 256 bytes*/
}],
},
"backgroundImageURL": "",
/*background picture URL*/
```

```
    }}  
  }  
}
```

### C.14 XML\_AdvanceConfiguration

XML message about advanced parameters

```
<AdvanceConfiguration version="2.0" xmlns="http://www.isapi.org/ver20/  
XMLSchema">  
  <headSensitivity>  
    <!--optional, xs:integer, head detection sensitivity, it is between 1 and 5  
and the default value is 3, the larger value indicates that the detection is  
more sensitive-->  
  </headSensitivity>  
  <humanSensitivity>  
    <!--optional, xs:integer, human body detection sensitivity, it is between 1  
and 5 and the default value is 3, the larger value indicates that the detection  
is more sensitive-->  
  </humanSensitivity>  
  <alarmAnalysisFrame>  
    <!--optional, xs:integer, number of frames required for alarm analysis, it  
is between 1 and 50 and the default value is 12, the larger value indicates  
that more frames need to be analyzed, and the alarm will be triggered more  
slowly-->  
  </alarmAnalysisFrame>  
  <FDLibList><!--optional, channel's linked face picture library list-->  
    <FDLib><!--list-->  
      <id><!--required, ro, xs:integer, No. of items in the list--></id>  
      <FDID><!--required, ro, xs:string, face picture library ID--></FDID>  
      <thresholdValue><!--optional, xs:integer, xs:integer, similarity  
threshold for comparison, which is between 0 and 100, the higher the threshold  
is, the more accurate the comparison is. If the threshold for all libraries is  
enabled, then threshold for single library will not take effect--></  
thresholdValue>  
      <customFaceLibID><!--optional, xs:string, custom face picture library ID,  
custom ID has priority over the FDID--></customFaceLibID>  
    </FDLib>  
  </FDLibList>  
  <FaceScore><!--face picture scoring configuration, this node is only  
available for PUT operation-->  
    <enabled><!--optional, xs:boolean, whether to enable face picture scoring  
configuration--></enabled>  
    <featureConfidence><!--optional, feature confidence-->  
      <medium><!--medium value-->  
        <leftInterval><!--required, xs:integer, left interval: [0-100]--></  
leftInterval>  
        <rightInterval><!--required, xs:integer, right interval: [0-100]--></  
rightInterval>  
      </medium>
```



```
</featureConfidence>
<pupillaryDistance ><!--optional, pupil distance-->
  <medium><!--medium value-->
    <leftInterval><!--required, xs:integer, left interval--></leftInterval>
    <rightInterval><!--required, xs:integer, right interval--></
rightInterval>
  </medium>
</pupillaryDistance>
<pitchAngle><!--optional, pitch angle-->
  <medium><!--medium value-->
    <leftInterval><!--required, xs:integer, left interval, unit: degree--></
leftInterval>
    <rightInterval><!--required, xs:integer, right interval, unit: degree--
></rightInterval>
  </medium>
</pitchAngle>
<yawAngle><!--optional, yaw angle-->
  <medium><!--medium value-->
    <leftInterval><!--required, xs:integer, left interval, unit: degree--></
leftInterval>
    <rightInterval><!--required, xs:integer, right interval, unit: degree--
></rightInterval>
  </medium>
</yawAngle>
</FaceScore>
</AdvanceConfiguration>
```

## C.15 XML\_AnalysisEngine

### XML message about analysis engine parameters

```
<AnalysisEngine version="2.0" xmlns="http://www.std-cgi.org/ver20/XMLSchema">
  <id>
    <!--required, xs: integer, engine ID-->
  </id>
  <mode>
    <!--required, xs: string, an intelligent analysis engine only supports one
mode: face_recognition, video_structure_preview, video_structure_record,
image_recognition_human, image_recognition_vehicle, safetyHelmetDetection,
personDensityDetection, leavePosition, peopleNumChange, behavior, retention,
situationAnalysis, cityManagement, face_human_recognition,
face_recognition_IES, OCR, face_recognition_IES_or_OCR, fireEscapeDetection,
takingElevatorDetection, mixed-->
  </mode>
  <state>
    <!--required, ro, xs: string, engine status: "loading"-initializing,
"analyzing"-running, "abnormal"-exception, "connect_boot_abnorma"-connecting to
boot program failed, "conig_boot_abnorma"-configuring boot program failed,
"connect_main_abnorma"-connecting main program failed, "conig_main_abnorma"-
configuring main program failed, "auth_abnorma"-authentication failed,
```

```
"reset_abnorma"-resetting failed, "DSP_status_abnorma"-DSP status exception,
"lib_ver_err_abnorma"-incorrect remote engine version-->
</state>
<utilization>
  <!--optional, ro, xs: float, engine usage, it is a read-only node and
accurate to 1 decimal place, e.g., "00.0"-->
</utilization>
<ChannelInfoList><!--optional, read-only, channel No. linked to the engine,
read-only node-->
  <ChannelInfo>
    <id>
      <!--required, xs: integer, NVR channel, including analog channel No.
and digital channel No.-->
    </id>
    <name>
      <!--optional, xs: string, channel name-->
    </name>
  </ChannelInfo>
</ChannelInfoList>
<repairTime><!--optional, read-only, xs: integer, number of automatic
repairs, from 1 to N: number of repairing attempts, -1: repairing failed--></
repairTime>
<channelNum><!--optional, xs:integer, supported number of channels to be
linked with the engine--></channelNum>
<analysisMode><!--optional, xs:string, analysis mode: realTimeMode(default),
autoSwitchMode--></analysisMode>
<AutoSwitchTime><!--optional, xs:integer, auto-switch time interval, unit:
second--></AutoSwitchTime>
</AnalysisEngine>
```

## C.16 XML\_AnalysisEnginesCap

XML message about configuration capability of analysis engine

```
<AnalysisEnginesCap version="2.0" xmlns="http://www.std-cgi.org/ver20/
XMLSchema">
  <supportEnginesMaxNumber><!--xs:required, xs:integer, maximum number of
supported engines--></supportEnginesMaxNumber>
  <mode
opt="face_recognition,video_structure_preview,video_structure_record,image_recog
nition_human,image_recognition_vehicle,safetyHelmetDetection,,personDensityDetec
tion,leavePosition,peopleNumChange,behavior,retention,situationAnalysis,cityMana
gement,face_human_recognition,face_recognition_IES,OCR,face_recognition_IES_or_O
CR,fireEscapeDetection,takingElevatorDetection,peopleCounting">
    <!--required, xs:string, engine working mode. Refer to the table below for
details-->
  </mode>
  <state opt="loading,analyzing,abnormal, connect_boot_abnorma,
conig_boot_abnorma, connect_main_abnorma, conig_main_abnorma, auth_abnorma,
reset_abnorma, DSP_status_abnorma, lib_ver_err_abnorma">
```

```

    <!--required, read-only, xs:string-->
</state>
<SingelModeCapList><!--number of channels supported by single engine-->
    <SingelModeCap>
        <Mode><!--required, xs:string--></Mode>
        <supportChannelNumber><!--required, xs:integer--></supportChannelNumber>
        <supportChannelNumberAutoSwitch>
            <!--optional, xs:integer, supported number of channels to be linked
with the engine (auto-switch mode); if the device does not support auto-switch,
this node will not be returned-->
            </supportChannelNumberAutoSwitch>
        </SingelModeCap>
    </SingelModeCapList>
    <isSupportCheckUtilization>
        <!--required, xs:boolean, whether to support displaying engine usage, this
node will not be returned if displaying usage is not supported-->
        </isSupportCheckUtilization>
    <isSupportCheckChannelsInfo>
        <!--required, xs:boolean, whether to support displaying channel information
linked to the engine, this node will not be returned if displaying channel No.
linked to the engine is not supported-->
        </isSupportCheckChannelsInfo>
    <repairTime min="-1" max="3">
        <!--optional, xs:integer, number of automatic repairs-->
        </repairTime>
    <isNeedReboot><!--optional, xs:boolean, whether needs to reboot the engine-->
</isNeedReboot>
    <engineSwitchPrompt opt="prompt1"><!--opt,wo,xs:string, engine switching
prompt--></engineSwitchPrompt>
    <supportChannelsConfiguration min="" max=""><!--optional, xs:integer,
supported number of channels to be linked with the engine--></
supportChannelsConfiguration>
    <analysisMode opt="realTimeMode,autoSwitchMode"><!--optional, xs:string,
analysis mode: realTimeMode(default), autoSwitchMode--></analysisMode>
    <eventSupportAutoSwitch
opt="cityManagement,fireEscapeDetection,behavior,peopleCounting,safetyHelmetDete
ction"><!--optional, xs:string, events that supports auto-switch. The
"peopleCounting" includes both people counting and regional people counting in
DVRs/NVRs--></eventSupportAutoSwitch>
    <AutoSwitchTime min="" max="" def=""><!--optional, xs:integer, auto-switch
time interval, unit: second--></AutoSwitchTime>
    <isSupportSchedule><!--xs:boolean, whether the device supports setting auto-
switch schedule--></isSupportSchedule>
</AnalysisEnginesCap>

```

### Engine Working Mode

face_recognition	face recognition
video_structure_preview	video structuring (real-time)

video_structure_record	video structuring (recording)
image_recognition_human	image recognition (human body)
image_recognition_vehicle	image recognition (vehicle)
safetyHelmetDetection	safety helmet detection
personDensityDetection	people density detection
leavePosition	absence detection
peopleNumChange	detection of the abnormal number of people
behavior	behavior analysis (including line crossing detection, intrusion detection, region entrance detection, region exit detection, loitering detection, people running detection, people gathering detection, violent motion detection, people falling down detection)
retention	people overstay detection
situationAnalysis	trend analysis
cityManagement	intelligent city management
face_human_recognition	face and body recognition
face_recognition_IES	face recognition + Intelligent Education System Analysis (IES)
OCR	optical character recognition
face_recognition_IES_or_OCR	face recognition + IES or OCR (face recognition + IER first)
fireEscapeDetection	fire escape detection
takingElevatorDetection	elevator detection
workBehavior	work behavior analysis (including the detection of the abnormal number of people, playing mobile phone detection, absence detection, sleep on duty detection, people retention detection)
streetBehavior	street behavior detection (including people running detection, people gathering detection, violent motion detection, people falling down detection)
AIOP	AI open platform
HEOP	Hikvision embedded open platform
peopleCounting	people counting (including both people counting and regional people counting in NVRs/DVRs)

## C.17 XML\_AttendedBaggageParam

XML message about rule parameters of object removal detection

```
<AttendedBaggageParam><!--dep-->
  <durationTime>
    <!--req, xs:integer, duration time, from 5 to 100 seconds, default value: 5
seconds-->
  </durationTime>
</AttendedBaggageParam>
```

## C.18 XML\_Cap\_RegionEntrance

XML message about configuration capability of region entrance detection by channel

```
<RegionEntrance version="2.0" xmlns="http://www.std-cgi.org/ver20/XMLSchema">
  <id><!--required, xs: string, channel No.--></id>
  <enabled opt="true,false">
    <!--required, xs: boolean, whether to enable region entrance detection-->
  </enabled>
  <normalizedScreenSize><!--required, read-only, normalized coordinates-->
    <normalizedScreenWidth>
      <!--required, read-only, xs: integer-->
    </normalizedScreenWidth>
    <normalizedScreenHeight>
      <!--required, reas-only, xs: integer-->
    </normalizedScreenHeight>
  </normalizedScreenSize>
  <RegionEntranceRegionList size="10"><!--required, detection region list. If
size is not returned, the default number of nodes is 4-->
    <RegionEntranceRegion>
      <id><!--required, xs: string, detection region No.--></id>
      <sensitivityLevel min="1" max="100">
        <!--required, xs: integer, sensitivity, which is between 1 and 100-->
      </sensitivityLevel>
      <RegionCoordinatesList size="4"><!--required, coordinates list of
detection region-->
        <RegionCoordinates><!--required, coordinates of detection region-->
          <positionX>
            <!--required, xs: integer; x-coordinate-->
          </positionX>
          <positionY>
            <!--required, xs: integer; y-coordinate-->
          </positionY>
        </RegionCoordinates>
      </RegionCoordinatesList>
      <detectionTarget opt="all,human,vehicle"><!--optional,
xs:string,"all,human,vehicle,others"--></detectionTarget>
      <alarmConfidence opt="low,mediumLow,mediumHigh,high" def="low"><!--
```

```

optional, xs:string, confidence of alarm notification--></alarmConfidence>
  <recordConfidence opt="low,mediumLow,mediumHigh,high" def="low">!--
optional, xs:string, confidence of recording video--></recordConfidence>
</RegionEntranceRegion>
</RegionEntranceRegionList>
<mutexAbility opt="PDC,videoFrameRate50,videoFrameRate60,19-1920*1080@50fps,
20-1920*1080@60fps,recordPlan"/>
  <!--optional, ro, xs: string, mutex capability, it indicates that which
function is mutually exclusive with the region entrance detection function:
"PDC,videoFrameRate50,videoFrameRate60,19-1920*1080@50fps,
20-1920*1080@60fps,scheduled recording"-->
  <isSupportMultiScene>
    <!--optional, xs: boolean, whether supports multiple scenes detection. This
function is supported by speed domes-->
  </isSupportMultiScene>
  <isSupportHumanMisinfoFilter opt="true">
    <!--optional, xs: boolean, whether supports preventing false human body
alarm, if supports, return true, if not supports, this node will not be
returned-->
  </isSupportHumanMisinfoFilter>
  <isSupportVehicleMisinfoFilter opt="true">
    <!--optional, xs: boolean, whether supports preventing false vehicle
alarm, if supports, return true, if not supports, this node will not be
returned-->
  </isSupportVehicleMisinfoFilter>
  <isSupportTargetMultiSelect>
    <!--optional, xs: boolean, whether supports selecting multiple detection
targets: "true"-support, no return-not support. If this node is "true", it
indicates that <detectedTarget> can be set to multiple values; otherwise, it
indicates that <detectedTarget> can only be set to one value-->
  </isSupportTargetMultiSelect>
  <isSupportAllDayUpload opt="true,false">
    <!--optional, xs: boolean, whether supports uploading all-day event, true-
support, false-not support-->
  </isSupportAllDayUpload>
</RegionEntrance>

```

## C.19 XML\_Cap\_RegionExiting

XML message about configuration capability of region exiting detection by channel

```

<RegionExiting version="2.0" xmlns="http://www.std-cgi.org/ver20/XMLSchema">
  <id>!--required, xs: string, channel No.--></id>
  <enabled opt="true,false">
    <!-- required, xs: boolean, whether to enable region exiting detection-->
  </enabled>
  <normalizedScreenSize>!--required, read-only, normalized coordinates-->
    <normalizedScreenWidth>
      <!--required, read-only, xs: integer-->
    </normalizedScreenWidth>

```

```
<normalizedScreenHeight>
  <!--required, reas-only, xs: integer-->
</normalizedScreenHeight>
</normalizedScreenSize>
<RegionExitingRegionList size="4"><!--required, detection region list-->
  <RegionExitingRegion>
    <id><!--required, xs: string, detection region No.--></id>
    <sensitivityLevel min="1" max="100">
      <!--required, xs: integer, sensitivity, which is between 1 and 100-->
    </sensitivityLevel>
    <RegionCoordinatesList size="10"><!--required, coordinates list of
detection region. If size is not returned, the default number of nodes is 4-->
      <RegionCoordinates><!--required, coordinates of detection region-->
        <positionX>
          <!--required, xs: integer; x-coordinate-->
        </positionX>
        <positionY>
          <!--required, xs: integer; y-coordinate-->
        </positionY>
      </RegionCoordinates>
    </RegionCoordinatesList>
    <detectionTarget opt="all,human,vehicle"><!--optional,
xs:string,"all,human,vehicle,others"--></detectionTarget>
    <alarmConfidence opt="low,mediumLow,mediumHigh,high" def="low"><!--
optional, xs:string, confidence of alarm notification--></alarmConfidence>
    <recordConfidence opt="low,mediumLow,mediumHigh,high" def="low"><!--
optional, xs:string, confidence of recording video--></recordConfidence>
  </RegionExitingRegion>
</RegionExitingRegionList>
<mutexAbility opt="PDC,videoFrameRate50,videoFrameRate60,19-1920*1080@50fps,
20-1920*1080@60fps,recordPlan"/>
  <!--optional, ro, xs: string, mutex capability, it indicates that which
function is mutually exclusive with the region exiting detection:
"PDC,videoFrameRate50,videoFrameRate60,19-1920*1080@50fps,
20-1920*1080@60fps,scheduled recording"-->
  <isSupportMultiScene>
    <!--optional, xs: boolean, whether supports multiple scenes. This function
is supported by speed domes-->
  </isSupportMultiScene>
  <isSupportHumanMisinfoFilter opt="true">
    <!--optional, xs: boolean, whether supports preventing false human body
alarm, if supports, return true, if not supports, this node will not be
returned-->
  </isSupportHumanMisinfoFilter>
  <isSupportVehicleMisinfoFilter opt="true">
    <!--optional, xs: boolean, whether supports preventing false vehicle alarm,
if supports, return true, if not supports, this node will not be returned-->
  </isSupportVehicleMisinfoFilter>
  <isSupportTargetMultiSelect>
    <!--optional, xs: boolean, whether supports selecting multiple detection
targets, true-support, no return-not support-->
  </isSupportTargetMultiSelect>
```

```
<isSupportAllDayUpload>
  <!--optional, xs: boolean, whether supports uploading all-day event, true-
support, false-not support-->
</isSupportAllDayUpload>
</RegionExiting>
```

## C.20 XML\_Cap\_RuleInfo

Capability message of setting behavior analysis in XML format.

```
<RuleInfo version="2.0" xmlns="http://www.isapi.com/ver20/XMLSchema">
  <ruleId opt="1,2,3,4,5,6,7,8"><!--required, xs:string--></ruleId>
  <ruleName min="" max=""><!--required, xs:string--></ruleName>
  <combinedRuleId opt="15,16"><!--required, xs:string--></combinedRuleId>
  <enabled opt="true,false"><!--required, xs:boolean--></enabled>
  <ruleType opt="region,line"><!--required, xs:string--></ruleType>
  <LineDetectionParam>
    <detectionTarget opt="all,human,vehicle, human_vehicle ">
      <!--detection target, node with underline indicates that it supports
multiple selections, e.g., human_vehicle indicates human and vehicle-->
    <directionSensitivity opt="left-right,right-left,any" def="any"><!--
required, xs:string--></directionSensitivity>
    <isSupportHumanMisinfoFilter>
      <!--optional, xs:boolean, whether supports false alarm filter of human:
true-supports, false or no return-not support-->
    </isSupportHumanMisinfoFilter>
    <isSupportVehicleMisinfoFilter>
      <!--optional, xs:boolean, whether supports false alarm filter of vehicle:
true-supports, false or no return-not support-->
    </isSupportVehicleMisinfoFilter>
    <sensitivity min="1" max="100" default="50"><!--required, xs:integer,
sensitivity, from 1 to 100--></sensitivity>
  </LineDetectionParam>
  <FieldDetectionParam>
    <durationTime min="1" max="100" def="5"><!--required,xs:integer--> </
durationTime>
    <sensitivityLevel min="1" max="100"><!--required, xs:integer, sensitivity
from 0 to 100--> </sensitivityLevel>
    <detectionTarget opt="all,human,vehicle, human_vehicle ">
      <!--detection target, node with underline indicates that it supports
multiple selections, e.g., human_vehicle indicates human and vehicle-->
    <isSupportHumanMisinfoFilter>
      <!--optional, xs:boolean, whether supports false alarm filter of human:
true-supports, false or no return-not support-->
    </isSupportHumanMisinfoFilter>
    <isSupportVehicleMisinfoFilter>
      <!--optional, xs:boolean, whether supports false alarm filter of vehicle:
true-supports, false or no return-not support-->
    </isSupportVehicleMisinfoFilter>
  </FieldDetectionParam>
```



```
<RegionEntranceParam>
  <detectionTarget opt="all,human,vehicle, human_vehicle "/>
    <!--detection target, node with underline indicates that it supports
multiple selections, e.g., human_vehicle indicates human and vehicle-->
    <isSupportHumanMisinfoFilter>
      <!--optional, xs:boolean, whether supports false alarm filter of human:
true-supports, false or no return-not support-->
    </isSupportHumanMisinfoFilter>
    <isSupportVehicleMisinfoFilter>
      <!--optional, xs:boolean, whether supports false alarm filter of vehicle:
true-supports, false or no return-not support-->
    </isSupportVehicleMisinfoFilter>
  </RegionEntranceParam>
  <RegionExitingParam>
    <detectionTarget opt="all,human,vehicle, human_vehicle "/><!--detection
target, node with underline indicates that it supports multiple selections,
e.g., human_vehicle indicates human and vehicle-->
    <isSupportHumanMisinfoFilter>
      <!--optional, xs:boolean, whether supports false alarm filter of human,
true-supports, false or no return-not support-->
    </isSupportHumanMisinfoFilter>
    <isSupportVehicleMisinfoFilter>
      <!--optional, xs:boolean, whether supports false alarm filter of vehicle,
true-supports, false or no return-not support-->
    </isSupportVehicleMisinfoFilter>
  </RegionExitingParam>
  <PeopleNumChangeParam>
    <peopleNumThreshold min="" max="" default=""><!--optional, xs:integer--></
peopleNumThreshold>
    <detectMode opt="greaterthan,lessThan,equal,notEqual"><!--optional,
xs:string--></detectMode>
    <noneStateEffectiveEnabled><!--optional,xs:boolean--></
noneStateEffectiveEnabled>
    <durationTime min="" max="" default=""><!--optional, xs:integer--></
durationTime>
  </PeopleNumChangeParam>
  <SpacingChangeParam>
    <spacingThreshold min="" max="" default=""><!--optional, xs: integer--></
spacingThreshold>
    <detectMode><!--optional,xs:string,"greaterthan,lessThan"--></detectMode>
    <durationTime min="" max="" default=""><!--optional, xs:integer--></
durationTime>
  </SpacingChangeParam>
  <ViolentMotionParam>
    <sensitivity min="" max="" default=""> <!--optional, xs:integer--> </
sensitivity>
  </ViolentMotionParam>
  <LeavePositionParam>
    <leaveDelay min="" max="" default=""><!--optional, xs:integer--></
leaveDelay>
    <onPosition min="" max="" default=""><!--optional, xs:integer--></
onPosition>
```

```

</LeavePositionParam>
<FailDownParam>
  <durationTime min="" max="" default=""><!--optional, xs:integer--></
durationTime>
  <sensitivity min="" max="" def=""><!--optional, xs:integer, sensitivity--></
sensitivity>
  <heightThreshold min="" max="" default=""><!--optional, xs:integer--></
heightThreshold>
  <stackTarget opt="true,false"><!--optional, xs:boolean, whether to overlay
alarm target frame--></stackTarget>
  <stackRule opt="true,false"><!--optional, xs:boolean, whether to overlay
alarm rule frame--></stackRule>
  <interval min="4" max="600" def="1">
    <!--optional, int, ro, alarm interval, value range: [4,600]-->1
  </interval>
</FailDownParam>
<RunningParam><!--running detection-->
  <speed min="" max="" default=""><!--optional, xs:integer--></speed>
  <durationTime min="" max="" def=""><!--optional, xs:integer, duration time--
></durationTime>
  <sensitivity min="" max="" def=""><!--optional, xs:integer, sensitivity--></
sensitivity>
  <stackTarget opt="true,false"><!--optional, xs:boolean, whether to overlay
alarm target frame--></stackTarget>
  <stackRule opt="true,false"><!--optional, xs:boolean, whether to overlay
alarm rule frame--></stackRule>
  <mode opt="single,many"><!--optional, xs:string, running mode, "single"-
single person running, "many"-multiple persons running--></mode>
  <peopleNum min="1" max="50" def="1">
    <!--optional, int, ro, the number of running people, value range: [1,50]--
>0
  </peopleNum>
</RunningParam>
<RetentionParam><!--dependent-->
  <durationTime min="" max="" default="">
    <!--required, xs:integer, duration time, from 60 seconds to 3600 seconds,
default value: 1800 seconds-->
  </RetentionParam>
<ParkingParam>
  <durationTime min="5" max="100"><!-- required,xs:integer--> </durationTime>
</ParkingParam>
<eventType
opt="none,lineDetection,fieldDetection,regionEntrance,regionExiting,parking,peop
leNumChange,spacingChange,violentMotion,leavePosition,failDown,running,retention
Param,playCellphone">
  <!--required, xs:string-->
</eventType>
<SizeFilter>
  <enabled opt="true,false"><!--required, xs:boolean--></enabled>
  <mode opt="pixels,actualSize"><!--required, xs:string, filter mode: "pixels"-
filter according to pixel, "actualSize"-filter according to actual size--></
mode>

```

```

    <ObjectSizeList size="4">
        <ObjectSize><!--only return the default value of supported detection
type-->
            <detectionTarget><!--optional,xs:string, detection target, "all"-all,
"human"-human, "vehicle"-vehicle, "human_vehicle"-human and vehicle--></
detectionTarget>
                <MaxObjectSize><!--maximum size:float-->
                    <positionX><!--required, xs:integer;coordinate--></positionX>
                    <positionY><!--required, xs:integer;coordinate--></positionY>
                    <width><!--required, xs:integer, default width value: when
detectionTarget values "all", it is 500, "human"-200, "vehicle"-500,
"human_vehicle"-500, unit: cm--></width>
                    <height><!--required, xs:integer, default height value: when
detectionTarget values "all", it is 100, "human"-300, "vehicle"-100,
"human_vehicle"-100, unit: cm --></height>
                </MaxObjectSize>
                <MinObjectSize><!--minimum size:float-->
                    <positionX><!--required, xs:integer;coordinate--></positionX>
                    <positionY><!--required, xs:integer;coordinate--></positionY>
                    <width><!--required, xs:integer, default width value: when
detectionTarget values "all", it is 60, "human"-60, "vehicle"-180,
"human_vehicle"-60, unit: cm--></width>
                    <height><!--required, xs:integer, default height value: when
detectionTarget values "all", it is 10, "human"-10, "vehicle"-140,
"human_vehicle"-10, unit: cm--></height>
                </MinObjectSize>
            </ObjectSize>
        </ObjectSizeList>
    </SizeFilter>
    <RuleRegion>
        <RegionCoordinatesList size="4">
            <RegionCoordinates>
                <positionX min="" max=""><!--required, xs:integer--></positionX>
                <positionY min="" max=""><!--required, xs:integer--></positionY>
            </RegionCoordinates>
        </RegionCoordinatesList>
    </RuleRegion>
    <backgroundSuppression opt="open,close,selfAdapt"><!--optional, xs:string,
background suppression--></backgroundSuppression>
    <PlayCellphoneParam><!--optional, playing mobile phone-->
        <durationTime min="1" max="3600" def=""><!--optional, xs:integer, duration
time--></durationTime>
        <stackTarget opt="true,false"><!--optional, xs:boolean, whether to overlay
alarm target frame--></stackTarget>
        <stackRule opt="true,false"><!--optional, xs:boolean, whether to overlay
alarm rule frame--></stackRule>
    </PlayCellphoneParam>
    <streetBehavior opt="running,group,violentMotion,failDown">
        <!--optional, xs:string, street behavior capability: "running"-fast moving,
"group"-people gathering, "violentMotion"-violent motion, "failDown"-failing
down-->
    </streetBehavior>

```

```
<workBehavior opt="playCellphone,peopleNumChange,leavePosition,retention">
  <!--optional, xs:string, working behavior capability: "playCellphone"-
playing mobile phone, "peopleNumChange"-number of persons changed,
"leavePosition"-absent, "retention"-loitering-->
</workBehavior>
<isSupportMultiRules><!--optional, xs:boolean, whether it supports multiple
rules for one event; by default, only one rule can be configured for one event--
></isSupportMultiRules>
<isSupportSchedulesEventType><!--optional, xs:boolean, whether to configure
arming schedule by event type; by default, it is configured by rule ID--></
isSupportSchedulesEventType>
<isSupportTriggerEventType><!--optional, xs:boolean, whether to configure
alarm linkage by event type; by default, it is configured by rule ID--></
isSupportTriggerEventType>
<analysisMode opt="realTimeMode,autoSwitchMode"><!--optional, xs:string,
analysis mode, the default mode is "realTimeMode"--></analysisMode>
</RuleInfo>
```

### Remarks

When the **mode** values "actualSize", the input value of **positionX** and **positionY** is 0.

## C.21 XML\_Cap\_Schedule

Schedule capability message in XML format

```
<Schedule version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id><!--req, xs:string, ID--></id>
  <eventType
opt="IO,VMD,videoloss,PIR,linedetection,fieldddetection,audioexception,facedetect
ion,regionEntrance,regionExiting,loitering,group,rapidMove,parking,unattendedBag
gage,attendedBaggage,storageDetection,shipsDetection,HUMANATTRIBUTE,humanAttribu
te,faceContrast,faceSnap, mixedTargetDetection">
    <!--opt, xs:string, event type-->
  </eventType>
  <inputIOPortID><!--ro, dep, xs:string, alarm input ID--></inputIOPortID>
  <outputIOPortID><!--ro, dep, xs:string, alarm output ID--></inputIOPortID>
  <videoInputChannelID><!--ro, dep, xs:string, video input channel ID--></
videoInputChannelID>
  <TimeBlockList size="8"><!--req-->
    <TimeBlock>
      <dayOfWeek opt="1,2,3,4,5,6,7">
        <!--opt, xs:integer, days of the week in ISO8601 time format, 1-Monday,
2-Tuesday, 3-Wednesday, 4-Thursday, 5-Friday, 6-Saturday, 7-Sunday-->
      </dayOfWeek>
      <TimeRange><!--req, time periods in arming schedule-->
        <beginTime><!--req, xs:time, start time of time period, in ISO8601 time
format--></beginTime>
        <endTime><!--req, xs:time, end time of time period, in ISO8601 time
format--></endTime>
      </TimeRange>
```

```

    </TimeBlock>
  </TimeBlockList>
  <HolidayBlockList><!--opt, holiday arming period-->
    <TimeBlock>
      <TimeRange><!--req, time period-->
        <beginTime><!--req, xs:time, start time, in ISO8601 time format--></
beginTime>
        <endTime><!--req, xs:time, end time, in ISO8601 time format--></
endTime>
      </TimeRange>
    </TimeBlock>
  </HolidayBlockList>
</Schedule>

```

## C.22 XML\_ChannelEventCap

XML message about event capability of a single channel

```

<ChannelEventCap version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <eventType opt="">
    <!--required, xs:string, event types supported by the channel; see remarks
for details-->
  </eventType>
  <shieldEventType
opt="behavior,faceSnap,humanRecognition,faceCapture,targetCapture">
    <!--required, xs:string, event types that support area shield-->
  </shieldEventType>
  <channelID><!--optional, xs:integer, current device channel No.--></channelID>
  <id><!--required, xs:integer--></id>
  <passThoughEventType
opt="faceSnapModeling,cityManagement,strudentsStoodUp,dredgerDetection,reverseEn
trance,teacherBehaviorDetect,personQueueDetection">
    <!--optional, xs:string, event types can be transmitted-->
  </passThoughEventType>
</ChannelEventCap>

```

### Remarks

The values of node **<eventType>** are shown in the table below.

eventType	Description
VMD (motionDetection)	Motion detection
cityManagement	Intelligent city management
framesPeopleCounting	Regional people counting
Shelteralarm (tamperDetection)	Video tampering alarm
videoLoss	Video loss


eventType	Description
ROI	Region of interest
facetedetection (faceDetection)	Face detection
intelliTrace	Auto-track
fielddetection (fieldDetection)	Intrusion detection
defocus (defocusDetection)	Defocus detection
audioexception (audioDetection)	Audio exception detection
scenechangedetection (sceneChangeDetection)	Sudden scene change detection
linedetection (lineDetection)	Line-crossing detection
regionEntrance	Region entrance detection
regionExiting	Region exiting detection
loitering	Loitering detection
group	People gathering detection
rapidMove	Fast moving detection
parking	Parking detection
unattendedBaggage	Unattended baggage
attendedBaggage	Object removal detection
peopleDetection	Human detection
storageDetection	HDD health detection
behavior	Behavior analysis
faceCapture(faceSnap)	Face capture
ANPR(vehicleDetection)	Vehicle detection
fireDetection	Fire detection
shipsDetection	Ship detection
PIR	PIR alarm
targetCapture (humanRecognition)	Target capture
alarmResult (faceContrast)	Face comparison
framesPeopleCounting	People counting in single frame

eventType	Description
hide	Tampering detection
audioLoss	Audio loss
cameraAngleAnomaly	Image exception
violentMotion	Violent motion
trial	Tailing
humanEnter	Human entrance
operateOverTime	Operation timed out
stickUp	Sticking scrip
installScanner	Installing scanner
faceDetect	ATM face detection
temperature (temperatureDetection)	Temperature difference detection
PeopleCounting	People counting
personQueueDetection	People queuing-up detection
heatmap	Heat map
mixedTargetDetection	Multi-target-type detection
MTD_face	Supported face attributes of multi-target-type detection (MTD)
MTD_human	Supported human attributes of multi-target-type detection (MTD)
MTD_vehicle	Supported motor vehicle attributes of multi-target-type detection (MTD)
MTD_nonMotor	Supported non-motor vehicle attributes of multi-target-type detection (MTD)
faceSnapModeling	Face capture modeling
HVTVehicleDetection	Mixed-traffic detection
PictureCaptureComparision	N:1 face comparison
IO	Sensor alarm
smokeDetection	Smoke detection
smokeAndFireDetection	Smoke and fire detection

eventType	Description
diskfull	HDD is full
diskerror	HDD error
nicbroken	Network disconnected
ipconflict	IP address conflicted
illaccess	Illegal access
personDensityDetection	People density detection
vehicleControl	Upload alarm of vehicle in allowlist and blocklist
vehicleRcogResult	Vehicle secondary recognition
cardMatch	Authentication
overSpeed	Overspeed alarm
highTempAlarm	High temperature alarm
abnormalAcceleration	Abnormal accelerated speed alarm
failDown	People falling down
leavePosition	Absence detection
peopleNumChange	The number of people changed
retention	Overstay detection
running	Running
thermometry	Temperature measurement
heatmapPDC	Heat map people counting
heatmapDuration	Heat map people staying duration
intersectionAnalysis	Intersection analysis
AID_abandonedObject	Thrown object in traffic events
AID_pedestrian	Pedestrian detection in traffic events
AID_congestion	Congestion detection in traffic events
AID_roadBlock	Roadblock detection in traffic events
AID_construction	Construction detection in traffic events
AID_trafficAccident	Traffic accident detection in traffic events
AID_fogDetection	Fog detection in traffic events



eventType	Description
TFS_illegalParking	Illegal parking of enforcement events
TFS_wrongDirection	Wrong-way driving of enforcement events
TFS_crossLane	Driving on lane line of enforcement events
TFS_laneChange	Lane change of enforcement events
TFS_vehicleExist	Vehicle existing of enforcement events
TFS_turnRound	Turning round of enforcement events
TFS_parallelParking	Parallel parking of enforcement events
TPS	Traffic data collection configuration
luma	Brightness exception
chroma	Color cast detection
snow	Snow noise
streak	Stripe noise
freeze	Image freeze
sigLose	Signal loss
clarity	Clarity exception
jitter	Image flutter
block	Video tampering
flowers	Blurred screen detection
noise	Image noise
ghost	Abnormal light spot detection
purple	Image purple edge detection
ICR	ICR exception detection
protectiveFilm	Protective film unremoved
certificateRevocation	Certificate is expired
dataPreAlarm	Traffic pre-alarm
vibrationDetection	Vibration detection
radarFieldDetection	Radar intrusion detection
radarLineDetection	Radar line crossing detection

eventType	Description
personArmingTrack	Person arming
manualPersonArmingTrack	Manual person arming
fireEscapeDetection	Fire engine access detection
takingElevatorDetection	Elevator detection
temperatureIntervalMeasurement	Interval temperature measurement
containerDetection	Container detection  <b>Note</b> If this node is not returned, you can refer to the value returned by URI (/ISAPI/Traffic/ContentMgmt/InputProxy/channels/<ID>/ocrScene/capabilities) to find whether the device supports container detection.
thermalVehicleDetection	Thermal vehicle detection
wasteGasDetection	Waste gas detection

### Note

The value in the bracket is the old version, and considering the compatibility, both new and old value will be returned.

## C.23 XML\_ChannelEventCapList

Message about event capability of all channels in XML format

```
<ChannelEventCapList version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <ChannelEventCap>
    <eventType opt="">
      <!--required, xs:string, event types supported by the channel; see
remarks for details-->
    </eventType>
    <shieldEventType
opt="behavior,faceSnap,humanRecognition,faceCapture,targetCapture">
      <!--required, xs:string, event types that support area shield-->
    </shieldEventType>
    <channelID><!--optional, xs:integer, current device channel No.--></
channelID>
    <id><!--required, xs:integer--></id>
    <passThoughEventType
opt="faceSnapModeling,cityManagement,strudentsStoodUp,dredgerDetection,reverseEn
trance,teacherBehaviorDetect,personQueueDetection">
```

```
    <!--optional, xs:string, event types can be transmitted-->
    </passThroughEventType>
  </ChannelEventCap>
</ChannelEventCapList>
```

### C.24 XML\_Desc\_EventAbility

Input description message for getting event capability.

```
<?xml version="1.0" encoding="utf-8"?>
<!--req, input description message for getting event capability.-->
<EventAbility version="2.0">
  <channelNO><!--req, channel No.--></channelNO>
</EventAbility>
```

### C.25 XML\_Desc\_VcaChanAbility

Input description message for getting intelligent device capability.

```
<?xml version="1.0" encoding="utf-8"?>
<!--req, description of input parameter pInBuf for getting intelligent device
capability-->
<VcaChanAbility version="2.0">
  <channelNO><!--req, channel No.--></channelNO>
</VcaChanAbility>
```

### C.26 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 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>
<UHFRFIDReader>
  <!--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>
  <isSupportHDDImpact><!--optional, xs:boolean, whether it supports HDD impact detection--></isSupportHDDImpact>
  <isSupportHDBadBlock><!--optional, xs:boolean, whether it supports HDD bad sector detection--></isSupportHDBadBlock>
  <isSupportSevereHDDFailure><!--optional, xs:boolean, whether it supports HDD severe fault detection--></isSupportSevereHDDFailure>
</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,riseTemperature,flame"><!--optional, xs:boolean, whether
it supports temperature alarm: "highTemperature" (high temperature alarm),
"riseTemperature" (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 dta--></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>
```

## C.27 XML\_EventAbility

EventAbility capability message in XML format

```
<EventAbility version="2.0">
  <channelNO><!--req, channel No.--></channelNO>
```

```
<ExceptionAlarm>
  <exceptionType
opt="diskFull,diskError,nicBroken,ipConflict,illAccess,videoMismatch,badVideo,recordingFailure,raid,resolutionMismatch,sparseException,POEPoweException"/>
  <alarmHandleType
opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp,SMS"/>
  <!--req, exception function linkage detailing, if resolved to the
capability below, it will be processed as exception alarm shown below; if not,
it will be processed according to alarmHandleType node-->
  <alarmRelateAct opt="ftp,email"/>
  <!--req, alarm linkage action: "ftp"- upload captured picture to FTP,
"email"-upload captured JPEG picture to email-->
  <DetailedExceptionAlarm>
    <DiskFull><!--req, HDD full-->
      <alarmHandleType
opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp"/>
    </DiskFull>
    <DiskError><!--req, HDD error-->
      <alarmHandleType
opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp"/>
    </DiskError>
    <NicBroken><!--req, network cable broken-->
      <alarmHandleType
opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp"/>
    </NicBroken>
    <IPConflict><!--req, IP conflict-->
      <alarmHandleType
opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp"/>
    </IPConflict>
    <IllAccess><!--req, illegal access-->
      <alarmHandleType
opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp"/>
    </IllAccess>
    <BadVideo><!--req, video loss-->
      <alarmHandleType
opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp"/>
    </BadVideo>
    <VideoMismatch><!--req, video not matched-->
      <alarmHandleType
opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp"/>
    </VideoMismatch>
    <RecordingFailure><!--req, video exception-->
      <alarmHandleType
opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp"/>
    </RecordingFailure>
    <Raid><!--req, Array exception-->
      <alarmHandleType
opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp"/>
    </Raid>
    <ResolutionMismatch><!--req, resolution not matched-->
      <alarmHandleType
opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp"/>
```



```

        </ResolutionMismatch>
        <SpareException><!--req, hot spare exception-->
            <alarmHandleType
opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp"/>
        </SpareException>
        <TemperatureException><!--req, temperature exception-->
            <alarmHandleType
opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp"/>
        </TemperatureException>
        <SubsystemException><!--req, sub system exception-->
            <alarmHandleType
opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp"/>
        </SubsystemException>
        <FanException><!--req, fan exception-->
            <alarmHandleType
opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp"/>
        </FanException>
    </DetailedExceptionAlarm>
</ExceptionAlarm>
<AlarmIn>
    <alarmTime>8</alarmTime>
    <!--req, number of arming interval -->
    <alarmHandleType
opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp,uploadcloud"/
>
    <!--req, handle -->
    <notSupportPTZLinkage>true</notSupportPTZLinkage>
    <!--req, dynamic display based on current capability-->
    <RelatePTZ>
        <!--req, dynamic display based on current capability-->
        <presetNo min="" max=""/>
        <!--req, link supported preset No.-->
        <CruiseNo min="" max=""/>
        <!--req, link supported patrol path No.-->
        <ptzTrackNo min="" max=""/>
        <!--req, link supported PTZ pattern No.-->
        <presetDurationTime min="0" max="20"/>
        <!--opt, preset dwell time -->
    </RelatePTZ>
    <inputType opt="switch,signal"/>
    <!--opt, alarm input type: switch- relay, signal- pulse signal-->
    <eventTypeNum min="1" max=""/><!--opt, supported number of composite event
types-->
    <eventType
opt="VMD,shelteralarm,facedetection,faceCapture,fieldddetection,linedetection,reg
ionEntrance,regionExitings,loitering,group,rapidMove,parking,unattendedBaggage,a
ttendedBaggage,ANPR,audioexception"/>
    <!--opt,composite event types, VMD-motion detection, shelteralarm-tampering
alarm, facedetection-face detection, faceCapture-face capture, fieldddetection-
intrusion, linedetection-line crossing, regionEntrance-region entrance,
regionExitings-region exiting, loitering-lotering detection, group-people
gathering, rapidMove-fast moving, parking-parking detection, unattendedBaggage-

```

```
unattended baggage detection, attendedBaggage-object removal detection, ANPR-
license plate recognition, audioexception-audio exception detection-->
    <alarmRelateAct opt="ftp,email"/>
    <!--req, alarm linkage action: ftp- upload captured picture to FTP, email-
upload captured JPEG picture to email-->
    </AlarmIn>
    <AlarmOut>
        <alarmTime>8</alarmTime>
        <!--req, the number of output acyivate time-->
        <pulseDuration opt="5,10,30,60,120,300,600>manual"/>
        <!--req, the alarm output delay, unit: s-->
    </AlarmOut>
    <FaceDetection>
        <!--req, face detection-->
        <detectFaceEnable opt="true,false"/>
        <!--req, enable face detection: true-yes, false-no -->
        <detectFaceSensitive min="0" max="9"/>
        <!--req,sensitivity -->
        <alarmHandleType
opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp,uploadcloud
"/>
        <!--req, handle-->
        <triggerRecord>true</triggerRecord>
        <!--req,enable or disable, true-support, false-not support-->
        <mutexAbility opt="traversingVirtualPlane,fieldDetection"/>
        <!--req mutex ability for line crossing detection and intrusion detection--
>
        <alarmRelateAct opt="ftp,email"/>
        <!--req, alarm linkage action: ftp- upload captured picture to FTP, email-
upload captured JPEG picture to email-->
    </FaceDetection>
    <Track>
        <!--req, intelligent track -->
        <trackEnable opt="true,false"/>
        <!--req, enable the configuration,true-enable, false- disable -->
        <stopTrackWhenFindFace opt="true,false"/>
        <!--req find face, whether stop tracking -->
        <stopTrackThreshold min="1" max="100"/><!--req stop threshold value, the
function is supported when the node exits -->
        <ManualTrace>
            <!--opt,support manual trace, the corresponding structure is
NET_DVR_PTZ_MANUALTRACE-->
            <enabled>true</enabled>
            <!--req, the function is supported when the node exit-->
            <trackType opt="normalTrack,highSpeedRoadTrack,urbanRoadTrack"/>
            <!--req trace type, the function is supported when the node exits, 0-
normally track, 1- highSpeedRoadTrack, 2- urbanRoadTrack-->
            <linkageTrack opt="ipd"/>
            <!--opt,"ipd means link to dome camera tracking"-->
        </ManualTrace>
    </Track>
    <VoiceDetection>
```

```

<!--req, voice exception detection -->
<enable opt="true,false"/>
<!--req, enable voice detection: true- yes,false- no -->
<Abnormal>
  <sensitivityLevel min="1" max="100"/>
  <!--req,sensitivity -->
  <audioMode opt="0,1,2"/>
  <!--req, audio detection mode: 0- enable sensitivity detection, 1- enable
db threshold detection, 2- enable both -->
  <enable opt="true,false"/>
  <!--req,enable audio detection: true=yes,false- no -->
  <threshold min="1" max="100"/>
  <!--req, audio threshold -->
</Abnormal>
<alarmTime>8</alarmTime>
<!--req, number of output activate time interval -->
<alarmHandleType
opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp,uploadcloud"/
>
  <!--req, handle-->
  <audioSteepDrop>
    <!--sound sharp rise or fall-->
    <enable opt="true,false"/><!--req, whether to enable,true- enable, false-
disable-->
    <sensitivityLevel min="1" max="100"/>
    <!--req, sensitivity-->
  </audioSteepDrop>
</VoiceDetection>
<TraversingVirtualPlane>
  <!--req, line crossing detection capability-->
  <enable opt="true,false"/>
  <!--req,false: disable; true: enable -->
<enableDualVca opt="true,false"/>
  <!--req enable supporting smart post search: true- enable, false- disable--
>
  <ALERTLINENum>8</ALERTLINENum>
  <!--req, the maximum number of warning line that traversing supported -->
  <AlertLine>
    <!--req, the node has the number according to the ALERTLINENum -->
    <ID>1</ID>
    <crossDirection opt="0,1,2"/>
    <!--req, cross direction: 0-both way, 1- from left to right, 2- from
right to left -->
    <sensitivityLevel min="0" max="100"/>
    <!--req,sensitivity -->
  </AlertLine>
  <alarmTime>8</alarmTime>
  <!--req, number of arming time interval -->
  <alarmHandleType
opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp,ptztrack,uplo
adcloud,SMS"/>
  <!--req, handle, picture: capture and send by e-mail(added because the lack

```

```
of uploadftp in the early time; if there is no string, judge the UploadFTP node
of devices soft hardware to find whether support capture and then uploaded to
FTP -->
    <supportLogNotCfg>true</supportLogNotCfg><!--req, support log not
configuration, no return means support configuration and log-->
    <holidayTimeSlotNum>8</holidayTimeSlotNum><!--req,the number of holiday
time for traversing plane detection -->
    <mutexAbility opt="faceDetection,19-1920*1080@50fps,
20-1920*1080@60fps,PDC,videoFrameRate50,videoFrameRate60,recordPlan"/>
    <!--req, mutually exclusive capability, face detection, CaptureMode
1920*1080@50fps, CaptureMode 1920*1080@60fps, PDC, videoFrameRate50,
videoFrameRate60, scheduled recording-->
    <detectionTarget opt="all,human,vehicle,others"/>
    <recogRuleType opt="vectorMode,slopeMode"/>
    <!--req, line crossing detection direction: vector or slope-->
    <triggerRecord>true</triggerRecord>
    <!--req, whether to enable triggered recording: true- supported, false- not
supported-->
    <alarmRelateAct opt="ftp,email"/>
    <!--req, alarm linkage action: ftp- upload captured picture to FTP, email-
upload captured JPEG picture to email-->
    <isSupportHumanMisinfoFilter>
        <!--opt, xs:boolean, "support preventing false human detection alarm or
not, if support, return true, if not, no return"-->
    </isSupportHumanMisinfoFilter>
    <isSupportVehicleMisinfoFilter>
        <!--opt, xs:boolean, "support preventing false vehicle detection alarm or
not, if support, return true, if not, no return"-->
    </isSupportVehicleMisinfoFilter>
    <isSupportAllDayUpload><!--opt, xs:boolean, whether to support uploading
all-day events: "true,false"--></isSupportAllDayUpload>
</TraversingVirtualPlane>
<FieldDetection>
    <!--req, intrusion detection-->
    <enable opt="true,false"/>
    <!--req,false: disable; true: enable -->
    <IntrusiongionNum>8</IntrusiongionNum>
    <!--req,the maximum warning line that field detection supported -->
    <Intrusiongion>
        <!--req,INTRUSIONREGIONNum, the node has the corresponding number -->
        <ID>1</ID>
        <regionNum min="3" max="10"/>
        <!--req,the valid point number of each field detection supported -->
        <duration min="1" max="100"/>
        <!--req, time triggered -->
        <sensitivityLevel min="0" max="100"/>
        <!--req, sensitivty -->
        <rate min = "0" max = "100"/>
        <!--req, rate: no alarm target size accounted for the proportion of alarm
area -->
    </Intrusiongion>
    <alarmTime>8</alarmTime>
```

```
<!--req,number of arming time interval -->
<alarmHandleType
opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp,ptztrack,uplo
adcloud,SMS"/>
<!--req, handle, picture: capture and send by e-mail(added because the lack
of uploadftp in the early time; if there is no string, judge the UploadFTP node
of devices soft hardware to find whether support capture and then uploaded to
FTP -->
<holidayTimeSlotNum>8</holidayTimeSlotNum><!--req,the number of holiday
time for traversing plane detection-->
<mutexAbility opt="faceDetection,19-1920*1080@50fps,
20-1920*1080@60fps,PDC,videoFrameRate50,videoFrameRate60,recordPlan"/>
<!--req, mutually exclusive capability, face detection, CaptureMode
1920*1080@50fps, CaptureMode 1920*1080@60fps, PDC, videoFrameRate50,
videoFrameRate60, scheduled recording-->
<detectionTarget opt="all,human,vehicle,others"/>
<notSupportTriggerRecord><!--req, "true"-not support--></
notSupportTriggerRecord>
<!--req, whether to enable triggered recording: true- supported, false- not
supported-->
<alarmRelateAct opt="ftp,email"/>
<!--req, alarm linkage action: ftp- upload captured picture to FTP, email-
upload captured JPEG picture to email-->
<isSupportHumanMisinfoFilter>
<!--opt, xs:boolean,"support preventing false human detection alarm or
not, if support, return true, if not, no return"-->
</isSupportHumanMisinfoFilter>
<isSupportVehicleMisinfoFilter>
<!--opt, xs:boolean,"support preventing false vehicle detection alarm or
not, if support, return true, if not, no return"-->
</isSupportVehicleMisinfoFilter>
<isSupportAllDayUpload><!--opt, xs:boolean, whether to support uploading
all-day events: "true,false"--></isSupportAllDayUpload>
</FieldDetection>
<DefocusDetection>
<!--req,defocus detection -->
<enable opt="true,false"/>
<!--req,false: disable; true: enable -->
<alarmHandleType
opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp,focus"/>
<!--req,handle, picture: capture and send by e-mail(added because the lack
of uploadftp in the early time; if there is no string, judge the UploadFTP node
of devices soft hardware to find whether support capture and then uploaded to
FTP -->
<sensitivityLevel min="1" max="100"/><!--req,sensitivity-->
<alarmRelateAct opt="ftp,email"/>
<!--req, alarm linkage action: ftp- upload captured picture to FTP, email-
upload captured JPEG picture to email-->
</DefocusDetection>
<SceneChangeDetection>
<!--req,scene change detection -->
<enable opt="true,false"/>
```

```
<!--req,false: disable; true: enable -->
<sensitiveLevel min="1" max="100"/>
<!--req,sensitivity -->
<alarmTime>8</alarmTime>
<!--req,number of arming time interval -->
<alarmHandleType
opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp,uploadcloud"/
>
  <!--req,handle -->
  <sceneChangeDetectionRecord>true</sceneChangeDetectionRecord><!--req video
linkage -->
  <alarmRelateAct opt="ftp,email"/>
  <!--req, alarm linkage action: ftp- upload captured picture to FTP, email-
upload captured JPEG picture to email-->
</SceneChangeDetection>
  <AudioLimitAlarm>
  <!--req, sound limit alarm-->
  <enable opt="true,false"/><!--req,false: disable, true: enable-->
  <!--req, alarm threshold, unit: dB-->
  <decibelLimit min="50" max="70"/>
  <alarmHandleType
opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp"/>
  <!--req, process type, picture: capture and email-->
</AudioLimitAlarm>
  <ButtonDownAlarm>
  <enable opt="true,false"/>
  <!--req,false: disable, true: enable-->
  <alarmHandleType
opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp"/>
  <!--req, process type, picture: capture and email-->
</ButtonDownAlarm>
  <VCADetection>
  <!--req, VCA detection-->
  <enable opt="true,false"/>
  <!--req,false: disable, true: enable-->
  <alarmTime>8</alarmTime>
  <!--req, number of arming periods-->
  <alarmHandleType
opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp"/>
  <!--req, process type, picture: capture and email-->
  <preset>true</preset>
  <!--preset, no return if not supported-->
  <cruise>true</cruise>
  <!--patrol point, no return if not supported-->
  <ptzTrack>true</ptzTrack>
  <!--PTZ track, no return if not supported-->
  <holidaySched>true</holidaySched>
  <!--holiday plan, no return if not supported-->
  <alarmRelateAct opt="ftp,email"/>
  <!--req, alarm linkage action: ftp- upload captured picture to FTP, email-
upload captured JPEG picture to email-->
</VCADetection>
```

```

<VehicleDetection>
  <DetectVehicleCalibration><!--req, detect vehicle calibration ability-->
    <widePixels>400</widePixels>
    <highPixels>400</highPixels>
  </DetectVehicleCalibration>
  <TriggerCond>
    <channel min="" max=""/><!--req, channel No.-->
    <detSenceID min="" max=""/><!--req, scene No.-->
    <triggerMode
opt="postIOSpeed,postSingleIO,postRS485,postRS485Radar,postVirtualcoil,epoliceIo
TrafficLights,epoliceRS485,peRS485,videoEpolice,postMPR,viaVtCoil,ipcHVT"/>
    <!--req, triggering mode-->
  </TriggerCond>
  <TriggerCfg>
    <enable opt="disable,enable"/><!--req, enable this triggering mode or
not-->
    <triggerType
opt="postIOSpeed,postSingleIO,postRS485,postRS485Radar,postVirtualcoil,epoliceIo
TrafficLights,epoliceRS485,peRS485,videoEpolice,postMPR,viaVtCoil,ipcHVT"/>
    <!--req, triggering mode-->
    <TriggerParam>
      <PostMPR>
        <enable opt="disable,enable"/>
        <!--req, enable or disable-->
        <laneNum min="" max=""/>
        <!--req, lane number-->
        <sourceType opt="MPR,IO,RS485"/>
        <!--req, signal source-->
        <LaneBoundaryLine>
          <Line>
            <Start>
              <x min="0.001" max="1"/>
              <!--req, X-axis, 0.001~1-->
              <y min="0.001" max="1"/>
              <!--req, Y-axis, 0.001~1-->
            </Start>
            <End>
              <x min="0.001" max="1"/>
              <!--req, X-axis, 0.001~1-->
              <y min="0.001" max="1"/>
              <!--req, Y-axis, 0.001~1-->
            </End>
          </Line>
        </LaneBoundaryLine>
        <PlateRecogParam>
          <defaultCHNLen min="0" max="3"/>
          <!--req, the string length of the chinese abbreviations
of the provience where the device is running-->
          <province
opt="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,
29,30,31,32,33,34,0xff"/>
          <!--req, provience index-->

```

```

        <enable opt="disable,enable"/>
        <!--req, enable the licensee plate recognition of this
area or not-->
        <!--req, recognition type-->
        <RecogMode>
            <plateRecogPos opt="fromFront,fromBack"/>
            <!--req, licensee plate recognition position,
fromFront-Front Plate, fromBack-Rear Plate-->
            <plateRecogType opt="big,little"/>
            <!--req, licensee plate recognition type, big-Big
Plate, little-Small Plate-->
            <vechileColorRecog opt="true,false"/>
            <!--req, vehicle color recognition: 0-Disable
vehicle color recognition when real plate or small plate recognition, 1-Enable
vehicle color recognition-->
            <farmVehicleRecog opt="true,false"/>
            <!--req, farm vehicle recognition-->
            <blurRecog opt="true,false"/>
            <!--req, fuzzy recognition-->
            <locationType opt="frameLocation,sceneLocation"/>
            <!--req, position type, frameLocation-Frame
Positioning, sceneLocation-Scene Positioning-->
            <recogType
opt="frameRecognition,sceneRecognition"/>
            <!--req, recognition type, frameRecognition-Frame
Recognition ,sceneRecognition-Scene Recognition-->
            <recogTime opt="daytime,night"/>
            <!--req, recognition time, daytime-Daytime, night-
Night-->
            <motorCarRecog opt="true,false"/>
            <!--req, motorcycle recognition-->
            <sceneType opt="epolice,gate"/>
            <!--req, scene type, EPolice-E-police/Multiframe,
gate-Monitoring Point-->
            <microPlateRecog opt="true,false"/>
            <!--req, smaller license plate recognition-->
            <region opt="Res,EU,ER,EUAndCIS"/>
            <!--req, area index: 0-Reserved, 1-Europe(EU), 2-
Russia(ER), 3-Europe&Russia(EU&CIS)-->
            <platePixelWidthMin min="" max=""/>
            <!--req, the pixel width of license plate,
corresponding to the wPlatePixelWidthMin in struct NET_ITC_PLATE_RECOG_PARAM-->
            <platePixelWidthMax min="" max=""/>
            <!--req,the pixel width of license plate,
corresponding to the wPlatePixelWidthMax in struct NET_ITC_PLATE_RECOG_PARAM-->
        </RecogMode>
        <vehicleLogoRecog opt="disable,enable"/>
        <!--req, vehicle logo recognition-->
    </PlateRecogParam>
    <maxLaneLineNum min="0" max="6"/>
    <!--req, the max. lane line number supported-->
    <LaneParam>

```



```

<laneNO min="" max=""/>
<!--req, related lane No.-->
<TssParamInfo>
  <IO>
    <no min="" max=""/>
    <triggerType opt="fallEdge, riseEdge"/>
  </IO>
  <RS485>
    <relateChan min="" max=""/>
  </RS485>
</TssParamInfo>
<carDriveDirection opt="unknown, uptodown, downtoup"/>
<!--vehicle driving direction, unknown-Unknown, uptodown-
Down Direction, downtoup-Up Direction-->
<LaneLine>
  <Line>
    <Start>
      <x min="0.001" max="1"/>
      <!--req, X-axis, 0.001~1-->
      <y min="0.001" max="1"/>
      <!--req, Y-axis, 0.001~1-->
    </Start>
    <End>
      <x min="0.001" max="1"/>
      <!--req, X-axis, 0.001~1-->
      <y min="0.001" max="1"/>
      <!--req, Y-axis, 0.001~1-->
    </End>
    <lineType
opt="unknown, ordinaryLanes, busLanes, fastLane, slowLane, motorcycleLane, nonmotorVeh
icleLane, reversedLane, banTrucksLane, mixLane"/>
      <!--req, lane line type, 0-Unknown, 1-The white
solid line between lanes, 2-Lane Stop Line, 3-Single Yellow Line, 4-Double
Yellow Line, 5-Guardrails on the Lane Line, 6-The lane line the vehicle cannot
cross, 7-Dotted Line-->
    </lineType>
  </Line>
</LaneLine>
<PlateRecog>
  <pointNum min="3" max="20"/>
  <!--req, efficient point, great than or equal to 3
points, if three points are in the same line, the area is invalid. If the lines
cross, the area is invalid. Support up to 20 points.-->
  <Pos>
    <x min="0.001" max="1"/>
    <!--req, X-axis, 0.001~1-->
    <y min="0.001" max="1"/>
    <!--req, Y-axis, 0.001~1-->
  </Pos>
</PlateRecog>
</LaneParam>
<autoBuildRecogArea>true</autoBuildRecogArea>
<!--opt, client generates the recognition area

```

```
automatically, SDK interface has no corresponding field information-->
    <brokenNetHttp>true</brokenNetHttp>
    <!--opt, support breakpoint resume or not-->
    <RodeType>
        <!--opt,Road Type-->
        <type opt="entrance,city,custom,alarmInput"/>
        <!--opt, road type: entrance- entrance, city- city
road, custom- custom, alarmInput- alarm input post, publicSecurity-->
        <Custom>
            <delayTime min="0" max="15000"/>
            <delayTimeUnit opt="ms"/>
        </Custom>
    </RodeType>

    <brokenNetHttp>true</brokenNetHttp>
    <!--opt, whether to support ANR for vehicle recognition-->

    <SnapLine>
        <!--opt, capture line-->
        <Line>
            <Start>
                <x min="0.001" max="1"/>
                <!--req, X coordinate, 0.001~1-->
                <y min="0.001" max="1"/>
                <!--req, Y coordinate, 0.001~1-->
            </Start>
            <End>
                <x min="0.001" max="1"/>
                <!--req, X coordinate, 0.001~1-->
                <y min="0.001" max="1"/>
                <!--req, Y coordinate, 0.001~1-->
            </End>
        </Line>
    </SnapLine>
    </PostMPR>
    </TriggerParam>
</TriggerCfg>

<!--req, capture triggering mode supports linkage of arming time
periods and uploading center-->
<GuardCond>
    <channel min="" max=""/>
    <!--req, channel No.-->
    <relateType opt="MPR"/>
    <!--req, relation type-->
</GuardCond>
<GuardCfg>
    <detSenceID min="" max=""/>
    <!--req, scene No.-->
    <alarmSchedDays min="0" max="7"/>
    <!--req, arming days, up to 7 days supported for one week -->
    <alarmSchedTimes min="0" max="8"/>
```

```

    <!--req, arming time periods, up to 8 periods supported for one
day-->
    <AlarmSched>
        <startHour min="0" max="24"/>
        <!--req, arming start time-->
        <startMin min="0" max="60"/>
        <!--req, arming start time-->
        <stopHour min="0" max="24"/>
        <!--req, arming end time-->
        <stopMin min="0" max="60"/>
        <!--req, arming end time-->
    </AlarmSched>
    <alarmHandleType
opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp,focus,PTZ,upl
oadcloud"/>
    <!--req, handling type-->
    <maxAlarmOutChannelNum min="0" max="64"/>
    <!--req, the max. alarm output channel to be triggered supported by
the device-->
    <alarmOutChannelNum min="" max=""/>
    <!--req, the currently set channel No. to be triggered-->
    <alarmOutTriggeredChannel min="" max=""/>
    <!--req, alarm channel to be triggered-->
    <direction>
        <!--opt,xs:string,"both,forward,reverse",triggering direction-->
    </direction>
    </GuardCfg>

    <!--req,OSD test overlay(support overlay of independent and composite
graph)-->
    <OverlapCond>
        <channel min="" max=""/>
        <!--req, channel No.-->
        <configMode opt="terminal,camera"/>
        <!--req, configuration mode, 0-Terminal, 1-Front-end(directly
connect to the front-end device or connect the terminal and front-end device)--
>
        <picModeType opt="smallPicture,bigPicture"/>
        <!--req, picture type, 0-Small Picture(Independent Graph), 1-Large
Picture(Composite Graph)-->
        <relateType opt="MPR,HVT"/>
        <!--req, MPR by default when no node is returned-->
    </OverlapCond>
    <OverlapCfg>
        <enable opt="disable,enable"/>
        <!--req, enable or not, 0-Disable, 1-Enable-->
        <OverlapItemParam>
            <overlapItemNum min="0" max="50"/>
            <!--req, up to 50 characters-->
            <SingleItem>
                <itemType
opt="unknown,place,crossingNo,deviceNo,directionNo,direction,laneNo,lane,capture

```

```
Time,captureTimeWithMS,plateNo,vehicleColor,vehicleType,vehicleBrand,vehicleSpeed,
speedLimitSign,vehicleLength,illegalcode,monitorInfo,illegalActivities,superSpeedRatio,
redStartTime,redStopTime,redBrightTime,securityCode,captureNo,safeBelt,monitorNo,sunVisor,
laneDirection,licensePlateColor,sceneNumber,sceneName,yellowSignCar,dangerousCar,carSubBrand,
vehicleDirection,validity,country,plateType,plateColor,plateCategory"/>
    <!--req, text type, 0-Unknow, 1-Place, 2-Intersection No., 3-Device No., 4-Direction No.,
    5-Direction, 6-Lane No., 7-Lane, 8-Capture Time(without millisecond),9-Capture Time(with millisecond),
    10-License Plate Number, 11-Vehicle Color, 12-Vehicle Type, 13-Vehicle Brand, 14-Vehicle Speed,
    15-Speed Limit Sign, 16-Vehicle Length(1 to 99m),17-Violation Code(traffic violation information is
    more useful than code, e.g. Normal, Low Speed, Overspeed, Opposite Direction, Run the Red Light,
    Occupying Lane, Over Yellow Lane Line),18-Monitoring Site Information, 19-Traffic Violation,
    20-Overspeed Ratio, 21-Red Light Start Time, 22-Red Light End Time, 23-Red Light Time, 24-Security
    Code, 25-Capture No.,26-Seat Belt, 27-Monitoring Site No., 28-Sun Shield, 29-Lane Direction,
    30-License Plate Color, 31-Scene No., 32-Scene Name, 33- Yellow Label Car, 34- Dangerous Goods
    Vehicle, 35- Vehicle Sub Brand, 36- vehicle direction, 38- confidence 40- Country, 41- License
    plate type 42- License plate size, 43- License plate color, 44- License plate additional information-->
    <changeLineNum min="0" max="10"/>
    <!--req, line feeds number after overlay[0 to 10](by default is 0)-->
    <spaceNum min="0" max="255"/>
    <!--req, space number after overlay [0 to255](by default is 0)-->
    <enablePos opt="0,1 "/>
    <!--req, enable coordinate display or not, 0-Disable, 1-Enable-->
    <startPosTop min="0" max="2448"/>
    <!--req, start top coordinate, only valid in picture internal overlay [0 to 2448](by default is 0)-->
    <startPosLeft min="0" max="2448"/>
    <!--req, start left coordinate, only valid in picture internal overlay [0 to 2448](by default is 0)-->
    </SingleItem>
    <linePercent min="0" max="100"/>
    <!--req, percentage of overlay lines(0 to 100),(by default is 100)-->
    <itemsStlye opt="0,1"/>
    <!--req, overlay mode, 0-Horizontal, 1-Vertical (by default is horizontal)-->
    <startPosTop min="0" max="2448"/>
    <!--req, start top coordinate, only valid in picture internal overlay[0 to 2448](by default is 0)-->
    <startPosLeft min="0" max="2448"/>
    <!--req, start left coordinate, only valid in picture internal overlay [0 to 2448](by default is 0)-->
    <charStyle opt="0,1"/>
    <!--req, font type, 0-Song Typeface, 1-Wei Typeface(default)-->
    <charSize min="" max=""/>
```

```

        <!--req, character length-->
        <charInterval min="0" max=""/>
        <!--req, character pitch, [0 to 16], Unit: Pixel(default)-->
        <foreClorRGB opt="B,G,R,White"/>
        <!--req, RGB of foreground color, bit0-7:(B) bit8-15:(G)
bit16-23:(R) (by default 0x00FFFFFF-White)-->
        <backClorRGB opt="B,G,R,Black"/>
        <!--req, RGB of background color, only valid in picture
external overlay, bit0-7:(B) bit8-15:(G) bit16-23:(R) (by default x00000000-
Black-->
        <colorAdapt opt="disable,enable"/>
        <!--req, color self-adaptive or not, 0-No, 1-Yes-->
        <paramFillZeroEnble opt="disable,enable"/>
        <!--req, enable parameter zerofilling, 0-Enable, 1-Disable
(annotation). Speed, speed limit, zerofill when less than 3 digits -->
        <plateLeftCornerEnable opt="disable,enable"/>
        <!--req, enable overlay of license plate window in the upper-
left corner, 0-Disable, 1-Enable-->
        <startSPicPosTop min="0" max="2448"/>
        <!--req, start top coordinate, only valid in picture internal
overlay [0 to 2448](by default is 0)-->
        <startSPicPosLeft min="0" max="2448"/>
        <!--req, start left coordinate, only valid in picture internal
overlay [0~2448](by default is 0)-->
        <osdLocate opt="inside,upper,lower"/>
        <!--req, OSD overlay position, 0-In the picture, 1-Picture top
edge, 2-Picture lower edge (for composite graph, out of the top edge)-->
    </OverlapItemParam>
    <OverlapInfoParam>
        <siteLen min="0" max="128"/>
        <!--req, position description length-->
        <roadNumLen min="0" max="32"/>
        <!--req, intersection No. length-->
        <instrumentNumLen min="0" max="32"/>
        <!--req, device No. length-->
        <directionLen min="0" max="32"/>
        <!--req, direction No. length-->
        <directionDescLen min="0" max="32"/>
        <!--req, direction description length-->
        <laneDesLen min="0" max="32"/>
        <!--req, lane description length-->
        <monitoringSite1Len min="0" max="44"/>
        <!--req, monitoring site 1 information length-->
        <monitoringSite2Len min="0" max="32"/>
        <!--req, monitoring site 2 information length-->
    </OverlapInfoParam>
</OverlapCfg>

<!--req, monitoring site parameter configuration-->
<MonitorCfg>
    <deviceType
opt="camera,coilTriggeringCamera,videoAnalysisCamera,ITC"/>

```

```
    <!--req, device type, 0-Camera for Monitoring, 1-Coil Triggered
Camera, 2-Video Analysis Camera, 3-All-in-one(ITC)-->
    <monitoringSiteIDLen min="0" max="48"/>
    <!--req, monitoring site No. length-->
    <deviceIDLen min="0" max="48"/>
    <!--req, device No. length-->
    <directionNo min="" max=""/>
    <!--req, monitoring direction No.-->
    <monitoringSInfoLen min="0" max="48"/>
    <!--req, monitoring site description length-->
</MonitorCfg>

<!--req,trigger capture manually-->
<ManualSnap>
    <enabled opt="disable,enable"/>
    <!--req, support triggering capture manually or not. If not
support, it will not display.-->
    <osdEnable opt="disable,enable"/>
    <!--req, enable OSD, false-Enable(default), true-Disable-->
</ManualSnap>

<!--req, captured picture parameter configuration-->
<SnapPicInfoCfg>
    <picSize min="64" max="2048"/>
    <!--req, picture size[64k,2048k]-->
    <picQuality min="1" max="100"/>
    <!--req, picture quality [1,100]-->
</SnapPicInfoCfg>
<mutexAbility opt="stream3,hvtVehicleDetection,heatMap"/>
<!--req, mutually exclusive ability:stream3(Stream 3), mixed-traffic
detection -->
</VehicleDetection>
<HVTVehicleDetection>
    <DetectVehicleCalibration>
        <!--req, detect vehicle calibration capability-->
        <widePixels>400</widePixels>
        <highPixels>400</highPixels>
    </DetectVehicleCalibration>
    <TriggerCond>
        <channel min="" max=""/>
        <!--req channel No.-->
        <detSenceID min="" max=""/>
        <!--req detect scene No.-->
        <triggerMode
opt="postIOSpeed,postSingleIO,postRS485,postRS485Radar,postVirtualcoil,epoliceIo
TrafficLights,
        epoliceRS485,peRS485,videoEpolice,postMPR,viaVtCoil,ipcHVT"/>
        <!--req, trigger mode-->
    </TriggerCond>
</TriggerCfg>
    <enable opt="disable,enable"/>
    <!--req, whether to enable the trigger mode-->
```

```
<triggerType
opt="postIOSpeed,postSingleIO,postRS485,postRS485Radar,postVirtualcoil,epoliceIo
TrafficLights,
    epoliceRS485,peRS485,videoEpolice,postMPR,viaVtCoil,ipcHVT"/>
<!--req, trigger mode -->
<TriggerParam>
    <PostIPCHVT>
        <enable opt="disable,enable"/>
        <!--req, enable-->
        <laneNum min="" max=""/>
        <!--req, number of lanes-->
        <LaneBoundaryLine>
            <Line>
                <Start>
                    <x min="0.001" max="1"/>
                    <!--req, X axis, 0.001 to 1-->
                    <y min="0.001" max="1"/>
                    <!--req, Y axis, 0.001 to 1-->
                </Start>
                <End>
                    <x min="0.001" max="1"/>
                    <!--req, X coordinate, 0.001~1-->
                    <y min="0.001" max="1"/>
                    <!--req, Y coordinate, 0.001~1-->
                </End>
            </Line>
        </LaneBoundaryLine>
        <PlateRecogParam>
            <defaultCHNLen min="0" max="3"/>
            <!--req, length of the string for province name shorthand-->
            <province
opt="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,
29,30,31,32,33,34,0xff"/>
            <!--req province index-->
            <enable opt="disable,enable"/>
            <!--req, whether to enable the region sign-->
            <!--detected type-->
            <RecogMode>
                <plateRecogPos opt="fromFront,fromBack"/>
                <!--plate recognition position, fromFront- from front, fromBack-
from back-->
                <plateRecogType opt="big,little"/>
                <!--plate recognition type, big- big plate, little- little plate--
>
                <vechileColorRecog opt="true,false"/>
                <!--vehicle color recognition: 0- disable, cannot be enabled for
recognition of small plate or from back, 1- enable-->
                <farmVehicleRecog opt="true,false"/>
                <!--farming vehicle recognition-->
                <blurRecog opt="true,false"/>
                <!--fuzzy recognition-->
                <locationType opt="frameLocation,sceneLocation"/>
```

```
<!--location type: frameLocation- frame location, sceneLocation-
scene location-->
<recogType opt="frameRecognition,sceneRecognition"/>
<!--recognition type: frameRecognition- frame recognition,
sceneRecognition- scene recognition-->
<recogTime opt="daytime,night"/>
<!--recognition time: daytime- day, night- night-->
<motorCarRecog opt="true,false"/>
<!--motorcycle recognition-->
<sceneType opt="epolice,gate"/>
<!--scene type: EPolice- e-police/multi-frame, gate- checkpoint--
>

<microPlateRecog opt="true,false"/>
<!--micro plate recognition-->
<region opt="Res,EU,ER"/>
<!--req, region index: 0- reserved, 1- Europe, 2- Russia-->
</RecogMode>
<vehicleLogoRecog opt="disable,enable"/>
<!--Logo recognition-->
</PlateRecogParam>
<maxLaneLineNum min="0" max="6"/>
<!--req max. supported number of lane lines-->
<LaneParam>
  <laneNO min="" max=""/>
  <!--req link lane No.-->
  <carDriveDirection opt="unknown,uptodown,downtoup"/>
  <!--vehicle driving direction, unknown- unknown, uptodown-
downward, downtoup- upward-->
  <LaneLine>
    <Line>
      <Start>
        <x min="0.001" max="1"/>
        <!--req X coordinate, 0.001~1-->
        <y min="0.001" max="1"/>
        <!--req Y coordinate, 0.001~1-->
      </Start>
      <End>
        <x min="0.001" max="1"/>
        <!--req X coordinate, 0.001~1-->
        <y min="0.001" max="1"/>
        <!--req Y coordinate, 0.001~1-->
      </End>
    </Line>
  </LaneLine>
  <PlateRecog>
    <pointNum min="3" max="20"/>
    <!--req valid point, >=3, if the three points are on the same
line or the lines crosses, the region is considered invalid, up to 20 points
are supported-->
    <Pos>
      <x min="0.001" max="1"/>
      <!--req X coordinate, 0.001~1-->
```



```
<y min="0.001" max="1"/>
<!--req Y coordinate, 0.001~1-->
</Pos>
</PlateRecog>
</LaneParam>
</PostIPCHVT>
</TriggerParam>
</TriggerCfg>

<!--req, capture trigger mode supports linkage to arming time periods and
uploading to center-->
<GuardCond>
  <channel min="" max=""/>
  <!--req, channel No.-->
  <relateType opt="MPR"/>
  <!--req, linkage type-->
</GuardCond>
<GuardCfg>
  <detSenceID min="" max=""/>
  <!--req, detection scene No.-->
  <alarmSchedDays min="0" max="7"/>
  <!--req, arming days, up to 7 days a week are supported-->
  <alarmSchedTimes min="0" max="8"/>
  <!--req, arming time periods, up to 8 periods a day are supported-->
  <AlarmSched>
    <startHour min="0" max="24"/>
    <!--req, arming start time-->
    <startMin min="0" max="60"/>
    <!--req, arming start time-->
    <stopHour min="0" max="24"/>
    <!--req, arming end time-->
    <stopMin min="0" max="60"/>
    <!--req, arming end time-->
  </AlarmSched>
  <alarmHandleType
opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp,focus,PTZ"/>
  <!--req, processing type-->
  <maxAlarmOutChannelNum min="0" max="64"/>
  <!--req, max number of triggered alarm output channels supported by
device-->
  <alarmOutChannelNum min="" max=""/>
  <!--req, current set alarm triggered channels-->
  <alarmOutTriggeredChannel min="" max=""/>
  <!--req, trigger alarm channel-->
  <direction>
    <!--opt,xs:string,"both,forward,reverse", trigger direction-->
  </direction>
</GuardCfg>

<!--req,OSD character overlay(independent image or composite image
overlay)-->
<OverlapCond>
```

```
<channel min="" max=""/>
<!--req channel No.-->
<configMode opt="terminal,camera"/>
<!--req configuration mode, 0- terminal, 1- front end(direct connection
or terminal connection)-->
<picModeType opt="smallPicture,bigPicture"/>
<!--req picture type: 0- small (independent), 1- big (composite)-->
<relateType opt="MPR,HVT"/>
<!-- default to MPR if the node is not returned-->
</OverlapCond>
<OverlapCfg>
  <enable opt="disable,enable"/>
  <!--req whether to enable, 0- disable, 1-enable-->
  <OverlapItemParam>
    <overlapItemNum min="0" max="50"/>
    <!--req max. 50 characters-->
    <SingleItem>
      <itemType
opt="unknown,place,crossingNo,deviceNo,directionNo,direction,laneNo,lane,capture
Time,captureTimeWithMS,plateNo,vehicleColor,vehicleType,vehicleBrand,vehicleSpee
d,speedLimitSign,vehicleLength,illegalcode,monitorInfo,illegalActivities,superSp
eedRatio,redStartTime,redStopTime,redBrightTime,securityCode,captureNo,safeBelt,
monitorNo,sunVisor,laneDirection,licensePlateColor,sceneNumber,sceneName,vehicle
Direction,validity"/>
      <!--req character type: 0- unknown, 1- location, 2- crossroad No., 3-
device No., 4- direction No., 5- direction, 6- lane No., 7- lane, 8- capture
time(without millisecond), 9- capture time (without millisecond), 10- plate
No., 11- vehicle color, 12- vehicle type, 13- vehicle brand, 14- vehicle speed,
15- speed limit sign, 16- vehicle length, 17- illegal code, 18- camera
information, 19- illegal conduct, 20- overspeed ratio, 21- red light start
time, 22- light off time, 23- red light duration, 24- anti-counterfeiting code,
25- capture code, 26- safety belt, 27- camera No.,28- sun shield, 29- lane
driving direction, 30- plate color, 31- scene No., 32- scene name, 36- vehicle
direction, 38- confidence -->
      <changeLineNum min="0" max="10"/>
      <!--req lines after item overlay [0-10](default to 0)-->
      <spaceNum min="0" max="255"/>
      <!--req spaces after item overlay [0-255](default to 0)-->
      <enablePos opt="0,1 "/>
      <!--req whether to enable coordinate display, 0- no, 1- yes-->
      <startPosTop min="0" max="2448"/>
      <!--req start top coordinate, only valid for internal picture overlay
[0-2448](default to 0)-->
      <startPosLeft min="0" max="2448"/>
      <!--req start left coordinate, only valid for internal picture
overlay [0-2448](default to 0)-->
    </SingleItem>
    <linePercent min="0" max="100"/>
    <!--req overlay line percentage (0-100),(default to 100)-->
    <itemsStlye opt="0,1"/>
    <!--req overlay type: 0- horizontal, 1- vertical(default to
horizontal)-->
```

```
<startPosTop min="0" max="2448"/>
<!--req start top coordinate, only valid for internal picture overlay
[0-2448](default to 0)-->
<startPosLeft min="0" max="2448"/>
<!--req start left coordinate, only valid for internal picture overlay
[0-2448](default to 0)-->
<charStyle opt="0,1"/>
<!--req character type, 0- Song 1- Wei(default)-->
<charSize min="" max=""/>
<!--req character length-->
<charInterval min="0" max=""/>
<!--req character space, [0-16], unit: pixel(default)-->
<foreClorRGB opt="B,G,R,White"/>
<!--req foreground color RGB value bit0-7:(B) bit8-15:(G) bit16-23:(R)
(default: 0x00FFFFFF-white)-->
<backClorRGB opt="B,G,R,Black"/>
<!--req background color RGB value, only valid for external picture
overlay, bit0-7:(B) bit8-15:(G) bit16-23:(R) (default: x00000000-black-->
<colorAdapt opt="disable,enable"/>
<!--req whether the color is self-adaptive 0- no, 1- yes-->
<paramFillZeroEnble opt="disable,enable"/>
<!--req enable parameter zero fill, 0- enable, 1- disable (detailed
notes) speed, speed limit, zero fill to 3 places-->
<plateLeftCornerEnable opt="disable,enable"/>
<!--req enable plate small picture overlay to upper left corner, 0-
disable, 1- enable-->
<startSPicPosTop min="0" max="2448"/>
<!--req start top coordinate, only valid for internal picture overlay
[0-2448](default to 0)-->
<startSPicPosLeft min="0" max="2448"/>
<!--req start left coordinate, only valid for internal picture overlay
[0-2448](default to 0)-->
<osdLocate opt="inside,upper,lower"/>
<!--req OSD overlay position 0- inside the picture, 1- top edge, 2-
bottom edge (top edge is exclusive for composite picture)-->
</OverlapItemParam>
<OverlapInfoParam>
  <siteLen min="0" max="128"/>
  <!--req, location description length-->
  <roadNumLen min="0" max="32"/>
  <!--req, crossroads No. length-->
  <instrumentNumLen min="0" max="32"/>
  <!--req, device No. length-->
  <directionLen min="0" max="32"/>
  <!--req, direction No. length-->
  <directionDescLen min="0" max="32"/>
  <!--req, direction description length-->
  <laneDesLen min="0" max="32"/>
  <!--req, lane description-->
  <monitoringSite1Len min="0" max="44"/>
  <!--req, camera 1 information length-->
  <monitoringSite2Len min="0" max="32"/>
```

```
<!--req, camera 2 information length-->
</OverlapInfoParam>
</OverlapCfg>

<!--req, camera parameter configuration-->
<MonitorCfg>
  <deviceType opt="camera,coilTriggeringCamera,videoAnalysisCamera,ITC"/>
  <!--req, device type, 0- surveillance camera; 1- coil trigger camera; 2-
video analysis camera, 3- ITC-->
<monitoringSiteIDLen min="0" max="48"/><!--req camera No. length-->
  <deviceIDLen min="0" max="48"/>
  <!--req, device No. length-->
  <directionNo min="" max=""/>
  <!--req, surveillance direction No.-->
  <monitoringSInfoLen min="0" max="48"/>
  <!--req, camera information description length-->
</MonitorCfg>

<!--req, manual trigger capture-->
<ManualSnap>
  <enabled opt="disable,enable"/>
  <!--req, whether to support manual trigger capture. It will not be
displayed if not supported-->
  <osdEnable opt="disable,enable"/>
  <!--req, enable OSD false- enable(default), true-disable-->
</ManualSnap>

<!--req, capture picture parameter configuration-->
<SnapPicInfoCfg>
  <picSize min="64" max="2048"/>
  <!--req picture size[64k,2048k]-->
  <picQuality min="1" max="100"/>
  <!--req picture quality[1,100]-->
</SnapPicInfoCfg>
  <mutexAbility opt="stream3,vehicleDetection"/>
  <!--req mutually exclusive capability: stream3(stream 3), mixed vehicle
detection -->
</HVTVehicleDetection>
  <!--req, support current trigger mode by default, device will be indicated if
the node is returned, supporting detailed function, the interface display is
based on capability-->
  <CurVehicleDetection>
    <getCfg>true</getCfg>
    <setCfg>true</setCfg>
  </CurVehicleDetection>
  <AlarmSearch><!--optional, capability of searching for alarm or event
information-->
    <alarmComm opt="json,faceSnap,faceContrast"><!--required, xs:string,
supported alarm type: "json"-alarm transmitted in JSON format, "faceSnap"-face
capture, "faceContrast"-face comparison--></alarmComm>
    <EventList><!--dependent, specific alarm type to be transmitted, this node
is valid only when <alarmComm> is "json"-->
```

```

    <Event>
      <type><!--required, xs:string, event type: "mixedTargetDetection"-multi-
target-type detection--></type>
      <subEventType opt="face,human,vehicle,radar"><!--optional, xs:string,
sub event type: 1-"face", 2-"human" (human body), 3-"vehicle", 4-"radar". The
index starts from 1--></subEventType>
    </Event>
  </EventList>
</AlarmSearch>
</EventAbility>

```

### C.28 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>

```

### C.29 XML\_EventTrigger

Linkage parameter message in XML format

```

<EventTrigger version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id><!--required, xs:string, ID--></id>

```

```
<eventType>
  <!--required, xs:string, see details in the "Remarks" below-->
</eventType>
<eventDescription><!--optional, xs:string--></eventDescription>
<inputIOPortID><!--dependent, xs:string, alarm input ID--></inputIOPortID>
<dynInputIOPortID><!--dependent, xs:string, dynamic alarm input ID--></
dynInputPortID>
<videoInputChannelID>
  <!--dependent, xs:string, video input channel ID, it is valid when
<eventType> is "VMD, videoloss, tamperdetection, regionEntrance, regionExiting,
loitering, group, rapidMove, parking, unattendedBaggage, attendedBaggage"-->
</videoInputChannelID>
<dynVideoInputChannelID><!--dependent, xs:string, dynamic video input channel
ID--></dynVideoInputChannelID>
<intervalBetweenEvents><!--optional, xs:integer, event time interval, unit:
second--></intervalBetweenEvents>
<WLSensorID><!--dependent, xs:string, ID--></WLSensorID>
<EventTriggerNotificationList/><!--optional, alarm/event linkage actions, see
details in the message of XML_EventTriggerNotificationList-->
</EventTrigger>
```

### Remarks

The node **<eventType>** can be the following values: IO, VMD, videoloss, raidfailure, recordingfailure, badvideo, POS, analytics, fanfailure, overheat, tamperdetection, diskfull, diskerror, nicbroken, ipconflict, illaccess, videomismatch, resolutionmismatch, radifailure, PIR, WLSensor, spareException, poePowerException, heatmap, counting, linedetection, fielddetection, regionEntrance, regionExiting, loitering, group,rapidMove, parking, unattendedBaggage, attendedBaggage, HUMANATTRIBUTE, blacklist, whitelist, peopleDetection, allVehicleList, otherVehicleList, vehicledetection, storageDetection, shipsDetection, humanAttribute, faceContrast, blacklistFaceContrast, whitelistFaceContrast, faceSnap, faceLib, personDensityDetection, personQueueDetecton, mixedTargetDetection, HVTVehicleDetection, illegalParking, pedestrian, trafficAccident, construction, roadblock, abandonedObject, parallelParking, parkingState, congestion, intersectionAnalysis, heatMap, thermometry, shipsFlowDetection, dredgerDetection, reverseEntrance, luma, highHDTemperature, lowHDTemperature, hdImpact, hdBadBlock, SevereHDFailure, safetyHelmetDetection, vibrationDetection, HBDLib,TMPA,faceThermometry,noMaskDetection, detectorTemp, detectorSmoke, detectorTamper, smokeFireRecognize, indoorPasswayBlock, detectorShelter, detectorMotion, fireNoRegulation.

### See Also

[XML\\_EventTriggerNotificationList](#)

## C.30 XML\_EventTriggerNotification

Event linkage notification message in XML format

```
<EventTriggerNotification><!--opt-->
  <id><!--required, xs:string, device ID--></id>
  <notificationMethod>
    <!--required, xs:string, linkage actions,
opt="email,IM,IO,syslog,HTTP,FTP,beep,ptz,record,monitorAlarm,center,
LightAudioAlarm,focus,trace,cloud,SMS,whiteLight,audio,whiteLight,faceContrast,s
iren,output"-->
  </notificationMethod>
  <notificationRecurrence>
    <!--optional, xs:string, "beginning,beginningandend,recurring"-->
  </notificationRecurrence>
  <notificationInterval><!--dependent, xs:integer, unit: millisecond--></
notificationInterval>
  <outputIOPortID><!--dependent, xs:string, video output No., it is required
only when notificationMethod is "IO"--></outputIOPortID>
  <dynOutputIOPortID><!--dependent, xs:string, dynamic video output No., it is
required only when notificationMethod is "IO"--></dynOutputIOPortID>
  <videoInputID><!--dependent, xs:string, video input No., it is required only
when notificationMethod is "record"--></videoInputID>
  <dynVideoInputID><!--dependent, xs:string, dynamic video input No., it is
required only when notificationMethod is "record"--></dynVideoInputID>
  <ptzAction><!--dependent, it is required only when notificationMethod is
"ptz"-->
    <ptzChannelID><!--required, xs:string, PTZ channel ID--></ptzChannelID>
    <actionName><!--required, xs:string, PTZ control type: "preset", "pattern",
"patrol"--></actionName>
    <actionNum><!--dependent, xs:integer></actionNum>
  </ptzAction>
  <WhiteLightAction><!--dependent, white light linkage parameters, this node is
valid when notificationMethod is "whiteLight"-->
    <whiteLightDurationTime><!--required, xs:integer, white light flashing
duration, it is between 1 and 60, unit: second--></whiteLightDurationTime>
  </WhiteLightAction>
  <cellphoneNumber><!--dependent, xs:string, min="0" max="11",cellphone number--
></cellphoneNumber-->
</EventTriggerNotification>
```

### C.31 XML\_EventTriggerNotificationList

EventTriggerNotificationList message in XML format

```
<EventTriggerNotificationList version="2.0" xmlns="http://www.isapi.org/ver20/
XMLSchema">
  <EventTriggerNotification/><!--opt, see details in the message of
XML_EventTriggerNotification-->
</EventTriggerNotificationList>
```

#### See Also

[XML\\_EventTriggerNotification](#)

## C.32 XML\_EventTriggersCap

XML message about linkage capabilities of different alarm categories

```
<EventTriggersCap version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <DiskfullTriggerCap><!--optional, xs: EventTriggerCapType--></
DiskfullTriggerCap>
  <DiskerrorTriggerCap><!--optional, xs: EventTriggerCapType--></
DiskerrorTriggerCap>
  <NicbrokenTriggerCap><!--optional, xs: EventTriggerCapType--></
NicbrokenTriggerCap>
  <IpconflictTriggerCap><!--optional, xs: EventTriggerCapType--></
IpconflictTriggerCap>
  <IllaccesTriggerCap><!--optional, xs: EventTriggerCapType--></
IllaccesTriggerCap>
  <BadvideoTriggerCap><!--optional, xs: EventTriggerCapType--></
BadvideoTriggerCap>
  <VideomismatchTriggerCap><!--optional, xs: EventTriggerCapType--></
VideomismatchTriggerCap>
  <IOTriggerCap><!--optional, xs: EventTriggerCapType--></IOTriggerCap>
  <LineDetectTriggerCap><!--optional, xs: EventTriggerCapType--></
LineDetectTriggerCap>
  <RegionEntranceTriggerCap><!--optional, xs: EventTriggerCapType--></
RegionEntranceTriggerCap>
  <RegionExitingTriggerCap><!--optional, xs: EventTriggerCapType--></
RegionExitingTriggerCap>
  <LoiteringTriggerCap><!--optional, xs: EventTriggerCapType--></
LoiteringTriggerCap>
  <GroupDetectionTriggerCap><!--optional, xs: EventTriggerCapType--></
GroupDetectionTriggerCap>
  <RapidMoveTriggerCap><!--optional, xs: EventTriggerCapType--></
RapidMoveTriggerCap>
  <ParkingTriggerCap><!--optional, xs: EventTriggerCapType--></
ParkingTriggerCap>
  <UnattendedBaggageTriggerCap><!--optional, xs: EventTriggerCapType--></
UnattendedBaggageTriggerCap>
  <AttendedBaggageTriggerCap><!--optional, xs: EventTriggerCapType--></
AttendedBaggageTriggerCap>
  <FireDetectionTriggerCap><!--optional, xs: EventTriggerCapType--></
FireDetectionTriggerCap>
  <FireDetectionCap><!--optional, xs: EventTriggerCapType--></FireDetectionCap>
  <StorageDetectionTriggerCap><!--optional, xs: EventTriggerCapType--></
StorageDetectionTriggerCap>
  <ShipsDetectionTriggerCap><!--optional, xs: EventTriggerCapType--></
ShipsDetectionTriggerCap>
  <ThermometryCap><!--optional, xs: EventTriggerCapType--></ThermometryCap>
  <VandalProofTriggerCap><!--optional, xs: EventTriggerCapType--></
VandalProofTriggerCap>
  <BlackListTriggerCap><!--opt, xs: EventTriggerCapType, configuration
capability of blocklist arming linkage--></BlackListTriggerCap>
  <WhiteListTriggerCap><!--opt, xs: EventTriggerCapType, configuration
```



```
capability of allowlist arming linkage--></WhiteListTriggerCap>
  <AllVehicleListTriggerCap><!--optional, xs:EventTriggerCapType, configuration
capability of other list arming linkage--></AllVehicleListTriggerCap>
  <OtherVehicleListTriggerCap><!--optional, xs:EventTriggerCapType--></
OtherVehicleListTriggerCap>
  <PeopleDetectionTriggerCap><!--optional, xs:EventTriggerCapType--></
PeopleDetectionTriggerCap>
  <PIRAlarmCap><!--optional, xs: EventTriggerCapType--></PIRAlarmCap>
  <TamperDetectionTriggerCap><!--optional, xs: EventTriggerCapType--></
TamperDetectionTriggerCap>
  <DefocusDetectionTriggerCap><!--optional, xs: EventTriggerCapType--></
DefocusDetectionTriggerCap>
  <FaceDetectionTriggerCap><!--optional, xs: EventTriggerCapType--></
FaceDetectionTriggerCap>
  <SceneChangeDetectionTriggerCap><!--optional, xs: EventTriggerCapType--></
SceneChangeDetectionTriggerCap>
  <VandalProofAlarmCap><!--optional, xs: EventTriggerCapType--></
VandalProofAlarmCap>
  <JudgmentTriggerCap><!--optional, xs: EventTriggerCapType--></
JudgmentTriggerCap>
  <FightingTriggerCap><!--optional, xs: EventTriggerCapType--></
FightingTriggerCap>
  <RisingTriggerCap><!--optional, xs: EventTriggerCapType--></RisingTriggerCap>
  <DozingTriggerCap><!--optional, xs: EventTriggerCapType--></DozingTriggerCap>
  <CountingTriggerCap><!--optional, xs: EventTriggerCapType--></
CountingTriggerCap>
  <VideoLossTriggerCap><!--optional, xs: EventTriggerCapType--></
VideoLossTriggerCap>
  <HideTriggerCap><!--optional, xs:EventTriggerCapType--></HideTriggerCap>
  <AlarmInTriggerCap><!--optional, xs: EventTriggerCapType--></
AlarmInTriggerCap>
  <VehicleDetectionTriggerCap><!--optional, xs: EventTriggerCapType--></
VehicleDetectionTriggerCap>
  <AudioExceptionCap><!--optional, xs: EventTriggerCapType--></
AudioExceptionCap>
  <FiledDetectTriggerCap><!--optional, xs: EventTriggerCapType--></
FiledDetectTriggerCap>
  <MotionDetectionTriggerCap><!--optional, xs: EventTriggerCapType--></
MotionDetectionTriggerCap>
  <TemperatureCap><!--optional, xs: EventTriggerCapType--></TemperatureCap>
  <IntelligentTriggerCap><!--optional, xs: EventTriggerCapType--></
IntelligentTriggerCap>
  <FaceContrastTriggerCap><!--optional, xs: EventTriggerCapType, face picture
comparison alarm linkage--></FaceContrastTriggerCap>
  <PersonDensityDetectionTriggerCap><!--optional, xs: EventTriggerCapType--></
PersonDensityDetectionTriggerCap>
  <PersonQueueDetectionTriggerCap><!--optional, xs: EventTriggerCapType, queue
management alarm linkage--></PersonQueueDetectionTriggerCap>
  <HumanRecognitionTriggerCap><!--optional, xs: EventTriggerCapType--></
HumanRecognitionTriggerCap>
  <FaceSnapTriggerCap><!--optional, xs: EventTriggerCapType--></
FaceSnapTriggerCap>
```

```
<isSupportWhiteLightAction>
  <!--dependent, xs: boolean, see details in EventTriggerCapType, it is valid
when isSupportWhiteLight is "true"-->
</isSupportWhiteLightAction>
<isSupportAudioAction>
  <!--dependent, xs: boolean, see details in EventTriggerCapType, it is valid
when isSupportBeep is "true"-->
</isSupportAudioAction>
<HFPDTriggerCap><!--optional, xs: EventTriggerCapType--></HFPDTriggerCap>
<MixedTargetDetectionCap><!--optional, xs: EventTriggerCapType--></
MixedTargetDetectionCap>
<HVTVehicleDetectionTriggerCap><!--optional, xs: EventTriggerCapType--></
HVTVehicleDetectionTriggerCap>
<VCATriggerCap><!--optional, xs: EventTriggerCapType--></VCATriggerCap>
<PIRCap><!--optional, xs: EventTriggerCapType--></PIRCap>
<IllegalParkingTriggerCap><!--optional, xs: EventTriggerCapType, whether it
supports illegal parking detection--></IllegalParkingTriggerCap>
<PedestrianTriggerCap><!--optional, xs: EventTriggerCapType, whether it
supports pedestrian detection--></PedestrianTriggerCap>
<TrafficAccidentTriggerCap><!--optional, xs: EventTriggerCapType, whether it
supports traffic accident detection--></TrafficAccidentTriggerCap>
<ConstructionTriggerCap><!--optional, xs: EventTriggerCapType, whether it
supports construction detection--></ConstructionTriggerCap>
<RoadBlockTriggerCap><!--optional, xs: EventTriggerCapType, whether it
supports roadblock detection--></RoadBlockTriggerCap>
<AbandonedObjectTriggerCap><!--optional, xs: EventTriggerCapType, whether it
supports objects dropped down detection--></AbandonedObjectTriggerCap>
<ParallelParkingTriggerCap><!--optional, xs: EventTriggerCapType, whether it
supports parallel parking detection--></ParallelParkingTriggerCap>
<ParkingStateTriggerCap><!--optional, xs: EventTriggerCapType, whether it
supports parking space status detection, currently this node is not supported--
></ParkingStateTriggerCap>
<CongestionTriggerCap><!--optional, xs: EventTriggerCapType, whether it
supports congestion detection--></CongestionTriggerCap>
<IntersectionAnalysisCap><!--optional, xs: EventTriggerCapType, whether it
supports intersection analysis--></IntersectionAnalysisCap>
<ShipsFlowDetectionTriggerCap><!--optional,xs:EventTriggerCapType, ship flow
detection--></ShipsFlowDetectionTriggerCap>
<dredgerDetectionTriggerCap><!--optional,xs:EventTriggerCapType, dredger
detection--></dredgerDetectionTriggerCap>
<voltageInstableTriggerCap><!--optional,xs:EventTriggerCapType, supply
voltage exception--></voltageInstableTriggerCap>
<HighHDDTemperatureTriggerCap><!--optional, xs:EventTriggerCapType, HDD high
temperature detection--></HighHDDTemperatureTriggerCap>
<LowHDDTemperatureTriggerCap><!--optional, xs:EventTriggerCapType, HDD low
temperature detection--></LowHDDTemperatureTriggerCap>
<HDImpactTriggerCap><!--optional, xs:EventTriggerCapType, HDD impact
detection--></HDImpactTriggerCap>
<HDBadBlockTriggerCap><!--optional, xs:EventTriggerCapType, HDD bad sector
detection--></HDBadBlockTriggerCap>
<SevereHDDFailureTriggerCap><!--optional, xs:EventTriggerCapType, HDD severe
fault detection--></SevereHDDFailureTriggerCap>
```

```
<HUMANATTRIBUTECap><!--optional, xs:EventTriggerCapType--></HUMANATTRIBUTECap>
<HumanAttributeTriggerCap><!--optional, xs:EventTriggerCapType, human body
attribute--></HumanAttributeTriggerCap>
<BlackListFaceContrastTriggerCap><!--opt, xs:EventTriggerCapType, alarm
linkage capability of blacklist face comparison--></
BlackListFaceContrastTriggerCap>
<FaceLibTriggerCap><!--optional, xs:EventTriggerCapType--></FaceLibTriggerCap>
<SafetyHelmetDetectionTriggerCap><!--optional, xs:EventTriggerCapType, alarm
linkage capability of hard hat detection--></SafetyHelmetDetectionTriggerCap>
<VibrationDetectionTriggerCap><!--optional, xs:EventTriggerCapType, alarm
linkage capability of vibration detection--></VibrationDetectionTriggerCap>
<RadarLineDetectionTriggerCap><!--optional, xs:EventTriggerCapType, alarm
linkage capability of radar line crossing detection--></
RadarLineDetectionTriggerCap>
<RadarFieldDetectionTriggerCap><!--optional, xs:EventTriggerCapType, alarm
linkage capability of radar intrusion detection--></
RadarFieldDetectionTriggerCap>
<HBDLibTriggerCap><!--optional, xs:EventTriggerCapType, alarm linkage
capability of human body picture library--></HBDLibTriggerCap>
<FaceThermometryCap><!--optional, xs:EventTriggerCapType--></
FaceThermometryCap>
<NoMaskDetectionTriggerCap><!--optional, xs:EventTriggerCapType, alarm
linkage capability of no wearing mask detection--></NoMaskDetectionTriggerCap>
<TMPATriggerCap><!--optional, xs:EventTriggerCapType, alarm linkage
capability of temperature measurement pre-alarm--></TMPATriggerCap>
<FireEscapeDetectionTriggerCap><!--optional, xs:EventTriggerCapType, alarm
linkage capability of fire engine access detection--></
FireEscapeDetectionTriggerCap>
<TakingElevatorDetectionTriggerCap><!--optional, xs:EventTriggerCapType,
alarm linkage capability of elevator detection--></
TakingElevatorDetectionTriggerCap>
<RuleTriggerCap><!--optional, linkage capability of rule triggered alarm -->
  <isSupportCityManagement>
    <!--optional, xs:boolean, whether the city management supports setting
linkage actions by area; if supports, the value is true, otherwise, this node
will not be returned-->
  </isSupportCityManagement>
</RuleTriggerCap>
<ThermalCalibrationFileExceptionCap><!--optional, xs:EventTriggerCapType,
alarm linkage capability of thermography calibration file exception--></
ThermalCalibrationFileExceptionCap>
</EventTriggersCap>
```

### See Also

#### [XML\\_EventTriggerCapType](#)

## C.33 XML\_EventTriggerCapType

XML message about capability of alarm linkage action types

```

<EventTriggerCapType version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <isSupportCenter><!--optional, xs:boolean--></isSupportCenter>
  <isSupportRecord><!--optional, xs:boolean--></isSupportRecord>
  <isSupportMonitorAlarm><!--optional, xs:boolean--></isSupportMonitorAlarm>
  <isSupportBeep><!--optional, xs: boolean, whether it supports audible
warning--></isSupportBeep>
  <isSupportIO><!--optional, xs:boolean--></isSupportIO>
  <isSupportFTP><!--optional, xs:boolean--></isSupportFTP>
  <isSupportEmail><!--optional, xs:boolean--></isSupEmail>
  <isSupportLightAudioAlarm><!--optional, xs:boolean--></
isSupportLightAudioAlarm>
  <isSupportFocus><!--optional, xs:boolean--></isSupportFocus>
  <isSupportPTZ><!--optional, xs:boolean--></isSupportPTZ>
  <maxPresetActionNum>
    <!--dependent, xs:integer, it is valid only when <isSupportPTZ> is "true"-->
  </maxPresetActionNum>
  <maxPatrolActionNum>
    <!--dependent, xs:integer, it is valid only when <isSupportPTZ> is "true"-->
  </maxPatrolActionNum>
  <maxPatternActionNum>
    <!--dependent, xs:integer, it is valid only when <isSupportPTZ> is "true"-->
  </maxPatternActionNum>
  <isSupportTrack><!--optional, xs:boolean, whether it supports PTZ linked
tracking--></isSupportTrack>
  <isSupportWhiteLight>
    <!--optional, xs: boolean, whether it supports supplement light alarm
linkage-->
  </isSupportWhiteLight>
  <isSupportCloud><!--optional, xs:boolean, whether it supports upload to the
cloud--></isSupportCloud>
  <targetNotificationInterval max="1000" min="0" default="30"><!--xs:integer,
range: [0, 1000], the default value is 30, unit: seconds, this node is valid
for <MotionDetectionTriggerCap> and <TamperDetectionTriggerCap> and this node
is valid when <isSupportPTZ> is "true"--></targetNotificationInterval>
  <direction opt="both,forward,reverse"><!--xs:string, triggering direction,
this node is valid for the node <BlackListTriggerCap>, <WhiteListTriggerCap>,
and <VehicleDetectionTriggerCap>--></direction>
  <presetDurationTime min="" max=""><!--dependent, xs:integer--></
presetDurationTime>
  <isSupportSMS><!--optional, xs:boolean, whether to support SMS (Short Message
Service)--></isSupportSMS>
  <maxCellphoneNum><!--dependent, xs:integer, the maximum number of cellphones,
which is node is valid only when <isSupportSMS> is "true"--></maxCellphoneNum>
  <isSupportOSD><!--optional, xs:boolean--></isSupportOSD>
  <isSupportAudio><!--optional, xs:boolean, whether it supports setting audio
alarm independently. If this node is set to "true", audio alarm and buzzer
alarm can be linked separately, and the linage method is audio--></
isSupportAudio>
  <AudioAction><!--dependent, this node is valid when <isSupportBeep> is "true"
or <isSupportAudio> is "true"-->
  <audioTypeList>
    <audioType><!--list-->

```

```

        <audioID><!--required, xs:integer, alarm sound type--></audioID>
        <audioDescription><!--required, xs:string, alarm sound description, it
should correspond to the alarm sound type--></audioDescription>
    </audioType>
</audioTypeList>
    <alarmTimes opt="0,1,2,3,4,5,6,7,8,9,255"><!--required, xs:integer, alarm
times, it is between 0 and 9, 255-continuous alarm, unit: time--></alarmTimes>
</AudioAction>
    <isSupportSMS><!--optional, xs:boolean --></isSupportSMS>
    <maxCellphoneNum><!--dependent, if <isSupportSMS> is true, xs:integer--></
maxCellphoneNum>
    <isNotSupportCenterModify><!--optional, xs:boolean, whether editing
configuration parameters of the monitoring center is not supported: "true"-yes
(configuration parameters of the monitoring center cannot be edited), "false"
or this node is not returned-no (configuration parameters of the monitoring
center can be edited)--></isNotSupportCenterModify>
    <isSupportMessageConfig>
        <!--optional, xs:boolean, whether it supports SMS configuration, if
supports, set cellphoneNumber to null-->
    </isSupportMessageConfig>
    <isSupportAnalogOutput><!--optional, xs:boolean, whether it supports IO
output of linkage analog channel--></isSupportAnalogOutput>
    <isSupportIOOutputUnify><!--optional, xs:boolean, whether it supports
configuration of IO output--></isSupportIOOutputUnify>
    <isSupportFaceContrast><!--optional, xs:boolean, whether it supports face
picture comparison linkage--></isSupportFaceContrast>
    <isSupportSiren><!--optional, xs:boolean, whether it supports siren linkage--
></isSupportSiren>
    <isSupportOutput><!--optional, xs:boolean, whether it supports relay linkage--
></isSupportOutput>
</EventTriggerCapType>

```

### C.34 XML\_FieldDetectionParam

XML message about rule parameters of intrusion detection

```

<FieldDetectionParam> <!--dep-->
    <durationTime> <!--req, xs: integer--> </durationTime>
    <sensitivityLevel><!--req, xs:integer, sensitivity, from 0 to 100--></
sensitivityLevel>
    <detectionTarget opt="all,human,vehicle, human_vehicle"/>
        <!--detection target, node with underline indicates that it supports
multiple selections, e.g., human_vehicle indicates human and vehicle-->
    <humanMisinfoFilterEnabled>
        <!--opt, xs:boolean,"true,false", whether to prevent false human body
alarm-->
    </humanMisinfoFilterEnabled>
    <vehicleMisinfoFilterEnabled>
        <!--opt, xs:boolean,"true,false", whether to prevent false vehicle alarm-->
    </vehicleMisinfoFilterEnabled>

```

```
</vehicleMisinfoFilterEnabled>  
</FieldDetectionParam>
```

### C.35 XML\_GroupParam

XML message about rule parameters of people gathering detection

```
<GroupParam><!--dep-->  
  <populDensity><!--dep, xs:integer, population density, from 1 to 10--></  
populDensity>  
</GroupParam>
```

### C.36 XML\_IntelliCap

XML message about intelligent capability

```
<IntelliCap version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">  
  <isFaceSupport><!--optional, xs:boolean, whether it supports face detection--  
></isFaceSupport>  
  <isBehaviorSupport><!--optional, xs:boolean, whether it supports behavior  
analysis--></isBehaviorSupport>  
  <isLineDetectionSupport><!--optional, xs:boolean, whether it supports line  
crossing detection--></isLineDetectionSupport>  
  <isFieldDetectionSupport><!--optional, xs:boolean, whether it supports  
intrusion detection--></isFieldDetectionSupport>  
  <isRegionEntranceSupport><!--optional, xs:boolean, whether it supports  
region entrance detection--></isRegionEntranceSupport>  
  <isRegionExitingSupport><!--optional, xs:boolean, whether it supports region  
exiting detection--></isRegionExitingSupport>  
  <isLoiteringSupport><!--optional, xs:boolean, whether it supports loitering  
detection--></isLoiteringSupport>  
  <isGroupSupport><!--optional, xs:boolean, whether it supports people  
gathering detection--></isGroupSupport>  
  <isRapidMoveSupport><!--optional, xs:boolean, whether it supports fast  
moving detection--></isRapidMoveSupport>  
  <isParkingSupport><!--optional, xs:boolean, whether it supports parking  
detection--></isParkingSupport>  
  <isUnattendedBaggageSupport><!--optional, xs:boolean, whether it supports  
unattended baggage detection--></isUnattendedBaggageSupport>  
  <isAttendedBaggageSupport><!--optional, xs:boolean, whether it supports  
object removal detection--></isAttendedBaggageSupport>  
  <isTeacherSupport><!--optional, xs:boolean, whether it supports teacher's  
behavior detection--></isTeacherSupport>  
  <isStudentSupport><!--optional, xs:boolean, whether it supports student's  
behavior detection--></isStudentSupport>  
  <isCombinedSupport><!--optional, xs:boolean, whether it supports combined  
detection--></isCombinedSupport>  
  <isTrafficSupport><!--optional, xs:boolean, whether it supports traffic  
detection--></isTrafficSupport>
```

```
<RestoreLib>
  <libName opt="studentsStoodUp,peopleCounting,shipDetection"><!--optional,
xs:string,--></libName>
</RestoreLib>
<RestoreDefParamForbid>
  <!--optional, xs:boolean "true, false", forbid restoring algorithm library
to default. When the node does not exist, it indicates restoring to default is
not supported. When the node exists and its value is "ture", it indicates
support, when the value is "false", it indicates not support-->
</RestoreDefParamForbid>
<RestoreAlgLibParam>
  <!--optional, the capabilities of restoring algorithm library parameters
according to algorithm library name-->
  <libName opt="faceSnap,HMS,behaviorAnalysis"><!--optional, xs:string--></
libName>
  </RestoreAlgLibParam>
  <isFaceCaptureStatisticsSupport><!--whether it supports face picture
statistics--></isFaceCaptureStatisticsSupport>
  <isSupportPersonQueueDetection><!--whether it supports queue management--></
isSupportPersonQueueDetection>
  <isSupportIntersectionAnalysis>
    <!--optional, xs: boolean, whether it supports intersection analysis-->
  </isSupportIntersectionAnalysis>
  <isSupportCityManagement><!--optional, xs:boolean, whether it supports
intelligent city management. Refer to URI: /ISAPI/Intelligent/channels/<ID>/
cityManagement/capabilities?format=json--></isSupportCityManagement>
  <mixedTargetDetectionWithoutAttribute><!--optional, xs: boolean--></
mixedTargetDetectionWithoutAttribute>
  <isSupportUploadFacePictureByForm><!--optional, xs:boolean, whether it
supports uploading face pictures by form--></isSupportUploadFacePictureByForm>
  <isSupportUploadFacePictureByUrl><!--optional, xs: boolean, whether it
supports uploading face pictures by URL--></isSupportUploadFacePictureByUrl>
  <isSupportUploadHumanPictureByForm><!--optional, xs:boolean, whether it
supports uploading human pictures in form--></isSupportUploadHumanPictureByForm>
  <isSupportMisinfoFilterStatisticalMode><!--optioal, xs:boolean, whether it
supports statistics of filtering face comparison--></
isSupportMisinfoFilterStatisticalMode>
  <isSupportFaceScore><!--optional, xs:boolean, whether it supports face
grading configuration (camera)--></isSupportFaceScore>
  <HumanRecognitionModeSearchCap>
    <searchTargetsNumMax><!--optional, xs:integer, maximum number of sample
pictures that can be imported for searching by picture--></searchTargetsNumMax>
    <HumanMode>
      <searchCond opt="age_group,gender,jacet_color,glass,bag,ride,unlimit"/
><!--setting multiple search conditions is supported-->
      <similarity min="0.0" max="100.0"/><!--optional, xs:float, similarity,
range: [0.0,100.0]-->
    </HumanMode>
    <HumanInfo><!--optional, person attribute information, it is required when
searching for picture by attributes-->
      <age_group
opt="child,infant,kid,middle,middleAged,old,prime,teenager,unknown,young">
```

```
    <!--optional, xs:string, age group:
"child,infant,kid,middle,middleAged,old,prime,teenager,unknown,young"-->
    </age_group>
    <gender opt="female,male,unknown"><!--optional, xs:string, gender:
"female,male,unknown"--></gender>
    <jacet_color
opt="black,black,blue,brown,cyan,deepBlue,deepGray,gray,green,mixture,orange,pin
k,purple,red,silver,unknown,white,yellow">
    <!--optional, xs:string, tops color-->
    </jacet_color>
    <glass opt="yes,no,unknown,sunglasses"><!--optional, xs:string, whether
the person is wearing glasses--></glass>
    <bag opt="yes,no,unknown"><!--optional, xs:string, whether the person is
carrying a bag--></bag>
    <ride opt="yes,no,unknown"><!--optional, xs:string, whether the person is
riding--></ride>
</HumanInfo>
    <positive opt="true,false"/><!--optional, xs:boolean, whether it is false
human body recognition alarm: "true"-yes, "false"-no-->
    <eventType opt="unlimit,humanRecognition">
    <!--optional, xs:string, event type: "unlimit"-no limit,
"humanRecognition"-human body detection alarm-->
    </eventType>
    <isSupportMultiChannelSearch>
    <!--optional, xs:boolean, whether it supports multi-channel search-->
    </isSupportMultiChannelSearch>
    <isSupportTotalSearchResult>
    <!--optional, xs:boolean, whether it supports limiting number of results
that can be obtained after a single search-->
    </isSupportTotalSearchResult>
</HumanRecognitionModeSearchCap>
<VehicleRecognitionModeSearchCap>
    <searchTargetsNumMax><!--optional, xs:integer, maximum number of sample
pictures that can be imported for searching by picture--></searchTargetsNumMax>
    <eventType opt="unlimit,vehicleBlackList,vehicleWhiteList"/><!--xs:string,
event type: "unlimit"-no limit, "vehicleBlackList"-vehicle blacklist,
"vehicleWhiteList"-vehicle allowlist-->
    <VehicleMode>
    <searchCond
opt="licensePlate,vehicleLogo,vehicleSubLogoRecog,vehicleType,vehicleColor,unlim
it"/>
    <similarity min="0.0" max="100.0"/><!--optional, xs:float, similarity,
range: [0.0,100.0]-->
    </VehicleMode>
    <isSupportMultiChannelSearch>
    <!--optional, xs:boolean, whether it supports multi-channel search-->
    </isSupportMultiChannelSearch>
    <isSupportTotalSearchResult>
    <!--optional, xs:boolean, whether it supports limiting number of results
that can be obtained after a single search-->
    </isSupportTotalSearchResult>
</VehicleRecognitionModeSearchCap>
```



```
<FaceContrastAnalyzeCap><!--capability of manual comparison and analysis of
face pictures-->
  <isSupportTrigger opt="false,true"><!--optional, xs:boolean--></
isSupportTrigger>
  <SearchCondition><!--optional-->
    <startTime><!--required, xs:time, ISO8601 time--></startTime>
    <endTime><!--required, xs:time, ISO8601 time--></endTime>
  </SearchCondition>
</FaceContrastAnalyzeCap>
<FramesPeopleCountingCap><!--capability of people counting statistics in a
single frame-->
  <Statistics>
    <SearchCondition>
      <startTime><!--required, xs:time,ISO8601 time--></startTime>
      <endTime><!--required, xs:time,ISO8601 time--></endTime>
    </SearchCondition>
  </Statistics>
  <MaskRegion><!--optional, shielded region-->
    <maxRegionNum><!--optional, xs:integer, number of regions--></
maxRegionNum>
    <Region>
      <vertexNum min="3" max="10"><!--optional, xs:integer, number of region
vertexes--></vertexNum>
    </Region>
  </MaskRegion>
</FramesPeopleCountingCap>
<FaceContrastPersonInfoExtend><!--configuration capability of face comparison
tag-->
  <personInfoCap>
    <maxPersonInfo min="0" max="4">
      <!--optional, xs:integer, maximum number of person tags-->
    </maxPersonInfo>
  </personInfoCap>
  <personInfoFDlibCap>
    <maxPersonInfo min="0" max="4">
      <!--optional, xs:integer, maximum number of person tags-->
    </maxPersonInfo>
  </personInfoFDlibCap>
</FaceContrastPersonInfoExtend>
<isSupportSafetyHelmetDetection>
  <!-- opt, xs:boolean, whether it supports hard hat detection-->
</isSupportSafetyHelmetDetection>
<isSupportDistanceRecognition><!--optional, xs:boolean, whether supports
distance measurement--></isSupportDistanceRecognition>
<isSupportATM><!--optional, xs:boolean, ATM intelligent configuraiton,
corresponds to URI: /ISAPI/Intelligent/ATM/capabilities?format=json--></
isSupportATM>
<isSupportFaceTemperature><!--optional, xs:boolean, whether the device
supports face thermography, corresponds to URI: /ISAPI/Intelligent/
faceTemperature/capabilities?format=json--></isSupportFaceTemperature>
<isSupportFireEscapeDetectionSearch>
  <!--optional, xs:boolean, whether the device supports searching for fire
```

```
engine access detection events of multiple channels (related URI: URI/ISAPI/Intelligent/fireEscapeDetection/search/capabilities?format=json)-->
</isSupportFireEscapeDetectionSearch>
<isSupportTakingElevatorDetectionSearch>
  <!--optional, xs:boolean, whether the device supports searching for
elevator detection events of multiple channels (related URI: /ISAPI/Intelligent/
takingElevatorDetection/search/capabilities?format=json)-->
</isSupportTakingElevatorDetectionSearch>
<isSupportCityManagementSearch>
  <!--optional, xs:boolean, whether the device supports searching for city
management events of multiple channels (related URI: /ISAPI/Intelligent/
cityManagement/search/capabilities?format=json)-->
</isSupportCityManagementSearch>
<isSupportSafetyHelmetDetectionSearch>
  <!--optional, xs:boolean, whether the device supports searching for hard
hat detection events of multiple channels (related URI: /ISAPI/Intelligent/
safetyHelmetDetection/search/capabilities?format=json)-->
</isSupportSafetyHelmetDetectionSearch>
<isSupportFramesPeopleCountingSearch>
  <!--optional, boolean, ro, whether it supports searching for the regional
people counting event (related URI: /ISAPI/Intelligent/framesPeopleCounting/
search/capabilities?format=json)-->true
</isSupportFramesPeopleCountingSearch>
<isSupportChannelPollingSchedules>
  <!--optional, xs:boolean, whether the device supports channel polling
configuration: "true,false"-->
</isSupportChannelPollingSchedules>
<AnalysisImageFaceCap>
  <!--optional, face picture analysis type (related URI:/ISAPI/Intelligent/
analysisImage/face)-->
  <modeling opt="false"><!--optional,xs:boolean, whether the device
supports modeling: "true,false"--></modeling>
</AnalysisImageFaceCap>
</IntelliCap>
```

### C.37 XML\_Intelli\_IntelliTraceCap

IntelliTraceCap capability message in XML format

```
<IntelliTraceCap version="1.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <IntelliTraceBlockList size="">
    <IntelliTraceBlock>
      <sid min="" max=""><!--req, xs:string, scene ID, ranges from 1 to 10,
supports up to 10 scene configurations--></sid>
      <sceneName min="" max=""><!--req, xs:string--></sceneName>
      <patrolId min="" max=""><!--opt, xs:integer, ranges from 0 to 10, 0
means no patrol--></patrolId>
      <dwelldtime min="" max=""><!--opt, xs:integer, the dwell time of the
patrol point, ranges from 30 to 720 (unit: second)--></dwelldtime>
      <traceEnable opt="true,false"><!--req, xs:string, "true, false"--></
```

```
traceEnable>
  <trackDuration min="" max=""><!--req, xs:integer, tracking time, ranges
from 0 to 300 (unit: second), 0 means continuous tracking--></trackDuration>
  <PTZLimitEnable opt="true,false"><!--req, xs:string, "true, false"--
><PTZLimitEnable>
    <SceneNameOverlay>
      <enabled opt="true,false"><!--req, xs:boolean--></enabled>
      <normalizedScreenSize>
        <normalizedScreenWidth><!--req, xs:integer--></normalizedScreenWidth>
        <normalizedScreenHeight><!--req, xs:integer--></
normalizedScreenHeight>
      </normalizedScreenSize>
      <positionX><!--req, xs:integer--></positionX>
      <positionY><!--req, xs:integer--></positionY>
    </SceneNameOverlay>
    <sceneRatio min="" max=""><!--ro, xs:integer, scene ratio--></sceneRatio>
  </IntelliTraceBlock>
</IntelliTraceBlockList>
</IntelliTraceCap>
```

### C.38 XML\_IntelliTraceBlock

IntelliTraceBlock message in XML format

```
<IntelliTraceBlock version="1.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <sid><!--req, xs:string, scene ID, ranges from 1 to 10, supports up to 10
scene configurations--></sid>
  <sceneName><!--req, xs:string--></sceneName>
  <patrolId><!--opt, xs:integer, ranges from 0 to 10, 0 means no patrol--></
patrolId>
  <dwelldtime><!--opt, xs:integer, the dwell time of the patrol point, ranges
from 30 to 720 (unit: second)--></dwelldtime>
  <traceEnable><!--req, xs:string, "true, false"--></traceEnable>
  <trackDuration><!--req, xs:integer, tracking time, ranges from 0 to 300
(unit: second), 0 means continuous tracking--></trackDuration>
  <PTZLimitEnable><!--req, xs:string, "true, false"--><PTZLimitEnable>
  <SceneNameOverlay>
    <enabled><!--req, xs:boolean--></enabled>
    <normalizedScreenSize>
      <normalizedScreenWidth><!--req, xs:integer--></normalizedScreenWidth>
      <normalizedScreenHeight><!--req, xs:integer--></normalizedScreenHeight>
    </normalizedScreenSize>
    <positionX><!--req, xs:integer--></positionX>
    <positionY><!--req, xs:integer--></positionY>
  </SceneNameOverlay>
  <sceneRatio><!--opt, xs:integer, scene ratio--></sceneRatio>
</IntelliTraceBlock>
```

## C.39 XML\_IntelliTraceBlockList

IntelliTraceBlockList message in XML format

```
<IntelliTraceBlockList version="1.0" xmlns="http://www.isapi.org/ver20/
XMLSchema" size="8">
  <IntelliTraceBlock/><!--opt, see details in the message of
XML_IntelliTraceBlock-->
</IntelliTraceBlockList>
```

### See Also

[XML\\_IntelliTraceBlock](#)

## C.40 XML\_LineDetectionParam

XML message about rule parameters of line crossing detection

```
<LineDetectionParam><!--dep-->
  <directionSensitivity><!--req, xs:string, "left-right,right-left,any"--></
directionSensitivity>
  <detectionTarget opt="all,human,vehicle, human_vehicle"/>
  <!--detection target, node with underline indicates that it supports
multiple selections, e.g., human_vehicle indicates human and vehicle-->
  <humanMisinfoFilterEnabled>
    <!--opt, xs:boolean,"true,false", whether enables false alarm filter of
human-->
  </humanMisinfoFilterEnabled>
  <vehicleMisinfoFilterEnabled>
    <!--opt, xs:boolean,"true,false", whether enables false alarm filter of
vehicle-->
  </vehicleMisinfoFilterEnabled>
  <sensitivity><!--req, xs:integer, sensitivity, from 1 to 100--></sensitivity>
</LineDetectionParam>
```

## C.41 XML\_LoiteringParam

XML message about rule parameters of loitering detection

```
<LoiteringParam><!--dep-->
  <durationTime>
    <!--req, xs: integer, duration time, from 1 second to 100 seconds, default
value: 1s-->
  </durationTime>
</LoiteringParam>
```

### C.42 XML\_ParkingParam

XML message about rule parameters of parking detection

```
<ParkingParam><!--dep-->
  <durationTime>
    <!--req, xs:integer, duration time, from 5 seconds to 100 seconds, default
value: 5s-->
  </durationTime>
</ParkingParam>
```

### C.43 XML\_RadarFieldDetectionScheduleList

XML message about arming schedule list of radar intrusion detection

```
<?xml version="1.0" encoding="utf-8"?>
<RadarFieldDetectionScheduleList version="2.0" xmlns="http://www.isapi.org/
ver20/XMLSchema">
  <Schedule/><!--optional, refer to the message XML_Schedule for details-->
</RadarFieldDetectionScheduleList>
```

#### See Also

[XML\\_Schedule](#)

### C.44 XML\_RadarLineDetectionScheduleList

XML message about arming schedule list of radar line crossing detection

```
<RadarLineDetectionScheduleList version="2.0" xmlns="http://www.isapi.org/ver20/
/XMLSchema">
  <Schedule/><!--optional, refer to the message XML_Schedule for details-->
</RadarLineDetectionScheduleList>
```

#### See Also

[XML\\_Schedule](#)

### C.45 XML\_RapidMoveParam

XML message about rule parameters of fast moving detection

```
<RapidMoveParam><!--dep-->
  <rapidMoveMode>
    <!--dep, xs:string, "pixels,actualSize", mode: pixels-pixel, actualSize-
actual size-->
  </rapidMoveMode>
</RapidMoveParam>
```

```
</rapidMoveMode>
<distanceThreshold>
  <!--dep, xs:integer, distance threshold, when rapidMoveMode is "pixels", it
ranges from 1 to 10, when rapidMoveMode is "actualSize", it ranges from 1 to
20-->
</distanceThreshold>
</RapidMoveParam>
```

### C.46 XML\_RegionEntrance

RegionEntrance message in XML format

```
<RegionEntrance version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id><!--req, xs: string--></id>
  <enabled><!--req, xs: boolean--></enabled>
  <normalizedScreenSize><!--req, ro-->
    <normalizedScreenWidth><!--req, read-only, xs: integer--></
normalizedScreenWidth>
    <normalizedScreenHeight><!--req, read-only, xs: integer--></
normalizedScreenHeight>
  </normalizedScreenSize>
  <RegionEntranceRegionList/><!--opt, refer to the message
XML_RegionEntranceRegionList for details-->
  <mutexAbility opt="PDC"/><!--opt, read-only, xs: string, "PDC"-people
counting statistics, which is not supported during region entrance detection-->
  <isSupportMultiScene><!--opt, xs: boolean, whether supports multiple scenes
detection--></isSupportMultiScene>
  <humanMisinfoFilterEnabled><!--opt, xs: boolean, "true,false", whether to
enable false human body alarm prevention--></humanMisinfoFilterEnabled>
  <vehicleMisinfoFilterEnabled><!--opt, xs: boolean, "true,false", whether to
enable false vehicle alarm prevention--></vehicleMisinfoFilterEnabled>
</RegionEntrance>
```

#### See Also

[XML\\_RegionEntranceRegionList](#)

### C.47 XML\_RegionEntranceRegion

XML message about parameters of a region entrance detection region by channel

```
<RegionEntranceRegion version="2.0" xmlns="http://www.isapi.org/ver20/
XMLSchema">
  <id><!--required, xs:string--></id>
  <sensitivityLevel>
    <!--optional, xs: integer, which is between 1and 100, 0-the sensitive is
lowest-->
  </sensitivityLevel>
  <RegionCoordinatesList size="10"><!--optional, if size is not returned, the
```

```
default number of nodes is 4-->
  <RegionCoordinates><!--optional-->
    <positionX><!--required, xs:integer;coordinate--></positionX>
    <positionY><!--required, xs:integer;coordinate --></positionY>
  </RegionCoordinates>
</RegionCoordinatesList>
  <detectionTarget><!--optional, xs: string, detection target type: "all",
"human", "vehicle", "others"-detection targets except human bodies and
vehicles. The value "all" is mutually exclusive with "others"--></
detectionTarget>
  <humanMisinfoFilterEnabled><!--optional, xs:boolean, "true,false", whether to
enable false alarm reduction of human body--></humanMisinfoFilterEnabled>
  <vehicleMisinfoFilterEnabled><!--optional, xs:boolean, "true,false", whether
to enable false alarm reduction of vehicle--></vehicleMisinfoFilterEnabled>
  <priority><!--optional, xs:string, priority: "low,middle,high"--></priority>
  <alarmConfidence><!--optional, xs:string, confidence of alarm notification:
"low", "mediumLow", "mediumHigh", "high", "low"--></alarmConfidence>
  <recordConfidence><!--optional, xs:string, confidence of recording video:
"low", "mediumLow", "mediumHigh", "high", "low"--></recordConfidence>
</RegionEntranceRegion>
```

### C.48 XML\_RegionEntranceRegionList

XML message about parameters of all region entrance detection regions by channel

```
<RegionEntranceRegionList version="2.0" xmlns="http://www.isapi.org/ver20/
XMLSchema">
  <RegionEntranceRegion/><!--optional, refer to the message
XML_RegionEntranceRegion for details-->
</RegionEntranceRegionList>
```

#### See Also

[XML\\_RegionEntranceRegion](#)

### C.49 XML\_RegionEntranceParam

XML message about rule parameters of region entrance detection

```
<RegionEntranceParam> <!--dep-->
  <detectionTarget opt="all,human,vehicle,human_vehicle"/>
  <!--detection target, node with underline indicates that it supports
multiple selections, e.g., human_vehicle indicates human and vehicle-->
  <humanMisinfoFilterEnabled>
    <!--opt, xs:boolean,"true,false", whether enables false alarm filter of
human-->
  </humanMisinfoFilterEnabled>
  <vehicleMisinfoFilterEnabled>
    <!-- opt, xs:boolean,"true,false", whether enables false alarm filter of
```

```
vehicle-->
  </vehicleMisinfoFilterEnabled>
</RegionEntranceParam>
```

### C.50 XML\_RegionExitingParam

XML message about rule parameters of region exiting detection

```
<RegionExitingParam><!--dep-->
  <detectedTarget opt="all,human,vehicle,human_vehicle"/>
  <!--detected target, node with underline indicates that it supports
multiple selections, e.g., human_vehicle indicates human and vehicle-->
  <humanMisinfoFilterEnabled>
    <!--opt, xs:boolean,"true,false", whether enables false alarm filter of
human-->
  </humanMisinfoFilterEnabled>
  <vehicleMisinfoFilterEnabled>
    <!--opt, xs:boolean,"true,false", whether enables false alarm filter of
vehicle-->
  </vehicleMisinfoFilterEnabled>
</RegionExitingParam>
```

### C.51 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>
```

### C.52 XML\_RuleInfo

Message about parameters of behavior analysis rule in XML format

```
<RuleInfo>
  <ruleId><!--required, xs:string, rule ID--></ruleId>
  <ruleName><!--required, xs:string, rule name--></ruleName>
  <enabled><!--required, xs:string, "true,false"--></enabled>
  <eventType>
    <!--required, xs:string,
"none,lineDetection,fieldDetection,regionEntrance,regionExiting,loitering,group,
rapidMove,parking,unattendedBaggage,attendedBaggage,teacher,student,peopleNumCha
nge,spacingChange,violentMotion,leavePosition,failDown,running,retentionParam,ad
vReachHight,situationAnalysis,advReachHight,situationAnalysis,toiletTarry,audioA
bnormal,standUp,getUp,playCellphone"-->
  </eventType>
  <ruleType><!--required, xs:string, "region, line"--></ruleType>
  <LineDetectionParam/><!--dependent, see details in
XML\_LineDetectionParam
-->
  <FieldDetectionParam/><!--dependent, see details in
XML\_FieldDetectionParam
-->
  <RegionEntranceParam/><!--dependent, see details in
XML\_RegionEntranceParam
-->
  <RegionExitingParam/><!--dependent, see details in
XML\_RegionExitingParam
-->
  <LoiteringParam/><!--dependent, see details in
XML\_LoiteringParam
-->
  <GroupParam/><!--dependent, see details in
XML\_GroupParam
-->
  <RapidMoveParam/><!--dependent, see details in
XML\_RapidMoveParam
-->
  <ParkingParam/><!--dependent, see details in
XML\_ParkingParam
-->
  <UnattendedBaggageParam/><!--dependent, see details in
XML\_UnattendedBaggageParam
-->
  <AttendedBaggageParam/><!--dependent, see details in
```

```

                                XML_AttendedBaggageParam
                                -->
<TeacherParam/> <!--dependent, see details in
                                XML_TeacherParam
                                -->
<StudentParam/> <!--dependent-->
<FailDownParam>
    <!--optional, object, parameters of falling down detection-->
    <durationTime><!--optional, int, duration time--></durationTime>
    <sensitivity><!--optional, int, sensitivity--></sensitivity>
    <heightThreshold><!--optional, int, the threshold height--></
heightThreshold>
    <stackTarget><!--optional, boolean, whether to overlay the alarm target
frame--></stackTarget>
    <stackRule><!--optional, boolean, whether to overlay the alarm rule frame--
></stackRule>
    <interval><!--optional, boolean, ro, alarm interval, value range: [4,600],
unit: s, unitType: time--></interval>
</FailDownParam>
<RunningParam/><!--dependent, see details in
                                XML_RunningParam
                                -->
<SizeFilter><!--dimension filter-->
<enabled><!--required, xs:string, "true,false"--></enabled>
<mode><!--required, xs:string, filter mode: "pixels, actualSize"--></mode>
<MaxObjectSize><!--maximum size, float -->
    <positionX><!--required, xs:integer; x-coordinate--></positionX>
    <positionY><!--required, xs:integer; y-coordinate--></positionY>
    <width><!--required, xs:integer--></width>
    <height><!--required, xs:integer--></height>
</MaxObjectSize>
<MinObjectSize><!--minimum size, float-->
    <positionX><!--required, xs:integer;x-coordinate--></positionX>
    <positionY><!--required, xs:integer;y-coordinate--></positionY>
    <width><!--required, xs: integer--></width>
    <height><!--required, xs: integer--></height>
</MinObjectSize>
</SizeFilter>
<RuleRegion><!--Region-->
    <RegionCoordinatesList>
        <RegionCoordinates><!--required-->
            <positionX><!--required, xs:integer;x-coordinate--></positionX>
            <positionY><!--required, xs:integer;y-coordinate--></positionY>
        </RegionCoordinates>
    </RegionCoordinatesList>
</RuleRegion>
<backgroundSuppression>
    <!--optional, xs:string, background suppression, "open,close,selfAdapt"-->
</backgroundSuppression>
<PlayCellphoneParam><!--optional, playing mobile phone detection-->
    <durationTime><!--optional, xs:integer, duration time of playing mobile
phone, value: [1, 3600], unit:second--></durationTime>

```

```
<stackTarget><!--optional, xs:boolean, whether to overlay alarm target
frame--></stackTarget>
<stackRule><!--optional, xs:boolean, whether to overlay alarm rule frame--
></stackRule>
</PlayCellphoneParam>
</RuleInfo>
```

### C.53 XML\_RuleNotification

XML message about alarm linkage parameters of behavior analysis

```
<RuleNotification version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <sid><!--req, xs: string--></sid>
  <RuleList>
    <RuleInfo>
      <ruleId><!--req, xs: string--></ruleId>
      <EventTriggerNotificationList>
        <EventTriggerNotification><!--opt-->
          <id><!--req, xs: string; ID--></id>
          <notificationMethod>
            <!--req, xs: string, linkage action,
"IO,email,record,center,cloud,audio"-->
          </notificationMethod>
          <notificationRecurrence>
            <!--opt, xs: string, alarm status,
"beginning,beginningandend,recurring"-->
          </notificationRecurrence>
          <notificationInterval>
            <!--dep, xs: integer, alarm interval, unit: milliseconds-->
          </notificationInterval>
          <outputIOPortID><!--dep, xs: string; alarm output ID--></
outputIOPortID>
          <dynOutputIOPortID><!--dep, xs: string; dynamic alarm output ID--></
dynOutputIOPortID>
        </EventTriggerNotification>
      </EventTriggerNotificationList>
    </RuleInfo>
  </RuleList>
  <EventList><!--set linkage action by event type; this node cannot exist with
the node RuleList at the same time-->
  <Event><!--list-->
    <eventType><!--req, xs:string, event type--></eventType>
    <mode><!--opt, xs:string, mode: leavePosition (default), sleep--></mode>
    <EventTriggerNotificationList>
      <EventTriggerNotification><!--list-->
        <id><!--req, xs:string, ID--></id>
        <notificationMethod><!--req, xs:string, linkage action: IO, email,
record, center, cloud, whiteLight, audio--></notificationMethod>
        <notificationRecurrence><!--opt, xs:string, linkage time point:
beginning, beginningandend, recurring--></notificationRecurrence>
```

```

        <notificationInterval>
            <!--dep, xs:integer, linkage interval, unit: millisecond; this node
is required when the value of <notificationMethod> is "IO"-->
        </notificationInterval>
        <outputIOPortID>
            <!--dep, xs:string; ID, alarm output No.; this node is required
when the value of <notificationMethod> is "IO"-->
        </outputIOPortID>
        <dynOutputIOPortID>
            <!--dep, xs:string; dynamic alarm output No.; this node is required
when the value of <notificationMethod> is "IO"-->
        </dynOutputIOPortID>
    </EventTriggerNotification>
</EventTriggerNotificationList>
</Event>
</EventList>
    <sceneName><!--optional, xs:string, scene name, which corresponds to <sid>--
></sceneName>
</RuleNotification>

```

### C.54 XML\_RuleSchedule

XML message about arming schedule parameters of behavior analysis

```

<?xml version="1.0" encoding="utf-8"?>
<RuleSchedule version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
    <sid><!--required, xs:string--></sid>
    <RuleList><!--rule information-->
        <RuleInfo><!--list-->
            <ruleId><!--required, xs:string--></ruleId>
            <Schedule>
                <TimeBlockList><!--required-->
                    <TimeBlock><!--list-->
                        <dayOfWeek><!--optional, xs:integer, days of the week, 1-Monday, 2-
Tuesday, 3-Wednesday, 4-Thursday, 5-Friday, 6-Saturday, 7-Sunday--></dayOfWeek>
                        <TimeRange><!--required-->
                            <beginTime><!--required, xs:time, ISO8601 time--></beginTime>
                            <endTime><!--required, xs:time, ISO8601 time--></endTime>
                        </TimeRange>
                    </TimeBlock>
                </TimeBlockList>
            </Schedule>
        </RuleInfo>
    </RuleList>
    <EventList><!--type of event to be set linkage action by; this node cannot
exist with the node RuleList at the same time-->
        <Event><!--list-->
            <eventType><!--required, xs:string,--></eventType>
            <mode><!--optional, xs:string, mode: leavePosition (default), sleep--></
mode>

```

```
<Schedule>
  <TimeBlockList><!--required-->
    <TimeBlock><!--list-->
      <dayOfWeek><!--optional, xs:integer, days of the week, 1-Monday, 2-
Tuesday, 3-Wednesday, 4-Thursday, 5-Friday, 6-Saturday, 7-Sunday--></dayOfWeek>
      <TimeRange><!--required-->
        <beginTime><!--required, xs:time, ISO8601 time--></beginTime>
        <endTime><!--required, xs:time, ISO8601 time--></endTime>
      </TimeRange>
    </TimeBlock>
  </TimeBlockList>
</Schedule>
</Event>
</EventList>
<sceneName><!--optional, xs:string, scene name, which corresponds to <sid>--
></sceneName>
</RuleSchedule>
```

### C.55 XML\_RunningParam

XML message about rule parameters of running detection

```
<RunningParam><!--running-->
  <speed><!--opt, xs:integer--></speed>
  <durationTime><!--optional, xs:integer, duration time--></durationTime>
  <sensitivity><!--optional, xs:integer, sensitivity--></sensitivity>
  <stackTarget><!--optional, xs:boolean, whether to overlay alarm target frame--
></stackTarget>
  <stackRule><!--optional, xs:boolean, whether to overlay alarm rule frame--></
stackRule>
  <mode><!--optional, xs:string, running mode, "single"-one person is running,
"many"-multiple persons are running--></mode>
  <peopleNum><!--optional, int, the number of running people, value range:
[1,50]-->0</peopleNum>
</RunningParam>
```

### C.56 XML\_SafetyHelmetDetection

XML message about hard hat detection parameters

```
<SafetyHelmetDetection version="2.0" xmlns="http://www.isapi.org/ver20/
XMLSchema">
  <normalizedScreenSize><!--required, ro, normalized value-->
    <normalizedScreenWidth>
      <!--required, xs:integer-->
    </normalizedScreenWidth>
    <normalizedScreenHeight>
      <!--required, xs:integer-->
    </normalizedScreenHeight>
```

```
</normalizedScreenSize>
<enabled>
  <!--optional, xs:string, whether to enable the rule: "true"-yes (default),
"false"-no-->
</enabled>
<SafetyHelmetDetectionRegionList/><!--optional, refer to
XML_SafetyHelmetDetectionRegionList for details-->
<stackTarget>
  <!--required, xs:boolean, whether to overlay alarm target frame: "true"-
yes, "false"-no-->
</stackTarget>
<stackRule>
  <!--required, xs:boolean, whether to overlay alarm rule frame: "true"-yes,
"false"-no-->
</stackRule>
<alarmLinkageDuration>
  <!--required, xs:string, alarm linkage duration: "5,10,30,60,120,300,600",
unit: second-->
</alarmLinkageDuration>
<ImageParam>
  <Resolution>
    <id>
      <!--required, xs:string-->
    </id>
    <name>
      <!--required, xs:string, opt="1080p, 720p, D1"-->
    </name>
    <value>
      <!--required, xs:string, opt="1920*1080,1280*720,740*576"-->
    </value>
  </Resolution>
  <quality>
    <!--required, xs:string, picture quality: "best", "good", "general"-
normal-->
  </quality>
  <alarmUpload>
    <!--required, xs:boolean, whether to upload alarm picture: "true"-yes,
"false"-no-->
  </alarmUpload>
  <localStorage>
    <!--optional, xs:boolean, whether to store pictures locally: "true"-yes
"false"-no-->
  </localStorage>
</ImageParam>
<videoOverlay><!--video overlay parameters-->
  <stackTarget><!--required, xs:boolean, whether to overlay the target frame--
></stackTarget>
  <stackRule><!--required, xs:boolean, whether to overlay rule frame--></
stackRule>
</videoOverlay>
<analysisMode>
  <!--optional, string, ro, analysis mode: "realTimeMode"-real time,
```

```
"autoSwitchMode"-auto switch. The engine also has its analysis mode (refer to
the <analysisMode> node in /ISAPI/Intelligent/analysisEngines/<ID>). The device
will be added to the engine with the same analysis mode via the load balancing
method. If the engine is not configured correspondingly, this node will be
failed to e returned, and the error code of taskNotMatchEngine will be
returned-->realTimeMode
    </analysisMode>
</SafetyHelmetDetection>
```

### C.57 XML\_SafetyHelmetDetectionCap

XML message about hard hat detection capability

```
<SafetyHelmetDetectionCap version="2.0" xmlns="http://www.isapi.org/ver20/
XMLSchema">
    <normalizedScreenSize><!--optional, ro, normalized value-->
        <normalizedScreenWidth>
            <!--required, xs:integer-->
        </normalizedScreenWidth>
        <normalizedScreenHeight>
            <!--required, xs:integer-->
        </normalizedScreenHeight>
    </normalizedScreenSize>
    <enabled opt="true,false"/><!--optional, xs:boolean, whether to enable the
rule: "true"-yes (default), "false"-no-->
    <SafetyHelmetDetectionRegionList size="8">
        <SafetyHelmetDetectionRegion>
            <id><!--optional, xs:string, ID--></id>
            <RegionCoordinatesList size=""><!--opt-->
                <RegionCoordinates><!--opt-->
                    <positionX>
                        <!--required, xs:integer, X-coordinate-->
                    </positionX>
                    <positionY>
                        <!--required, xs:integer, Y-coordinate-->
                    </positionY>
                </RegionCoordinates>
            </RegionCoordinatesList>
            <minRegionCoordinatesNum>
                <!--optional, minimum number of nodes-->
            </minRegionCoordinatesNum>
            <maxRegionCoordinatesNum>
                <!--optional, maximum number of nodes-->
            </maxRegionCoordinatesNum>
        </SafetyHelmetDetectionRegion>
    </SafetyHelmetDetectionRegionList><!--opt-->
    <stackTarget opt="true,false">
        <!--required, xs:boolean, whether to overlay alarm target frame: "true"-
yes, "false"-no-->
    </stackTarget>
```

```
<stackRule opt="true,false">
  <!--required, xs:boolean, whether to overlay alarm rule frame: "true"-yes,
"false"-no-->
</stackRule>
<alarmLinkageDuration opt="5,10,30,60,120,300,600">
  <!--required, xs:string, alarm linkage duration, unit: second-->
</alarmLinkageDuration>
<ImageParam>
  <ResolutionList size="3">
    <Resolution>
      <id>
        <!--required, xs:string-->
      </id>
      <name>
        <!--required, xs:string, opt="1080p, 720p, D1"-->
      </name>
      <value>
        <!--required, xs:string, opt="1920*1080, 1280*720, 740*576"-->
      </value>
    </Resolution>
  </ResolutionList>
  <quality opt="best,good,general">
    <!--required, xs:string, picture quality: "best", "good", "general"-
normal-->
  </quality>
  <alarmUpload opt="true,false">
    <!--required, xs:boolean, whether to upload alarm picture: "true"-yes,
"false"-no-->
  </alarmUpload>
  <localStorage opt="true,false">
    <!--optional, xs:boolean, whether to store pictures locally: "true"-yes
"false"-no-->
  </localStorage>
</ImageParam>
<AdvanceConfiguration><!--opt-->
  <headSensitivity min="1" max="5" def="3"><!--optional, xs:integer, head
detection sensitivity, larger value indicates more sensitive detection--></
headSensitivity>
  <humanSensitivity min="1" max="5" def="3"><!--optional, xs:integer, human
body detection sensitivity, larger value indicates more sensitive detection--></
humanSensitivity>
  <alarmAnalysisFrame min="1" max="50" def="12"><!--optional, xs:integer,
number of frames required for alarm analysis, larger value indicates that more
frames are needed to be analyzed and more slowly the alarm will be triggered--
></alarmAnalysisFrame>
  <FDLibList size=""><!--optional, channel's linked face picture library
list-->
  <FDLib><!--list-->
    <id min="" max=""><!--required, ro, xs:integer, No. of items in the
list--></id>
    <FDID min="" max=""><!--required, ro, xs:string, face picture library
ID--></FDID>
```



```

        <thresholdValue min="" max=""><!--optional, xs:integer, similarity
threshold for comparison, which is between 0 and 100, the higher the threshold
is, the more accurate the comparison is. If the threshold for all libraries is
enabled, then threshold for single library will not take effect--></
thresholdValue>
        <customFaceLibID min="" max=""><!--optional, xs:string, custom face
picture library ID, custom ID has priority over the FDID--></customFaceLibID>
    </FDLib>
</FDLibList>
</AdvanceConfiguration>
<SearchDescriptionCap><!--optional, ability of searching for safety helmet
event-->
    <timeSpanMaxNum><!--required, xs:integer, the maximum number of time
periods in which search can be conducted simultaneously--></timeSpanMaxNum>
    <resultMaxNum><!--required, xs:integer, the maximum number of items that
can be searched for at a time--></resultMaxNum>
    <faceContrastResult opt="all,success,failed"><!--required, xs:string,
comparison result: all-all are succeeded or failed; success, failed--></
faceContrastResult>
    <FDLibList size=""><!--optional, list library-->
        <FDLib><!--list-->
            <FDID min="" max=""><!--required, read-only, xs:string, list library
ID--></FDID>
        </FDLib>
    </FDLibList>
    <name min="" max=""><!--optional, xs:string, name; this node is valid when
the FDLibList does exist--></name>
</SearchDescriptionCap>
<videoOverlay><!--video overlay parameters-->
    <stackTarget opt="true,false"><!--required, xs:boolean, whether to overlay
the target frame--></stackTarget>
    <stackRule opt="true,false"><!--required, xs:boolean, whether to overlay
rule frame--></stackRule>
</videoOverlay>
<analysisMode opt="realTimeMode,autoSwitchMode">
    <!--optional, string, ro, analysis mode: "realTimeMode"-real time,
"autoSwitchMode"-auto switch. The engine also has its analysis mode (refer to
the <analysisMode> node in /ISAPI/Intelligent/analysisEngines/<ID>). The device
will be added to the engine with the same analysis mode via the load balancing
method. If the engine is not configured correspondingly, this node will be
failed to e returned, and the error code of taskNotMatchEngine will be
returned-->test
</analysisMode>
</SafetyHelmetDetectionCap>

```

## C.58 XML\_Schedule

Schedule message in XML format

```
<Schedule version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id><!--required, xs:string, ID--></id>
  <eventType>
    <!--optional, xs:string, alarm/event types, see details in the "Remarks"
below-->
  </eventType>
  <inputIOPortID><!--read-only, dependent, xs:string, alarm input No.--></
inputIOPortID>
  <outputIOPortID><!--read-only, dependent, xs:string, alarm output No.--></
inputIOPortID>
  <videoInputChannelID><!--read-only, dependent, xs:string, video input channel
ID--></videoInputChannelID>
  <TimeBlockList size="8"><!--required-->
    <TimeBlock><!--list-->
      <dayOfWeek>
        <!--optional, xs:integer, day of the week based on ISO8601,
"1"=Monday, ...-->
      </dayOfWeek>
      <TimeRange><!--required-->
        <beginTime><!--required, xs:time, ISO 8601 time--></beginTime>
        <endTime><!--required, xs:time, ISO 8601 time--></endTime>
      </TimeRange>
      <CustomExtension>
        <vehicleDetectSceneID>
          <!--required, xs:interger-->
        </vehicleDetectSceneID>
      </CustomExtension>
    </TimeBlock>
  </TimeBlockList>
  <HolidayBlockList><!--optional-->
    <TimeBlock><!--list-->
      <TimeRange><!--required-->
        <beginTime><!--required, xs:time, ISO 8601 time--></beginTime>
        <endTime><!--required, xs:time, ISO 8601 time--></endTime>
      </TimeRange>
    </TimeBlock>
  </HolidayBlockList>
</Schedule>
```

### Remarks

The node **<eventType>** can be set to the following values: IO, VMD, videoloss, PIR, linedetection, fieldddetection, audioexception, facedetection, regionEntrance, regionExiting, loitering, group, rapidMove, parking, unattendedBaggage, attendedBaggage, storageDetection, shipsDetection, HUMANATTRIBUTE, humanAttribute, faceContrast, faceSnap, faceLib, whitelistFaceContrast, personDensityDetection, personQueueDetection, mixedTargetDetection, fireDetection, illegalParking, pedestrian, trafficAccident, construction, roadblock, abandonedObject, parallelParking, parkingState, congestion, intersectionAnalysis, heatMap, reverseEntrance, vehicledetect, safetyHelmetDetection, vibrationDetection, TMPA, faceThermometry, HBDLib,

detectorTemp, detectorSmoke, detectorTamper, smokeFireRecognizes, smokeFireRecognize, indoorPasswayBlock, detectorShelter, detectorMotion, fireNoRegulation, peopleDetections.

### C.59 XML\_SmartCap

SmartCap capability message in XML format

```
<SmartCap version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <isSupportROI><!--opt, xs:boolean, whether to support ROI (Region of
Interest)--></isSupportROI>
  <isSupportFaceDetect><!--opt, xs:boolean, whether to support face detection--
></isSupportFaceDetect>
  <isSupportIntelliTrace><!--opt, xs:boolean--></isSupportIntelliTrace>
  <isSupportFieldDetection><!--opt, xs:boolean, whether to support region
detection--></isSupportFieldDetection>
  <isSupportDefocusDetection><!--opt, xs:boolean--></isSupportDefocusDetection>
  <isSupportAudioDetection><!--opt, xs:boolean--></isSupportAudioDetection>
  <isSupportSceneChangeDetection><!--opt, xs:boolean, whether to support scene
change detection--></isSupportSceneChangeDetection>
  <isSupportLineDetection><!--opt, xs:boolean--></isSupportLineDetection>
  <isSupportRegionEntrance><!--opt, xs:boolean--></isSupportRegionEntrance>
  <isSupportRegionExiting><!--opt, xs:boolean--></isSupportRegionExiting>
  <isSupportLoitering><!--opt, xs:boolean--></isSupportLoitering>
  <isSupportGroup><!--opt, xs:boolean--></isSupportGroup>
  <isSupportRapidMove><!--opt, xs:boolean--></isSupportRapidMove>
  <isSupportParking><!--opt, xs:boolean--></isSupportParking>
  <isSupportUnattendedBaggage><!--opt, xs:boolean--></
isSupportUnattendedBaggage>
  <isSupportAttendedBaggage><!--opt, xs:boolean--></isSupportAttendedBaggage>
  <isSupportPeopleDetection><!--opt, xs:boolean--></isSupportPeopleDetection>
  <isSupportStorageDetection><!--opt, xs:boolean--></isSupportStorageDetection>
  <isSupportShipsDetection><!--opt, xs:boolean--></isSupportShipsDetection>
  <isSupportSmartCalibration><!--opt, xs:boolean--></isSupportSmartCalibration>
  <isSupportShield><!--opt, xs:boolean, whether to support shielded area--></
isSupportShield>
  <isSupportAlgVersion><!--opt, xs:boolean, whether to support algorithm
library version--></isSupportAlgVersion>
  <isSupportVideoOverlap><!--opt, xs:boolean, whether to support text overlay--
></isSupportVideoOverlap>
  <isSupportParkingState><!--opt, xs:boolean, whether to support parking space
status detection--></isSupportParkingState>
  <isSupportChannelResource><!--opt, xs:boolean--></ isSupportChannelResource>
  <isSupportAnalysisUnitSwitch opt="true,false"><!--opt, xs:boolean, whether to
support analysis unit switch--></isSupportAnalysisUnitSwitch>
  <isSupportHFPD><!--opt, xs:boolean, whether to support frequently appeared
person detection--></isSupportHFPD>
  <isSupportLFPD><!--opt, xs:boolean, whether it supports low frequency person
detection, related URI: /ISAPI/SDT/LFPD?format=json--></isSupportLFPD>
  <isSupportImageROI><!--opt, xs:boolean, whether to support smartJpeg (image
```

```
ROI (Region of Interest)--></isSupportImageROI>
</SmartCap>
```

## C.60 XML\_SubscribeEvent

SubscribeEvent message in XML format

```
<SubscribeEvent version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema" >
  <heartbeat>
    <!--optional, xs:integer, heartbeat interval, unit: second, the default
value is 30s-->
  </heartbeat>
  <eventMode>
    <!--required, xs:string, "all"-upload all alarms/events, "list"-upload
specified alarm/event-->
  </eventMode>
  <EventList>
    <Event><!--uploading mode of specified alarm/event, this node exists only
when eventMode is "list"-->
      <type>
        <!--required, xs:string, alarm/event types, which are obtained from the
capability, refer to Alarm/Event Types for Subscription for its values-->
      </type>
      <minorAlarm>
        <!--opt, xs:string, minor alarm type: "0x400,0x401,0x402,0x403", see
details in Access Control Event Type. This node is required when type is
"AccessControllerEvent"-->
      </minorAlarm>
      <minorException>
        <!--opt, xs:string, minor exception type: "0x400,0x401,0x402,0x403",
see details in Access Control Event Type. This node is required when type is
"AccessControllerEvent"-->
      </minorException>
      <minorOperation>
        <!--opt, xs:string, minor operation type: "0x400,0x401,0x402,0x403",
see details in Access Control Event Type. This node is required when type is
"AccessControllerEvent"-->
      </minorOperation>
      <minorEvent>
        <!--opt, xs:string, minor event type: "0x01,0x02,0x03,0x04", see
details in Access Control Event Type. This node is required when type is
"AccessControllerEvent"-->
      </minorEvent>
      <pictureURLType>
        <!--opt, xs:string, alarm picture format: "binary"-binary, "localURL"-
device local URL, "cloudStorageURL"-cloud storage URL-->
      </pictureURLType>
    </Event>
  </EventList>
</channels>
```

```
<!--optional, xs:string, event linked channel information, and multiple
channels can be linked, each channel is separated by comma, e.g., "1,2,3,4..."-->
</channels>
<channels>
  <!--optional, xs:string, specify channels (each channel is separated by
comma, e.g., "1,2,3,4...") to be armed, this node does not exist if you want to
arm all channels, and if this node exists, the sub node <channels> in the node
<Event> is invalid-->
</channels>
<identityKey max="64"/>
  <!--opt, xs: string, interaction command of subscription, supports
subscribing comparison results of face picture library (importing with this
command), the maximum length is 64-->
</SubscribeEvent>
```

### C.61 XML\_TeacherParam

XML message about rule parameters of teacher behavior analysis

```
<TeacherParam><!--dep-->
  <durationTime><!--req, xs: integer--> </durationTime>
</TeacherParam>
```

### C.62 XML\_UnattendedBaggageParam

XML message about rule parameters of unattended baggage detection

```
<UnattendedBaggageParam><!--dep-->
  <durationTime>
    <!--req, xs:integer, duration time, from 5 to 100 seconds, default value: 5
seconds-->
  </durationTime>
</UnattendedBaggageParam>
```

### C.63 XML\_VcaChanAbility

VcaChanAbility message in XML format

```
<?xml version="1.0" encoding="UTF-8"?>
<VcaChanAbility><!--req, VCA capability-->
  <channelNO><!--channel No.--></channelNO>
  <restartLib><!--whether to support restarting algorithm library--></
restartLib>
  <vcaVersion><!--whether to support viewing version information of algorithm
library--></vcaVersion>
  <syncChannelName><!--whether to support setting the channel name of
synchronization analyzer to the camera name--></syncChannelName>
```

```

<VcaDrawMode><!--intelligent information overlay-->
  <dspEncAddTarget opt="true,false"/><!--character overlay target-->
  <dspEncAddRule opt="true,false"/><!--character overlay rule-->
  <dspPicAddTarget opt="true,false"/><!--picture overlay target-->
  <dspPicAddRule opt="true,false"/><!--picture overlay rule-->
</VcaDrawMode>
<AlarmPicResolutionList><!--alarm picture resolution-->
  <picResolutionEntry index="3" name="UXGA" resolution="1600*1200"/>
  <picResolutionEntry index="4" name="SVGA" resolution="800*600"/>
  <picResolutionEntry index="5" name="HD720P" resolution="1280*720"/>
  <picResolutionEntry index="6" name="VGA" resolution="640*480"/>
</AlarmPicResolutionList>
<GlobalSizeFilter><!--global size filter-->
  <mode opt="imagePixel,realWorld,default" default="imagePixel"/>
  <!--filter mode, "imagePixel"-by pixel size,"realWorld"-by actual size,
"default"-->
  <minRect><!--minimum target frame--></minRect>
  <maxRect><!--maximum target frame--></maxRect>
</GlobalSizeFilter>
<Behavior><!--behavior analysis-->
  <EventType-->
    <BlackboardWriting><!--blackboard writing-->
      <Region>
        <vertexNum min="3" max="10"/><!--number of region vertexes-->
      </Region>
    </BlackboardWriting>
    <Lecture><!--teaching-->
      <Region>
        <vertexNum min="3" max="10"/><!--number of region vertexes-->
      </Region>
      <sensitivity min="1" max="5" default="5"/><!--sensitivity-->
      <duration min="0" max="10" default="5"/>
      <!--triggered time threshold of behavior analysis event-->
      <trackingMode opt="auto,horizontal,vertical" default="auto"/>
      <!--tracking mode-->
      <zoomMode opt="fixed,auto" default="fixed"/>
      <!--zooming mode-->
      <SizeFilter>
        <mode opt="imagePixel,realWorld,default" default="imagePixel"/>
        <!--filter mode, "imagePixel"-by pixel size, "realWorld"-by actual
size, "default"-->
        <minRect><!--minimum target frame--></minRect>
        <maxRect><!--maximum target frmae--></maxRect>
      </SizeFilter>
    </Lecture>
    <Answer><!--answering-->
      <Region>
        <vertexNum min="3" max="10"/><!--number of region vertexes-->
      </Region>
      <sensitivity min="1" max="5" default="5"/><!--sensitivity-->
      <SizeFilter>
        <mode opt="imagePixel,realWorld,default" default="imagePixel"/>

```

```
    <!--filter mode, "imagePixel"-by pixel size, "realWorld"-by actual
size, "default"-->
    <minRect><!--minimum target frame--></minRect>
    <maxRect><!--maximum target frmae--></maxRect>
</SizeFilter>
</Answer>
<TraversePlane><!--plane crossing-->
    <planeBottom><!--plane bottom border--></planeBottom>
    <crossDirection opt="bothDirection,leftToRight,rFaceSnapightToLeft"
default="bothDirection"/>
    <!--crossing direction, "bothDirection"-dual-direction, "leftToRight"-
left to right, "rightToLeft"-right to left-->
    <sensitivity min="1" max="5" default="5"/><!--sensitivity-->
    <planeHeight min="0" max="255" default="5"/><!--plane height-->
    <detectionTarget opt="all,human,vehicle, human_vehicle"/>
    <!--detection target: "all", "human", "vehicle", "human_vehicle"-human
and vehicle-->
    <eventPriority opt="low,mid,high"/>
    <!--opt, event priority: "low", "mid"-medium, "high"-->
    <isSupportHumanMisinfoFilter>
        <!--opt, xs:boolean, whether to support preventing false human body
alarms, "true"-yes, "false"-no-->
    </isSupportHumanMisinfoFilter>
    <isSupportVehicleMisinfoFilter>
        <!--opt, xs:boolean, whether to support preventing false vehicle
alarms, "true"-yes, "false"-no-->
    </isSupportVehicleMisinfoFilter>
</TraversePlane>
<EnterArea><!--region entrance-->
    <Region>
        <vertexNum min="3" max="10"/><!--number of region vertexes-->
    </Region>
    <detectionTarget opt="all,human,vehicle, human_vehicle"/>
    <!--detection target: "all", "human", "vehicle", "human_vehicle"-human
and vehicle-->
    <eventPriority opt="low,mid,high"/>
    <!--opt, event priority: "low", "mid"-medium, "high"-->
    <isSupportHumanMisinfoFilter>
        <!--opt, xs:boolean, whether to support preventing false human body
alarms, "true"-yes, "false"-no-->
    </isSupportHumanMisinfoFilter>
    <isSupportVehicleMisinfoFilter>
        <!--opt, xs:boolean, whether to support preventing false vehicle
alarms, "true"-yes, "false"-no-->
    </isSupportVehicleMisinfoFilter>
</EnterArea>
<ExitArea><!--region exiting-->
    <Region>
        <vertexNum min="3" max="10"/><!--number of region vertexes-->
    </Region>
    <detectionTarget opt="all,human,vehicle"/>
    <!--detection target: "all", "human", "vehicle", "human_vehicle"-human
```

```

and vehicle-->
    <eventPriority opt="low,mid,high"/>
    <!--opt, event priority: "low", "mid"-medium, "high"-->
    <isSupportHumanMisinfoFilter>
        <!--opt, xs:boolean, whether to support preventing false human body
alarms, "true"-yes, "false"-no-->
    </isSupportHumanMisinfoFilter>
    <isSupportVehicleMisinfoFilter>
        <!--opt, xs:boolean, whether to support preventing false vehicle
alarms, "true"-yes, "false"-no-->
    </isSupportVehicleMisinfoFilter>
</ExitArea>
<Intrusion><!--intrusion-->
    <Region>
        <vertexNum min="3" max="10"/><!--number of region vertexes-->
    </Region>
    <duration min="0" max="100" default="5"/>
    <!--triggered time threshold of intrusion alarm-->
    <sensitivity min="1" max="100" default="50"/><!--sensitivity-->
    <rate min="1" max="100" default="5"/><!--percentage-->
    <detectionTarget opt="all,human,vehicle,human_vehicle"/>
    <!--detection target: "all", "human", "vehicle", "human_vehicle"-human
and vehicle-->
    <eventPriority opt="low,mid,high"/>
    <!--opt, event priority: "low", "mid"-medium, "high"-->
    <isSupportHumanMisinfoFilter>
        <!--opt, xs:boolean, whether to support preventing false human body
alarms, "true"-yes, "false"-no-->
    </isSupportHumanMisinfoFilter>
    <isSupportVehicleMisinfoFilter>
        <!--opt, xs:boolean, whether to support preventing false vehicle
alarms, "true"-yes, "false"-no-->
    </isSupportVehicleMisinfoFilter>
</Intrusion>
<Loiter><!--loitering-->
    <Region>
        <vertexNum min="3" max="10"/><!--number of region vertexes-->
    </Region>
    <duration min="1" max="120" default="10"/><!--triggered time threshold--
>
    <eventPriority opt="low,mid,high"/>
    <!--opt, event priority: "low", "mid"-medium, "high"-->
    <loiterDistance min="100" max="5000" default="1000"/>
    <!--opt, total distance threshold of loitering, unit: cm-->
</Loiter>
<LeftTake><!--object left and removal-->
    <Region>
        <vertexNum min="3" max="10"/><!--number of region vertexes-->
    </Region>
    <duration min="1" max="120" default="10"/><!--triggered time threshold--
>
    <eventPriority opt="low,mid,high"/>

```



```
<!--opt, event priority: "low", "mid"-medium, "high"-->
</LeftTake>
<Parking><!--parking-->
  <Region>
    <vertexNum min="3" max="10"/><!--number of region vertexes-->
  </Region>
  <duration min="1" max="100" default="10"/><!--triggered time threshold--
>
  <eventPriority opt="low,mid,high"/>
  <!--opt, event priority: "low", "mid"-medium, "high"-->
</Parking>
<Run><!--running-->
  <Region>
    <vertexNum min="3" max="10"/><!--number of region vertexes-->
  </Region>
  <distance min="0.1" max="1.0" default="0.5"/>
  <!--maximum distance of people running-->
  <mode opt="imagePixel,realWorld" default="imagePixel"/>
  <!--filter mode: "imagePixel"-by pixel size, "realWorld"-by actual
size-->
  <ImagePixelMode><!--parameters for filter mode by pixel size-->
    <distance min="0.1" max="1.0" default="0.5"/>
    <!--maximum distance of people running-->
  </ImagePixelMode>
  <RealWorldMode><!--parameters for filter mode by actual size-->
    <distance min="1" max="20" default="10"/>
    <!--maximum distance of people running-->
  </RealWorldMode>
  <detectionTarget opt="all,human,vehicle,human_vehicle"/>
  <!--detection target: "all", "human", "vehicle", "human_vehicle"-human
and vehicle-->
  <eventPriority opt="low,mid,high"/>
  <!--opt, event priority: "low", "mid"-medium, "high"-->
</Run>
<HighDensity><!--people gathering-->
  <Region>
    <vertexNum min="3" max="10"/><!--number of region vertexes-->
  </Region>
  <density min="0.1" max="1.0" default="0.5"/><!--density ratio-->
  <duration min="20" max="360" default="20"/>
  <!--triggered parameter threshold of people gathering alarm-->
  <eventPriority opt="low,mid,high"/>
  <!--opt, event priority: "low", "mid"-medium, "high"-->
</HighDensity>
<ViolentMotion><!--violent motion-->
  <Region>
    <vertexNum min="3" max="10"/><!--number of region vertexes-->
  </Region>
  <duration min="1" max="120" default="50"/><!--triggered time threshold--
>
  <sensitivity min="1" max="5" default="5"/><!--sensitivity-->
  <mode opt="video,audio,videoAndAudio" default="video"/>
```

```
<!--"video"-video only mode, "audio"-audio only mode, "videoAndAudio"-
video and audio mode-->
</ViolentMotion>
<ReachHeight><!--climbing-->
  <planeBottom><!--plane climbing--></planeBottom>
  <duration min="1" max="120" default="10"/>
  <!--triggered threshold of climbing alarm-->
</ReachHeight>
<GetUp><!--getting up-->
  <Region>
    <vertexNum min="3" max="10"/><!--number of region vertexes-->
  </Region>
  <duration min="1" max="100" default="10"/>
  <!--triggered threshold of getting up alarm-->
  <sensitivity min="1" max="10" default="10"/><!--sensitivity-->
  <mode opt="overBed,areaMove,sitting" default="overBed"/>
  <!--"overBed"-wide bed mode, "areaMove"-bunk bed mode, "sitting"-wide
bed sitting and getting up mode-->
  <eventPriority opt="low,mid,high"/>
  <!--opt, event priority: "low", "mid"-medium, "high"-->
</GetUp>
<Left><!--unattended baggage-->
  <Region>
    <vertexNum min="3" max="10"/><!--number of region vertexes-->
  </Region>
  <duration min="10" max="100" default="10"/>
  <!--triggered threshold of unattended baggage alarm-->
  <sensitivity min="1" max="5" default="5"/><!--sensitivity-->
  <eventPriority opt="low,mid,high"/>
  <!--opt, event priority: "low", "mid"-medium, "high"-->
</Left>
<Take><!--object removal-->
  <Region>
    <vertexNum min="3" max="10"/><!--number of region vertexes-->
  </Region>
  <duration min="10" max="100" default="10"/>
  <!--triggered threshold of object removal alarm-->
  <sensitivity min="1" max="5" default="5"/><!--sensitivity-->
  <eventPriority opt="low,mid,high"/>
  <!--opt, event priority: "low", "mid"-medium, "high"-->
</Take>
<LeavePosition><!--absence-->
  <Region>
    <vertexNum min="3" max="10"/><!--number of region vertexes-->
  </Region>
  <leaveDelay min="1" max="1800" default="120"/>
  <!--absence alarm time-->
  <staticDealy min="1" max="1800" default="120"/>
  <!--sleeping alarm time-->
  <mode opt="leave,sleep,leaveAndSleep" default="leave"/>
  <!--"leave"-absence, "sleep"-sleeping, "leaveAndSleep"-absence and
sleeping-->
```

```
<personType opt="single,couple" default="single"/>
<!--duty mode: "single"-single people on duty, "couple"-double people
on duty-->
  <OnPosition min="1" max="10" default="1"/><!--number of people on duty--
>
</LeavePosition>
<Trail><!--tailgating-->
  <Region>
    <vertexNum min="3" max="10"/><!--number of region vertexes-->
  </Region>
  <sensitivity min="1" max="5" default="5"/><!--sensitivity-->
</Trail>
<KeyPersonGetUp><!--key person getting up-->
  <Region>
    <vertexNum min="3" max="10"/><!--number of region vertexes-->
  </Region>
  <duration min="1" max="100" default="20"/>
  <!--triggered threshold of getting up alarm-->
  <sensitivity min="1" max="10" default="5"/><!--sensitivity-->
  <mode opt="overBed,areaMove,sitting" default="overBed"/>
  <!--"overBed"-wide bed mode, "areaMove"-bunk bed mode, "sitting"-wide
bed sitting and getting up mode-->
</KeyPersonGetUp>
<FallDown><!--people falling down-->
  <Region>
    <vertexNum min="3" max="10"/><!--number of region vertexes-->
  </Region>
  <duration min="1" max="60" default="30"/><!--triggered time threshold-->
  <sensitivity min="1" max="5" default="5"/><!--sensitivity-->
  <heightThreshold min="10" max="100"/>
  <!--triggered height threshold of people falling down alarm, unit: cm-->
</FallDown>
<AudioAbnormal><!--sudden change of sound intensity-->
  <decibel min="1" max="255" default="50"/><!--sound intensity-->
  <sensitivity min="1" max="5" default="5"/><!--sensitivity-->
  <audioMode opt="sensitivity,decibel,sensitivityAndDecibel"
default="sensitivity"/>
  <!--"sensitivity"-sensitivity detection, "decibel"-decibel threshold
detection, "sensitivityAndDecibel"-sensitivity and decibel threshold detection--
>
  <enabled><!--enable mark--></enabled>
  <threshold min="0" max="100" default="80"/><!--sound threshold-->
</AudioAbnormal>
<ADVReachHeight><!--polyline climbing-->
  <Region>
    <vertexNum min="2" max="10"/><!--number of region vertexes-->
  </Region>
  <crossDirection opt="bothDirection,leftToRight,rightToLeft"
default="bothDirection"/>
  <!--crossing direction: "bothDirection"-dual-direction, "leftToRight"-
left to right, "rightToLeft"-right to left-->
</ADVReachHeight>
```

```
<ToiletTarry><!--in-toilet overtime-->
  <Region>
    <vertexNum min="3" max="10"/><!--number of region vertexes-->
  </Region>
  <duration min="1" max="3600" default="600"/><!--time of in-toilet
overtime-->
</ToiletTarry>
<YardTarry><!--playground overstay-->
  <Region>
    <vertexNum min="3" max="10"/><!--number of region vertexes-->
  </Region>
  <duration min="1" max="120" default="60"/><!--time of playground
overstay-->
</YardTarry>
<ADVTraversePlane><!--plane polyline crossing-->
  <Region>
    <vertexNum min="2" max="10"/><!--number of region vertexes-->
  </Region>
  <crossDirection opt="bothDirection,leftToRight,rightToLeft"
default="bothDirection"/>
  <!--crossing direction: "bothDirection"-dual-direction, "leftToRight"-
left to right, "rightToLeft"-right to left-->
  <sensitivity min="1" max="5" default="5"/><!--sensitivity-->
</ADVTraversePlane>
<OverTime><!--operation timeout-->
  <Region>
    <vertexNum min="3" max="10"/><!--number of region vertexes-->
  </Region>
  <duration min="4" max="60000" default="30000"/>
  <!--triggered time threshold of operation timeout alarm-->
</OverTime>
<StickUp><!--sticking-->
  <Region>
    <vertexNum min="3" max="10"/><!--number of region vertexes-->
  </Region>
  <duration min="4" max="60" default="30"/><!--triggered time threshold-->
  <sensitivity min="1" max="5" default="5"/><!--sensitivity-->
</StickUp>
<InstallScanner><!--installing card reader-->
  <Region>
    <vertexNum min="3" max="10"/><!--number of region vertexes-->
  </Region>
  <duration min="4" max="60" default="30"/><!--card reading duration-->
  <sensitivity min="1" max="5" default="5"/><!--sensitivity-->
</InstallScanner>
<Standup><!--people standing up-->
  <Region>
    <vertexNum min="3" max="10"/><!--number of region vertexes-->
  </Region>
  <sensitivity min="1" max="100" default="5"/><!--sensitivity-->
  <duration min="1" max="3600" default="2"/><!--triggered time threshold--
>
```

```

    <heightThreshold min="0" max="250" default="130"/><!--height threshold-->
>
</Standup>
<PeopleNumChange><!--number of people exception-->
    <Region>
        <vertexNum min="3" max="10"/><!--number of region vertexes-->
    </Region>
    <sensitivity min="1" max="100" default="5"/><!--sensitivity-->
    <peopleNumThreshold min="0" max="5" default="1"/>
    <!--number of people threshold-->
    <detectMode opt="greaterthan,lessthan,equal,notEqual"/>
    <!--detection mode: "greaterthan"-greater than, "lessthan"-less than,
"equal"-equal to, "notEqual"-not equal to-->
    <noneStateEffective opt="true,false"/>
    <!--whether it is valid when there are no people-->
    <duration min="1" max="3600" default="2"/><!--triggered time threshold-->
>
</PeopleNumChange>
<SpacingChange><!--space distance exception-->
    <Region>
        <vertexNum min="3" max="10"/><!--number of region vertexes-->
    </Region>
    <sensitivity min="1" max="100" default="5"/><!--sensitivity-->
    <spacingThreshold min="0" max="10.0" default="1.0"/><!--space distance
threshold-->
    <detectMode opt="greaterthan,lessthan,equal,notEqual"/>
    <!--detection mode: "greaterthan"-greater than, "lessthan"-less than,
"equal"-equal to, "notEqual"-not equal to-->
    <duration min="1" max="3600" default="2"/><!--triggered time threshold-->
>
</SpacingChange>
<CombinedRule><!--combined rule-->
    <supportCombinedRule min="" max=""/>
    <relateRuleID min="" max=""/>
    <ruleSequence opt="ascending, ascendingOrDescending"/>
    <minTimeInterval min="" max=""/>
    <maxTimeInterval min="" max=""/>
    <rule1ID min="" max=""/>
    <rule2ID min="" max=""/>
</CombinedRule>
<SitQuietly><!--opt, sitting quietly-->
    <duration min="" max=""/><!--req, duration -->
</SitQuietly>
<HighDensityStatus><!--people gathering status-->
    <Region>
        <vertexNum min="3" max="10"/><!--number of region vertexes-->
    </Region>
    <density min="0.1" max="1.0" default="0.5"/><!--density ratio-->
    <sensitivity min="1" max="5"/><!--sensitivity-->
</HighDensityStatus>
<FakeCard><!--fake card detection-->
    <Region>

```

```
<vertexNum min="3" max="10"/> <!--number of region vertexes-->
</Region>
<sensitivity min="1" max="5" default="3"/><!--sensitivity-->
<eventPriority opt="low,mid,high"/><!--optional, event priority: 0-low,
1-medium, 2-high"-->
</FakeCard>
</EventType>
<BehaviorRule><!--behavior rule-->
  <picProcType opt="notProcess,upload"/>
  <!--picture processing method: "notProcess"-not process, "upload"-upload--
>
  <uploadLastAlarm opt="false,true"/><!--upload the latest alarm-->
  <picRecordEnable opt="false,true"/><!--picture storage-->
  <JpegParam>
    <picSize><!--picture size--></picSize>
    <picQuality opt="best,better,normal"/>
    <!--picture quality: "best", "better", "normal"-->
  </JpegParam>
  <maxRelSnapChanNum><!--maximum number of capture linkage channels--></
maxRelSnapChanNum>
  <RuleEntryList>
    <maxRuleNum><!--maximum number of rules--></maxRuleNum>
    <RuleEntry>
      <eventType opt="traversePlane, enterArea, exitArea, intrusion,
loiter, leftAndTake, parking, run, highDensity, violentMotion, reachHeight,
getup, left, take, leavePosition, trail, keyPersonGetup, fallDown,
audioAbnormal, advReachHeight, toiletTarry, yardTarry, advTraversePlane,
humanEnter, overTime, stickup, installScanner, standup, peopleNumChange,
spacingChange, lecture, answer, combinedRule, sitQuietly, writing, fakeCard"/>
      <!--event type: "traversePlane"-plane crossing, "enterArea"-region
entrance, "exitArea"-region exiting, "intrusion", "loiter"-loitering,
"leftAndTake"-object left and removal, "parking", "run"-running, "highDensity"-
occupant density in the region, "violentMotion"-violent motion, "reachHeight"-
climbing, "getup"-getting up, "left"-unattended baggage, "take"-object removal,
"leavePosition"-absence, "trail"-tailgating, "keyPersonGetup"-key person
getting up, "fallDown"-people falling down, "audioAbnormal"-sudden change of
sound intensity, "advReachHeight"-polyline climbing, "toiletTarry"-in-toilet
overtime, "yardTarry"-playground overstay, "advTraversePlane"-plane polyline
crossing, "humanEnter"-people approaching ATM, "overtime"-operation timeout,
"stickup"-sticking, "installScanner"-installing card reader, "standup"-standing
up, "peopleNumChange"-number of people change, "spacingChange"-space distance
change, "lecture"-teaching, "answer"-answering questions, "combinedRule"-
combined rule, "sitQuietly"-sitting quietly, "writing"-writing on the
blackboard, "fakeCard"-fake card detection-->
      <ruleNameLength min="0" max="32"/><!--rule name length-->
      <SizeFilter>
        <mode opt="imagePixel,realWorld,default" default="imagePixel"/>
        <!--filter mode: "imagePixel"-by pixel size,"realWorld"-by actual
size, "default"-->
        <minRect><!--minimum target frame--></minRect>
        <maxRect><!--maximum target frame--></maxRect>
      </SizeFilter>
```

```

        <AlarmTime>
            <timeSegNum min="2" max="8"/><!--number of arming period per day-->
        </AlarmTime>
        <alarmHandleType
opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp"/>
            <!--"monitor"-warning on the monitor, "audio"-audible warning,
"center"-notify surveillance center, "alarmout"-trigger alarm output, "picture"-
capture JPEG picture and send by email, "wirelesslight"-wireless light and
audio alarm, "uploadftp"-capture picture and upload to FTP-->
            <relRecordChan><!--alarm triggered channel recording--></
relRecordChan>
            <alarmDelay opt="5,10,30,60,120,300,600"/><!--alarm delay-->
            <FilterStrategy>
                <strategy opt="disabled,widthAndHeight,targetArea"
default="disabled"/>
                <!--filter strategy: "disabled"-disable, "widthAndHeight"-height
and width filter, "targetArea"-target area filter-->
            </FilterStrategy>
            <TriggerParam>
                <tiggerMode opt="disabled,trackPoint,targetArea"
default="disabled"/>
                <!--triggering mode: "disabled"-disable, "trackPoint"-track point,
"targetArea"-target area-->
                <triggerPoint opt="center,up,down" default="center"/>
                <!--triggered point: "center", "up", "down"-->
                <triggerArea min="0" max="100" default="50"/>
                <!--triggered target area percentage-->
            </TriggerParam>
            <intervalTime min="" max="" default=""/>
            <!--interval between two alarms (unit: second), ranges from 1 to
7200, the default value is 600-->
            <heightLimit min="" max=""/>
            <!--minimum height of the detected target (unit: cm), ranges from 0
to 250, the default value is 80, target lower than the minimum height will not
be detected-->
            <fallDownMaxRuleNum><!--maximum number of rules for people falling
down detection, if this node is not returned, the maximum number depends on
maxRuleNum--></ fallDownMaxRuleNum>
            <violentMotionMaxRuleNum><!--maximum number of rules for violent
motion, if this node is not returned, the maximum number depends on maxRuleNum--
></ violentMotionMaxRuleNum>
        </RuleEntry>
    </RuleEntryList>
</BehaviorRule>
<MaskRegion><!--shielded region-->
    <maxRegionNum><!--number of regions--></maxRegionNum>
    <Region>
        <vertexNum min="3" max="10"/><!--number of region vertexes-->
    </Region>
</MaskRegion>
<EnterRegion><!--region entrance-->
    <maxRegionNum><!--number of regions--></maxRegionNum>

```

```

    <Region>
      <vertexNum min="3" max="10"/><!--number of region vertexes-->
    </Region>
  </EnterRegion>
  <Calibration><!--calibration-->
    <BehaviorIn><!--behavior calibration in indoor scene-->
      <enabled opt="true,false"/><!--whether to enable-->
      <calSampleNum min="2" max="5"/><!--number of calibration samples-->
      <CalSample><!--calibration sample-->
        <targetRect><!--target frame--></targetRect>
        <LineSegment>
          <lineMode opt="heightLine,lengthLine"/>
          <!--"heightLine"-height sample line, "lengthLine"-length sample
line-->
            <value min="1" max="1000"/><!--height or length-->
          </LineSegment>
        </CalSample>
      <CameraParam>
        <cameraHeight min="2" max="50"/><!--camera height-->
        <cameraPitchAngle min="1" max="89"/><!--camera tilt angle-->
        <horizonLine min="0.0" max="1.0"/><!--horizon line in the scene-->
      </CameraParam>
    </BehaviorIn>
    <BehaviorOut><!--behavior calibration in outdoor scene-->
      <enabled opt="true,false"/><!--whether to enable-->
      <lineSegNum min="4" max="8"/><!--number of sample lines-->
      <LineSegment>
        <lineMode opt="heightLine,lengthLine"/>
        <!--"heightLine"-height sample line, "lengthLine"-length sample line--
>
          <value min="1" max="1000"/><!--height or length-->
        </LineSegment>
      <CameraParam>
        <cameraHeight min="2" max="50"/><!--camera height-->
        <cameraPitchAngle min="1" max="89"/><!--camera tilt angle-->
        <horizonLine min="0.0" max="1.0"/><!--horizon line in the scene-->
      </CameraParam>
    </BehaviorOut>
    <calibVerify><!--whether to support calibration verification--></
calibVerify>
  </Calibration>
  <Scene>
    <maxSceneNum><!--maximum number of scenes--></maxSceneNum>
    <uploadSceneID><!--whether uploads event/alarm scene ID: "true"--></
uploadSceneID>
  </Scene>
</Behavior>
<Traffic><!--traffic-->
  <Scene><!--scene parameter-->
    <maxSceneNum><!--maximum number of scenes--></maxSceneNum>
    <maxSceneTimeSegNum><!--maximum number of time period in traffic scene--
></maxSceneTimeSegNum>

```



```

    <SceneParam>
      <direction><!--detection direction--></direction>
      <sceneID><!--scene ID--></sceneID>
      <sceneNameLength min="0" max="32"/><!--scene name length-->
      <ptzPos><!--PTZ coordinate--></ptzPos>
      <trackTime min="5" max="300"/><!--speed dome tracking time-->
    </SceneParam>
  </Scene>
  <MaskRegion><!--shielded region-->
    <maxRegionNum><!--number of regions--></maxRegionNum>
    <Region>
      <vertexNum min="3" max="10"/><!--number of region vertexes-->
    </Region>
  </MaskRegion>
  <ReferenceRegion><!--reference region-->
    <maxRegionNum><!--number of regions--></maxRegionNum>
    <Region>
      <vertexNum min="3" max="10"/><!--number of region vertexes-->
    </Region>
  </ReferenceRegion>
  <Calibration><!--calibration-->
    <enabled opt="true,false"/><!--whether to enable-->
    <calibPointNum min="4" max="4"/><!--number of calibration points-->
    <width><!--width--></width>
    <height><!--height--></height>
    <calibVerify><!--whether to support calibration verification--></
calibVerify>
  </Calibration>
  <LaneCfg><!--lane configuration-->
    <maxLaneNum><!--maximum number of lanes--></maxLaneNum>
    <LaneParam>
      <laneNameLength min="0" max="32"/><!--lane name length-->
      <flowDirection><!--traffic flow direction in the lane--></flowDirection>
      <Region>
        <vertexNum min="3" max="10"/><!--number of region vertexes-->
      </Region>
    </LaneParam>
  </LaneCfg>
  <AidRule><!--traffic event rules-->
    <picProcType opt="notProcess,upload"/>
    <!--picture processing method: "notProcess"-not process, "upload"-upload--
>
    <JpegParam>
      <picSize><!--picture size--></picSize>
      <picQuality opt="best,better,normal"/>
      <!--picture quality: "best", "better", "normal"-->
    </JpegParam>
    <RuleEntryList>
      <maxRuleNum><!--maximum number of rules--></maxRuleNum>
      <RuleEntry>
        <ruleNameLength min="0" max="32"/><!--rule name length-->
        <eventType>

```

```
opt="congestion,parking,inverse,pedestrian,debris,smoke,overLine,vehicleControlList,speed,illegalLaneChange,turnAround"/>
    <!--event type: "congestion", "parking", "inverse"-driving in the
opposite direction, "pedestrian", "debris"-thrown object, "smoke", "overLine"-
driving over the line, "vehicleControlList"-blocklist, "speed"-speeding,
"illegalLaneChange"-illegal lane change, "turnAround"-turning around-->
    <SizeFilter>
        <mode opt="imagePixel,realWorld,default" default="imagePixel"/>
        <!--filter mode, "imagePixel"-by pixel size, "realWorld"-by actual
size, "default"-->
        <minRect><!--minimum target frame--></minRect>
        <maxRect><!--maximum target frame--></maxRect>
    </SizeFilter>
    <Region>
        <vertexNum min="3" max="10"/><!--number of region vertexes-->
    </Region>
    <AidParam>
        <parkingDuration min="10" max="120" default="100"/>
        <!--parking duration-->
        <pedestrianDuration min="1" max="120" default="100"/>
        <!--pedestrian duration-->
        <debrisDuration min="10" max="120" default="100"/>
        <!--thrown object duration-->
        <congestionLength min="5" max="200" default="100"/>
        <!--congestion length threshold-->
        <congestionDuration min="10" max="120" default="60"/>
        <!--congestion duration-->
        <inverseDuration min="1" max="10" default="5"/>
        <!--duration of driving in the opposite direction-->
        <inverseDistance min="2" max="100" default="50"/>
        <!--distance threshold of driving in the opposite direction-->
        <inverseAngleTolerance min="90" max="180" default="100"/>
        <!--allowed angle deviation-->
        <illegalParkingTime min="4" max="60" default="10"/>
        <!--illegal parking time-->
        <illegalParkingPicNum min="1" max="6" default="4"/>
        <!--number of illegal parking pictures-->
        <mergePic><!--whether to support joint picture--></mergePic>
    </AidParam>
    <AlarmTime>
        <timeSegNum min="2" max="8"/>
        <!--number of arming period per day-->
    </AlarmTime>
    <alarmHandleType
opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp"/>
    <!--"monitor"-warning on the monitor, "audio"-audible warning,
"center"-notify surveillance center, "alarmout"-trigger alarm output, "picture"-
capture JPEG picture and send by email, "wirelesslight"-wireless light and
audio alarm, "uploadftp"-capture picture and upload to FTP-->
    <relRecordChan><!--alarm triggered channel recording--></
relRecordChan>
    <laneNo min="1" max="99"/><!--lane No.-->
```

```

        </RuleEntry>
    </RuleEntryList>
</AidRule>
<TpsRule><!--traffic data statistics rule-->
    <RuleEntryList>
        <maxRuleNum><!--maximum number of rules--></maxRuleNum>
        <RuleEntry>
            <laneID><!--lane ID--></laneID>
            <calcType
opt="laneVolume, laneVelocity, timeHeadway, spaceHeadway, timeOccupancyRatio, spaceOc
cupancyRatio, queue, vehicleType, trafficState"/>
            <!--statistics parameter type: "laneVolume"-lane traffic,
"laneVelocity"-lane speed, "timeHeadway"-time headway, "spaceHeadway"-space
headway, "timeOccupancyRatio"-lane occupancy percentage (time),
"spaceOccupancyRatio"-lane occupancy percentage (space), "queue"-queue length,
"vehicleType"-vehicle type, "trafficState"-traffic status-->
            <SizeFilter>
                <mode opt="imagePixel, realWorld, default" default="imagePixel"/>
                <!--filter mode, "imagePixel"-by pixel size, "realWorld"-by actual
size, "default"-->
                <minRect><!--minimum target frame--></minRect>
                <maxRect><!--maximum target frame--></maxRect>
            </SizeFilter>
            <Region>
                <vertexNum min="3" max="10"/><!--number of region vertexes-->
            </Region>
            <AlarmTime>
                <timeSegNum min="2" max="8"/>
                <!--number of arming period per day-->
            </AlarmTime>
            <alarmHandleType
opt="monitor, audio, center, alarmout, picture, wirelesslight, uploadftp"/>
            <!--"monitor"-warning on the monitor, "audio"-audible warning,
"center"-notify surveillance center, "alarmout"-trigger alarm output, "picture"-
capture JPEG picture and send by email, "wirelesslight"-wireless light and
audio alarm, "uploadftp"-capture picture and upload to FTP-->
        </RuleEntry>
    </RuleEntryList>
</TpsRule>
<forensicsMode opt="manual, auto"/>
<!--evidence capture mode: "manual", "auto"-automatic-->
<trafficSceneMode opt="freeway, tunnel, bridge"/>
<!--traffic scene mode: "freeway"-outdoor scene in the freeway, "tunnel"-
tunnel scene in the freeway, "bridge"-bridge scene in the freeway-->
<ITCTriggerCfg><!--video triggering parameter-->
    <triggerMode opt="VIAVirtualCoil"/>
    <!--triggering mode: "VIAVirtualCoil"-VIA (Vehicle characteristics
Intelligent Analysis) virtual coil triggering-->
    <VIAVirtualCoilParam><!--VIA virtual coil triggering parameter-->
        <laneNum><!--number of lanes--></laneNum>
        <laneBoundaryLine><!--lane boundary line, which is the left boundary
line of the leftmost lane--></laneBoundaryLine>

```

```
<VIALane>
  <maxLaneNum><!--maximum number of lanes--></maxLaneNum>
  <associatedLaneNO><!--lane No. linkage--></associatedLaneNO>
  <LaneLogicParam>
    <laneUseage
opt="unknown,carriageWay,bus,fast,slow,motor,nonMotor,reverse,banTrucks,mix"/>
    <!--lane type: "unknown", "carriageWay"-normal lane, "bus"-bus
lane, "fast"-express lane, "slow"-slow lane, "motor"-motorcycle lane,
"nonMotor"-non-motor vehicle lane, "reverse"-opposite lane, "banTruck"-non-
truck lane, "mix"-all-purpose lane-->
    <laneDirection
opt="unknown,left,straight,leftStraight,right,leftRight,rightStraight,leftRightS
traight,leftWait,straightWait,forward,backward,bothway"/>
    <!--lane direction: "unknown", "left"-turning left, "straight"-
going straight, "leftStraight"-turning left and going straight, "right"-turning
right, "leftRight"-turning left and turning right, "rightStraight"-turning
right and going straight, "leftRightStraight"-turning left, turning right and
going straight, "leftWait"-waiting to turn left, "straight"-waiting to go
straight, "forward"-driving forward, "backward"-driving backward, "bothway"-
bidirectional driving-->
    <carDriveDirection opt="unknown,uptodown,downtoup"/>
    <!--driving direction: "unknown", "uptodown"-driving in the down
direction, "downtoup"-driving in the up direction-->
  </LaneLogicParam>
  <laneLine><!--lane line--></laneLine>
  <Region>
    <vertexNum min="3" max="20"/>
    <!--plate recognition area, number of region vertexes-->
  </Region>
</VIALane>
<PlateRecogParam>
  <provinceAbbreviation><!--Chinese character abbreviation of the
province where the device operates--></provinceAbbreviation>
  <RecogMode><!--recognition mode-->
    <plateRecogPos opt="fromFront,fromBack"/>
    <!--plate recognition position, "fromFront"-front plate
recognition, "fromBack"-rear plate recognition-->
    <plateRecogType opt="big,little"/>
    <!--plate recognition type, "big"-large license plate, "little"-
small license plate-->
    <locationType opt="sceneLocation,frameLocation"/>
    <!--location type, "sceneLocation"-scene location, "frameLocation"-
frame location-->
    <recogType opt="sceneRecognition,frameRecognition"/>
    <!--recognition type, "sceneRecognition"-scene recognition,
"frameRecognition"-frame recognition-->
    <recogTime opt="daytime,night"/>
    <!--recognition time, "daytime", "night"-->
    <sceneType opt="EPolice,gate"/>
    <!--scene type, "EPolice"-e-police, "gate"-checkpoint-->
    <microPlateRecog opt="true,false"/>
    <!--smaller license plate recognition-->
```

```

        <farmVehicleRecog opt="true,false"/>
        <!--farm vehicle recognition-->
        <vechileColorRecog opt="true,false"/>
        <!--vehicle color recognition-->
        <motorCarRecog opt="true,false"/>
        <!--motorcycle recognition-->
        <blurRecog opt="true,false"/>
        <!--fuzzy recognition-->
    </RecogMode>
    <vehicleLogoRecog opt="true,false"/><!--vehicle logo recognition-->
</PlateRecogParam>
</VIAVirtualCoilParam>
</ITCTriggerCfg>
</Traffic>
<PDC><!--people counting statistics-->
    <PDCType opt="smart, professionalIntelligence"/>
    <!--intelligent people counting type: "smart"-smart device,
"professionalIntelligence"-professional intelligent iDS device-->
    <PDCRule>
        <Region>
            <vertexNum min="4" max="10"/><!--number of region vertexes-->
        </Region>
        <Line>
            <support opt="leftLine,rightLine"/>
        </Line>
        <enterDirection><!--people counting entrance direction--></enterDirection>
        <dayStartTime><!--daytime start time--></dayStartTime>
        <nightStartTime><!--night start time--></nightStartTime>
        <AlarmTime>
            <timeSegNum min="2" max="8"/><!--number of arming period per day-->
        </AlarmTime>
        <alarmHandleType
opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp"/>
        <!--req, handling type-->
        <detecteSensitive min="" max=""/>
        <!--target detection sensitivity, ranges from 1 to 100, the default value
is 50-->
        <generatieSpeedSpace min="" max=""/>
        <!--target generating speed (space), ranges from 1 to 100, the default
value is 50-->
        <generatieSpeedTime min="" max=""/>
        <!--target generating speed (time), ranges from 1 to 100, the default
value is 50-->
        <countSpeed min="" max=""/>
        <!--counting speed, ranges from 1 to 100, the default value is 50-->
        <detecteType opt="auto,head,shoulder"/>
        <!--target detection type: "auto"-automatic detection, "head"-head
detection, "shoulder"-head and shoulder detection, the default value is "auto"--
>
        <targetSizeCorrect min="" max=""/>
        <!--target size correction, ranges from 1 to 100, the default value is
50-->

```

```

</PDCRule>
<Calibration>
  <calibRectNum min="1" max="6"/>
  <!--number of calibration frames-->
  <calibLine><!--whether to support calibration line--></calibLine>
  <calibVerify><!--whether to support calibration verification--></
calibVerify>
</Calibration>
<resetCounter><!--reset counting parameter--></resetCounter>
<resetCounterMode opt="timing,manual"/><!--reset counting mode-->
<OSD>
  <enable opt="disable,enable"/><!--req, enable people counting OSD
display-->
  <topLeftPoint><!--req, whether to support the top left coordinate--></
topLeftPoint>
  <OSDType opt="enter,leave,enterLeave,none,peoplePassing,
enterPeoplePassing">
    <!--opt, xs:string, display type: "enter"-number of people entered,
"leave"-number of people left, "enterLeave"-number of people entered and left,
"none", "peoplePassing"-number of people passed, "enterPeoplePassing"-number of
people entered and passed-->
    </OSDType>
  </OSD>
  <mutexAbility
opt="fieldDetection,traversingVirtualPlane,regionEntrance,regionExiting,loiterin
g,group,rapidMove,parking,unattendedBaggage,attendedBaggage"/><!--req, mutex
ability-->
  <brokenNetHttp><!--opt, whether to support ANR--></brokenNetHttp>
  <SecUploadEnable opt="true,false"/>
  <!--opt, xs:boolean, enable to upload every second-->
  <DataUploadCycle opt="1,5,10,15,20,30,60"/>
  <!--opt, xs:inter, period of uploading people counting detection statistics
(unit: minute)-->
  <isSupportRecommendedValue>
    <!--opt, xs:boolean, return true if supported, otherwise this node will
not be returned-->
  </isSupportRecommendedValue>
  <isSupportFlashRemoveCouting>
    <!--opt, xs:boolean, clear people counting statistics in Flash, return
true if supported, otherwise this node will not be returned-->
  </isSupportFlashRemoveCouting>
  <InterferenceSuppression><!--opt, interference suppression-->
    <shadow opt="true,false">
      <!--opt, xs:boolean, shadow-->
    </shadow>
    <loitering opt="true,false">
      <!--opt, xs:boolean, loitering-->
    </loitering>
    <cart opt="true,false">
      <!--opt, xs:boolean, pushing the vehicle-->
    </cart>
  </InterferenceSuppression>

```

```
<EmailReport><!--opt, report by email-->
  <DayReport opt="true,false">
    <!--opt, xs:boolean, daily people counting report-->
  </DayReport>
  <WeekReport opt="true,false">
    <!--opt, xs:boolean, weekly people counting report-->
  </WeekReport>
  <MonthReport opt="true,false">
    <!--opt, xs:boolean, monthly people counting report-->
  </MonthReport>
  <YearReport opt="true,false">
    <!--opt, xs:boolean, annual people counting report-->
  </YearReport>
</EmailReport>
<CountingCalibrate><!--opt, installation calibration verification-->
  <calibrateType opt="automatic>manual,no">
    <!--opt, xs:string, "automatic"-automatic calibration, "manual"-manual
calibration, "no"-no calibration-->
  </calibrateType>
  <SetupParam><!--req, installation parameters-->
    <height min="" max="">
      <!--opt, xs:float, height, unit: cm-->
    </height>
    <tiltAngle min="" max="">
      <!--opt, ro, xs:float, tilt angle-->
    </tiltAngle>
    <heelAngle min="" max="">
      <!--opt, ro, xs:float, heel angle-->
    </heelAngle>
  </SetupParam>
  <CountingArea><!--req, ro, counting area (red frame)-->
    <RegionCoordinatesList size="">
      <RegionCoordinates><!--req, region coordinate points-->
        <positionX>
          <!--req, xs:integer, X-coordinate-->
        </positionX>
        <positionY>
          <!--req, xs:integer, Y-coordinate-->
        </positionY>
      </RegionCoordinates>
    </RegionCoordinatesList>
  </CountingArea>
  <AutomaticCalib><!--opt, it is valid when calibrateType is "Automatic"-->
    <CalibRegion><!--opt, calibration region (green frame)-->
      <RegionCoordinatesList size="">
        <RegionCoordinates><!--req, region coordinate points-->
          <positionX>
            <!--req, xs:integer, X-coordinate-->
          </positionX>
          <positionY>
            <!--req, xs:integer, Y-coordinate-->
          </positionY>
        </RegionCoordinates>
      </CalibRegion>
    </AutomaticCalib>
  </CountingArea>
```

```

        </RegionCoordinates>
    </RegionCoordinatesList>
</CalibRegion>
</AutomaticCalib>
</CountingCalibrate>
<HeightFilterOverlay><!--opt-->
    <enable opt="true,false" default="">
        <!--opt, xs:boolean-->
    </enable>
    <heightFilter min="" max="" default="">
        <!--opt, xs:integer-->
    </heightFilter>
</HeightFilterOverlay>
<isSupportPosInfoOverlay opt="true,false">
    <!--opt, xs:boolean-->
</isSupportPosInfoOverlay>
<isSupportCalibrate opt="true,false">
    <!--opt, xs:boolean, whether to support calibration-->
</isSupportCalibrate>
<isSupportSearchPeoplePassing opt="true,false">
    <!--opt, xs:boolean, whether to support searching people passed-->
</isSupportSearchPeoplePassing>
<countingType opt="none,alarmInputTrigger,videoTrigger" />
    <!--opt, xs:string, triggering counting mode: "none", "alarmInputTrigger"-
alarm input triggering, "videoTrigger"-VCA triggering-->
    <alarmInputTrigger_signalType opt="level,pulses" />
    <!--opt, xs:string, signal type: "level"-level, "pulses"-pulse-->
    <videoTrigger_RS485Transmission opt="true,false" />
    <!--opt, xs:boolean, enable RS-485 data transmission-->
    <isSupportTriggerPeopleCountingDataSearch><!--opt, whether to support
triggering searching people counting statistics--></
isSupportTriggerPeopleCountingDataSearch>
    <isSupportDailyResetTime><!--opt, whether to support scheduled resetting--
></isSupportDailyResetTime>
    <Polyline><!--opt, whether to support polyline detection-->
        <maxPointNumber>
            <!--req, xs:integer, supported number of polyline endpoints-->
        </maxPointNumber>
    </Polyline>
    <isSupportBusDetectionArea><!--opt, whether to support mobile bus entry/
exit detection area--></isSupportBusDetectionArea>
    <isSupportStreamOverlayRuleInfos><!--opt, whether to support stream VCA
information overlay--></isSupportStreamOverlayRuleInfos>
    <isSupportInterfaceSegmentation><!--opt, whether to support window division
by 4200 software--></isSupportInterfaceSegmentation>
    <isSupportMobileDescribeChange><!--opt, this node is used to mark the
edited the parameter name of people counting camera interface, which can
distinguish the baseline product and mobile product--></
isSupportMobileDescribeChange>
    <isSupportAdvanceConfiguration><!--opt, this node is used to mark whether
the camera supports advanced configuration--></isSupportAdvanceConfiguration>
    <isSupportSearchDoorStates><!--opt, this node is used to support detecting

```



```
the door status triggered by mobile signal--></isSupportSearchDoorStates>
  <isSupportMultiChannelSearch><!--opt, whether supports people counting of
multiple channels--></isSupportMultiChannelSearch>
</PDC>
  <HeatMapDetection><!--req, heat map detection-->
    <enable opt="true,false"/><!--req, "false"-disable, "true"-enable-->
    <detSenceID min="" max=""/><!--req, detection scene ID-->
    <heatMapRegionNum><!--req, number of heat map detection regions--></
heatMapRegionNum>
    <HeatMapParam><!--req, corresponding number of this node-->
      <regionNum min="3" max="10"/>
      <!--req, number of valid points supported by each heat map detection
region-->
      <targetTrackEnable opt="true,false"/>
      <!--req, target track: "true"-open, "false"-close, the default value is
"false"-->
      <sensitivityLevel min="0" max="100"/>
      <!--req, sensitivity-->
      <backgroundUpdateRate min="1" max="100"/>
      <!--req, background updating rate, ranges from 1 to 100, the default
value is 50-->
      <sceneChangeLevel min="1" max="100"/>
      <!--req, scene change level, ranges from 1 to 100, the default value is
50-->
      <minTargetSize min="1" max="100"/>
      <!--req, minimum target size, ranges from 1 to 100, the default value is
50-->
    </HeatMapParam>
    <alarmTime><!--req, number of arming period--></alarmTime>
    <alarmHandleType
opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp"/>
    <!--req, handling type: "monitor"-warning on the monitor, "audio"-audible
warning, "center"-notify surveillance center, "alarmout"-trigger alarm output,
"picture"-capture JPEG picture and send by email (Uploadftp was missing in
earlier versions and was added later. If this string cannot be resolved,
firstly check whether to support capturing pictures and uploading to FTP, and
then resolve the uploadftp node of the software and hardware capability of the
device.), "wirelesslight"-wireless light and audio alarm, "uploadftp"-capture
picture and upload to FTP-->
    <brokenNetHttp><!--opt, whether to support ANR--></brokenNetHttp>
    <uploadHeatMapResultType opt="0,1"/><!--opt, heat map type, 0-standard heat
map, 1-heat map of people counting and dwell duration-->
    <confidence min="0.00" max="100.00"/><!--opt, whether supports setting
target picture confidence-->
    <searchType opt="duration,PDC"/><!--opt, statistics types, dwell duration
statistics or people counting statistics--->
  </HeatMapDetection>
  <Face><!--facial detection-->
    <FaceDetect>
      <eventType
opt="abnormalFace,normalFace,multiFace,sunglassesFace,callFace"/>
      <!--"abnormalFace"-abnormal face, "normalFace"-normal face, "multiFace"-
```

```

multiple faces, "sunglassesFace"-face with sunglasses, "callFace"-face on the
phone-->
    <uploadLastAlarm opt="false,true"/><!--upload the latest alarm-->
    <uploadFacePic opt="false,true"/><!--upload face sub-picture-->
    <picRecordEnable opt="false,true"/><!--picture storage-->
    <ruleNameLength min="0" max="32"/><!--rule name length-->
    <Region>
        <vertexNum min="3" max="10"/><!--number of region vertexes-->
    </Region>
    <picProcType opt="notProcess,upload"/>
    <!--picture processing method: "notProcess"-not process, "upload"-upload--
>
    <sensitivity min="1" max="100" default="10"/><!--sensitivity-->
    <duration min="4" max="60" default="30"/><!--triggered time threshold of
face alarm-->
    <JpegParam>
        <picSize><!--picture size--></picSize>
        <picQuality opt="best,better,normal"/>
        <!--picture quality: "best", "better", "normal"-->
    </JpegParam>
    <SizeFilter>
        <mode opt="imagePixel,realWorld,default" default="imagePixel"/>
        <!--filter mode, "imagePixel"-by pixel size, "realWorld"-by actual
size, "default"-->
        <minRect><!--minimum target frame--></minRect>
        <maxRect><!--maximum target frame--></maxRect>
    </SizeFilter>
    <AlarmTime>
        <timeSegNum min="2" max="8"/><!--number of arming period per day-->
    </AlarmTime>
    <alarmHandleType
opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp"/>
    <!--"monitor"-warning on the monitor, "audio"-audible warning, "center"-
notify surveillance center, "alarmout"-trigger alarm output, "picture"-capture
JPEG picture and send by email, "wirelesslight"-wireless light and audio alarm,
"uploadftp"-capture picture and upload to FTP-->
    <relRecordChan><!--alarm triggered channel recording--></relRecordChan>
    <alarmDelay opt="5,10,30,60,120,300,600"/><!--alarm delay-->
    <FaceInPicture><!--face picture-in-picture-->
        <enabled opt="true,false"/><!--whether to enable-->
        <backChannel><!--background channel--></backChannel>
        <position opt="topLeft,topRight,bottomLeft,bottomRight"
default="bottomRight"/>
        <!--overlay position: "topLeft"-top left, "topRight"-top right,
"bottomLeft"-bottom left, "bottomRight"-bottom right-->
        <division opt="1/4,1/9,1/16"/><!--window division ratio-->
        <IPChannel>
            <isSupport><!--whether to support IP channel, newly added to
DeepinMind ATM, old devices only support face PIP of analog channel--></
isSupport>
            <isSupportBackChannel><!--whether to support setting IP channel as
the background channel, newly added to DeepinMind ATM, old devices only support

```

```
setting analog channel as the background channel--></isSupportBackChannel>
</IPChannel>
</FaceInPicture>
<maxRelSnapChanNum><!--maximum number of capture linkage channels--></
maxRelSnapChanNum>
<isSupportEventTypeEx><!--whether the device supports EventTypeEx
extension field. It is used to determine the compatibility of iVMS-4200--></
isSupportEventTypeEx>
<AlgParam>
  <!--opt, configuration display capability of face detection algorithm
library, only used in ATM project. The configuration items not contained in the
capability will display by default without being determined by the capability-->
  <FaceDetectAlg><!--opt, face detection algorithm capability-->
    <FaceDetectList size=""><!--opt, face detection algorithm capability
list-->
      <FaceDetect><!--opt-->
        <eventType
opt="abnormalFace,normalFace,multiFace,sunglassesFace,callFace"/>
        <!--"abnormalFace"-abnormal face, "normalFace"-normal face,
"multiFace"-multiple faces, "sunglassesFace"-face with sunglasses, "callFace"-
face on the phone-->
        <sensitivity min="1" max="5" default="3"/><!--sensitivity-->
        <duration min="30" max="1800" default="60"/>
        <!--triggered time threshold of face alarm, unit: second-->
      </FaceDetect>
    </FaceDetectList>
  </FaceDetectAlg>
</AlgParam>
<NotDisplayDefaultDetectRegion><!--The rule configuration will not
display the default detection range. It is mainly used to determine the
compatibility of iVMS-4200--></NotDisplayDefaultDetectRegion>
</FaceDetect>
<FaceSnap><!--face capture-->
  <snapTime min="0" max="10" default="5"/><!--capture times-->
  <snapInterval min="0" max="255" default="24"/><!--capture interval, unit:
frame-->
  <snapThreshold min="0" max="100" default="80"/><!--capture threshold-->
  <generateRate min="1" max="5" default="3"/><!--target generating rate-->
  <sensitivity min="1" max="5" default="5"/><!--sensitivity-->
  <referenceBright min="0" max="100" default="80"/><!--reference
brightness-->
  <matchType opt="alarmRealtime,alarmAfterDisappear"
default="alarmRealtime"/>
  <!--comparison alarm mode: "alarmRealtime"-real-time alarm,
"alarmAfterDisappear"-alarm after the target disappears-->
  <matchThreshold min="0" max="100" default="80"/>
  <!--real-time comparison threshold-->
  <JpegParam>
    <picSize><!--picture size--></picSize>
    <picQuality opt="best,better,normal"/><!--picture, "best", "better",
"normal"-->
  </JpegParam>
```

```

<RuleEntryList>
  <maxRuleNum><!--maximum number of rules--></maxRuleNum>
  <RuleEntry>
    <SizeFilter>
      <mode opt="imagePixel,realWorld,default" default="imagePixel"/>
      <!--filter mode, "imagePixel"-by pixel size, "realWorld"-by actual
size, "default"-->
      <minRect><!--minimum target frame--></minRect>
      <maxRect><!--maximum target frame--></maxRect>
      <PupillaryDistance><!--pupil distance limit-->
        <width min="" max=""/><!--width limit-->
      </PupillaryDistance>
    </SizeFilter>
    <Region>
      <vertexNum min="3" max="10"/><!--number of region vertexes-->
    </Region>
  </RuleEntry>
  <autoROI opt="false,true"/>
  <!--whether to enable automatic face ROI switch: "false"-no, "true"-
yes-->
</RuleEntryList>
<faceExposure opt="false,true"/>
<!--whether to enable face exposure: "false"-no, "true"-yes-->
<faceExposureMinDuration min="" max=""/><!--minimum face exposure
duration-->
<backgroundPic opt="open,close"/>
<!--req, whether to enable uploading background picture: "open"(default),
"close"-->
<faceFilteringTime min="" max="" default="">
  <!--opt, face dwell time filter, ranges from 0 to 100, the default
value is 5-->
</faceFilteringTime>
<isSupportMultiScene opt="true,false">
  <!--opt, xs:boolean, whether to support face multi-scene configuration--
>
</isSupportMultiScene>
<FaceContrast>
  <!--opt, face comparison-->
  <enabled opt="false,true"/>
</FaceContrast>
<brokenNetHttp><!--opt, whether to support ANR--></brokenNetHttp>
</FaceSnap>
<SnapDevAccess><!--capture access parameters-->
  <devIP opt="ipv4,ipv6"/><!--IP address-->
  <devPort min="8000" max="65535"/><!--port-->
  <usernameLength min="0" max="32"/><!--username length-->
  <passwordLength min="0" max="16"/><!--password length-->
</SnapDevAccess>
<SavePathCfg><!--storage path parameters-->
  <partitionNum><!--number of partitions--></partitionNum>
  <SinglePath>
    <type opt="snapFace,blacklistAlarm,snapFaceAndBlacklistAlarm"

```

```
default="snapFaceAndBlacklistAlarm"/>
    <!--storage type: "snapFace"-capture face, "blacklistAlarm"-blocklist
alarm, "snapFaceAndBlacklistAlarm"-capture face and blocklist alarm-->
    <saveAlarmPic><!--whether to support saving offline alarm picture--></
saveAlarmPic>
    <diskDriver min="0" max="32"/><!--disk driver symbol-->
    <reservedSpace min="10" max="20" default="10"/><!--reserved space,
unit: G-->
    </SinglePath>
</SavePathCfg>
<MaskRegion><!--shielded region-->
    <maxRegionNum><!--number of regions--></maxRegionNum>
    <Region>
        <vertexNum min="3" max="10"/><!--number of region vertexes-->
    </Region>
</MaskRegion>
</Face>
<VQD><!--video quality diagnostics-->
    <VQDEventType
opt="blur,luma,chroma,snow,streak,freeze,signalLoss,ptzControl,sceneChange,video
Abnormal,videoBlock"/>
    <!--"blur"-blurred image, "luma"-brightness exception, "chroma"-color cast,
"snow"-snow noise, "streak"-stripe noise, "freeze"-frame freezing, "signalLoss"-
signal loss, "ptzControl"-uncontrollable PTZ, "sceneChange"-sudden scene
change, "videoAbnormal"-video exception, "videoBlock"-video tampering-->
    <VQDRule>
        <EventParam>
            <threshold min="0" max="100"/><!--alarm threshold-->
            <triggerMode opt="continuous,single"/>
            <!--alarm triggering mode: "continuous"-continuous triggering, "single"-
single triggering-->
            <uploadPic opt="false,true"/><!--upload alarm picture-->
            <timeInterval min="0" max="3600" default="120"/>
            <!--time interval of continuous alarm triggering, unit: second-->
        </EventParam>
        <AlarmTime>
            <timeSegNum min="2" max="8"/><!--number of arming period per day-->
        </AlarmTime>
        <alarmHandleType
opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp"/>
        <!--"monitor"-warning on the monitor, "audio"-audible warning, "center"-
notify surveillance center, "alarmout"-trigger alarm output, "picture"-capture
JPEG picture and send by email, "wirelesslight"-wireless light and audio alarm,
"uploadftp"-capture picture and upload to FTP-->
        <relRecordChan><!--alarm triggered channel recording--></relRecordChan>
    </VQDRule>
    <downloadAlarmPic><!--download VQD alarm picture--></downloadAlarmPic>
</VQD>
<ObjectColor><!--object color-->
    <objectType opt="coat"/><!--object type, "coat"-->
    <colorMode opt="value,picture" default="value"/>
    <!--coloring mode: "value"-color value, "picture"-->
```

```

    <colorValue>
      <brightness min="0" max="255"/><!--brightness-->
      <saturation min="0" max="255"/><!--saturation-->
      <hue min="0" max="255"/><!--hue-->
    </colorValue>
    <colorPicture>
      <picType opt="jpg"/><!--picture type-->
      <picWidth min="100" max="400"/><!--picture width-->
      <picHeight min="100" max="300"/><!--picture height-->
    </colorPicture>
  </ObjectColor>
  <AuxArea><!--auxiliary region-->
    <maxAreaNum><!--number of regions--></maxAreaNum>
    <areaType opt="overlapRegion,bedLocation"/>
    <!--region type, "overlapRegion"-common region, "bedLocation"-bed location-->
  >
  <Region>
    <vertexNum min="3" max="10"/><!--number of region vertexes-->
  </Region>
</AuxArea>
<BaselineScene><!--baseline scene-->
  <operation opt="update"/><!--operation, "update"-->
  <downloadBaselineScenePic><!--download baseline scene picture--></
downloadBaselineScenePic>
</BaselineScene>
  <channelWorkMode opt="independent, master, slave"/>
  <!--channel working mode: "independent"-independent mode, "master"-master
mode, "slave"-slave mode-->
  <SlaveChannel><!--slave channel-->
    <maxChanNum><!--maximum number of slave channels--></maxChanNum>
    <chanType opt="local, remote"/>
    <!--slave channel type: "local"-local device channel, "remote"-remote
device channel-->
    <RemoteChannel><!--remote channel-->
      <devAddress opt="ipv4, ipv6, domainName"/>
      <!--address type: "ipv4"-IPv4, "ipv6"-IPv6, "domainName"-device domain
name-->
      <devPort min="8000" max="65535"/><!--port-->
      <userNameLength min="0" max="32"/><!--username length-->
      <passwordLength min="0" max="16"/> <!--password length-->
      <channelNumber><!--channel No.--></channelNumber>
    </RemoteChannel>
  </SlaveChannel>
  <VcaKeyParam>
    <!--VCA algorithm library keyword parameter, the default type is integer
currently, other types such as float and string need extra APIs-->
    <!--translation suffix explanation: "cn"-Chinese, "en"-English-->
    <VcaKeyParamEntry>
      <index><!--keyword index value--></index>
      <PlainText>
        <translationCn><!--Chinese name--></translationCn>
        <translationEn><!--English name--></translationEn>
      </PlainText>
    </VcaKeyParamEntry>
  </VcaKeyParam>

```

```

        </PlainText>
        <value min="1" max="100" default="50"/><!--value range-->
    </VcaKeyParamEntry>
    <VcaKeyParamEntry>
        <index><!--keyword index value--></index>
        <PlainText>
            <translationCn><!--Chinese name--></translationCn>
            <translationEn><!--English name--></translationEn>
        </PlainText>
        <value min="1" max="100" default="50"/>
    </VcaKeyParamEntry>
    <VcaKeyParamEntry>
        <index><!--keyword index value--></index>
        <PlainText>
            <translationCn><!--Chinese name--></translationCn>
            <translationEn><!--English name--></translationEn>
        </PlainText>
        <value min="1" max="100" default="50"/>
    </VcaKeyParamEntry>
</VcaKeyParam>
<SwitchLamp>
    <enabled opt="true,false"/>
    <triggerMode opt="continue,once"/>
    <!--req, triggering mode: "continuous"-continuous triggering, "single"-
single triggering -->
    <timeInterval min="0" max="3600" default="120"/>
    <!--req, time interval of continuous alarm triggering, unit: second-->
    <uploadPic opt="false,true"/><!--req, upload alarm picture-->
    <AlarmTime>
        <timeSegNum min="2" max="8"/>
        <!--req, number of arming period per day-->
    </AlarmTime>
    <alarmHandleType
opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp"/>
    <!--"monitor"-warning on the monitor, "audio"-audible warning, "center"-
notify surveillance center, "alarmout"-trigger alarm output, "picture"-capture
JPEG picture and send by email, "wirelesslight"-wireless light and audio alarm,
"uploadftp"-capture picture and upload to FTP-->
    <relRecordChan><!--req, alarm triggered channel recording--></relRecordChan>
</SwitchLamp>
<FaceCaptureStatistics><!--req, face capture statistics-->
    <isSupportStartTime>true</isSupportStartTime>
    <reportType opt="daily,weekly,monthly,yearly"/><!--req, report type-->
    <statType opt="age,gender,numberOfPeople"/><!--req, statistics type-->
</FaceCaptureStatistics>
<ColorFilter><!--req, color filter>
    <enable opt="true,false"/>
    <ruleID min="" max=""/>
    <sequenceNo min="" max=""/>
    <PicInfo>
        <picType opt="jpg"/>
        <width>true</width>

```

```
<height>true</height>
<picDataLen min="" max=""/>
</PicInfo>
</ColorFilter>
<HumanRecognition><!--human recognition capability-->
  <MaskRegion><!--shielded region-->
    <maxRegionNum><!--number of regions--></maxRegionNum>
    <Region>
      <vertexNum min="3" max="10"/><!--number of region vertexes-->
    </Region>
  </MaskRegion>
</HumanRecognition>
<AtmCabinKeyParam><!--ATM safety cabin key parameters>
  <VcaKeyParam>
    <VcaKeyParamEntry>
      <index><!--index No. of limitation on the number of people of violent
motion alarm--></index>
      <PlainText>
        <translationCn>limitation on the number of people of violent motion
alarm</translationCn>
        <!--Chinese name-->
        <translationEn>limitation on the number of people of violent motion
alarm</translationEn>
        <!--English name-->
      </PlainText>
      <value min="0" max="32" default="2"/>
    </VcaKeyParamEntry>
    <VcaKeyParamEntry>
      <index><!--index No. of time threshold of tailgating alarm--></index>
      <PlainText>
        <translationCn>time threshold of tailgating alarm</translationCn>
        <!--Chinese name-->
        <translationEn>time threshold of tailgating alarm</translationEn>
        <!--English-->
      </PlainText>
      <value min="1" max="10000" default="1400"/><!--unit: ms-->
    </VcaKeyParamEntry>
    <VcaKeyParamEntry>
      <index><!--index No. of filtering unattended baggage false alarm--></
index>
      <PlainText>
        <translationCn>filtering unattended baggage false alarm</
translationCn>
        <!--Chinese name-->
        <translationEn>filtering unattended baggage false alarm</
translationEn>
        <!--English name-->
      </PlainText>
      <enabled opt="true,false"/>
    </VcaKeyParamEntry>
  </VcaKeyParam>
</AtmCabinKeyParam>
```



```
<AtmSurroundKeyParam><!--ATM environment key parameters>
  <VcaKeyParam>
    <VcaKeyParamEntry>
      <index><!--index No. of filtering unattended baggage false alarm--></
index>
      <PlainText>
        <translationCn>filtering unattended baggage false alarm</
translationCn>
        <!--Chinese name-->
        <translationEn>filtering unattended baggage false alarm</
translationEn>
        <!--English name-->
      </PlainText>
      <enabled opt="true,false"/>
    </VcaKeyParamEntry>
  </VcaKeyParam>
</AtmSurroundKeyParam>
<AtmPanelKeyParam><!--ATM panel key parameters>
  <VcaKeyParam>
    <VcaKeyParamEntry>
      <index><!--index No. of filtering sticking scrip false alarm--></index>
      <PlainText>
        <translationCn>filtering sticking scrip false alarm</translationCn>
        <!--Chinese name-->
        <translationEn>filtering sticking scrip false alarm</translationEn>
        <!--English name-->
      </PlainText>
      <enabled opt="true,false"/>
    </VcaKeyParamEntry>
  </VcaKeyParam>
</AtmPanelKeyParam>
</VcaChanAbility>
```

### C.64 XML\_VideoCap

#### VideoCap message in XML format

```
<VideoCap version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <videoInputPortNums><!--opt, xs: integer--><videoInputPortNums>
  <videoOutputPortNums><!--opt, xs: integer--><videoOutputPortNums>
  <isSupportHeatmap><!--opt, xs: boolean, whether to support heat map function--
></isSupportHeatmap>
  <isSupportCounting><!--opt, xs: boolean--></isSupportCounting>
  <countingType><!--dep, xs: string, "human,object"--></countingType>
  <isSupportPreviewSwitch><!--opt, xs: boolean, whether to support live view
switch--></isSupportPreviewSwitch>
  <isSupportRecodStatus>
    <!--opt, xs: boolean, whether supports getting recording status-->
  </isSupportRecodStatus>
  <isSupportPrivacyMask>
```

```
<!--opt, xs: boolean, whether supports priavte mask function-->
</isSupportPrivacyMask>
<isSupportBinocularPreviewSwitch>
  <!--opt, xs: boolean, whether supports auto-switch during the live view
of dual lens camera-->
</isSupportBinocularPreviewSwitch>
<isSupportCalibCheck>
  <!--opt, xs: boolean, whether supports calibration verification-->
</isSupportCalibCheck>
<isSupportPIP><!--opt, xs: boolean, "true, false"--></isSupportPIP>
<channelFlexible opt="name,enable,online,linknum">
  <!--capability of getting channel status by condition-->
</channelFlexible >
<isSupportFocusVideoMode>
  <!--opt, xs: boolean, video focus mode for installation and debug-->
</isSupportFocusVideoMode>
<isSupportExternalChannel>
  <!--opt, xs: boolean, whether supports extending analog channel-->
</isSupportExternalChannel>
<isSupportMultiChannelCounting>
  <!--opt, xs: boolean, whether supports people counting of multiple
channels-->
</isSupportMultiChannelCounting>
<isSupportCountingCollection>
  <!--opt, xs:boolean, whether supports people counting data replenishment-->
</isSupportCountingCollection>
<isSupportHeatmapCollection>
  <!--opt, xs:boolean, whether supports heat map data replenishment-->
</isSupportHeatmapCollection>
<OSDLanguage opt="GBK,EUC-KR" def="GBK"/>
<isSupportInitLens><!--req, xs:boolean, whether to support initializing lens--
></isSupportInitLens>
  <isSupportOneFocus><!--req, xs:boolean, whether to support one-touch
focusing--></isSupportOneFoucs>
  <notSupportFocus><!--req, xs:boolean, the focus capability is not supported--
></notSupportFoucs>
  <notSupportIris><!--req, xs:boolean, the iris capability is not supported--
></notSupportIris>
  <isSupportCapturePicOverlays><!--opt, xs:boolean, whether to support text
overlay on the captured picture--></isSupportCapturePicOverlays>
  <isSupportMergePicOverlays><!--opt, xs:boolean, whether to support text
overlay on the composite picture--></isSupportMergePicOverlays>
</VideoCap>
```

### C.65 XML\_ZoomRatioCap

ZoomRatioCap message in XML format

```
<ZoomRatioCap version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <isSupportZoomRatioInScene>
```

```
<!--opt, xs:boolean, whether supports zooming ratio configuration of patrol
tracking for multiple scenes. When this node exists and its value is "true", it
indicates support.-->
</isSupportZoomRatioInScene>
<isSupportZoomRatioGoto>
  <!--opt, xs:boolean, whether supports optimizing zooming ratio of patrol
tracking for multiple scenes. When this node exists and its value is "true", it
indicates support.-->
  </isSupportZoomRatioGoto>
</ZoomRatioCap>
```

## Appendix D. Appendixes

### D.1 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.
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.

Error Name	Error Code	Error Description
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.
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.

Error Name	Error Code	Error Description
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.
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.

Error Name	Error Code	Error Description
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.
NET_DVR_BINDSOCKET_ERROR	72	Failed to bind socket.
NET_DVR_SOCKETCLOSE_ERROR	73	Socket disconnected. Network disconnected or the destination is unreachable.

Error Name	Error Code	Error Description
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.
NET_DVR_CALLINEINVALID	92	The height line is too concentrated, or the length line is not inclined enough.



Error Name	Error Code	Error Description
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.
NET_DVR_LOAD_HCPLAYBACK_SDK_ERROR	110	Failed to load the playback component.

Error Name	Error Code	Error Description
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.
NET_DVR_COM_VER_MISMATCH_HCPLAYBACK	139	Playback component version and SDK version mismatched.

Error Name	Error Code	Error Description
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.
NET_DVR_EMAIL_SERVER_NOT_CONFIG_DNS	168	The server uses domain name without configuring DNS, the domain name may be invalid.

Error Name	Error Code	Error Description
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.
NET_ERR_MAX_PORT_MULTIPLEX	183	Maximum number of multiplexed ports reaches.
NET_ERR_CREATE_PORT_MULTIPLEX	184	Creating port multiplier failed.

Error Name	Error Code	Error Description
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.
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.

Error Name	Error Code	Error Description
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.
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.

Error Name	Error Code	Error Description
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).
NET_DVR_LANE_NOT_READY	312	Lane not configured.
NET_DVR_RULE_INCLUDE_TWO_WAY	313	Two different directions are contained in event rule.

Error Name	Error Code	Error Description
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.
NET_DVR_TRIAL_OVERDUE	328	Trial period is ended.
NET_DVR_CONFIG_FILE_CONFLICT	329	Device type and configuration file conflicted.



Error Name	Error Code	Error Description
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.
NET_DVR_IMAGE_RESOLUTION_ERROR	347	Invalid picture resolution

Error Name	Error Code	Error Description
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.
NET_DVR_SPLIT_WINDOW_NOT_SUPPORT_TRANSPARENT	726	For windows whose window division is enabled, transparency configuration is not supported.

Error Name	Error Code	Error Description
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.
NET_SDK_ERR_INQUEST_RESUMING	815	Inquest resuming.
NET_SDK_ERR_RECORD_BACKUPING	816	Inquest video backing up.

Error Name	Error Code	Error Description
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.
NET_DVR_ERR_CALIB_FAIL_PLEASEAGAIN	839	Calibration failed, please try again.

Error Name	Error Code	Error Description
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.
NET_DVR_FACELIB_PIC_IMPORTING_ERROR	859	The pictures is importing to face picture library.

Error Name	Error Code	Error Description
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.
NET_DVR_ERR_URL_DOWNLOAD_NOTSTART	879	URL download has not started.

Error Name	Error Code	Error Description
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.
NET_ERR_OUTPUT_BOARD_DATA_OVERFLOW	913	Output board data overflow
NET_ERR_SAME_USER_NAME	914	Duplicate user name

Error Name	Error Code	Error Description
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.
NET_ERR_NO_ZERO_CHAN	953	The device has no zero channel.
NEED_RECONNECT	955	Need redirection (for transcoding system)



Error Name	Error Code	Error Description
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
XML_ANALYZE_FIND_LOCALXML_ERROR	1002	Failed to find related local xml
XML_ANALYZE_LOAD_LOCALXML_ERROR	1003	Loading local xml error

Error Name	Error Code	Error Description
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.
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

Error Name	Error Code	Error Description
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
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

Error Name	Error Code	Error Description
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_IOABILITY	1413	Lack of IO ability.
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.

Error Name	Error Code	Error Description
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.
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.

Error Name	Error Code	Error Description
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.
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.

Error Name	Error Code	Error Description
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.
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.

Error Name	Error Code	Error Description
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.
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.



Error Name	Error Code	Error Description
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_RADAR_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.
NET_DVR_ERR_BIGCARSIGNSPEED_GREATERTHAN_LIMITSPEED	1614	Big car sign speed limit greater than speed limit value.
NET_DVR_ERR_BIGCARSIGNSPEED_GREATERTHAN_CARSIGNSPEED	1615	Big car sign speed limit is greater than the car sign speed limit value.

Error Name	Error Code	Error Description
NET_DVR_ERR_BIGCARLIMITSPEED_GREATERTHAN_CARLIMITSPEED	1616	Big car speed limit value is greater than the car speed limit value.
NET_DVR_ERR_BIGCARLOWSPEEDLIMIT_GREATERTHAN_CARLOWSPEEDLIMIT	1617	Big car low speed limit value is greater than the car low speed limit value.
NET_DVR_ERR_CARLIMITSPEED_GREATERTHAN_EXCEPHIGHSPEED	1618	Car speed limit greater than exception high speed value.
NET_DVR_ERR_BIGCARLIMITSPEED_GREATERTHAN_EXCEPHIGHSPEED	1619	Big car speed limit greater than exception high speed value.
NET_DVR_ERR_STOPLINE_MORETHAN_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
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

Error Name	Error Code	Error Description
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.
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

Error Name	Error Code	Error Description
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
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.

Error Name	Error Code	Error Description
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
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_PLAYLOAD	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.

Error Name	Error Code	Error Description
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
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.

Error Name	Error Code	Error Description
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.
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

Error Name	Error Code	Error Description
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).
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



Error Name	Error Code	Error Description
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

Error Name	Error Code	Error Description
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_DESCRIBESENDDTIMEOUT	411	Sending "RTSP DESCRIBE" is timeout.
NET_DVR_RTSP_DESCRIBESENDERROR	412	Failed to send "RTSP DESCRIBE".
NET_DVR_RTSP_DESCRIBERECDTIMEOUT	413	Receiving "RTSP DESCRIBE" is timeout.
NET_DVR_RTSP_DESCRIBERECDATALOST	414	Receiving data of "RTSP DESCRIBE" error.
NET_DVR_RTSP_DESCRIBERECDERROR	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_SETUPSENDDTIMEOUT	421	Sending "RTSP SETUP" is timeout.
NET_DVR_RTSP_SETUPSENDERROR	422	Sending "RTSP SETUP" error.
NET_DVR_RTSP_SETUPRECDTIMEOUT	423	Receiving "RTSP SETUP" is timeout.
NET_DVR_RTSP_SETUPRECDATALOST	424	Receiving data of "RTSP SETUP" error.
NET_DVR_RTSP_SETUPRECDERROR	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

Error Name	Error Code	Error Description
		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_PLAYRECVERROR	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_TEARDOWNRECVERROR	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.

Error Name	Error Code	Error Description
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.
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.

Error Name	Error Code	Error Description
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_NOTSUPORT	601	Not support.
NET_AUDIOINTECOM_ERR_ALLOC_MEMERY	602	Memory allocation error.
NET_AUDIOINTECOM_ERR_PARAMETER	603	Parameter error.
NET_AUDIOINTECOM_ERR_CALL_ORDER	604	API calling order error.
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

Error Name	Error Code	Error Description
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.
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.

**D.2 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.
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.



Sub Status Code	Error Code	Description
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)
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.

Sub Status Code	Error Code	Description
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).
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).

Sub Status Code	Error Code	Description
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.
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.

Sub Status Code	Error Code	Description
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.
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.

Sub Status Code	Error Code	Description
fileAlreadyExists	0x40001007	The file already exists.
invalidID	0x40001008	Invalid ID.
backupnodeNotAllowe Log	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.
exportItemsExceedLimi t	0x40001010	No more items can be exported.
noFiles	0x40001011	The file does not exist.
beingExportedByAnoth erUser	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.
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.
unitGetNetworkInfoErr or	0x4000101E	Getting the network information of data analysis server failed
unitSetNetworkInfoErr or	0x4000101F	Setting the network information of data analysis server failed

Sub Status Code	Error Code	Description
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.
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.

Sub Status Code	Error Code	Description
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.
userCannotCreatAdmin	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.
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.

Sub Status Code	Error Code	Description
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.
mgmtConnectFailed	0x40001062	Connecting to system failed. Check the network port.
mgmtLoginFailed	0x40001063	Logging in to system failed. Check the user name and password.
TDACConnectFailed	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.



Sub Status Code	Error Code	Description
IDNotExist	0x40001068	The server ID does not exist.
serviceNumberReachedLimit	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.
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.
cloudAlalyseOnline	0x4000107B	The cloud analytic server is online.
virIP&mainIPnotSameNetSegment	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.

Sub Status Code	Error Code	Description
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 blacklist library can be armed.
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.

Sub Status Code	Error Code	Description
searchByPictureTimed Out	0x40001094	Searching picture by picture timeout. Search again.
comparisonTimeRange Error	0x40001095	Comparison time period error.
selectMonitorNumber UpperLimit	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.
1V1FacePictureComparisonFailed	0x400010A1	Comparison 1 VS 1 face picture failed.
libraryArmed	0x400010A2	The blacklist 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.

Sub Status Code	Error Code	Description
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.
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.

Sub Status Code	Error Code	Description
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.
getBigDataCloudAnalysisFailed	0x400010CD	Getting big data cloud analytic information failed.
setBigDataCloudAnalysisFailed	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.

Sub Status Code	Error Code	Description
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.
bigDataServerConnect Failed	0x400010D8	Connecting to big data server failed.
microVideoCloudRequ estInfoBuildFailed	0x400010D9	Adding response information of micro video cloud failed.
microVideoCloudRespo nseInfoBuildFailed	0x400010DA	Parsing response information of micro video cloud failed.
transcodingServerRequ estInfoBuildFailed	0x400010DB	Adding response information of transcoding server failed.
transcodingServerResp onseInfoParseFailed	0x400010DC	Parsing response information of transcoding server failed.
transcodingServerOffli ne	0x400010DD	Transcoding server is offline.
microVideoCloudOfflin e	0x400010DE	Micro video cloud is offline.
UPSServerOffline	0x400010DF	UPS monitor server is offline.
statisticReportRequestI nfoBuildFailed	0x400010E0	Adding response information of statistics report failed.
statisticReportRespons eInfoParseFailed	0x400010E1	Parsing response information of statistics report failed.
DisplayConfigInfoBuild Failed	0x400010E2	Adding display configuration information failed.
DisplayConfigInfoParse Failed	0x400010E3	Parsing display configuration information failed.
DisplayConfigInfoSaveF ailed	0x400010E4	Saving display configuration information failed.
notSupportDisplayConf igType	0x400010E5	The display configuration type is not supported.
passError	0x400010E7	Incorrect password.

Sub Status Code	Error Code	Description
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.
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.

Sub Status Code	Error Code	Description
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.
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.



Sub Status Code	Error Code	Description
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.
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.

Sub Status Code	Error Code	Description
managementNodeDoesNotSupportThisOperation	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.
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.

Sub Status Code	Error Code	Description
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.
editingStreamingTypeIsNotSupported	0x40001140	Editing streaming type is not supported.
switchingScheduledTaskToTemporaryTaskIsNotSupported	0x40001141	Switching scheduled task to temporary task is not supported.
switchingTemporaryTaskToScheduledTaskIsNotSupported	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.

Sub Status Code	Error Code	Description
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.
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.

Sub Status Code	Error Code	Description
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.
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.

Sub Status Code	Error Code	Description
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.
gettingDetailsOfSearchedHigh-frequencyPeopleDetectionLogsFailed.	0x40001174	Getting the search result details of frequently appeared person alarms failed.
theArmedCamerasAlreadyExistInTheControlCenter	0x40001175	Some cameras in control center are already armed.
disarmingFailedTheCamerasNotArmed	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.

Sub Status Code	Error Code	Description
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.
frequentlyAppearedPersonAlarmIsAlreadyConfiguredForListLibrary	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.
settingAlgorithmTypeForDeviceFailed	0x40001187	Allocating task resource failed.
gettingHighFrequencyPersonDetectionAlarmInformationFailed	0x40001188	Setting frequently appeared person alarm failed.
invalidSearchCondition	0x40001189	Invalid result.
theTaskIsNotCompleted	0x4000118B	The task is not completed.
resourceOverRemainLimit	0x4000118C	No more resource can be pre-allocated.
frequentlyAppearedPersonAlarmIsAlreadyConfiguredForTheCameraDisarmFirstAndTryAgain	0x4000118D	The frequently appeared person alarm of this camera is configured. Delete the arming information and try again.

Sub Status Code	Error Code	Description
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.
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.
polygonNotAllowCrossing	0x400015C2	Polygons are not allowed to cross.



Sub Status Code	Error Code	Description
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.
MaximumNumberOfDevices	0x400019F1	The maximum number of devices reached.
StandbyMachineCannotBeDeleted	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.

Sub Status Code	Error Code	Description
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.
audiosPlayingPleaseWait	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.

Sub Status Code	Error Code	Description
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.
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.

Sub Status Code	Error Code	Description
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.
masterSlavelsEnable	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?
radarMoudleConnectFail	0x40008033	Radar module communication failed.

Sub Status Code	Error Code	Description
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 (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.

Sub Status Code	Error Code	Description
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.
connectServerFail	0x60000009	Connecting to server failed.

Sub Status Code	Error Code	Description
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.
incorrentUserNameOrPassword	0x60000012	Incorrect user name or password.
invalidStoragePoolOfCloudServer	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.
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.

Sub Status Code	Error Code	Description
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
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.



Sub Status Code	Error Code	Description
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.
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.

Sub Status Code	Error Code	Description
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.
humanMisInfoFilterEnabledChanNumError	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.
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

Sub Status Code	Error Code	Description
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.
AdaptiveStreamNotEnabled	0x60002001	The stream self-adaptive function is not enabled.
AdaptiveStreamAndVariableBitrateEnabled	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).

Sub Status Code	Error Code	Description
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.
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.

Sub Status Code	Error Code	Description
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 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.

