



Device Network SDK (Thermal)

Developer Guide

Legal Information

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

Contents

Chapter 1 Overview	1
1.1 Introduction	1
1.2 Update History	1
Chapter 2 Manually Measure Temperature	13
Chapter 3 Measure Real-Time Temperature	21
Chapter 4 Alarm/Event Configuration	25
4.1 Configure Temperature Screening Alarm	25
4.2 Configure Temperature Measurement Alarm in Normal Mode	30
4.3 Configure Temperature Measurement Alarm in Expert Mode	35
4.4 Configure Fire and Smoke Alarm	40
4.5 Configure Ship Flow Detection Alarm	44
4.6 Configure Dredger Detection Alarm	49
Chapter 5 Alarm/Event Receiving	55
5.1 Receive Alarm/Event in Arming Mode	55
5.2 Receive Alarm/Event in Listening Mode	58
5.3 Subscribe Alarm/Event in Arming Mode	61
Chapter 6 Other Thermal Applications	64
Chapter 7 API Reference	69
7.1 NET_DVR_CaptureJPEGPicture_WithAppendData	69
7.2 NET_DVR_Cleanup	69
7.3 NET_DVR_CloseAlarmChan_V30	70
7.4 NET_DVR_GetAlarmSubscribe	70
7.5 NET_DVR_GetLastError	71
7.6 NET_DVR_GetErrorMsg	71
7.7 NET_DVR_GetDeviceAbility	72
7.8 NET_DVR_GetDVRConfig	72

7.9 NET_DVR_GetNextRemoteConfig	74
7.10 NET_DVR_GetSTDAbility	75
7.11 NET_DVR_GetSTDConfig	75
7.12 NET_DVR_Init	76
7.13 NET_DVR_Login_V40	77
7.14 NET_DVR_Logout	77
7.15 NET_DVR_PTZPreset_Other	78
7.16 NET_DVR_SetAlarmSubscribe	79
7.17 NET_DVR_SetDVRConfig	80
7.18 NET_DVR_SetSDKInitCfg	80
7.19 NET_DVR_SetSTDConfig	82
7.20 NET_DVR_STDXMLConfig	83
7.21 NET_DVR_SetDVRMessageCallBack_V50	84
7.21.1 MSGCallBack	91
7.22 NET_DVR_SetupAlarmChan_V50	92
7.23 NET_DVR_StartListen_V30	93
7.24 NET_DVR_StartRemoteConfig	94
7.24.1 fRemoteConfigCallback	95
7.25 NET_DVR_StopListen_V30	96
7.26 NET_DVR_StopRemoteConfig	96
Appendix A. Data Structure	98
A.1 NET_ALARM_CVR_SUBINFO_UNION	98
A.2 NET_ALARM_RECORD_EXCEPTION	98
A.3 NET_ALARM_RECORDFILE_LOSS	99
A.4 NET_ALARM_RESOURCE_USAGE	99
A.5 NET_ALARM_STREAM_EXCEPTION	100
A.6 NET_DVR_ALARM_FIXED_HEADER	100
A.7 NET_DVR_ALARM_ISAPI_INFO	103

A.8 NET_DVR_ALARM_ISAPI_PICDATA	104
A.9 NET_DVR_ALARMINFO_DEV	105
A.10 NET_DVR_ALARMINFO_DEV_V40	105
A.11 NET_DVR_ALARMINFO_V30	106
A.12 NET_DVR_ALARMINFO_V40	107
A.13 NET_DVR_ALARMER	108
A.14 NET_DVR_ALARMSTRATEGY_PARAM	109
A.15 NET_DVR_CRUISECHAN_INFO	110
A.16 NET_DVR_DEVICEINFO_V30	110
A.17 NET_DVR_DEVICEINFO_V40	114
A.18 NET_DVR_DPC_PARAM	116
A.19 NET_DVR_EVENT_SCHEDULE	118
A.20 NET_DVR_ETHERNET_V30	118
A.21 NET_DVR_EVENT_TRIGGER	119
A.22 NET_DVR_FACE_THERMOMETRY_ALARM	119
A.23 NET_DVR_FIREDETECTION_ALARM	121
A.24 NET_DVR_FIREDETECTION_CFG	123
A.25 NET_DVR_FOCUSMODE_CFG	124
A.26 NET_DVR_HANDLEEXCEPTION_V41	125
A.27 NET_DVR_INIT_CFG_ABILITY	126
A.28 NET_DVR_IPADDR_UNION	127
A.29 NET_DVR_JPEGPICTURE_WITH_APPENDDATA	127
A.30 NET_DVR_LINEPOLYGON_THERM_CFG	128
A.31 NET_DVR_LLI_PARAM	129
A.32 NET_DVR_LLPOS_PARAM	129
A.33 NET_DVR_LOCAL_SDK_PATH	129
A.34 NET_DVR_MIME_UNIT	130
A.35 NET_DVR_NETCFG_V50	130

A.36 NET_DVR_POINT_THERM_CFG	132
A.37 NET_DVR_PPPOECFG	132
A.38 NET_DVR_PRESETCHAN_INFO	133
A.39 NET_DVR_PTZTRACKCHAN_INFO	133
A.40 NET_DVR_REALTIME_THERMOMETRY_COND	133
A.41 NET_DVR_SCHEDTIME	134
A.42 NET_DVR_SETUPALARM_PARAM_V50	135
A.43 NET_DVR_SHIPSDTECTION_ALARM	138
A.44 NET_DVR_SHIPSINFO	139
A.45 NET_DVR_SMOKEDTECTION_ALARM	140
A.46 NET_DVR_SMOKEDTECTION_CFG	140
A.47 NET_DVR_STD_ABILITY	140
A.48 NET_DVR_STD_CONFIG	141
A.49 NET_DVR_TEMPERATURE_COLOR	142
A.50 NET_DVR_TIME	143
A.51 NET_DVR_TIME_EX	143
A.52 NET_DVR_THERMAL_PIP	144
A.53 NET_DVR_THERMINTELL_PARAM	145
A.54 NET_DVR_THERMOMETRY_ALARM	145
A.55 NET_DVR_THERMOMETRY_ALARMRULE	147
A.56 NET_DVR_THERMOMETRY_ALARMRULE_PARAM	147
A.57 NET_DVR_THERMOMETRY_BASICPARAM	148
A.58 NET_DVR_THERMOMETRY_COND	151
A.59 NET_DVR_THERMOMETRY_DIFF_ALARM	152
A.60 NET_DVR_THERMOMETRY_DIFFCOMPARISON	153
A.61 NET_DVR_THERMOMETRY_DIFFCOMPARISON_PARAM	153
A.62 NET_DVR_THERMOMETRY_MODE	154
A.63 NET_DVR_THERMOMETRY_PRESETINFO	155

A.64 NET_DVR_THERMOMETRY_PRESETINFO_PARAM	155
A.65 NET_DVR_THERMOMETRYRULE_TEMPERATURE_INFO	157
A.66 NET_DVR_THERMOMETRY_TRIGGER_COND	157
A.67 NET_DVR_THERMOMETRY_UPLOAD	157
A.68 NET_DVR_USER_LOGIN_INFO	159
A.68.1 fLoginResultCallBack	161
A.69 NET_DVR_XML_CONFIG_INPUT	161
A.70 NET_DVR_XML_CONFIG_OUTPUT	162
A.71 NET_PTZ_INFO	162
A.72 NET_SDK_CALLBACK_STATUS_NORMAL	163
A.73 NET_SDK_MANUAL_THERMOMETRY	163
A.74 NET_SDK_MANUALTHERM_BASICPARAM	164
A.75 NET_SDK_MANUALTHERM_RULE	164
A.76 NET_SDK_POINT_THERMOMETRY	165
A.77 NET_SDK_REGION_THERMOMETRY	165
A.78 NET_VCA_DEV_INFO	166
A.79 NET_VCA_POINT	167
A.80 NET_VCA_POLYGON	167
A.81 NET_VCA_RECT	168
Appendix B. Request URIs	169
B.1 /ISAPI/Event/notification/subscribeEvent	169
B.2 /ISAPI/Event/notification/subscribeEvent/<ID>	169
B.3 /ISAPI/Event/notification/subscribeEventCap	170
B.4 /ISAPI/Event/notification/unSubscribeEvent	170
B.5 /ISAPI/Event/schedules/dredgerDetection/<ID>/scene/<SID>	171
B.6 /ISAPI/Event/schedules/TMPA	171
B.7 /ISAPI/Event/schedules/TMPA/<ID>	172
B.8 /ISAPI/Event/schedules/shipsFlowDetection/<ID>/scene/<SID>	173

B.9 /ISAPI/Event/triggersCap	174
B.10 /ISAPI/Event/triggers/<eventType>-<ID>	174
B.11 /ISAPI/Event/triggers/dredgerDetection/<ID>/scene/<SID>	175
B.12 /ISAPI/Event/triggers/shipsFlowDetection/<ID>/scene/<SID>	176
B.13 /ISAPI/System/capabilities	176
B.14 /ISAPI/Thermal/capabilities	177
B.15 /ISAPI/Thermal/channels/<ID>/blackBody	177
B.16 /ISAPI/Thermal/channels/<ID>/blackBody/capabilities	178
B.17 /ISAPI/Thermal/channels/<ID>/burningPrevention	178
B.18 /ISAPI/Thermal/channels/<ID>/burningPrevention/capabilities	179
B.19 /ISAPI/Thermal/channels/<ID>/faceThermometry	179
B.20 /ISAPI/Thermal/channels/<ID>/faceThermometry/capabilities	180
B.21 /ISAPI/Thermal/channels/<ID>/faceThermometry/regions	180
B.22 /ISAPI/Thermal/channels/<ID>/faceThermometry/regions/<ID>	181
B.23 /ISAPI/Thermal/channels/<ID>/faceThermometry/regions/<ID>/detectionInfo	182
B.24 /ISAPI/Thermal/channels/<ID>/fireDetection	182
B.25 /ISAPI/Thermal/channels/<ID>/fireDetection/advanceParam/capabilities?format=json	183
B.26 /ISAPI/Thermal/channels/<ID>/fireDetection/advanceParam?format=json	184
B.27 /ISAPI/Thermal/channels/<ID>/fireDetection/capabilities	184
B.28 /ISAPI/Thermal/channels/<ID>/historyTemperature	185
B.29 /ISAPI/Thermal/channels/<ID>/historyTemperature/capabilities	185
B.30 /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/capabilities?format=json	185
B.31 /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/captureRatio/capabilities? format=json	186
B.32 /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/captureRatio?format=json	187
B.33 /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/dredgerDetection/regions/<ID>? format=json	187

B.34 /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/dredgerDetection/regions/capabilities?format=json	188
B.35 /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/dredgerDetection/regions?format=json	189
B.36 /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/goto?format=json	189
B.37 /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/resetCount?format=json	190
B.38 /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/shipFlowDetection/regions/<ID>?format=json	190
B.39 /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/shipFlowDetection/regions/capabilities?format=json	191
B.40 /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/shipFlowDetection/regions?format=json	192
B.41 /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/shipsDetectionCount?format=json	193
B.42 /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/shipsFunctionMode/capabilities?format=json	193
B.43 /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/shipsFunctionMode?format=json	194
B.44 /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>?format=json	194
B.45 /ISAPI/Thermal/channels/<ID>/shipsDetection/advanceParam/capabilities?format=json	195
B.46 /ISAPI/Thermal/channels/<ID>/shipsDetection/advanceParam?format=json	196
B.47 /ISAPI/Thermal/channels/<ID>/shipsDetection/basicParam/capabilities?format=json ..	197
B.48 /ISAPI/Thermal/channels/<ID>/shipsDetection/basicParam?format=json	197
B.49 /ISAPI/Thermal/channels/<ID>/shipsDetection/capabilities?format=json	198
B.50 /ISAPI/Thermal/channels/<ID>/shipsDetection/sceneTrace/capabilities?format=json ..	198
B.51 /ISAPI/Thermal/channels/<ID>/shipsDetection/sceneTrace?format=json	199
B.52 /ISAPI/Thermal/channels/<ID>/shipsDetection?format=json	200
B.53 /ISAPI/Thermal/channels/<ID>/thermometry/<SID>/capabilities	201
B.54 /ISAPI/Thermal/channels/<ID>/thermometry/<SID>/alarmRules/temperatureIntervalMeasurement/capabilities?format=json	201

B.55 /ISAPI/Thermal/channels/<ID>/thermometry/<SID>/alarmRules/ temperatureIntervalMeasurement?format=json	202
B.56 /ISAPI/Thermal/channels/<ID>/thermometry/basicParam	202
B.57 /ISAPI/Thermal/channels/<ID>/thermometry/basicParam/capabilities	203
B.58 /ISAPI/Thermal/channels/<ID>/thermometry/pixelToPixelParam	204
B.59 /ISAPI/Thermal/channels/<ID>/thermometry/pixelToPixelParam/capabilities	204
B.60 /ISAPI/Thermal/channels/<ID>/thermometry/OffLineCapture?format=json	205
B.61 /ISAPI/Thermal/channels/<ID>/temperatureCorrect/capabilities?format=json	205
B.62 /ISAPI/Thermal/channels/<ID>/temperatureCorrect?format=json	206
B.63 /ISAPI/Thermal/Power	206
B.64 /ISAPI/Thermal/Power/capabilities	207
Appendix C. Appendixes	208
C.1 Request and Response Messages	208
C.1.1 JSON_advanceParam	208
C.1.2 JSON_advanceParamCap	208
C.1.3 JSON_basicParam	208
C.1.4 JSON_basicParamCap	209
C.1.5 JSON_Cap_functionMode	209
C.1.6 JSON_DredgerDetectionRule	210
C.1.7 JSON_DredgerDetectionRuleList	210
C.1.8 JSON_DredgerDetectionRuleListCap	211
C.1.9 JSON_EventNotificationAlert_Alarm/EventInfo	212
C.1.10 JSON_EventNotificationAlert_dredgerDetection	213
C.1.11 JSON_EventNotificationAlert_temperatureIntervalMeasurementMsg	215
C.1.12 JSON_fireDetection_AdvanceParam	218
C.1.13 JSON_fireDetection_AdvanceParamCap	218
C.1.14 JSON_functionMode	219
C.1.15 JSON_ResponseStatus	219

C.1.16 JSON_ShipDetectionCount	219
C.1.17 JSON_ShipFlowDetectionRule	220
C.1.18 JSON_ShipFlowDetectionRuleList	221
C.1.19 JSON_ShipFlowDetectionRuleListCap	222
C.1.20 JSON_shipsDetectionCap	223
C.1.21 JSON_ShipsDetectionCaptureRatio	224
C.1.22 JSON_ShipsDetectionCaptureRatioCap	224
C.1.23 JSON_ShipsDetectionScene	224
C.1.24 JSON_ShipsDetectionSceneCap	225
C.1.25 JSON_ShipsDetectionSceneList	226
C.1.26 JSON_ShipsDetectionSceneTraceList	227
C.1.27 JSON_ShipsDetectionTraceListCap	228
C.1.28 JSON_thermometryOffLineCapture	229
C.1.29 JSON_TemperatureCorrect	229
C.1.30 JSON_TemperatureCorrectCap	230
C.1.31 JSON_TemperatureIntervalMeasurementAlarmRuleCap	231
C.1.32 JSON_TemperatureIntervalMeasurementAlarmRule	232
C.1.33 XML_BurningPrevention	233
C.1.34 XML_BurningPreventionCap	234
C.1.35 XML_CAMERAPARA	235
C.1.36 XML_Cap_FaceThermometry	253
C.1.37 XML_Cap_FireDetection	256
C.1.38 XML_Cap_ManualThermBasic	257
C.1.39 XML_Cap_ManualThermometry	258
C.1.40 XML_Cap_Power	260
C.1.41 XML_Cap_ThermalBlackBody	260
C.1.42 XML_Cap_ThermIntell	261
C.1.43 XML_Cap_ThermalPip	261

C.1.44 XML_Cap_ThermometryAlarmRule	262
C.1.45 XML_Cap_ThermometryBasicParam	264
C.1.46 XML_Cap_ThermometryMode	267
C.1.47 XML_Cap_ThermometryScene	268
C.1.48 XML_DeviceCap	270
C.1.49 XML_EventNotificationAlert_AlarmEventInfo	279
C.1.50 XML_EventNotificationAlert_SubscriptionHeartbeat	280
C.1.51 XML_EventTrigger	280
C.1.52 XML_EventTriggerNotification	281
C.1.53 XML_EventTriggerNotificationList	282
C.1.54 XML_EventTriggersCap	282
C.1.55 XML_EventTriggerCapType	285
C.1.56 XML_FaceThermDetectionInfo	286
C.1.57 XML_FaceThermometry	288
C.1.58 XML_FaceThermometryRegionList	289
C.1.59 XML_FireDetection	289
C.1.60 XML_HistoryTemperatureCap	290
C.1.61 XML_HistoryTemperatureDescription	291
C.1.62 XML_HistoryTemperatureResult	291
C.1.63 XML_PixelToPixelParam	291
C.1.64 XML_PixelToPixelParamCap	292
C.1.65 XML_Power	293
C.1.66 XML_Schedule	293
C.1.67 XML_SubscribeEventResponse	294
C.1.68 XML_SubscribeEventCap	295
C.1.69 XML_SubscribeEvent	297
C.1.70 XML_ThermalBlackBody	298
C.1.71 XML_ThermalCap	298

C.1.72 XML_ThermIntell	301
C.1.73 XML_ThermometryAlarmRule	301
C.1.74 XML_ThermometryBasicParam	302
C.1.75 XML_ThermometryMode	305
C.1.76 XML_ThermometryRegion	305
C.1.77 XML_TMPAScheduleList	307
C.1.78 XML_ResponseStatus	307
C.2 Device Network SDK Errors	308
C.3 Response Codes of Text Protocol	350

Chapter 1 Overview

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

1.1 Introduction

Thermal application developed by ISAPI protocol contains temperature screening and temperature measurement, which is mainly for thermal products. You can configure the temperature screening alarm and thermography alarm (including temperature alarm and temperature difference alarm) and receive the alarms if triggered.

1.2 Update History

The update history shows the summary of changes in HCNetSDK with different versions, the related products or projects, and the updated time (which is also the build No. of HCNetSDK).

Summary of Changes in Version 6.1.5.26_Nov., 2020

Related Product: DS-2TA21-2AVF, DS-2TA21-3AVF, DS-2TB21-3AVF, DS-2TB21-3AVF/IN, DS-2TD3017T-2/V, and DS-2TD3017T03/V Thermographic Cube Camera with Software Version 5.5.12

1. Extended the structure about real-time temperature measurement condition
NET_DVR_REALTIME_THERMOMETRY_COND (related API: **NET_DVR_StartRemoteConfig**): added one member **fTemperatureDiff** (temperature difference).
2. Extended the capability of temperature measurement basic parameters
XML_Cap_ThermometryBasicParam and temperature measurement basic parameters **XML_ThermometryBasicParam** (related API: **NET_DVR_STDXMLConfig**; related URIs: **/ISAPI/Thermal/channels/<ID>/thermometry/basicParam** and **/ISAPI/Thermal/channels/<ID>/thermometry/basicParam/capabilities**):
extended the value of node **<temperatureRange>** (temperature range): "100-550";
added three nodes: **<alarmMode>** (alarm mode), **<NormalRulesColor>** (normal rule color), and **<NormalTemperatureIntervalMeasurement>** (normal interval temperature measurement).
3. Added URIs for configuring rules of interval temperature measurement alarm (related API: **NET_DVR_STDXMLConfig**):
Get capability: GET **/ISAPI/Thermal/channels/<ID>/thermometry/<SID>/alarmRules/temperatureIntervalMeasurement/capabilities?format=json** ;
Get or set rules of interval temperature measurement alarm: GET or PUT **/ISAPI/Thermal/channels/<ID>/thermometry/<SID>/alarmRules/temperatureIntervalMeasurement?format=json** .

4. Extended configuration capability of pixel-to-pixel temperature measurement parameters ***XML_PixelToPixelParamCap*** and pixel-to-pixel temperature measurement parameters ***XML_PixelToPixelParam*** (related API: ***NET_DVR_STDXMLConfig*** ; related URIs: ***/ISAPI/Thermal/channels/<ID>/thermometry/pixelToPixelParam/capabilities*** and ***/ISAPI/Thermal/channels/<ID>/thermometry/pixelToPixelParam***):
added three nodes **<rulesOverlayEnabled>** (whether to enable thermography rule overlay on picture), **<visiblePicResolution>** (visible light picture resolution), and **<thermalPicResolution>** (thermal picture resolution).
5. Extended thermal capability ***XML_ThermalCap*** (related API: ***NET_DVR_STDXMLConfig*** ; related URI: ***/ISAPI/Thermal/capabilities***):
added one node **<isSupportTemperatureIntervalMeasurement>** (whether device supports interval temperature measurement).
6. Added one event type "temperatureIntervalMeasurement" (interval temperature measurement), see ***JSON_EventNotificationAlert_temperatureIntervalMeasurementMsg*** for details.

Summary of Changes in Version 6.1.5.25_Sept., 2020

Related Product: DS-2TA03-15SVI, DS-2TA06-25SVI, and DS-2TD2036T-7/V Thermographic Automation Camera with Software Version 5.5.8

1. Extended the capability of temperature measurement basic parameters ***XML_Cap_ThermometryBasicParam*** and temperature measurement basic parameters ***XML_ThermometryBasicParam*** (related URIs: ***/ISAPI/Thermal/channels/<ID>/thermometry/basicParam/capabilities*** and ***/ISAPI/Thermal/channels/<ID>/thermometry/basicParam*** ; related API: ***NET_DVR_STDXMLConfig***):
added three nodes: **<calibrationFileVersion>** (calibration file version information), **<alarmInterval>** (temperature measurement interval), and **<rulesOverlayMode>** (rule overlay mode).
2. Extended the capability of pixel-to-pixel temperature measurement ***XML_PixelToPixelParamCap*** (related URI: ***/ISAPI/Thermal/channels/<ID>/thermometry/pixelToPixelParam/capabilities*** ; related API: ***NET_DVR_STDXMLConfig***):
added two sub nodes **<visiblePicEnabled>** (whether the thermal camera returns visible light picture) and **<captureMode>** (image capture mode) to node **<JpegPictureWithAppendData>**.
3. Extended the pixel-to-pixel temperature measurement parameters ***XML_PixelToPixelParam*** (related URI: ***/ISAPI/Thermal/channels/<ID>/thermometry/pixelToPixelParam*** ; related API: ***NET_DVR_STDXMLConfig***):
added one sub node **<visiblePicEnabled>** (whether the thermal camera returns visible light picture) to node **<JpegPictureWithAppendData>**.

Summary of Changes in Version 6.1.4.20_June, 2020

Related Product: DS-2TD1217B series and DS-2TD2636B series Thermal & Optical Bi-spectrum Network Camera

1. Extended structure about temperature measurement alarm information
NET_DVR_THERMOMETRY_ALARM (related API: **NET_DVR_SetDVRMessageCallBack_V50** with command 0x5212-"COMM_THERMOMETRY_ALARM"):
added four members: **dwVisibleChannel** (visible light channel No.), **dwRelativeTime** (relative time), **dwAbsTime** (absolute time), and **fAlarmRuleTemperature** (rule temperature of temperature measurement alarm) by reserved 16 bytes.
2. Extended the configuration capability of temperature measurement basic parameters
XML_Cap_ThermometryBasicParam (related API: **NET_DVR_STDXMLConfig** ; related URI: GET /**ISAPI/Thermal/channels/<ID>/thermometry/basicParam/capabilities**) and temperature measurement basic parameters **XML_ThermometryBasicParam** (related API: **NET_DVR_STDXMLConfig** ; related URI: **/ISAPI/Thermal/channels/<ID>/thermometry/basicParam**):
added two nodes **<distanceMode>** (distance mode) and **<faceTemperatureInfoUploadEnabled>** (whether to enable uploading face temperature information).
3. Extended the structure of real-time temperature information
NET_DVR_THERMOMETRY_UPLOAD (related API: **NET_DVR_StartRemoteConfig** with command of 3629-"NET_DVR_GET_REALTIME_THERMOMETRY"):
added two members **byFaceSnapThermometryEnabled** (whether to enable uploading captured face picture with temperature information) and **struFaceRect** (face thumbnail rectangle).
4. Extended the temperature screening configuration capability **XML_Cap_FaceThermometry** (related API: **NET_DVR_STDXMLConfig** ; related URI: GET **/ISAPI/Thermal/channels/<ID>/faceThermometry/capabilities**):
added one sub node **<MaxPupilParam>** (maximum pupil distance) to **<ThermometryRegion>** of **<FaceThermometryRegionList>**;
added four nodes: **<faceSnapUploadEnabled>** (whether to enable uploading captured face picture), **<maxTemperatureCoordinatesEnabled>** (whether to enable displaying the maximum temperature position), **<faceRectShowEnabled>** (whether to enable displaying a frame on the target person), and **<faceTemperatureShowEnabled>** (whether to enable displaying face temperature).
5. Extended the temperature screening parameters **XML_FaceThermometry** (related API: **NET_DVR_STDXMLConfig** ; related URI: **/ISAPI/Thermal/channels/<ID>/faceThermometry**):
added four nodes: **<faceSnapUploadEnabled>** (whether to enable uploading captured face picture), **<maxTemperatureCoordinatesEnabled>** (whether to enable displaying the maximum temperature position), **<faceRectShowEnabled>** (whether to enable displaying a frame on the target person), and **<faceTemperatureShowEnabled>** (whether to enable displaying face temperature).
6. Extended the alarm linkage capabilities **XML_EventTriggersCap** (related API: **NET_DVR_STDXMLConfig** ; related URI: GET **/ISAPI/Event/triggersCap**):
added two nodes **<NoMaskDetectionTriggerCap>** (alarm linkage capability of no wearing mask detection) and **<TMPATriggerCap>** (alarm linkage capability of temperature measurement pre-alarm).
7. Extended the linkage parameter message **XML_EventTrigger** (related API: **NET_DVR_STDXMLConfig** ; related URI: **/ISAPI/Event/triggers/<eventType>-<ID>**):

- added two event type "noMaskDetection" (no wearing mask detection) and "TMPA" (temperature measurement pre-alarm) to the node **<eventType>**.
8. Extended the black body configuration capability **XML_Cap_ThermalBlackBody** (related API: **NET_DVR_STDXMLConfig** ; related URI: GET **/ISAPI/Thermal/channels/<ID>/blackBody/capabilities**) and black body configurations **XML_ThermalBlackBody** (related API: **NET_DVR_STDXMLConfig** ; related URI: **/ISAPI/Thermal/channels/<ID>/blackBody**) :
added two nodes **<enabled>** (whether to enable black body) and **<BlackBodyReigon>** (black body detection area).
 9. Added two URIs of configuring the arming schedule for temperature measurement pre-alarm (related API: **NET_DVR_STDXMLConfig**) :
Get or set the arming schedule of temperature measurement pre-alarm for a specified channel: GET or PUT **/ISAPI/Event/schedules/TMPA/<ID>** ;
Get or set arming schedules of temperature measurement pre-alarm for all channels in a batch: GET or PUT **/ISAPI/Event/schedules/TMPA** .
 10. Extended the temperature screening rule configurations of a specific detection region **XML_ThermometryRegion** (related API: **NET_DVR_STDXMLConfig** ; related URI: **/ISAPI/Thermal/channels/<ID>/faceThermometry/regions/<ID>**) :
added one node **<MaxPupilParam>** (maximum pupil distance).
 11. Extended the thermal capability **XML_ThermalCap** (related API: **NET_DVR_STDXMLConfig** ; related URI: GET **/ISAPI/Thermal/capabilities**) :
added one node **<isSupportFaceSnapThermometry>** (whether device supports uploading captured face picture with temperature information).

Summary of Changes in Version 6.1.4.15_May, 2020

Related Product: TB-4117-3-S Temperature Screening Thermal Camera Module with Software Version 5.5.2

1. Extended the thermal capability **XML_ThermalCap** (related API: **NET_DVR_STDXMLConfig** ; related URI: GET **/ISAPI/Thermal/capabilities**) :
added one node **<isSupportThermalTemperatureCorrect>** (whether device supports temperature calibration).
2. Added a function of temperature calibration (related API: **NET_DVR_STDXMLConfig**) :
Get temperature calibration configuration capability: GET **/ISAPI/Thermal/channels/<ID>/temperatureCorrect/capabilities?format=json** ;
Get or set temperature calibration configuration parameters: GET or POST **/ISAPI/Thermal/channels/<ID>/temperatureCorrect?format=json** .

Summary of Changes in Version 6.1.4.10_Feb., 2020

Related Product: Thermal & Optical Bi-spectrum Network Speed Dome; Thermal & Optical Bi-spectrum Positioning System; Thermal & Optical Bi-spectrum Network Stable PTZ Camera in Version 5.5.25

Added URIs of configuring fire detection advanced parameters: (related API: **NET_DVR_STDXMLConfig**) :

Get capability: GET */ISAPI/Thermal/channels/<ID>/fireDetection/advanceParam/capabilities?format=json* ;

Get or set advanced parameters of fire detection: GET or PUT */ISAPI/Thermal/channels/<ID>/fireDetection/advanceParam?format=json* .

Summary of Changes in Version 6.1.0.X_Nov., 2019

Related Product: DS-2TD11 Series, DS-2TD12 Series, DS-2TD21 Series, and DS-2TD16 Series Thermal Network Bullet Camera in Version 5.5.24

1. Extended the preset (scene) configuration capability **XML_Cap_ThermometryScene** (related API: **NET_DVR_STDXMLConfig** ; related URI: */ISAPI/Thermal/channels/<ID>/thermometry/<SID>/capabilities*):
added a sub node **<RegionBoundary>** (rule region boundary) to the node **<ThermometryRegion>** of **<ThermometryRegionList>**.
2. Extended the error codes in **Device Network SDK Errors** :
added one error code "3013"-NET_DVR_RULE_SHIELDMASK_CONFLICT_ERROR (the rule region conflicts with the shielded area).

Summary of Changes in Version 6.1.0.X_Aug., 2019

Related Product Type: DS-2TA03-7AVI, DS-2TA03-15VI, DS-2TA03-15SVI, DS-2TA06-25SVI, DS-2TA06-25VI, and DS-2TA03-25SVI Thermographic Automation Thermal Camera in Version 2.2

1. Edited the burning prevention capability message **XML_BurningPreventionCap** (related API: **NET_DVR_STDXMLConfig** ; related URI: GET */ISAPI/Thermal/channels/<ID>/burningPrevention/capabilities*):
changed the default value of node **<closedDuration>** (closed status duration) to "10".

Summary of Changes in Version 6.1.0.25_July, 2019

Related Product: Thermographic Cube Camera in Version 1.0

1. Extended the capability of picture-in-picture configuration **XML_Cap_ThermalPip** (related API: **NET_DVR_GetSTDAbility** , capability type: "NET_DVR_GET_THERMAL_PIP_CAPABILITIES" (6767)):
added one node **<distance>** (fusion distance).
2. Extended the thermal picture-in-picture configuration structure **NET_DVR_THERMAL_PIP** (related API: **NET_DVR_GetSTDConfig** with command 6768-"NET_DVR_GET_THERMAL_PIP" and **NET_DVR_SetSTDConfig** with command 6769-"NET_DVR_SET_THERMAL_PIP"):
added one member **fDistance** (fusion distance).
3. Extended the basic configuration capability of temperature measurement **XML_Cap_ThermometryBasicParam** (related API: **NET_DVR_GetSTDAbility** , capability type: "NET_DVR_GET_THERMOMETRY_BASICPARAM_CAPABILITIES" (3620)):
added two temperature ranges "20-350" and "20-45" to the node **<temperatureRange>**.
4. Extended the basic parameters structure of temperature measurement **NET_DVR_THERMOMETRY_BASICPARAM** (related API: **NET_DVR_GetSTDConfig** with command 3621-"NET_DVR_GET_THERMOMETRY_BASICPARAM" and **NET_DVR_SetSTDConfig** with command 3622-"NET_DVR_SET_THERMOMETRY_BASICPARAM"):

added two temperature ranges "7" (from 20 to 350) and "8" (from 20 to 45)"to parameter **byThermometryRange**.

Summary of Changes in Version 6.1.0.25_July, 2019

Related Product: Thermal & Optical Bi-spectrum Network Speed Dome; Thermal & Optical Bi-spectrum Positioning System; Thermal & Optical Bi-spectrum Network Stable PTZ Camera in Version 5.5.19

1. Extended the thermal capability **XML_ThermalCap** (related API: **NET_DVR_STDXMLConfig** ; related URI: **/ISAPI/Thermal/capabilities**):
added one node: **<isSupportShipsDetectionWithScene>** (whether supports ship detection BY scene).
2. Added the function of configuring ship flow detection alarm, see details in **Configure Ship Flow Detection Alarm** .
3. Added the function of configuring dredger detection alarm, see details in **Configure Dredger Detection Alarm**
4. Extended the configuration capability of fire and smoke detection **XML_Cap_FireDetection** (related API: **NET_DVR_GetSTDAbility** ; capability type: 3635-"NET_DVR_GET_FIREDETECTION_CAPABILITIES"):
added five nodes: **<verificationSensitivity>** (sensitivity of double verification), **<fireAlgorithmModel>** (fire detection algorithm mode), **<agriculturalMachineryFilterEnabled>** (enable agricultural machinery filter), **<waterReflectionEnabled>** (enable water reflection), and **<patrolSensitivity>** (patrol sensitivity).
5. Extended the structure about fire and smoke detection parameters **NET_DVR_FIREDETECTION_CFG** (related APIs: **NET_DVR_GetSTDConfig** with command 3638-"NET_DVR_GET_FIREDETECTION_TRIGGER"; **NET_DVR_SetSTDConfig** with command 3639- "NET_DVR_SET_FIREDETECTION_TRIGGER"):
added five members: **byverificationSensitivity** (sensitivity of double verification), **byFireAlgorithmMode** (fire detection algorithm mode), **byAgriculturalMachineryFilterEnabled** (enable agricultural machinery filter or not), **byWaterReflectionEnabled** (enable water reflection or not), and **byPatrolSensitivity** (patrol sensitivity) with 5 reserved bytes.
6. Extended the burning prevention capabilities **XML_BurningPreventionCap** (related API: **NET_DVR_STDXMLConfig** ; related URI: **/ISAPI/Thermal/channels/<ID>/burningPrevention/capabilities**):
added three nodes: **<protectionMode>** (protection mode), **<burningRecoveryEnabled>** (enable burning recovery or not), and **<movementDuration>** (duration of lens movement for burning protection).
7. Extended the burning prevention parameters **XML_BurningPrevention** (related API: **NET_DVR_STDXMLConfig** ; related URI: **/ISAPI/Thermal/channels/<ID>/burningPrevention**):
added three nodes: **<protectionMode>** (protection mode), **<burningRecoveryEnabled>** (enable burning recovery or not), and **<movementDuration>** (duration of lens movement for burning protection).

Summary of Changes in Version 6.0.2.40_May, 2019

Related Product Type: Thermal Network Bullet Camera; Thermal Bi-spectrum Network Bullet Camera; Thermal & Optical Network Turret Camera; Thermal Box Camera V5.5.18

1. Extended the preset (scene) configuration capability ***XML_Cap_ThermometryScene*** (related API: ***NET_DVR_GetSTDAbility*** with command 3623-"NET_DVR_GET_THERMOMETRY_SCENE_CAPABILITIES"): added one node **<emissivityMode>** (emissivity type)
2. Extended the thermometry preset information structure ***NET_DVR_THERMOMETRY_PRESETINFO_PARAM*** (related API: ***NET_DVR_GetSTDConfig*** and ***NET_DVR_SetSTDConfig*** with command 3624-"NET_DVR_GET_THERMOMETRY_PRESETINFO" and 3625-"NET_DVR_SET_THERMOMETRY_PRESETINFO"): added one parameter **byemissivityMode** (emissivity type).
3. Extended the capability of thermometry basic configuration ***XML_Cap_ThermometryBasicParam*** (request API: ***NET_DVR_GetSTDAbility*** with command 3620-***NET_DVR_GET_THERMOMETRY_BASICPARAM_CAPABILITIES***): added two nodes **<emissivityMode>** (emissivity type) and **<displayTemperatureInOpticalChannelEnabled>** (display the temperature information of optical channel).
4. Extended the thermometry basic parameters structure ***NET_DVR_THERMOMETRY_BASICPARAM*** (related API: ***NET_DVR_GetSTDConfig*** and ***NET_DVR_SetSTDConfig*** with command 3621-"NET_DVR_GET_THERMOMETRY_BASICPARAM" and 3622-"NET_DVR_SET_THERMOMETRY_BASICPARAM"): added two parameters **byemissivityMode** (emissivity type) and **bydisplayTemperatureInOpticalChannelEnabled** (display the temperature information of optical channel) with two reserved bytes.

Summary of Changes in Version 6.0.2.30_03/2019

Related Project Info: DS-9600NI Series, DS-7700NI Series, DS-7600NI Series, and DS-8600NI-K Series Network Video Recorder V4.21.000

1. Extended structures of temperature alarm details and temperature difference alarm details, i.e., ***NET_DVR_THERMOMETRY_ALARM*** and ***NET_DVR_THERMOMETRY_DIFF_ALARM*** : added one parameter **byPicTransType** (picture transmission method).
2. Extended the structure of fire and smoke alarm details ***NET_DVR_FIREDETECTION_ALARM*** : added one parameter **byPicTransType** (picture transmission method).

Summary of Changes in Version 6.0.2.151 (Windows)_April, 2019

Related Product Type: Handheld Thermography Camera in Version 1.1

1. Extended the message of camera power capability ***XML_Cap_Power*** and message of camera power parameters ***XML_Power*** :

- added one node <**batteryPower**> (battery percentage).
- Extended the thermal capability message XML_ThermalCap (related API: **NET_DVR_STDXMLConfig** ; related URI: GET **/ISAPI/Thermal/capabilities**): added one node <**isSupporttthermometryOffLineCapture**> (whether device supports offline capture).
 - Added one URI of capturing one picture offline (related API: **NET_DVR_STDXMLConfig**): GET **/ISAPI/Thermal/channels/<ID>/thermometry/OffLineCapture?format=json**

Summary of Changes in Version 6.0.2.20 (Windows)_02/2019

Related Product Type: Thermographic Automation Thermal Camera V2.1

Related Product Model: DS-2TA03-7AVI, DS-2TA03-15VI, DS-2TA03-15SVI, DS-2TA06-25SVI, DS-2TA06-25VI

- Extended the structure of real-time temperature information **NET_DVR_THERMOMETRY_UPLOAD** (related API: **NET_DVR_StartRemoteConfig** with command of 3629-NET_DVR_GET_REALTIME_THERMOMETRY): added one member **byIsFreezedata** (whether supports freezing data) with one reserved byte.
- Extended the structure of thermometry rule temperature information **NET_DVR_THERMOMETRYRULE_TEMPERATURE_INFO** (related API: **NET_DVR_GetDVRConfig** with the command 23001-NET_DVR_GET_THERMOMETRYRULE_TEMPERATURE_INFO): added one member **byIsFreezedata** (whether supports freezing data) with one reserved byte.

Summary of Changes in Version 6.0.2.5 (Windows)_01/2019

Related Product Type: Network Thermographic Automation Camera

Related Product Model: DS-2TA03-4AUM, DS-2TA03-7AUF, DS-2TA03-7HUF, DS-2TA03-10AUF, DS-2TA03-15AUF, DS-2TA03-15HUF, DS-2TA06-7AXF, DS-2TA06-15AXF

- Extended the JPEG picture information structure **NET_DVR_JPEGPICTURE_WITH_APPENDDATA** (related API: **NET_DVR_CaptureJPEGPicture_WithAppendData**): added one member **byIsFreezedata** (whether supports freezing data).
- Extended the structure of real-time temperature information **NET_DVR_THERMOMETRY_UPLOAD** (related API: **NET_DVR_StartRemoteConfig** with command of 3629-NET_DVR_GET_REALTIME_THERMOMETRY): added one member **byIsFreezedata** (whether supports freezing data) with one reserved byte.
- Extended the structure of thermometry rule temperature information **NET_DVR_THERMOMETRYRULE_TEMPERATURE_INFO** (related API: **NET_DVR_GetDVRConfig** with the command 23001-NET_DVR_GET_THERMOMETRYRULE_TEMPERATURE_INFO): added one member **byIsFreezedata** (whether supports freezing data) with one reserved byte.

Summary of Changes in Version 6.0.2.5 (Windows)_01/2019

Related Product Type: Thermal & Optical Bi-spectrum Network Speed Dome, Thermal & Optical Bi-spectrum Positioning System, Thermal & Optical Bi-spectrum Network Stable PTZ Camera,

Related Product Model: DS-2TD4237-xx/V2 series, DS-2TD4136-xx/V2 series, DS-2TD6236-xx/V2 series, DS-2TD6236-xx/V2 series, DS-2TD8166xx/V2 series, DS-2TD8136xx/V2 series

1. Extended structure of basic temperature measurement parameters
NET_DVR_THERMOMETRY_BASICPARAM (related API: **NET_DVR_GetSTDConfig**, **NET_DVR_SetSTDConfig**):
added two members: **dwAlertFilteringTime** (temperature pre-alarm dwell time) and **dwAlarmFilteringTime** (temperature alarm dwell time) with eight reserved bytes.
2. Extended the structure of temperature alarm rule parameters
NET_DVR_THERMOMETRY_ALARMRULE_PARAM (related API: **NET_DVR_GetSTDConfig**, **NET_DVR_SetSTDConfig**):
added two members: **dwAlertFilteringTime** (temperature pre-alarm dwell time) and **dwAlarmFilteringTime** (temperature alarm dwell time) with eight reserved bytes.
3. Extended the structure of temperature difference alarm rule
NET_DVR_THERMOMETRY_DIFFCOMPARISON_PARAM (related API: **NET_DVR_GetSTDConfig**, **NET_DVR_SetSTDConfig**):
added one member **dwAlarmFilteringTime** (temperature alarm dwell time) with four reserved bytes.
4. Extended the fire detection capability message **XML_Cap_FireDetection** :
added one node **<smokeAuxiliaryDetectionEnabled>** (whether to enable fire and smoke detection).
5. Extended the structure of fire and smoke configuration parameters
NET_DVR_FIREDETECTION_CFG (related API: **NET_DVR_GetSTDConfig**, **NET_DVR_SetSTDConfig**):
added one member **bySmokeAuxiliaryDetectionEnabled** (whether to enable fire and smoke detection).
6. Added a integration flow and description of fire and smoke alarm configuration, refer to **Configure Fire and Smoke Alarm** for details.

Summary of Changes in Version 6.0.0.20 (Windows/Linux)_12/2018

Related Product Type: Thermal Network Bullet Camera; Thermal Bi-spectrum Network Bullet Camera; Thermal & Optical Network Dome Camera V5.5.16

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 message of picture-in-picture configuration capability **XML_Cap_ThermalPip** (related API: **NET_DVR_GetSTDAbility** with the command of 6767-**NET_DVR_GET_THERMAL_PIP_CAPABILITIES**):
added two nodes **<imageFusionRatio>** (image fusion ratio) and **<borderFusionRatio>** (border fusion ratio).
2. Extended the structure of thermal picture-in-picture configuration **NET_DVR_THERMAL_PIP** (related API: **NET_DVR_GetSTDConfig** and **NET_DVR_SetSTDConfig**):
added two parameters **byImageFusionRatio** (image fusion ration) and **byBorderFusionRatio** (boarder fusion ration).
3. Extended the fire detection configuration capability **XML_Cap_FireDetection** (related API: **NET_DVR_GetSTDAbility** with the command of 3635-**NET_DVR_GET_FIREDETECTION_CAPABILITIES**):

added one node **<fireSourceDetection>** (fire source detection mode).

4. Extended the fire detection configuration structure **NET_DVR_FIREDETECTION_CFG** (related API: **NET_DVR_GetSTDConfig** with command: "3636-NET_DVR_GET_FIREDETECTION"; **NET_DVR_SetSTDConfig** with command: "3637-NET_DVR_SET_FIREDETECTION"): added one parameter **byFireSourceDetection** (fire source detection mode).

Summary of Changes in Version 6.0.0.10 (Windows/Linux)_10/2018

Related Product Type: Thermographic Automation Thermal Camera

Related Product Model: DS-2TA03-15SVI, DS2TA06-25SVI

1. Extended the capability of thermometry basic parameters configuration
XML_Cap_ThermometryBasicParam :
added ten nodes: **<isSupportAlertOutputIOPortList>**, **<isSupportAlarmOutputIOPortList>**, **<alertFilteringTime>** (temperature pre-alarm dwell time), **<alarmFilteringTime>** (temperature alarm dwell time), **<displayMaxTemperatureEnabled>** (whether displays the maximum temperature), **<displayMinTemperatureEnabled>** (whether displays the minimum temperature), **<displayAverageTemperatureEnabled>** (whether displays the average temperature), **<thermometryInfoDisplayposition>** (thermometry information overlay position), **<calibrationCoefficientEnabled>** (whether enables calibration coefficient), **<calibrationCoefficient>** (calibration coefficient).
2. Extended the structure of thermometry basic parameters
NET_DVR_THERMOMETRY_BASICPARAM (related API: **NET_DVR_GetSTDConfig** , **NET_DVR_SetSTDConfig**):
added four members with four reserved bytes: **byDisplayMaxTemperatureEnabled** (whether displays the maximum temperature), **byDisplayMinTemperatureEnabled** (whether displays the minimum temperature), **byDisplayAverageTemperatureEnabled** (whether displays the average temperature), **byThermometryInfoDisplayposition** (thermometry information overlay position).
3. Extended the capability of thermometry preset (scene) configuration
XML_Cap_ThermometryScene :
added one sub node **<distanceUnit>** to node **<ThermometryRegion>**.
4. Extended the thermometry preset information structure
NET_DVR_THERMOMETRY_PRESETINFO_PARAM (related API: **NET_DVR_GetSTDConfig** **NET_DVR_SetSTDConfig**):
added one parameter **<byDistanceUnit>** with one reserved byte.

Summary of Changes in Version 5.3.6.40 (Windows/Linux)_09/2018

Related Product Type: Network Thermographic Automation Camera DS-2TA03-4AUM

1. Extended the capability of temperature measurement basic parameters configuration
XML_Cap_ThermometryBasicParam and thermometry basic parameters
XML_ThermometryBasicParam : added nine nodes: **<emissivity>**, **<distanceUnit>** (distance unit), **<distance>**, **<reflectiveEnable>** (whether enables temperature reflection), **<reflectiveTemperature>** (reflective temperature), **<alert>** (pre-alarm threshold), **<alarm>**

- (alarm threshold), <**thermalOpticalTransmittance**> (optical transmissivity), <**externalOpticsWindowCorrection**> (external optical temperature).
2. Extended the basic parameter configuration structure of temperature measurement **NET_DVR_THERMOMETRY_BASICPARAM** (related API: **NET_DVR_GetSTDConfig** , **NET_DVR_SetSTDConfig**):
added nine members: **fEmissivity** (emissivity), **byDistanceUnit** (distance unit), **byReflectiveEnabled** (whether enables reflecting temperature), **wDistance** (distance), **fReflectiveTemperature** (reflected temperature), **fAlert** (pre-alarm temperature threshold), **fAlarm** (alarm temperature threshold), **fThermalOpticalTransmittance** (optical transmissivity), **fExternalOpticsWindowCorrection** (external optical temperature).
 3. Extended the capability of thermometry preset (scene) configuration **XML_Cap_ThermometryScene** :
added three nodes: <**maxPointNum**> (maximum number of point rules that can be configured), <**maxLineNum**> (maximum number of line rules that can be configured), <**maxRegionNum**> (maximum number of frame rules that can be configured).
 4. Extended the capability of temperature measurement mode configuration **XML_Cap_ThermometryMode** and temperature measurement mode configuration parameters **XML_ThermometryMode** :
added one node <**thermometryROIEnabled**> (whether enables ROI temperature measurement).
 5. Extended the structure of temperature measurement mode configuration **NET_DVR_THERMOMETRY_MODE** (related API: **NET_DVR_GetSTDConfig** , **NET_DVR_SetSTDConfig**):
added one parameter **byThermometryROIEnabled** (whether to enable ROI temperature measurement).
 6. Added the function of capturing JPEG picture with pixel-to-pixel thermometry data **NET_DVR_CaptureJPEGPicture_WithAppendData** .
 7. Extended the structure about focus mode configuration of speed dome **NET_DVR_FOCUSMODE_CFG** (related API: **NET_DVR_GetSTDConfig** **NET_DVR_SetSTDConfig**):
added two parameters **byFocusMode** (focus mode) and **dwRelativeFocusPos** (relative focus value).
 8. Extended the structure of real-time temperature information **NET_DVR_THERMOMETRY_UPLOAD** (related API: **NET_DVR_StartRemoteConfig** , **NET_DVR_StopRemoteConfig**):
added two members **struHighestPoint**(coordinates of highest temperature position for thermometry by frame/line) and **struLowestPoint** (coordinates of lowest temperature position for thermometry by frame/line).
 9. Added the function of manually getting thermometry rule temperature information. Related API **NET_DVR_GetDVRConfig** with command 23001-
NET_DVR_GET_THERMOMETRYRULE_TEMPERATURE_INFO.
 10. Extended the camera parameter capability set **XML_CAMERAPARA** : added two sub nodes <**FocusModeSet**> (focus mode) and <**relativeFocusPos**> (relative focus sensitivity) to <FocusMode>.

Summary of Changes in Version 5.3.6.152 (Windows/Linux)_07/2018

1. Extended the body thermometry alarm information structure
NET_DVR_FACE_THERMOMETRY_ALARM (related API: **NET_DVR_SetDVRMessageCallback_V50** and **NET_DVR_StartListen_V30**):
added 11 parameters: **byAlarmRule** (alarm rule), **fAlarmTemperature** (alarm triggered temperature), **fRuleTemperature** (rule temperature), **fMinTemperature** (minimum temperature), **fAverageTemperature** (average temperature), **struMinTemperaturePoint** (coordinates of lowest temperature position), **struMaxTemperaturePoint** (coordinates of highest temperature position), **dwHighTemperatureTargetImageLen** (length of high temperature object thumbnail), **pHighTemperatureTargetImage** (pointer to high temperature object thumbnail), and **struHighTemperatureTargetRegion** (coordinates of high temperature object thumbnail).
2. Extended the capability of temperature measurement configuration
XML_Cap_ThermometryBasicParam :
added one temperature range "-20 to 120" to node <temperatureRange>.
3. Extended the basic parameter configuration structure of temperature measurement
NET_DVR_THERMOMETRY_BASICPARAM (related API: **NET_DVR_GetSTDConfig** **NET_DVR_SetSTDConfig**):
added one value "6-(-20 to 120)" to **byThermometryRange** (temperature range).

Summary of Changes in Version 5.3.3.5 (Windows/Linux)_12/2017

1. New document

Chapter 2 Manually Measure Temperature

When you enable the manual thermometry function of device, you can click any position on the interface to show the real temperature.

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

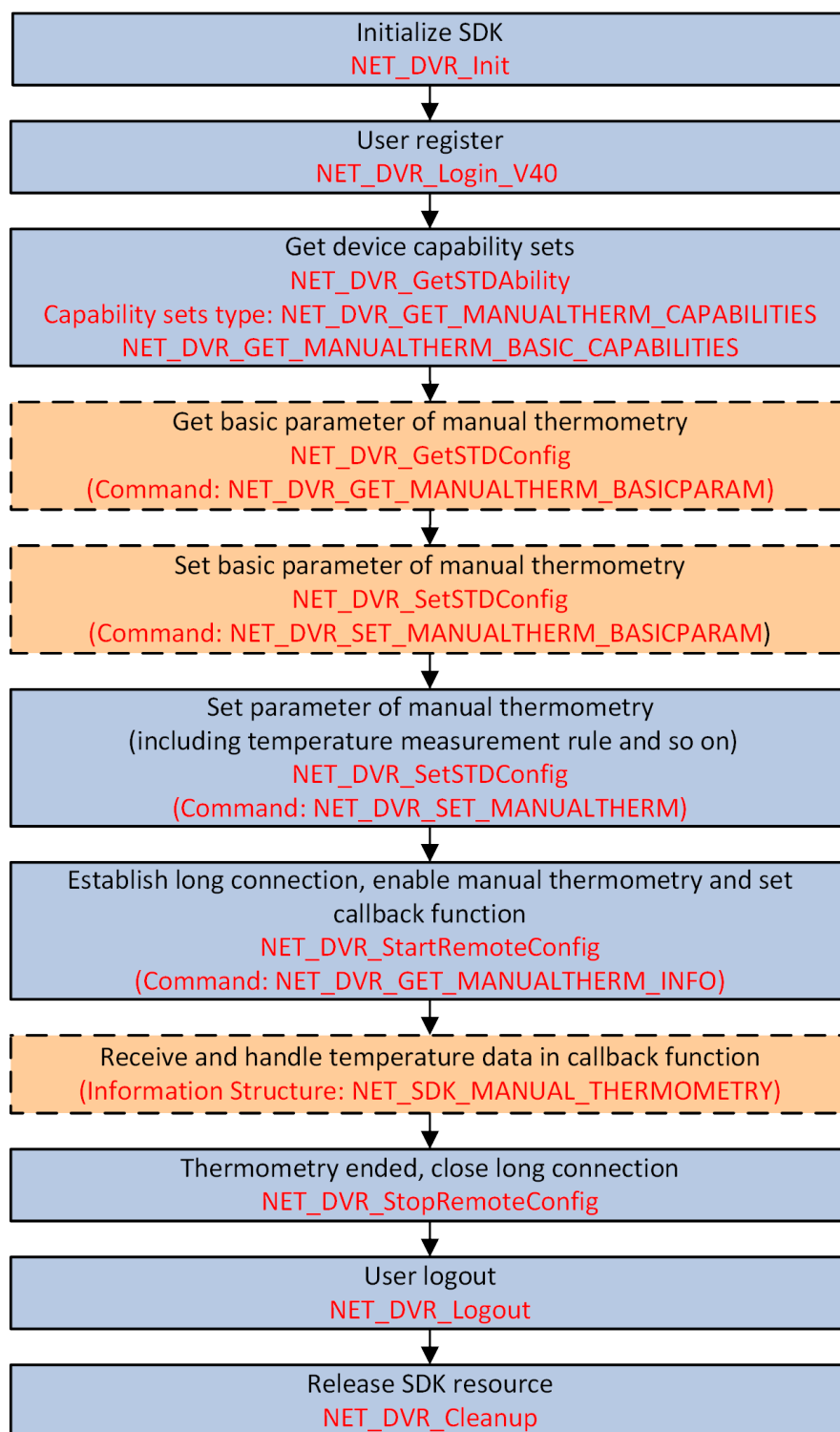


Figure 2-1 Programming Flow of Manually Measuring Temperature

1. Call **NET_DVR_GetSTDAbility** with the command of

NET_DVR_GET_MANUALTHERM_CAPABILITIES

(command No.: 6707), and set condition parameter **IpCondBuffer** in the structure of **NET_DVR_STD_ABILITY** to 4-byte channel No. for getting the manual thermometry capability to check whether the device supports manual thermometry.

The manual thermometry capability is returned in **XML_Cap_ManualThermometry** by the output parameter **IpOutBuffer** in the structure of **NET_DVR_STD_ABILITY**.

2. Set manual thermometry basic parameters.

1) Call *NET_DVR_GetSTDAbility* with the command of

NET_DVR_GET_MANUALTHERM_BASIC_CAPABILITIES

(command No.: 6715), and set condition parameter **IpCondBuffer** in the structure of **NET_DVR_STD_ABILITY** to 4-byte channel No. for getting the basic parameters configuration capability of manual thermometry.

The basic parameters configuration capability of manual thermometry is returned in **XML_Cap_ManualThermBasic** by the output parameter **IpOutBuffer** in the structure of **NET_DVR_STD_ABILITY**.

2) *NET_DVR_GetSTDConfig* with the command of

NET_DVR_GET_MANUALTHERM_BASICPARAM

(command No.: 6717) and set the condition parameter **IpCondBuffer** in the structure of **NET_DVR_STD_CONFIG** to 4-byte channel No. for getting the manual thermometry basic parameters for reference.

The manual thermometry basic parameters **NET_SDK_MANUALTHERM_BASICPARAM** are returned by the output parameter **IpOutBuffer** in the structure of **NET_DVR_STD_CONFIG**.

3) Call *NET_DVR_SetSTDConfig* with the command of

NET_DVR_SET_MANUALTHERM_BASICPARAM

(command No.: 6716), set the condition parameter **IpCondBuffer** and input parameter **IpInBuffer** in the structure of **NET_DVR_STD_CONFIG** to 4-byte channel No. and

NET_SDK_MANUALTHERM_BASICPARAM for setting manual thermometry basic parameters.

3. Call *NET_DVR_SetSTDConfig* with the command of

NET_DVR_SET_MANUALTHERM

(command No.: 6708), set the condition parameter **IpCondBuffer** and input parameter

IpInBuffer in the structure of **NET_DVR_STD_CONFIG** to 4-byte channel No. and

NET_SDK_MANUAL_THERMOMETRY for setting manual thermometry rules, including rule ID, detection type, and so on.

4. Call *NET_DVR_StartRemoteConfig* with the command of

NET_DVR_GET_MANUALTHERM_INFO

(command No.: 6706) and set the input parameters **IpInBuffer** to

NET_DVR_REALTIME_THERMOMETRY_COND.

The manual thermometry result **NET_SDK_MANUAL_THERMOMETRY** is returned in callback function **fRemoteConfigCallback**.

5. Call *NET_DVR_StopRemoteConfig* to disconnect the persistent connection to stop remote configuration, and release resources.

Example

Sample Code of Manually Measuring Temperature

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

//Macro Definition of temporal 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)

int iNum=0;
#define ISAPI_OUT_LEN 3 * 1024 * 1024
#define ISAPI_STATUS_LEN 8*1024

void CALLBACK GetManualThermInfoCallback(DWORD dwType, void* lpBuffer, DWORD dwBufLen, void* pUserData)
{
    if (dwType == NET_SDK_CALLBACK_TYPE_DATA)
    {
        LPNET_SDK_MANUAL_THERMOMETRY lpManualThermometry = new NET_SDK_MANUAL_THERMOMETRY;
        memcpy(lpManualThermometry, lpBuffer, sizeof(*lpManualThermometry));

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

        printf("Manual Temperature Measurement Result: dwChannel[%d]byThermometryUnit[d%]dwAbsTime[%4.4d%2.2d%2.2d%2.2d%2.2d%2.2d]\n",
            lpManualThermometry->dwChannel, lpManualThermometry->byThermometryUnit, struAbsTime.dwYear,
            struAbsTime.dwMonth, struAbsTime.dwDay, struAbsTime.dwHour, struAbsTime.dwMinute,
            struAbsTime.dwSecond);

        if (lpManualThermometry != NULL)
        {
            delete lpManualThermometry;
            lpManualThermometry = NULL;
        }
    }
    else if (dwType == NET_SDK_CALLBACK_TYPE_STATUS)
    {
        DWORD dwStatus = *(DWORD*)lpBuffer;
        if (dwStatus == NET_SDK_CALLBACK_STATUS_SUCCESS)
```

```
{
    printf("dwStatus:NET_SDK_CALLBACK_STATUS_SUCCESS\n");
}
else if (dwStatus == NET_SDK_CALLBACK_STATUS_FAILED)
{
    DWORD dwErrCode = *(DWORD*)((char *)lpBuffer + 4);
    printf("NET_DVR_GET_MANUALTHERM_INFO failed, Error code %d\n", dwErrCode);
}
}
}

void main()
{
    DWORD dwChannel = 2; //Thermal imaging channel

    char *m_pOutBuf = new char[ISAPI_OUT_LEN];
    memset(m_pOutBuf, 0, ISAPI_OUT_LEN);

    char *m_pStatusBuf = new char[ISAPI_STATUS_LEN];
    memset(m_pStatusBuf, 0, ISAPI_STATUS_LEN);

    //-----
    //Initialize
    NET_DVR_Init();

    //Set connected time and reconnected time
    NET_DVR_SetConnectTime(2000, 1);
    NET_DVR_SetReconnect(10000, true);

    //-----
    //Register device (it is not required when listening alarm)
    LONG lUserID;
    NET_DVR_DEVICEINFO_V30 struDeviceInfo;
    lUserID = NET_DVR_Login_V30("10.8.10.199 ", 8000, "admin", "abcd1234", &struDeviceInfo);
    if (lUserID < 0)
    {
        printf("Login error, %d\n", NET_DVR_GetLastError());
        NET_DVR_Cleanup();
        return;
    }

    //Capability sets of manual temperature measurement
    NET_DVR_STD_ABILITY struStdAbility = {0};
    struStdAbility.lpCondBuffer = &dwChannel;
    struStdAbility.dwCondSize = sizeof(DWORD);

    struStdAbility.lpOutBuffer   = m_pOutBuf;
    struStdAbility.dwOutSize    = ISAPI_OUT_LEN;
    struStdAbility.lpStatusBuffer = m_pStatusBuf;
    struStdAbility.dwStatusSize = ISAPI_STATUS_LEN;

    if(!NET_DVR_GetSTDAbility(lUserID,NET_DVR_GET_MANUALTHERM_CAPABILITIES,&struStdAbility))
```

```
{
    printf("NET_DVR_GET_MANUALTHERM_CAPABILITIES failed, error code: %d\n", NET_DVR_GetLastError());
}
else
{
    printf("NET_DVR_GET_MANUALTHERM_CAPABILITIES is successful!");
}

//Configure basic parameter of manual temperature measurement
NET_DVR_STD_CONFIG struStdConfig = {0};
struStdConfig.lpCondBuffer = &dwChannel;
struStdConfig.dwCondSize = sizeof(dwChannel);
struStdConfig.lpInBuffer = NULL;
struStdConfig.dwInSize = 0;

NET_SDK_MANUALTHERM_BASICPARAM struManualThermBasicParam = {0};
struStdConfig.lpOutBuffer = (LPVOID)&struManualThermBasicParam;
struStdConfig.dwOutSize = sizeof(struManualThermBasicParam);

struStdConfig.lpStatusBuffer = m_pStatusBuf;
struStdConfig.dwStatusSize = ISAPI_STATUS_LEN;

DWORD dwReturned=0;
if(!NET_DVR_GetSTDConfig(IUserID,NET_DVR_GET_MANUALTHERM_BASICPARAM,&struStdConfig))
{
    printf("NET_DVR_GET_MANUALTHERM_BASICPARAM failed, error code: %d\n", NET_DVR_GetLastError());
}
else
{
    printf("NET_DVR_GET_MANUALTHERM_BASICPARAM is successful!");
}

struManualThermBasicParam.wDistance = 20; //Distance (m), ranging from 0 to 10000
struStdConfig.lpInBuffer = (LPVOID)&struManualThermBasicParam;
struStdConfig.dwInSize = sizeof(struManualThermBasicParam);

if(!NET_DVR_SetSTDConfig(IUserID,NET_DVR_SET_MANUALTHERM_BASICPARAM,&struStdConfig))
{
    printf("NET_DVR_SET_MANUALTHERM_BASICPARAM failed, error code: %d\n", NET_DVR_GetLastError());
}
else
{
    printf("NET_DVR_SET_MANUALTHERM_BASICPARAM is successful!");
}

//Configure rule for manual temperature measurement
NET_SDK_MANUAL_THERMOMETRY struManualTherm = {0};
struManualTherm.dwSize = sizeof(struManualTherm);
struManualTherm.dwChannel = dwChannel;
struManualTherm.byThermometryUnit = 0; //Temperature unit: 0-Centigrade, 1-Fahrenheit, 2-Kelvin
struManualTherm.struRuleInfo.byRuleID = 1;
struManualTherm.struRuleInfo.byEnable=1;
```

```
strcpy(struManualTherm.struRuleInfo.szRuleName, "TestName");
struManualTherm.struRuleInfo.byRuleCalibType = 0;
struManualTherm.struRuleInfo.struPointTherm.struPoint.fX = 0.5; //Normalized value, ranging from 0.001 to 1.
struManualTherm.struRuleInfo.struPointTherm.struPoint.fY = 0.5; //Normalized value, ranging from 0.001 to 1.

struStdConfig.lpCondBuffer = &dwChannel;
struStdConfig.dwCondSize = sizeof(dwChannel);
struStdConfig.lpInBuffer = (LPVOID)&struManualTherm;;
struStdConfig.dwInSize = sizeof(struManualTherm);

if(!NET_DVR_SetSTDConfig(IUserID,NET_DVR_SET_MANUALTHERM,&struStdConfig))
{
    printf("NET_DVR_SET_MANUALTHERM failed, error code: %d\n", NET_DVR_GetLastError());
}
else
{
    printf("NET_DVR_SET_MANUALTHERM is successful!");
}

//Enable manual temperature measurement
NET_DVR_REALTIME_THERMOMETRY_COND struThermCond = {0};
struThermCond.dwSize = sizeof(struThermCond);
struThermCond.byRuleID = 1;    //Rule ID, 0-Get all rules, the rule ID starts from 1.
struThermCond.dwChan = dwChannel; //Start from 1, 0xffffffff- Get all channels

LONG ManualHandle = NET_DVR_StartRemoteConfig(IUserID, NET_DVR_GET_MANUALTHERM_INFO,
&struThermCond, sizeof(struThermCond), GetManualThermInfoCallback, NULL);
if (ManualHandle >= 0)
{
    printf("NET_DVR_GET_MANUALTHERM_INFO failed, error code: %d\n", NET_DVR_GetLastError());
}
else
{
    printf("NET_DVR_GET_MANUALTHERM_INFO is successful!");
}

Sleep(5000); //Wait for receiving manual measurement result

//Close the handle created by long connection configuration API, and release resource.
if(!NET_DVR_StopRemoteConfig(ManualHandle))
{
    printf("NET_DVR_StopRemoteConfig failed, error code: %d\n", NET_DVR_GetLastError());
}

//User logout, if the user is not login, skip this step.
NET_DVR_Logout(IUserID);

//Release SDK resource
NET_DVR_Cleanup();

if (m_pOutBuf != NULL)
{

```



```
    delete []m_pOutBuf;
    m_pOutBuf = NULL;
}

if (m_pStatusBuf != NULL)
{
    delete []m_pStatusBuf;
    m_pStatusBuf = NULL;
}

return;
}
```

What to do next

Call ***NET_DVR_Logout*** and ***NET_DVR_Cleanup*** to log out and release the resources.

Chapter 3 Measure Real-Time Temperature

When the thermometry mode is configured, you can start measure real-time temperature.

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

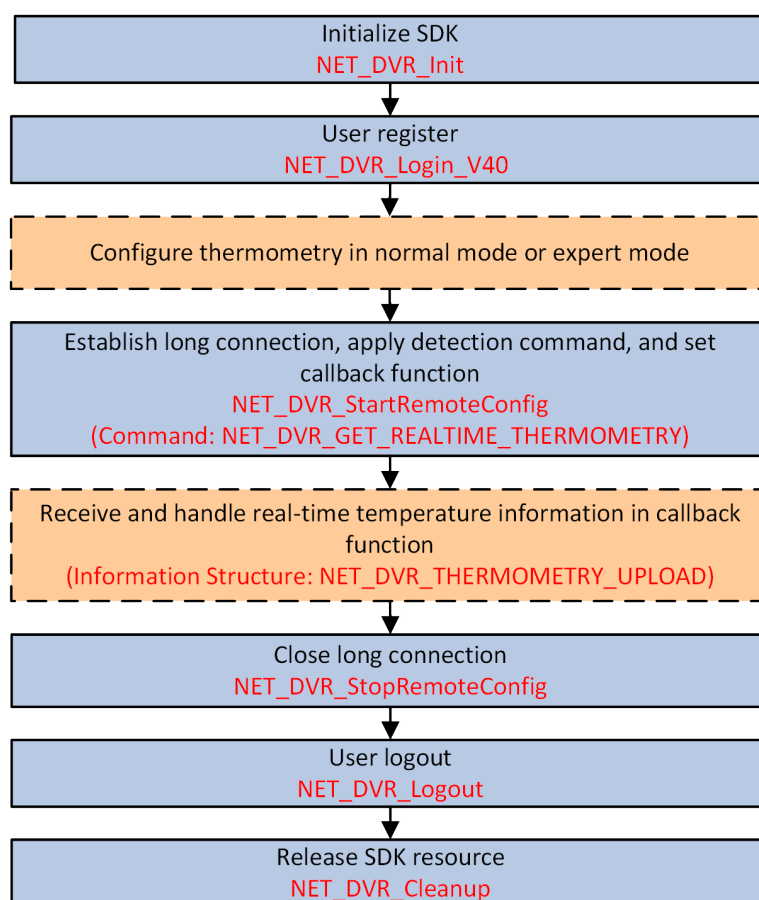


Figure 3-1 Programming Flow of Measuring Real-Time Temperature

1. Call **NET_DVR_StartRemoteConfig** with the command of **NET_DVR_GET_REALTIME_THERMOMETRY** (command No.: 3629) and set the input parameters **lpInBuffer** to for establishing long connection.
The real-time thermometry result is returned in the configured callback function **fRemoteConfigCallback**.
2. Call **NET_DVR_StopRemoteConfig** to disconnect the persistent connection to stop remote configuration, and release resources.

Example

Sample Code of Measuring Real-Time Temperature

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

//Macro Definition of temporal 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)

int iNum=0;
#define ISAPI_OUT_LEN 3 * 1024 * 1024
#define ISAPI_STATUS_LEN 8*1024

void CALLBACK GetThermInfoCallback(DWORD dwType, void* lpBuffer, DWORD dwBufLen, void* pUserData)
{
    if (dwType == NET_SDK_CALLBACK_TYPE_DATA)
    {
        LPNET_DVR_THERMOMETRY_UPLOAD lpThermometry = new NET_DVR_THERMOMETRY_UPLOAD;
        memcpy(lpThermometry, lpBuffer, sizeof(*lpThermometry));

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

        printf("Real-time temperature measurement result:
byRuleID[%d]wPresetNo[%d]byRuleCalibType[%d]byThermometryUnit[d%]byDataType[d%]"
        "dwAbsTime[%4.4d%2.2d%2.2d%2.2d%2.2d%2.2d]\n", lpThermometry->byRuleID, lpThermometry->wPresetNo,
        lpThermometry->byRuleCalibType,lpThermometry->byThermometryUnit, lpThermometry->byDataType,
        struAbsTime.dwYear, struAbsTime.dwMonth, struAbsTime.dwDay,
        struAbsTime.dwHour, struAbsTime.dwMinute, struAbsTime.dwSecond);

        if(lpThermometry->byRuleCalibType==0) //Measure temperature by point
        {
            printf("Information of Measuring Temperature by Point: fTemperature[%d]\n", lpThermometry->struPointThermCfg.fTemperature);
        }

        if((lpThermometry->byRuleCalibType==1) || (lpThermometry->byRuleCalibType==2)) //Measure temperature by
        frame or line
```

```
{
    printf("Information of Measuring Temperature by Frame or Line:
fMaxTemperature[%d]fMinTemperature[%d]fAverageTemperature[%d]fTemperatureDiff[%d]\n",
        lpThermometry->struLinePolygonThermCfg.fMaxTemperature,lpThermometry-
>struLinePolygonThermCfg.fMinTemperature,
        lpThermometry->struLinePolygonThermCfg.fAverageTemperature,lpThermometry-
>struLinePolygonThermCfg.fTemperatureDiff);
}

if (lpThermometry != NULL)
{
    delete lpThermometry;
    lpThermometry = NULL;
}
}
else if (dwType == NET_SDK_CALLBACK_TYPE_STATUS)
{
    DWORD dwStatus = *(DWORD*)lpBuffer;
    if (dwStatus == NET_SDK_CALLBACK_STATUS_SUCCESS)
    {
        printf("dwStatus:NET_SDK_CALLBACK_STATUS_SUCCESS\n");
    }
    else if (dwStatus == NET_SDK_CALLBACK_STATUS_FAILED)
    {
        DWORD dwErrCode = *(DWORD*)((char *)lpBuffer + 4);
        printf("NET_DVR_GET_MANUALTHERM_INFO failed, Error code %d\n", dwErrCode);
    }
}
}

void main()
{
    DWORD dwChannel = 2; //Thermal imaging channel

    //-----
    //Initialize
    NET_DVR_Init();

    //Set connected time and reconnected time
    NET_DVR_SetConnectTime(2000, 1);
    NET_DVR_SetReconnect(10000, true);

    //-----
    //Register device (it is not required when listening alarm)
    LONG lUserID;
    NET_DVR_DEVICEINFO_V30 struDeviceInfo;
    lUserID = NET_DVR_Login_V30("10.8.10.199 ", 8000, "admin", "abcd1234", &struDeviceInfo);
    if (lUserID < 0)
    {
        printf("Login error, %d\n", NET_DVR_GetLastError());
        NET_DVR_Cleanup();
        return;
    }
}
```

```
}

//Enable real-time temperature measurement
NET_DVR_REALTIME_THERMOMETRY_COND struThermCond = {0};
struThermCond.dwSize = sizeof(struThermCond);
struThermCond.byRuleID = 1;    //Rule ID, 0-Get All Rules, the ID starts from 1.
struThermCond.dwChan = dwChannel; //Start from 1, 0xffffffff- Get All Channels

LONG lHandle = NET_DVR_StartRemoteConfig(lUserID, NET_DVR_GET_REALTIME_THERMOMETRY,
&struThermCond, sizeof(struThermCond), GetThermInfoCallback, NULL);
if (lHandle >= 0)
{
    printf("NET_DVR_GET_REALTIME_THERMOMETRY failed, error code: %d\n", NET_DVR_GetLastError());
}
else
{
    printf("NET_DVR_GET_REALTIME_THERMOMETRY is successful!");
}

Sleep(5000); //Wait for receiving real-time temperature measurement result

//Close the handle created by long connection configuration API, and release resource.
if(!NET_DVR_StopRemoteConfig(lHandle))
{
    printf("NET_DVR_StopRemoteConfig failed, error code: %d\n", NET_DVR_GetLastError());
}

//User logout, if the user is not login, skip this step.
NET_DVR_Logout(lUserID);

//Release SDK resource
NET_DVR_Cleanup();

return;
}
```

What to do next

Call **NET_DVR_Logout** and **NET_DVR_Cleanup** to log out and release the resources.

Chapter 4 Alarm/Event Configuration

Before the alarms can be triggered or the events can be detected, you must configure parameters, such as detection rule, alarm threshold, arming schedule, linkage action, and so on, for different alarms or events. This chapter shows the configuration processes of general alarms or events with multiple types.



Note

Currently, only the configuration of behavior analysis alarm is provided. For other types of alarm or event configurations, refer to the user manuals of typical HCNetsDK integrations.

4.1 Configure Temperature Screening Alarm

Temperature screening can detect the human body temperatures at large-scale area in the public places (e.g., customs, airports, schools, hospitals) with high people density, and help to fast find and track the person with high body temperature. Then, the alarm will be remotely uploaded to the monitor center via the network for real-time analysis and handling. This function is mainly applied to the control and prevention of some serious febrile infectious diseases (e.g., SARS, Avian Influenza, Ebola).

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

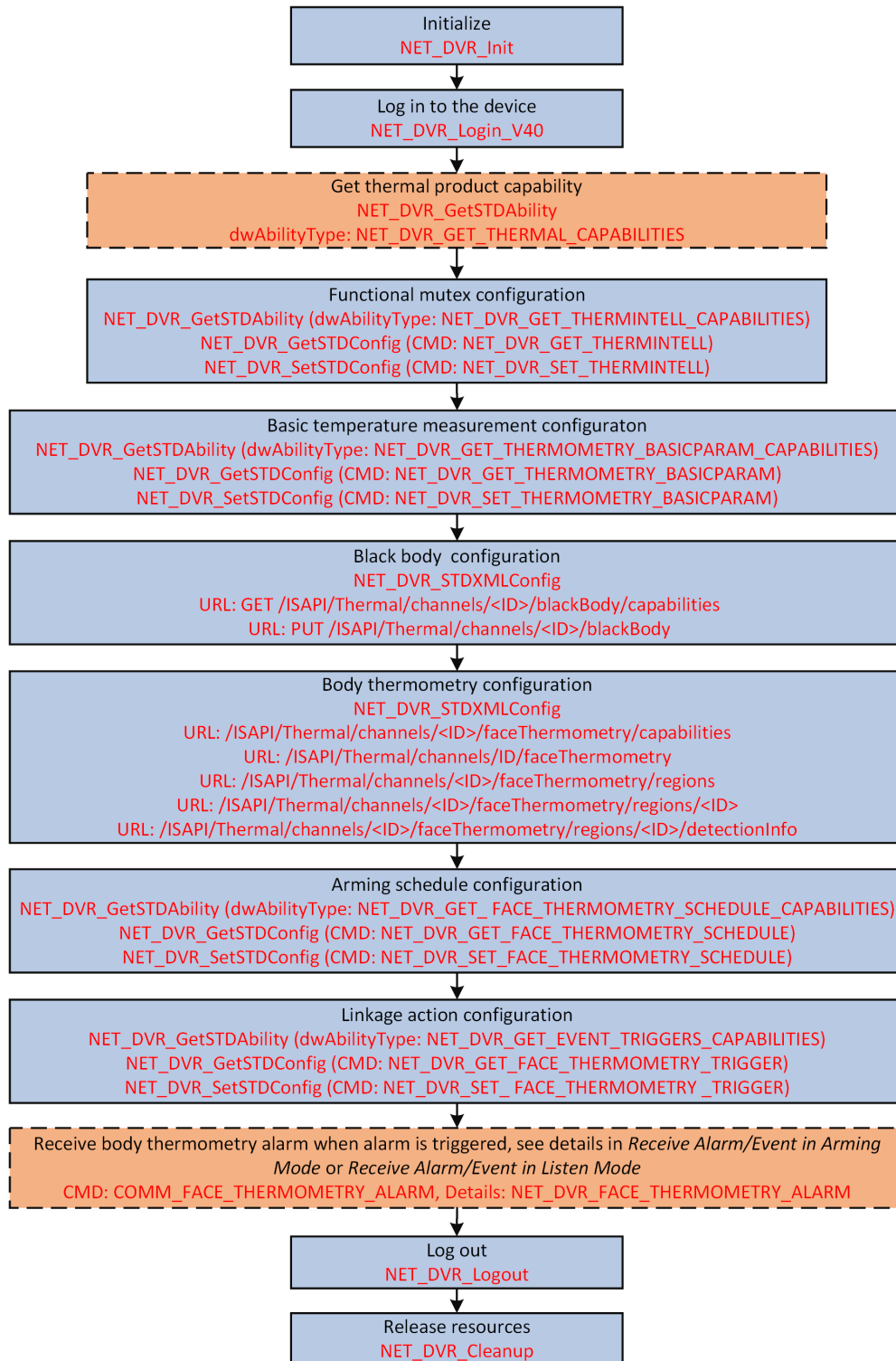


Figure 4-1 Programming Flow of Configuring Temperature Screening Alarm

1. **Optional:** Call **NET_DVR_GetSTDAbility** , set the **dwAbilityType** to **NET_DVR_GET_THERMAL_CAPABILITIES** (value: 3634), and set condition parameter **IpCondBuffer** in the structure of **NET_DVR_STD_ABILITY** to "NULL" for getting the thermal capability to check if the following functions are supported.

The thermal capability (**XML_ThermalCap**) is returned by the output parameter **IpOutBuffer** in the structure of **NET_DVR_STD_ABILITY** .
2. Configure functional mutex parameters.
 - 1) **Optional:** Call **NET_DVR_GetSTDAbility** , set the **dwAbilityType** to **NET_DVR_GET_THERMINTELL_CAPABILITIES** (value: 6711), and set condition parameter **IpCondBuffer** in the structure of **NET_DVR_STD_ABILITY** to 4-byte channel No. for getting the functional mutex capability to check if it is supported.

The functional mutex capability (**XML_Cap_ThermIntell**) is returned by the output parameter **IpOutBuffer** in the structure of **NET_DVR_STD_ABILITY** .
 - 2) **Optional:** Call **NET_DVR_GetSTDConfig** with **NET_DVR_GET_THERMINTELL** (command No.: 6712) and set the condition parameter **IpCondBuffer** in the structure of **NET_DVR_STD_CONFIG** to 4-byte channel No. for getting the existing or configured functional mutex parameters for reference.

The functional mutex parameters (**NET_DVR_THERMINTELL_PARAM**) are returned by the output parameter **IpOutBuffer** in the structure of **NET_DVR_STD_CONFIG** .
 - 3) Call **NET_DVR_SetSTDConfig** with **NET_DVR_SET_THERMINTELL** (command No.: 6713), set the condition parameter **IpCondBuffer** and input parameter **IpInBuffer** in the structure of **NET_DVR_STD_CONFIG** to 4-byte channel No. and **NET_DVR_THERMINTELL_PARAM** for setting functional mutex parameters.
3. Configure basic temperature measurement parameters.
 - 1) **Optional:** Call **NET_DVR_GetSTDAbility** , set the **dwAbilityType** to **NET_DVR_GET_THERMOMETRY_BASICPARAM_CAPABILITIES** (value: 3620), and set condition parameter **IpCondBuffer** in the structure of **NET_DVR_STD_ABILITY** to 4-byte channel No. for getting the basic temperature measurement capability to check if it is supported.

The basic temperature measurement capability (**XML_Cap_ThermometryBasicParam**) is returned by the output parameter **IpOutBuffer** in the structure of **NET_DVR_STD_ABILITY** .
 - 2) **Optional:** Call **NET_DVR_GetSTDConfig** with **NET_DVR_GET_THERMOMETRY_BASICPARAM** (command No.: 3621) and set the condition parameter **IpCondBuffer** in the structure of **NET_DVR_STD_CONFIG** to 4-byte channel No. for getting the existing or configured basic temperature measurement parameters for reference.

The basic temperature measurement parameters (**NET_DVR_THERMOMETRY_BASICPARAM**) are returned by the output parameter **IpOutBuffer** in the structure of **NET_DVR_STD_CONFIG** .
 - 3) Call **NET_DVR_SetSTDConfig** with **NET_DVR_SET_THERMOMETRY_BASICPARAM**

(command No.: 3622), set the condition parameter **IpCondBuffer** and input parameter **IpInBuffer** in the structure of **NET_DVR_STD_CONFIG** to 4-byte channel No. and **NET_DVR_THERMOMETRY_BASICPARAM** for setting basic temperature measurement parameters.

4. Configure black body as the reference of temperature screening.

- 1) **Optional:** Call **NET_DVR_STDXMLConfig** to transmit the request URI: GET **/ISAPI/Thermal/channels/<ID>/blackBody/capabilities** for getting the black body capability to check if it is supported.

The black body capability is returned in the message of **XML_Cap_ThermalBlackBody**.

- 2) **Optional:** Call **NET_DVR_STDXMLConfig** to transmit the request URI: GET **/ISAPI/Thermal/channels/<ID>/blackBody** for getting the existing or configured black body parameters for reference.

The black body parameters are returned in the message of **XML_ThermalBlackBody**.

- 3) Call **NET_DVR_STDXMLConfig** to transmit the request URI: PUT **/ISAPI/Thermal/channels/<ID>/blackBody** and set the request message to **XML_ThermalBlackBody** for setting the black body parameters.

5. Configure temperature screening parameters.

- 1) **Optional:** Call **NET_DVR_STDXMLConfig** to transmit the request URI: GET **/ISAPI/Thermal/channels/<ID>/faceThermometry/capabilities** for getting the temperature screening capability to check if it is supported.

The temperature screening capability is returned in the message of **XML_Cap_FaceThermometry**.

- 2) Call **NET_DVR_STDXMLConfig** to transmit the request URI: PUT **/ISAPI/Thermal/channels/<ID>/faceThermometry** and set the request message to **XML_FaceThermometry** for setting the temperature screening parameters.

- 3) Call **NET_DVR_STDXMLConfig** to transmit the request URIs: PUT **/ISAPI/Thermal/channels/<ID>/faceThermometry/regions** or PUT **/ISAPI/Thermal/channels/<ID>/faceThermometry/regions/<ID>**, and then set the request messages to **XML_FaceThermometryRegionList** and **XML_ThermometryRegion**, respectively, for setting temperature measurement rules of all detection regions or a specific detection region.

- 4) **Optional:** Call **NET_DVR_STDXMLConfig** to transmit the request URI: GET **/ISAPI/Thermal/channels/<ID>/faceThermometry/regions/<ID>/detectionInfo** for getting the temperature screening results by channel or by detection region.

The temperature screening results are returned in the message of **XML_FaceThermDetectionInfo** with form format (see the example in the message).



Note

Before setting the temperature screening parameters (include temperature measurement rules), you can call each configuration URI by GET method to get the existing or configured parameters for reference.

6. Configure arming schedule for temperature screening alarm.

- 1) **Optional:** Call **NET_DVR_GetSTDAbility** , set the **dwAbilityType** to **NET_DVR_GET_FACE_THERMOMETRY_SCHEDULE_CAPABILITIES** (value: 4402), and set condition parameter **IpCondBuffer** in the structure of **NET_DVR_STD_ABILITY** to 4-byte channel No. for getting the arming schedule configuration capability to check if it is supported.

The arming schedule configuration capability (**XML_Schedule**) is returned by the output parameter **IpOutBuffer** in the structure of **NET_DVR_STD_ABILITY** .

- 2) **Optional:** Call **NET_DVR_GetSTDConfig** with **NET_DVR_GET_FACE_THERMOMETRY_SCHEDULE** (command No.:4403) and set the condition parameter **IpCondBuffer** in the structure of **NET_DVR_STD_CONFIG** to 4-byte channel No. for getting the existing 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 of **NET_DVR_STD_CONFIG** .

- 3) Call **NET_DVR_SetSTDConfig** with **NET_DVR_SET_FACE_THERMOMETRY_SCHEDULE** (command No.: 4404), set the condition parameter **IpCondBuffer** and input parameter **IpInBuffer** in the structure of **NET_DVR_STD_CONFIG** to 4-byte channel No. and **NET_DVR_EVENT_SCHEDULE** for setting arming schedule.

7. Configure linkage action for temperature screening alarm.

- 1) **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 of **NET_DVR_STD_ABILITY** to "NULL" for getting the linkage action configuration capability to check if it is supported.

The linkage action configuration capability (**XML_EventTriggersCap**) is returned by the output parameter **IpOutBuffer** in the structure of **NET_DVR_STD_ABILITY** .

- 2) **Optional:** Call **NET_DVR_GetSTDConfig** with **NET_DVR_GET_FACE_THERMOMETRY_TRIGGER** (command No.: 4405) and set the condition parameter **IpCondBuffer** in the structure of **NET_DVR_STD_CONFIG** to 4-byte channel No. for getting the existing or configured linkage action parameters for reference.

The linkage action parameters (**NET_DVR_EVENT_TRIGGER**) are returned by the output parameter **IpOutBuffer** in the structure of **NET_DVR_STD_CONFIG** .

- 3) Call **NET_DVR_SetSTDConfig** with **NET_DVR_SET_FACE_THERMOMETRY_TRIGGER** (command No.: 4401), set the condition parameter **IpCondBuffer** and input parameter **IpInBuffer** in the structure of **NET_DVR_STD_CONFIG** to "NULL" and **NET_DVR_EVENT_TRIGGER** for setting linkage action.



Note

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

8. **Optional:** Receive the temperature screening 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 temperature screening alarm should be set to **COMM_FACE_THERMOMETRY_ALARM** (command No.: 0x4994) in the APIs of **NET_DVR_SetDVRMessageCallBack_V50** and **NET_DVR_StartListen_V30** . For the alarm details, refer to structure **NET_DVR_FACE_THERMOMETRY_ALARM** .

What to do next

Call **NET_DVR_Logout** and **NET_DVR_Cleanup** to log out and release the resources.

4.2 Configure Temperature Measurement Alarm in Normal Mode

Temperature measurement is mainly applied to the electric and industrial automation fields, which provides accurate and stable temperature measurement for the mid-to-high end manufacturers (e.g., intelligent robot) to realize the process control, automatic detection, and status monitoring of electromechanical devices. You can set the industry thermometry in normal mode to measure the global temperature during the industrial production. When the measured temperature does not meet the configured temperature threshold, the alarm will be triggered and uploaded to monitoring center if configured. And then, you can receive the 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

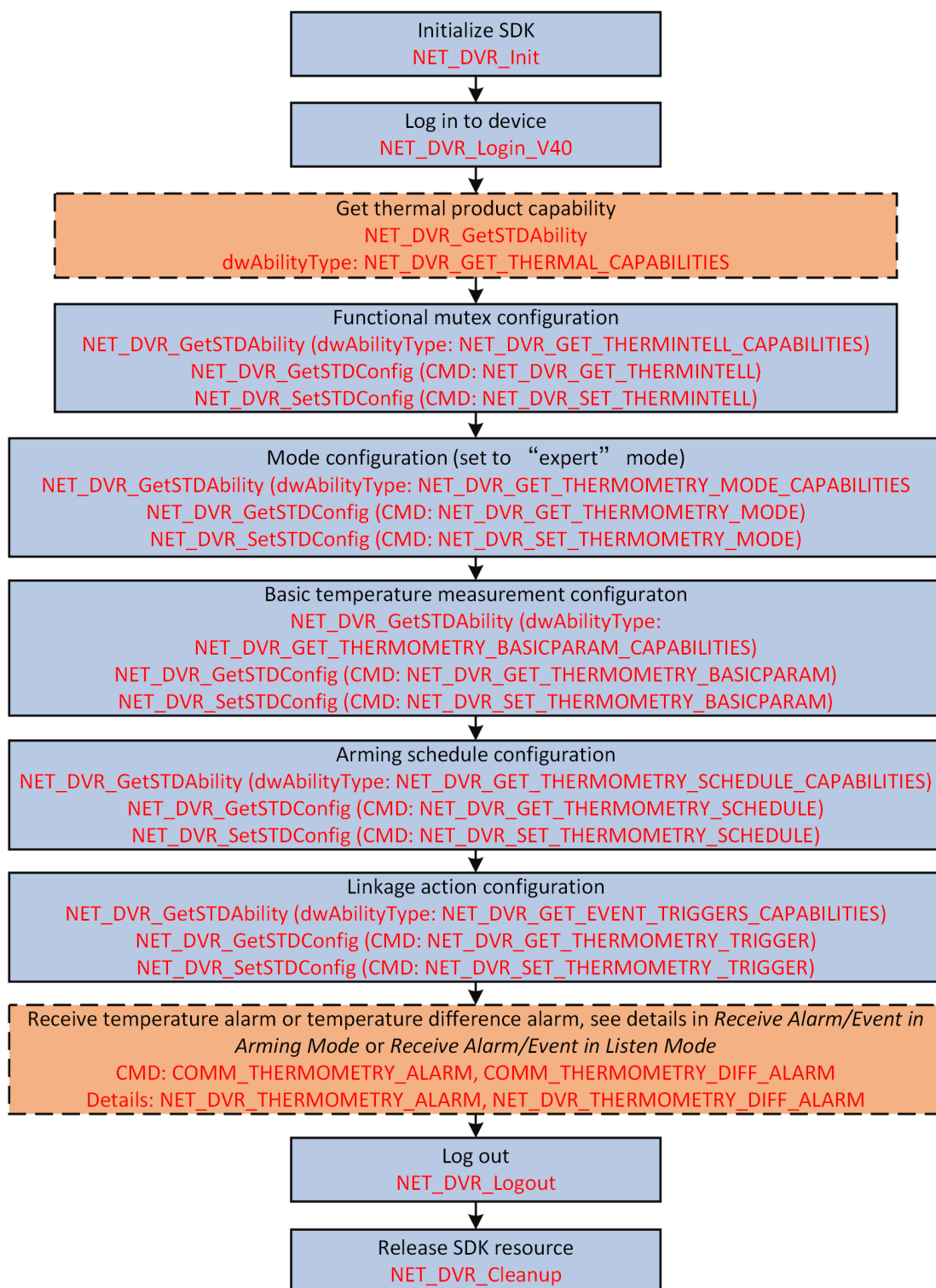


Figure 4-2 Programming Flow of Configuring Temperature Measurement Alarm in Normal Mode

1. **Optional:** Call **NET_DVR_GetSTDAbility** , set the **dwAbilityType** to **NET_DVR_GET_THERMAL_CAPABILITIES** (value: 3634), and set condition parameter **lpCondBuffer** in the structure of **NET_DVR_STD_ABILITY** to "NULL" for getting the thermal capability to check if the following functions are supported.
The thermal capability (**XML_ThermalCap**) is returned by the output parameter **lpOutBuffer** in the structure of **NET_DVR_STD_ABILITY** .
2. Configure functional mutex parameters.
 - 1) **Optional:** Call **NET_DVR_GetSTDAbility** , set the **dwAbilityType** to **NET_DVR_GET_THERMINTELL_CAPABILITIES** (value: 6711), and set condition parameter **lpCondBuffer** in the structure of **NET_DVR_STD_ABILITY** to 4-byte channel No. for getting the functional mutex capability to check if it is supported.
The functional mutex capability (**XML_Cap_ThermIntell**) is returned by the output parameter **lpOutBuffer** in the structure of **NET_DVR_STD_ABILITY** .
 - 2) **Optional:** Call **NET_DVR_GetSTDConfig** with **NET_DVR_GET_THERMINTELL** (command No.: 6712) and set the condition parameter **lpCondBuffer** in the structure of **NET_DVR_STD_CONFIG** to 4-byte channel No. for getting the existing or configured functional mutex parameters for reference.
The functional mutex parameters (**NET_DVR_THERMINTELL_PARAM**) are returned by the output parameter **lpOutBuffer** in the structure of **NET_DVR_STD_CONFIG** .
 - 3) Call **NET_DVR_SetSTDConfig** with **NET_DVR_SET_THERMINTELL** (command No.: 6713), set the condition parameter **lpCondBuffer** and input parameter **lpInBuffer** in the structure of **NET_DVR_STD_CONFIG** to 4-byte channel No. and **NET_DVR_THERMINTELL_PARAM** for setting functional mutex parameters.
3. Configure temperature measurement mode.
 - 1) **Optional:** Call **NET_DVR_GetSTDAbility** , set the **dwAbilityType** to **NET_DVR_GET_THERMOMETRY_MODE_CAPABILITIES** (value: 6764), and set condition parameter **lpCondBuffer** in the structure of **NET_DVR_STD_ABILITY** to 4-byte channel No. for getting the capability of temperature measurement mode configuration.
The capability of temperature measurement mode configuration (**XML_Cap_ThermometryMode**) is returned by the output parameter **lpOutBuffer** in the structure of **NET_DVR_STD_ABILITY** .
 - 2) **Optional:** Call **NET_DVR_GetSTDConfig** with **NET_DVR_GET_THERMOMETRY_MODE** (command No.: 6765) and set the condition parameter **lpCondBuffer** in the structure of **NET_DVR_STD_CONFIG** to 4-byte channel No. for getting the existing or configured temperature measurement mode for reference.
The temperature measurement mode (**NET_DVR_THERMOMETRY_MODE**) are returned by the output parameter **lpOutBuffer** in the structure of **NET_DVR_STD_CONFIG** .
 - 3) Call **NET_DVR_SetSTDConfig** with **NET_DVR_SET_THERMOMETRY_MODE**

(command No.: 6766), set the condition parameter **IpCondBuffer** and input parameter **IpInBuffer** in the structure of **NET_DVR_STD_CONFIG** to 4-byte channel No. and **NET_DVR_THERMOMETRY_MODE** for setting the temperature measurement mode to "normal".

4. Configure basic temperature measurement parameters.

- 1) **Optional:** Call **NET_DVR_GetSTDAbility** , set the **dwAbilityType** to **NET_DVR_GET_THERMOMETRY_BASICPARAM_CAPABILITIES** (value: 3620), and set condition parameter **IpCondBuffer** in the structure of **NET_DVR_STD_ABILITY** to 4-byte channel No. for getting the basic temperature measurement capability to check if it is supported.

The basic temperature measurement capability (**XML_Cap_ThermometryBasicParam**) is returned by the output parameter **IpOutBuffer** in the structure of **NET_DVR_STD_ABILITY** .

- 2) **Optional:** Call **NET_DVR_GetSTDConfig** with **NET_DVR_GET_THERMOMETRY_BASICPARAM** (command No.: 3621) and set the condition parameter **IpCondBuffer** in the structure of **NET_DVR_STD_CONFIG** to 4-byte channel No. for getting the existing or configured basic temperature measurement parameters for reference.

The basic temperature measurement parameters (**NET_DVR_THERMOMETRY_BASICPARAM**) are returned by the output parameter **IpOutBuffer** in the structure of **NET_DVR_STD_CONFIG** .

- 3) Call **NET_DVR_SetSTDConfig** with **NET_DVR_SET_THERMOMETRY_BASICPARAM** (command No.: 3622), set the condition parameter **IpCondBuffer** and input parameter **IpInBuffer** in the structure of **NET_DVR_STD_CONFIG** to 4-byte channel No. and **NET_DVR_THERMOMETRY_BASICPARAM** for setting basic temperature measurement parameters.

5. Configure arming schedule for temperature measurement alarm.

- 1) **Optional:** Call **NET_DVR_GetSTDAbility** , set the **dwAbilityType** to **NET_DVR_GET_THERMOMETRY_SCHEDULE_CAPABILITIES** (value: 6750), and set condition parameter **IpCondBuffer** in the structure of **NET_DVR_STD_ABILITY** to 4-byte channel No. for getting the arming schedule configuration capability to check if it is supported.

The arming schedule configuration capability (**XML_Schedule**) is returned by the output parameter **IpOutBuffer** in the structure of **NET_DVR_STD_ABILITY** .

- 2) **Optional:** Call **NET_DVR_GetSTDConfig** with **NET_DVR_GET_THERMOMETRY_SCHEDULE** (command No.: 6751) and set the condition parameter **IpCondBuffer** in the structure of **NET_DVR_STD_CONFIG** to 4-byte channel No. for getting the existing 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 of **NET_DVR_STD_CONFIG** .

- 3) Call **NET_DVR_SetSTDConfig** with **NET_DVR_SET_THERMOMETRY_SCHEDULE**

(command No.:6752), set the condition parameter **IpCondBuffer** and input parameter **IpInBuffer** in the structure of **NET_DVR_STD_CONFIG** to 4-byte channel No. and **NET_DVR_EVENT_SCHEDULE** for setting arming schedule.

6. Configure linkage action for temperature measurement alarm.

- 1) **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 of **NET_DVR_STD_ABILITY** to "NULL" for getting the linkage action configuration capability to check if it is supported.

The linkage action configuration capability (**XML_EventTriggersCap**) is returned by the output parameter **IpOutBuffer** in the structure of **NET_DVR_STD_ABILITY** .

- 2) **Optional:** Call **NET_DVR_GetSTDConfig** with **NET_DVR_GET_THERMOMETRY_TRIGGER**

(command No.: 3632) and set the condition parameter **IpCondBuffer** in the structure of **NET_DVR_STD_CONFIG** to **NET_DVR_THERMOMETRY_TRIGGER_COND** for getting the existing or configured linkage action parameters for reference.

The linkage action parameters (**NET_DVR_EVENT_TRIGGER**) are returned by the output parameter **IpOutBuffer** in the structure of **NET_DVR_STD_CONFIG** .

- 3) Call **NET_DVR_SetSTDConfig** with

NET_DVR_SET_THERMOMETRY_TRIGGER

(command No.: 3633), set the condition parameter **IpCondBuffer** and input parameter **IpInBuffer** in the structure of **NET_DVR_STD_CONFIG** to **NET_DVR_THERMOMETRY_TRIGGER_COND** and **NET_DVR_EVENT_TRIGGER** for setting linkage action.



Note

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

7. **Optional:** Receive the temperature measurement alarm (i.e., temperature alarm and temperature difference 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 related temperature measurement alarm should be set to **COMM_THERMOMETRY_ALARM**

(command No.: 0x5212) or

COMM_THERMOMETRY_DIFF_ALARM

(command No.: 0x5211) in the APIs of **NET_DVR_SetDVRMessageCallBack_V50** and

NET_DVR_StartListen_V30 . For alarm details, refer to the structure

NET_DVR_THERMOMETRY_ALARM or **NET_DVR_THERMOMETRY_DIFF_ALARM** .

What to do next

Call **NET_DVR_Logout** and **NET_DVR_Cleanup** to log out and release the resources.

4.3 Configure Temperature Measurement Alarm in Expert Mode

Temperature measurement is mainly applied to the electric and industrial automation fields, which provides accurate and stable temperature measurement for the mid-to-high end manufacturers (e.g., intelligent robot) to realize the process control, automatic detection, and status monitoring of electromechanical devices. You can set the temperature measurement to expert mode to measure the temperature according to areas or presets during the industrial production. When the measured temperature does not meet the configured temperature threshold, the alarm will be triggered and uploaded to monitoring center if configured. And then, you can receive the 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

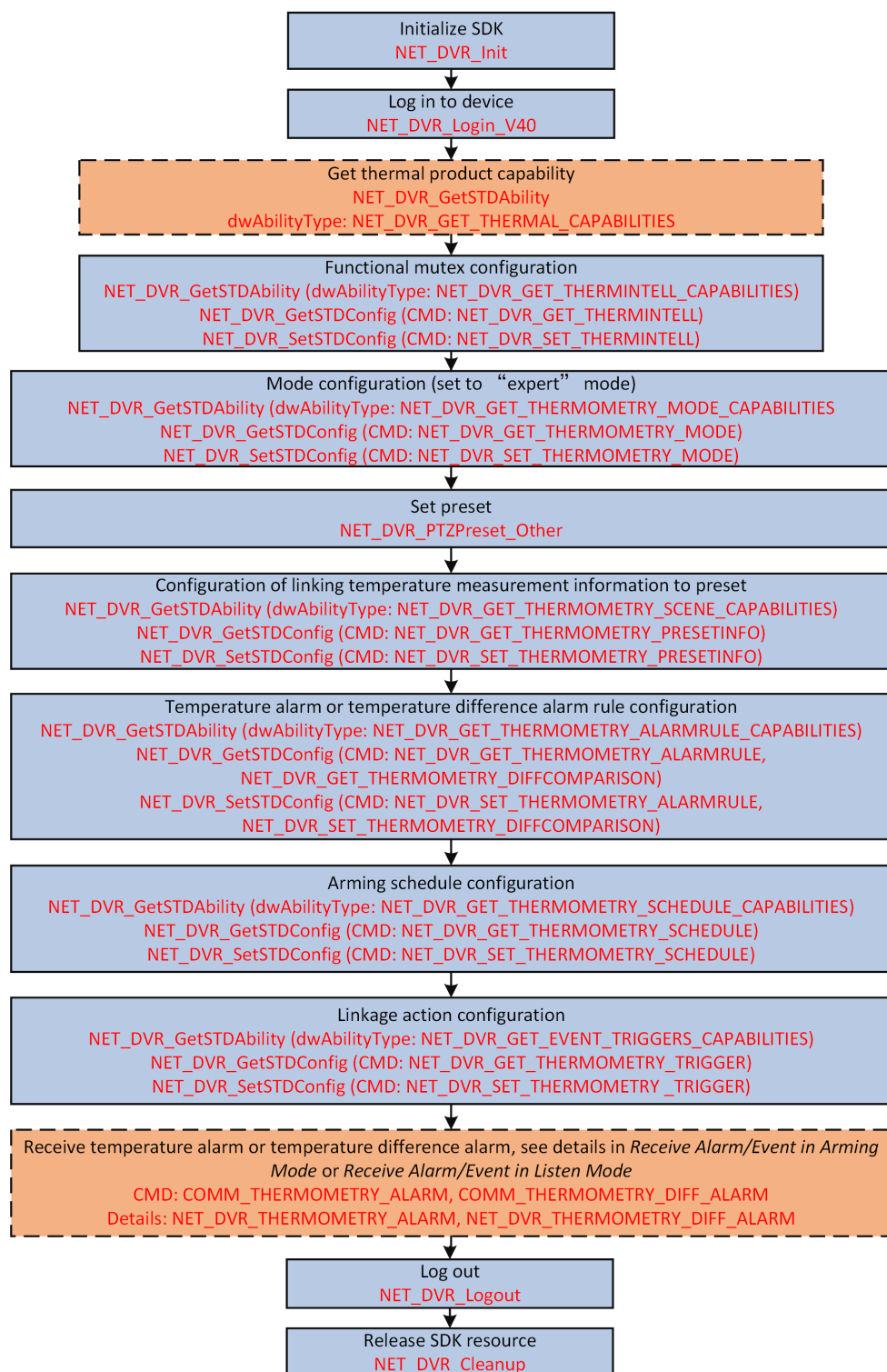


Figure 4-3 Programming Flow of Configuring Temperature Measurement Alarm in Expert Mode

1. **Optional:** Call **NET_DVR_GetSTDAbility** , set the **dwAbilityType** to **NET_DVR_GET_THERMAL_CAPABILITIES** (value: 3634), and set condition parameter **IpCondBuffer** in the structure of to "NULL" for getting the thermal capability to check if the following functions are supported.

The thermal capability (**XML_ThermalCap**) is returned by the output parameter **IpOutBuffer** in the structure of .
2. Configure functional mutex parameters.
 - 1) **Optional:** Call **NET_DVR_GetSTDAbility** , set the **dwAbilityType** to **NET_DVR_GET_THERMINTELL_CAPABILITIES** (value: 6711), and set condition parameter **IpCondBuffer** in the structure of to 4-byte channel No. for getting the functional mutex capability to check if it is supported.

The functional mutex capability (**XML_Cap_ThermIntell**) is returned by the output parameter **IpOutBuffer** in the structure of .
 - 2) **Optional:** Call **NET_DVR_GetSTDConfig** with **NET_DVR_GET_THERMINTELL** (command No.: 6712) and set the condition parameter **IpCondBuffer** in the structure of to 4-byte channel No. for getting the existing or configured functional mutex parameters for reference.

The functional mutex parameters () are returned by the output parameter **IpOutBuffer** in the structure of .
 - 3) Call **NET_DVR_SetSTDConfig** with **NET_DVR_SET_THERMINTELL** (command No.: 6713), set the condition parameter **IpCondBuffer** and input parameter **IpInBuffer** in the structure of to 4-byte channel No. and for setting functional mutex parameters.
3. Configure temperature measurement mode.
 - 1) **Optional:** Call **NET_DVR_GetSTDAbility** , set the **dwAbilityType** to **NET_DVR_GET_THERMOMETRY_MODE_CAPABILITIES** (value: 6764), and set condition parameter **IpCondBuffer** in the structure of to 4-byte channel No. for getting the capability of temperature measurement mode configuration.

The capability of temperature measurement mode configuration (XML_Cap_ThermometryMode) is returned by the output parameter **IpOutBuffer** in the structure of .
 - 2) **Optional:** Call **NET_DVR_GetSTDConfig** with **NET_DVR_GET_THERMOMETRY_MODE** (command No.: 6765) and set the condition parameter **IpCondBuffer** in the structure of to 4-byte channel No. for getting the existing or configured temperature measurement mode for reference.

The temperature measurement mode () are returned by the output parameter **IpOutBuffer** in the structure of .
 - 3) Call **NET_DVR_SetSTDConfig** with **NET_DVR_SET_THERMOMETRY_MODE**

(command No.: 6766), set the condition parameter **IpCondBuffer** and input parameter **IpInBuffer** in the structure of to 4-byte channel No. and for setting the temperature measurement mode to "expert".

4. Call **NET_DVR_PTZPreset_Other** to set a preset.

5. Configure linkage between temperature measurement information and preset.

- 1) **Optional:** Call **NET_DVR_GetSTDAbility** , set the **dwAbilityType** to **NET_DVR_GET_THERMOMETRY_SCENE_CAPABILITIES** (value: 3623), and set condition parameter **IpCondBuffer** in the structure of to 4-byte channel No. for getting the configuration capability of linkage between temperature measurement information and preset.

The configuration capability of linkage between temperature measurement information and preset (**XML_Cap_ThermometryScene**) is returned by the output parameter **IpOutBuffer** in the structure of .

- 2) **Optional:** Call **NET_DVR_GetSTDConfig** with **NET_DVR_GET_THERMOMETRY_PRESETINFO**

(command No.: 3624) and set the condition parameter **IpCondBuffer** in the structure of to for getting the existing or configured linkage parameters between temperature measurement information and preset for reference.

The linkage parameters () are returned by the output parameter **IpOutBuffer** in the structure of .

- 3) Call **NET_DVR_SetSTDConfig** with **NET_DVR_SET_THERMOMETRY_PRESETINFO**

(command No.: 3625), set the condition parameter **IpCondBuffer** and input parameter **IpInBuffer** in the structure of to and for linking the temperature measurement information to the preset.

6. Configure rules of temperature alarm or temperature difference alarm.

- a. Call **NET_DVR_GetSTDAbility** , set the **dwAbilityType** to **NET_DVR_GET_THERMOMETRY_ALARMRULE_CAPABILITIES** (value: 3626), and set condition parameter **IpCondBuffer** in the structure of to 4-byte channel No. for getting alarm rule configuration capability.



The alarm rule configuration capability (**XML_Cap_ThermometryAlarmRule**) is returned by the output parameter **IpOutBuffer** in the structure of .

- b. Call **NET_DVR_GetSTDConfig** with **NET_DVR_GET_THERMOMETRY_ALARMRULE**

(command No.: 3627) and set the condition parameter **IpCondBuffer** in the structure of to for getting the existing or configured temperature alarm rules.



The temperature alarm rules () are returned by the output parameter **IpOutBuffer** in the structure of .

- c. Call **NET_DVR_SetSTDConfig** with

NET_DVR_SET_THERMOMETRY_ALARMRULE

(command No.: 3628), set the condition parameter **IpCondBuffer** and input parameter **IpInBuffer** in the structure of to and for setting the temperature alarm rules.

- a. Call **NET_DVR_GetSTDAbility** , set the **dwAbilityType** to **NET_DVR_GET_THERMOMETRY_ALARMRULE_CAPABILITIES** (value: 3626), and set condition parameter **IpCondBuffer** in the structure of to 4-byte channel No. for getting alarm rule configuration capability.



Note

The alarm rule configuration capability (**XML_Cap_ThermometryAlarmRule**) is returned by the output parameter **IpOutBuffer** in the structure of .

- b. Call **NET_DVR_GetSTDConfig** with **NET_DVR_GET_THERMOMETRY_DIFFCOMPARISON** (command No.: 3630) and set the condition parameter **IpCondBuffer** in the structure of to for getting the existing or configured temperature difference alarm rules.



Note

The temperature difference alarm rules () are returned by the output parameter **IpOutBuffer** in the structure of .

- c. Call **NET_DVR_SetSTDConfig** with **NET_DVR_SET_THERMOMETRY_DIFFCOMPARISON** (command No.: 3631), set the condition parameter **IpCondBuffer** and input parameter **IpInBuffer** in the structure of to and for setting the temperature difference alarm rules.

7. Configure arming schedule for temperature measurement alarm.

- 1) **Optional:** Call **NET_DVR_GetSTDAbility** , set the **dwAbilityType** to **NET_DVR_GET_THERMOMETRY_SCHEDULE_CAPABILITIES** (value: 6750), and set condition parameter **IpCondBuffer** in the structure of to 4-byte channel No. for getting the arming schedule configuration capability to check if it is supported.

The arming schedule configuration capability (**XML_Schedule**) is returned by the output parameter **IpOutBuffer** in the structure of .

- 2) **Optional:** Call **NET_DVR_GetSTDConfig** with **NET_DVR_GET_THERMOMETRY_SCHEDULE** (command No.: 6751) and set the condition parameter **IpCondBuffer** in the structure of to 4-byte channel No. for getting the existing or configured arming schedule parameters for reference.

The arming schedule parameters () are returned by the output parameter **IpOutBuffer** in the structure of .

- 3) Call **NET_DVR_SetSTDConfig** with **NET_DVR_SET_THERMOMETRY_SCHEDULE** (command No.:6752), set the condition parameter **IpCondBuffer** and input parameter **IpInBuffer** in the structure of to 4-byte channel No. and for setting arming schedule.

8. Configure linkage action for temperature measurement alarm.

- 1) **Optional:** Call **NET_DVR_GetSTDAbility** , set the **dwAbilityType** to **NET_DVR_GET_EVENT_TRIGGERS_CAPABILITIES** (value: 3501), and set condition parameter **lpCondBuffer** in the structure of to "NULL" for getting the linkage action configuration capability to check if it is supported.
The linkage action configuration capability (**XML_EventTriggersCap**) is returned by the output parameter **lpOutBuffer** in the structure of .
- 2) **Optional:** Call **NET_DVR_GetSTDConfig** with **NET_DVR_GET_THERMOMETRY_TRIGGER** (command No.: 3632) and set the condition parameter **lpCondBuffer** in the structure of to for getting the existing or configured linkage action parameters for reference.
The linkage action parameters () are returned by the output parameter **lpOutBuffer** in the structure of .
- 3) Call **NET_DVR_SetSTDConfig** with **NET_DVR_SET_THERMOMETRY_TRIGGER** (command No.: 3633), set the condition parameter **lpCondBuffer** and input parameter **lpInBuffer** in the structure of to and for setting linkage action.



Note

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

9. **Optional:** Receive the temperature measurement alarm (i.e., temperature alarm and temperature difference 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 related temperature measurement alarm should be set to **COMM_THERMOMETRY_ALARM** (command No.: 0x5212) or **COMM_THERMOMETRY_DIFF_ALARM** (command No.: 0x5211) in the APIs of **NET_DVR_SetDVRMessageCallback_V50** and **NET_DVR_StartListen_V30** . For alarm details, refer to the structure or .

What to do next

Call **NET_DVR_Logout** and **NET_DVR_Cleanup** to log out and release the resources.

4.4 Configure Fire and Smoke Alarm

In the actual fire detection, the fire source may be concealed or the distance is too far to be detected, so the fire alarm may failed to be triggered sometimes. To avoid the above situation, you can configure the fire and smoke alarm, which includes fire source detection and smoke detection, once the fire source or smoke is detected, the alarm will be 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

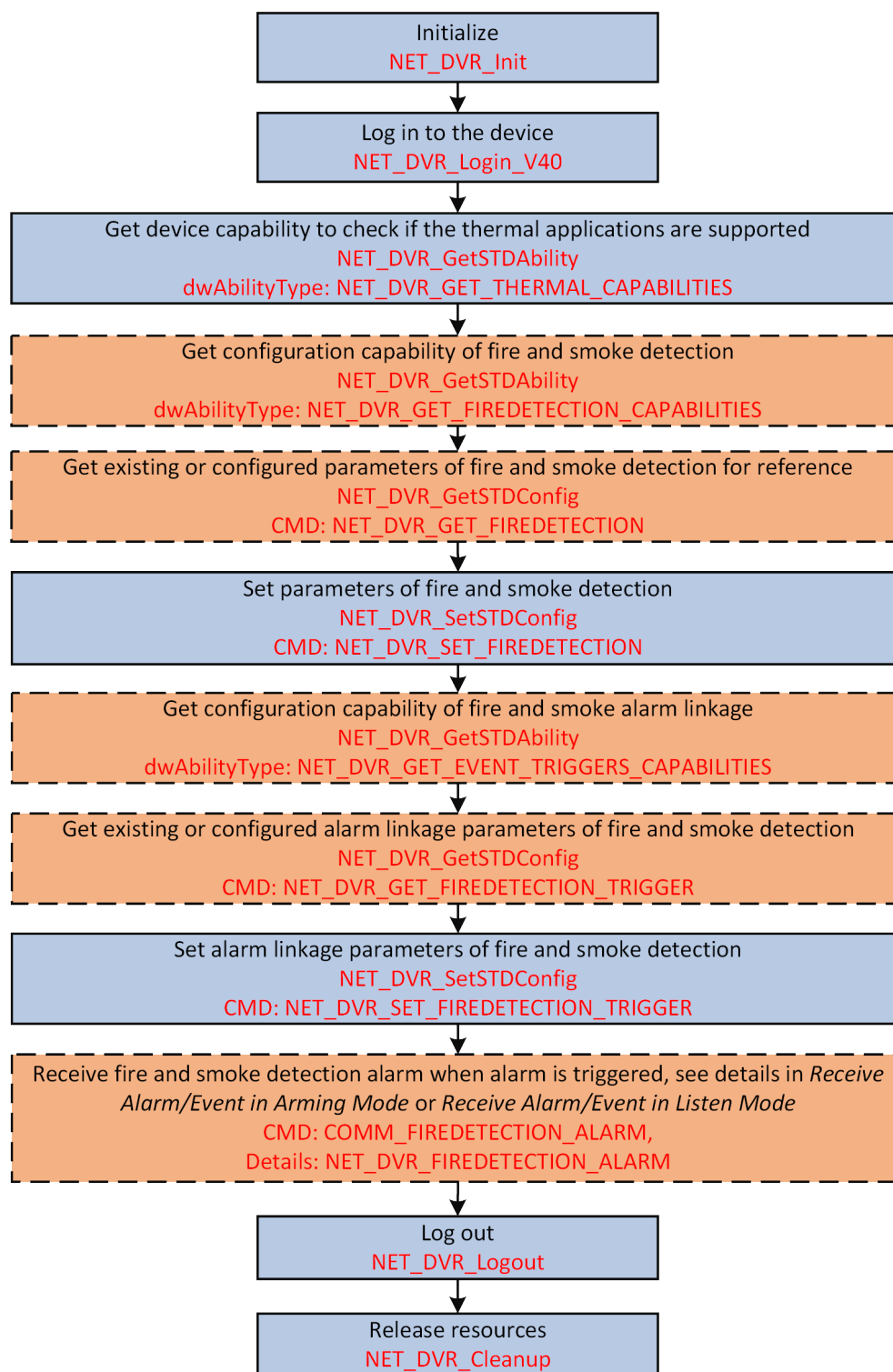


Figure 4-4 Programming Flow of Configuring Fire and Smoke Detection Alarm

1. Call **NET_DVR_GetSTDAbility** and set **dwAbilityType** to "NET_DVR_GET_THERMAL_CAPABILITIES" (value: 3634) for getting the thermal capability.
The thermal capability is returned in the message **XML_ThermalCap** by **lpOutBuffer** of **NET_DVR_STD_ABILITY**.
If the node **<isSupportFireDetection>** is returned in the thermal capability and its value is "true", it indicates that the device supports fire and smoke detection, and you can perform the following steps.
Otherwise, the fire and smoke detection is not supported by the device, you should end this task.
2. **Optional:** Perform one of the following operations to get configuration capability of fire and smoke detection.
 - Call **NET_DVR_GetSTDAbility**, set **dwAbilityType** to "NET_DVR_GET_FIREDETECTION_CAPABILITIES" (value: 3635), and set **lpCondBuffer** in **NET_DVR_STD_ABILITY** to 4-byte channel No.
The configuration capability of fire and smoke detection is returned in the message **XML_Cap_FireDetection** by **lpOutBuffer** of **NET_DVR_STD_ABILITY**.
 - Call **NET_DVR_STDXMLConfig** to transmit GET **/ISAPI/Thermal/channels/<ID>/fireDetection/capabilities**.
The capability is returned in **XML_Cap_FireDetection** by **lpOutBuffer**.
3. **Optional:** Perform one of the following operations to get the default or configured fire and smoke detection parameters for reference.
 - Call **NET_DVR_GetSTDConfig** with "NET_DVR_GET_FIREDETECTION" (command No.: 3636) and set **lpCondBuffer** in **NET_DVR_STD_CONFIG** to 4-byte channel No.
The fire and smoke detection parameters is returned in the structure **NET_DVR_FIREDETECTION_CFG** by **lpOutBuffer** of **NET_DVR_STD_CONFIG**.
 - Call **NET_DVR_STDXMLConfig** to transmit GET **/ISAPI/Thermal/channels/<ID>/fireDetection** and set **lpInBuffer** to "NULL".
The fire detection parameters are returned in **XML_FireDetection** by **lpOutBuffer**.
4. Perform one of the following operations to set the fire and smoke detection parameters.
 - Call **NET_DVR_SetSTDConfig** with "NET_DVR_SET_FIREDETECTION" (command No.: 3637), and set **lpCondBuffer** and **lpInBuffer** of **NET_DVR_STD_CONFIG** to 4-byte channel No and **NET_DVR_FIREDETECTION_CFG**, respectively.
 - Call **NET_DVR_STDXMLConfig** to transmit PUT **/ISAPI/Thermal/channels/<ID>/fireDetection** and set **lpInBuffer** to **XML_FireDetection**.
5. **Optional:** Call **NET_DVR_GetSTDAbility** and set **dwAbilityType** to "NET_DVR_GET_EVENT_TRIGGERS_CAPABILITIES" (value: 3501) for getting the configuration capability of fire and smoke alarm linkage.
The configuration capability of fire and smoke alarm linkage is returned in the message **XML_EventTriggersCap** by **lpOutBuffer** of **NET_DVR_STD_ABILITY**.

6. **Optional:** Call **NET_DVR_GetSTDConfig** with "NET_DVR_GET_FIREDETECTION_TRIGGER" (command No.: 3638) and set **lpCondBuffer** of **NET_DVR_STD_CONFIG** to 4-byte channel No. for getting the configured or default parameters of fire and smoke alarm linkage for reference. The parameters of fire and smoke alarm linkage is returned in the structure **NET_DVR_EVENT_TRIGGER** by **lpOutBuffer** of **NET_DVR_STD_CONFIG**.
7. Call **NET_DVR_SetSTDConfig** with "NET_DVR_SET_FIREDETECTION_TRIGGER" (command No.: 3639), and set **lpCondBuffer** and **lpInBuffer** of **NET_DVR_STD_CONFIG** to 4-byte channel No and **NET_DVR_EVENT_TRIGGER**, respectively, for setting fire and smoke alarm linkage.



Note

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

-
8. Set **lCommand** in alarm callback function (**MSGCallBack**) to "COMM_FIREDETECTION_ALARM" (command No.: 0x4991) and receive fire and smoke 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 fire and smoke alarm details is returned in the structure

NET_DVR_FIREDETECTION_ALARM.

What to do next

Call **NET_DVR_Logout** and **NET_DVR_Cleanup** to log out and release the resources.

4.5 Configure Ship Flow Detection Alarm

Ship flow detection can calculate the amount of ships passing through a predefined area, and triggers an alarm when the ship crosses the predefined line. It is widely applied to river monitoring under severe weather conditions, such as heavy rain, heavy fog, and so on.

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

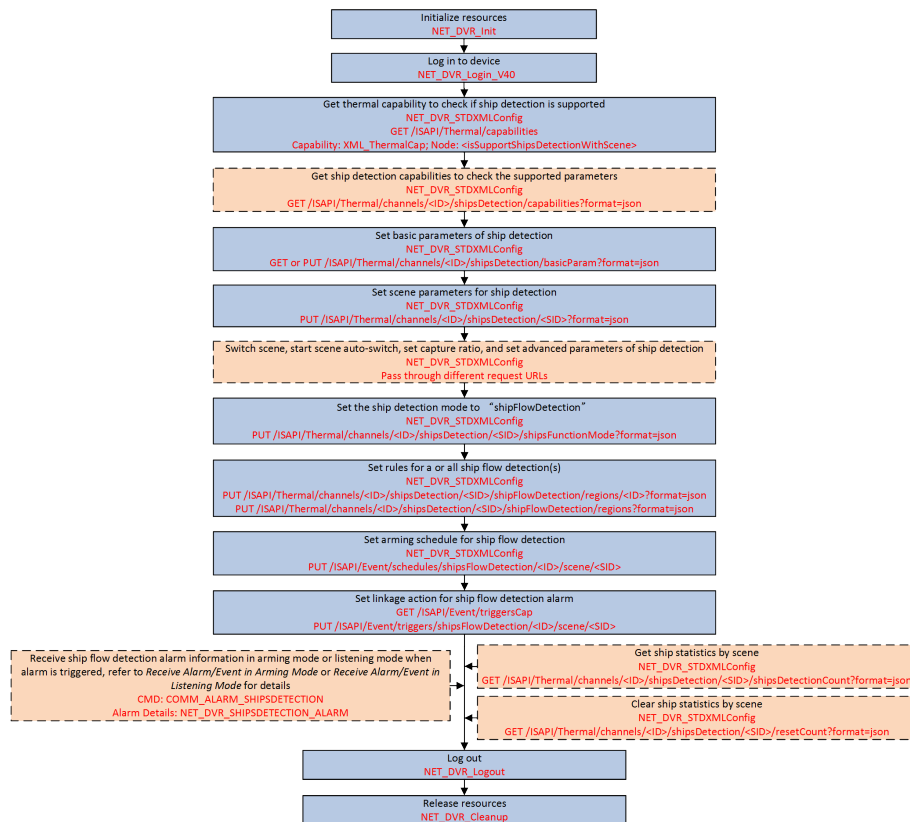


Figure 4-5 Programming Flow of Configuring Ship Flow Detection Alarm

1. Call **NET_DVR_STDXMLConfig** to transmit the request URI: **GET /ISAPI/Thermal/capabilities** for getting the thermal capability to check if the device supports ship detection by scene. The thermal capability is returned in the message **XML_ThermalCap** by **IpOutBuffer**. If supports, the node **<isSupportShipsDetectionWithScene>** is returned in the message and its value is "true", and then you can perform the following steps. Otherwise, ship detection by scene is not supported by device, please end this task.
2. **Optional:** Call **NET_DVR_STDXMLConfig** to transmit the request URI: **GET /ISAPI/Thermal/channels/<ID>/shipsDetection/capabilities?format=json** for getting the ship detection capability to check the supported parameters. The ship detection capability is returned in the message **JSON_shipsDetectionCap** by **IpOutBuffer**.
3. Set basic parameters of ship detection.
 - 1) **Optional:** Call **NET_DVR_STDXMLConfig** to transmit the request URI: **GET /ISAPI/Thermal/channels/<ID>/shipsDetection/basicParam/capabilities?format=json** for getting basic configuration capability of ship detection to check the supported parameters. The basic configuration capability of ship detection is returned in the message **JSON_basicParamCap** by **IpOutBuffer**.

- 2) **Optional:** Call **NET_DVR_STDXMLConfig** to transmit the request URI: GET **/ISAPI/Thermal/channels/<ID>/shipsDetection/basicParam?format=json** for getting default or configured basic parameters of ship detection for reference.
The basic parameters of ship detection are returned in the message **JSON_basicParam** by **lpOutBuffer**.
- 3) Call **NET_DVR_STDXMLConfig** to transmit the request URI: PUT **/ISAPI/Thermal/channels/<ID>/shipsDetection/basicParam?format=json** and set **lpInBuffer** to **JSON_basicParam** for setting basic parameters for ship detection.
4. Set scene parameters for ship detection.
 - 1) **Optional:** Call **NET_DVR_STDXMLConfig** to transmit the request URI: GET **/ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/capabilities?format=json** for getting scene configuration capability.
The scene configuration capability is returned in the message **JSON_ShipsDetectionSceneCap** by **lpOutBuffer**.
 - 2) **Optional:** Call **NET_DVR_STDXMLConfig** to transmit the request URI: GET **/ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>?format=json** or **/ISAPI/Thermal/channels/<ID>/shipsDetection?format=json** for getting default or configured parameters of a or all scene(s) for reference.
The parameters of a or all scene(s) are returned in the message **JSON_ShipsDetectionScene** or **JSON_ShipsDetectionSceneList** by **lpOutBuffer**.
 - 3) Call **NET_DVR_STDXMLConfig** to transmit the request URI: PUT **/ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>?format=json** or **/ISAPI/Thermal/channels/<ID>/shipsDetection?format=json** for creating a or all scenes(s).
 - 4) Call **NET_DVR_STDXMLConfig** to transmit the request URI: PUT **/ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>?format=json** or **/ISAPI/Thermal/channels/<ID>/shipsDetection?format=json** and set **lpInBuffer** to **JSON_ShipsDetectionScene** or **JSON_ShipsDetectionSceneList** for setting parameters of a or all scenes.
 - 5) **Optional:** Call **NET_DVR_STDXMLConfig** to transmit the request URI: DELETE **/ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>?format=json** or **/ISAPI/Thermal/channels/<ID>/shipsDetection?format=json** for deleting a or all scenes(s).
5. **Optional:** Perform the following operation(s) to switch scene, start scene auto-switch, set capture ratio for scene, and set advanced parameters for ship detection.

Switch Scene Call **NET_DVR_STDXMLConfig** to transmit the request URI: PUT **/ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/goto?format=json**

Scene Auto-Switch a. Call **NET_DVR_STDXMLConfig** to transmit the request URI: GET **/ISAPI/Thermal/channels/<ID>/shipsDetection/sceneTrace/capabilities?format=json** for getting scene auto-switch capability.
b. Call **NET_DVR_STDXMLConfig** to transmit the request URI: GET **/ISAPI/Thermal/channels/<ID>/shipsDetection/sceneTrace?format=json** for

getting default or configured scene auto-switch parameters for reference.

- c. Call **NET_DVR_STDXMLConfig** to transmit the request URI: **PUT /ISAPI/Thermal/channels/<ID>/shipsDetection/sceneTrace?format=json** and set **lpInBuffer** to **JSON_ShipsDetectionSceneTraceList** for starting scene auto-switch.

Capture Ratio

- a. Call **NET_DVR_STDXMLConfig** to transmit the request URI: **GET /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/captureRatio/capabilities?format=json** for getting configuration capability of capture ratio.
- b. Call **NET_DVR_STDXMLConfig** to transmit the request URI: **GET /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/captureRatio?format=json** for getting default or configured capture ratio for reference.
- c. Call **NET_DVR_STDXMLConfig** to transmit the request URI: **PUT /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/captureRatio?format=json** and set **lpInBuffer** to **JSON_ShipsDetectionCaptureRatio** for setting capture ratio.

Advanced Configuration

- a. Call **NET_DVR_STDXMLConfig** to transmit the request URI: **GET /ISAPI/Thermal/channels/<ID>/shipsDetection/advanceParam/capabilities?format=json** for getting advanced configuration capability of ship detection to check the supported parameters.
- b. Call **NET_DVR_STDXMLConfig** to transmit the request URI: **GET /ISAPI/Thermal/channels/<ID>/shipsDetection/advanceParam?format=json** for getting default or configured advanced parameters of ship detection for reference.
- c. Call **NET_DVR_STDXMLConfig** to transmit the request URI: **PUT /ISAPI/Thermal/channels/<ID>/shipsDetection/advanceParam?format=json** and set **lpInBuffer** to **JSON_advanceParam** for setting advanced parameters of ship detection.

6. Set function mode to "shipFlowDetection" to detect the ship flow.

- 1) **Optional:** Call **NET_DVR_STDXMLConfig** to transmit the request URI: **GET /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/shipsFunctionMode/capabilities?format=json** for getting function mode capability to check the supported modes.

The function mode capability is returned in the message **JSON_Cap_functionMode** by **lpOutBuffer**.

- 2) **Optional:** Call **NET_DVR_STDXMLConfig** to transmit the request URI: **GET /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/shipsFunctionMode?format=json** for getting default or configured function mode for reference.

The function mode is returned in the message **JSON_functionMode** by **lpOutBuffer**.

- 3) Call **NET_DVR_STDXMLConfig** to transmit the request URI: PUT **/ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/shipsFunctionMode?format=json** and set **lpInBuffer** to **JSON_functionMode** for setting the function mode to "shipFlowDetection".
7. Set ship flow detection rules.
 - 1) **Optional:** Call **NET_DVR_STDXMLConfig** to transmit the request URI: GET **/ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/shipFlowDetection/regions/capabilities?format=json** for getting rule configuration capability of ship flow detection to check the supported rule parameters.
 The rule configuration capability of ship flow detection is returned in the message **JSON_ShipFlowDetectionRuleListCap** by **lpOutBuffer**.
 - 2) **Optional:** Call **NET_DVR_STDXMLConfig** to transmit the request URIs: GET **/ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/shipFlowDetection/regions/<ID>?format=json** or **/ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/shipFlowDetection/regions?format=json** for getting default or configured rules of a or all ship flow detection(s) for reference.
 The rules of a or all ship flow detection(s) are returned in the message **JSON_ShipFlowDetectionRule** or **JSON_ShipFlowDetectionRuleList** by **lpOutBuffer**.
 - 3) Call **NET_DVR_STDXMLConfig** to transmit the request URIs: PUT **/ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/shipFlowDetection/regions/<ID>?format=json** or **/ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/shipFlowDetection/regions?format=json** and set **lpInBuffer** to **JSON_ShipFlowDetectionRule** or **JSON_ShipFlowDetectionRuleList** for setting rules of a or all ship flow detection(s).
8. Call **NET_DVR_STDXMLConfig** to transmit the request URI: PUT **/ISAPI/Event/schedules/shipsFlowDetection/<ID>/scene/<SID>** for setting the arming schedule of ship flow detection.



Note

Before setting arming schedule, you can transmit this URI by GET method to get default or configured arming schedule for reference.

9. Call **NET_DVR_STDXMLConfig** to transmit the request URI: PUT **/ISAPI/Event/triggers/shipsFlowDetection/<ID>/scene/<SID>** for setting linkage action of ship flow detection alarm.



Note

- Before setting linkage action, you can transmit this URI by GET method to get default or configured linkage actions for reference.
- To receive the alarm via platform, the linkage action must be set to "center".

10. **Optional:** Perform the following operation(s) after configuring ship flow detection alarm.

**Get Ship
Statistics by
Scene**

Call **NET_DVR_STDXMLConfig** to transmit the request URI: GET **/ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/shipsDetectionCount?format=json**

Clear Ship Statistics by Scene	Call <i>NET_DVR_STDXMLConfig</i> to transmit the request URI: <i>PUT /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/resetCount?format=json</i>
Receive Alarm in Arming or Listening Mode	Perform the flow of <i>Receive Alarm/Event in Arming Mode</i> or <i>Receive Alarm/Event in Listening Mode</i> and set lCommand in the registered alarm callback function (<i>MSGCallback</i>) to "COMM_ALARM_SHIPSDTECTION" (command No.: 0x4521).



Note

The ship flow detection alarm information is returned in the structure ***NET_DVR_SHIPSDTECTION_ALARM*** by **pAlarmInfo** of alarm callback function.

What to do next

Call ***NET_DVR_Logout*** and ***NET_DVR_Cleanup*** to log out and release the resources.

4.6 Configure Dredger Detection Alarm

Dredger detection can detect the ship staying time to distinguish if the ship is dredger, when the ship staying time exceeds the configured threshold, the dredger detection alarm will be triggered. It is widely applied to river monitoring under severe weather conditions, such as heavy rain, heavy fog, and so on.

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

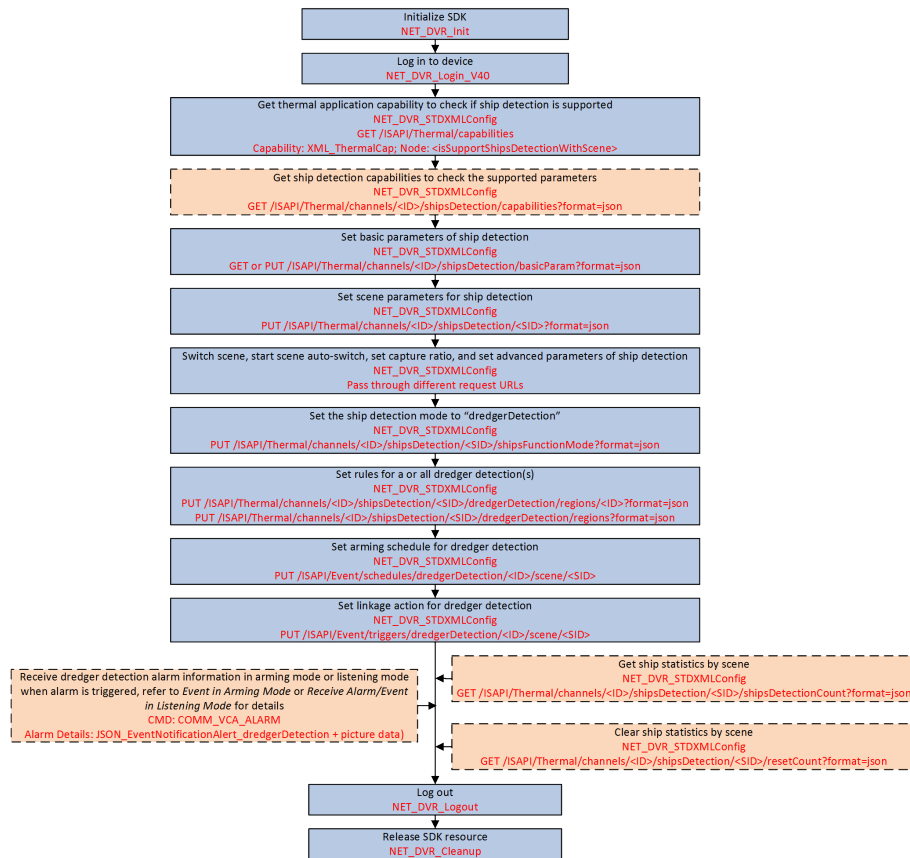


Figure 4-6 Programming Flow of Configuring Dredger Detection Alarm

1. Call **NET_DVR_STDXMLConfig** to transmit the request URI: **GET /ISAPI/Thermal/capabilities** for getting the thermal capability to check if the device supports ship detection by scene.
The thermal capability is returned in the message **XML_ThermalCap** by **IpOutBuffer**.
If supports, the node **<isSupportShipsDetectionWithScene>** is returned in the message and its value is "true", and then you can perform the following steps.
Otherwise, ship detection by scene is not supported by device, please end this task.
2. **Optional:** Call **NET_DVR_STDXMLConfig** to transmit the request URI: **GET /ISAPI/Thermal/channels/<ID>/shipsDetection/capabilities?format=json** for getting the ship detection capability to check the supported parameters.
The ship detection capabilities are returned in the message **JSON_shipsDetectionCap** by **IpOutBuffer**.
3. Set basic parameters of ship detection.
 - 1) Call **NET_DVR_STDXMLConfig** to transmit the request URI: **GET /ISAPI/Thermal/channels/<ID>/shipsDetection/basicParam/capabilities?format=json** for getting basic configuration capability of ship detection to check the supported parameters.
The basic configuration capability of ship detection is returned in the message **JSON_basicParamCap** by **IpOutBuffer**.

- 2) Call **NET_DVR_STDXMLConfig** to transmit the request URI: GET */ISAPI/Thermal/channels/<ID>/shipsDetection/basicParam?format=json* for getting default or configured basic parameters of ship detection for reference.
The basic parameters of ship detection are returned in the message **JSON_basicParam** by **lpOutBuffer**.
- 3) Call **NET_DVR_STDXMLConfig** to transmit the request URI: PUT */ISAPI/Thermal/channels/<ID>/shipsDetection/basicParam?format=json* for setting basic parameters of ship detection.
4. Set scene parameters for ship detection.
 - 1) **Optional:** Call **NET_DVR_STDXMLConfig** to transmit the request URI: GET */ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/capabilities?format=json* for getting scene configuration capability.
The scene configuration capability is returned in the **JSON_ShipsDetectionSceneCap** by **lpOutBuffer**.
 - 2) **Optional:** Call **NET_DVR_STDXMLConfig** to transmit the request URI: GET */ISAPI/Thermal/channels/<ID>/shipsDetection/basicParam?format=json* or */ISAPI/Thermal/channels/<ID>/shipsDetection?format=json* for getting default or configured scene parameters of a or all scene(s) for reference
The parameters of a or all scene(s) are returned in the message **JSON_ShipsDetectionScene** or **JSON_ShipsDetectionSceneList** by **lpOutBuffer**.
 - 3) Call **NET_DVR_STDXMLConfig** to transmit the request URI: PUT */ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>?format=json* or */ISAPI/Thermal/channels/<ID>/shipsDetection?format=json* for creating a or all scenes(s).
 - 4) Call **NET_DVR_STDXMLConfig** to transmit the request URI: PUT */ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>?format=json* or */ISAPI/Thermal/channels/<ID>/shipsDetection?format=json* and set **lpInBuffer** to **JSON_ShipsDetectionScene** or **JSON_ShipsDetectionSceneList** for setting parameters of a or all scenes.
 - 5) **Optional:** Call **NET_DVR_STDXMLConfig** to transmit the request URI: DELETE */ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>?format=json* or */ISAPI/Thermal/channels/<ID>/shipsDetection?format=json* for deleting a or all scenes(s).
5. **Optional:** Perform the following operation(s) to switch scene, start auto-switch, set capture ratio for scene, and set advanced parameters for ship detection.

Switch Scene	Call NET_DVR_STDXMLConfig to transmit the request URI: PUT <i>/ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/goto?format=json</i>
Scene Auto-Switch	<ol style="list-style-type: none"> a. Call NET_DVR_STDXMLConfig to transmit the request URI: GET <i>/ISAPI/Thermal/channels/<ID>/shipsDetection/sceneTrace/capabilities?format=json</i> for getting scene auto-switch capability. b. Call NET_DVR_STDXMLConfig to transmit the request URI: GET <i>/ISAPI/Thermal/channels/<ID>/shipsDetection/sceneTrace?format=json</i> for getting default or configured scene auto-switch parameters for reference. c. Call NET_DVR_STDXMLConfig to transmit the request URI: PUT <i>/ISAPI/Thermal/channels/<ID>/shipsDetection/sceneTrace?format=json</i> and set

lpInBuffer to *JSON_ShipsDetectionSceneTraceList* for starting scene auto-switch.

Capture Ratio

- a. Call **NET_DVR_STDXMLConfig** to transmit the request URI: GET */ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/captureRatio/capabilities?format=json* for getting configuration capability of capture ratio.
- b. Call **NET_DVR_STDXMLConfig** to transmit the request URI: GET */ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/captureRatio?format=json* for getting default or configured capture ratio for reference.
- c. Call **NET_DVR_STDXMLConfig** to transmit the request URI: PUT */ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/captureRatio?format=json* and set **lpInBuffer** to *JSON_ShipsDetectionCaptureRatio* for setting capture ratio.

Advanced parameters

- a. Call **NET_DVR_STDXMLConfig** to transmit the request URI: GET */ISAPI/Thermal/channels/<ID>/shipsDetection/advanceParam/capabilities?format=json* for getting advanced configuration capability of ship detection to check the supported parameters.
- b. Call **NET_DVR_STDXMLConfig** to transmit the request URI: GET */ISAPI/Thermal/channels/<ID>/shipsDetection/advanceParam?format=json* for getting default or configured advanced parameters of ship detection for reference.
- c. Call **NET_DVR_STDXMLConfig** to transmit the request URI: PUT */ISAPI/Thermal/channels/<ID>/shipsDetection/advanceParam?format=json* and set **lpInBuffer** to *JSON_advanceParam* for setting advanced parameters of ship detection.

6. Set the function mode to "dredgerDetection" to detect the dredger.

- 1) **Optional:** Call **NET_DVR_STDXMLConfig** to transmit the request URI: GET */ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/shipsFunctionMode/capabilities?format=json* for getting function mode capability to check the supported modes.

The function mode capability is returned in the message *JSON_Cap_functionMode* by **lpOutBuffer**.

- 2) **Optional:** Call **NET_DVR_STDXMLConfig** to transmit the request URI: GET */ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/shipsFunctionMode?format=json* for getting default or configured function mode for reference.

The function mode is returned in the message *JSON_functionMode* by **lpOutBuffer**.

- 3) Call **NET_DVR_STDXMLConfig** to transmit the request URI: PUT */ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/shipsFunctionMode?format=json* and set **lpInBuffer** to *JSON_functionMode* for setting the function mode to "dredgerDetection".

7. Set dredger detection rules.

- 1) **Optional:** Call **NET_DVR_STDXMLConfig** to transmit the request URI: GET */ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/dredgerDetection/regions/capabilities?format=json*

for getting configuration capability of dredger detection rules to check the supported rule parameters.

The rule configuration capability of dredger detection is returned in the message **JSON_DredgerDetectionRuleListCap** by **IpOutBuffer**.

- 2) **Optional:** Call **NET_DVR_STDXMLConfig** to transmit the request URI: GET **/ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/dredgerDetection/regions/<ID>?format=json** or **/ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/dredgerDetection/regions?format=json** for getting default or configured rules of a or all dredger detection(s) for reference.

The rules of a or all dredger detection(s) are returned in the message **JSON_DredgerDetectionRule** or **JSON_DredgerDetectionRuleList** by **IpOutBuffer**.

- 3) Call **NET_DVR_STDXMLConfig** to transmit the request URI: PUT **/ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/dredgerDetection/regions/<ID>?format=json** or **/ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/dredgerDetection/regions?format=json** and set **IpInBuffer** to **JSON_DredgerDetectionRule** or **JSON_DredgerDetectionRuleList** for setting rules of a or all dredger detection(s).
8. Call **NET_DVR_STDXMLConfig** to transmit the request URI: PUT **/ISAPI/Event/schedules/dredgerDetection/<ID>/scene/<SID>** for setting the arming schedule of dredger detection alarm.



Note

Before setting arming schedule, you can transmit this URI by GET method to get default or configured arming schedule for reference.

9. Call **NET_DVR_STDXMLConfig** to transmit the request URI: PUT **/ISAPI/Event/triggers/dredgerDetection/<ID>/scene/<SID>** for setting linkage action of dredger detection alarm.



Note

- Before setting linkage action, you can transmit this URI by GET method to get default or configured linkage actions for reference.
 - To receive the alarm via platform, the linkage action must be set to "center".
-

10. Perform the following operation(s) after configuring the dredger detection alarm.

Get Ship Statistics by Scene	Call NET_DVR_STDXMLConfig to transmit the request URI: GET /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/shipsDetectionCount?format=json
Clear Ship Statistics by Scene	Call NET_DVR_STDXMLConfig to transmit the request URI: PUT /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/resetCount?format=json
Receive Alarm in Arming or Listening Mode	Perform the flow of Receive Alarm/Event in Arming Mode or Receive Alarm/Event in Listening Mode and set lCommand in the registered

alarm callback function (***MSGCallback***) to "COMM_VCA_ALARM"
(command No.: 0x4993).



Note

The dredger detection alarm information is returned in the message
JSON_EventNotificationAlert_dredgerDetection .

What to do next

Call ***NET_DVR_Logout*** and ***NET_DVR_Cleanup*** to log out and release the resources.

Chapter 5 Alarm/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.

5.1 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

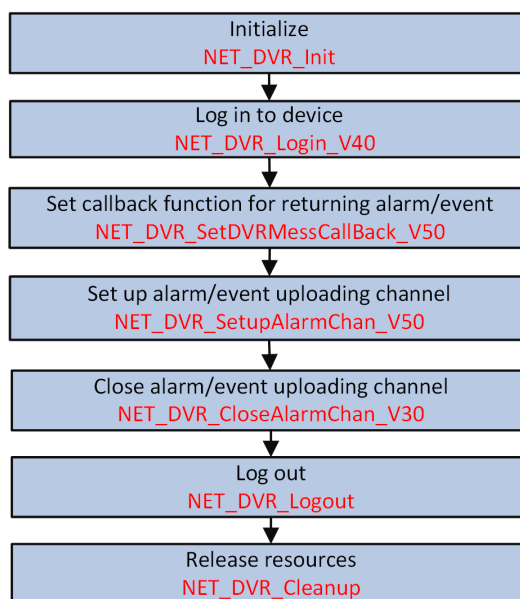


Figure 5-1 Programming Flow of Receiving Alarm/Event in Arming Mode

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 IUserID;
    //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};
    IUserID = NET_DVR_Login_V40(&struLoginInfo, &struDeviceInfoV40);
    if (IUserID < 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 IHandle = -1;
    if (0 == strlen(szSubscribe))
    {
        //Arm
        IHandle = NET_DVR_SetupAlarmChan_V50(IUserID, &struSetupParamV50, NULL, strlen(szSubscribe));
    }
    else
    {
        //Subscribe
    }
```

```
    IHandle = NET_DVR_SetupAlarmChan_V50(IUserID, &struSetupParamV50, szSubscribe, strlen(szSubscribe));
}

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

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

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

What to do next

Call **NET_DVR_Logout** and **NET_DVR_Cleanup** to log out and release resources.

5.2 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

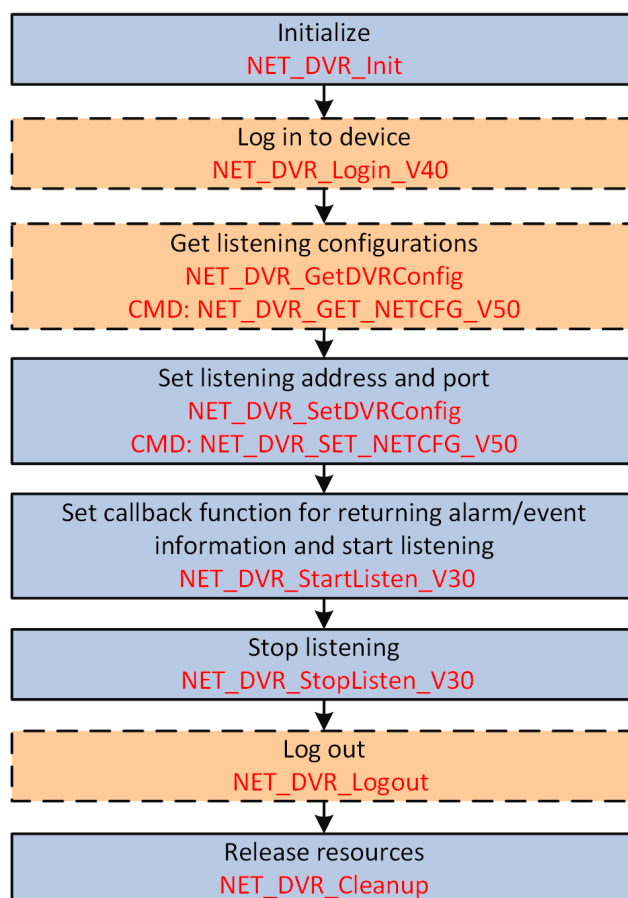


Figure 5-2 Programming Flow of Receiving Alarm/Event in Listening Mode

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.

5.3 Subscribe Alarm/Event in Arming Mode

For arming mode, the platform will connect to the devices automatically and send commands to the devices for uploading alarm/event information when the alarm is triggered or event occurred. To reduce the CPU and bandwidth usage of platform, and improve the device processing performance, the platform can subscribe alarm/event types to receive alarm/event information as required.

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

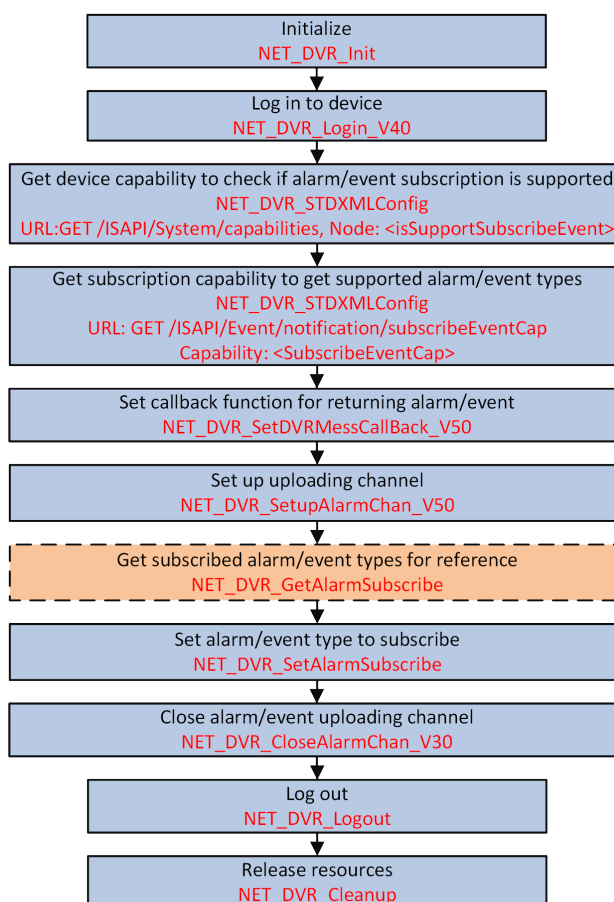


Figure 5-3 Programming Flow of Subscribing Alarm/Event in Arming Mode

1. Call **NET_DVR_STDXMLConfig** to pass through the request URL: GET **/ISAPI/System/capabilities** for getting device capability to check if alarm/event subscription is supported.

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

If the node **<isSupportSubscribeEvent>** is also returned in the message and its value is "true", it indicates that alarm/event subscription is supported by device, and you can continue to perform the following steps;

Otherwise, alarm/event subscription is not supported, please end the task.

2. Call **NET_DVR_STDXMLConfig** to pass through the request URL: GET **/ISAPI/Event/notification/subscribeEventCap** for getting subscription capability, which contains supported alarm/event types.

The alarm/event subscription capability is returned in the message **XML_SubscribeEventCap** by the output parameter (**lpOutputParam**) pointer.

3. Call **NET_DVR_SetDVRMessageCallBack_V50** to set callback function for returning alarm/event information or subscription failed 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.
- 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 *Supported Alarm/Event Types* for details).
- To receive the subscription result (subscription failed), the parameter **lCommand** (data type to be uploaded) in the configured callback function should be set to "COMM_ALARM_SUBSCRIBE_EVENT". And the result is returned in the message **XML_SubscribeEventResponse**

-
4. Call **NET_DVR_SetupAlarmChan_V50** to set up alarm/event uploading channel.
 5. **Optional:** Call **NET_DVR_GetAlarmSubscribe** to get subscribed alarm/event types for reference.
 6. Call **NET_DVR_SetAlarmSubscribe** to set alarm/event type to subscribe.
 7. Call **NET_DVR_CloseAlarmChan_V30** to close alarm/event uploading channel and finishing receiving.

What to do next


Call **NET_DVR_Logout** and **NET_DVR_Cleanup** to log out and release resources.

Chapter 6 Other Thermal Applications

Capture JPEG Picture

Function	API
Get JPEG picture with pixel-to-pixel temperature measurement data	<ol style="list-style-type: none"> 1. Call NET_DVR_GetSTDAbility with "NET_DVR_GET_THERMAL_PIP_CAPABILITIES" (command No.: 6767) and set IpCondBuffer in NET_DVR_STD_ABILITY to 4-byte channel No. The picture-in-picture configuration capability is returned by IpOutBuffer in the message XML_Cap_ThermalPip. 2. Call NET_DVR_CaptureJPEGPicture_WithAppendData to capture the JPEG picture with pixel-to-pixel temperature measurement data. The picture information is returned in the structure NET_DVR_JPEGPICTURE_WITH_APPENDDATA.

Get Temperature Measurement Information

Function	API
Manually get temperature information and temperature measurement rule	<p>Call NET_DVR_GetDVRConfig with "NET_DVR_GET_THERMOMETRYRULE_TEMPERATURE_INFO" (command No.: 23001).</p> <p>The temperature information and temperature measurement rules are returned by IpOutBuffer in NET_DVR_THERMOMETRYRULE_TEMPERATURE_INFO.</p> <p> Note You can check whether the product supports manual temperature measurement via the node <ManualThermCap> in XML_ThermalCap.</p>
Get real-time temperature measurement information	<ol style="list-style-type: none"> 1. Call NET_DVR_GetSTDAbility and set dwAbilityType to "NET_DVR_GET_THERMAL_CAPABILITIES" to get the thermal capability to check whether the device supports getting real-time temperature measurement information. The capability is returned in the message XML_ThermalCap, and if the device supports this function, the node

Function	API
	<p><isSupportRealtimeThermometry> is returned and its value is "true".</p> <p>2. Call NET_DVR_StartRemoteConfig with "NET_DVR_GET_REALTIME_THERMOMETRY" (command No.: 3629) and set IpInBuffer to NET_DVR_REALTIME_THERMOMETRY_COND. The real-time temperature measurement information is returned in the structure NET_DVR_THERMOMETRY_UPLOAD.</p>

Pixel-to-Pixel Temperature Measurement Configuration

Function	API
Get configuration capability of pixel-to-pixel temperature measurement	<p>Call NET_DVR_STDXMLConfig to transmit GET /ISAPI/Thermal/channels/<ID>/thermometry/pixelToPixelParam/capabilities. The capability is returned by IpOutBuffer in XML_PixelToPixelParamCap.</p>
Get pixel-to-pixel temperature measurement parameters	<p>Call NET_DVR_STDXMLConfig to transmit GET /ISAPI/Thermal/channels/<ID>/thermometry/pixelToPixelParam. The pixel-to-pixel temperature measurement parameters are returned by IpOutBuffer in XML_PixelToPixelParam.</p>
Set pixel-to-pixel temperature measurement parameters	<p>Call NET_DVR_STDXMLConfig to transmit PUT /ISAPI/Thermal/channels/<ID>/thermometry/pixelToPixelParam and set IpInBuffer to XML_PixelToPixelParam.</p>

Configure Picture-in-Picture


Function	API
Get picture-in-picture configuration capability	<p>Call NET_DVR_GetSTDAbility with "NET_DVR_GET_THERMAL_PIP_CAPABILITIES" (command No.: 6767) and set IpCondBuffer in NET_DVR_STD_ABILITY to 4-byte channel No. The picture-in-picture configuration capability is returned by the IpOutBuffer in the message XML_Cap_ThermalPip.</p>
Get picture-in-picture parameters	<p>Call NET_DVR_GetSTDConfig with "NET_DVR_GET_THERMAL_PIP" (command No.: 6768) and set IpCondBuffer in NET_DVR_STD_CONFIG to 4-byte channel No.</p>

Function	API
	The picture-in-picture configuration parameters is returned by the IpOutBuffer in NET_DVR_THERMAL_PIP .
Set picture-in-picture parameters	Call NET_DVR_SetSTDConfig with "NET_DVR_SET_THERMAL_PIP" (command No.: 6769), and set IpCondBuffer and IpInBuffer in NET_DVR_STD_CONFIG to 4-byte channel No and NET_DVR_THERMAL_PIP , respectively.

Get History Temperature Data

Function	API
Get history temperature capability	Call NET_DVR_STDXMLConfig to transmit GET /ISAPI/Thermal/channels/<ID>/historyTemperature/capabilities and set IpInBuffer to "NULL". The capability is returned by IpOutBuffer in XML_HistoryTemperatureCap .
Get history temperature	Call NET_DVR_STDXMLConfig to transmit POST /ISAPI/Thermal/channels/<ID>/historyTemperature and set IpInBuffer to XML_HistoryTemperatureDescription . The history temperature is returned by IpOutBuffer in XML_HistoryTemperatureResult .

Burning Prevention

Function	API
Get burning prevention capabilities	Call NET_DVR_STDXMLConfig to transmit GET /ISAPI/Thermal/channels/<ID>/burningPrevention/capabilities and set IpInBuffer to "NULL". The capability is returned by IpOutBuffer in XML_BurningPreventionCap . <div>  Note Check whether the device supports the burning prevention function via node <isSupportBurningPrevention> in XML_ThermalCap . </div>
Get burning prevention parameters	Call NET_DVR_STDXMLConfig to transmit GET /ISAPI/Thermal/channels/<ID>/burningPrevention and set IpInBuffer to "NULL".

Function	API
	The burning prevention parameters are returned by IpOutBuffer in <i>XML_BurningPrevention</i> .
Set burning prevention parameters	Call NET_DVR_STDXMLConfig to transmit PUT <i>/ISAPI/Thermal/channels/<ID>/burningPrevention</i> and set IpInBuffer to <i>XML_BurningPrevention</i> .

Advanced Parameters Configuration of Fire Detection

Function	API
Get capability of configuring advanced parameters of fire detection	Call NET_DVR_STDXMLConfig to transmit GET <i>/ISAPI/Thermal/channels/<ID>/fireDetection/advanceParam/capabilities?format=json</i> and set IpInBuffer to "NULL". The capability is returned by IpOutBuffer in <i>JSON_fireDetection_AdvanceParamCap</i> .
Get advanced parameters of fire detection	Call NET_DVR_STDXMLConfig to transmit GET <i>/ISAPI/Thermal/channels/<ID>/fireDetection/advanceParam?format=json</i> and set IpInBuffer to "NULL". The advanced parameters of fire detection are returned by IpOutBuffer in <i>JSON_fireDetection_AdvanceParam</i> .
Set advanced parameters of fire detection	Call NET_DVR_STDXMLConfig to transmit PUT <i>/ISAPI/Thermal/channels/<ID>/fireDetection/advanceParam?format=json</i> and set IpInBuffer to <i>JSON_fireDetection_AdvanceParam</i> .

Temperature Measurement Pre-Alarm

Function	API
Get arming schedule of temperature measurement pre-alarm for a specified channel	Call NET_DVR_STDXMLConfig to transmit GET <i>/ISAPI/Event/schedules/TMPA/<ID></i>
Set arming schedule of temperature measurement pre-alarm for a specified channel	Call NET_DVR_STDXMLConfig to transmit PUT <i>/ISAPI/Event/schedules/TMPA/<ID></i>
Get arming schedule of temperature measurement	Call NET_DVR_STDXMLConfig to transmit GET <i>/ISAPI/Event/schedules/TMPA</i>

Function	API
pre-alarm for all channels in a batch	
Set arming schedule of temperature measurement pre-alarm for all channels in a batch	Call NET_DVR_STDXMLConfig to transmit PUT /ISAPI/Event/schedules/TMPA

Temperature Calibration

Function	API
Get temperature calibration capability	Call NET_DVR_STDXMLConfig to transmit request URI: GET /ISAPI/Thermal/channels/<ID>/temperatureCorrect/capabilities?format=json
Get temperature calibration configuration parameters	Call NET_DVR_STDXMLConfig to transmit request URI: GET /ISAPI/Thermal/channels/<ID>/temperatureCorrect?format=json
Set temperature calibration parameters	Call NET_DVR_STDXMLConfig to transmit request URI: POST /ISAPI/Thermal/channels/<ID>/temperatureCorrect?format=json

Interval Temperature Measurement Alarm

Function	API
Get configuration capability of interval temperature measurement alarm	Call NET_DVR_STDXMLConfig to transmit request URI: GET /ISAPI/Thermal/channels/<ID>/thermometry/<SID>/alarmRules/temperatureIntervalMeasurement/capabilities?format=json
Get configuration parameters of interval temperature measurement alarm	Call NET_DVR_STDXMLConfig to transmit request URI: GET /ISAPI/Thermal/channels/<ID>/thermometry/<SID>/alarmRules/temperatureIntervalMeasurement?format=json
Set configuration parameters of interval temperature measurement alarm	Call NET_DVR_STDXMLConfig to transmit request URI: PUT /ISAPI/Thermal/channels/<ID>/thermometry/<SID>/alarmRules/temperatureIntervalMeasurement?format=json .

Chapter 7 API Reference

7.1 NET_DVR_CaptureJPEGPicture_WithAppendData

Call this API to capture a JPEG format picture with additional pixel-to-pixel thermometry data

API Definition

```
BOOL NET_DVR_CaptureJPEGPicture_NEW(  
    LONG                IUserID,  
    LONG                IChannel,  
    NET_DVR_JPEGPICTURE_WITH_APPENDDATA *IpJpegWithAppend  
);
```

Parameters

IUserID

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

IChannel

[IN] Channel number.

IpJpegWithAppend

[IN] JPEG picture parameters, see structure for details.

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.

7.2 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.

7.3 NET_DVR_CloseAlarmChan_V30

Close alarm uploading channel.

API Definition

```
BOOL NET_DVR_CloseAlarmChan_V30(  
    LONG   IAlarmHandle  
);
```

Parameters

IAlarmHandle

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**.

7.4 NET_DVR_GetAlarmSubscribe

Get the event/alarm subscription parameters.

API Definition

```
BOOL NET_DVR_GetAlarmSubscribe(  
    LONG   IAlarmHandle,  
    char    *pData,  
    DWORD   dwDataLen  
);
```

Parameters

IAlarmHandle

[IN] Value returned by **NET_DVR_SetupAlarmChan_V50**

pData

[OUT] Pointer to data buffer, see details in ***XML_SubscribeEvent***

dwDataLen

[IN] Size of data buffer, unit: byte, it cannot be 0.

Return Values

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

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

7.5 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.

7.6 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.

7.7 NET_DVR_GetDeviceAbility

Get the device capabilities.

API Definition

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

Parameters

IUserID

[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.

7.8 NET_DVR_GetDVRConfig

Get the device configuration information.

API Definition

```
BOOL NET_DVR_GetDVRConfig(  
    LONG    IUserID,  
    DWORD   dwCommand,  
    LONG    IRuleID,  
    LONG    IChannel,  
    LPVOID   lpOutBuffer,  
    DWORD   dwOutBufferSize,  
    LPDWORD  lpBytesReturned  
);
```

Parameters

IUserID

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

dwCommand

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

IRuleID

[IN] Rule ID.

IChannel

[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

7.9 NET_DVR_GetNextRemoteConfig

Get the next search result.

API Definition

```
LONG NET_DVR_GetNextRemoteConfig(  
    LONG    IHandle,  
    void    *IpOutBuff,  
    DWORD   dwOutBuffSize  
);
```

Parameters

IHandle

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

IpOutBuff

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

dwOutBuffSize

[IN] Buffer size.

Return Values

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

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

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

Remarks

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

7.10 NET_DVR_GetSTDAbility

Get the device capabilities.

API Definition

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

Parameters

IUserID

[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.

7.11 NET_DVR_GetSTDConfig

Get the device configuration information.

API Definition

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

Parameters

IUserID

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

dwCommand

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

IpConfigParam

[IN][OUT] Set input and output parameters, which are different according to different configuration functions. For different configuration functions, the **IpCondBuffer** and **IpOutBuffer** in the **IpConfigParam** are also different. See the structure **NET_DVR_STD_CONFIG** for details.



Note

When getting configuration parameters, the **IpInBuffer** in the **IpConfigParam** 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

7.12 NET_DVR_Init

Initialize the programming environment before calling other APIs.

API Definition

```
BOOL NET_DVR_Init(  
);
```

Return Values

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

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

The available error codes of this API are 0, 41, and 53. See details in **Device Network SDK Errors** .

Remarks

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

See Also

NET_DVR_Cleanup

7.13 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

7.14 NET_DVR_Logout

Log out from devices.

API Definitions

```
BOOL NET_DVR_Logout(  
    LONG  IUserID  
);
```

Parameters

IUserID

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

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, 3, 7, 8, 9, 10, 14, 17, 41, 44, 47, 72, and 73. See details in **Device Network SDK Errors** .

7.15 NET_DVR_PTZPreset_Other

Call this API to set and call preset.

API Definition

```
BOOL NET_DVR_PTZPreset_Other(  
    LONG  IUserID,  
    LONG  IChannel,  
    DWORD dwPTZPresetCmd,  
    DWORD dwPresetIndex  
);
```

Parameters

IUserID

[IN] Value returned by **NET_DVR_Login_V40**

IChannel

[IN] Channel No.

dwPTZPresetCmd

[IN] Preset commands, see details in the following table:

Command	Command No.	Description
SET_PRESET	8	Set preset.
CLE_PRESET	9	Clear preset.
GOTO_PRESET	39	Call preset.

dwPresetIndex

[IN] Preset No. (starts from 1), up to 300 presets can be supported.

Return Values

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

If returning failed, you can call ***NET_DVR_GetLastError*** to get the error code.

Remarks

- The preset commands correspond to the control codes between device and PTZ, the device will send the control codes to PTZ according to the configured decoder type and address. If the decoder of device and PTZ mismatches, you should configure the device decoder again. If the PTZ decoder is not supported by device, it is not available to call this API to set and call preset.
- If you call this API to control the PTZ, the device directly returns *TRUE* after receiving the control command. But for ***NET_DVR_PTZPreset***, the device returns *TRUE* only when operation completed.

7.16 NET_DVR_SetAlarmSubscribe

Set the event/alarm subscription parameters.

API Definition

```
BOOL NET_DVR_SetAlarmSubscribe(  
    LONG    IAlarmHandle,  
    char    *pData,  
    DWORD   dwDataLen  
);
```

Parameters

IAlarmHandle

[IN] Value returned by ***NET_DVR_SetupAlarmChan_V50***

pData

[IN] Pointer to data buffer, see details in ***XML_SubscribeEvent***

dwDataLen

[IN] Size of data buffer, unit: byte

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.

7.17 NET_DVR_SetDVRConfig

Set the device parameters.

API Definition

```
BOOL NET_DVR_SetDVRConfig(  
    LONG    IUserID,  
    DWORD   dwCommand,  
    LONG    IChannel,  
    LPVOID   lpInBuffer,  
    DWORD   dwInBufferSize  
);
```

Parameters

IUserID

[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

7.18 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 7-1 NET_SDK_INIT_CFG_TYPE

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

lpInBuff

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

Return Values

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

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

Remarks

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

7.19 NET_DVR_SetSTDConfig

Set the device parameters.

API Definition

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

Parameters

IUserID

[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

7.20 NET_DVR_STDXMLConfig

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

API Definition

```
BOOL NET_DVR_STDXMLConfig(  
    LONG                IUserID,  
    const NET_DVR_XML_CONFIG_INPUT    *IpInputParam,  
    NET_DVR_XML_CONFIG_OUTPUT    *IpOutputParam  
);
```

Parameters

IUserID

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

IpInputParam

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

IpOutputParam

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

Return Values

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

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

Remarks

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

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

Parameter of NET_DVR_STDXMLConfig		Component of Text Protocol
IpOutputParam	IpOutBuffer (see in structure <i>NET_DVR_XML_CONFIG_OUTPUT</i>)	Response Message
	IpStatusBuffer (see in structure <i>NET_DVR_XML_CONFIG_OUTPUT</i>)	Response Message

7.21 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 ((struTSPlateResult.struPicInfo[i].dwDataLen != 0)&&(struTSPlateResult.struPicInfo[i].byType == 1) |
(struTSPlateResult.struPicInfo[i].byType == 2))
{
    sprintf(filename,"testITSpic%d_%d.jpg",iNum,i);
    fSnapPic=fopen(filename,"wb");
    fwrite(struTSPlateResult.struPicInfo[i].pBuffer, struTSPlateResult.struPicInfo[i].dwDataLen,1,fSnapPic);
    iNum++;
    fclose(fSnapPic);
}
//License plate thumbnails
if ((struTSPlateResult.struPicInfo[i].dwDataLen != 0)&&(struTSPlateResult.struPicInfo[i].byType == 0))
{
    sprintf(filename,"testPicPlate%d_%d.jpg",iNum,i);
    fSnapPicPlate=fopen(filename,"wb");
    fwrite(struTSPlateResult.struPicInfo[i].pBuffer, struTSPlateResult.struPicInfo[i].dwDataLen, 1, \
fSnapPicPlate);
    iNum++;
    fclose(fSnapPicPlate);
}
//Processing other data...
}
break;
}
default:
    break;
}
}

void CALLBACK MessageCallbackNo2(LONG ICommand, 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 funcion via message mode (PostMessage).

    switch(ICommand)

```

```
{
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_ITS_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(IUserID);
    NET_DVR_Cleanup();
    return;
}

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

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

See Also

NET_DVR_SetupAlarmChan_V50

7.21.1 MSGCallback

Alarm/event information callback function.

Callback Function Definition

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

Parameters

ICommand

[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 **ICommand**.

dwBufLen

[OUT] Size of alarm/event information buffer.

pUser

[OUT] User data.

7.22 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          IUserID,  
    NET_DVR_SETUPALARM_PARAM_V50 IpSetupParam,  
    char          *pData,  
    DWORD         dwDataLen,  
);
```

Parameters**IUserID**

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

IpSetupParam

[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**.

7.23 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**).

7.24 NET_DVR_StartRemoteConfig

Enable remote configuration.

API Definition

```
LONG NET_DVR_StartRemoteConfig(  
    LONG        IUserID,  
    DWORD       dwCommand,  
    LPVOID       lpInBuffer,  
    DWORD       dwInBufferLen,  
    fRemoteConfigCallback  cbStateCallback,  
    LPVOID       pUserData  
);
```

Parameters

IUserID

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

dwCommand

[IN] Configuration commands. For different functions, the commands and **lpInBuffer** are different.

lpInBuffer

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

dwInBufferLen

[IN] Size of input buffer.

cbStateCallback

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

pUserData

[OUT] User data.

Return Values

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

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

Remarks

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

7.24.1 fRemoteConfigCallback

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

Callback Function Definition

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

Parameters

dwType

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

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

NET_SDK_CALLBACK_TYPE_STATUS

Connection status.

NET_SDK_CALLBACK_TYPE_PROGRESS

Connection progress.

NET_SDK_CALLBACK_TYPE_DATA

Related data to be called back.

lpBuffer

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

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

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

dwBufLen

[OUT] Buffer size.

pUserData

[OUT] User data.

7.25 NET_DVR_StopListen_V30

Stop listening (supports multiple threads).

API Definition

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

Parameters

lListenHandle

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*** .

7.26 NET_DVR_StopRemoteConfig

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

API Definition

```
BOOL NET_DVR_StopRemoteConfig(  
    LONG  IHandle  
);
```

Parameters

IHandle

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

Return Values

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

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

Appendix A. Data Structure

A.1 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	<i>NET_ALARM_RECORD_FILE_LOSS</i>	Video loss alarm information, the value of dwAlarmType in <i>NET_DVR_ALARMINFO_DEV_V40</i> is 8.
struStreamException	<i>NET_ALARM_STREAM_EXCEPTION</i>	Streaming exception alarm information, the value of dwAlarmType in <i>NET_DVR_ALARMINFO_DEV_V40</i> is 9.
struResourceUsage	<i>NET_ALARM_RESOURCE_USAGE</i>	Resource usage alarm information, the value of dwAlarmType in <i>NET_DVR_ALARMINFO_DEV_V40</i> is 10.
struRecordException	<i>NET_ALARM_RECORD_EXCEPTION</i>	Recording exception alarm information, the value of dwAlarmType in <i>NET_DVR_ALARMINFO_DEV_V40</i> is 12.

A.2 NET_ALARM_RECORD_EXCEPTION

Structure about Recording Exception Alarm Information

Member	Data Type	Description
byReason	BYTE	Exception reason: 0-video volume full, 1-video volume exception, 2-no available video volume.
byRes1	BYTE[]	Reserved, set to 0. The maximum array length is 3 bytes.
sVolumeName	BYTE[]	Video volume name, the maximum array length is "MAX_VOLUMENAME_LEN" (32 bytes).

Member	Data Type	Description
dwVolumeID	DWORD	Video volume ID, or HDD No.
byRes	BYTE[]	Reserved, set to 0. The maximum array length is 452 bytes.

A.3 NET_ALARM_RECORDFILE_LOSS

Structure about Video Loss Alarm Information

Member	Data Type	Description
struInspectStart	<i>NET_DVR_TIME_EX</i>	Start time of video loss check.
struInspectEnd	<i>NET_DVR_TIME_EX</i>	End time of video loss check.
struIP	<i>NET_DVR_IPADDR_UNION</i>	IP address of video loss channel.
dwChanNo	DWORD	Channel No.
dwIDIndex	DWORD	Encoder ID.
sName	BYTE[]	Encoder name, the maximum array length is "STREAM_ID_LEN" (32 bytes).
struLossStartTime	<i>NET_DVR_TIME_EX</i>	Start time of video loss.
struLossEndTime	<i>NET_DVR_TIME_EX</i>	End time of video loss.
dwLostNum	DWORD	Number of lost video files, 0xffffffff-all video files are lost.
byRes	BYTE[]	Reserved, set to 0. The maximum array length is 240 bytes.

A.4 NET_ALARM_RESOURCE_USAGE

Structure about Resource Usage Alarm Information

Member	Data Type	Description
byLevel	BYTE	Usage alarm level: 0-normal, 1-alarm level 1, 2-alarm level 2, 3-alarm level 3.
byRes	BYTE[]	Reserved, set to 0. The maximum array length is 491 bytes.

A.5 NET_ALARM_STREAM_EXCEPTION

Structure about Video Exception Alarm Information

Member	Data Type	Description
strulP	<i>NET_DVR_IPADDR_UNION</i>	IP address of video exception channel.
dwChanNo	DWORD	Channel No.
dwIDIndex	DWORD	Encoder ID.
sName	BYTE[]	Encoder name, the maximum array length is "STREAM_ID_LEN" (32 bytes).
byExceptionCase	BYTE	Exception reason: 0-data writing exception, 1-network exception.
byRes	BYTE[]	Reserved, set to 0. The maximum array length is 307 bytes.

A.6 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-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	<i>NET_DVR_TIME_EX</i>	Alarm time

Member	Data Type	Description
uStruAlarm	Union (Table 8-1)	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 byISO8601 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 byISO8601 is "1".
byRes	Array of BYTE	Reserved, set to 0. The maximum size is 5 bytes.

Table A-1 Union about Alarm Information Structures (uStruAlarm)

Member	Data Type	Description
byUnionLen	Array of BYTE	Union size, which is 116 bytes.
struIOAlarm	Struct (Table 8-2)	Structure about alarm input parameters
struAlarmChannel	Struct (Table 8-3)	Structure about alarm channel parameters
struAlarmHardDisk	Struct (Table 8-4)	Structure about HDD alarm parameters
struRecordingHost	Struct (Table 8-5)	Structure about alarm parameters of education sharing system
struVoltageInstable	Struct (Table 8-6)	Structure about alarm parameters of supply voltage exception
struPTLocking	Struct (Table 8-7)	Structure about parameters of PTZ locked alarm

Table A-2 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 pAlarmData in

Member	Data Type	Description
		NET_DVR_ALARMINFO_V40 , 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 pAlarmData of NET_DVR_ALARMINFO_V40 , each channel is represented by 4 bytes.

Table A-3 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 pAlarmData of NET_DVR_ALARMINFO_V40 , 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.
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-4 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 pAlarmData of NET_DVR_ALARMINFO_V40 , each alarm HDD is represented by 4 bytes.

Table A-5 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	NET_DVR_TIME_EX	Recording end time.

Table A-6 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-7 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.
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.7 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 pPicPackData returned). When this member is 1, only one structure of NET_DVR_ALARM_ISAPI_PICDATA will be returned by pPicPackData . When this member is larger than 1, multiple structures of NET_DVR_ALARM_ISAPI_PICDATA will be returned by pPicPackData .
byRes	Array of BYTE	Reserved, set to 0. The maximum size is 2 bytes.
pPicPackData	void*	Alarm picture structure, see NET_DVR_ALARM_ISAPI_PICDATA for details.
byRes	Array of BYTE	Reserved. The maximum size is 32 bytes.

Remarks

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

A.8 NET_DVR_ALARM_ISAPI_PICDATA

Structure about Alarm Picture Data Transmitted Based on Text Protocol

Member	Data Type	Description
dwPicLen	DWORD	Alarm picture data length.
byRes	Array of BYTE	Reserved, set to 0. The maximum size is 4 bytes.
szFilename	Array of char	Picture file saving path, including file name. The maximum size is 256 bytes.
pPicData	BYTE*	Pointer that pointing to the uploaded image data.

A.9 NET_DVR_ALARMINFO_DEV**Device Alarm Information Structure**

Memeber	Data Type	Description
dwAlarmType	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.
struTime		Alarm time
byRes	Array of BYTE	Reserved, set to 0.
dwNumber	DWORD	Number of alarm triggered channels.
pNO	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.10 NET_DVR_ALARMINFO_DEV_V40

Structure about CVR Alarm Information

Member	Data Type	Description
dwAlarmType	DWORD	Alarm categories: 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 monitoring alarm, 10-usage alarm, 11-CVR exception recovered, 12-recording exception.
struTime	<i>NET_DVR_TIME</i>	Alarm time
uSubAlarmInfo	<i>NET_ALARM_CVR_SUBINFO_UNION</i>	CVR alarm information structure, and it is valid when the alarm type is 8, 9, 10, and 12.
byRes	Array of BYTE	Reserved, set to 0. The maximum size is 256 bytes.
dwNumber	DWORD	Number of alarm triggered channels.
pNO	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.11 NET_DVR_ALARMINFO_V30**Structure About Uploaded Alarm Information**

Member	Data Type	Description
dwAlarmType	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

Member	Data Type	Description
		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.
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.12 NET_DVR_ALARMINFO_V40

Structure About Uploaded Alarm Information

Member	Data Type	Description
struAlarmFixedHeader	NET_DVR_ALRAM_FIXED_HEADER	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-8 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.13 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

Member	Data Type	Description
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 NET_DVR_Login_V40 , 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.
byIpProtocol	BYTE	Network protocol: 0-IPv4, 1-IPv6
byRes2	Array of BYTE	Reserved, set to 0.

A.14 NET_DVR_ALARMSTRATEGY_PARAM

Alarm Strategy Structure

Member	Data Type	Description
byStrategyType	BYTE	Strategy types: 0-any alarm, 1-linked alarm, 2-multi-system alarm, 3-specified fire source detection alarm, 4-specified smoke detection alarm.
byRes	Array of BYTE	Reserved.

See Also

NET_DVR_FIREDETECTION_CFG

A.15 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.16 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-Hikvision Private Protocol (default), 1-RTSP, 2-Hikvision Private Protocol+RTSP
bySubProto	BYTE	Transmission protocol type of sub-stream: 0-Hikvision Private Protocol (default), 1-RTSP, 2-Hikvision Private Protocol+RTSP
bySupport	BYTE	Capabilities, if the result of bitwise operation is 0, it refers that the capability is not supported,

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

Member	Data Type	Description
		<ul style="list-style-type: none"> bySupport2&0x1: whether supports getting stream via URL. bySupport2&0x2: whether supports FTP with version 40. bySupport2&0x4: whether supports ANR. bySupport2&0x20: whether supports getting device status. bySupport2&0x40: whether supports encrypting stream.
wDevType	WORD	Device model
bySupport3	BYTE	<p>Extended capabilities, if the result of bitwise operation is 0, it refers that the capability is not supported, while, if the result is 1, it indicates that the capability is supported.</p> <ul style="list-style-type: none"> bySupport3&0x1: whether supports multi-stream. bySupport3&0x4: whether supports configuring by group (e.g., image, alarm input, alarm output, user, device status, JPEG picture capture, continuous and scheduled capture, .HDD group management, and so on). bySupport3&0x20: whether supports getting stream via DDNS.
byMultiStreamProto	BYTE	<p>Whether supports multi-stream, if the result of bitwise operation is 0, it refers to not support, if the result is 1, it refers to support.</p> <ul style="list-style-type: none"> byMultiStreamProto&0x1: whether supports third-stream. byMultiStreamProto&0x2: whether supports fourth-stream. byMultiStreamProto&0x40: whether supports main stream. byMultiStreamProto&0x80: whether supports sub-stream.
byStartDChan	BYTE	Start No. of digital channel, 0-no digital channel (e.g., DVR, network camera).

Member	Data Type	Description
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 supported, if the result is 1, it indicates that the capability is supported. <ul style="list-style-type: none"> bySupport4&0x01: whether all stream types support RTSP and Hikvision Private Protocol. bySupport4&0x02: whether the device supports transmitting form format data via API (NET_DVR_STDXMLConfig). bySupport4&0x10: whether supports loading network disk by domain name.
byLanguageType	BYTE	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"> byLanguageType ==0: this field is not supported by device. byLanguageType&0x1: whether supports Chinese. byLanguageType&0x2: whether supports English.
byVoiceInChanNum	BYTE	Number of audio input channels
byStartVoiceInChanNo	BYTE	Start No. of audio input channel, 0-invalid.
byRes3	Array of BYTE	Reserved, set to 0.
byMirrorChanNum	BYTE	Number of mirror channels
wStartMirrorChanNo	WORD	Start No. of mirror channel
byRes2	Array of BYTE	Reserved, set to 0.

Remarks

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

See Also**NET_DVR_DEVICEINFO_V40****A.17 NET_DVR_DEVICEINFO_V40****Device Parameter Structure (V40)**

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

Member	Data Type	Description
		operation: 0-not support, 1-support. The result of bySupport&0x1 indicates that this member is reserved; the result of bySupport&0x2 indicates that whether it supports uploading changes: 0-not support, 1-support. This member is the capability set extension.
byLoginMode	BYTE	Login mode: 0-login via private protocol, 1-login via text protocol. For private protocol, the default login port number is 8000, and for text protocol, the default login port number is 80 or 443.
dwOEMCode	DWORD	OEM code.
iResidualValidity	int	Remaining valid days of the user's password, unit: day. If the negative number is returned, it indicates that the password being used has expired. For example, if -3 is returned, it indicates that the password being used has expired for three days.
byResidualValidity	BYTE	Whether the member iResidualValidity is valid: 0-invalid, 1-valid.
bySingleStartDTalkChannel	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.
bySingleDTalkChannels	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

Member	Data Type	Description
		new password" each time he/she logs in to the device until he/she changes the password).
bySupportStreamEncrypt	BYTE	Whether it supports stream encryption, it depends on the result of bitwise AND (&) operation: 0-no, 1-yes. The result of bySupportStreamEncrypt&0x1 indicates whether to support RTP/TLS streaming, the result of bySupportStreamEncrypt&0x2 indicates whether to support SRTP/UDP streaming, and the result of bySupportStreamEncrypt&0x4 indicates whether to support SRTP/MULTICAST streaming.
byRes2	Array of BYTE	Reserved, set to 0.

Remarks

- Four character types are allowed in the password, including digits, lowercase letters, uppercase letters and symbols. The maximum password length is 16 bits, and there are four password strength levels, see details below:
 - Level 0 (Risky Password): The password length is less than 8 bits, or only contains one kind of the character types. Or the password is the same with the user name, or is the mirror writing of the user name.
 - Level 1 (Weak Password): The password length is more than or equal to 8 bits, and contains two kinds of the character types. Meanwhile, the combination should be (digits + lowercase letters) or (digits + uppercase letters).
 - Level 2 (Medium Password): The password length is more than or equal to 8 bits, and contains two kinds of the character types. Meanwhile, the combination cannot be (digits + lowercase letters) and (digits + uppercase letters).
 - Level 3 (Strong Password): The password length is more than or equal to 8 bits, and at least contains three kinds of the character types.
- For login via text protocol, the following parameters are not supported: **bySupportLock**, **byRetryLoginTime**, **byPasswordLevel**, **byProxyType**, **dwSurplusLockTime**, **byCharEncodeType**, and **bySupportDev5**.

A.18 NET_DVR_DPC_PARAM

Structure of Defective Point Correction Parameters

Member	Data Type	Description
dwSize	DWORD	Structure size
dwChannel	DWORD	Channel number
wCtrlType	WORD	Control type, details as follows:
byDPCMode	BYTE	DPC Mode: 0-Manual, 1-Auto Manual Correction: default mode, correct manually by using wCtrlType operation type Auto Correction: correct automatically, in this mode, the parameters wCtrlType and struPoint are invalid.
byRes	BYTE	Reserved, set to 0.
struPoint	NET_VCA_POINT	The input plane coordinates of the image, normalize to 0-1. It is valid when wCtrlType value "DPC_POINT".
byRes1	Array of BYTE	Reserved, set to 0.

Remarks

Control type, details as follows:

wCtrlType Macro Definition	Value	Description
DPC_CORRECT	1	Defective pixel correction
DPC_CORRECT_CANCEL	2	Cancel correction
DPC_CROSS_DISPALY_OPEN	3	Enable the crossed display of defective pixel detection.
DPC_CROSS_DISPALY_CLOSE	4	Disable the crossed display of defective pixel detection.
DPC_POINT	5	Defective pixel correction coordinates
DPC_UP	6	The defective pixel coordinate is upward offset.
DPC_DOWN	7	The defective pixel coordinate is downward offset.
DPC_RIGHT	8	The defective pixel coordinate is offset towards right.

wCtrlType Macro Definition	Value	Description
DPC_LEFT	9	The defective pixel coordinate is offset towards left.
DPC_ALL_CORRECT	10	All the defective pixels correction.
DPC_SAVE	11	Save the defective pixels.

A.19 NET_DVR_EVENT_SCHEDULE

Arming Schedule Parameter Structure

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

A.20 NET_DVR_ETHERNET_V30

Ethernet Configuration Structure

Member	Data Type	Description
struDVRIP	NET_DVR_IPADDR_UNION	Device IP address
struDVRIPMask	NET_DVR_IPADDR_UNION	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.

Member	Data Type	Description
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.21 NET_DVR_EVENT_TRIGGER

Structure About Event Linkage Configuration

Member	Data Type	Description
dwSize	DWORD	Structure size.
struHandleException	Array of NET_DVR_HANDLEEXCEPTION_V41	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 NET_DVR_PRESETCHAN_INFO	Information of channel that called preset
struCruiseChanInfo	Array of NET_DVR_CRUISECHAN_INFO	Information of channel that called patrol
struPtzTrackInfo	Array of NET_DVR_PTZTRACKCHAN_INFO	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.22 NET_DVR_FACE_THERMOMETRY_ALARM

Body Thermometry Alarm Information Structure

Member	Data Type	Description
dwSize	DWORD	Structure size
dwChannel	DWORD	Channel No.
byRuleID	BYTE	Body thermometry rule ID, its value is between from 1 and 40.
byRes1	Array of BYTE	Reserved, set to 0
byRuleName	Array of BYTE	Rule name
dwRelativeTime	DWORD	Time of UTC \pm 00:00
dwAbsTime	DWORD	Local time
byFaceDetectionState	BYTE	Face detection status: 0-no face detected, 1-face detected. When it is 0, the dwFacelImageLen is 0, and struFaceRect is null.
byThermometryUnit	BYTE	Thermometry unit: 0-Celsius ($^{\circ}$ C), 1-Fahrenheit ($^{\circ}$ F), 2- Kelvin(K)
byAlarmRule	BYTE	Alarm rule: 0-max. temperature higher than, 1-max.temperature lower than
byRes2	Array of BYTE	Reserved, set to 0
fAlarmTemperature	float	Alarm triggered temperature, ranges from -20.0 to 120.0, corrects to one decimal place.
fRuleTemperature	float	Rule temperature, ranges from -20.0 to 60.0, corrects to one decimal place.
dwVisibleLightImageLen	DWORD	Visible light picture size
pVisibleLightImage	BYTE*	Visible light picture pointer
dwFacelImageLen	DWORD	Face thumbnail size
pFacelImage	BYTE*	Face thumbnail pointer
struFaceRegion	NET_VCA_RECT	Face thumbnail coordinates in captured picture
fMinTemperature	float	Minimum temperature, ranges from -20.0 to 120.0, corrects to one decimal place.
fAverageTemperature	float	Average temperature, ranges from -20.0 to 120.0, corrects to one decimal place.

Member	Data Type	Description
struMinTemperaturePoint	NET_VCA_POINT	Coordinates of lowest temperature position
struMaxTemperaturePoint	NET_VCA_POINT	Coordinates of highest temperature position
byRes	Array of BYTE	Reserved, set to 0

Remarks

When no face detected, the **byFaceDetectionState** is "0", **dwfacelImageLen** is "0", and **struFaceRect** is "null"; when the face is detected, the **byFaceDetectionState** is "1", **dwfacelImageLen** is the face thumbnail size, and **struFaceRect** is the actual position of face thumbnail in visible light.

A.23 NET_DVR_FIREDETECTION_ALARM

Structure About Fire and Smoke Detection Alarm Information

Member	Data Type	Description
dwSize	DWORD	Structure size.
dwRelativeTime	DWORD	Time of UTC \pm 00:00, which is valid only when the value of byTimeDiffFlag is "1".
dwAbsTime	DWORD	Local time.
struDevInfo	NET_VCA_DEV_INFO	Front-end device information
wPanPos	WORD	Panning parameter (horizontal, actual value \times 100).
wTiltPos	WORD	Tilting parameter (vertical, (actual value + 360) \times 100).
wZoomPos	WORD	Zooming parameter (actual value \times 100).
byPicTransType	BYTE	Picture transmission format: 0-binary data, 1-URL.
byRes1	BYTE	Reserved, set to 0.
dwPicDataLen	DWORD	Size of thermal alarm picture.
pBuffer	BYTE*	Buffer pointer for saving thermal picture data.
struRect	NET_VCA_RECT	Coordinates of fire source frame

Member	Data Type	Description
struPoint	NET_VCA_POINT	Coordinates of the maximum temperature point in the fire source frame
wFireMaxTemperature	WORD	Maximum temperature, range: [300, 400], unit: °C.
wTargetDistance	WORD	Target range: [100, 10000], unit: m.
byStrategyType	BYTE	Strategy type: 0-any alarm, 1-fire source+smoke detection alarm, 2-multiple systems alarm, 3-specified fire source detection alarm, 4-specified smoke detection alarm.
byAlarmSubType	BYTE	Alarm minor type: 0-fire source detection alarm, 1-smoke detection alarm, 2-fire source +smoke detection alarm.
byPTZPosExEnable	BYTE	Whether to enable PTZ coordinates extension: 0-no (the PTZ position is based on wPanPos , wTiltPos , wZoomPos), 1-yes (the PTZ position is based on struPtzPosEx).
byRes2	BYTE	Reserved, set to 0.
struPtzPosEx	NET_PTZ_INFO	Extended PTZ coordinates, which supports correcting to three decimal places.
dwVisiblePicLen	DWORD	Size of visible light picture.
pVisiblePicBuf	BYTE*	Buffer pointer for saving visible light picture data.
pSmokeBuf	BYTE*	Pointer of smoke detection alarm information, which points to the structure NET_DVR_SMOKEDETECTION_ALARM . It is valid when the byAlarmSubType is "1" and "2".
wDevInfolvmsChannelEx	WORD	Extended the parameter bylvmsChannel in NET_VCA_DEV_INFO , its value range is extended.
byRes3	BYTE	Reserved, set to 0.
byFireScanWaitMode	BYTE	Fire source scanning mode: 0-auto, 1-manual.
dwVisibleChannel	DWORD	No. of visible light channel.
byTimeDiffFlag	BYTE	Whether the time difference parameter is valid: 0-invalid, 1-valid.

Member	Data Type	Description
cTimeDifferenceH	signed 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 byTimeDiffFlag is "1".
cTimeDifferenceM	signed 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 byTimeDiffFlag is "1".
byRes	Array of BYTE	Reserved, set to 0.

A.24 NET_DVR_FIREDETECTION_CFG

Structure About the Configuration Parameters of Fire and Smoke Detection

Member	Data Type	Description
dwSize	DWORD	Structure size.
byEnabled	BYTE	Whether to enable dynamic fire source detection: 0-no, 1-yes.
bySensitivity	BYTE	Detection sensitivity, which ranges from 1 to 100.
byFireComfirmTime	BYTE	Fire source detection duration, which ranges from 0 to 120, unit: second, default: 5s.
byFireRegionOverlay	BYTE	Whether to display frame of fire source: 0-no, 1-yes.
byDetectionMode	BYTE	Fire detection mode: 0-secondary detection, 1-detect by single frame.
byFireFocusMode	BYTE	Focus mode: 0-auto, 1-patrol.
byFireZoomMode	BYTE	Zooming mode: 0-auto, 1-manual.
byFirezoomLevel	BYTE	Zooming rate, which ranges from 1 to 100. It is valid when byFireZoomMode is "1".
bySmokeFireEnabled	BYTE	Whether to enable smoke and fire detection: 0-no, 1-yes.
byFireManualWaitEnabled	BYTE	Fire detection waiting mode: 0-auto, 1-manual.

Member	Data Type	Description
byCancelRepeatedAlarmEnabled	BYTE	Whether to cancel repeat alarm: 0-no, 1-yes.
byApplicationSceneMode	BYTE	Scene mode: 0-reserved, 1-forest, 2-frame field, 3-city, 4-indoor or perimeter.
dwInstallationHeight	DWORD	Height of mounting position.
byFireSourceDetection	BYTE	Fire source detection mode: 0-dynamic fire source, 1-smoking
bySmokeAuxiliaryDetectionEnabled	BYTE	Whether to enable fire and smoke detection: 0-no, 1-yes. It is valid when the detection mode is multiple frame.
byverificationSensitivity	BYTE	Sensitivity of double verification, ranges from 1 to 100, the default value is 50.
byFireAlgorithmMode	BYTE	Fire detection algorithm mode: 0-invalid, 1-pattern recognition, 2-machine learning
byAgriculturalMachineryFilterEnabled	BYTE	Enable agricultural machinery filter or not: 0-no, 1-yes
byWaterReflectionEnabled	BYTE	Enable water reflection or not: 0-no, 1-yes
byPatrolSensitivity	BYTE	Patrol sensitivity, only valid for fire detection, ranges from 1 to 100
byRes	Array of BYTE	Reserved.
struAlarmStrategy	NET_DVR_ALARMSTRATEGY_PARAM	Alarm strategy
struSmokeCfg	NET_DVR_SMOKEDETECTION_CFG	Smoke detection configuration structure

A.25 NET_DVR_FOCUSMODE_CFG

Structure About Cocus Mode Configuration of Speed Dome

Member	Data Type	Description
dwSize	DWORD	Structure size.
byFocusMode	BYTE	Focus mode: 0-auto, 1-manual, 2-semiautomatic.

Member	Data Type	Description
byAutoFocusMode	BYTE	Auto focus mode: 0-close, 1-mode A, 2-mode B, 3-mode AB, 4-mode C.
wMinFocusDistance	WORD	Minimum focusing distance, unit: CM, 0- auto, 0xffff- infinite.
byZoomSpeedLevel	BYTE	Zoom speed, ranges from 1 to 3.
byFocusSpeedLevel	BYTE	Focus speed, ranges from 1 to 3.
byOpticalZoom	BYTE	Optical zoom, ranges from 0 to 255.
byDigitalZoom	BYTE	Digital zoom, ranges from 0 to 255.
fOpticalZoomLevel	float	Optical zoom, ranges from 1 to 32, minimum interval is 0.5
dwFocusPos	DWORD	Focus value, range: [0x1000,0xC000].
byFocusDefinitionDisplay	BYTE	Whether to display focus definition value: 0-no, 1-yes.
byFocusSensitivity	BYTE	Focus sensitivity, range: [0,2], it is valid when byFocusMode is 0 or 2.
byRes1	Array of BYTE	Reserved, set to 0.
dwRelativeFocusPos	BYTE	Relative focus value, low 16 bytes indicate focus value (ranges from 0 to 4000), and high 16 bytes indicate temperature value under current focus.
byRes	Array of BYTE	Reserved, set to 0.

A.26 NET_DVR_HANDLEEXCEPTION_V41

Exception Information Structure

Member	Data Type	Description
dwHandleType	DWORD	Handling types, see details below: <ul style="list-style-type: none">• 0x00: no response• 0x01: display alarm on monitor screen• 0x02: audio warning• 0x04: upload to center• 0x08: trigger alarm output

Member	Data Type	Description
		<ul style="list-style-type: none"> • 0x10: send picture with JPEG format by e-mail • 0x20: trigger wireless sound and light alarm • 0x40: trigger e-map (supported by PCNVR only) • 0x200: capture picture and upload to FTP • 0x400: focus mode linkage (for defocus detection) • 0x800: PTZ linkage (speed dome tracks the target) • 0x1000: capture picture and upload to cloud storage. • 0x10000: message alarm <p>E.g., if dwHandleType 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	Maximum 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. byRelAlarmOut[i]==3 indicates that the alarm output No.4 is triggered.
byRes	Array of BYTE	Reserved, set to 0.

A.27 NET_DVR_INIT_CFG_ABILITY

Initialization Capability Structure

Member	Data Type	Description
enumMaxLoginUsersNum	INIT_CFG_MAX_NUM	<p>Maximum number of users can log in, see details below:</p> <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,</pre>

Member	Data Type	Description
		INIT_CFG_NUM_20480 = 20480 }INIT_CFG_MAX_NUM
enumMaxAlarmNum	INIT_CFG_MAX_NUM	Maximum number of alarm channels, see details below: 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
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.28 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.29 NET_DVR_JPEGPICTURE_WITH_APPENDDATA

JPEG Picture Information Structure

Member	Data Type	Description
dwSize	DWORD	Structure size
dwChannel	DWORD	Channel No.
dwJpegPicLen	DWORD	JPEG picture length

Member	Data Type	Description
pJpegPicBuff	char*	Pointer to JPEG picture
dwJpegPicWidth	DWORD	Picture width
dwJpegPicHeight	DWORD	Picture height
dwP2PDataLen	DWORD	Length of pixel-to-pixel thermometry data.
pP2PDataBuff	char*	Pointer to pixel-to-pixel thermometry data.
byIsFreezedata	BYTE	Whether freezes the data: 0-no, 1-yes.  Note Freeze here indicates that during adjusting the shutter or changing temperature level, the device keeps a record of the raw data without updating. In the process, the image and temperature measurement information remains unchanged.
byRes	Array of BYTE	Reserved, and set to 0.

A.30 NET_DVR_LINEPOLYGON_THERM_CFG

Structure About Real-Time Information of Thermometry by Line or Frame

Member	Data Type	Description
fMaxTemperature	float	The highest temperature
fMinTemperature	float	The lowest temperature
fAverageTemperature	float	The average temperature
fTemperatureDiff	float	The temperature difference
struRegion	NET_VCA_POLYGON	Coordinates of temperature measuring area, it is valid when the rule calibration type is line or frame.
byRes	Array of BYTE	Reserved, set to 0.

See Also

NET_DVR_THERMOMETRY_UPLOAD

A.31 NET_DVR_LLI_PARAM

Longitude and Latitude Parameter Structure

Member	Data Type	Description
fSec	float	Second, range: [0.000000, 60.000000]
byDegree	BYTE	Degree, range of latitude: [0, 90], range of longitude: [0, 180]
byMinute	BYTE	Minute, range: [0, 59]
byRes	Array of BYTE	Reserved, set to 0. The maximum size is 6 bytes.

A.32 NET_DVR_LLPOS_PARAM

Position Information (Longitude and Latitude) Structure

Member	Data Type	Description
byLatitudeType	BYTE	Latitude type: 0-north, 1-south.
byLongitudeType	BYTE	Longitude type: 0-east, 1-west.
byRes1	Array of BYTE	Reserved, set to 0.
struLatitude	NET_DVR_LLI_PARAM	Latitude information.
struLongitude	NET_DVR_LLI_PARAM	Longitude information.
byRes	Array of BYTE	Reserved, set to 0. The maximum size is 16 bytes.

A.33 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.34 NET_DVR_MIME_UNIT

Input Content Details Structure of Message Transmission API (NET_DVR_STDXMLConfig)

Member	Data Type	Description
szContentType	Array of char	Content type (corresponds to Content-Type field in the message), e.g., text/json. text/xml, and so on. The content format must be supported by HTTP.
szName	Array of char	Content name (corresponds to name field in the message), e.g., name="upload".
szFilename	Array of char	Content file name (corresponds to filename field in the message), e.g., filename="C:\Users\test\Desktop\11.txt".
dwContentLen	DWORD	Content size
pContent	char*	Data point
bySelfRead	BYTE	0-External file, 1-Internal data, whose address is specified by szFilename .
byRes	Array of BYTE	Reserved. Set to 0. Maximum: 15 bytes.

See Also

NET_DVR_XML_CONFIG_INPUT

A.35 NET_DVR_NETCFG_V50

Network Configuration Structure

Member	Data Type	Description
dwSize	DWORD	Structure size.
struEtherNet	Array of NET_DVR_ETHERNET_V30	Ethernet interface
struRes1	Array of	Reserved, set to 0.
struAlarmHostIpAddr	NET_DVR_IPADDR_UNION	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	NET_DVR_IPADDR_UNION	IP address of domain name server 1
struDnsServer2IpAddr	NET_DVR_IPADDR_UNION	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	NET_DVR_IPADDR_UNION	Multicast group address
struGatewayIpAddr	NET_DVR_IPADDR_UNION	Gateway address
struPPPoE	NET_DVR_PPPOECFG	PPPoE parameters
byEnablePrivateMulticastDiscovery	BYTE	Private multicast search (SADP): 0-default, 1-enable, 2-disable
byEnableOnvifMulticastDiscovery	BYTE	Onvif multicast search (SADP): 0-default, 1-enable, 2-disable

Member	Data Type	Description
wAlarmHost2IpPort	WORD	Port No. of listening host 2.
struAlarmHost2IpAddr	NET_DVR_IPADDR_UNION	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.36 NET_DVR_POINT_THERM_CFG

Structure About Real-Time Information of Thermometry By Point.

Member	Data Type	Description
fTemperature	float	Current temperature.
struPoint	NET_VCA_POINT	Coordinate of thermometry by point, it is valid when the rule calibration type is point.
byRes	BYTE	Reserved, set to 0.

See Also

NET_DVR_THERMOMETRY_UPLOAD

A.37 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	NET_DVR_IPADDR_UNION	PPPoE IP address

A.38 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.39 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.40 NET_DVR_REALTIME_THERMOMETRY_COND**Structure About Real-Time Temperature Measurement Condition**

Member	Data Type	Description
dwSize	DWORD	Structure size.
dwChan	DWORD	Channel No., starts from 1, 0xffffffff-get all channels.
byRuleID	BYTE	Rule ID, which starts from 1, 0-get all rules.

Member	Data Type	Description
byMode	BYTE	<p>Persistent connection mode: 0-reserved, 1-scheduled mode, 2-temperature difference mode.</p> <p>1-scheduled mode: the device uploads the measured temperature data of each rule per second, including the highest temperature, the lowest temperature, the average temperature, and the temperature difference.</p> <p>2-temperature difference mode: if the difference of the highest temperature, the lowest temperature, the average temperature, or the temperature difference between the previous second and the next second is larger than or equal to two centigrade, the device will upload the highest temperature, the lowest temperature, and the average temperature; if the temperature difference is smaller than two centigrade for one hour or more, the device will upload the highest temperature, the lowest temperature, the average temperature, and the temperature difference.</p>
wInterval	WORD	Uploading interval, range: [1,3600]s, default value: 3600s; this node is supported by temperature difference mode only.
fTemperatureDiff	float	Temperature difference
byRes	Array of BYTE	Reserved, set to 0. The maximum size is 56 bytes.

See Also

NET_DVR_StartRemoteConfig

A.41 NET_DVR_SCHEDULETIME

Structure About Start and End Time Parameters

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

A.42 NET_DVR_SETUPALARM_PARAM_V50**Arming Parameter Structure**

Member	Data Type	Description
dwSize	DWORD	Structure size.
byLevel	BYTE	Arming priority: 0-high, 1-medium, 2-low.
byAlarmInfoType	BYTE	Intelligent traffic alarm information type: 0-old (NET_DVR_PLATE_RESULT),1-new (NET_ITS_PLATE_RESULT).
byRetAlarmTypeV40	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: NET_DVR_ALARMINFO_V30); 1- alarm information is uploaded in variable size (alarm type: COMM_ALARM_V40, alarm information structure: NET_DVR_ALARMINFO_V40).
byRetDevInfoVersion	BYTE	Alarm types of CVR: 0-COMM_ALARM_DEVICE (alarm information structure: NET_DVR_ALARMINFO_DEV), 1-COMM_ALARM_DEVICE_V40 (alarm information structure: NET_DVR_ALARMINFO_DEV_V40).
byRetVQDAlarmType	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_

Member	Data Type	Description
		ALARM, including camera information and captured pictures)
byFaceAlarmDetection	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).
bySupport	BYTE	Capabilities, which is represented by bit: <ul style="list-style-type: none"> • bit0-whether to upload picture: 0-yes, 1-no • bit1-whether to enable ANR: 0-no, 1-yes • 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. • 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.
byBrokenNetHttp	BYTE	ANR type, which is represented by bit and should be supported by device: <ul style="list-style-type: none"> • bit0-whether to enable ANR for ANPR: 0-no, 1-yes. • bit1-whether to enable ANR for people counting: 0-no, 1-yes. • bit2-whether to enable ANR for heat map: 0-no, 1-yes. • bit3-whether to enable ANR for face capture: 0-no, 1-yes. • bit4-whether to enable ANR for face picture comparison: 0-no, 1-yes. • bit5-whether to enable ANR for JSON message transmission: 0-no, 1-yes. • bit6: whether to enable ANR for uploading heat map data by dwell time duration and by people quantity: 0-no, 1-yes. • bit7: whether to enable ANR for uploading intersection analysis result: 0-no, 1-yes.

Member	Data Type	Description
wTaskNo	BYTE	Task No.
byDeployType	BYTE	Arming type: 0-arm via client software, 1-real-time arming.
bySubScripton	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.
byRes1	Array [BYTE]	Reserved, set to 0. The maximum size is 2 bytes.
byAlarmTypeURL	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"> • bit0-type of captured face pictures: 0-binary data, 1-URL • bit1-type of picture uploaded in message: 0-binary, 1-URL • bit2-type of picture uploaded for face picture comparison: 0-binary, 1-URL
byCustomCtrl	BYTE	Custom control type, which is represented by bit, bit0-whether to upload the face thumbnail of the front passenger: 0-no, 1-yes
byRes4	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.43 NET_DVR_SHIPSDTECTION_ALARM

Structure About Ship Detection Alarm Information

Member	Data Type	Description
dwSize	DWORD	Structure size.
struDevInfo	NET_VCA_DEV_INFO	Front-end device information
dwRelativeTime	DWORD	Time of UTC \pm 00:00, which is valid only when the value of byTimeDiffFlag is "1".
dwAbsTime	DWORD	Local time.
byShipsNum	BYTE	Number of ships (number of line crossed ships).
byShipsNumHead	BYTE	Number of ships (calculate according to ship's bow).
byShipsNumEnd	BYTE	Number of ships (calculate according to ship's stern).
byPicTransType	BYTE	Picture transmission type:-binary data, 1-URL.
struShipInfo	Array of NET_DVR_SHIPSINFO	Ship information, up to 20 ships are supported.
dwPicLen	DWORD	Size of visible light picture.
dwThermalPicLen	DWORD	Thermal picture size.
pPicBuffer	char*	Pointer of visible light picture data.
pThermalPicBuffer	char*	Pointer of thermal picture data.
wDevInfolvmsChannel Ex	WORD	Extended the parameter bylvmsChannel in NET_VCA_DEV_INFO , its value range is extended.
byTimeDiffFlag	BYTE	Whether the time difference parameter is valid: 0-invalid, 1-valid.

Member	Data Type	Description
cTimeDifferenceH	signed 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 byTimeDiffFlag is "1".
cTimeDifferenceM	signed 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 byTimeDiffFlag is "1".
sceneName	char	Scene name, up to 32 characters are supported.
SID	BYTE	Scene ID.
byRes	Array of BYTE	Reserved, set to 0.

A.44 NET_DVR_SHIPSINFO

Ship Information Structure

Member	Data Type	Description
fShipsLength	float	Ship length, the value is between 1 and 1000.0, which corrects to one decimal place, unit: m.
fShipsHeight	float	Ship height, the value is between 1 and 1000.0, which corrects to one decimal place, unit: m.
fShipsWidth	float	Ship width, the value is between 1 and 1000.0, which corrects to one decimal place, unit: m.
fShipsSpeed	float	Ship speed, the value is between 1 and 1000.0, which corrects to one decimal place, unit: m/s.
byShipsDirection	BYTE	Ship direction (on the picture): 0-up, 1-down, 2-left, 3-right
byShipsDetState	BYTE	Ship detection mode: 0-line crossing, 1-ship's bow, 2-ship's stern
byTriggerLineID	BYTE	Detection line ID
byRes	Array of BYTE	Reserved, set to 0.
struShipsRect	NET_VCA_POLYGON	Ship region coordinates (normalized)

See Also***NET_DVR_SHIPSDETECTION_ALARM*****A.45 NET_DVR_SMOKEDETECTION_ALARM****Structure About the Smoke Detection Alarm Information**

Member	Data Type	Description
struPTZPos	<i>NET_PTZ_INFO</i>	PTZ position coordinates of visible light device
struThermalPTZPos	<i>NET_PTZ_INFO</i>	PTZ position coordinates of thermal camera
struLLPos	<i>NET_DVR_LLPOS_PARAMETER</i>	Longitude and latitude information of device
struSmokePos	<i>NET_VCA_RECT</i>	Smoke region coordinates in the image
byRes	Array of BYTE	Reserved, set to 0.

See Also***NET_DVR_FIREDETECTION_ALARM*****A.46 NET_DVR_SMOKEDETECTION_CFG****Structure About Sensitivity Parameter of Smoke Detection**

Member	Data Type	Description
byEnable	BYTE	Whether to enable: 0-no, 1-yes.
bySensitivity	BYTE	Detection sensitivity, ranges from 1 to 100, set to 50 by default.
byRes	Array of BYTE	Reserved.

See Also***NET_DVR_FIREDETECTION_CFG*****A.47 NET_DVR_STD_ABILITY**

Input and Output Parameter Structure for Getting Capabilities

Member	Data Type	Description
lpCondBuffer	LPVOID	Condition parameters (ASCII character format), e.g., the channel No., it can be set to "null".
dwCondSize	DWORD	Buffer size of condition parameters.
lpOutBuffer	LPVOID	Output parameters buffer (the parameter is returned in the message with XML format), it cannot be set to "null".
dwOutSize	DWORD	Output buffer size.
lpStatusBuffer	LPVOID	Get the returned status parameters (<i><code>XML_ResponseStatus</code></i>) when getting capabilities failed. It can be set to null.
dwStatusSize	DWORD	Status buffer size.
dwRetSize	DWORD	Obtained data size (if the capability is obtained, the value refers to the size of lpOutBuffer ; if getting failed, the value refers to the size of lpStatusBuffer).
byRes	Array [BYTE]	Reserved. The maximum size is 32 bytes.

Remarks

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

A.48 NET_DVR_STD_CONFIG**Structure About Configuring Input and Output Parameters**

Member	Data Type	Description
lpCondBuffer	LPVOID	Condition parameters, e.g., channel No., it can be set to "NULL".
dwCondSize	DWORD	Size of buffer for storing condition parameters
lpInBuffer	LPVOID	Input parameters (a structure)
dwInSize	DWORD	Size of buffer for storing input parameters
lpOutBuffer	LPVOID	Output parameters (a structure)

Member	Data Type	Description
dwOutSize	DWORD	Size of buffer for storing output parameters
lpStatusBuffer	LPVOID	Returned status parameters in XML format, it can be set to NULL.
dwStatusSize	DWORD	Size of buffer for storing status parameters
lpXmlBuffer	LPVOID	Request or response message in XML format, it is valid when byDataType is 1.
dwXmlSize	DWORD	Size of memory pointed by lpXmlBuffer .
byDataType	BYTE	Input or output parameter type: 0-valid when the input or output parameters is a structure; 1-valid when the input or output parameters is a XML message.
byRes	Array [BYTE]	Reserved, set to 0. The maximum size is 32 bytes.

A.49 NET_DVR_TEMPERATURE_COLOR

Temperature Alarm Color Structure

Member	Data Type	Description
byType	BYTE	<p>Temperature alarm color types: 0-no alarm, 1-high temperature alarm, 2-low temperature alarm, 3-temperature range alarm, 4-thermal insulation alarm</p> <ul style="list-style-type: none">• For 1-high temperature alarm, iHighTemperature is valid. When the actual temperature is higher than this value, it will be marked by color.• For 2-low temperature alarm, iLowTemperature is valid. When the actual temperature is lower than this value, it will be marked by color.• For 3-temperature range alarm, both the iHighTemperature and iLowTemperature are

Member	Data Type	Description
		valid. When the actual temperature is in this range, it will be marked by color. <ul style="list-style-type: none">For 4-thermal insulation alarm, both the iHighTemperature and iLowTemperature are valid. When the actual temperature is not in this range, it will be marked by color.
byRes1	Array of BYTE	Reserved, set to 0
iHighTemperature	int	High temperature, ranges from -273 to 10000.
iLowTemperature	int	Low temperature, ranges from -273 to 10000
byRes	Array of BYTE	Reserved, set to 0.

See Also***NET_DVR_THERMOMETRY_BASICPARAM***

A.50 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.51 NET_DVR_TIME_EX

Extended Time Parameter Structure

Member	Data Type	Description
wYear	WORD	Year
byMonth	BYTE	Month

Member	Data Type	Description
byDay	BYTE	Day
byHour	BYTE	Hour
byMinute	BYTE	Minute
bySecond	BYTE	Second
byRes	BYTE	Reserved.

A.52 NET_DVR_THERMAL_PIP

Thermal Picture-in-Picture Configuration Structure

Member	Data Type	Description
dwSize	DWORD	Structure size.
byEnable	BYTE	Whether to enable: 0-no, 1-yes.
byPipMode	BYTE	Picture-in-picture mode: 0-overlay mode, 1-fusion mode, 2-normal mode
byOverlapType	BYTE	Overlay type, it is valid when byPipMode is "0": 0-overlay visible light picture on thermal picture, 1-overlay thermal picture on visible light picture.
byTransparency	BYTE	Transparency, ranges from 0 to 100.
struPipRegion	NET_VCA_POLYGON	Picture-in-picture region
byImageFusionRatio	BYTE	Image fusion ration, ranges from 0 to 100, default value is 75. It is valid when byPipMode is "1-fusion mode".
byBorderFusionRatio	BYTE	Boarder fusion ration, ranges from 0 to 100, default value is 100. It is valid when byPipMode is "1-fusion mode".
byRes1	Array of BYTE	Reserved, set to 0.
fDistance	float	Fusion distance, it is valid when byPipMode is "1", ranges from 0.1 m to 4.0 m.
byRes	Array of BYTE	Reserved, set to 0.

See Also*NET_DVR_GetSTDConfig**NET_DVR_SetSTDConfig***A.53 NET_DVR_THERMINTELL_PARAM****Structure About Thermal Intelligent Mutex Configuration**

Member	Data Type	Description
dwSize	DWORD	Structure size
byIntellType	BYTE	Intelligent function types: 0-temperature measurement+behavior analysis (default), 1-ship detection, 2-fire source detection, 3-PIP (picture in picture), 4-body thermometry, 5-temperature measurement+fire and smoke detection, 6-temperature measurement+fire source detection, 7-behavior analysis
byRes	Array of BYTE	Reserved, set as 0.

Related API*NET_DVR_GetSTDConfig**NET_DVR_SetSTDConfig***A.54 NET_DVR_THERMOMETRY_ALARM****Temperature Measurement Alarm Information Structure**

Member	Data Type	Description
dwSize	DWORD	Structure size
dwChannel	DWORD	Channel No.
byRuleID	BYTE	Body thermometry rule ID, from 1 to 40
byThermometryUnit	BYTE	Temperature unit: 0-Celsius (°C), 1-Fahrenheit (°F), 2-Kelvin(K)
wPresetNo	WORD	Preset No.
struPtzInfo	<i>NET_PTZ_INFO</i>	PTZ information
byAlarmLevel	BYTE	Alarm level: 0-prealarm, 1-alarm

Member	Data Type	Description
byAlarmType	BYTE	Alarm type: 0-highest temperature, 1-lowest temperature, 2-average temperature, 3-temperature differences, 4-sudden rise in temperature, 5-sudden reduction in temperature
byAlarmRule	BYTE	Alarm rule: 0-larger than, 1-smaller than
byRuleCalibType	BYTE	Rule calibration type: 0-point, 1-rectangle, 2-line
struPoint	<i>NET_VCA_POINT</i>	Coordinate of temperature measuring point (it is valid when byRuleCalibType is "0").
struRegion	<i>NET_VCA_POLYGON</i>	Coordinate of temperature measuring region (it is valid when byRuleCalibType is "1" or "2").
fRuleTemperature	float	Configured rule temperature, corrects to one decimal place, which ranges from -40.0 to 1000.0.
fCurrTemperature	float	Current temperature, corrects to one decimal place, which ranges from -40.0 to 1000.0.
dwPicLen	DWORD	Visible light picture data length.
dwThermalPicLen	DWORD	Thermal picture data length
dwThermalInfoLen	DWORD	Additional information length.
pPicBuff	char*	Visible light picture pointer, which is used to save the visible light picture data with JPEG format.
pThermalPicBuff	char*	Thermal picture pointer, used to save the thermal imaging picture data with JPEG format
pThermalInfoBuff	char*	Additional information pointer, which is used to save the thermal information
struHighestPoint	<i>NET_VCA_POINT</i>	Coordinate of highest temperature position on the measuring region, it is valid when byRuleCalibType is "1" or "2").
fToleranceTemperature	float	Temperature tolerance, corrects to one decimal place, which ranges from -40.0 to 1000.0.

Member	Data Type	Description
dwAlertFilteringTime	DWORD	Temperature prealarm filtering time, unit: second, which ranges from 0 to 200 seconds, it is 0 second by default.
dwAlarmFilteringTime	DWORD	Temperature alarm filtering time, unit: second, which ranges from 0 to 200 seconds, it is 0 second by default.
dwTemperatureSuddenChangeCycle	DWORD	Period of temperature sudden change, unit: second.
fTemperatureSuddenChangeValue	float	Picture transmission method: 0-binary, 1-url
byPicTransType	BYTE	Temperature difference, corrects to one decimal place and it should be larger than 0.
byRes1	Array of BYTE	Reserved, set to 0
dwVisibleChannel	DWORD	Visible light channel No.
dwRelativeTime	DWORD	Relative time
dwAbsTime	DWORD	Absolute time
fAlarmRuleTemperature	float	Rule temperature of TMA (temperature measurement alarm)
byRes	Array of BYTE	Reserved, set to 0.

A.55 NET_DVR_THERMOMETRY_ALARMRULE

Structure About Configuring Thermometry Alarm Rule for Preset

Member	Data Type	Description
dwSize	DWORD	Structure size.
struThermometryAlarmRuleParam	NET_DVR_THERMOMETRY_ALARMRULE_PARAM	Thermometry alarm rule parameters of preset, each array indicates one rule.
byRes	Array of BYTE	Reserved, set to 0.

A.56 NET_DVR_THERMOMETRY_ALARMRULE_PARAM

Structure About Alarm Rule When Measuring Temperature by Preset

Member	Data Type	Description
byEnabled	BYTE	Whether to enable: 0-no, 1-yes
byRuleID	BYTE	Rule ID
byRule	BYTE	Temperature comparison mode: 0-high temperature higher than, 1-high temperature lower than, 2-low temperature higher than, 3-low temperature lower than, 4-average temperature higher than, 5-average temperature lower than, 6-temperature difference higher than, 7-temperature difference lower than.
byRes	Array of BYTE	Reserved, set to 0.
byReflectiveEnabled	BYTE	Whether to enable: 0-no, 1-yes
fReflectiveTemperature	float	Reflective temperature
szRuleName	Array of char	Rule name.
fAlert	float	Pre-alarm temperature
fAlarm	float	Alarm temperature.
fThreshold	float	Temperature threshold. E.g., if the configured alarm temperature is 60 °C, and the temperature threshold is set to 5 °C, the alarm will be triggered when the actual temperature is between 55 °C and 65 °C.
dwAlertFilteringTime	DWORD	Temperature pre-alarm dwell time
dwAlarmFilteringTime	DWORD	Temperature alarm dwell time
byRes1	Array of BYTE	Reserved, set to 0.

See Also***NET_DVR_THERMOMETRY_ALARMRULE*****A.57 NET_DVR_THERMOMETRY_BASICPARAM**

Structure About Basic Parameter of Temperature Measurement

Member	Data Type	Description
dwSize	DWORD	Structure size.
byEnabled	BYTE	Whether to enable: 0-no, 1-yes.
byStreamOverlay	BYTE	Whether to display temperature on the video: 0-no, 1-yes.
byPictureOverlay	BYTE	Whether to display temperature on picture: 0-no, 1-yes.
byThermometryRange	BYTE	Temperature range (default unit: °C. For other kinds of units, you should transform by yourself): 0-default value, 1-(-20 to 150), 2-(0 to 550) , 3-(0 to 650), 4-(20 to 650), 5-(-20 to 150), 6-(-20 to 120), 7-(20 to 350), 8-(20 to 45), 0xff-auto.
byThermometryUnit	BYTE	Temperature unit: 0-Celsius (°C), 1-Fahrenheit (°F), 2-Kelvin (K).
byThermometryCurve	BYTE	Temperature curve: 0-reserved, 1-mode 1 (transverse temperature curve), 2- mode 2 (longitudinal temperature curve).
byFireImageModea	BYTE	Thermal picture mode: 0-reserved, 1-black and white mode, 2-thermal detection mode, 3-fire ground mode
byShowTempStripEnable	BYTE	Whether to enable displaying temperature bar: 0-no, 1-yes.
fEmissivity	float	Emissivity (the radiating energy capability of object, which is accurate to two decimal places), which ranges from 0.01 to 1.00.
byDistanceUnit	BYTE	Distance unit: 0-meter, 1-feet, 2-centimeter.
byEnviroHumidity	BYTE	Distance unit: 0-meter, 1-feet
byRes2	Array of BYTE	Reserved, set to 0.
struTempColor	NET_DVR_TEMPERATURE_COLOR	Temperature alarm color
iEnviroTemperature	int	Environment humidity, ranges from -273 to 10000 degrees centigrade.

Member	Data Type	Description
iCorrectionVolume	int	Temperature correction, ranges from -100 to 100.
bySpecialPointThermType	BYTE	Display specific temperature measurement point, indicated by bit: <ul style="list-style-type: none"> bit0-whether to display the point with medium temperature: 0-no, 1-yes bit1-whether to display the point with highest temperature: 0-no, 1-yes bit2-whether to display the point with lowest temperature: 0-no, 1-yes
byReflectiveEnabled	BYTE	Whether enables reflecting temperature: 0-no, 1-yes.
wDistance	WORD	Distance, unit: meter, ranges from 0 to 10000.
fReflectiveTemperature	float	Reflected temperature, corrects to one decimal place.
fAlert	float	Pre-alarm temperature threshold, ranges from -100.0 to 1000.0 degrees centigrade (corrects to one decimal place).
fAlarm	float	Alarm temperature threshold, ranges from -100.0 to 1000.0 degrees centigrade (corrects to one decimal place).
fThermalOpticalTransmittance	float	Optical transmissivity, which corrects to three decimal places, from 0.001 to 1.000, the default value is 1.000 .
fExternalOpticsWindowCorrection	float	External optical temperature is between -40.0 °C and 80.0 °C. The default value is 20 °C. It corrects to three decimal places.
byDisplayMaxTemperatureEnabled	BYTE	Whether displays the maximum temperature: 0-no, 1-yes.
byDisplayMinTemperatureEnabled	BYTE	Whether displays the minimum temperature: 0-no, 1-yes.
byDisplayAverageTemperatureEnabled	BYTE	Whether displays the average temperature: 0-no, 1-yes.
byThermometryInfoDisplayposition	BYTE	Thermometry information overlay position: 0-reserved, 1-near target, 2-on the top left corner of screen

Member	Data Type	Description
dwAlertFilteringTime	BYTE	Temperature pre-alarm dwell time, unit: second
dwAlarmFilteringTime	BYTE	Temperature alarm dwell time, unit: second
byemissivityMode	BYTE	Emissivity type: 1-rougher 0.95, 2-rough 0.80, 3-smooth 0.60, 4-smoother 0.30, 0xff-customized setting, ranges from 0.01 to 1.00, the larger the value, the higher the roughness
bydisplayTemperatureInOpticalChannelEnabled	BYTE	Display the temperature information of optical channel or not: 0-not display, 1-display
byRes	Array of BYTE	Reserved, set to 0.

Remarks

For "normal" temperature measurement mode, this structure is invalid.

A.58 NET_DVR_THERMOMETRY_COND

Thermometry Condition Structure

Member	Data Type	Description
dwSize	DWORD	Structure size
dwChannel	DWORD	Channel No.
wPresetNo	WORD	Preset No.
byRes	Array of BYTE	Reserved, set to 0.

Remarks

The temperature measurement function is based on the preset, you need to set the preset before configuring other parameters, call API **NET_DVR_PTZPreset_Other** to set preset. Get the device supported number of presets via the device capability by calling API: **NET_DVR_GetDeviceAbility**, capability type (**dwAbilityType**): DEVICE_ABILITY_INFO, related node: <maxThermometryPresetNum>.

Related API

NET_DVR_GetSTDConfig

NET_DVR_SetSTDConfig

A.59 NET_DVR_THERMOMETRY_DIFF_ALARM

Structure About Information of Temperature Difference Alarm

Member	Data Type	Description
dwSize	DWORD	Structure size
dwChannel	DWORD	Channel No.
byAlarmID1	BYTE	Alarm 1
byAlarmID2	BYTE	Alarm 2
wPresetNo	WORD	Preset No.
byAlarmLevel	BYTE	Alarm level: 0-prealarm, 1-alarm
byAlarmType	BYTE	Alarm type: 0-highest temperature, 1-lowest temperature, 2-average temperature, 3-temperature differences, 4-sudden rise in temperature, 5-sudden reduction in temperature
byAlarmRule	BYTE	Alarm rule: 0-larger than, 1-smaller than
byRuleCalibType	BYTE	Rule calibration type: 0-point, 1-rectangle, 2-line
struPoint	Array of NET_VCA_POINT	Coordinate of temperature measuring point (it is valid when byRuleCalibType is "0").
struRegion	Array of NET_VCA_POLYGON	Coordinate of temperature measuring region (it is valid when byRuleCalibType is "1" or "2").
fRuleTemperatureDiff	float	Rule temperature difference, corrects to one decimal place, ranges from -40.0 to 1000.0.
fCurTemperatureDiff	float	Current temperature difference, corrects to one decimal place, ranges from -40.0 to 1000.0.
struPtzInfo	NET_PTZ_INFO	PTZ position information, see details in structure .
dwPicLen	DWORD	Visible light picture data length.
dwThermalPicLen	DWORD	Thermal picture data length.
dwThermalInfoLen	DWORD	Additional information length.
pPicBuff	char*	Visible light picture pointer, used to save the visible light picture data with JPEG format.

Member	Data Type	Description
pThermalPicBuff	char*	Thermal imaging picture pointer, used to save the thermal picture data with JPEG format.
pThermalInfoBuff	char*	Pointer to thermal additional information, which is used to save the thermal information.
byThermometryUnit	BYTE	Thermometry unit
byPicTransType	BYTE	Picture transmission method: 0-binary, 1-url
byRes1	Array of BYTE	Reserved, set to 0.
fToleranceTemperature	float	Temperature tolerance, corrects to one decimal place, ranges from -40.0 to 1000.0.
dwAlarmFilteringTime	DWORD	Temperature alarm filtering time, unit: second, ranges from 0 to 200 seconds, it is 0 second by default.
byRes	Array of BYTE	Reserved, set to 0.

A.60 NET_DVR_THERMOMETRY_DIFFCOMPARISON

Structure About Configuring Temperature Difference Alarm Rule Based on Preset

Member	Data Type	Description
dwSize	DWORD	Structure size.
struDiffComparison	NET_DVR_THERMOMETRY_DIFFCOMPARISON_PARAM	Temperature difference alarm rule, each array indicates one rule
byRes	BYTE	Reserved, set to 0.

Related API

NET_DVR_GetSTDConfig

NET_DVR_SetSTDConfig

A.61 NET_DVR_THERMOMETRY_DIFFCOMPARISON_PARAM

Structure About Temperature Difference Alarm Rule Information Based on Preset

Member	Data Type	Description
byEnabled	BYTE	Whether to enable: 0-no, 1-yes.
byRuleID	BYTE	Rule ID
byAlarmID1	BYTE	Alarm 1
byAlarmID2	BYTE	Alarm 2
byRule	BYTE	Alarm temperature comparison mode: 0-high temperature higher than, 1-high temperature lower than, 2-low temperature higher than, 3-low temperature lower than, 4-average temperature higher than, 5-average temperature lower than, 6-temperature difference higher than, 7-temperature difference lower than.
byRes	Array of BYTE	Reserved, set to 0.
fTemperatureDiff	float	Temperature difference
dwAlarmFilteringTime	DWORD	Temperature alarm dwell time
byRes1	Array of BYTE	Reserved, set to 0.

Remarks

Temperature difference parameters **byAlarmID1** and **byAlarmID2** correspond to the thermometry alarm parameter **byRuleID**, which refers to the comparison between two specified temperature difference alarm rules.

See Also

NET_DVR_THERMOMETRY_DIFFCOMPARISON

A.62 NET_DVR_THERMOMETRY_MODE**Temperature Measurement Mode Structure**

Member	Data Type	Description
dwSize	DWORD	Structure size
byMode	BYTE	Temperature measurement mode: 0-normal mode, 1-expert mode

Member	Data Type	Description
byThermometryROIEnabled	BYTE	Whether to enable ROI temperature measurement: 0-reserved, 1-no, 2-yes.
byRes	Array of BYTE	Reserved, set to 0

Remarks

- Normal mode: Configure overall temperature measurement without distinguishing rule and preset. In normal mode, the related configuration in the basic parameter configuration structure **NET_DVR_THERMOMETRY_BASICPARAM** is valid.
- Expert mode: Configure temperature measurement configuration by rule and preset. In expert mode, the related configuration in the preset configuration structure **NET_DVR_THERMOMETRY_PRESETINFO** is valid.
- The parameters **byThermometryROIEnabled** and **byMode** are mutually exclusive.

Related API*NET_DVR_GetSTDConfig**NET_DVR_SetSTDConfig*

A.63 NET_DVR_THERMOMETRY_PRESETINFO

Structure About Thermometry Preset Configuration

Member	Data Type	Description
dwSize	DWORD	Structure size.
wPresetNo	WORD	Scene No. (preset No.).
byRes	Array of BYTE	Reserved, set to 0.
struPresetInfo	Array of NET_DVR_THERMOMETRY_PRESETINFO_PARAM	Thermometry preset information, each array indicates one thermometry information. Currently each preset supports up to 10 rectangle frames, 10 points, and one line.

Related API*NET_DVR_GetSTDConfig**NET_DVR_SetSTDConfig*

A.64 NET_DVR_THERMOMETRY_PRESETINFO_PARAM

Thermometry Preset Information Structure

Member	Data Type	Description
byEnabled	BYTE	Whether to enable: 0-no, 1-yes.
byRuleID	BYTE	rule ID, which starts from 1, and 0- invalid.
wDistance	WORD	Distance (unit: m), range: [0, 10000]
fEmissivity	float	Emissivity (Emissivity refers to a material's effectiveness in emitting energy. Here the value of emissivity is accurate to two decimal places), range: [0.01, 1.00].
byDistanceUnit	BYTE	Distance unit: 0-meter, 1-feet, 2-centimeter
byRes	Array of BYTE	Reserved, set to 0.
byReflectiveEnabled	BYTE	whether to enable reflecting temperature: 0-no, 1-yes.
fReflectiveTemperature	float	Reflective temperature. Here the value of reflective temperature is accurate to one decimal place.
szRuleName	char	Rule name.
byemissivityMode	BYTE	Emissivity type: 1-rougher 0.95, 2-rough 0.80, 3-smooth 0.60, 4-smoother 0.30, 0xff-customized setting, ranges from 0.01 to 1.00, the larger the value, the higher the roughness
byRes1	Array of BYTE	Reserved, set to 0.
byRuleCalibType	BYTE	Rule calibration types: 0-point, 1-frame, 2-line.
struPoint	NET_VCA_POINT	Point coordinate, it is valid when byRuleCalibType is "0".
struRegion	NET_VCA_POLYGON	Region/line coordinate, it is valid when byRuleCalibType is "1" or "2".

Remarks

For region temperature measurement, the maximum temperature, the minimum temperature, the average temperature, and the temperature difference will display; for line temperature measurement, the maximum temperature and the minimum temperature will display; for point temperature measurement, only the average temperature will display.

A.65 NET_DVR_THERMOMETRYRULE_TEMPERATURE_INFO

Structure of Thermometry Rule Temperature Information

Member	Data Type	Description
fMaxTemperature	float	The highest temperature, corrects to one decimal place.
fMinTemperature	float	The lowest temperature, corrects to one decimal place.
fAverageTemperature	float	The average temperature, corrects to one decimal place.
struHighestPoint	NET_VCA_POINT	Coordinates of the highest temperature position
struLowestPoint	NET_VCA_POINT	Coordinates of the lowest temperature position
byIsFreezedata	BYTE	Whether supports freezing data: 0-no, 1-yes
byRes	Array of BYTE	Reserved, set to 0.

A.66 NET_DVR_THERMOMETRY_TRIGGER_COND

Structure about Thermometry Linkage Configuration Condition

Member	Data Type	Description
dwSize	DWORD	Structure size.
dwChan	DWORD	Channel No.
dwPreset	DWORD	Preset No.
byRes	Array of BYTE	Reserved, set to 0.

Related API

NET_DVR_GetSTDConfig

NET_DVR_SetSTDConfig

A.67 NET_DVR_THERMOMETRY_UPLOAD

Structure of Real-Time Temperature Information

Member	Data Type	Description
dwSize	DWORD	Structure size.
dwRelativeTime	DWORD	Time of UTC \pm 00:00.
dwAbsTime	DWORD	Local time.
szRuleName	Array of char	Rule name.
byRuleID	BYTE	Rule ID.
byRuleCalibType	BYTE	Rule calibration type: 0-point, 1-frame, 2-line.
wPresetNo	WORD	Preset number.
struPointThermCfg	NET_DVR_POINT_THERM_CFG	Temperature measuring by point, it is valid when "byRuleCalibType" is "0".
struLinePolygonThermCfg	NET_DVR_LINEPOLYGON_THERM_CFG	Temperature measuring by frame or line, valid when "byRuleCalibType" is "1" or "2".
byThermometryUnit	BYTE	Unit: 0-Degree Centigrade, 1-Degree Fahrenheit, 2-Kelvin.
byDataType	BYTE	Data status type: 0-measuring, 1-started, 2-ended.
byRes1	BYTE	Reserved, set to 0.
bySpecialPointThermType	BYTE	Whether supports measuring temperature by special point (indicated by bit): bit0-medium temperature point measurement: 0-not support, 1-support; bit1-highest temperature point measurement: 0-not support, 1-support; bit2- lowest temperature point measurement: 0-not support, 1-support.
fCenterPointTemperature	BYTE	Medium temperature (check whether it is supported via the value of "bySpecialPointThermType"), which corrects to one decimal place.
fHighestPointTemperature	BYTE	The highest temperature (check whether it is supported via the value of "bySpecialPointThermType"), which corrects to one decimal place.

Member	Data Type	Description
fLowestPointTemperature	BYTE	The lowest temperature (check whether it is supported via the value of "bySpecialPointThermType"), which corrects to one decimal place.
struHighestPoint	NET_VCA_POINT	The coordinates of highest temperature position for thermometry by frame/line (it is valid when the "byRuleCalibType" is "1-frame" or "2-line").
struLowestPoint	NET_VCA_POINT	The coordinates of lowest temperature position for thermometry by frame/line (it is valid when the "byRuleCalibType" is "1-frame" or "2-line").
byIsFreezedata	BYTE	Whether supports freezing data: 0-no, 1-yes.
byFaceSnapThermometryEnabled	BYTE	Whether to enable uploading captured face picture with temperature information: 1-enable, 0-disable.
byRes2	Array of BYTE	Reserved, set to 0.
dwChan	DWORD	Channel No.
struFaceRect	NET_VCA_RECT	Face thumbnail rectangle
dwTimestamp	DWORD	DSP timestamp
byRes	Array of BYTE	Reserved, set to 0, the maximum size is 68 bytes

Remarks

Check whether the device supports this function via the device capability set (***XML_ThermalCap***), API: ***NET_DVR_GetSTDAbility*** , capability set: NET_DVR_GET_THERMAL_CAPABILITIES, node: <ThermalCap>.

A.68 NET_DVR_USER_LOGIN_INFO

Structure About Login Parameters

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

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

A.68.1 fLoginResultCallBack

Login Status Callback Function

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

A.69 NET_DVR_XML_CONFIG_INPUT

Input Parameter Structure of Message Transmission API (NET_DVR_STDXMLConfig)

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

Related API*NET_DVR_STDXMLConfig***A.70 NET_DVR_XML_CONFIG_OUTPUT****Output Parameter Structure of Message Transmission API (NET_DVR_STDXMLConfig)**

Member	Data Type	Description
dwSize	DWORD	Structure size.
lpOutBuffer	void*	Buffer for storing output parameters (response messages), which is allocated when passing through URL by GET method.
dwOutBufferSize	DWORD	Output buffer size.
dwReturnedXMLSize	DWORD	Actual size of response message.
lpStatusBuffer	void*	Response status (ResponseStatus message). This parameter will not be assigned if performing GET operation succeeded, and you can also set it to "NULL" if not required.
dwStatusSize	DWORD	Size of response status buffer.
byRes	Array of BYTE	Reserved, set to 0.

Related API*NET_DVR_STDXMLConfig***A.71 NET_PTZ_INFO****PTZ Parameter (Thermal Camera Position Information) Structure**

Member	Data Type	Description
fPan	float	Panning parameter (horizontal parameter), range: [0, 36.000], which is accurate to 3 decimal places.
fTilt	float	Tilting parameter (vertical parameter), range: [-90.000, 270.000], which is accurate to 3 decimal places.

Member	Data Type	Description
fZoom	float	Zooming parameter, range: [0, 100000], which is accurate to 3 decimal places.
dwFocus	DWORD	Focus parameter, focus range: normalization [0, 100000].
byRes	Array of BYTE	Reserved.

A.72 NET_SDK_CALLBACK_STATUS_NORMAL

Enumeration About Persistent Connection Status

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

A.73 NET_SDK_MANUAL_THERMOMETRY

Structure of Manual Thermometry Configuration Parameters

Member	Data Type	Description
dwSize	DWORD	Structure size
dwChannel	DWORD	Channel No
dwRelativeTime	DWORD	Time of UTC ± 00:00
dwAbsTime	DWORD	Local time
byThermometryUnit	BYTE	Temperature measurement unit: 0- degrees Centigrade, 1- degrees Fahrenheit, 2- Kelvin.
byDataType	BYTE	Data status: 0-detecting, 1-start, 2-end (read-only).

Member	Data Type	Description
byRes1	Array of BYTE	Reserved, set to 0.
struRuleInfo	NET_SDK_MANUALTHERM_RULE	Manual thermometry rule, see details in structure .
byRes	Array of BYTE	Reserved, set to 0.

See Also**NET_SDK_MANUAL_THERMOMETRY**

A.74 NET_SDK_MANUALTHERM_BASICPARAM

Structure of Manual Thermometry Basic Parameters

Member	Data Type	Description
dwSize	DWORD	Structure size
wDistance	WORD	Distance, unit: m, ranges from 0 to 10000
byRes1	Array of BYTE	Reserved, set to 0
fEmissivity	float	Emissivity, the ability of a surface to radiate energy, accurate to 2 decimal places, ranges from 0.01 to 1.00
byRes	Array of BYTE	Reserved, set to 0

Related API**NET_DVR_GetSTDConfig****NET_DVR_SetSTDConfig**

A.75 NET_SDK_MANUALTHERM_RULE

Manual Thermometry Rule Structure

Member	Data Type	Description
byRuleID	BYTE	Rule ID, 0-invalid, starts from 1
byEnable	BYTE	Enable or not: 0-no, 1-yes
byRes1	Array of BYTE	Reserved, set to 0
szRuleName	Array of char	Rule name

Member	Data Type	Description
byRuleCalibType	BYTE	Rule calibration type: 0-point, 1-frame, 2-line
byRes2	BYTE	Reserved, set to 0
struPointTherm	<i>NET_SDK_POINT_THERMOMETRY</i>	Point temperature measurement, it is valid when byRuleCalibType values "0".
struRegionTherm	<i>NET_SDK_REGION_THERMOMETRY</i>	Region temperature measurement, it is valid when byRuleCalibType values "1" or "2".
byRes	Array of BYTE	Reserved, and set to 0

See Also

NET_SDK_MANUAL_THERMOMETRY

A.76 NET_SDK_POINT_THERMOMETRY

Structure of Thermometry by Point

Member	Data Type	Description
fPointTemperature	float	Current temperature, corrects to one decimal place, ranges from -40 °C to +1000 °C, (floating number +100) × 10
struPoint	<i>NET_VCA_POINT</i>	Point coordinates
byRes	BYTE	Reserved, and set to 0.

See Also

NET_SDK_MANUAL_THERM_RULE

A.77 NET_SDK_REGION_THERMOMETRY

Region Thermometry Structure

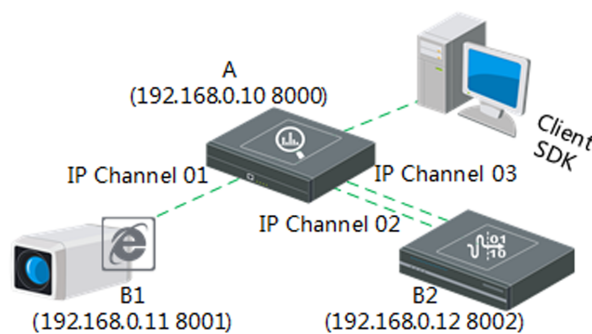
Member	Data Type	Description
fMaxTemperature	float	The max. temperature, corrects to 1 decimal place, ranges from -40 °C to +1000 °C, (floating number +100) × 10
fMinTemperature	float	The min. temperature, corrects to 1 decimal place, ranges from -40 °C to +1000 °C, (floating number +100) × 10
fAverageTemperature	float	The average temperature, corrects to 1 decimal place, ranges from -40 °C to +1000 °C, (floating number +100) × 10
fTemperatureDiff	float	Temperature difference, corrects to 1 decimal place, ranges from -40 °C to +1000 °C, (floating number +100) × 10
struRegion	NET_VCA_POLYGON	Region or line parameters
byRes	Array of BYTE	Reserved, set to 0.

See Also**NET_SDK_MANUALTHERM_RULE****A.78 NET_VCA_DEV_INFO****Structure About Camera Information**

Member	Data Type	Description
struDevIP	NET_DVR_IPADDR_UNION	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 **bylvmsChannel** 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; **bylvmsChannel** 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; **bylvmsChannel** 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 **bylvmsChannel** is 3 in the received alarm message.

A.79 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.80 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 <i>NET_VCA_POINT</i>	Boundary point of polygon, up to 10 points

A.81 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.

Appendix B. Request URIs

B.1 /ISAPI/Event/notification/subscribeEvent

Subscribe events/alarms in arming mode.

Request URI Definition

Table B-1 POST /ISAPI/Event/notification/subscribeEvent

Method	POST
Description	Subscribe events/alarms in arming mode.
Query	None.
Request	<i>XML_SubscribeEvent</i>
Response	Succeeded: <i>XML_SubscribeEventResponse</i> or <i>XML_EventNotificationAlert_SubscriptionHeartbeat</i> and alarm/event details message Failed: <i>XML_ResponseStatus</i>

Remarks

The ***XML_EventNotificationAlert_SubscriptionHeartbeat*** and alarm/event details message is uploaded repeatedly. The default time interval of uploading heartbeat information is 30s.

B.2 /ISAPI/Event/notification/subscribeEvent/<ID>

Get or set alarm/event subscription parameters.

Request URI Definition

Table B-2 GET /ISAPI/Event/notification/subscribeEvent/<ID>

Method	GET
Description	Get alarm/event subscription parameters.
Query	None.
Request	None.
Response	Succeeded: <i>XML_SubscribeEvent</i> Failed: <i>XML_ResponseStatus</i>

Table B-3 PUT /ISAPI/Event/notification/subscribeEvent/<ID>

Method	PUT
Description	Set alarm/event subscription parameters.
Query	None.
Request	<i>XML_SubscribeEvent</i>
Response	<i>XML_ResponseStatus</i>

Remarks

The <ID> in the request URI refers to the subscription No. which is returned by the device. After the persistent connection for receiving events or alarms in arming mode is closed, the device will release the resource used by the <ID>.

B.3 /ISAPI/Event/notification/subscribeEventCap

Get event/alarm subscription capability.

Request URI Definition**Table B-4 GET /ISAPI/Event/notification/subscribeEventCap**

Method	GET
Description	Get event/alarm subscription capability.
Query	None.
Request	None.
Response	Succeeded: <i>XML_SubscribeEventCap</i> Failed: <i>XML_ResponseStatus</i>

B.4 /ISAPI/Event/notification/unSubscribeEvent

Cancel subscribing alarm/event.

Request URI Definition**Table B-5 PUT /ISAPI/Event/notification/unSubscribeEvent**

Method	PUT
Description	Cancel subscribing alarm/event.
Query	None.

Request	None.
Response	<i>XML_ResponseStatus</i>

B.5 /ISAPI/Event/schedules/dredgerDetection/<ID>/scene/<SID>

Operations about arming schedule of dredger detection in specified scene.

Request URL Definition

Table B-6 GET /ISAPI/Event/schedules/dredgerDetection/<ID>/scene/<SID>

Method	GET
Description	Get the arming schedule of dredger detection in specified scene.
Query	None.
Request	None.
Response	Succeeded: <i>XML_Schedule</i> Failed: <i>XML_ResponseStatus</i>

Table B-7 PUT /ISAPI/Event/schedules/dredgerDetection/<ID>/scene/<SID>

Method	PUT
Description	Set the arming schedule of dredger detection in specified scene.
Query	None.
Request	<i>XML_Schedule</i>
Response	<i>XML_ResponseStatus</i>

Remarks

- The <ID> in the URL refers to the channel No.
- The <SID> in the URL refers to the scene ID.

B.6 /ISAPI/Event/schedules/TMPA

Get or set arming schedules of temperature measurement pre-alarm for all channels in a batch.

Request URI Definition**Table B-8 GET /ISAPI/Event/schedules/TMPA**

Method	GET
Description	Get arming schedules of temperature measurement pre-alarm for all channels in a batch.
Query	None.
Request	None.
Response	Succeeded: <i>XML_TMPAScheduleList</i> Failed: <i>XML_ResponseStatus</i>

Table B-9 PUT /ISAPI/Event/schedules/TMPA

Method	PUT
Description	Set arming schedules of temperature measurement pre-alarm for all channels in a batch.
Query	None.
Request	<i>XML_TMPAScheduleList</i>
Response	<i>XML_ResponseStatus</i>

B.7 /ISAPI/Event/schedules/TMPA/<ID>

Get or set the arming schedule of temperature measurement pre-alarm for a specified channel.

Request URI Definition**Table B-10 GET /ISAPI/Event/schedules/TMPA/<ID>**

Method	GET
Description	Get the arming schedule of temperature measurement pre-alarm for a specified channel.
Query	None.
Request	None.
Response	Succeeded: <i>XML_Schedule</i> Failed: <i>XML_ResponseStatus</i>

Table B-11 PUT /ISAPI/Event/schedules/TMPA/<ID>

Method	PUT
Description	Set the arming schedule of temperature measurement pre-alarm for a specified channel.
Query	None.
Request	<i>XML_Schedule</i>
Response	<i>XML_ResponseStatus</i>

Remarks

The <ID> in the request URI refers to temperature measurement pre-alarm ID, which consists of event type and video input channel No., e.g., TMPA-1.

B.8 /ISAPI/Event/schedules/shipsFlowDetection/<ID>/scene/<SID>

Operations about arming schedule of ship flow detection in specified scene.

Request URL Definition**Table B-12 GET /ISAPI/Event/schedules/shipsFlowDetection/<ID>/scene/<SID>**

Method	GET
Description	Get the arming schedule of ship flow detection in specified scene.
Query	None.
Request	None.
Response	Succeeded: <i>XML_Schedule</i> Failed: <i>XML_ResponseStatus</i>

Table B-13 PUT /ISAPI/Event/schedules/shipsFlowDetection/<ID>/scene/<SID>

Method	PUT
Description	Set the arming schedule of ship flow detection in specified scene.
Query	None.
Request	<i>XML_Schedule</i>
Response	<i>XML_ResponseStatus</i>

Remarks

- The <ID> in the URL refers to the channel No.
- The <SID> in the URL refers to the scene ID.

B.9 /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: <i>XML_EventTriggersCap</i> Failed: <i>XML_ResponseStatus</i>

B.10 /ISAPI/Event/triggers/<eventType>-<ID>

Get, set, or delete the alarm linkage action by channel.

Request URI Definition

Table B-15 GET /ISAPI/Event/triggers/<eventType>-<ID>

Method	GET
Description	Get the alarm linkage action by channel.
Query	None
Request	None
Response	Succeeded: <i>XML_EventTrigger</i> Failed: <i>XML_ResponseStatus</i>

Table B-16 PUT /ISAPI/Event/triggers/<eventType>-<ID>

Method	PUT
Description	Set the alarm linkage action by channel.
Query	None
Request	<i>XML_EventTrigger</i>
Response	<i>XML_ResponseStatus</i>

Table B-17 DELETE /ISAPI/Event/triggers/<eventType>-<ID>

Method	DELETE
Description	Delete the alarm linkage action by channel.
Query	None
Request	None
Response	<i>XML_ResponseStatus</i>

Remarks

The <eventType> in the request URI refers to the predefined event or alarm type name, and the <ID> is the No. of the event detection channel. For example, if the No. of the face capture channel is 101, the "<eventType>-<ID>" is "faceSnap-101".

B.11 /ISAPI/Event/triggers/dredgerDetection/<ID>/scene/<SID>

Operations about linkage configurations of dredger detection in specified scene.

Request URL Definition**Table B-18 GET /ISAPI/Event/triggers/dredgerDetection/<ID>/scene/<SID>**

Method	GET
Description	Get the linkage configurations of dredger detection in specified scene.
Query	None.
Request	None.
Response	Succeeded: <i>XML_EventTrigger</i> Failed: <i>XML_ResponseStatus</i>

Table B-19 PUT /ISAPI/Event/triggers/dredgerDetection/<ID>/scene/<SID>

Method	PUT
Description	Set the linkage parameters of dredger detection in specified scene.
Query	None.
Request	<i>XML_EventTrigger</i>
Response	<i>XML_ResponseStatus</i>

Remarks

- The <ID> in the URL refers to the channel No.
- The <SID> in the URL refers to the scene ID.

B.12 /ISAPI/Event/triggers/shipsFlowDetection/<ID>/scene/<SID>

Get or set the linkage configurations of ship flow detection in specified scene.

Request URL Definition**Table B-20 GET /ISAPI/Event/triggers/shipsFlowDetection/<ID>/scene/<SID>**

Method	GET
Description	Get the linkage configurations of ship detection in specified scene.
Query	None.
Request	None.
Response	Succeeded: <i>XML_EventTrigger</i> Failed: <i>XML_ResponseStatus</i>

Table B-21 PUT /ISAPI/Event/triggers/shipsFlowDetection/<ID>/scene/<SID>

Method	PUT
Description	Set the linkage parameters of ship detection in specified scene.
Query	None.
Request	<i>XML_EventTrigger</i>
Response	<i>XML_ResponseStatus</i>

Remarks

- The <ID> in the URL refers to the channel No.
- The <SID> in the URL refers to the scene ID.

B.13 /ISAPI/System/capabilities

Get device capability.

Request URI Definition

Table B-22 GET /ISAPI/System/capabilities

Method	GET
Description	Get device capability.
Query	None
Request	None.
Response	Succeeded: <i>XML_DeviceCap</i> Failed: <i>XML_ResponseStatus</i>

B.14 /ISAPI/Thermal/capabilities

Get thermal capability.

Request URI Definition

Table B-23 GET /ISAPI/Thermal/capabilities

Method	GET
Description	Get the thermal capability.
Query	None.
Request	None.
Response	<i>XML_ThermalCap</i>

B.15 /ISAPI/Thermal/channels/<ID>/blackBody

Operations about black body configuration.

Request URL Definition

Table B-24 GET /ISAPI/Thermal/channels/<ID>/blackBody

Method	GET
Description	Get the black body parameters of a specific camera.
Query	None.
Request	None.
Response	<i>XML_ThermalBlackBody</i>

Table B-25 PUT /ISAPI/Thermal/channels/<ID>/blackBody

Method	PUT
Description	Set the black body parameters of a specific camera.
Query	None.
Request	<i>XML_ThermalBlackBody</i>
Response	<i>XML_ResponseStatus</i>

Remarks

The <ID> in the URL is the channel ID.

B.16 /ISAPI/Thermal/channels/<ID>/blackBody/capabilities

Get black body capability.

Request URL Definition**Table B-26 GET /ISAPI/Thermal/channels/<ID>/blackBody/capabilities**

Method	GET
Description	Get the black body capability.
Query	None.
Request	None.
Response	<i>XML_Cap_ThermalBlackBody</i>

Remarks

The <ID> in the URL is the channel ID.

B.17 /ISAPI/Thermal/channels/<ID>/burningPrevention

Get or set configuration parameters of burning prevention.

Request URL Definition**Table B-27 GET /ISAPI/Thermal/channels/<ID>/burningPrevention**

Method	GET
Description	Get configuration parameters of burning prevention.
Query	None.

Request	None.
Response	<i>XML_BurningPrevention</i>

Table B-28 PUT /ISAPI/Thermal/channels/<ID>/burningPrevention

Method	PUT
Description	Set the burning prevention parameters.
Query	None.
Request	<i>XML_BurningPrevention</i>
Response	<i>XML_ResponseStatus</i>

Remarks

The <ID> in the URL is the channel ID.

B.18 /ISAPI/Thermal/channels/<ID>/burningPrevention/capabilities

Get burning prevention capabilities.

Request URL Definition**Table B-29 GET /ISAPI/Thermal/channels/<ID>/burningPrevention/capabilities**

Method	GET
Description	Get burning prevention capabilities.
Query	None.
Request	None.
Response	<i>XML_BurningPreventionCap</i>

B.19 /ISAPI/Thermal/channels/<ID>/faceThermometry

Operations about temperature screening configuration.

Request URI Definition**Table B-30 GET /ISAPI/Thermal/channels/<ID>/faceThermometry**

Method	GET
Description	Get the configuration parameters of temperature screening of a specific camera.

Query	None.
Request	None.
Response	<i>XML_FaceThermometry</i>

Table B-31 PUT /ISAPI/Thermal/channels/<ID>/faceThermometry

Method	PUT
Description	Set the parameters of temperature screening of a specific camera.
Query	None.
Request	<i>XML_FaceThermometry</i>
Response	<i>XML_ResponseStatus</i>

Remarks

The <ID> in the URI refers to channel ID.

B.20 /ISAPI/Thermal/channels/<ID>/faceThermometry/capabilities

Get the temperature screening capability by channel.

Request URI Definition**Table B-32 GET /ISAPI/Thermal/channels/<ID>/faceThermometry/capabilities**

Method	GET
Description	Get the temperature screening capability by channel.
Query	None.
Request	None.
Response	<i>XML_Cap_FaceThermometry</i>

Remarks

The <ID> in the URI refers to channel ID.

B.21 /ISAPI/Thermal/channels/<ID>/faceThermometry/regions

Operations about the temperature screening rule configurations of all detection regions.

Request URI Definition**Table B-33 GET /ISAPI/Thermal/channels/<ID>/faceThermometry/regions**

Method	GET
Description	Get the temperature screening rule parameters of all detection regions.
Query	None.
Request	None.
Response	<i>XML_FaceThermometryRegionList</i>

Table B-34 PUT /ISAPI/Thermal/channels/<ID>/faceThermometry/regions

Method	PUT
Description	Set the temperature screening rules for all detection regions.
Query	None.
Request	<i>XML_FaceThermometryRegionList</i>
Response	<i>XML_ResponseStatus</i>

Remarks

The <ID> in the URL refers to the channel ID.

B.22 /ISAPI/Thermal/channels/<ID>/faceThermometry/regions/<ID>

Operations about the temperature screening rule configurations of a specific detection region.

Request URI Definition**Table B-35 GET /ISAPI/Thermal/channels/<ID>/faceThermometry/regions/<ID>**

Method	GET
Description	Get the temperature screening rule parameters of a specific detection region.
Query	None.
Request	None.
Response	<i>XML_ThermometryRegion</i>

Table B-36 PUT /ISAPI/Thermal/channels/<ID>/faceThermometry/regions/<ID>

Method	PUT
Description	Set the temperature screening rules of a specific detection region.
Query	None.
Request	<i>XML_ThermometryRegion</i>
Response	<i>XML_ResponseStatus</i>

Remarks

The first <ID> in the URI is the channel ID, and the second <ID> is the detection region ID.

B.23 /ISAPI/Thermal/channels/<ID>/faceThermometry/regions/<ID>/detectionInfo

Get the temperature screening results of a specific detection region.

Request URI Definition

Table B-37 GET /ISAPI/Thermal/channels/<ID>/faceThermometry/regions/<ID>/detectionInfo

Method	GET
Description	Get the temperature screening results of a specific detection region.
Query	None.
Request	None.
Response	<i>XML_FaceThermDetectionInfo</i>

Remarks

The first <ID> in the URI is channel ID, and second <ID> is the detection region ID.

B.24 /ISAPI/Thermal/channels/<ID>/fireDetection

Operations about the parameter information of fire and smoke detection.

Request URL Definition**Table B-38 GET /ISAPI/Thermal/channels/<ID>/fireDetection**

Method	GET
Description	Get the parameter information of fire and smoke detection.
Query	None.
Request	None.
Response	<i>XML_FireDetection</i>

Table B-39 PUT /ISAPI/Thermal/channels/<ID>/fireDetection

Method	PUT
Description	Set the parameter information of fire and smoke detection.
Query	None.
Request	<i>XML_FireDetection</i>
Response	<i>XML_ResponseStatus</i>

B.25 /ISAPI/Thermal/channels/<ID>/fireDetection/advanceParam/capabilities?format=json

Get the capability of configuring advanced parameters of fire detection.

Request URI Definition**Table B-40 GET /ISAPI/Thermal/channels/<ID>/fireDetection/advanceParam/capabilities?format=json**

Method	GET
Description	Get the capability of configuring advanced parameters of fire detection.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <i>JSON_fireDetection_AdvanceParamCap</i> Failed: <i>JSON_ResponseStatus</i>

B.26 /ISAPI/Thermal/channels/<ID>/fireDetection/advanceParam?format=json

Get and set the advanced parameters of fire detection.

Request URI Definition

Table B-41 GET /ISAPI/Thermal/channels/<ID>/fireDetection/advanceParam?format=json

Method	GET
Description	Get the advanced parameters of fire detection.
Query	format : determine the format of request or response message.
Request	None.
Response	Succeeded: <i>JSON_fireDetection_AdvanceParam</i> Failed: <i>JSON_ResponseStatus</i>

Table B-42 PUT /ISAPI/Thermal/channels/<ID>/fireDetection/advanceParam?format=json

Method	PUT
Description	Set the advanced parameters of fire detection.
Query	format : determine the format of request or response message.
Request	<i>JSON_fireDetection_AdvanceParam</i>
Response	<i>JSON_ResponseStatus</i>

B.27 /ISAPI/Thermal/channels/<ID>/fireDetection/capabilities

Get the configuration capability of fire and smoke detection.

Request URL Definition

Table B-43 GET /ISAPI/Thermal/channels/<ID>/fireDetection/capabilities

Method	GET
Description	Get the configuration capability of fire and smoke detection.
Query	None.
Request	None.
Response	<i>XML_Cap_FireDetection</i>

B.28 /ISAPI/Thermal/channels/<ID>/historyTemperature

Get history temperature.

Request URL Definition

Table B-44 POST /ISAPI/Thermal/channels/<ID>/historyTemperature

Method	POST
Description	Get history temperature.
Query	None.
Request	<i>XML_HistoryTemperatureDescription</i>
Response	<i>XML_HistoryTemperatureResult</i>

B.29 /ISAPI/Thermal/channels/<ID>/historyTemperature/capabilities

Get the history temperature capabilities.

Request URL Definition

Table B-45 GET /ISAPI/Thermal/channels/<ID>/historyTemperature/capabilities

Method	GET
Description	Get the history temperature capabilities.
Query	None.
Request	None.
Response	<i>XML_HistoryTemperatureCap</i>

B.30 /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/capabilities?format=json

Get the scene parameters configuration capabilities of ship detection.

Request URL Definition

Table B-46 GET /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/capabilities?format=json

Method	GET
Description	Get the scene parameters configuration capabilities of ship detection.
Query	format : determine the format of request or response message.
Request	None.
Response	Succeeded: <i>JSON_ShipsDetectionSceneCap</i> Failed: <i>JSON_ResponseStatus</i>

Remarks

- The <ID> in the URL refers to the channel No.
- The <SID> in the URL refers to the scene ID.

B.31 /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/captureRatio/capabilities?format=json

Get the capture ratio configuration capabilities.

Request URL Definition

Table B-47 GET /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/captureRatio/capabilities?format=json

Method	GET
Description	Get the capture ratio configuration capabilities.
Query	format : determine the format of request or response message.
Request	None.
Response	Succeeded: <i>JSON_ShipsDetectionCaptureRatioCap</i> Failed: <i>JSON_ResponseStatus</i>

Remarks

- The <ID> in the URL refers to the channel No.
- The <SID> in the URL refers to the scene ID.

B.32 /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/captureRatio?format=json

Get or set the capture ratio.

Request URL Definition

Table B-48 GET /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/captureRatio?format=json

Method	GET
Description	Get the capture ratio.
Query	format : determine the format of request or response message.
Request	None.
Response	Succeeded: <i>JSON_ShipsDetectionCaptureRatio</i> Failed: <i>JSON_ResponseStatus</i>

Table B-49 PUT /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/captureRatio?format=json

Method	PUT
Description	Set the capture ratio.
Query	format : determine the format of request or response message.
Request	<i>JSON_ShipsDetectionCaptureRatio</i>
Response	<i>JSON_ResponseStatus</i>

Remarks

- The <ID> in the URL refers to the channel No.
- The <SID> in the URL refers to the scene ID.

B.33 /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/dredgerDetection/regions/<ID>?format=json

Get or set a specified rule of dredger detection.

Request URL Definition

Table B-50 GET /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/dredgerDetection/regions/<ID>?format=json

Method	GET
Description	Get a specified rule of dredger detection.

Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <i>JSON_DredgerDetectionRule</i> Failed: <i>JSON_ResponseStatus</i>

Table B-51 PUT /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/dredgerDetection/regions/<ID>?format=json

Method	GET
Description	Get a specified rule of dredger detection.
Query	format: determine the format of request or response message.
Request	None.
Response	<i>JSON_ResponseStatus</i>

Remarks

- The first <ID> in the URL refers to the channel No., the second <ID> in the URL refers to the rule ID
- The <SID> in the URL refers to the scene ID.

B.34 /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/dredgerDetection/regions/capabilities?format=json

Get the configuration capabilities of dredger detection rules.

Request URL Definition

Table B-52 GET /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/dredgerDetection/regions/capabilities?format=json

Method	GET
Description	Get the configuration capabilities of dredger detection rules.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <i>JSON_DredgerDetectionRuleListCap</i> Failed: <i>JSON_ResponseStatus</i>

Remarks

- The <ID> in the URL refers to the channel No.
- The <SID> in the URL refers to the scene ID.

B.35 /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/dredgerDetection/regions?format=json

Get or set all rules of dredger detection.

Request URL Definition

Table B-53 GET /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/dredgerDetection/regions?format=json

Method	GET
Description	Get all rules of dredger detection.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <i>JSON_DredgerDetectionRuleList</i> Failed: <i>JSON_ResponseStatus</i>

Table B-54 PUT /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/dredgerDetection/regions?format=json

Method	PUT
Description	Set all rules of dredger detection.
Query	format: determine the format of request or response message.
Request	<i>JSON_DredgerDetectionRuleList</i>
Response	<i>JSON_ResponseStatus</i>

Remarks

- The <ID> in the URL refers to the channel No.
- The <SID> in the URL refers to the scene ID.

B.36 /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/goto?format=json

Switch scene for ship detection.

Request URL Definition**Table B-55 PUT /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/goto?format=json**

Method	PUT
Description	Switch scene for ship detection.
Query	format : determine the format of request or response message.
Request	None.
Response	<i>JSON_ResponseStatus</i>

Remarks

- The <ID> in the URL refers to the channel No.
- The <SID> in the URL refers to the scene ID.

B.37 /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/resetCount?format=json

Clear the ship statistics by scene.

Request URL Definition**Table B-56 PUT /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/resetCount?format=json**

Method	PUT
Description	Clear the ship statistics by scene.
Query	format : determine the format of request or response message.
Request	None.
Response	<i>JSON_ResponseStatus</i>

Remarks

- The <ID> in the URL refers to the channel No.
- The <SID> in the URL refers to the scene ID.

B.38 /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/shipFlowDetection/regions/<ID>?format=json

Get or set a specified rule of ship flow detection.

Request URL Definition**Table B-57 GET /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/shipFlowDetection/
regions/<ID>?format=json**

Method	GET
Description	Get a specified rule of ship flow detection.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <i>JSON_ShipFlowDetectionRule</i> Failed: <i>JSON_ResponseStatus</i>

**Table B-58 PUT /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/shipFlowDetection/
regions/<ID>?format=json**

Method	PUT
Description	Set a specified rule of ship flow detection.
Query	format: determine the format of request or response message.
Request	<i>JSON_ShipFlowDetectionRule</i>
Response	<i>JSON_ResponseStatus</i>

Remarks

- The first <ID> in the URL refers to the channel No., the second <ID> in the URL refers to the rule ID
- The <SID> in the URL refers to the scene ID.

**B.39 /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/
shipFlowDetection/regions/capabilities?format=json**

Get the configuration capabilities of ship flow detection rules.

Request URL Definition**Table B-59 GET /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/shipFlowDetection/
regions/capabilities?format=json**

Method	GET
Description	Get the configuration capabilities of ship flow detection rules.
Query	format: determine the format of request or response message.

Request	None.
Response	Succeeded: <i>JSON_ShipFlowDetectionRuleListCap</i> Failed: <i>JSON_ResponseStatus</i>

Remarks

- The <ID> in the URL refers to the channel No.
- The <SID> in the URL refers to the scene ID.

B.40 /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/shipFlowDetection/regions?format=json

Get or set all rules of ship flow detection.

Request URL Definition

Table B-60 GET /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/shipFlowDetection/regions?format=json

Method	GET
Description	Get all rules of ship flow detection.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <i>JSON_ShipFlowDetectionRuleList</i> Failed: <i>JSON_ResponseStatus</i>

Table B-61 PUT /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/shipFlowDetection/regions?format=json

Method	PUT
Description	Set all rules of ship flow detection.
Query	format: determine the format of request or response message.
Request	<i>JSON_ShipFlowDetectionRuleList</i>
Response	<i>JSON_ResponseStatus</i>

Remarks

- The <ID> in the URL refers to the channel No.
- The <SID> in the URL refers to the scene ID.

B.41 /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/shipsDetectionCount?format=json

Get the ship statistics by scene.

Request URL Definition

Table B-62 GET /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/shipsDetectionCount?format=json

Method	GET
Description	Get the ship statistics by scene.
Query	format : determine the format of request or response message.
Request	None.
Response	Succeeded: <i>JSON_ShipDetectionCount</i> Failed: <i>JSON_ResponseStatus</i>

Remarks

- The <ID> in the URL refers to the channel No.
- The <SID> in the URL refers to the scene ID.

B.42 /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/shipsFunctionMode/capabilities?format=json

Get the configuration capabilities of scene function mode.

Request URL Definition

Table B-63 GET /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/shipsFunctionMode/capabilities?format=json

Method	GET
Description	Get the configuration capabilities of scene function mode.
Query	format : determine the format of request or response message.
Request	None.
Response	Succeeded: <i>JSON_Cap_functionMode</i> Failed: <i>JSON_ResponseStatus</i>

Remarks

- The <ID> in the URL refers to the channel No.
- The <SID> in the URL refers to the scene ID.

B.43 /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/shipsFunctionMode?format=json

Get or set the scene function mode of ship detection.

Request URL Definition

Table B-64 GET /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/shipsFunctionMode?format=json

Method	GET
Description	Get the scene function mode of ship detection.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <i>JSON_functionMode</i> Failed: <i>JSON_ResponseStatus</i>

Table B-65 PUT /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>/shipsFunctionMode?format=json

Method	PUT
Description	Set the scene function mode of ship detection.
Query	format: determine the format of request or response message.
Request	None.
Response	<i>JSON_functionMode</i>

Remarks

- The <ID> in the URL refers to the channel No.
- The <SID> in the URL refers to the scene ID.

B.44 /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>?format=json

Operations about specified ship detection scene.

Request URL Definition

Table B-66 GET /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>?format=json

Method	GET
Description	Get the parameters of specified ship detection scene.
Query	format : determine the format of request or response message.
Request	None.
Response	Succeeded: <i>JSON_ShipsDetectionScene</i> Failed: <i>JSON_ResponseStatus</i>

Table B-67 PUT /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>?format=json

Method	PUT
Description	Add a new scene, or set specified scene parameters.
Query	format : determine the format of request or response message.
Request	<i>JSON_ShipsDetectionScene</i>
Response	<i>JSON_ResponseStatus</i>

Table B-68 DELETE /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>?format=json

Method	DELETE
Description	Delete the specified scene.
Query	format : determine the format of request or response message.
Request	<i>JSON_ShipsDetectionScene</i>
Response	<i>JSON_ResponseStatus</i>

Remarks

- The <ID> in the URL refers to the channel No.
- The <SID> in the URL refers to the scene ID.
- You can add a new scene via URL: PUT /ISAPI/Thermal/channels/<ID>/shipsDetection/<SID>?format=json. When adding a new scene, you should input only the node **sid** (scene ID) and set other required nodes to NULL in the message *JSON_ShipsDetectionScene* . The optional nodes can not be applied.

B.45 /ISAPI/Thermal/channels/<ID>/shipsDetection/advanceParam/capabilities?format=json

Get the capabilities of advanced parameters configuration for ship detection.

Request URL Definition**Table B-69 GET /ISAPI/Thermal/channels/<ID>/shipsDetection/advanceParam/capabilities?format=json**

Method	GET
Description	Get the capabilities of advanced parameters configuration for ship detection.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <i>JSON_advanceParamCap</i> Failed: <i>JSON_ResponseStatus</i>

Remarks

The <ID> in the URL refers to the channel No.

B.46 /ISAPI/Thermal/channels/<ID>/shipsDetection/advanceParam?format=json

Get or set the advanced parameters of ship detection.

Request URL Definition**Table B-70 GET /ISAPI/Thermal/channels/<ID>/shipsDetection/advanceParam?format=json**

Method	GET
Description	Get the advanced parameters of ship detection.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <i>JSON_advanceParam</i> Failed: <i>JSON_ResponseStatus</i>

Table B-71 PUT /ISAPI/Thermal/channels/<ID>/shipsDetection/advanceParam?format=json

Method	PUT
Description	Set the advanced parameters of ship detection.
Query	format: determine the format of request or response message.
Request	<i>JSON_advanceParam</i>
Response	<i>JSON_ResponseStatus</i>

Remarks

The <ID> in the URL refers to the channel No.

B.47 /ISAPI/Thermal/channels/<ID>/shipsDetection/basicParam/capabilities?format=json

Get the basic parameters configuration capability of ship detection.

Request URL Definition

Table B-72 GET /ISAPI/Thermal/channels/<ID>/shipsDetection/basicParam/capabilities?format=json

Method	GET
Description	Get the basic parameters configuration capability of ship detection.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <i>JSON_basicParamCap</i> Failed: <i>JSON_ResponseStatus</i>

Remarks

The <ID> in the URL refers to the channel No.

B.48 /ISAPI/Thermal/channels/<ID>/shipsDetection/basicParam?format=json

Get or set the ship detection basic parameters.

Request URL Definition

Table B-73 GET /ISAPI/Thermal/channels/<ID>/shipsDetection/basicParam?format=json

Method	GET
Description	Get the ship detection basic parameters.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <i>JSON_basicParam</i> Failed: <i>JSON_ResponseStatus</i>

Table B-74 PUT /ISAPI/Thermal/channels/<ID>/shipsDetection/basicParam?format=json

Method	PUT
Description	Set the ship detection basic parameters.
Query	format : determine the format of request or response message.
Request	<i>JSON_basicParam</i>
Response	<i>JSON_ResponseStatus</i>

Remarks

The <ID> in the URL refers to the channel No.

B.49 /ISAPI/Thermal/channels/<ID>/shipsDetection/capabilities?format=json

Get the ship detection capabilities.

Request URL Definition

Table B-75 GET /ISAPI/Thermal/channels/<ID>/shipsDetection/capabilities?format=json

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

Remarks

The <ID> in the URL refers to the channel No.

B.50 /ISAPI/Thermal/channels/<ID>/shipsDetection/sceneTrace/capabilities?format=json

Get the scene auto-switch capabilities.

Request URL Definition**Table B-76 GET /ISAPI/Thermal/channels/<ID>/shipsDetection/sceneTrace/capabilities?format=json**

Method	GET
Description	Get the scene auto-switch capabilities.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <i>JSON_ShipsDetectionTraceListCap</i> Failed: <i>JSON_ResponseStatus</i>

Remarks

The <ID> in the URL refers to the channel No.

B.51 /ISAPI/Thermal/channels/<ID>/shipsDetection/sceneTrace?format=json

Get all scenes information or start the scene auto-switch.

Request URL Definition**Table B-77 GET /ISAPI/Thermal/channels/<ID>/shipsDetection/sceneTrace?format=json**

Method	GET
Description	Get all scenes information.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: Failed: <i>JSON_ResponseStatus</i>

Table B-78 PUT /ISAPI/Thermal/channels/<ID>/shipsDetection/sceneTrace?format=json

Method	PUT
Description	Start the scene auto-switch.
Query	format: determine the format of request or response message.
Request	
Response	<i>JSON_ResponseStatus</i>

Remarks

The <ID> in the URL refers to the channel No.

B.52 /ISAPI/Thermal/channels/<ID>/shipsDetection?format=json

Operations about all ship detection scenes.

Request URL Definition**Table B-79 GET /ISAPI/Thermal/channels/<ID>/shipsDetection?format=json**

Method	GET
Description	Get parameters of all ship detection scenes.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <i>JSON_ShipsDetectionSceneList</i> Failed: <i>JSON_ResponseStatus</i>

Table B-80 PUT /ISAPI/Thermal/channels/<ID>/shipsDetection?format=json

Method	PUT
Description	Add new scenes, or set the parameters of all scenes.
Query	format: determine the format of request or response message.
Request	<i>JSON_ShipsDetectionSceneList</i>
Response	<i>JSON_ResponseStatus</i>

Table B-81 DELETE /ISAPI/Thermal/channels/<ID>/shipsDetection?format=json

Method	DELETE
Description	Delete all scenes.
Query	format: determine the format of request or response message.
Request	<i>JSON_ShipsDetectionSceneList</i>
Response	<i>JSON_ResponseStatus</i>

Remarks

The <ID> in the URL refers to the channel No.

B.53 /ISAPI/Thermal/channels/<ID>/thermometry/<SID>/capabilities

Get the preset (scene) configuration capability of temperature measurement.

Request URI Definition

Table B-82 GET /ISAPI/Thermal/channels/<ID>/thermometry/<SID>/capabilities

Method	GET
Description	Get the preset (scene) configuration capability of temperature measurement.
Query	None.
Request	None.
Response	<i>XML_Cap_ThermometryScene</i>

Remarks

The <ID> in the URI refers to the video input channel ID, and the <SID> in the URI is the scene ID (or preset No.).

B.54 /ISAPI/Thermal/channels/<ID>/thermometry/<SID>/alarmRules/temperatureIntervalMeasurement/capabilities?format=json

Get the configuration capability of interval temperature measurement alarm.

Request URI Definition

Table B-83 GET /ISAPI/Thermal/channels/<ID>/thermometry/<SID>/alarmRules/temperatureIntervalMeasurement/capabilities?format=json

Method	GET
Description	Get the configuration capability of interval temperature measurement alarm.
Query	format : determine the format of request or response message.
Request	None.
Response	Succeeded: <i>JSON_TemperatureIntervalMeasurementAlarmRuleCap</i> Failed: <i>JSON_ResponseStatus</i>

Remarks

- The <ID> in the request URI refers to channel ID.
- The <SID> in the request URI refers to scene ID.

B.55 /ISAPI/Thermal/channels/<ID>/thermometry/<SID>/alarmRules/temperatureIntervalMeasurement?format=json

Get or set the configuration parameters of interval temperature measurement alarm.

Request URI Definition

Table B-84 GET /ISAPI/Thermal/channels/<ID>/thermometry/<SID>/alarmRules/temperatureIntervalMeasurement?format=json

Method	GET
Description	Get the configuration parameters of interval temperature measurement alarm.
Query	format : determine the format of request or response message.
Request	None.
Response	Succeeded: <i>JSON_TemperatureIntervalMeasurementAlarmRule</i> Failed: <i>JSON_ResponseStatus</i>

Table B-85 PUT /ISAPI/Thermal/channels/<ID>/thermometry/<SID>/alarmRules/temperatureIntervalMeasurement?format=json

Method	PUT
Description	Set the parameters of interval temperature measurement alarm.
Query	format : determine the format of request or response message.
Request	<i>JSON_TemperatureIntervalMeasurementAlarmRule</i>
Response	<i>JSON_ResponseStatus</i>

Remarks

- The <ID> in the request URI refers to channel ID.
- The <SID> in the request URI refers to scene ID.

B.56 /ISAPI/Thermal/channels/<ID>/thermometry/basicParam

Operations about basic configuration of temperature measurement.

Request URL Definition**Table B-86 GET /ISAPI/Thermal/channels/<ID>/thermometry/basicParam**

Method	GET
Description	Get the basic parameters of temperature measurement.
Query	None.
Request	None.
Response	<i>XML_ThermometryBasicParam</i>

Table B-87 PUT /ISAPI/Thermal/channels/<ID>/thermometry/basicParam

Method	PUT
Description	Set the basic parameters of temperature measurement.
Query	None.
Request	<i>XML_ThermometryBasicParam</i>
Response	<i>XML_ResponseStatus</i>

Remarks

The <ID> in the URL is the channel ID.

B.57 /ISAPI/Thermal/channels/<ID>/thermometry/basicParam/capabilities

Get the capability of basic configuration of temperature measurement.

Request URL Definition**Table B-88 GET /ISAPI/Thermal/channels/<ID>/thermometry/basicParam/capabilities**

Method	GET
Description	Get the capability of basic configuration of temperature measurement.
Query	None.
Request	None.
Response	<i>XML_Cap_ThermometryBasicParam</i>

Remarks

The <ID> in the URL refers to the channel ID.

B.58 /ISAPI/Thermal/channels/<ID>/thermometry/pixelToPixelParam

Get or set the pixel-to-pixel temperature measurement parameters.

Request URI Definition

Table B-89 GET /ISAPI/Thermal/channels/<ID>/thermometry/pixelToPixelParam

Method	GET
Description	Get the pixel-to-pixel temperature measurement parameters.
Query	None.
Request	None.
Response	<i>XML_PixelToPixelParam</i>

Table B-90 PUT /ISAPI/Thermal/channels/<ID>/thermometry/pixelToPixelParam

Method	PUT
Description	Set the pixel-to-pixel temperature measurement parameters.
Query	None.
Request	<i>XML_PixelToPixelParam</i>
Response	<i>XML_ResponseStatus</i>

Remarks

The <ID> in the URI refers to channel ID.

B.59 /ISAPI/Thermal/channels/<ID>/thermometry/pixelToPixelParam/capabilities

Get the configuration capability of pixel-to-pixel temperature measurement parameters.

Request URI Definition

Table B-91 GET /ISAPI/Thermal/channels/<ID>/thermometry/pixelToPixelParam/capabilities

Method	GET
Description	Get the configuration capability of pixel-to-pixel temperature measurement parameters.
Query	None.

Request	None.
Response	<i>XML_PixelToPixelParamCap</i>

Remarks

The <ID> in the URL is the channel ID.

B.60 /ISAPI/Thermal/channels/<ID>/thermometry/OffLineCapture?format=json

Device captures one picture offline.

Request URL Definition

Table B-92 GET /ISAPI/Thermal/channels/<ID>/thermometry/OffLineCapture?format=json

Method	GET
Description	Device captures one picture offline.
Query	None.
Request	None.
Response	Succeeded: <i>JSON_thermometryOffLineCapture</i> Failed: <i>JSON_ResponseStatus</i>

Remarks

The <ID> in the URL refers to the channel ID.

B.61 /ISAPI/Thermal/channels/<ID>/temperatureCorrect/capabilities?format=json

Get the temperature calibration capability.

Request URI Definition

Table B-93 GET /ISAPI/Thermal/channels/<ID>/temperatureCorrect/capabilities?format=json

Method	GET
Description	Get the temperature calibration capability.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <i>JSON_TemperatureCorrectCap</i>

	Failed: <i>JSON_ResponseStatus</i>
--	------------------------------------

Remarks

The <ID> in the URI refers to channel No.

B.62 /ISAPI/Thermal/channels/<ID>/temperatureCorrect?format=json

Get or set the temperature calibration configuration parameters

Request URI Definition**Table B-94 GET /ISAPI/Thermal/channels/<ID>/temperatureCorrect?format=json**

Method	GET
Description	Get the temperature calibration configuration parameters.
Query	format: determine the format of request or response message.
Request	None.
Response	Succeeded: <i>JSON_TemperatureCorrect</i> Failed: <i>JSON_ResponseStatus</i>

Table B-95 POST /ISAPI/Thermal/channels/<ID>/temperatureCorrect?format=json

Method	POST
Description	Set the temperature calibration parameters.
Query	format: determine the format of request or response message.
Request	<i>JSON_TemperatureCorrect</i>
Response	<i>JSON_ResponseStatus</i>

Remarks

The <ID> in the URI refers to channel No.

B.63 /ISAPI/Thermal/Power

Get the camera battery parameters.

Request URL Definition

Table B-96 GET /ISAPI/Thermal/Power

Method	GET
Description	Get the camera battery parameters.
Query	None.
Request	None.
Response	Succeeded: <i>XML_Power</i> Failed: <i>XML_ResponseStatus</i>

B.64 /ISAPI/Thermal/Power/capabilities

Get the configuration capability of camera battery.

Request URL Definition

Table B-97 GET /ISAPI/Thermal/Power/capabilities

Method	GET
Description	Get the configuration capability of camera battery.
Query	None.
Request	None.
Response	Succeeded: <i>XML_Cap_Power</i> Failed: <i>XML_ResponseStatus</i>

Appendix C. Appendixes

C.1 Request and Response Messages

C.1.1 JSON_advanceParam

advanceParam message in JSON format

```
{
  "advanceParam":{
    "targetDetectionThreshold":,
    /*optional, integer type, target detection sensitivity threshold, ranges from 0 to 4, the default value is 3*/
    "targetGenerationSpeed":
    /*optional, integer type, target generation speed, ranges from 0 to 4, the default value is 3*/
  }
}
```

C.1.2 JSON_advanceParamCap

advanceParamCap message in JSON format

```
{
  "advanceParamCap":{
    "targetDetectionThreshold":{
    /*optional, integer type, target detection sensitivity threshold, ranges from 0 to 4, the default value is 3*/
      "@min":0,
      "@max":4,
      "def":3
    },
    "targetGenerationSpeed":{
    /*optional, integer type, target generation speed, ranges from 0 to 4, the default value is 3*/
      "@min":0,
      "@max":4,
      "def":3
    }
  }
}
```

C.1.3 JSON_basicParam

basicParam message in JSON format

```
{
  "basicParam":{
```

```
"enabled":,  
/*required, boolean type, enable ship detection*/  
"horizontalHeight":,  
/*required, float type, device horizontal height, corrects to one decimal place*/  
"displayShipInfoOnStreamEnabled":,  
/*optional, boolean type, enable displaying ship information on stream*/  
"backToSceneTime":  
/*optional, integer type, back to scene time, ranges from 0 to 100, the default value is 15, unit: second*/  
}  
}
```

C.1.4 JSON_basicParamCap

basicParamCap message in JSON format

```
{  
  "basicParamCap":{  
    "enabled":{  
      /*required, boolean type, enable ship detection*/  
      "@opt":"true,false"  
    },  
    "horizontalHeight":{  
      /*required, float type, device horizontal height, corrects to one decimal place*/  
      "@min":0.0,  
      "@max":100.0,  
      "def":0.0  
    },  
    "displayShipInfoOnStreamEnabled":{  
      /*optional, boolean type, enable displaying ship information on stream*/  
      "@opt":"true,false"  
    },  
    "backToSceneTime":{  
      /*optional, integer type, back to scene time, ranges from 0 to 100, the default value is 15, unit: second*/  
      "@min":0,  
      "@max":100,  
      "def":15  
    }  
  }  
}
```

C.1.5 JSON_Cap_functionMode

functionMode capability message in JSON format

```
{  
  "functionMode":{  
    /*required, string type, scene function mode of ship detection: shipFlowDetection-ship flow detection mode,  
    dredgerDetection-dredger detection mode*/  
    "@opt":"shipFlowDetection,dredgerDetection"  
  }  
}
```

```
}  
}
```

C.1.6 JSON_DredgerDetectionRule

DredgerDetectionRule message in JSON format

```
{  
  "DredgerDetectionRule":{  
    "id":,  
    /*required, integer type, rule ID*/  
    "enabled":,  
    /*required, boolean type, enable rule*/  
    "overStayingAlarmTime":,  
    /*optional, integer type, overstay alarm time, ranges from 60 to 3600, the default value is 180, unit: second*/  
    "RegionCoordinatesList":[  
      {  
        /*optional, list of detection region coordinates*/  
        "positionX":,  
        /*required, integer type, X-coordinate*/  
        "positionY":  
        /*required, integer type, Y-coordinate*/  
      }  
    ],  
    "TriggerLineList":[  
      {  
        "id":,  
        /*required, integer type, trigger line ID*/  
        "enabled":,  
        /*required, boolean type, enable trigger line*/  
        "MinimumTargetSize":{  
          /*required, the minimum target size*/  
          "width":,  
          /*required, integer type, width, ranges from 0 to 100, the default value is 52*/  
          "height":35  
        },  
        /*required, integer type, height, ranges from 0 to 100, the default value is 35*/  
      },  
    ]  
  }  
}
```

C.1.7 JSON_DredgerDetectionRuleList

DredgerDetectionRuleList message in JSON format

```
{  
  "DredgerDetectionRuleList":[  
    {  
      /*required, integer type, rule ID*/  
      "id":,  
      /*required, boolean type, enable rule*/  
      "enabled":,  
      /*required, integer type, overstay alarm time, ranges from 60 to 3600, the default value is 180, unit: second*/  
      "overStayingAlarmTime":,  
      /*optional, list of detection region coordinates*/  
      "RegionCoordinatesList":[  
        {  
          /*optional, list of detection region coordinates*/  
          "positionX":,  
          /*required, integer type, X-coordinate*/  
          "positionY":  
          /*required, integer type, Y-coordinate*/  
        }  
      ],  
      "TriggerLineList":[  
        {  
          "id":,  
          /*required, integer type, trigger line ID*/  
          "enabled":,  
          /*required, boolean type, enable trigger line*/  
          "MinimumTargetSize":{  
            /*required, the minimum target size*/  
            "width":,  
            /*required, integer type, width, ranges from 0 to 100, the default value is 52*/  
            "height":35  
          },  
          /*required, integer type, height, ranges from 0 to 100, the default value is 35*/  
        },  
      ]  
    },  
  ]  
}
```

```
"DredgerDetectionRule":{
  "id":,
  /*required, integer type, rule ID*/
  "enabled":,
  /*required, boolean type, enable rule*/
  "overStayingAlarmTime":,
  /*optional, integer type, overstay alarm time, ranges from 60 to 3600, the default value is 180, unit: second*/
  "RegionCoordinatesList":[
    {
  /*optional, list of detection region coordinates*/
    "positionX":,
    /*required, integer type, X-coordinate*/
    "positionY":
  /*required, integer type, Y-coordinate*/
    }
  ],
  "TriggerLineList":[
    {
      "id":,
      /*required, integer type, trigger line ID*/
      "enabled":,
      /*required, boolean type, enable trigger line*/
      "MinimumTargetSize":{
        /*required, the minimum target size*/
        "width":,
        /*required, integer type, width, ranges from 0 to 100, the default value is 52*/
        "height":
        /*required, integer type, height, ranges from 0 to 100, the default value is 35*/
      },
    }
  ]
}
}
```

C.1.8 JSON_DredgerDetectionRuleListCap

DredgerDetectionRuleListCap message in JSON format

```
{
  "DredgerDetectionRuleListCap":{
    "maxSize":5,
    /*required, integer type, the maximum number of supported dredger detection rules*/
    "DredgerDetectionRuleCap":{
      "enabled":{
        /*required, boolean type, enable rule*/
        "@opt":"true,false"
      },
      "overStayingAlarmTime":{
        /*optional, integer type, overstay alarm time, ranges from 60 to 3600, the default value is 180, unit: second*/

```



```

    "@min":60,
    "@max":3600,
    "@def":180
  },
  "RegionCoordinatesList":{
/*optional, list of detection region coordinates*/
    "maxSize":5,
/*required, integer type, the number of supported regions*/
    "positionX":{
/*required, integer type, X-coordinate*/
      "@min":0,
      "@max":1000
    },
    "positionY":{
/*required, integer type, Y-coordinate*/
      "@min":0,
      "@max":1000
    }
  },
  "TriggerLineList":{
    "maxSize":5,
/*required, integer type, the number of supported trigger lines*/
    "enabled":{
/*required, boolean type, enable trigger line*/
      "@opt":["true,false"]
    },
    "MinimumTargetSize":{
/*required, the minimum target size*/
      "width":{
/*required, integer type, width, ranges from 0 to 100, the default value is 52*/
        "@min":0,
        "@max":100,
        "@def":52
      },
      "height":{
/*required, integer type, height, ranges from 0 to 100, the default value is 35*/
        "@min":0,
        "@max":100,
        "@def":35
      }
    }
  }
}
}
}
}

```

C.1.9 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.1.10 JSON_EventNotificationAlert_dredgerDetection

The information of dredger detection alarm is uploaded in the JSON format, see details below.

```
{
  "ipAddress": "",
  /*required, alarm device IPV4 address, string type, the maximum length is 32*/
  "ipv6Address": "",
  /*optional, alarm device IPV6 address, string type, the maximum length is 128*/
  "portNo": "",
  /*optional, alarm device port No., integer32 type*/
  "protocol": "",
  /*optional, protocol type: HTTP or HTTPS, string type, the maximum length is 32*/
  "macAddress": "",
  /*optional, MAC address, string type, the maximum length is 32*/
  "channelID": ,
```

```
/*optional, channel ID number, which triggers the alarm, integer32 type*/
"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, alarm triggered time, ISO8601 time format, string type, the maximum length is 32*/
"activePostCount":,
/*required, the number of one alarm uploaded times, integer32 type*/
"eventType": "",
/*required, event type: dredgerDetection-dredger detection, string type, the maximum length is 128*/
"eventState": "",
/*required, event triggering status: active-triggered, inactive-not triggered (heartbeat data), string type, the maximum
length is 32*/
"eventDescription": "",
/*required, event description: dredger detection alarm)*/
"DetectionRegionList": [{
/*ship detection region list*/
  "DetectionRegionEntry": {
    "RegionCoordinatesList": [{
/*ship detection region*/
      "RegionCoordinates": {
        "positionX":,
/*required, float type, X-coordinate*/
        "positionY":
/*required, float type, Y-coordinate*/
      }
    }],
    "ShipsDetection": {
      "shipsInfoList": [{
        "shipsInfo": {
          "id":,
/*required, integer type, ship ID*/
          "shipsLength":,
/*required, float type, ship length*/
          "shipsHeight":,
/*required, float type, ship height*/
          "shipsWidth":
/*required, float type, ship width*/
        }
      }
    }
  }
}],
"sid":,
/*optional, integer type, scene ID*/
"sceneName": ""
/*optional, string type, scene name*/
}
```

C.1.11 JSON_EventNotificationAlert_temperatureIntervalMeasurementMsg

JSON message about details of interval temperature measurement alarm

```
Accept: text/html, application/xhtml+xml,
Accept-Language: en-US
Content-Type: multipart/form-data; boundary=MIME_boundary
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0)
Accept-Encoding: gzip, deflate
Host: 10.10.36.29:8080
Content-Length: 9907
//In listening mode, the uploaded message contains Content-Length, and its value is the length of message and
pictures; in arming mode, the uploaded message does not contain Content-Length//
Connection: Keep-Alive
Cache-Control: no-cache
//Contents above are HTTP header contents, and below are detailed alarm information//

--MIME_boundary
Content-Disposition: form-data; name="temperatureIntervalMeasurement"
Content-Type: application/json
Content-Length: 9907    //JSON message length//
{
  "ipAddress": "172.6.64.7",
  /*required, string, IPv4 address of alarm device; the maximum length is 32 bytes*/
  "ipv6Address": "",
  /*optional, string, IPv6 address of alarm device; the maximum length is 128 bytes*/
  "portNo": 80,
  /*optional, integer32, port No. of alarm device*/
  "protocol": "HTTP",
  /*optional, string, "HTTP"-for device network SDK, "HTTPS", "EHome"-for ISUP SDK; the maximum length is 32 bytes*/
  "macAddress": "01:17:24:45:D9:F4",
  /*optional, string, MAC address; the maximum length is 32 bytes*/
  "channelID": 1,
  /*optional, integer32, No. of alarm triggered channel*/
  "dateTime": "2004-05-03T17:30:08+08:00",
  /*required, string, alarm triggered time (ISO 8601 format); the maximum length is 32 bytes*/
  "activePostCount": 1,
  /*required, integer32, uploaded times of one alarm*/
  "eventType": "temperatureIntervalMeasurement",
  /*required, string, triggered event type, here it should be set to "temperatureIntervalMeasurement" (interval
  temperature measurement); the maximum length is 128 bytes*/
  "eventState": "active",
  /*required, string, event status: "active"-occured, "inactive"-unoccurred; the maximum length is 32 bytes*/
  "eventDescription": "Temperature Interval Measurement",
  /*required, string, event description, the maximum length is 128 bytes*/
  "deviceID": "test0123",
  /*optional, string, PUID; this node should be returned when transmitting ISAPI evnet informtion via ISUP SDK, and the
  value is same as deviceID of URI: /ISAPI/System/Network/Ehome*/
  "TemperatureIntervalMeasurement":{
    "mode": "normal",
    /*optional, string, mode: "normal", "expert"*/
```

```
"thermometryUnit": "celsius",
/*optional, string, temperature unit: "celsius, fahrenheit, kelvin"*/
"NormalMode":{
/*optional, temperature measurement information in normal mode*/
"alarmType": "",
/*optional, string, alarm type: "highestTemp"-the highest temperature, "lowestTemp"-the lowest temperature*/
"currTemperature": 35.0,
/*optional, float, current temperature, value range: [-20,550], unit: Celsius; the value should be accurate to one decimal place*/
"CurrTemperaturePoint":{
/*optional, current temperature point coordiantes*/
"positionX":0.123,
/*required, float, X-coordiante, the value is normalized, range: [0,1]; the value should be accurate to three decimal places; the reference origin is the upper left corner of image*/
"positionY":0.123
/*required, float, Y-coordiante, the value is normalized, range: [0,1]; the value should be accurate to three decimal places; the reference origin is the upper left corner of image*/
},
"TemperatureInterval":{
/*optional, temperature interval information*/
"id": 1,
/*optional, int, No. */
"name": "",
/*optional, string, interval name*/
"minTemperature": 35.0,
/*optional, float, the minimum temperature, range: [-20,550], unit: Celsius; the value should be accurate to one decimal place*/
"maxTemperature": 37.0
/*optional, float, the maximum temperature, range: [-20,550], unit: Celsius; the value should be accurate to one decimal place*/
}
},
"ExpertMode":{
/*optional, temperature measurement information in expert mode*/
"DetectionRegionList":[{
"DetectionRegionEntry":{
"id": 1,
/*optional, int, rule ID*/
"name": "",
/*optional, string, rule name*/
"type": "",
/*optional, string, rule type: "point", "line", "region"-area*/
"Point":{
/*optional, point coordiantes*/
"positionX":0.123,
/*required, float, X-coordiante, the value is normalized, range: [0,1]; the value should be accurate to three decimal places; the reference origin is the upper left corner of image*/
"positionY":0.123
/*required, float, Y-coordiante, the value is normalized, range: [0,1]; the value should be accurate to three decimal places; the reference origin is the upper left corner of image*/
},
"Line":{
```

```
/*optional, line coordinates*/
  "Point":{
/*optional, point coordiantes*/
  "positionX":0.123,
/*required, float, X-coordiante, the value is normalized, range: [0,1]; the value should be accurate to three decimal
places; the reference origin is the upper left corner of image*/
  "positionY":0.123
/*required, float, Y-coordiante, the value is normalized, range: [0,1]; the value should be accurate to three decimal
places; the reference origin is the upper left corner of image*/
  }
  },
  "Region":{{
/*optional, area coordinates*/
    "Point":{
/*optional, point coordinates*/
      "positionX":0.123,
/*required, float, X-coordiante, the value is normalized, range: [0,1]; the value should be accurate to three decimal
places; the reference origin is the upper left corner of image*/
      "positionY":0.123
/*required, float, Y-coordiante, the value is normalized, range: [0,1]; the value should be accurate to three decimal
places; the reference origin is the upper left corner of image*/
    }
  },
  "alarmType": "",
/*optional, string, alarm type: "highestTemp"-the highest temperature, "lowestTemp"-the lowest temperature*/
  "currTemperature": 35.0,
/*optional, float, current temperature, value range: [-20,550], unit: Celsius; the value should be accurate to one
decimal place*/
  "CurrTemperaturePoint":{
/*optional, current temperature point coordiantes*/
    "positionX":0.123,
/*required, float, X-coordiante, the value is normalized, range: [0,1]; the value should be accurate to three decimal
places; the reference origin is the upper left corner of image*/
    "positionY":0.123
/*required, float, Y-coordiante, the value is normalized, range: [0,1]; the value should be accurate to three decimal
places; the reference origin is the upper left corner of image*/
  },
  "TemperatureInterval":{
/*optional, temperature interval information*/
    "id": 1,
/*optional, int, No.*/
    "name": "",
/*optional, string, interval name*/
    "minTemperature": 35.0,
/*optional, float, the minimum temperature, range: [-20,550], unit: Celsius; the value should be accurate to one
decimal place*/
    "maxTemperature": 37.0
/*optional, float, the maximum temperature, range: [-20,550], unit: Celsius; the value should be accurate to one
decimal place*/
  }
}
}}
```

```
,
  "VisibleLightImage":{
/*optional, visible picture*/
    "resourcesContentType":"binary",
/*dependent, string, resource transmission type: "url", "binary"; when the returned alarm message consists additional
resource (e.g., picture), this node should be returned*/
    "resourcesContent": "",
/*dependent, string, resource ID; when the returned alarm message consists additional resource (e.g., picture), this
node should be returned; when the value of resourcesContentType is "binary", it must be the same as Content-ID of
picture, when the value of resourcesContentType is "url", its value is the picture URL */
  }
  "ThermalImage":{
/*optional, thermal picture*/
    "resourcesContentType":"binary",
/*dependent, string, resource transmission type: "url", "binary"; when the returned alarm message consists additional
resource (e.g., picture), this node should be returned*/
    "resourcesContent": "",
/*dependent, string, resource ID; when the returned alarm message consists additional resource (e.g., picture), this
node should be returned; when the value of resourcesContentType is "binary", it must be the same as Content-ID of
picture, when the value of resourcesContentType is "url", its value is the picture URL */
  }
}
}
```

C.1.12 JSON_fireDetection_AdvanceParam

JSON message about advanced parameters of fire detection

```
{
  "AdvanceParam": {
    "minPixel":
/*optional, int, the minimum pixel of fire detection, range: [1,2]*/
  }
}
```

C.1.13 JSON_fireDetection_AdvanceParamCap

JSON message about capability of configuring advanced parameters of fire detection.

```
{
  "AdvanceParamCap": {
    "minPixel": {
/*optional, int, the minimum pixel of fire detection, range: [1,2]*/
      "@min": 1,
      "@max": 2
    }
  }
}
```

C.1.14 JSON_functionMode

functionMode message in JSON format

```
{
  "functionMode":""
  /*required, string type, scene function mode of ship detection: shipFlowDetection-ship flow detection mode,
  dredgerDetection-dredger detection mode*/
}
```

C.1.15 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*/
}
```



Note

See ***Response Codes of Text Protocol*** for details about the status codes, sub status codes, error codes, and error descriptions.

C.1.16 JSON_ShipDetectionCount

ShipDetectionCount message in JSON format

```
{
  "ShipsDetectionCount":{
    "sid";
    /*required, integer type, scene ID*/
  }
```



```
"upShipsCount":,
/*required, integer type, number of upstream ships*/
"downShipsCount":,
/*required, integer type, number of downstream*/
"leftShipsCount":,
/*required, integer type, number of ships on the left*/
"rightShipsCount":,
/*required, integer type, number of ships on the right*/
"totalCount":,
/*required, integer type, total number of ships*/
"beginTime":""
/*required, string type, ISO8601 time format, the maximum length is 32 bits*/
}
}
```

C.1.17 JSON_ShipFlowDetectionRule

ShipFlowDetectionRule message in JSON format

```
{
  "ShipFlowDetectionRule":{
    "id":,
    /*required, integer type, rule ID*/
    "enabled":,
    /*required, boolean type, enable the rule*/
    "regionEnOrExAlarmEnabled":,
    /*optional, boolean type, enable alarm of region entrance or region exit*/
    "RegionCoordinatesList":[
      {
        /*optional, list of detection region coordinates*/
        "positionX":,
        /*required, integer type, X-coordiante*/
        "positionY":
        /*required, integer type, Y-coordinate*/
      }
    ],
    "TriggerLineList":[
      {
        "id":,
        /*required, integer type, trigger line ID*/
        "enabled":,
        /*required, boolean type, enable trigger line*/
        "MinimumTargetSize":{
          /*required, the minimum target size*/
          "width":,
          /*required, integer type, width, ranges from 0 to 100, the value is 52 by default*/
          "height":
          /*required, integer type, height, ranges from 0 to 100, the value is 35 by default*/
        },
        "TriggerLineCoordinatesList":[
          {
```

```
    "positionX":,  
    /*required, integer type, X-coordinate*/  
    "positionY":  
    /*required, integer type, Y-coordinate*/  
  }  
]  
}  
]  
}  
}
```

C.1.18 JSON_ShipFlowDetectionRuleList

ShipFlowDetectionRuleList message in JSON format

```
{  
  "ShipFlowDetectionRuleList": [  
    {  
      "ShipFlowDetectionRule": {  
        "id":,  
        /*required, integer type, rule ID*/  
        "enabled":,  
        /*required, boolean type, enable the rule*/  
        "regionEnOrExAlarmEnabled":,  
        /*optional, boolean type, enable alarm of region entrance or region exit*/  
        "RegionCoordinatesList": [  
          {  
            /*optional, list of detection region coordinates*/  
            "positionX":,  
            /*required, integer type, X-coordiante*/  
            "positionY":  
            /*required, integer type, Y-coordinate*/  
          }  
        ],  
        "TriggerLineList": [  
          {  
            "id":,  
            /*required, integer type, trigger line ID*/  
            "enabled":,  
            /*required, boolean type, enable trigger line*/  
            "MinimumTargetSize": {  
            /*required, the minimum target size*/  
              "width":,  
            /*required, integer type, width, ranges from 0 to 100, the value is 52 by default*/  
              "height":  
            /*required, integer type, height, ranges from 0 to 100, the value is 35 by default*/  
            },  
            "TriggerLineCoordinatesList": [  
              {  
                "positionX":,  
                /*required, integer type, X-coordinate*/
```

```
        "positionY":  
/*required, integer type, Y-coordinate*/  
        }  
    ]  
}  
]  
}  
}  
]  
}
```

C.1.19 JSON_ShipFlowDetectionRuleListCap

ShipFlowDetectionRuleListCap message in JSON format

```
{  
  "ShipFlowDetectionRuleListCap":{  
    "maxSize":5,  
/*required, integer type, the maximum number of supported ship flow detection rules*/  
    "ShipFlowDetectionRuleCap":{  
      "enabled":{  
/*required, boolean type, enable ship flow detection rule*/  
        "@opt":"true,false"  
      },  
      "regionEnOrExAlarmEnabled":{  
/*optional, boolean type, enable alarm of region entrance or region exit*/  
        "@opt":"true,false"  
      },  
      "RegionCoordinatesList":{  
/*optional, list of detection region coordinates*/  
        "maxSize":5,  
/*required, integer type, the maximum number of supported regions*/  
        "positionX":{  
/*required, integer type, X-coordinate*/  
          "@min":0,  
          "@max":1000  
        },  
        "positionY":{  
/*required, integer type, Y-coordinate*/  
          "@min":0,  
          "@max":1000  
        }  
      },  
      "TriggerLineList":{  
        "maxSize":5,  
/*required, integer type, the number of supported trigger lines*/  
        "enabled":{  
/*required, boolean type, enable trigger line*/  
          "@opt":"true,false"  
        },  
        "MinimumTargetSize":{
```

```
/*required, the minimum target size*/
    "width":{
/*required, integer type, width, ranges from 0 to 100, the value is 52 by default*/
        "@min":0,
        "@max":100,
        "@def":52
    },
    "height":{
/*required, integer type, height, ranges from 0 to 100, the value is 35 by default*/
        "@min":0,
        "@max":100,
        "@def":35
    },
    },
    "TriggerLineCoordinatesList":{
        "positionX":{
/*required, integer type, X-coordinate*/
            "@min":0,
            "@max":1000
        },
        "positionY":{
/*required, integer type, Y--coordinate*/
            "@min":0,
            "@max":1000
        }
    }
}
}
}
}
```

C.1.20 JSON_shipsDetectionCap

shipsDetectionCap message in JSON format

```
{
  "shipsDetectionCap":{
    "isSupportBasicConfig":,
/*required, boolean type, whether supports basic parameters configuration of ship detection*/
    "isSupportAdvanceConfig":,
/*optional, boolean type, whether supports advanced parameters configuration of ship detection*/
    "isSupportSceneConfig":,
/*optional, boolean type, whether supports scene parameters configuration of ship detection*/
    "isSupportCaptureRatio":,
/*optional, boolean type, whether supports capture ratio configuration of ship detection*/
    "isSupportFunctionMode":,
/*optional, boolean type, whether supports function mode configuration of ship detection*/
    "isSupportShipsFlowDetectionRuleConfig":,
/*optional, boolean type, whether supports rule configuration of ship flow detection mode*/
    "isSupportDredgerDetectionRuleConfig":,
/*optional, boolean type, whether supports rule configuration of dredger detection mode*/
  }
```

```
"isSupportShipsDetectionCount":,
/*optional, boolean type, whether supports ship counting*/
"isSupportSceneCruise":,
/*optional, boolean type, whether supports scene auto-switch*/
"sceneRatioId":{
/*optional, string type, the ID of channel, which supports scene ratio*/
"@opt":""
},
"shipsDetectionSceneNum":
/*optional, integer type, the maximum number of supported scenes for ship detection*/
}
}
```

C.1.21 JSON_ShipsDetectionCaptureRatio

ShipsDetectionCaptureRatio message in JSON format

```
{
  "ShipsDetectionCaptureRatio":{
    "centerCaptureEnabled":,
/*required, boolean type, enable center capture mode*/
    "captureRatio":
/*optional, integer type, capture ratio*/
  }
}
```

C.1.22 JSON_ShipsDetectionCaptureRatioCap

ShipsDetectionCaptureRatioCap message in JSON format

```
{
  "ShipsDetectionCaptureRatioCap":{
    "centerCaptureEnabled":"true,false",
/*required, boolean type, enable center capture mode*/
    "captureRatio":{
/*optional, integer type, capture ratio*/
      "@min":1,
      "@max":32
    }
  }
}
```

C.1.23 JSON_ShipsDetectionScene

ShipsDetectionScene message in JSON format

```
{
  "ShipsDetectionScene":{
    "sid";
    /*required, integer type, scene ID*/
    "sceneName":"","
    /*optional, string type, scene name*/
    "sceneRatio";
    /*optional, integer type, scene ratio (actual ratio of each channel). You can get the channel No. via URL: /ISAPI/
    Thermal/shipsDetection/capabilities?format=json*/
    "SceneNameOverlay":{
    /*optional, scene name on-screen display*/
    "enabled";
    /*required, enable scene name OSD*/
    "normalizedScreenSize":{
    /*normalized size*/
    "normalizedScreenWidth";
    /*optional, float type, normalized width*/
    "normalizedScreenHeight":
    /*optional, float type, normalized height*/
    },
    "positionX";
    /*optional, integer type, original point X-coordinate*/
    "positionY":
    /*optional, integer type, original point Y-coordinate*/
    }
  }
}
```

C.1.24 JSON_ShipsDetectionSceneCap

ShipsDetectionSceneCap message in JSON format

```
{
  "ShipsDetectionSceneCap":{
    "sid";
    /*required, integer type, scene ID*/
    "sceneName":"","
    /*optional, string type, scene name*/
    "sceneRatio":{
    /*optional, integer type, scene ratio (actual ratio of each channel). You can get the channel No. via URL: /ISAPI/
    Thermal/shipsDetection/capabilities?format=json*/
    "@min";
    "@max";
    "def";
    },
    "SceneNameOverlay":{
    /*optional, scene name on-screen display*/
    "enabled":{
    /*required, enable scene name OSD*/
    "@opt":"true,false"
```

```

    },
    "normalizedScreenSize":{
/*normalized size*/
    "normalizedScreenWidth":{
/*optional, float type, normalized width*/
        "@min":0,
        "@max":1,
    },
    "normalizedScreenHeight":{
/*optional, float type, normalized height*/
        "@min":0,
        "@max":1,
    }
    },
    "positionX":{
/*optional, integer type, original point X-coordinate*/
        "@min":,
        "@max":,
    },
    "positionY":{
/*optional, integer type, original point Y-coordinate*/
        "@min":,
        "@max":,
    }
    }
}

```

C.1.25 JSON_ShipsDetectionSceneList

ShipsDetectionSceneList message in JSON format

```

{
  "ShipsDetectionSceneList":[{
    "ShipsDetectionScene":{
      "sid",
/*required, integer type, scene ID*/
      "sceneName": "",
/*optional, string type, scene name*/
      "sceneRatio",
/*optional, integer type, scene ratio (actual ratio of each channel). You can get the channel No. via URL: /ISAPI/
Thermal/shipsDetection/capabilities?format=json*/
      "SceneNameOverlay":{
/*optional, scene name on-screen display*/
        "enabled":,
/*required, enable scene name OSD*/
        "normalizedScreenSize":{
/*normalized size*/
          "normalizedScreenWidth",
/*optional, float type, normalized width*/
          "normalizedScreenHeight":

```

```
/*optional, float type, normalized height*/
    },
    "positionX":,
/*optional, integer type, original point X-coordinate*/
    "positionY":
/*optional, integer type, original point Y-coordinate*/
    }
}
}
}}
}
```

C.1.26 JSON_ShipsDetectionSceneTraceList

ShipsDetectionSceneTraceList message in JSON format

```
{
  "ShipsDetectionSceneTraceList": [{
    "ShipsDetectionSceneTrace": {
      "sid":,
/*required, integer type, scene ID*/
      "sceneName": "",
/*required, string type, scene name*/
      "patrolId":,
/*optional, integer type, patrol sequence ID, it is the scene serial number in the patrol*/
      "functionMode": "",
/*optional, string type, function mode: shipFlowDetection-ship flow detection mode, dredgerDetection-dredger
detection mode*/
      "dwelltime":,
/*optional, integer type, scene dwell time, ranges from 30s to 3600s*/
      "overStayingAlarmTime":,
/*optional, integer type, ranges from 60s to 3600s. The scene dwell time should be no less than overstay alarm time*/
      "SceneNameOverlay": {
/*optional, scene name OSD*/
        "enabled":,
/*required, enable scene name OSD*/
        "ScreenSize": {
/*normalized size*/
          "ScreenWidth": 1,
/*optional, integer type, normalized width*/
          "ScreenHeight": 1
/*optional, integer type, normalized height*/
        },
        "positionX": 1,
/*optional, integer type, original point X-coordinate*/
        "positionY": 1
/*optional, integer type, original point Y-coordiante*/
      }
    }
  ]
}
```



```
}}  
}
```

C.1.27 JSON_ShipsDetectionTraceListCap

ShipsDetectionTraceListCap message in JSON format

```
{  
  "ShipsDetectionTraceListCap":{  
    "maxSize":"","  
    /*required, integer type, the maximum number of supported scenes*/  
    "ShipsDetectionTraceCap":{  
      "functionMode":{  
        /*optional, string type, supported function mode*/  
        "@opt":"shipFlowDetection"  
      },  
      "dwelltime":{  
        /*optional, integer type, scene dwell time, ranges from 30s to 3600s*/  
        "@min":30,  
        "@max":3600  
      },  
      "overStayingAlarmTime":{  
        /*optional, integer type, overstay alarm time, ranges from 60s to 3600s. The scene dwell time should be no less than  
        overstay alarm time*/  
        "@min":60,  
        "@max":3600  
      },  
      "SceneNameOverlay":{  
        /*optional, scene name OSD*/  
        "enabled":{  
          /*required, enable scene name OSD*/  
          "opt":"true,false"  
        },  
        "normalizedScreenSize":{  
          /*normalized size*/  
          "normalizedScreenWidth":{  
            /*optional, integer type, normalized width*/  
            "@min":0,  
            "@max":1  
          },  
          "normalizedScreenHeight":{  
            /*optional, integer type, normalized height*/  
            "@min":0,  
            "@max":1  
          }  
        },  
        "positionX":{  
          /*optional, integer type, original point X-coordinate*/  
          "@min":,  
          "@max":  
        },  
      },  
    },  
  },  
}
```

```
"positionY":{  
/*optional, integer type, original point Y-coordiante*/  
  "@min":,  
  "@max":  
}  
}  
}  
}  
}
```

C.1.28 JSON_thermometryOffLineCapture

thermometryOffLineCapture message in JSON format

```
{  
  "thermometryOffLineCapture":{  
    "channel": ,  
/*required, integer32, channel ID*/  
    "thermometryOffLineCaptureDataLen": ,  
/*optional, integer32, the size of offline captured picture*/  
  }  
}
```

C.1.29 JSON_TemperatureCorrect

JSON message about temperature calibration configuration parameters

```
{  
  "TemperatureCorrect": {  
    "enabled": ,  
/*optional, boolean, whether to enable the function*/  
    "streamOverlay": ,  
/*optional, string, whether to display calibrated temperature on video*/  
    "correctEnabled": ,  
/*optional, string, whether to enable temperature calibration*/  
    "emissivity": ,  
/*optional, float, emissivity, range: [0.01,1.00]*/  
    "distance": ,  
/*optional, float, distance: range: [0.3,2]m*/  
    "temperature": ,  
/*optional, float, temperature, range: [30.0,50.0]°C*/  
    "CentrePoint":{  
/*optional, normalized coordinates of black body center point, from 0 to 1000*/  
      "CalibratingCoordinates":{  
        "positionX": ,  
/*required, int, x-coordinate*/  
        "positionY": ,  
/*required, int, y-coordinate*/  
      }  
    }  
  }  
}
```

```
    },
    "correctTemperature":
/*optional, float, temperature calibration coefficient, range: [-99.0,-99.0]°C*/
  }
}
```

C.1.30 JSON_TemperatureCorrectCap

JSON message about temperature calibration capability

```
{
  "TemperatureCorrectCap":{
    "enabled":{
/*optional, boolean, whether to enable the function*/
      "@opt":[true,false]
    },
    "streamOverlay":{
/*optional, string, whether to display calibrated temperature on video*/
      "@opt":[true,false]
    },
    "correctEnabled":{
/*optional, string, whether to enable temperature calibration*/
      "@opt":[true,false]
    },
    "emissivity":{
/*optional, float, emissivity, range: [0.01,1.00]*/
      "@min":0.01,
      "@max":1
    },
    "distance":{
/*optional, float, distance: range: [0.3,2]m*/
      "@min":0.3,
      "@max":2
    },
    "temperature":{
/*optional, float, temperature, range: [30.0,50.0]°C*/
      "@min":30,
      "@max":50
    },
    "CentrePoint":{
/*optional, normalized coordinates of black body center point, from 0 to 1000*/
      "CalibratingCoordinates":{
        "positionX": ,
/*required, int, x-coordinate*/
        "positionY":
/*required, int, y-coordinate*/
      }
    },
    "correctTemperature":{
/*optional, float, temperature calibration coefficient, range: [-99.0,-99.0]°C*/
      "@min":-99,
```

```

    "@max": 99
  }
}
}

```

C.1.31 JSON_TemperatureIntervalMeasurementAlarmRuleCap

JSON message about configuration capability of interval temperature measurement rule

```

{
  "TemperatureIntervalMeasurementAlarmRuleCap": {
    "RulesList": {
      "maxSize": 21,
      /*required, int, the maximum number of rules*/
      "Rule": {
        "id": {
          /*optional, int, rule ID*/
          "@min": 1,
          "@max": 21
        },
        "enabled": {
          /*optional, boolean, whether to enable*/
          "@opt": [true, false]
        },
        "alarmType": {
          /*optional, string, alarm type: "highestTemp"-the highest temperature, "lowestTemp"-the lowest temperature*/
          "@opt": ["highestTemp", "lowestTemp"],
          "#text": "highestTemp"
        },
        "TemperatureIntervalList": {
          /*temperature interval list; up to four intervals are supported*/
          "maxSize": 4,
          /*required, int, the maximum number of intervals*/
          "TemperatureInterval": {
            "id": {
              /*optional, int, No.*/
              "@min": 1,
              "@max": 4
            },
            "enabled": {
              /*optional, boolean, whether to enable*/
              "@opt": [true, false]
            },
            "name": {
              /*optional, string, interval name*/
              "@min": 1,
              "@max": 32
            },
            "minTemperature": {
              /*optional, float, the minimum temperature, range: [-20,550], unit: Celsius; the value should be accurate to one decimal place*/

```

```

    "@min": -20,
    "@max": 550
  },
  "maxTemperature": {
/*optional, float, the maximum temperature, range: [-20,550], unit: Celsius; the value should be accurate to one decimal place*/
    "@min": -20,
    "@max": 550
  },
  "alarmColor": {
/*optional, alarm color of temperature interval*/
    "R": 1,
/*optional, int*/
    "G": 1,
/*optional, int*/
    "B": 1
/*optional, int*/
  },
  "AlarmOutputIOPortList":{
/*optional, alarm output port list*/
    "maxSize": 2,
/*required, int, the maximum number of ports*/
    "OutputIOPort":{
/*optional, int, port No.*/
      "portID": {
        "@min": 1,
        "@max": 2
      },
      "enabled": {
/*required, boolean, whether to enable*/
        "@opt":[true,false]
      }
    }
  }
}
}
}
}
}
}
}

```

C.1.32 JSON_TemperatureIntervalMeasurementAlarmRule

JSON message about interval temperature measurement rule

```
{
  "TemperatureIntervalMeasurementAlarmRule":{
    "RulesList":[{
      "Rule":{
        "id": 1,
        /*required, int, rule ID*/

```

```

    "enabled": true,
    /*optional, boolean, whether to enable*/
    "alarmType": "",
    /*optional, string, alarm type: "highestTemp"-the highest temperature, "lowestTemp"-the lowest temperature*/
    "TemperatureIntervalList":[{
    /*temperature interval list; up to four intervals are supported*/
        "TemperatureInterval":{"
            "id": 1,
            /*required, int, No.*/
            "enabled": true,
            /*optional, boolean, whether to enable*/
            "name": "",
            /*optional, string, interval name*/
            "minTemperature": 35.0,
            /*optional, float, the minimum temperature, range: [-20,550], unit: Celsius; the value should be accurate to one decimal place*/
            "maxTemperature": 37.0,
            /*optional, float, the maximum temperature, range: [-20,550], unit: Celsius; the value should be accurate to one decimal place*/
            "alarmColor":{"
            /*optional, alarm color of temperature interval*/
                "R": 1,
                /*optional, int*/
                "G": 1,
                /*optional, int*/
                "B": 1
                /*optional, int*/
            },
            "AlarmOutputIOPortList":[{
            /*optional, alarm output port list*/
                "OutputIOPort":{"
                    "portID": 1,
                    /*required, int, port No.*/
                    "enabled": true
                    /*required, boolean, whether to enable*/
                }
            }]
        }
    }]
}
}

```

C.1.33 XML_BurningPrevention

BurningPrevention message in XML format.

```
<BurningPrevention version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <enabled><!--req, xs:boolean "true,false"--></enabled>
  <mode><!--opt, xs:string, mode: manual, automatic--></mode>
```

```
<closedDuration><!--opt, xs:integer, close duration, unit: second--></closedDuration>
<shutterStatus><!--opt, xs:string, shutter status: closed, open--></shutterStatus>
<protectionMode>
  <!--opt, xs:string, protection mode: lensMovement-lens movement, shutterClose-close shutter-->
</protectionMode>
<burningRecoveryEnabled>
  <!--opt, xs:boolean, enable burning recovery or not-->
</burningRecoveryEnabled>
<movementDuration>
  <!--dep, xs:integer, duration of lens movement for burning protection, unit: minute-->
</movementDuration>
</BurningPrevention>
```

Remarks

- When **protectionMode** is "lensMovement", the node **movementDuration** is valid.
- When **protectionMode** is "shutterClose", the node **closedDuration**, **shutterStatus**, and **burningRecoveryEnabled** are valid.
- When **mode** is "automatic", the shutter is open, and cannot be configured.

C.1.34 XML_BurningPreventionCap

BurningPreventionCap message in XML format

```
<BurningPreventionCap version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <enabled opt="true,false"><!--req, xs:boolean--></enabled>
  <mode opt="manual,automatic"><!--opt, xs:string, mode: manual, automatic--></mode>
  <closedDuration min="5" max="60" default="10">
    <!--opt, xs:integer, closed status duration, unit: second-->
  </closedDuration>
  <shutterStatus opt="closed,open"><!--opt, xs:string, shutter status: closed, open--></shutterStatus>
  <protectionMode opt="lensMovement,shutterClose">
    <!--opt, xs:string, protection mode: lensMovement-lens movement, shutterClose-close shutter-->
  </protectionMode>
  <burningRecoveryEnabled opt="true,false">
    <!--opt, xs:boolean, enable burning recovery or not-->
  </burningRecoveryEnabled>
  <movementDuration min="" max="" default="">
    <!--opt, xs:integer, duration of lens movement for burning protection, unit: minute-->
  </movementDuration>
</BurningPreventionCap>
```

Remarks

- When **protectionMode** is "lensMovement", the node **movementDuration** is valid.
- When **protectionMode** is "shutterClose", the node **closedDuration**, **shutterStatus**, and **burningRecoveryEnabled** are valid.
- When **mode** is "automatic", the shutter is open, and cannot be configured.

C.1.35 XML_CAMERAPARA

CAMERAPARA message in XML format

```
<xml version="1.0" encoding="utf-8"?>
<!--req, camera parameter capability set description -->
<CAMERAPARA version="2.0">
  <ChannelList>
    <ChannelEntry>
      <ChannelNumber>1</ChannelNumber>
      <IPStartChanNoDefault><!-- optional, xs:integer, start digital channel No. The device's digital channel No. will start
from the returned No.--></IPStartChanNoDefault>
      <PowerLineFrequencyMode><!-- req, format -->
        <isNotSupportDigitalChanCfg opt="true,false"/><!-- optional, whether setting digital channels is not supported:
true=yes (not supported), false=no (supported). If this field is not returned, it indicates that setting digital channels is
supported-->
        <Range>0,1</Range><!-- req, 0-50HZ; 1-60HZ -->
        <Default>0</Default><!-- req, default value -->
      </PowerLineFrequencyMode>
      <CaptureMode>
        <!--req, correspond to byCaptureMode in NET_DVR_CAMERAPARAMCFG_EX-->
        <!--req, the device supports captureModePWithIndex and captureModeNWithIndex, when returning
captureModeP and captureModeN, the client resolves the capability set, first with captureModePWithIndex and
captureModeNWithIndex, or captureModeP and captureModeN if the former nodes are not supported-->
        <!--req, 0-close, 1-640*480@25fps, 2-640*480@30ps, 3-704*576@25fps, 4-704*480@30fps,
5-1280*720@25fps, 6-1280*720@30fps, 7-1280*720@50fps, 8-1280*720@60fps, 9-1280*960@15fps,
10-1280*960@25fps, 11-1280*960@30fps, 12-1280*1024@25fps, 13-1280*1024@30fps, 14-1600*900@15fps,
15-1600*1200@15fps, 16-1920*1080@15fps, 17-1920*1080@25fps, 18-1920*1080@30fps, 19-1920*1080@50fps,
20-1920*1080@60fps, 21-2048*1536@15fps, 22-2048*1536@20fps, 23-2048*1536@24fps, 24-2048*1536@25fps,
25-2048*1536@30fps, 26-2560*2048@25fps, 27-2560*2048@30fps, 28-2560*1920@7.5fps, 29-3072*2048@25fps,
30-3072*2048@30fps, 31-2048*1536@12.5, 32-2560*1920@6.25, 33-1600*1200@25, 34-1600*1200@30,
35-1600*1200@12.5, 36-1600*900@12.5, 37-1600@900@15, 38-800*600@25, 39-800*600@30fps,
136-640*960@25fps, 137-640*960@24fps, 142-2992*2192@25fps, 143-2992*2192@30fps, 158-384*288@8.3fps,
159-640*512@8.3fps, 160-160*120@8.3fps, 161-1024*768@8.3fps, 162-640*480@8.3fps-->
        <captureModeP opt="close,640*480@25fps,640*480@30ps,704*576@25fps,704*480@30fps,1280*720@25fps,
1280*720@30fps,1280*720@50fps,1280*720@60fps,1280*960@15fps,1280*960@25fps, 1280*960@30fps,
1280*1024@25fps,1280*1024@30fps,1600*900@15fps,1600*1200@15fps, 1920*1080@15fps,1920*1080@25fps,
1920*1080@30fps,1920*1080@50fps,1920*1080@60fps, 2048*1536@15fps,2048*1536@20fps,2048*1536@24fps,
2048*1536@25fps,2048*1536@30fps, 2560*2048@25fps,2560*2048@30fps,2560*1920@7.5fps,3072*2048@25fps,
3072*2048@30fps, 2048*1536@12.5fps,2560*1920@6.25fps,1600*1200@25fps,1600*1200@30fps,
1600*1200@12.5fps, 1600*900@12.5fps,1600@900@15fps,800*600@25fps,800*600@30fps,640*960@25fps,
640*960@24fps"/>
        <!--req, The value of captureMode in P standard-->
        <captureModeN opt="close,640*480@25fps,640*480@30ps,704*576@25fps,704*480@30fps,
1280*720@25fps, 1280*720@30fps,1280*720@50fps,1280*720@60fps,1280*960@15fps,1280*960@25fps,
1280*960@30fps,1280*1024@25fps,1280*1024@30fps,1600*900@15fps,1600*1200@15fps, 1920*1080@15fps,
1920*1080@25fps,1920*1080@30fps,1920*1080@50fps,1920*1080@60fps, 2048*1536@15fps,2048*1536@20fps,
2048*1536@24fps,2048*1536@25fps,2048*1536@30fps, 2560*2048@25fps,2560*2048@30fps,2560*1920@7.5fps,
3072*2048@25fps,3072*2048@30fps, 2048*1536@12.5fps,2560*1920@6.25fps,1600*1200@25fps,
1600*1200@30fps,1600*1200@12.5fps, 1600*900@12.5fps,1600@900@15fps,800*600@25fps,800*600@30fps,
640*960@25fps,640*960@24fps"/>
      </CaptureMode>
    </ChannelEntry>
  </ChannelList>
</CAMERAPARA>
```



```
<!--req, The value of captureMode in N standard-->
<captureModePWithIndex opt="0-close, 1-640*480@25fps,2-640*480@30ps,3-704*576@25fps,
4-704*480@30fps,5-1280*720@25fps, 6-1280*720@30fps,7-1280*720@50fps,8-1280*720@60fps,
9-1280*960@15fps,10-1280*960@25fps, 11-1280*960@30fps,12-1280*1024@25fps,13-1280*1024@30fps,
14-1600*900@15fps,15-1600*1200@15fps, 16-1920*1080@15fps,17-1920*1080@25fps,18-1920*1080@30fps,
19-1920*1080@50fps,20-1920*1080@60fps, 21-2048*1536@15fps,22-2048*1536@20fps,23-2048*1536@24fps,
24-2048*1536@25fps,25-2048*1536@30fps, 26-2560*2048@25fps,27-2560*2048@30fps,28-2560*1920@7.5fps,
29-3072*2048@25fps,30-3072*2048@30fps, 31-2048*1536@12.5fps,32-2560*1920@6.25fps,33-1600*1200@25fps,
34-1600*1200@30fps,35-1600*1200@12.5fps, 36-1600*900@12.5fps,37-1280*960@12.5fps,38-800*600@25fps,
39-800*600@30fps,40-4000*3000@12.5fps, 41-4000*3000@15fps,42-4096*2160@20fps,43-3840*2160@20fps,
44-960*576@25fps,45-960*480@30fps, 46-752*582@25fps,47-768*494@30fps,48-2560*1440@25fps,
49-2560*1440@30fps,50-720P@100fps, 51-720P@120fps,52-2048*1536@50fps,53-2048*1536@60fps,
54-3840*2160@25fps,55-3840*2160@30fps, 56-4096*2160@25fps,57-4096*2160@30fps,58-1280*1024@50fps,
59-1280*1024@60fps,60-3072*2048@50fps, 61-3072*2048@60fps,62-3072*1728@25fps,63-3072*1728@30fps,
64-3072*1728@50fps,65-3072*1728@60fps, 66-336*256@50fps,67-336*256@60fps,68-384*288@50fps,
69-384*288@60fps,70-640*512@50fps, 71-640*512@60fps,72-2592*1944@25fps,73-2592*1944@30fps,
74-2688*1536@25fps,75-2688*1536@30fps, 76-2592*1944@20fps,77-2592*1944@15fps,78-2688*1520@20fps,
79-2688*1520@15fps,80-2688*1520@25fps, 81-2688*1520@30fps,82-2720*2048@25fps,83-2720*2048@30fps,84-
336*256@25fps,85-384*288@25fps, 86-640*512@25fps,87-1280*960@50fps,88-1280*960@60fps,
89-1280*960@100fps,90-1280*960@120fps, 91-4000*3000@20fps,141-2688*1520@12.5fps"/>
<!--req, captureMode value with index in P standard-->
<captureModeNWithIndex opt="0-close,1-640*480@25fps,2-640*480@30ps,3-704*576@25fps,
4-704*480@30fps,5-1280*720@25fps, 6-1280*720@30fps,7-1280*720@50fps,8-1280*720@60fps,
9-1280*960@15fps,10-1280*960@25fps, 11-1280*960@30fps,12-1280*1024@25fps,13-1280*1024@30fps,
14-1600*900@15fps,15-1600*1200@15fps, 16-1920*1080@15fps,17-1920*1080@25fps,18-1920*1080@30fps,
19-1920*1080@50fps,20-1920*1080@60fps, 21-2048*1536@15fps,22-2048*1536@20fps,23-2048*1536@24fps,
24-2048*1536@25fps,25-2048*1536@30fps, 26-2560*2048@25fps,27-2560*2048@30fps,28-2560*1920@7.5fps,
29-3072*2048@25fps,30-3072*2048@30fps, 31-2048*1536@12.5fps,32-2560*1920@6.25fps,33-1600*1200@25fps,
34-1600*1200@30fps,35-1600*1200@12.5fps, 36-1600*900@12.5fps,37-1280*960@12.5fps,38-800*600@25fps,
39-800*600@30fps,40-4000*3000@12.5fps, 41-4000*3000@15fps,42-4096*2160@20fps,43-3840*2160@20fps,
44-960*576@25fps,45-960*480@30fps, 46-752*582@25fps,47-768*494@30fps,48-2560*1440@25fps,
49-2560*1440@30fps,50-720P@100fps, 51-720P@120fps,52-2048*1536@50fps,53-2048*1536@60fps,
54-3840*2160@25fps,55-3840*2160@30fps, 56-4096*2160@25fps,57-4096*2160@30fps,58-1280*1024@50fps,
59-1280*1024@60fps,60-3072*2048@50fps, 61-3072*2048@60fps,62-3072*1728@25fps,63-3072*1728@30fps,
64-3072*1728@50fps,65-3072*1728@60fps, 66-336*256@50fps,67-336*256@60fps,68-384*288@50fps,
69-384*288@60fps,70-640*512@50fps, 71-640*512@60fps,72-2592*1944@25fps,73-2592*1944@30fps,
74-2688*1536@25fps,75-2688*1536@30fps, 76-2592*1944@20fps,77-2592*1944@15fps,78-2688*1520@20fps,
79-2688*1520@15fps,80-2688*1520@25fps, 81-2688*1520@30fps,82-2720*2048@25fps,83-2720*2048@30fps,84-
336*256@25fps,85-384*288@25fps, 86-640*512@25fps,87-1280*960@50fps,88-1280*960@60fps,
89-1280*960@100fps,90-1280*960@120fps, 91-4000*3000@20fps,141-2688*1520@12.5fps"/>
<!--req, captureMode value with index in N standard-->

<!--req, to enable 3D noise reduction, SMD, rotation or WDR in 1080p50/1080p60 mode, the prompt will show
"please set capture mode with normal frame rate"-->
<CaptureModelIndex19>
<!--1920*1080@50fps-->
<!--req, mutually exclusive capability, digital noise reduction, line crossing detection, rotation, WDR access
capability-->
<mutexAbility opt="digitalNoiseReduction,traversingVirtualPlane,fieldDetection,corridorMode,WDR"/>
</CaptureModelIndex19>

<CaptureModelIndex20>
```

```
<!--req, 1920*1080@60fps-->
<!--req, mutually exclusive capability, digital noise reduction, line crossing detection, rotation, WDR access
capability-->
<mutexAbility opt="digitalNoiseReduction,traversingVirtualPlane,fieldDetection,corridorMode,WDR"/>
</CaptureModelIndex20>

<!--req, to enable rotation or WDR in 720p50/720p60 mode, the prompt will show "please set capture mode with
normal frame rate"-->
<CaptureModelIndex7>
<!--1280*720@50fps-->
<!--req mutually exclusive capability rotation WDR access capability-->
<mutexAbility opt="corridorMode,WDR"/>
</CaptureModelIndex7>

<CaptureModelIndex8>
<!--1280*720@60fps-->
<!--req mutually exclusive capability rotation WDR access capability-->
<mutexAbility opt="corridorMode,WDR"/>
</CaptureModelIndex8>

<CaptureModelIndex52>
<!--2048*1536@50fps-->
<!--req mutually exclusive capability rotation WDR access capability-->
<mutexAbility opt="WDR"/>
</CaptureModelIndex52>

<CaptureModelIndex53>
<!--2048*1536@60fps -->
<!--req mutually exclusive capability rotation WDR access capability-->
<mutexAbility opt="WDR"/>
</CaptureModelIndex53>

<CaptureModelIndex87>
<!--1280*960@50fps -->
<!--req mutually exclusive capability WDR access capability-->
<mutexAbility opt="WDR"/>
</CaptureModelIndex87>

<CaptureModelIndex88>
<!--1280*960@60fps -->
<!--req mutually exclusive capability WDR access capability-->
<mutexAbility opt="WDR"/>
</CaptureModelIndex88>

<CaptureModelIndex89>
<!--1280*960@100fps -->
<!--req mutually exclusive capability WDR access capability-->
<mutexAbility opt="WDR"/>
</CaptureModelIndex89>

<CaptureModelIndex90>
<!--1280*960@120fps -->
```

```

<!--req mutually exclusive capability WDR access capability-->
<mutexAbility opt="WDR"/>
</CaptureModelIndex90>
<WhiteBalance><!-- req, white balance -->
  <WhiteBalanceMode><!-- req, white balance mode -->
    <!-- req, 0-Manual, 1-AWB1, 2-AWB2, 3-Automatic Control(4~9:reserved), 11-Auto Trace,12-One Push,13-Indoor,
14-Outdoor, 15-Outdoor Auto, 16- SodiumLight Auto -->
    <!-- req, 10~16: new options for speed domes -->
    <Range>1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16</Range>
    <Default>1</Default><!-- req, default value -->
  </WhiteBalanceMode>
  <WhiteBalanceModeRGain><!-- req, R gain of white balance -->
    <Min>0</Min><!-- req, minimum value -->
    <Max>255</Max><!-- req, maximum value -->
    <Default>100</Default><!-- req, default value -->
  </WhiteBalanceModeRGain>
  <WhiteBalanceModeBGain><!-- req, B gain of white balance -->
    <Min>0</Min><!-- req, minimum value -->
    <Max>255</Max><!-- req, maximum value -->
    <Default>100</Default><!-- req, default value -->
  </WhiteBalanceModeBGain>
</WhiteBalance>

<!-- req, supported by IPC only -->
<Exposure> <!-- req, exposure -->
  <ExposureMode> <!-- req, exposure mode, reserved currently -->
    <Range>0,1</Range><!-- req, 0-manual exposure,1-auto exposure -->
    <Default>0</Default><!-- req, default value -->
  </ExposureMode>

  <ExposureSet>
    <!-- req, exposure time, 0(index): auto*8(display on the client), 40000*8 us(the actual value)-->
    <!-- req,0-auto*8(40000*8us),1-auto*5(40000*5us),2-auto*4(40000*4us),3-auto*2(40000*2us),-->
    <!-- req,4-1/25(40000us),5-1/50(20000us),6-1/100(10000us),7-1/250(4000us),8-1/500(2000us), -->
    <!-- req,9-1/750(1333us),10-1/1000(1000us),11-1/2000(500us),12-1/4000(250us),-->
    <!-- req,13-1/10,000(100us),14-1/100,000(10us), 17-1/150, 18-1/200, 20-1-1000000us, 21-1/75, 22-1/125,
23-1/175, 24-1/225,25-1/300, 26-1/400 -->
    <Range>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14</Range>
    <Default>4</Default><!-- req, default value-->
  </ExposureSet>
  <DynamicAbility>
    <!--req, IPC 5.1.0 supports to get or set abilities dynamically-->
    <dynamicAbilityLinkTo opt="wdrEnable,irisType"/>
    <!--req, Dynamic related items, WDR enable, the structure used for defining the type of len is
NET_DVR_CAMERAPARAMCFG_EX, the parameters is struWdr.byWDREnabled and struWdr.byIrisMode -->
  </DynamicAbility>
</Exposure>
  <exposureUSERSET><!--req, customized exposure time-->
    <Min>1</Min><!--req, minimum value-->
    <Max>40000</Max><!--req, maximum value-->
    <Default>20000</Default><!--req, default value-->
  </exposureUSERSET>
  <ExposureTarget> <!--req, reserved-->

```

```

    <Min>0</Min><!-- req, minimum value -->
    <Max>2000000</Max><!-- req, maximum value -->
    <Default>1000000</Default><!-- req, default value -->
  </ExposureTarget>
</Exposure>
<IrisMode> <!--req, Lens mode-->
  <!--req, 0- auto iris, 1- manual iris, 2- Piris1"Tamron 2.8-8mm F1.2 (M13VP288-IR) ", 3- Union 3-9mm F1.6-2.7
(T5280-PQ1),4- Union 2.8-12mm F1.6-2.7(T5289-PQ1), 5- HIK 3.8-16mm F1.5 (HV3816P-8MPIR), 6-HIK 11-40mm F1.7,
7- HIK 2.7-12mm F1.2 (TV2712P-MPIR), 8- MZ5721D-12MPIR, 9- MZ1555D-12MPIR, 10- MZ5721D-12MPIR(RS485), 11-
MZ1555D-12MPIR(RS485)-->
  <Range>0,1,2,3,4,5,6,7,8,9,10,11</Range>
  <Default>1</Default><!-- req, default value -->
  <Piris>
    <!--req valid when IrisMode>=2-->
    <Piris1><!--req, Tamron 2.8-8mm F1.2 (M13VP288-IR) -->
      <modeType opt="automatic, manual"/><!--req, 0-auto, 1-manual-->
      <PirisAperture min="" max=""/><!--req, level range: 1 to 100 (can be configured under the manual mode)-->
    </Piris1>
    <Piris2><!--req, Union 3-9mm F1.6-2.7 (T5280-PQ1)-->
      <modeType opt="automatic, manual"/><!--req, 0-auto, 1-manual-->
      <PirisAperture min="" max=""/><!--req, level range: 1 to 100 (can be configured under the manual mode)-->
    </Piris2>
    <Piris3><!--req, Union 2.8-12mm F1.6-2.7 (T5289-PQ1)-->
      <modeType opt="automatic, manual"/><!--req, 0-auto, 1-manual-->
      <PirisAperture min="" max=""/><!--req, level range: 1 to 100 (can be configured under the manual mode)-->
    </Piris3>
    <Piris4><!--req, HIK 3.8-16mm F1.5 (HV3816P-8MPIR)-->
      <modeType opt="automatic, manual"/><!--req, 0-auto, 1-manual-->
      <PirisAperture min="" max=""/><!--req, level range: 1 to 100 (can be configured under the manual mode)-->
    </Piris4>
    <Piris6><!--req,HIK 11-40mm F1.7 (HV1140P-8MPIR)-->
      <modeType opt="automatic, manual"/><!--req, 0-auto, 1-manual-->
      <PirisAperture min="" max=""/><!--req, level range: 1 to 100 (can be configured under the manual mode)-->
    </Piris6>
    <Piris7><!--req, HIK 2.7-12mm F1.2 (TV2712P-MPIR) -->
      <modeType opt="automatic, manual"/><!--req, 0-auto, 1-manual-->
      <PirisAperture min="" max=""/><!--req, level range: 1 to 100 (can be configured under the manual mode)-->
    </Piris7>
  </Piris>
</IrisMode>

<AutoApertureLevel> <!-- req, auto aperture sensitivity -->
  <Min>0</Min><!-- req, minimum value -->
  <Max>15</Max><!-- req, maximum value -->
  <Default>7</Default><!-- req, default value -->
</AutoApertureLevel>

<FocusMode> <!--req, reserved-->
  <!--req, 0- manual focus; 1-auto focus; 2- auto back focus-->
  <Range>0,1,2</Range>
  <Default>0</Default><!--req, default value-->
  <GainLevel><!--req, gain, ranges from 0 to 100-->

```

```

    <Min>0</Min><!--req, minimum value-->
    <Max>100</Max><!--req, maximum value-->
    <Default>50</Default><!-- req, default value-->
</GainLevel>
<BrightnessLevel><!--req, brightness, ranges from 0 to 100 -->
    <Min>0</Min><!-- req, minimum value -->
    <Max>100</Max><!-- req, maximum value -->
    <Default>50</Default><!-- req, default value -->
</BrightnessLevel>
<ContrastLevel><!--req, contrast, ranges from 0 to 100-->
    <Min>0</Min><!--req, minimum value-->
    <Max>100</Max><!--req, maximum value-->
    <Default>50</Default><!--req, default value-->
</ContrastLevel>
<SharpnessType><!--sharpness type-->
    <Range><!--sharpness type range (for speed dome): 0-automatic, 1-manual--></Range>
    <Default><!--default value--></Default>
</SharpnessType>
<SharpnessLevel><!--req, sharpness-->
    <!--req, IPC(min/max); speed dome/zoom camera(Range), from 0 to 100-->
    <Min>0</Min><!-- req, minimum value -->
    <Max>100</Max><!-- req, maximum value -->
    <Range>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17</Range>
    <!-- req, speed dome: 0- auto, 1- manual, 2-1, 3-2, 4-3, 5-4, 6-5, 7-6, 8-7, 9-8, 10-9, 11-10, 12-11, 13-12, 14-13,
15-14, 16-15, 17-16 -->
    <Default>50</Default><!-- req, default value -->
</SharpnessLevel>
<HorizonAperture><!-- req, horizontal sharpness -->
    <!-- req, special for speed dome: 1~64 -->
    <Min>1</Min><!-- req, minimum value -->
    <Max>64</Max><!-- req, maximum value -->
    <Default>10</Default><!-- req, default value -->
</HorizonAperture>
<VerticalAperture><!-- req, vertical sharpness -->
    <!-- req, special for speed dome: 1~64 -->
    <Min>1</Min><!-- req, minimum value -->
    <Max>64</Max><!-- req, maximum value -->
    <Default>10</Default><!-- req, default value -->
</VerticalAperture>

<LaserConfig>
    <controlMode opt="auto,manual"/>
    <autoMode><!--req,Control Mode-->
        <sensitivity min="0" max="100"/><!--req,laser light sensitivity-->
        <triggerMode opt=" cameraModuleTrigger, photoresistanceTrigger"/>
        <!--req laser light triggering mode-->
        <limitBrightness min="0" max="100"/>
        <!--req,laser light brightness limitation-->
        <angle min="1" max="36"/><!--req,laser light angle-->
        <enable opt="true,false" />
        <!-- dep, enable manual control laser: 0- No, 1- Yes -->
        <illumination min="0" max="100" /><!-- dep, laser light strength limit-->
    </autoMode>
</LaserConfig>

```

```

<lightAngle min="0" max="100" /><!-- dep, light angle -->
</autoMode>

<manualMode>
  <sensitivity min="0" max="100" /><!--req,laser light sensitivity-->
  <triggerMode opt=" cameraModuleTrigger, photoresistanceTrigger"/>
  <!--req laser light triggering mode-->
  <brightness min="0" max="255" /><!--req,laser light brightness-->
  <angle min="1" max="36" /><!--req,laser light angle-->
</manualMode>
</LaserConfig>

<ChromaSuppress><!-- req, color suppression -->
  <!--req, special for speed dome: 0~100 -->
  <Min>0</Min><!--req, minimum value -->
  <Max>100</Max><!--req, maximum value -->
  <Default>50</Default><!--req, default value -->
</ChromaSuppress>
<SaturationLevel><!--req, saturation, from 0 to 100 -->
  <Min>0</Min><!--req, minimum value -->
  <Max>100</Max><!--req, maximum value -->
  <Default>50</Default><!--req, default value -->
</SaturationLevel>
<HueLevel><!--req, hue, from 0 to 100-->
  <Min>0</Min><!--req, minimum value -->
  <Max>100</Max><!--req, maximum value -->
  <Default>50</Default><!--req, default value -->
</HueLevel>
<GammaCorrection><!--req, gamma correction-->
  <GammaCorrectionEnabled><!--req, 0-disable, 1-enable-->
  <Range>0,1</Range>
  <Default>0</Default><!--req, default value-->
</GammaCorrectionEnabled>
  <GammaCorrectionLevel><!--req, the level of Gamma correction-->
  <Min>0</Min>
  <Max>100</Max>
  <Default>50</Default><!--req, default value -->
</GammaCorrectionLevel>
</GammaCorrection>
<WDR><!--req, wide dynamic range-->
  <WDREnabled><!--req, 0-disable, 1-enable, 2-auto-->
  <Range>0,1,2</Range>
  <Default>0</Default><!--req, default value-->
</WDREnabled>
  <isNotSupportDigitalChanCfg opt="true,false" /><!--optional, whether setting digital channels is not supported:
true=yes (not supported), false=no (supported). If this field is not returned, it indicates that setting digital channels is
supported-->
  <WDRLevel1><!--req, level 1 of wide dynamic range, from 0 to 15-->
  <Min>0</Min><!-- req, minimum value-->
  <Max>15</Max><!-- req, maximum value-->
  <Range>0,1,2</Range><!--req, speed dome: 0- low, 1- medium, 2- high-->
  <Default>0</Default><!--req, default value-->

```

```

</WDRLevel1>
<WDRLevel2><!--req, level 2 of wide dynamic range, from 0 to 15-->
  <Min>0</Min><!--req, minimum value-->
  <Max>15</Max><!--req, maximum value-->
  <Default>0</Default><!--req, default value-->
</WDRLevel2>
<WDRContrastLevel><!--req, contrast of wide dynamic range, from 0 to 100 -->
  <Min>0</Min><!-- req, minimum value -->
  <Max>100</Max><!-- req, maximum value -->
  <Default>50</Default><!-- req, default value -->
</WDRContrastLevel>
</WDR>
<DayNightFilter><!--req, day and night switch -->
  <DayNightFilterType><!--req, day and night switch mode -->
    <!--req, 0- day,1- night, 2- auto, 3- timing, 4- triggered by alarm input , 5- Auto mode 2(no photosensitivity)-->
    <Range>0,1,2,3,4</Range>
    <Default>2</Default><!--req, default value -->
  </DayNightFilterType>
  <SwitchSchedule>
    <SwitchScheduleEnabled><!--req, reserved -->
      <Range>0,1</Range><!--req, 0- disable 1- enable -->
      <Default>1</Default><!--req, default value -->
    </SwitchScheduleEnabled>
    <DayToNightFilterLevel><!--req, sensitivity of switching day to night -->
      <Range>0,1,2,3,4,5,6,7,8,9,10,11,12</Range>
      <!--req, 0, 1, 2, 3, 4, 5, 6, 7, 10-low, 11-medium, 12-high -->
      <!-- req, (10~12: new options for speed domes) -->
      <Default>3</Default><!-- req, default value -->
    </DayToNightFilterLevel>
    <NightToDayFilterLevel><!-- req, sensitivity of switching night to day -->
      <Range>0,1,2,3,4,5,6,7,8,9,10,11,12</Range>
      <!-- req, 0, 1, 2, 3, 4, 5, 6, 7, 10-low, 11-medium, 12-high -->
      <!-- req, (10~12: new options for speed domes) -->
      <Default>3</Default><!-- req, default value -->
    </NightToDayFilterLevel>
    <DayNightFilterTime><!-- req, filtering time of switching day to night -->
      <!-- req, IPC(min/max); speed dome/zoom camera(Range) -->
      <Min>10</Min><!-- req, minimum value -->
      <Max>120</Max><!-- req, maximum value -->
      <Range>0,1,2,3,4,5,6,7</Range>
      <!-- req, zoom camera/speed dome: 0-2S, 1-3S, 2-5S, 3-10S, 4-15S, 5-30S, 6-45S, 7-60S -->
      <Default>55</Default><!-- req, default value -->
    </DayNightFilterTime>
    <NightDayFilterTime>
      <!-- req, IPC(min/max); speed dome/zoom camera(Range) -->
      <Min>0</Min><!-- req, minimum value -->
      <Max>120</Max><!-- req, maximum value -->
      <Range>0,1,2,3,4,5,6,7</Range>
      <!-- req, zoom camera/speed dome:0-2S,1-3S,2-5S,3-10S,4-15S,5-30S,6-45S,7-60S -->
      <Default>55</Default><!-- req, default value -->
    </NightDayFilterTime>
  <TimeSchedule><!--2012-08-29-->

```

```

    <BeginTime>1</BeginTime><!--req, 1 means it supports the beginning time-->
    <EndTime>1</EndTime><!--req, 1 means it supports the ending time-->
  </TimeSchedule>
</SwitchSchedule>
<AlarmInTrigType><!--2012-08-29-->
  <Range>0,1</Range><!--req, triggered status of alarm input: 0- day, 1- night-->
</AlarmInTrigType>
<DayNightFilterandGain>
  <!--opt, whether to support setting day/night auto-switch and gain simultaneously-->
  <enabled><!--req, if this function is supported, this node must exist and be set to "true"--></enabled>
</DayNightFilterandGain>
</DayNightFilter>
<Backlight><!-- req, backlight compensation -->
  <BacklightMode><!-- req, option of backlight compensation -->
    <!-- req, 0-closed, 1-UP, 2-DOWN, 3-LEFT, 4-RIGHT, 5-MIDDLE, 6-customized, 10-open, 11-auto, 12- multi-zone
backlight compensation -->
    <!-- req, (10~12: new options for speed domes, when the value is 10(open), it supports to adjust the
compensation level) -->
    <Range>0,1,2,3,4,5,6,7,8,9,10,11,12</Range>
    <Default>0</Default><!-- req, default value -->
  </BacklightMode>
  <BacklightLevel><!-- req, backlight compensation level -->
    <!-- req, IPC, 0-15 -->
    <Min>0</Min><!-- req, minimum value -->
    <Max>15</Max><!-- req, maximum value -->
    <Range>0,1,2</Range>
    <!-- req, speed dome/zoom camera: 0-low, 1-medium, 2-high -->
    <Default>0</Default><!-- req, default value -->
  </BacklightLevel>
</Backlight>
<LowLightLimit> <!--req, low illumination electronic shutter-->
  <LowLightLimitEnabled><!--req, enable: 0-closed, 1-open-->
    <Range>0,1</Range>
    <Default>0</Default><!-- req, default value -->
  </LowLightLimitEnabled>
  <LowLightLimitLevel> <!-- req, the level of low illumination electronic shutter -->
    <!-- req, speed dome 0- slow shutter*2, 1-slow shutter*3, 2-slow shutter*4, 3-slow shutter*8, 4-slow
shutter*16, 5-slow shutter*32, 6-1, 7-2, 8-3, 9-4, 10-5, 11-6, 12-low, 13-medium, 14-high -->
    <Range>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14</Range>
    <Default>0</Default><!-- req, default value -->
  </LowLightLimitLevel>
</LowLightLimit>
<ImageStabilize>
  <ImageStabilizeLevel> <!--req, image stabilization level-->
    <Range>0,1,2</Range><!--req, speed dome: 0-low, 1-medium, 2-high-->
    <Default>0</Default><!--req, default value -->
  </ImageStabilizeLevel>
</ImageStabilize>
<CameraIRCCorrection> <!--req, the movement infrared correction function-->
  <Range>0,1,2</Range><!--req, speed dome: 0-auto, 1-open, 2-closed-->
  <Default>0</Default><!--req, default value-->
</CameraIRCCorrection>

```



```

<HighSensitivitySupport>
  <!--req, whether supports setting high sensitivity: 1- support, no this node if not support-->
  <Range>1</Range>
</HighSensitivitySupport>
<InitializeLensSupport>
  <!--req, whether supports initializing the Lens: 1-support, no this node if not support-->
  <Range>1</Range>
</InitializeLensSupport>
<CameraResetSupport>
  <!--req, whether supports rebooting movement: 1-support, no this node if not support-->
  <Range>1</Range>
</CameraResetSupport>
<CameraRestoreSupport>
  <!--req, whether supports resuming movement to the factory settings: 1-support, no this node if not support-->
  <Range>1</Range>
</CameraRestoreSupport>

<Mirror> <!--req, mirror-->
  <Range>0,1,2,3</Range><!--req, 0-off, 1-leftright, 2-updown, 3-center-->
  <Default>0</Default><!-- req, default value -->
</Mirror>

<EPTZ><!-- req, E-PTZ -->
  <!-- req, 1-support, and there is no this node if not support -->
  <Range>1</Range>
</EPTZ>

<LOCALOUTPUT><!--req, local output-->
  <!--req, 0-not support, 1-support-->
  <!--req, 365: mini-dome and cube camera don't support local output -->
  <!--req, 6467:BNC-0,1, 10-closed, 11-scaling output, 12-cropping output,-->
  <!--req, 13-cropping and scaling output (10~13: special for speed dome);-->
  <!--req, HDMI®-0:not support,20:HDMI®(720P50),21:HDMI®(720P60),22:HDMI®(1080I60)-->
  <!--req, 23 : HDMI®(1080I50), 24 : HDMI®(1080P24), 25 : HDMI®(1080P25),-->
  <!--req, 26:HDMI®(1080P30), 27 : HDMI®(1080P50), 28 : HDMI®(1080P60)-->
  <Range>0,1,10,11,12,13,20,21,22,23,24,25,26,27,28</Range>
  <Default>1</Default><!-- req, default value -->
</LOCALOUTPUT>

<DigitalNoiseReduction><!--req, noise reduction-->
  <DigitalNoiseReductionEnable>
    <!-- req, 0-closed,1-normal mode,2-expert mode,(3~9:reserved),10-open -->
    <!-- req, (10: new options for speed domes, when the value is 10(open), it supports to adjust noise reduction
level (that is, DigitalNoiseReductionLevel is valid) ) -->
    <Range>0,1,2,3,4,5,6,7,8,9,10</Range>
    <Default>0</Default><!-- req, default value -->
  </DigitalNoiseReductionEnable>
  <DigitalNoiseReductionLevel>
    <!--req, digital noise reduction level in normal mode, from 0 to 100-->
    <Min>0</Min><!-- req, minimum value -->
    <Max>100</Max><!-- req, maximum value -->
  </DigitalNoiseReductionLevel>

```

```

    <!-- req, speed dome: 0-low,1-medium,2-high,
3-1,4-2,5-3,6-4,7-5,8-6,9-7,10-8,11-9,12-10,13-11,14-12,15-13,16-14,17-15 -->
    <Range>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17</Range>
    <Default>50</Default><!-- req, default value -->
</DigitalNoiseReductionLevel>
<DigitalNoiseSpectralLevel>
    <!-- req, spatial digital noise reduction level in expert mode, from 0 to 100-->
    <Min>0</Min><!-- req, minimum value -->
    <Max>100</Max><!-- req, maximum value -->
    <Default>50</Default><!-- req, default value -->
</DigitalNoiseSpectralLevel>
<DigitalNoiseTemporalLevel>
    <!-- req, temporal digital noise reduction level in expert mode, from 0 to 100-->
    <Min>0</Min><!-- req, minimum value -->
    <Max>100</Max><!-- req, maximum value -->
    <Default>50</Default><!-- req, default value -->
</DigitalNoiseTemporalLevel>
<DigitalNoiseRemove2DEnable><!--whether to enable 2D noise reduction for captured frames: 0-disable, 1-
enable-->
    <Range>0,1</Range>
    <Default>0</Default><!--default value-->
</DigitalNoiseRemove2DEnable>
<DigitalNoiseRemove2DLevel><!--2D noise reduction level for captured frames, which is between 0 and 100-->
    <Min>0</Min><!--minimum value-->
    <Max>100</Max><!--maximum value-->
    <Default>50</Default><!--default value-->
</DigitalNoiseRemove2DLevel>
<mutexAbility opt="1920*1080@50fps,1920*1080@60fps"/>
    <!--req, if 1920*1080@50fps or 1920*1080@60fps needs to be enabled after 3D noise reduction, SMD,
rotation, or WDR (wide dynamic range) is enabled, users will be prompted to disable 3D noise reduction, SMD,
rotation, and WDR in the self-adaptive mode and schedule mode first. This node is mutually exclusive with
CaptureMode-->
</DigitalNoiseReduction>
<SceneMode><!--req, scene mode: 0-outdoor, 1-indoor-->
    <Range>0,1</Range><!--req, 0-outdoor, 1-indoor, 2-default, 3-low light-->
    <Default>0</Default><!--req, default value-->
</SceneMode>
<ColorRange><!--req, color scale range-->
    <Range>0,1</Range><!--req, 0:16-235, 1:0-255-->
    <Default>0</Default><!--req, default value-->
</ColorRange>
<DigitalZoom><!--req, digital zoom, special for thermal network camera-->
    <Range>0,1,2,3,4,5</Range>
    <!-- req, digital zoom: 0-closed, 1-x2, 2-x4, 3-x8, 4-x16, 5-x32 -->
    <Default>0</Default><!--req, default value-->
</DigitalZoom>
<DeadPixelDetect><!--req, dead pixel detection, 1-support, and there is no this node if not support-->
    <Range>1</Range>-->
</DeadPixelDetect>

<LINEENCODING><!--req, whether it supports line coding capacity: 1-support, and there is no this node if not
support-->

```

```

    <Range>1</Range>
  </LINEENCODING>
  <!--req, whether it supports one-key focus or not: 1- support, and there is no this node if not support-->
  <OnepushFocus>1</OnepushFocus>

  <Dehaze><!--req, de-haze-->
    <DehazeEnable>0,1</DehazeEnable>
    <!--req, enable de-haze mode or not: 0-no, 1-adaptive mode-->
  </Dehaze>

  <!--req, the following from dimmer mode to auto shutter compensation are special for thermal network camera--
>
  <DimmerMode>
    <!--req, dimmer mode: 0-semiautomatic, 1-automatic-->
    <Range>0,1</Range>
    <Default>0</Default><!--req, default value-->
  </DimmerMode>

  <PaletteMode>
    <!-- req, palette: 0- white heat, 1-black heat, 2-palette2, ..., 8-palette8, 9-fusion 1, 10-rainbow, 11-fusion 2, 12-
iron red 1, 13-iron red 2, 14-sepia, 15-color 1, 16-color 2, 17-ice & fire, 18-rain, 19-red hot, 20-green hot, 21-dark blue,
22-color 3-->
    <Range>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22</Range>
    <Default>0</Default><!-- req, default value -->
  </PaletteMode>

  <EnhancedMode>
    <!-- req, enhanced mode(detection object surrounding): 0- not enhanced, 1-1, 2-2, 3-3, 4-4 -->
    <Range>0,1,2,3,4</Range>
    <Default>0</Default><!-- req, default value -->
  </EnhancedMode>

  <FilterSwitch>
    <!-- req, filter switch: 1-support -->
    <Range>1</Range>
  </FilterSwitch>

  <FocusSpeed>
    <!-- req, focus speed: 0~10 -->
    <Min>0</Min><!-- req, minimum value -->
    <Max>10</Max><!-- req, maximum value -->
    <Default>5</Default><!-- req, default value -->
  </FocusSpeed>

  <AutoCompensationInterval><!--req, time interval of auto shutter compensation-->
    <!-- req, timing auto shutter compensation: 1~120, unit: minute -->
    <Min>1</Min><!-- req, minimum value -->
    <Max>120</Max><!-- req, maximum value -->
    <Default>60</Default><!-- req, default value -->
  </AutoCompensationInterval>

  <SmartIR><!--2012-08-29-->

```

```
<Range>0,1</Range><!--req,SMART IR: 0-closed, 1-open-->
<modeType opt="automatic, manual"><!--req,0-auto 1-manual><-req valid when switch is on>
<IRDistance min="" max=""/><!--req,level 1-100(can be set in manual mode)-->
<ShortIRDistance min="" max=""/>
  <!--req,the level of short light 1-100(can be set in manual mode)-->
<LongIRDistance min="" max=""/>
  <!--req,the level of long light 1-100(can be set in manual mode)-->
</SmartIR>

<Illumination><!--2012-08-29-->
  <Range>0,1</Range><!--req,low light: 0-closed, 1-open-->
</Illumination>

<LightInhibit><!--2012-08-29-->
  <LightInhibitEnable opt="true,false"/>
  <!--req, enable high light compensation: 0-closed, 1-open-->
  <isNotSupportDigitalChanCfg opt="true,false"/><!--optional, whether setting digital channels is not supported:
true-yes (not supported), false-no (supported). If this field is not returned, it indicates that setting digital channels is
supported-->
  <level min="0" max="100"/>
  <!--req,high light compensation level-->
</LightInhibit>

<GrayLevel><!--2012-08-29-->
  <Range>0,1</Range><!--req,grayscale value range,0- [0,255], 1- [16,235]-->
</GrayLevel>

<AutoFocusMode><!--req, focus mode of zoom camera and speed dome-->
  <FocusModeSet>
    <!-- req, focus mode: 0-auto,1-manual,2-once,3-semiautomatic -->
    <Range>0,1,2,3</Range>
    <Default>0</Default><!-- req, default value -->
  </FocusModeSet>
  <AFModeChoose>
    <!-- req, auto focus mode: 0-closed, 1-mode A, 2-mode B, 3-mode AB, 4-mode C -->
    <Range>0,1,2,3,4</Range>
    <Default>0</Default><!-- req, default value -->
  </AFModeChoose>
  <MinFocusDistance>
    <!-- req, minimum focusing distance: 0- automatic, 0xffff- unlimited -->
    <Range>0,1,2,5,10,30,50,100,150,200,300,500,600,800,1000,2000,0xffff</Range>
    <Default>0</Default><!-- req, default value -->
  </MinFocusDistance>
  <ZoomSpeedLevel> <!-- req, zoom speed -->
    <!-- req, 0-0, 1-1, 2-2, 3-3, 4-4, 10-low, 11-medium, 12-high -->
    <!-- req, (0-4: special for zoom camera, 10-12: special for speed dome) -->
    <Range>0,1,2,3,4,5,6,7,8,9,10,11,12</Range>
    <Default>0</Default><!-- req, default value -->
  </ZoomSpeedLevel>
  <FocusSpeedLevel>
    <!-- req, focus speed: 0-low, 1-medium, 2-high -->
    <Range>0,1,2</Range>
```

```

    <Default>0</Default><!-- req, default value -->
  </FocusSpeedLevel>
</AutoFocusMode>
<assistFocus opt="true"/><!--req, whether to enable assist zoom: 0-No, 1-Yes -->
<focusSensitivity min="0" max="2" def="1"/>
  <!--opt, focus sensitivity, ranges from 0 to 2, it is valid when the focus mode is auto or semi-auto-->
<relativeFocusPos min="0" max="4000" def=""/>
  <!--opt, xs:integer, relative focus sensitivity, low 16 bytes indicate focus value (ranges from 0 to 4000), and high
16 bytes indicate temperature value under current focus, it is valid when the focus mode is manual or semi-auto-->
</FocusMode>

<AutoExposureMode>
  <!-- req, exposure and gain control of zoom camera and speed dome-->
  <ExposureSet>
    <!-- req, exposure mode: 0-manual mode, 1-auto exposure, 2-aperture priority, 3-shutter priority, 4-gain priority
-->
    <Range>0,1,2,3,4</Range>
    <Default>0</Default><!-- req, default value -->
  </ExposureSet>
  <IrisSet> <!-- req, aperture -->
    <!-- req, 0-F1.2,1-F1.4,2-F1.6,3-F1.67,4-F1.8,5-F1.85,6-F1.96,7-F2.0,8-F2.11 -->
    <!-- req, 9-F2.2,10-F2.4,11-F2.41,12-F2.64,13-F2.8,14-F2.86,15-F3.13,16-F3.2 -->
    <!-- req, 17-F3.4,18-F3.53,19-F3.7,20-F3.95,21-F4.0,22-F4.4,23-F4.49,24-F4.8 -->
    <!-- req, 25-F5.35,26-F5.6,27-F6.38,28-F6.4,29-F6.8,30-F7.90,31-F8.0,32-F8.8 -->
    <!-- req, 33-F9.6,34-F11,35-F11.06,36-F12,37-F14,38-F16,39-F16.60,40-F17 -->
    <!-- req, 41-F19,42-F22,43-F24,44-F33.19,45-F34 -->

<Range>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,3
9,40,41,42,43,44,45</Range>
    <Default>0</Default><!-- req, default value -->
  </IrisSet>
  <ShutterSet> <!-- req, shutter -->
    <!-- req,0-closed, 1-auto x1,2-auto x2,3-auto x4,4-auto x8,5-auto x16,6-auto x32,-->
    <!-- req,7-auto x64, 8-auto x128, 9-1/1, 10-1/2, 11-1/3, 12-1/4, 13-1/6, 14-1/8,-->
    <!-- req,15-1/12, 16-1/15, 17-1/25, 18-1/30, 19-1/50, 20-1/60, 21-1/75, 22-1/90,-->
    <!-- req,23-1/100, 24-1/120, 25-1/125, 26-1/150, 27-1/180, 28-1/200, 29-1/215,-->
    <!-- req,30-1/250, 31-1/300, 32-1/350, 33-1/425, 34-1/500, 35-1/600, 36-1/725,-->
    <!-- req,37-1/1000, 38-1/1250, 39-1500, 40-1/1750, 41-1/2000, 42-1/2500, 43-3000,-->
    <!-- req,44-1/3500, 45-1/4000, 46-1/6000, 47-1/10000, 48-1/30000, 49-1/100000 -->

<Range>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,2/9,30,31,32,33,34,35,36,37,38,
39,40,41,42,43,44,45,46,47,48,49</Range>
    <Default>0</Default><!-- req, default value -->
  </ShutterSet>
  <GainSet><!-- req, gain: 0~100 -->
  <Min>0</Min>
  <Max>100</Max>
  <!-- req, : 0-closed, 1-low, 2-medium, 3-high, 4-0, 5-1, 6-2, 7-3, 8-4, 9-5, 10-6, 11-7, 12-8, 13-9, 14-10, 15-11,
16-12, 17-13, 18-14, 19-15 -->
  <Range>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19</Range>
  <!-- req, (0~19: special for speed dome) -->
  <Default>50</Default><!-- req, default value -->

```

```

</GainSet>
<GainLimit> <!-- req, gain limit -->
  <!-- req, speed dome -->
  <Min>0</Min><!-- req, minimum value -->
  <Max>0x0f</Max><!-- req, maximum value -->
  <Default>0</Default><!-- req, default value -->
</GainLimit>
<ExposureComp>
  <!-- req, exposure compensation: 0~100 -->
  <Min>0</Min>
  <Max>100</Max>
  <Default>50</Default><!-- req, default value -->
  <!-- req, : 0-closed, 1-low, 2-medium, 3-high, 4-0, 5-1, 6-2, 7-3, 8-4, 9-5, 10-6, 11-7, 12-8, 13-9, 14-10, 15-11,
16-12, 17-13, 18-14 -->
  <!-- req, (0~18: special for speed dome) -->
  <Range>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19</Range>
</ExposureComp>
</AutoExposureMode>

<ZoomPara>
  <!-- req, zoom parameter of zoom camera/speed dome -->
  <ZoomDisplay>
    <!-- req, zoom display: 0-closed, 1-open -->
    <Range>0,1</Range>
    <Default>0</Default><!-- req, default value -->
  </ZoomDisplay>
  <ZoomLimit>
    <!-- req, zoom camera(Range); speed dome(min,max)-->
    <!-- req, zoom limit of zoom camera: 0-10,1-18,2-20,3-22,4-23,5-30,6-36,7-37,8-38,9-39,10-40,11-unlimited -->
    <Range>0,1,2,3,4,5,6,7,8,9,10,11</Range>
    <!-- req, optical zoom of speed dome: 1~50 -->
    <Min>1</Min>
    <Max>50</Max>
    <Default>1</Default><!-- req, default value -->
  </ZoomLimit>
  <DigitalZoom>
    <!-- req, zoom camera(Range);speed dome(min,max) -->
    <!-- req, digital zoom of zoom camera: 0-closed, 1- *2, 2-*4, 3-*8, 4-*10, 5-*12 -->
    <Range>0,1,2,3,4,5</Range>
    <!-- req, digital zoom of speed dome: 0~30 -->
    <Min>1</Min>
    <Max>30</Max>
    <Default>1</Default><!-- req, default value -->
  </DigitalZoom>
</ZoomPara>

<SnapExposure>
  <!-- req, exposure control general triggered snapshot -->
  <SnapMode>
    <!-- req, snapshot mode: 0- snapshot mode 1, 1- snapshot mode 2, 2- snapshot mode 3 -->
    <Range>0,1,2</Range>
    <Default>0</Default><!-- req, default value -->

```

```

</SnapMode>
<SnapGain1>
  <!-- req, snapshot gain 1: 0-100 -->
  <Min>0</Min>
  <Max>100</Max>
  <Default>50</Default><!-- req, default value -->
</SnapGain1>
<SnapGain2>
  <!-- req, snapshot gain 2: 0-100 -->
  <Min>0</Min>
  <Max>100</Max>
  <Default>50</Default><!-- req, default value -->
</SnapGain2>
</SnapExposure>

<!--req, dynamic contrast ratio level of intelligent traffic camera-->
<DynamicContrast>
  <DynamicContrastLevel>
    <Min>0</Min>
    <Max>100</Max>
    <Default>50</Default><!--req, default value-->
  </DynamicContrastLevel>
</DynamicContrast>

<!--req, Rotation Mode-->
<CorridorMode>
  <corridorModeFunEnable opt="true,false"/>
  <!--req,Enable or not, true-Enable, false-Disable-->
  <!--req,If enable 1080p50/1080p60 after the 3D DNR, SMD, rotation or WDR is enabled, the prompt will pop up
"Close the 3D DNR, SMD, rotation and WDR under the self-adaptive and continuous mode first."-->
  <!--req,If enable 720p50/720p60 after the rotation or WDR is enabled, the prompt will pop up "Close rotation and
WDR under the self-adaptive and continuous mode first."-->
  <mutexAbility opt="19-1920*1080@50fps,20-1920*1080@60fps,8-1280*720@60fps,7-1280*720@50fps"/>
  <!--req Mutex among the CaptureMode 1920*1080@50fps, 1920*1080@60fps, 1280*720@60fps and
1280*720@50fps, prompt will pop up when enabled any of the modes.-->
</CorridorMode>

<ISPAdvanceCfg><!--req,ISP supports return or not.-->
  <ISPSupportMode opt="dayMode,nightMode"/>
  <!--req Mode supported by ISP-->
  <workMode opt="auto,schedule"/>
  <!--req,0-Auto,1-Scheduled switch-->
  <TimeSchedule>
    <beginTime opt="hour,min,sec,millisec"/>
    <!--req Type of start time period-->
    <endTime opt="hour,min,sec,millisec"/>
    <!--req Type of end time period-->
  </TimeSchedule>
  <ISPCfgSupport opt="whiteBalanceMode,whiteBalanceModeRGain,whiteBalanceModeBGain,exposureSet,
    exposureUserSet,gainLevel,brightnessLevel,contrastLevel,sharpnessLevel,WDREnabled,
    WDRLevel1,WDRLevel2,WDRContrastLevel,backlightMode,backlightLevel,position1,
    position2,imageStabilizeEnable,imageStabilizeLevel,digitalNoiseReductionEnable,

```

```
        digitalNoiseReductionLevel,digitalNoiseSpectralLevel,digitalNoiseTemporalLevel,
        dehazeEnable,dehazeLevel,lightInhibitEnable,lightInhibitLevel,grayLevel"/>
</ISPAdvanceCfg>

<supportCCDFunc opt="whiteBalance,exposure,WDR,dayNightFilter,gammaCorrection,digitalNoiseReduction,
        backLight,lowLight,focus,infrared,domeAemode,dehaze,parkAction,elecStab,other,ISP,laser"/>
<!--req supported front-end parameter capabilities.-->

<!--req, Illumination Enhancement Capture Camera v3.5-->
<BrightCompensate>
    <brightCompensate min="0" max="100"/>
</BrightCompensate>

<!--req, Exposure Control Capture Camera v3.5-->
<ExposureSegment>
    <exposureSegmentEnable opt="true,false"/>
    <!--req,Enable or not, true-Enable, false-Disable-->
</ExposureSegment>

<LensDistortionCorrection>
    <enable opt="false,true" default="false"/>
    <!--req,Lens Distortion Correction (0-Disable/1-Enable)-->
    <mutexAbility opt="WDR"/>
    <!--req,mutex in WDR -->
    <level min="1" max="3">
        <!--opt, xs:integer, Distortion Correction Level, 1-3-->
    </level>
    <zoomedInDistantView>
        <!--dep, xs:integer, Remote zoom, takes effect when Distortion Correction is enabled.-->
        <enabled>
            <!--req, xs:bool, "true,false"-->
        </enabled>
        <level min="1" max="3">
            <!--opt, xs:integer, Correction level of remote zoom, 1-3-->
        </level>
    </zoomedInDistantView>
    <horizontalFOV min="0" max="100">
        <!--opt, xs:integer, Horizontal FOV[0-100]-->
    </horizontalFOV>
    <verticalFOV min="0" max="100">
        <!--opt, xs:integer, Vertical FOV[0,100]-->
    </verticalFOV>
</LensDistortionCorrection>
<BrightnessSuddenChangeSuppressionCap>
    <enabled opt="true,false">
        <!--req, xs:boolean, brightness sudden change suppression-->
    </enabled>
</BrightnessSuddenChangeSuppressionCap>

<DPCParam>
    <!--req,DPC-->
    <ctrltype
```



```
opt="correct,cancelCorrect,crossDisplayOpen,crossDisplayClose,point,up,down,right,left,allCorrect,save"/>
  <!--req,correction, cancel correction, enable/disable DPC cross display, DPC coordinate, up-forward offset of DPC
coordinate,
  down-forward offset of DPC coordinate, right-forward offset of DPC coordinate, left-forward offset of DPC
coordinate, DPC all, save defective Pixel-->
  <dpcMode opt="manual,auto" def="auto"/>
  <!--req,xs:string,"manual-Manual Correction, auto-Auto Correction, if device does not support this node, all will
be handled manually"-->
</DPCParam>

<FFCParam>
  <mode opt="schedule,temperature"/>
  <!--req,1-Continuous mode, 2-Temperature difference mode, 3-Close-->
  <ScheduleMode>
    <compensateTime opt="10,20,30,40,50,60,120,180,240"/>
    <compensateTimeUnit opt="min"/>
    <!--req,min-->
  </ScheduleMode>
  <FFCManualCtrl opt="true"/>
  <!--req,FFC Manual Control-->
  <FFCBackCompCtrl opt="true"/>
  <!--req,FFC Background Compensation control-->
</FFCParam>

<DDEParam>
  <mode opt="off,normal,expert"/>
  <!--req,1-Close, 2-Normal Mode, 3-Expert Mode-->
  <normalLevel min="1" max="100"/>
  <!--req,Level settings under normal mode-->
  <expertLevel min="1" max="100"/>
  <!--req,Level settings under expert mode-->
</DDEParam>

<AGCParam>
  <scene opt="normal,highlight,manual"/>
  <!--req,1-Normal scene, 2-Highlight scene, 3- Manual scene-->
  <ManualMode>
    <lightLevel min="1" max="100"/>
    <!--req,Brightness level-->
    <gainLevel min="1" max="100"/>
    <!--req,Gain level-->
  </ManualMode>
</AGCParam>
  <fusionMode/> <!--opt, xs:string, visual and thermal image fusion mode: "thermal"-thermal mode, "fusion"-fusion
mode, "PIP"-picture in picture mode, "Visible"-visible mode, "fusionB/W"-black and white fusion mode, "city",
"jungle", "desert", "sea", "snow"-->
  <ThermometryAGC>
    <mode opt = "close,auto,manual">
      <!--opt, xs:string-->
    </mode>
    <highTemperature min="-273" max="10000">
      <!--dep, xs:integer-->
```

```

</highTemperature>
<lowTemperature min="-273" max="10000">
  <!--dep, xs:integer-->
</lowTemperature>
</ThermometryAGC>

<isSupportGPSControl>
  <!--optional, boolean, whether the device supports GPS control capability-->
</isSupportGPSControl>

<gearRange>
  <!--optional, xs:integer, the number of ranges supported by the device, e.g., when the value is 3,it indicates
supported three ranges-->
  <gearRange>

</ChannelEntry>
</ChannelList>
</CAMERAPARA>

```

C.1.36 XML_Cap_FaceThermometry

XML message about temperature screening configuration capability

```

<FaceThermometry version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <faceThermometryEnabled opt="true,false">
    <!--required, xs:boolean, whether enables temperature screening: true=yes, false=no-->
  </faceThermometryEnabled>
  <thermometrShowEnabled opt="true,false">
    <!--required, xs:boolean, whether enables temperature OSD: true=yes, false=no-->
  </thermometrShowEnabled>
  <alarmEnabled opt="true,false">
    <!--required, xs:boolean, whether enables alarm subscription: true=yes, false=no-->
  </alarmEnabled>
  <alarmIntervalTime min="0.5" max="600" def="1">
    <!--dependent,endent, xs:float, unit:s, alarm interval time, which is valid in non-card mode, the interval is between
0.5s and 60s, corrects to one decimal place. The default time interval is 1 second-->
  </alarmIntervalTime>
  <normalizedScreenSize>
    <!--required, read-only; it is the multiples of normalized coordinates returned by device-->
    <normalizedScreenWidth>
      <!--required, read-only, xs:integer, normalized screen width-->
    </normalizedScreenWidth>
    <normalizedScreenHeight>
      <!--required, read-only, xs:integer, normalized screen height-->
    </normalizedScreenHeight>
  </normalizedScreenSize>
  <FaceThermometryRegionList size="1">
    <!--optional, temperature screening rule list-->
    <ThermometryRegion>
      <!--optional, temperature screening rule-->
      <id min="1" max="40">

```

```

    <!--required, xs:integer, rule ID-->
  </id>
  <name min="0" max="32">
    <!--optional, xs:string, rule name-->
  </name>
  <sensitivity min="1" max="5" def="3">
    <!--required, xs:integer, sensitivity, normalized value, ranges from 1 to 5, default value:3-->
  </sensitivity>
  <PupilParam>
    <!--required, pupil parameters-->
    <pupilDistance min="42" max="1000">
      <!--required, xs:integer, pupil distance, normalized value, range: [0,1000], valid range: [42,1000]-->
    </pupilDistance>
    <Region>
      <!--required, pupil region coordinates, normalized value, range: [0,1000],-->
      <RegionCoordinatesList size="">
        <!--required-->
        <RegionCoordinates>
          <!--optional, pupil region coordinates-->
          <positionX>
            <!--required, xs:integer, X-coordinate-->
          </positionX>
          <positionY>
            <!--required, xs:integer, Y-coordinate-->
          </positionY>
        </RegionCoordinates>
      </RegionCoordinatesList>
    </Region>
  </PupilParam>
  <MaxPupilParam><!--optional, the maximum pupil distance-->
    <pupilDistance min="10" max="625">
      <!--optional, xs:integer, pupil distance range, normalized value, range: [0,1000], valid range: [10,625], default
value: 625-->
    </pupilDistance>
    <Region>
      <RegionCoordinatesList size=""><!--req-->
      <RegionCoordinates><!--list, pupil region coordinates-->
        <positionX><!--required, xs:integer, coordinate, X-coordinate--></positionX>
        <positionY><!--required, xs:integer, coordinate, Y-coordinate--></positionY>
      </RegionCoordinates>
    </RegionCoordinatesList>
  </Region>
</MaxPupilParam>
<targetSpeed min="1" max="5" def="3">
  <!--required, xs:integer, target generating speed, ranges from 1 to 5, and default value is 3-->
</targetSpeed>
<alarmTemperature min="0.0" max="60.0">
  <!--optional, xs:float, alarm triggered temperature, unit:°C, ranges from -20.0 to 60.0, and corrects to one decimal
place-->
</alarmTemperature>
<type opt="region">
  <!--required, xs:string, rule type-->

```

```

</type>
<Region>
  <!--required, rule region coordinates-->
  <RegionCoordinatesList size="">
    <!--dependent-->
    <RegionCoordinates>
      <!--opt-->
      <positionX>
        <!--required, xs:integer;coordinate-->
      </positionX>
      <positionY>
        <!--required, xs:integer;coordinate-->
      </positionY>
    </RegionCoordinates>
  </RegionCoordinatesList>
</Region>
<RegionBoundary>
  <RegionCoordinatesList size="">
    <!--dependent-->
    <RegionCoordinates>
      <!--opt-->
      <positionX>
        <!--required, xs:integer;coordinate-->
      </positionX>
      <positionY>
        <!--required, xs:integer;coordinate-->
      </positionY>
    </RegionCoordinates>
  </RegionCoordinatesList>
</RegionBoundary>
<alarmRule opt="highestGreater,highestLess">
  <!--optional, xs:string, alarm rule: highestGreater-Max. temperature higher than, highestLess-Max. temperature
lower than-->
</alarmRule>
<alert min="-100.0" max="1000.0"><!--optional, xs:float, pre-alarm threshold, unit: Celsius, corrects to one
decimal place-->
</alert>
</ThermometryRegion>
</FaceThermometryRegionList>
<isSupportFaceThermDetectionInfo>
  <!--optional, xs:boolean, whether supports getting temperature screening result, corresponds to URL of /ISAPI/
Thermal/channels/<ID>/faceThermometry/regions/<ID>/detectionInfo-->
</isSupportFaceThermDetectionInfo>
<imageQuality opt="high,medium,low">
  <!--optional, xs:string-->
</imageQuality>
<mode opt="barrierPassing, targeting">
  <!--optional, xs:string, temperature screening mode: "barrierPassing"-barrier passing (transmit the custom in
airport), "targeting"-targeting (detect the person with high temperature)-->
</mode>
<faceSnapUploadEnabled opt="true,false">
  <!--optional, xs:boolean, whether to enable uploading captured face picture: true-enable, false-disable-->

```

```
</faceSnapUploadEnabled>
<maxTemperatureCoordinatesEnabled>
  <!--optional, xs:boolean, whether to enable displaying the maximum temperature position: true-enable, false-
disable-->
</maxTemperatureCoordinatesEnabled>
<faceRectShowEnabled opt="true,false">
  <!--optional, xs:boolean, whether to enable displaying a frame on the target person: true-enable, false:disable-->
</faceRectShowEnabled>
<faceTemperatureShowEnabled opt="true,false">
  <!--optional, xs:boolean, whether to enable displaying face temperature: true-enable, false:disable-->
</faceTemperatureShowEnabled>
</FaceThermometry>
```

C.1.37 XML_Cap_FireDetection

FireDetection capability message in XML format

```
<FireDetection version="2.0" xmlns="http://www.std-cgi.org/ver20/XMLSchema">
  <enabled><!--req, xs:boolean--></enabled>
  <sensitivity min="1" max="100"><!--req, xs:integer, detection sensitivity, which ranges from 1 to 10--></sensitivity>
  <fireComfirmTime min="0" max="120"><!--opt, xs:integer, ranges from 0 to 120--></fireComfirmTime>
  <fireRegionOverlay><!--opt, xs:boolean, ro, display fire frame on stream--></fireRegionOverlay>
  <fireFrameDis><!--opt, xs:boolean, ro, display fire frame--></fireFrameDis>
  <fireMaxTemp><!--opt, xs:boolean, ro--></fireMaxTemp>
  <fireMaxTempPosition><!--opt, xs:boolean, ro--></fireMaxTempPosition>
  <fireDistance><!--opt, xs:boolean, ro--></fireDistance>
  <detectionMode opt="multipleFarme,singleFarme">
    <!--opt, xs:string, fire detection mode: multipleFarme-multiple frame, singleFarme-single frame-->
  </detectionMode>
  <fireFocusMode opt="auto,cruise"><!--opt, xs:string--></fireFocusMode>
  <FireZoom>
    <zoomMode opt="auto, manual"><!--req, xs:string--></zoomMode>
    <zoomLevel min="1" max="100"><!--dep, xs:integer--></zoomLevel>
  </FireZoom>
  <AlarmStrategy>
    <strategyType opt="any,cooperate,multisystem,appointFire,appointSmoke"><!--req, xs:string--></strategyType>
  </AlarmStrategy>
  <SmokeDetection>
    <enabled><!--req, xs:boolean--></enabled>
    <sensitivity min="1" max="100"><!--opt, xs:integer, ranges from 1 to 100--></sensitivity>
    <patrolSensitivity min="1" max="100" default="50"><!--opt, xs:integer, sensitivity of patrol detection, ranges from 1
to 100--> </patrolSensitivity>
    <doubleCheckSensitivity min="1" max="100" default="50"><!--opt, xs:integer, sensitivity of double filtering, ranges
from 1 to 100--> </doubleCheckSensitivity>
    <displaySmokeInfoOnStreamEnabled opt="true,false"><!--opt, xs:boolean, overlay smoke information on stream--
></displaySmokeInfoOnStreamEnabled>
  </SmokeDetection>
  <smokeFireEnabled><!--opt, xs:boolean--></smokeFireEnabled>
  <ApplicationScene>
    <mode opt="forest-Fire_Prevention,strawBurning,high-building,Indoor/Perimeter"><!--opt, xs:string--></mode>
    <InstallationHeight min="1" max="500"><!--xs:interger, it is valid only when mode is "strawBurning", "high-
```

```

building", or "Indoor/Perimeter", ranges from 1 to 500, unit: m--></InstallationHeight>
</ApplicationScene>
<cancelRepeatedAlarmEnabled opt="true, false">
  <!--opt, xs:boolean, cancel repeated alarm, it is valid only when detectionMode is "multipleFarme"-->
</cancelRepeatedAlarmEnabled>
<fireManualWaitEnabled opt="true,false"><!--opt,xs:boolean,--></fireManualWaitEnabled>
<isSupportFireScanStart opt="true,false"><!--opt,xs:boolean, whether the fire continue scan command is supported,
related URI : /ISAPI/Thermal/channels/<ID>/fireScanStart--></isSupportFireScanStart>
<isSupportFireScanState opt="true,false"><!--opt,xs:boolean, whether the fire scan status command is supported, ,
related URI : /ISAPI/Thermal/channels/<ID>/fireScanState--></isSupportFireScanState>
<displayFireInfoOnStreamEnabled opt="true,false"><!--opt, xs:boolean, overlay fire source information on stream--
></displayFireInfoOnStreamEnabled>
<fireSourceDetection opt="dynamicFire, smokingMode"><!--opt, xs:string, fire source detection mode: dynamic fire
source, smoking--></fireSourceDetection>
<smokeAuxiliaryDetectionEnabled opt="true,false">
  <!--dep, xs:boolean, enable fire and smoke detection or not, it is valid only when detectionMode is
"multipleFarme"-->
</smokeAuxiliaryDetectionEnabled>
<verificationSensitivity min="1" max="100" default="50">
  <!--opt, xs:integer, sensitivity of double verification, ranges from 1 to 100-->
</verificationSensitivity>
<fireAlgorithmModel opt="patternRecognition, machineLearning">
  <!--opt, xs:string, fire detection algorithm mode: "patternRecognition"-pattern recognition, "machineLearning"-
machine learning-->
</fireAlgorithmModel>
<agriculturalMachineryFilterEnabled>
  <!--opt, xs:boolean, enable agricultural machinery filter-->
</agriculturalMachineryFilterEnabled>
<waterReflectionEnabled><!--opt, xs:boolean, enable water reflection--></waterReflectionEnabled>
<patrolSensitivity min="1" max="100" default="50">
  <!--opt, xs:integer, patrol sensitivity, only valid for fire detection, ranges from 1 to 100-->
</patrolSensitivity>
</FireDetection>

```

C.1.38 XML_Cap_ManualThermBasic

ManualThermBasic capability message in XML format

```

<ManualThermBasic version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id min="" max="">
    <!-- req ,xs:integer, channel No. -->
  </id>
  <distance min="0" max="10000">
    <!-- req ,xs:integer; distance, unit: m-->
  </distance>
  <emissivity min="0.01" max="1.00">
    <!-- req ,xs:float, emissivity -->
  </emissivity>
</ManualThermBasic>

```

C.1.39 XML_Cap_ManualThermometry

ManualThermometry capability message in XML format

```
<ManualThermometry version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id min="" max="">
    <!-- req ,xs:integer -->
  </id>
  <currentTime>
    <!-- req, xs:time, ISO8601 time -->
  </currentTime>
  <temperatureUnit opt="degreeCentigrade,degreeFahrenheit,degreeKelvin">
    <!-- req ,xs:string -->
  </temperatureUnit>
  <normalizedScreenSize>
    <!-- req, ro -->
    <normalizedScreenWidth>
      <!-- req, ro,xs:integer -->
    </normalizedScreenWidth>
    <normalizedScreenHeight>
      <!-- req, ro,xs:integer -->
    </normalizedScreenHeight>
  </normalizedScreenSize>
  <!-- <ThermometryRuleList size = "10"> -->
  <ThermometryRuleList>
    <ThermometryRule>
      <ruleId min="1" max="10">
        <!-- req ,xs:integer -->
      </ruleId>
      <enabled opt="true,false">
        <!-- req ,xs:boolean -->
      </enabled>
      <name min="0" max="32">
        <!-- req, xs:string -->
      </name>
      <type opt="point,region,line">
        <!-- req, xs:string"point,region,line" -->
      </type>
      <Point>
        <TempValue min="-40.0" max="1000.0">
          <!-- dep, xs:float "-40.0 .. 1000.0" ro-->
        </TempValue>
        <CalibratingCoordinates>
          <!-- dep -->
          <positionX>
            <!-- req, xs:integer;coordinate -->
          </positionX>
          <positionY>
            <!-- req, xs:integer;coordinate -->
          </positionY>
        </CalibratingCoordinates>
      </Point>
    </ThermometryRule>
  </ThermometryRuleList>
</ManualThermometry>
```

```
</Point>
<Region>
  <highestTempValue min="-40.0" max="1000.0">
    <!-- dep, xs: float ro-->
  </highestTempValue>
  <lowestTempValue min="-40.0" max="1000.0">
    <!-- dep, xs: float ro-->
  </lowestTempValue>
  <averageTempValue min="-40.0" max="1000.0">
    <!-- dep, xs: float ro-->
  </averageTempValue>
  <diffTempValue min="-40.0" max="1000.0">
    <!-- dep, xs: float ro-->
  </diffTempValue>
  <RegionCoordinatesList>
    <!-- dep -->
    <RegionCoordinates>
      <!-- opt -->
      <positionX>
        <!-- req, xs:integer;coordinate -->
      </positionX>
      <positionY>
        <!-- req, xs:integer;coordinate -->
      </positionY>
    </RegionCoordinates>
  </RegionCoordinatesList>
</Region>
<Line>
  <highestTempValue min="-40.0" max="1000.0">
    <!-- dep, xs: float ro-->
  </highestTempValue>
  <lowestTempValue min="-40.0" max="1000.0">
    <!-- dep, xs: float ro-->
  </lowestTempValue>
  <RegionCoordinatesList size="">
    <!-- dep -->
    <RegionCoordinates>
      <!-- opt -->
      <positionX>
        <!-- req, xs:integer;coordinate -->
      </positionX>
      <positionY>
        <!-- req, xs:integer;coordinate -->
      </positionY>
    </RegionCoordinates>
  </RegionCoordinatesList>
</Line>
</ThermometryRule>
</ThermometryRuleList>
</ManualThermometry>
```


C.1.40 XML_Cap_Power

Power capability message in XML format

```
<?xml version="1.0" encoding="utf-8"?>
<Power version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <powerSwitch>
    <!--req, xs:boolean, switch on/off: "true"-switch on, wake up device, "false"-switch off, device will be in sleep mode-->
  </powerSwitch>
  <batteryPower min="1" max="100">
    <!--opt, xs:integer, battery percentage-->
  </batteryPower>
</Power>
```

C.1.41 XML_Cap_ThermalBlackBody

ThermalBlackBody capability message in XML format.

```
<ThermalBlackBody version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <emissivity min="0.01" max="1" def="0.97">
    <!--required, xs:float, black body emissivity is between 0.01 and 1, which corrects to two decimal places. The default value is 0.97-->
  </emissivity>
  <distance min="0.0" max="10.0" def="2.0">
    <!--required, xs: float, the distance between lens and black body, ranges from 0.0 m to 10.0 m, which corrects to one decimal place. The default value is 2.0 m, unit:m-->
  </distance>
  <temperature min="30.0" max="50.0" def="35.0">
    <!--required, xs:float, black body temperature is between 30.0 °C and 50.0 °C, which corrects to one decimal place. The default value is 35.0 °C, unit: °C-->
  </temperature>
  <CentrePoint><!--required, center position of black body. After clicking this position, the normalized coordinate information (between 0 and 1000) will be applied-->
    <CalibratingCoordinates><!--dep-->
      <positionX><!--required, xs:integer; coordinate--></positionX>
      <positionY><!--required, xs:integer; coordinate--></positionY>
    </CalibratingCoordinates>
  </CentrePoint>
  <normalizedScreenSize><!--required, ro, read-only, it is the multiples of normalized coordinates returned by device-->
    <normalizedScreenWidth><!--required, ro,xs:integer--></normalizedScreenWidth>
    <normalizedScreenHeight><!--required, ro,xs:integer--></normalizedScreenHeight>
  </normalizedScreenSize>
  <enabled opt="true,false"><!--optional, xs:boolean, whether to enable black body--></enabled>
  <BlackBodyReigon><!--optional, black body detection area-->
    <type opt="point,region"><!--required, xs:string, "point", "region", area type: "point", "region"--></type>
    <Point><!--dependent, point coordinate, it is valid when value of type is "point"-->
      <CalibratingCoordinates><!--dependent, point coordinate-->
        <positionX><!--required, xs:integer; coordinate--></positionX>
```

```
<positionY><!--required, xs:integer; coordinate--></positionY>
</CalibratingCoordinates>
</Point>
<Region><!--dependent, frame coordinate, it is valid when value of type is "region"-->
  <RegionCoordinatesList size="4"><!--dependent, list of region coordinates-->
    <RegionCoordinates><!--list, optional-->
      <positionX><!--required, xs:integer; X-coordinate--></positionX>
      <positionY><!--required, xs:integer; Y-coordinate--></positionY>
    </RegionCoordinates>
  </RegionCoordinatesList>
</Region>
</BlackBodyReigon>
</ThermalBlackBody>
```

C.1.42 XML_Cap_ThermIntell

ThermIntell capability message in XML format.

```
<ThermIntell version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id><!--req, xs:integer--></id>
  <intellType
    opt="thermometryAndSmart,shipsDetection,fireDetection,pip,faceThermometry,thermometryAndSmokeFireDetection
,thermometryAndFireDetection,basicBehavior,thermometry" def="thermometryAndSmart">
    <!--req, xs:string,smart function resource configuration type: "thermometryAndSmart"-temperature measurement
+behavior analysis (default), "shipsDetection"-ship detection, "fireDetection"-fire source detection, "pip"-picture in
picture function, "faceThermometry"-temperature screening, "thermometryAndSmokeFireDetection"-temperature
measurement+smoke and fire source detection, "thermometryAndFireDetection"-temperature measurement+fire
source detection, "basicBehavior"-behavior analysis, "thermometry"-temperature measurement-->
  </intellType>
</ThermIntell>
```

C.1.43 XML_Cap_ThermalPip

ThermalPip capability message in XML format

```
<ThermalPip version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <!--required, capability of picture-in-picture configuration-->
  <enabled><!--required, xs:boolean, enable or not--></enabled>
  <pipMode opt="overlap,fusion, normal" def="normal">
    <!--required, xs:string, picture-in-picture mode: overlap-overlay mode, fusion-integration mode, normal-normal
mode-->
  </pipMode>
  <overlapType opt="visibleOverlapThermal,thermalOverlapVisible">
    <!--required, xs:string, overlay type: visibleOverlapThermal-overlay visible light on thermal imaging picture,
thermalOverlapVisible-overlay thermal imaging on visible light-->
  </overlapType>
  <transparency min="0" max="100">
    <!--opt, xs: integer, transparency-->
  </transparency>
```

```

<normalizedScreenSize><!--required, read-only, screen with normalized coordinates-->
  <normalizedScreenWidth><!--required, read-only, xs:integer--></normalizedScreenWidth>
  <normalizedScreenHeight><!--required, read-only, xs:integer--></normalizedScreenHeight>
</normalizedScreenSize>
<PipRegion> <!--dependent, picture-in-picture area-->
  <RegionCoordinatesList size="4"><!--dependent, coordinates list of picture-in-picture area-->
    <RegionCoordinates><!--optional, picture-in-picture area coordinate-->
      <positionX><!--required, xs:integer;x-coordinate--> </positionX>
      <positionY><!--required, xs:integer;y-coordinate--> </positionY>
    </RegionCoordinates>
  </RegionCoordinatesList>
</PipRegion>
<imageFusionRatio min="0" max="100" default="75">
  <!--dependent, xs: integer, image fusion ratio, it is valid when pipMode is "fusion"-->
</imageFusionRatio>
<borderFusionRatio min="0" max="100" default="100">
  <!--dependent, xs: integer, border fusion ratio, it is valid when pipMode is "fusion"-->
</borderFusionRatio>
<distance><!--dependent, xs:float, fusion distance, it is valid when pipMode is "fusion", ranges from 0.1 m to 4.0 m-->
</distance>
<borderEnhancementSensitivity min="0" max="100" default="80">
  <!--dependent, xs:integer, border enhancement sensitivity. The lower the sensitivity, the less details and lower noise; the higher the sensitivity, the more details and higher noise. It is valid when the value of pipMode is "fusion"-->
</borderEnhancementSensitivity>
</ThermalPip>

```

C.1.44 XML_Cap_ThermometryAlarmRule

XML message about alarm rules parameters of temperature measurement preset

```

<ThermometryAlarmRule version="2.0">
  <ThermometryAlarmModeList size="">
    <!--req, alarm rules parameters of temperature measurement preset -->
    <ThermometryAlarmMode>
      <id min="" max="">
        <!--req, xs:inter, rule ID -->
      </id>
      <enabled opt="true,false">
        <!--req, xs:boolean, enable: false-no, true-yes-->
      </enabled>
      <name min="0" max="32">
        <!--req, xs:string, ro, rule name-->
      </name>
      <pointRule opt="averageGreater,averageLess" def="averageGreater">
        <!--req, xs:string, alarm temperature comparison mode of point thermography: averageGreater-average temperature higher than (default), averageLess-average temperature lower than-->
      </pointRule>
      <lineRule opt="highestGreater,highestLess,lowestGreater,lowestLess, averageGreater,averageLess" def="averageGreater">
        <!--req, xs:string, alarm temperature comparison mode of line thermography: highestGreater-Max. temperature higher than, highestLess-Max. temperature lower than, lowestGreater-Min. temperature lower than, lowestLess-Min.

```

```

temperature lower than, averageGreater-average temperature higher than (default), averageLess-average
temperature lower than-->
  </lineRule>
  <regionRule
opt="highestGreater,highestLess,lowestGreater,lowestLess,averageGreater,averageLess,diffTempGreater,diffTempLess"
def="averageGreater">
  <!--req, xs:string, alarm temperature comparison of frame thermography: highestGreater-Max. temperature
higher than, highestLess-Max. temperature lower than, lowestGreater-Min. temperature lower than, lowestLess-Min.
temperature lower than, averageGreater-average temperature higher than (default), averageLess-average
temperature lower than, diffTempGreater-temperature difference higher than, diffTempLess-temperature difference
lower than-->
  </regionRule>
  <alert min="0" max="32">
    <!--req, xs: float, pre-alarm temperature-->
  </alert>
  <alarm min="0" max="32">
    <!--req, xs: float, alarm temperature-->
  </alarm>
  <threshold min="0" max="32">
    <!--req, xs: float, threshold temperature-->
  </threshold>
  <isSupportAlertOutputIOPortList>
    <!--opt, xs:boolean, "true", if supports, this node will be returned and the value is "true", if not support, it will not
be returned-->
  </isSupportAlertOutputIOPortList>
  <isSupportAlarmOutputIOPortList>
    <!--opt, xs:boolean, "true", if supports, this node will be returned and the vlue is "true", if not support, it will not
be returned-->
  </isSupportAlarmOutputIOPortList>
  <alertFilteringTime min="0" max="200" default="0">
    <!--opt, xs:integer, temperature pre-alarm dwell time, unit: s-->
  </alertFilteringTime>
  <alarmFilteringTime min="0" max="200" default="0">
    <!--opt, xs:integer, temperature pre-alarm dwell time, unit:s-->
  </alarmFilteringTime>
  <visibleLightLinkageEnabled opt="true, false">
    <!--opt, xs:boolean, visible light linkage-->
  </visibleLightLinkageEnabled>
  <TemperatureSuddenChange>
    <mode opt="close, temperatureSuddenIncrease, temperatureSuddenDecrease">
      <!--req, xs:string, temperature sudden change mode-->
    </mode>
    <cycle opt="1, 5, 10, 30, 60"><!--opt, xs:integer, cycle period, unit: second--></cycle>
    <alert min="" max=""><!--opt, xs: float, pre-alarm threshold--></alert>
    <alarm min="" max=""><!--opt, xs: float, alarm threshold--></alarm>
  </TemperatureSuddenChange>
</ThermometryAlarmMode>
</ThermometryAlarmModeList>
<TemperatureDifferenceComparisonList size="">
  <!--req, temperature difference configuration parameters of temperature measurement preset-->
  <TemperatureDifferenceComparison>
    <id min="" max="">

```

```

    <!--req, xs:inter, rule ID-->
</id>
<enabled opt="true,false">
    <!--req, xs:boolean, enable: false-no, true-yes-->
</enabled>
<ruleID1 min="" max="">
    <!--req, xs:string, alarm point 1-->
</ruleID1>
<ruleID2 min="" max="">
    <!--req, xs:string, alarm point 2-->
</ruleID2>
<pointRule opt="averageGreater,averageLess" def="averageGreater">
    <!--req, xs:string, alarm temperature comparison modes of point thermography: averageGreater-average
temperature higher than (default), averageLess-average temperature lower than-->
</pointRule>
<lineRule opt="highestGreater,highestLess,lowestGreater,lowestLess, averageGreater,averageLess"
def="averageGreater">
    <!--req, xs:string, alarm temperature comparison modes of line thermography: highestGreater-Max. temperature
higher than, highestLess-Max. temperature lower than, lowestGreater-Min. temperature higher than, lowestLess-Min.
temperature lower than, averageGreater-average temperature higher than (default), averageLess-average
temperature lower than -->
</lineRule>
<regionRule
opt="highestGreater,highestLess,lowestGreater,lowestLess,averageGreater,averageLess,diffTempGreater,diffTempLess"
def="averageGreater">
    <!--req, xs:string, alarm temperature comparison modes of frame thermography: highestGreater-Max.
temperature higher than, highestLess-Max. temperature lower than, lowestGreater-Min. temperature lower than,
lowestLess-Min. temperature lower than, averageGreater-average temperature higher than (Default), averageLess-
average temperature lower than, diffTempGreater-temperature difference higher than, diffTempLess-temperature
difference lower than-->
</regionRule>
<temperatureDifference min="0" max="32">
    <!--req, xs: float, temperature difference-->
</temperatureDifference>
</TemperatureDifferenceComparison>
</TemperatureDifferenceComparisonList>
</ThermometryAlarmRule>

```

C.1.45 XML_Cap_ThermometryBasicParam

XML message about capability of temperature measurement basic parameters

```

<ThermometryBasicParam version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
<id><!--req, xs:integer, channel number--></id>
<enabled opt="true,false"><!--req, xs:boolean, enable or not: false-no, true-yes--></enabled>
<streamOverlay opt="true,false"><!--req, xs:boolean, whether displays temperature information on the stream: false-
no, true-yes--></streamOverlay>
<pictureOverlay opt="true,false"><!--req, xs:boolean, whether displays temperature information on the captured
picture: false-no, true-yes--></pictureOverlay>
<temperatureRange
opt="-20-150,0-550,0-650,-4-302,32-1022,32-1200,-20-650,-20-1500,automatic,-20-120,20-350,20-45,20-350,30-45,1

```

```

00-550">
  <!--req, xs:string, temperature range-->
</temperatureRange>
<temperatureUnit opt="degreeCentigrade,degreeFahrenheit,degreeKelvin">
  <!--req, xs:string, temperature unit: degreeCentigrade-Celsius(°C), degreeFahrenheit-Fahrenheit(°F), degreeKelvin-
Kelvin(K)-->
</temperatureUnit>
<temperatureCurve opt="close,transverseTemperatureTrend,longitudinalTemperatureTrend">
  <!--opt, xs:string, temperature curve modes: close-closed, transverseTemperatureTrend-transverse temperature
mode, longitudinalTemperatureTrend-longitudinal temperature mode-->
</temperatureCurve>
<fireImageMode opt="blackWhite,thermalProbe,fireGround">
  <!--opt, xs:string, fire detection modes: blackWhite-black and white mode, thermalProbe-thermal detection mode,
fireGround-fire scene mode-->
</fireImageMode>
<emissivity min="0.01" max="1.00" default="0.96">
  <!--opt, xs:float, the emissivity is between 0.01 and 1.00, which corrects to two decimal places. This parameter is
used by the device (i.e., DS-2TF03-260V/GLT, DS-2TF03-167V/GLT) that does not support regular frame and preset-->
</emissivity>
<distanceUnit opt="meter,feet,centimeter">
  <!--req, xs:string-->
</distanceUnit>
<TemperatureColor>
  <!--opt, set the alarm information color for the temperature measurement-->
  <type opt="highTemperature,lowTemperature,rangeTemperature,heatPreservation">
    <!--req, xs:string, set temperature alarm types: highTemperature field-high temperature alarm, when the
measured temperature is higher than the configured value of highTemperature field, the measured temperature will
be marked by color; lowTemperature-low temperature alarm, when the measured temperature is lower than the
configured value of lowTemperature field, the measured temperature will be marked by color; rangeTemperature-
range temperature alarm, when the measured temperature is between the configured values of highTemperature
field and lowTemperature field, the measured temperature will be marked by color; heatPreservation-insulation
alarm, when the measured temperature is not between the configured values of highTemperature field and
lowTemperature field, the measured temperature will be marked by color-->
  </type>
  <highTemperature min="-273" max="10000"><!--dep, xs:integer--></highTemperature>
  <lowTemperature min="-273" max="10000"><!--dep, xs:integer--></lowTemperature>
</TemperatureColor>
<enviroTemperature min="-273" max="10000"><!--opt, xs:integer, environment temperature, unit: °C--></
enviroTemperature>
<enviroHumidity min="0" max="100"><!--opt, xs:integer, environment humidity, unit:%--></enviroHumidity>
<correctionVolume min="-100" max="100" def="0"><!--opt, xs:integer, temperature correction--></
correctionVolume>
<specialPointThermType opt="centerPoint,highestPoint,lowestPoint">
  <!--req, xs:string, display the special point temperature, central point temperature, highest temperature, lowest
temperature. Supports multiple selections-->
</specialPointThermType>
<distance min="0" max="50"><!--req, xs:integer; unit:m, the distance range is [0, 10000]--></distance>
<reflectiveEnable><!--req, xs:boolean, whether enables temperature reflection--></reflectiveEnable>
<reflectiveTemperature min="" max=""><!--opt, xs:float, reflective temperature, which corrects to one decimal place--
></reflectiveTemperature>
<alert min="-73.3" max="1000.0"><!--opt, xs: float, pre-alarm threshold--></alert>
<alarm min="-73.3" max="1000.0"><!--opt, xs: float, alarm threshold--></alarm>

```

```

<showTempStripEnable><!--opt, xs:boolean, whether enables displaying temperature bar--></showTempStripEnable>
<thermalOpticalTransmittance min="0.001" max="1.000" def="1.000">
  <!--opt, xs: float, optical transmissivity is between 0.001 and 1.000, which corrects to three decimal places. The
  default value is 1.000-->
</thermalOpticalTransmittance>
<externalOpticsWindowCorrection min="-40.0" max="80.0" def="20.0">
  <!--opt, xs: float, external optical temperature is between -40.0 °C and 80.0 °C. The default value is 20 °C-->
</externalOpticsWindowCorrection>
<isSupportAlertOutputIOPortList><!--opt, xs:boolean, "true", true indicates support, no return indicates not support--
></isSupportAlertOutputIOPortList>
<isSupportAlarmOutputIOPortList><!--opt, xs:boolean, "true", true indicates support, no return indicates not
support--></isSupportAlarmOutputIOPortList>
<alertFilteringTime min="0" max="200" default="0"><!--opt, xs:integer, temperature pre-alarm dwell time, unit:s--></
alertFilteringTime>
<alarmFilteringTime min="0" max="200" default="0"><!--opt, xs:integer, temperature alarm dwell time, unit:s--></
alarmFilteringTime>
<displayMaxTemperatureEnabled opt="true,false"><!--opt, xs:boolean, whether displays the maximum temperature--
></displayMaxTemperatureEnabled>
<displayMinTemperatureEnabled opt="true,false"><!--opt, xs:boolean, whether displays the minimum temperature--
></displayMinTemperatureEnabled>
<displayAverageTemperatureEnabled opt="true,false"><!--opt, xs:boolean, whether displays the average
temperature--></displayAverageTemperatureEnabled>
<thermometryInfoDisplayposition opt="rules_around,top_left_of_screen">
  <!--opt, xs:string, position of temperature measurement information overlay-->
</thermometryInfoDisplayposition>
<calibrationCoefficientEnabled opt="true,false"><!--opt, xs:boolean, whether enables calibration coefficient--></
calibrationCoefficientEnabled>
<calibrationCoefficient min="0.00" max="30.00"><!--dep, xs:float, calibration coefficient, ranges from 0 to 30,
corrects to two decimal places--></calibrationCoefficient>
<emissivityMode opt="rougher,rough,smooth,smoother,customsettings">
  <!--opt,xs:string, emissivity type: "rougher"-rougher 0.95, "rough"-rough 0.80, "smooth"-smooth 0.60, "smoother"-
smoother 0.30, "customsettings"-customized value, ranges from 0.01 to 1.00, the larger the value, the higher the
roughness-->
</emissivityMode>
<displayTemperatureInOpticalChannelEnabled opt="true,false">
  <!--opt,xs:boolean, display the temperature information of optical channel-->
</displayTemperatureInOpticalChannelEnabled>
<distanceMode opt="selfAdapt,fixed"><!--optional, xs:string, distance mode: "selfAdapt"-self-adaption, "fixed"-fixed
distance--></distanceMode>
<faceTemperatureInfoUploadEnabled opt="true,false"><!--optional, xs:boolean, whether to enable uploading face
temperature information: true-enable, false-disable--></faceTemperatureInfoUploadEnabled>
<calibrationFileVersion><!--optional, xs:string, read-only, calibration file version information--></
calibrationFileVersion>
<alarmInterval min="1" max="300"><!--optional, xs:integer, temperature measurement interval, unit: second--></
alarmInterval>
<rulesOverlayMode opt="all,alarm"><!--optional, xs:string, rule overlay mode: all (all rules), alarm (triggered alarm
rule)--></rulesOverlayMode>
<toleranceTemperature min="1" max="5"><!--optional, xs:float, tolerance temperature, value range: [1,5], unit:
Celsius--></toleranceTemperature>
<alarmMode opt="temperatureIntervalMeasurement,alarm_alert"><!--optional, xs:string, alarm mode:
"temperatureIntervalMeasurement" (temperature range measurement), "alarm_alert" (pre-alarm/alarm)--></
alarmMode>

```

```

<NormalRulesColor><!--optional, normal rule color; this node is valid when the value of alarmMode is
"temperatureIntervalMeasurement"-->
  <R><!--required, xs:integer--></R>
  <G><!--required, xs:integer--></G>
  <B><!--required, xs:integer--></B>
</NormalRulesColor>
<NormalTemperatureIntervalMeasurement><!--optional, normal temperature range measurement-->
  <alarmType opt="highestTemp,lowestTemp"><!--optional, xs:string, alarm type: "highestTemp" (the highest
temperature), "lowestTemp" (the lowest temperature)--></alarmType>
  <TemperatureIntervalList size="4"><!--optional, temperature range list; up to 4 temperature ranges are allowed-->
    <TemperatureInterval>
      <id min="1" max="4"><!--optional, xs:integer, No.--></id>
      <enabled opt="true,false"><!--optional, xs:boolean, whether to enable--></enabled>
      <name min="" max=""><!--optional, xs:string, range name--></name>
      <minTemperature min="" max=""><!--optional, xs:float, the lowest temperature; value range: [-20,550], unit:
Celsius; the value should be accurate to one decimal place--></minTemperature>
      <maxTemperature min="" max=""><!--optional, xs:float, the highest temperature; value range: [-20,550], unit:
Celsius; the value should be accurate to one decimal place--></maxTemperature>
      <AlarmColor><!--optional, alarm color of temperature range-->
        <R><!--required, xs:integer--></R>
        <G><!--required, xs:integer--></G>
        <B><!--required, xs:integer--></B>
      </AlarmColor>
      <AlarmOutputIOPortList size=""><!--optional, alarm output port list-->
        <OutputIOPort><!--list-->
          <portID min="" max=""><!--required, xs:string, port No.--></portID>
          <enabled opt="true,false"><!--required, xs:boolean, whether to enable--></enabled>
        </OutputIOPort>
      </AlarmOutputIOPortList>
    </TemperatureInterval>
  </TemperatureIntervalList>
</NormalTemperatureIntervalMeasurement>
</ThermometryBasicParam>

```

Remarks

The following nodes are not supported by the thermographic automation thermal camera (DS-2TA03-15SVI, DS2TA06-25SVI): **<TemperatureColor>**, **<specialPointThermType>**, and **<reflectiveEnable>**.

C.1.46 XML_Cap_ThermometryMode

XML message about temperature measurement mode capability

```

<ThermometryMode version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <!--req, capability sets of temperature measurement mode-->
  <mode opt="normal,expert" def="normal">
    <!--req, xs:string, temperature measurement: normal-normal mode, expert-expert mode-->
    <!--Normal mode: configure overall temperature measurement without distinguishing rule and preset. In normal
mode, the related configuration in the basic parameter configuration structure is valid. Expert mode: configure
temperature measurement configuration by rule and preset. In expert mode, the related configuration in the preset

```



```
configuration structure is valid-->
</mode>
<thermometryROIEnabled opt="true,false"><!--opt,xs:boolean, whether enables ROI temperature measurement--></
thermometryROIEnabled>
</ThermometryMode>
```

Remarks

The ROI temperature measurement and mode configuration is mutually exclusive, ROI temperature measurement is mainly applied to temperature measurement.

C.1.47 XML_Cap_ThermometryScene

ThermometryScene capability message in XML format.

```
<ThermometryScene version="2.0">
  <normalizedScreenSize>
    <!--req, ro, coordinate normalization-->
    <normalizedScreenWidth>
      <!--req, ro, xs:integer, normalized width-->
    </normalizedScreenWidth>
    <normalizedScreenHeight>
      <!--req, ro, xs:integer, normalized height-->
    </normalizedScreenHeight>
  </normalizedScreenSize>
  <ThermometryRegionList size="">
    <ThermometryRegion>
      <id min="1" max="21">
        <!--req, xs:integer, channel number-->
      </id>
      <enabled opt="true,false">
        <!--req, xs:boolean, enable or not: false-No, true-Yes-->
      </enabled>
      <name min="0" max="32">
        <!--req, xs:string, rule name-->
      </name>
      <emissivity min="0.01" max="1.00">
        <!--req, xs:float, emissivity-->
      </emissivity>
      <distance min="0" max="50">
        <!--req, xs:integer, unit:m, distance-->
      </distance>
      <reflectiveEnable>
        <!--req, xs:boolean, enable reflective temperature? false-No, true-Yes-->
      </reflectiveEnable>
      <reflectiveTemperature min="" max="">
        <!--opt, xs:float, reflective temperature-->
      </reflectiveTemperature>
      <type opt="point,region,line">
        <!--req, xs:string, rule calibration type: point-point, region-frame, line-line-->
      </type>
```

```
<Point>
  <!--dep, temperature measurement by point, display the average temperature-->
  <TempValue min="-273.0" max="10000.0">
    <!--dep, xs:float, ro-->
  </TempValue>
  <CalibratingCoordinates>
    <!--dep, point coordinates-->
    <positionX>
      <!--req, xs:integer;coordinate -->
    </positionX>
    <positionY>
      <!--req, xs:integer;coordinate -->
    </positionY>
  </CalibratingCoordinates>
</Point>
<Region>
  <!--dep, temperature measurement by frame, display the maximum, minimum, average temperature and the
temperate difference-->
  <highestTempValue min="-273.0" max="10000.0">
    <!--dep, xs: float, ro, maximum temperature-->
  </highestTempValue>
  <lowestTempValue min="-273.0" max="10000.0">
    <!--dep, xs: float, ro, minimum temperature-->
  </lowestTempValue>
  <averageTempValue min="-273.0" max="10000.0">
    <!--dep, xs: float, ro, average temperature-->
  </averageTempValue>
  <diffTempValue min="-273.0" max="10000.0">
    <!--dep, xs: float, ro, temperature difference-->
  </diffTempValue>
  <RegionCoordinatesList size="">
    <!--dep, area coordinates list of the frame-->
    <RegionCoordinates>
      <!--opt-->
      <positionX>
        <!--req, xs:integer;coordinate -->
      </positionX>
      <positionY>
        <!--req, xs:integer;coordinate -->
      </positionY>
    </RegionCoordinates>
  </RegionCoordinatesList>
</Region>
<Line>
  <!--dep, temperature measurement by line, display the maximum and minimum temperature-->
  <highestTempValue min="-273.0" max="10000.0">
    <!--dep, xs: float, ro, maximum temperature-->
  </highestTempValue>
  <lowestTempValue min="-273.0" max="10000.0">
    <!--dep, xs: float, ro, minimum temperature-->
  </lowestTempValue>
  <RegionCoordinatesList size="">
```

```

<!--dep, endpoint coordinates list of the line-->
<RegionCoordinates><!--opt-->
  <positionX><!--req, xs:integer;coordinate--></positionX>
  <positionY><!--req, xs:integer;coordinate--></positionY>
</RegionCoordinates>
</RegionCoordinatesList>
</Line>
<distanceUnit opt="meter,feet,centimeter"><!--opt, xs:string--></distanceUnit>
<emissivityMode opt="rougher,rough,smooth,smoother,customsettings">
  <!--opt,xs:string, emissivity type: "rougher"-rougher 0.95, "rough"-rough 0.80, "smooth"-smooth 0.60,
"smoother"-smoother 0.30, "customsettings"-customized value, ranges from 0.01 to 1.00, the larger the value, the
higher the roughness-->
</emissivityMode>
<RegionBoundary><!--opt, rule region boundary-->
  <RegionCoordinatesList size=""><!--dep-->
    <RegionCoordinates><!--opt-->
      <positionX><!--req, xs:integer; x-coordinate--></positionX>
      <positionY><!--req, xs:integer; y-coordinate--></positionY>
    </RegionCoordinates>
  </RegionCoordinatesList>
</RegionBoundary>
</ThermometryRegion>
<maxPointNum><!--req, xs:integer, maximum number of point rules that can be configured--></maxPointNum>
<maxLineNum><!--req, xs:integer, maximum number of line rules that can be configured--></maxLineNum>
<maxRegionNum><!--req, xs:integer, maximum number of frame rules that can be configured--></maxRegionNum>
</ThermometryRegionList>
</ThermometryScene>

```

C.1.48 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>
  </SysCap>
</DeviceCap>

```

```
</RebootConfigurationCap>
<isSupportExternalDevice><!--optional, xs:boolean--></isSupportExternalDevice>
<isSupportChangedUpload>
  <!--optional, xs:boolean, whether it supports uploading status changes-->
</isSupportChangedUpload>
<isSupportGettingWorkingStatus>
  <!--optional, xs:boolean, whether it supports getting device status-->
</isSupportGettingWorkingStatus>
<isSupportGettingChannelInfoByCondition>
  <!--optional, xs:boolean-->
</isSupportGettingChannelInfoByCondition>
<isSupportDiagnosedDataParameter>
  <!--optional, xs:boolean-->
</isSupportDiagnosedDataParameter>
<isSupportSimpleDevStatus>
  <!--optional, xs:boolean, whether it supports getting device working status-->
</isSupportSimpleDevStatus>
<isSupportFlexible>
  <!--optional, xs:boolean, whether it supports getting channel status by condition-->
</isSupportFlexible>
<isSupportPTZChannels>
  <!--optional, xs:boolean, whether it supports returning PTZ channel (which is different from the video channel)-->
</isSupportPTZChannels>
<isSupportSubscribeEvent>
  <!--optional, xs:boolean, whether it supports alarm or event subscription: "true,false"-->
</isSupportSubscribeEvent>
<isSupportDiagnosedData>
  <!--optional, xs:boolean, "true,false", whether it supports diagnosis data-->
</isSupportDiagnosedData>
<isSupportTimeCap>
  <!--optional, xs:boolean, whether it supports time capability-->
</isSupportTimeCap>
<isSupportThermalStreamData>
  <!--optional, xs:boolean, whether it supports uploading thermal stream data in real-time. If it is supported, the
returned value is "true"; otherwise, this node will not be returned-->
</isSupportThermalStreamData>
<isSupportPostUpdateFirmware>
  <!--optional,xs:boolean,"true,false", whether it supports upgrading the firmware-->
</isSupportPostUpdateFirmware>
<isSupportPostConfigData>
  <!--optional, xs:boolean,"true,false", whether it supports importing or exporting the configuration file-->
</isSupportPostConfigData>
<isSupportUserLock>
  <!--optional, xs:boolean,"true,false", whether it supports locking user-->
</isSupportUserLock>
<isSupportModuleLock><!--optional, xs:boolean, whether it supports locking the module: "true,false"--></
isSupportModuleLock>
<isSupportSoundCfg><!--optional, xs:boolean--></isSupportSoundCfg>
<isSupportMetadata>
  <!--optional, xs:boolean, if it is supported, return "true", otherwise, this node will not be returned-->
</isSupportMetadata>
<isSupportShutdown><!--optional, xs:boolean, whether it supports shutdown configuration--></
```

```
isSupportShutdown>
  <supportSmartOverlapChannles opt="1"/><!--optional, xs:boolean, whether it supports stream configuration of
smart events. If this function is supported, this node and the corresponding channel ID will be returned; otherwise,
this node will not be returned-->
  <isSupportConsumptionMode><!--optional, xs:boolean, whether it supports switching power consumption
mode:true (yes), this node is not returned (no). Related URI: /ISAPI/System/consumptionMode/capabilities?
format=json--></isSupportConsumptionMode>
  <isSupportManualPowerConsumption><!--optional, xs:boolean, whether it supports control the power
consumption mode manually: true (yes), this node is not returned (no)--></isSupportManualPowerConsumption>
</SysCap>
<voicetalkNums><!--optional, xs:integer, the number of two-way audio channels--></voicetalkNums>
<isSupportSnapshot><!--optional, xs:boolean, whether it supports capture: "true, false"--></isSupportSnapshot>
<SecurityCap/><!--optional, security capability-->
<EventCap/><!--optional, event capability-->
<ITCCap><!--optional--></ITCCap>
<ImageCap/><!--optional, image capability-->
<RacmCap/><!--optional, storage capability-->
<PTZCtrlCap>
  <isSupportPatrols><!--optional, xs:boolean--></isSupportPatrols>
</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-->
<WLAAlarmCap/><!--optional, wireless alarm capability-->
<SecurityCPCapabilities/><!--optional, security control panel capability-->
<isSupportGIS>
  <!--optional, xs:boolean, whether it supports GIS capability-->
</isSupportGIS>
<isSupportCompass>
  <!--optional, xs:boolean-->
</isSupportCompass>
<isSupportRoadInfoOverlays>
  <!--optional, xs:boolean-->
</isSupportRoadInfoOverlays>
<isSupportFaceCaptureStatistics>
  <!--optional, xs:boolean-->
</isSupportFaceCaptureStatistics>
<isSupportExternalDevice>
  <!--optional, xs:boolean-->
</isSupportExternalDevice>
<isSupportElectronicsEnlarge>
  <!--optional, xs:boolean, whether it supports digital zoom-->
</isSupportElectronicsEnlarge>
<isSupportRemoveStorage>
  <!--optional, xs:boolean-->
</isSupportRemoveStorage>
<isSupportCloud>
  <!--optional, xs:boolean-->
```

```
</isSupportCloud>
<isSupportRecordHost>
  <!--optional, xs:boolean-->
</isSupportRecordHost>
<isSupportEagleEye>
  <!--optional, xs:boolean, whether it supports PanoVu series camera-->
</isSupportEagleEye>
<isSupportPanorama>
  <!--optional, xs:boolean, whether it supports panorama-->
</isSupportPanorama>
<isSupportFirmwareVersionInfo>
  <!--optional, xs:boolean, whether it supports displaying firmware version information-->
</isSupportFirmwareVersionInfo>
<isSupportExternalWirelessServer>
  <!--optional, xs: boolean-->
</isSupportExternalWirelessServer>
<isSupportSetupCalibration>
  <!--optional, xs:boolean, whether it supports setting calibration-->
</isSupportSetupCalibration>
<isSupportGetmutexFuncErrMsg>
  <!--optional, xs:boolean, whether it supports getting mutex information-->
</isSupportGetmutexFuncErrMsg>
<isSupportTokenAuthenticate><!--optional, xs:boolean--></isSupportTokenAuthenticate>
<isSupportStreamDualVCA><!--optional, xs:boolean--></isSupportStreamDualVCA>
<isSupportlaserSpotManual>
  <!--optional, boolean, whether it supports laser spot configuration-->
</isSupportlaserSpotManual>
<isSupportRTMP><!--optional, xs:boolean--></isSupportRTMP>
<isSupportTraffic><!--optional, xs:boolean--></isSupportTraffic>
<isSupportLaserSpotAdjustment>
  <!--optional, boolean, whether it supports adjusting laser spot size-->
</isSupportLaserSpotAdjustment>
<VideoIntercomCap/><!--optional, video intercom capability-->
<isSupportSafetyCabin>
  <!--optional, xs:boolean-->
</isSupportSafetyCabin>
<isSupportPEA>
  <!--optional, xs:boolean, whether it supports one-touch security control panel capability-->
</isSupportPEA>
<isSupportCurrentLock>
  <!--optional, xs:boolean, whether it supports locking current configuration-->
</isSupportCurrentLock>
<isSupportGuardAgainstTheft>
  <!--optional, xs:boolean, whether it supports device anti-theft configuration-->
</isSupportGuardAgainstTheft>
<isSupportPicInfoOverlap>
  <!--optional, xs:boolean, whether it supports picture information overlay-->
</isSupportPicInfoOverlap>
<isSupportPlay>
  <!--optional, xs: boolean, whether it supports live view: "true,false"-->
</isSupportPlay>
<isSupportPlayback>
```

```
<!--optional, xs: boolean, whether it supports playback: "true,false"-->
</isSupportPlayback>
<UHF RFIDReader>
  <!--optional, supported capability of UHF RFID card reader-->
  <isSupportBasicInformation>
    <!--optional, xs:boolean, whether it supports basic parameters of UHF RFID card reader-->
  </isSupportBasicInformation>
  <isSupportHardDiskStorageTest>
    <!--optional, xs:boolean, whether it supports hard disk storage test of UHF RFID card reader-->
  </isSupportHardDiskStorageTest>
</UHF RFIDReader>
<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>
<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,not
SlowZebraCrossing,overtakeRightSide,lowSpeed,dragRacing,changeLaneContinuously,SSharpDriving,largeVehicleOccup
yLine,jamCrossLine">
  <!--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,notSupport"/><!--
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
<isSupportEncryption> 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 slave access control devices or peripheral modules: "true"-yes, this
node is not returned-no-->
</isSupportAcsUpdate>
<isSupportAccessControlCap>
  <!--optional, whether it supports access control capability: "true"-yes, this node is not returned-no-->
</isSupportAccessControlCap>
<isSupportIDCardInfoEvent><!--optional, whether it supports ID card swiping event: "true"-yes. This node will not be
returned if this function is not supported--></isSupportIDCardInfoEvent>
<OpenPlatformCap><!--optional, embedded open platform capability, refer to the message XML_OpenPlatformCap
for details-->
<isSupportInstallationAngleCalibration>
  <!--optional, xs:boolean, whether it supports installation angle calibration-->
</isSupportInstallationAngleCalibration>
```

```

<isSupportZeroBiasCalibration>
  <!--optional, xs:boolean, whether it supports zero bias calibration-->
</isSupportZeroBiasCalibration>
<isSupportDevStatus><!--optional, xs:boolean, whether device supports getting device status--></
isSupportDevStatus>
  <isSupportRadar><!--optional, xs:boolean, whether it supports the security radar--></isSupportRadar>
  <isSupportRadarChannels><!--optional, xs:boolean, whether it supports getting radar channels--></
isSupportRadarChannels>
  <radarIPDForm><!--optional, xs:string, radar form: "single"-single radar, "double_diagonal"-two radars forming an
180° diagonal, "double_vertical"-two radars forming a 90° vertical angle--></radarIPDForm>
  <isSupportRadarFieldDetection><!--optional, xs:boolean, whether it supports intrusion detection (radar)--></
isSupportRadarFieldDetection>
  <isSupportRadarLineDetection><!--optional, xs:boolean, whether it supports line crossing detection (radar)--></
isSupportRadarLineDetection>
  <mixedTargetDetectionWebNoDisplay><!--optional, xs:boolean, whether to enable not displaying multi-target-type
recognition--></mixedTargetDetectionWebNoDisplay>
  <SHMCap><!--opt-->
    <isSupportHighHDDTemperature><!--optional, xs:boolean, whether it supports HDD high temperature detection--></
isSupportHighHDDTemperature>
    <isSupportLowHDDTemperature><!--optional, xs:boolean, whether it supports HDD low temperature detection--></
isSupportLowHDDTemperature>
    <isSupportHDImpact><!--optional, xs:boolean, whether it supports HDD impact detection--></isSupportHDImpact>
    <isSupportHDBadBlock><!--optional, xs:boolean, whether it supports HDD bad sector detection--></
isSupportHDBadBlock>
    <isSupportSevereHDFailure><!--optional, xs:boolean, whether it supports HDD severe fault detection--></
isSupportSevereHDFailure>
  </SHMCap>
  <isSupportBVCorrect><!--optional, xs:boolean, whether it supports configuring camera correction parameters--></
isSupportBVCorrect>
  <guideEventSupport opt="linkageCapture">
    <!--optional,xs:string, events which support quick setup by instruction, "linkageCapture"-capture by linkage-->
  </guideEventSupport>
  <isSupportAutoSwitch><!--optional, xs:boolean, whether it supports auto switch--> true</isSupportAutoSwitch>
  <isSupportDataPrealarm><!--optional,xs:boolean, whether it supports traffic pre-alarm event--></
isSupportDataPrealarm>
  <supportGISEvent opt="AID,TPS,ANPR,mixedTargetDetection">
    <!--optional, xs:string, event types that support GIS information access: AID (corresponding SDK event:
COMM_ALARM_AID_V41), TPS (corresponding SDK event: COMM_ALARM_TPS_REAL_TIME), ANPR (corresponding
SDK event: COMM_ITS_PLATE_RESULT), mixedTargetDetection-mixed targets detection-->
  </supportGISEvent>
  <isSupportIntelligentMode><!--optional, xs:boolean, whether it supports intelligent scene switch (related URI:/ISAPI/
System/IntelligentSceneSwitch?format=json)--></isSupportIntelligentMode>
  <isSupportCertificateCaptureEvent><!--optional, xs:boolean, whether it supports certificate capture and comparison
events: true=yes. If this function is not supported, this node will not be returned--></
isSupportCertificateCaptureEvent>
  <isSupportAlgorithmsInfo><!--optional, xs:boolean, whether it supports getting the algorithm library version
information: true=yes. If this function is not supported, this node will not be returned--></isSupportAlgorithmsInfo>
  <isSupportVibrationDetection><!--optional, xs:boolean, whether it supports vibration detection--></
isSupportVibrationDetection>
  <isSupportFaceTemperatureMeasurementEvent><!--optional, xs:boolean, whether it supports uploading face
thermography events (eventType: "FaceTemperatureMeasurementEvent")--></
isSupportFaceTemperatureMeasurementEvent>

```

```

<isSupportQRCodeEvent><!--optional, xs:boolean, whether it supports uploading QR code events (eventType:
"QRCodeEvent")--></isSupportQRCodeEvent>
<isSupportPersonArmingTrack><!--optional, xs:boolean, whether device supports person arming (related URI: /ISAPI/
Intelligent/channels/<ID>/personArmingTrack/capabilities?format=json)--></isSupportPersonArmingTrack>
<isSupportManualPersonArmingTrack><!--optional, xs:boolean, whether device supports manual person arming
(related URI: /ISAPI/Intelligent/channels/<ID>/manualPersonArmingTrack?format=json)--></
isSupportManualPersonArmingTrack>
<isSupportGPSCalibrationMode><!--optional, xs:boolean, whether device supports GPS calibration (related URI: /
ISAPI/System/GPSCalibration/channels/<ID>/mode?format=json)--></isSupportGPSCalibrationMode>
<isSupportGPSVerification><!--optional, xs:boolean, whether device supports GPS verification (related URI: /ISAPI/
System/GPSVerification/channels/<ID>/points?format=json)--></isSupportGPSVerification>
<isSupportHBDLib><!--optional, xs:boolean, whether device supports human body picture library (related URI: /ISAPI/
Intelligent/HBDLib/capabilities?format=json)--></isSupportHBDLib>
<isSupportFireEscapeDetection><!--optional, xs:boolean, whether the device supports fire engine access detection
(related URI: /ISAPI/Intelligent/channels/<ID>/fireEscapeDetection/capabilities?format=json)--></
isSupportFireEscapeDetection>
<isSupportTakingElevatorDetection><!--optional, xs:boolean, whether the device supports elevator detection
(related URI: /ISAPI/Intelligent/channels/<ID>/takingElevatorDetection/capabilities?format=json)--></
isSupportTakingElevatorDetection>
<isSupportSSDFileSystemUpgrade><!--optional, xs:boolean, whether the device supports SSD file system upgrade
(related URI: /ISAPI/System/SSDFileSystem/upgrade?format=json)--></isSupportSSDFileSystemUpgrade>
<isSupportSSDFileSystemFormat><!--optional, xs:boolean, whether the device supports SSD file system formatting
(related URI: /ISAPI/System/SSDFileSystem/format?format=json)--></isSupportSSDFileSystemFormat>
<isSupportSSDFileSystemCapacity><!--optional, xs:boolean, whether the device supports getting space distribution
information of SSD file system (related URI: /ISAPI/System/SSDFileSystem/capacity?format=json)--></
isSupportSSDFileSystemCapacity>
<isSupportAIOpenPlatform><!--optional, xs:boolean, whether the device supports AI open platform capabilities; if
supports, this node will be returned and its value is true; if not, this node will not be returned--></
isSupportAIOpenPlatform>
<isSupportPictureDownloadError><!--optional, xs:boolean, whether the device supports reporting picture download
failure--></isSupportPictureDownloadError>
<characteristicCode min="1" max="128"><!--optional, xs:string, device attribute code (related URI: /ISAPI/System/
deviceInfo/characteristicCode?format=json)--></characteristicCode>
</DeviceCap>

```

C.1.49 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.1.50 XML_EventNotificationAlert_SubscriptionHeartbeat

Heartbeat information message returned when subscribing alarm/event, and it is in XML format.

```
<EventNotificationAlert version="2.0" xmlns="http://www.isapi.com/ver20/XMLSchema">
  <ipAddress>10.17.133.46</ipAddress>
  <portNo>80</portNo>
  <protocol>HTTP</protocol>
  <macAddress>44:19:b6:6d:24:85</macAddress>
  <channelID>1</channelID>
  <dateTime>2017-05-04T11:20:02+08:00</dateTime>
  <activePostCount>0</activePostCount>
  <eventType>heartBeat</eventType>
  <eventState>active</eventState>
  <eventDescription>heartBeat</eventDescription>
</EventNotificationAlert>
```

See Also

XML_EventNotificationAlert_AlarmEventInfo

C.1.51 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

See Also

XML_EventTriggerNotificationList

C.1.52 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,siren,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.1.53 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.1.54 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>
<PIRAAlarmCap><!--optional, xs: EventTriggerCapType--></PIRAAlarmCap>
<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>
<SevereHDFailureTriggerCap><!--optional, xs:EventTriggerCapType, HDD severe fault detection--></
SevereHDFailureTriggerCap>
<HUMANATTRIBUTECap><!--optional, xs:EventTriggerCapType--></HUMANATTRIBUTECap>
<HumanAttributeTriggerCap><!--optional, xs:EventTriggerCapType, human body attribute--></
HumanAttributeTriggerCap>
<BlackListFaceContrastTriggerCap><!--opt, xs:EventTriggerCapType, alarm linkage capability of blocklist 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.1.55 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 linkage 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 surveillance
center is not supported: "true"-yes (configuration parameters of the surveillance center cannot be edited), "false" or
this node is not returned-no (configuration parameters of the surveillance 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.1.56 XML_FaceThermDetectionInfo

XML message about temperature screening result.

```

<FaceThermDetectionInfo version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id><!--req, xs:integer, rule ID--> </id>
  <name><!--req, xs:string, rule name--> </name>
  <time><!--req, xs:ISO8601_time, detection time--></time>
  <faceDetectionState>
    <!--req, xs:boolean, face detection status: false-no face detected, true-face detected-->
  </faceDetectionState>
  <alarmTemperature>
    <!--opt, xs:float, alarm triggered temperature, ranges from 20.0°C to 150.0°C-->
  </alarmTemperature>
  <visibleLightImageLen><!--opt, xs:integer, the size of visible light image binary data--></visibleLightImageLen>

```

```
<faceImageLen>
  <!--opt, xs:integer, size of face thumbnail binary data, that is the size of Opaque Data(visibleLightImage), if no
Opaque Data(visibleLightImage), it values 0-->
</faceImageLen>
<FaceRect>
  <!--opt, face thumbnail coordinates-->
  <height><!--req, xs:float--></height>
  <width><!--req, xs:float--></width>
  <x><!--req, xs:float--></x>
  <y><!--req, xs:float--></y>
</FaceRect>
<minTemperature> <!--opt, xs:float, -20.0~150.0°C--> </minTemperature>
<ruleTemperature> <!--opt, xs:float, 0.0°C~60.0°C--> </ruleTemperature>
<averageTemperature> <!--opt, xs:float, -20.0~150.0°C--> </averageTemperature>
<MinTemperaturePoint>
  <!--opt, the coordinates of lowest temperature position-->
  <x><!--req, xs:float--></x>
  <y><!--req, xs:float--></y>
</MinTemperaturePoint>
<MaxTemperaturePoint>
  <!--opt, the coordinates of highest temperature position-->
  <x><!--req, xs:float--></x>
  <y><!--req, xs:float--></y>
</MaxTemperaturePoint>
<alarmRule>
  <!--opt, xs:string, alarm rule: highestGreater-max. temperature higher than, highestLess-min. temperature lower
than-->
</alarmRule>
<highTemperatureTargetImageLen><!--opt, xs:integer--></ highTemperatureTargetImageLen>
<HighTemperatureTargetRect>
  <!--opt, the coordiantes of high object thumbnail-->
  <height><!--req, xs:float--></height>
  <width><!--req, xs:float--></width>
  <x><!--req, xs:float--></x>
  <y><!--req, xs:float--></y>
</HighTemperatureTargetRect>
</FaceThermDetectionInfo>
```

Example

Transmit Result Information in Form Format

```
Accept: text/html, application/xhtml+xml, */*
Accept-Language: en-US
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
Accept-Encoding: gzip, deflate
Host: 10.17.133.46
DNT: 1
Connection: Keep-Alive
Cookie: language=zh; sdMarkMenu=8%3Avehicle; sdMarkTab_1_0=0%3AsettingBasic;
sdMarkTab_6_0=5%3AeventException; sdMarkTab_6_1=6%3AsmartLoiterDetection;
sdMarkTab_7_0=1%3AplanCapture; sdMarkTab_7_1=0%3AstorageManageHarddisk;
sdMarkTab_8=0%3AvehicleParam; WebSession=f81610c130711300cf30
```

```
HTTP/1.1 200 OK
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary=boundary

--boundary
Content-Type: application/xml; charset="UTF-8"
Content-Length: 480

<FaceThermDetectionInfo>
--boundary
Content-Disposition: form-data;
Content-Type: image/jpeg
Content-Length: 480

Opaque Data(visibleLightImage)
--boundary
Content-Disposition: form-data;
Content-Type: image/jpeg
Content-Length: 480

Opaque Data(facelImage)
--boundary
Content-Disposition: form-data;
Content-Type: image/jpeg
Content-Length: 480

Opaque Data(highTemperatureTargetImage)// High temperature object thumbnail
--boundary--
```

C.1.57 XML_FaceThermometry

XML message about temperature screening parameters

```
<FaceThermometry version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id><!--required, xs:integer, channel No.--></id>
  <faceThermometryEnabled><!--required, xs:boolean, whether enables temperature screening: true=yes, false=no--></faceThermometryEnabled>
  <thermometrShowEnabled><!--required, xs:boolean, whether enables temperature OSD: true=yes, false=no--></thermometrShowEnabled>
  <alarmEnabled><!--required, xs:bool, enable alarm subscription, opt="true,false"--></alarmEnabled>
  <alarmIntervalTime min="0.5" max="600" def="1"><!--dep, xs:float, alarm interval time, which is valid in non-card mode, the interval is between 0.5 s and 60 s. The default time interval is 1s, unit:s--></alarmIntervalTime>
  <normalizedScreenSize><!--required, read-only, read-only, it is the multiples of normalized coordinates returned by device-->
    <normalizedScreenWidth><!--required, read-only, xs:integer--></normalizedScreenWidth>
    <normalizedScreenHeight><!--required, read-only, xs:integer--></normalizedScreenHeight>
  </normalizedScreenSize>
  <FaceThermometryRegionList/><!--optional, temperature screening rule list-->
  <imageQuality><!--optional, xs:string, image quality, including high, medium, and low, "high,medium,low"--></imageQuality>
```

```
<mode><!--optional, xs:string, temperature screening mode: "barrierPassing"-barrier passing (transmit the custom in
airport), "targeting"-targeting (detect the person with high temperature)--></mode>
<faceSnapUploadEnabled><!--optional, xs:boolean, whether to enable uploading captured face picture: true-enable,
false-disable--></faceSnapUploadEnabled>
<maxTemperatureCoordinatesEnabled><!--optional, xs:boolean, whether to enable displaying the maximum
temperature position: true-enable, false-disable--></maxTemperatureCoordinatesEnabled>
<faceRectShowEnabled><!--optional, xs:boolean, whether to enable displaying a frame on the target person: true-
enable, false:disable--></faceRectShowEnabled>
<faceTemperatureShowEnabled><!--optional, xs:boolean, whether to enable displaying face temperature: true-
enable, false:disable--></faceTemperatureShowEnabled>
</FaceThermometry>
```

C.1.58 XML_FaceThermometryRegionList

XML message about temperature screening rule list

```
<FaceThermometryRegionList version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <ThermometryRegion/><!--opt, single rule information of temperature screening-->
</FaceThermometryRegionList>
```

C.1.59 XML_FireDetection

FireDetection message in XML format

```
<FireDetection version="2.0" xmlns="http://www.std-cgi.org/ver20/XMLSchema">
  <enabled><!--req, xs:boolean--></enabled>
  <sensitivity><!--req, xs:integer, ranges from 1 to 100--></sensitivity>
  <fireComfirmTime><!--opt, xs:integer, ranges from 0 to 120--></fireComfirmTime>
  <fireRegionOverlay><!--opt, xs:boolean--></fireRegionOverlay>
  <detectionMode><!--opt, xs:string, "multipleFarme, singleFarme"--></detectionMode>
  <fireFocusMode><!--opt, xs:string, "auto, cruise"--></fireFocusMode>
  <FireZoom>
    <zoomMode><!--req, xs:string, "auto, manual"--></zoomMode>
    <zoomLevel><!--dep, xs:integer, ranges from 1 to 100--></zoomLevel>
  </FireZoom>
  <AlarmStrategy>
    <strategyType><!--req, xs:string, "any, cooperate, multisystem, appointFire, appointSmoke"--></strategyType>
    <alarmType><!--dep, xs:string, "fire, smoke"--></strategyType>
  </AlarmStrategy>
  <SmokeDetection>
    <enabled><!--req, xs:boolean--></enabled>
    <sensitivity><!--opt, xs:integer, ranges from 1 to 100--></sensitivity>
    <patrolSensitivity><!--opt, xs:integer, sensitivity of patrol detection, ranges from 1 to 100--></patrolSensitivity>
    <doubleCheckSensitivity><!--opt, xs:integer, sensitivity of double filtering, ranges from 1 to 100--></
doubleCheckSensitivity>
    <displaySmokeInfoOnStreamEnabled><!--opt, xs:boolean, overlay smoke information on stream--></
displaySmokeInfoOnStreamEnabled>
  </SmokeDetection>
  <smokeFireEnabled><!--req, xs:boolean--></smokeFireEnabled>
```

```

<ApplicationScene>
  <mode><!--opt, xs:string. "forest-Fire_Prevention, strawBurning, high-building, Indoor/Perimeter"--></mode>
  <InstallationHeight><!--xs:integer, it is valid only when mode is "strawBurning", "high-building", or "Indoor/
Perimeter", ranges from 1 to 500, unit: m--></InstallationHeight>
</ApplicationScene>
<cancelRepeatedAlarmEnabled>
  <!--opt, xs:boolean, cancel repeated alarm, it is valid only when detectionMode is "multipleFarme"-->
</cancelRepeatedAlarmEnabled>
<displayFireInfoOnStreamEnabled><!--opt, xs:boolean, overlay fire source information on stream--></
displayFireInfoOnStreamEnabled>
<smokeAuxiliaryDetectionEnabled>
  <!--dep, xs:boolean, enable fire and smoke detection or not, it is valid only when detectionMode is
"multipleFarme"-->
</smokeAuxiliaryDetectionEnabled>
<verificationSensitivity>
  <!--opt, xs:integer, sensitivity of double verification, ranges from 1 to 100-->
</verificationSensitivity>
<fireAlgorithmMode>
  <!--opt, xs:string, fire detection algorithm mode: "patternRecognition"-pattern recognition, "machineLearning"-
machine learning-->
</fireAlgorithmMode>
<agriculturalMachineryFilterEnabled>
  <!--opt, xs:boolean, enable agricultural machinery filter-->
</agriculturalMachineryFilterEnabled>
<waterReflectionEnabled><!--opt, xs:boolean, enable water reflection--></waterReflectionEnabled>
<patrolSensitivity>
  <!--opt, xs:integer, patrol sensitivity, only valid for fire detection, ranges from 1 to 100-->
</patrolSensitivity>
<fireManualWaitEnabled opt="true,false"><!--opt,xs:boolean,--></fireManualWaitEnabled>
<isSupportFireScanStart opt="true,false"><!--opt,xs:boolean, whether the fire continue scan command is supported,
related URI : /ISAPI/Thermal/channels/<ID>/fireScanStart--></isSupportFireScanStart>
<isSupportFireScanState opt="true,false"><!--opt,xs:boolean, whether the fire scan status command is supported, ,
related URI : /ISAPI/Thermal/channels/<ID>/fireScanState--></isSupportFireScanState>
</FireDetection>

```

C.1.60 XML_HistoryTemperatureCap

HistoryTemperatureCap message in XML format

```

<HistoryTemperatureCap version="2.0" xmlns="http://www.hikvision.com/ver20/XMLSchema">
  <id min="1" max="40">
    <!--req, xs:integer, preset ID-->
  </id>
  <thermometryUnit opt = "celsius,fahrenheit,kelvin">
    <!--req, temperature measurement unit, xs:string:"celsius,fahrenheit,kelvin"-->
  </thermometryUnit>
  <ruleId min = "1" max = "10">
    <!--req, xs:integer -->
  </ruleId>
  <tempValue min="-40.0" max="1000.0">
    <!--req, xs: float ro-->
  </tempValue>
</HistoryTemperatureCap>

```

```
</tempValue>
</HistoryTemperatureCap>
```

C.1.61 XML_HistoryTemperatureDescription

HistoryTemperatureDescription message in XML format

```
<HistoryTemperatureDescription version="2.0" xmlns="http://www.hikvision.com/ver20/XMLSchema">
  <id><!--req, xs:integer, preset ID--></id>
  <startTime><!--req,xs:time, ISO8601 time--></startTime>
  <RuleIdList><!--req-->
    <ruleId><!-- req ,xs:integer--></ruleId>
  </RuleIdList>
</HistoryTemperatureDescription>
```

C.1.62 XML_HistoryTemperatureResult

HistoryTemperatureResult message in XML format

```
<HistoryTemperatureResult version="2.0" xmlns="http://www.hikvision.com/ver20/XMLSchema">
  <thermometryUnit>
    <!--req,xs:string,"celsius,fahrenheit,kelvin"-->
  </thermometryUnit>
  <RuleInfoList><!--req-->
    <RuleInfo>
      <ruleId><!--req ,xs:integer--></ruleId>
      <TempInfoList>
        <TempInfo>
          <time><!--req,xs:time, ISO8601 time--></time>
          <tempValue>
            <!--dep, xs:float, "-40.0 .. 1000.0", corrects to one decimal place, ro-->
          </tempValue>
        </TempInfo>
      </TempInfoList>
    </RuleInfo>
  </RuleInfoList>
</HistoryTemperatureResult>
```

C.1.63 XML_PixelToPixelParam

XML message about parameters of pixel-to-pixel temperature measurement

```
<PixelToPixelParam version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id><!--req, xs:integer, channel No.--></id>
  <maxFrameRate>
    <!--opt, xs: integer, maximum frame rate-->
  </maxFrameRate>
```



```

<reflectiveEnable>
  <!--opt, xs: boolean, enable reflection temperature or not-->
</reflectiveEnable>
<reflectiveTemperature>
  <!--opt, xs: float, reflection temperature-->
</reflectiveTemperature>
<emissivity>
  <!--opt, xs: float, emissivity-->
</emissivity>
<distance>
  <!--req, xs: integer, unit: m, distance-->
</distance>
<refreshInterval>
  <!--opt, xs: integer, the refresh interval of pixel-to-pixel temperature measurement-->
</refreshInterval>
<distanceUnit>
  <!--opt, xs:string, distance unit, opt="meter,feet,centimeter"-->
</distanceUnit>
<temperatureDataLength>
  <!--opt, xs:integer, temperature data length-->
</temperatureDataLength>
<JpegPictureWithAppendData>
  <!--opt, configuration of getting JPEG picture with pixel-to-pixel temperature measurement data-->
  <jpegPicEnabled><!--opt, xs:boolean, whether returns JPEG picture--></jpegPicEnabled>
  <visiblePicEnabled><!--optional, xs:boolean, whether the thermal camera returns visible light picture--></
visiblePicEnabled>
  <rulesOverlayEnabled><!--optional, xs:boolean, whether to enable thermography rule overlay on picture--></
rulesOverlayEnabled>
  <visiblePicResolution><!--optional, xs:boolean, visible light picture resolution: 600*800, 1200*1600--></
visiblePicResolution>
  <thermalPicResolution><!--optional, xs:boolean, thermal picture resolution: 160*120, 640*480--></
thermalPicResolution>
</JpegPictureWithAppendData>
</PixelToPixelParam>

```

C.1.64 XML_PixelToPixelParamCap

XML message about capability of pixel-to-pixel temperature measurement

```

<PixelToPixelParamCap version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id><!--required, xs:integer, channel No.--></id>
  <maxFrameRate opt="">
    <!--optional, xs:integer, maximum frame rate-->
  </maxFrameRate>
  <reflectiveEnable opt="true false">
    <!--optional, xs:boolean, enable reflection temperature or not-->
  </reflectiveEnable>
  <reflectiveTemperature>
    <!--optional, xs:float, reflection temperature-->
  </reflectiveTemperature>
  <emissivity min="0.01" max="1.00">

```

```
<!--optional, xs:float, emissivity-->
</emissivity>
<distance min="0" max="50">
  <!--required, xs:integer, unit: m, distance-->
</distance>
<refreshInterval min="1" max="500">
  <!--optional, xs:integer, the refresh interval of pixel-to-pixel temperature measurement-->
</refreshInterval>
<distanceUnit opt="meter,feet,centimeter">
  <!--optional, xs:string, distance unit-->
</distanceUnit>
<temperatureDataLength opt="2,4">
  <!--optional, xs:integer, temperature data length-->
</temperatureDataLength>
<JpegPictureWithAppendData>
  <!--optional, configuration of getting JPEG picture with pixel-to-pixel temperature measurement data-->
  <jpegPicEnabled opt="true,false"><!--optional, xs:boolean, whether returns JPEG picture--></jpegPicEnabled>
  <visiblePicEnabled opt="true,false"><!--optional, xs:boolean, whether the thermal camera returns visible light
picture--></visiblePicEnabled>
  <captureMode opt="standard"><!--optional, xs:string, image capture mode: standard--></captureMode>
  <rulesOverlayEnabled opt="true,false"><!--optional, xs:boolean, whether to enable thermography rule overlay on
picture--></rulesOverlayEnabled>
  <visiblePicResolution opt="600*800,1200*1600"><!--optional, xs:boolean, visible light picture resolution: 600*800,
1200*1600--></visiblePicResolution>
  <thermalPicResolution opt="160*120,640*480"><!--optional, xs:boolean, thermal picture resolution: 160*120,
640*480--></thermalPicResolution>
</JpegPictureWithAppendData>
</PixelToPixelParamCap>
```

C.1.65 XML_Power

Power message in XML format

```
<?xml version="1.0" encoding="utf-8"?>
<Power version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <powerSwitch>
    <!--req, xs:boolean, switch on/off: "true"-switch on, wake up device, "false"-switch off, device will be in sleep mode-->
  >
  </powerSwitch>
  <batteryPower><!--opt, ro, xs:integer, battery percentage, range: [1,100]--></batteryPower>
</Power>
```

C.1.66 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.--></outputIOPortID>
<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, ISO8601 time--></beginTime>
      <endTime><!--required, xs:time, ISO8601 time--></endTime>
    </TimeRange>
    <CustomExtension>
      <vehicleDetectSceneID>
        <!--required, xs:interger-->
      </vehicleDetectSceneID>
    </CustomExtension>
  </TimeBlock>
</TimeBlockList>
<HolidayBlockList><!--optional-->
  <TimeBlock><!--list-->
    <TimeRange><!--required-->
      <beginTime><!--required, xs:time, ISO8601 time--></beginTime>
      <endTime><!--required, xs:time, ISO8601 time--></endTime>
    </TimeRange>
  </TimeBlock>
</HolidayBlockList>
</Schedule>

```

Remarks

The node **<eventType>** can be set to the following values: IO, VMD, videoloss, PIR, linedetection, fielddetection, 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.

C.1.67 XML_SubscribeEventResponse

SubscribeEventResponse message in XML format

```

<SubscribeEventResponse>
  <id><!--req, xs:integer, subscription ID--></id>
  <FailedEventList>
    <!--opt, list of subscription failed events. When subscription failed, it should be returned, and the upper layer can

```

check whether all event/alarm subscriptions are succeeded via the existence of node **FailedEventList**-->

```

<Event>
  <type>
    <!--req, xs:string, refer to Supported Alarm/Event Types for details-->
  </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, opt="binary,localURL,cloudStorageURL", alarm picture transmission mode: "binary"-binary,
"localURL"-device local URL, "cloudStorageURL"-cloud storage URL-->
  </pictureURLType>
  <channels>
    <!--opt, xs:string, "1,2,3,4...", event related channel ID, supports multiple channels, and the channel ID is
separated by commas-->
  </channels>
  <subStatusCode>
    <!--req, string, subscription failure error code-->
  </subStatusCode>
</Event>
</FailedEventList>
</SubscribeEventResponse>

```

C.1.68 XML_SubscribeEventCap

SubscribeEventCap capability message in XML format

```

<SubscribeEventCap version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <heartbeat min="" max="" />
    <!--optional, heartbeat time interval, unit: second-->
  <format opt="xml,json"/><!--req, supported message format-->
  <channelMode opt="all,list" />
    <!--required, channel subscription mode: "all"-subscribe events/alarms of all channels, "list"-subscribe events/
alarms of specific channels-->
  <eventMode opt="all,list" />
    <!--required, event subscription mode: "all"-subscribe all event types (must be supported), "list"-subscribe specific
event types, if "list" is returned, "all" will also be returned-->
    <!--if both the channelMode and eventMode returns "all", it indicates that the device does not support subscribing

```

```

event/alarm by event type or channel-->
<EventList><!--required, dependent, upload mode of specified alarms/events, it is valid only when eventMode is
"list"-->
  <Event><!--required-->
    <type><!--required, xs:string, refer to Supported Alarm/Event Types for details--></type>
    <minorAlarm opt="0x400,0x401,0x402,0x403">
      <!--opt, xs:string, minor alarm type, see details in Access Control Event Type. This node is required when type is
"AccessControllerEvent"-->
    </minorAlarm>
    <minorException opt="0x400,0x401,0x402,0x403">
      <!--opt, xs:string, minor exception type, see details in Access Control Event Type. This node is required when type
is "AccessControllerEvent"-->
    </minorException>
    <minorOperation opt="0x400,0x401,0x402,0x403">
      <!--opt, xs:string, minor operation type, see details in Access Control Event Type. This node is required when type
is "AccessControllerEvent"-->
    </minorOperation>
    <minorEvent opt="0x01,0x02,0x03,0x04">
      <!--opt, xs:string, minor event type, see details in Access Control Event Type. This node is required when type is
"AccessControllerEvent"-->
    </minorEvent>
    <pictureURLType opt="binary,localURL,cloudStorageURL" def=""/>
    <!--opt, xs:string, alarm picture format: "binary"-binary, "localURL"-device local URL, "cloudStorageURL"-cloud
storage URL, and the def is followed by the default format-->
  </Event>
</EventList>
<pictureURLType opt="binary,localURL,cloudStorageURL" def=""/>
  <!--opt, xs:string, alarm picture format: "binary"-binary picture, "localURL"-device local URL, "cloudStorageURL"-
cloud storage URL. This node is the method of uploading all pictures related to the event. If this node is applied,
<pictureURLType> in <Event> is invalid; otherwise, pictures will be uploaded using the default method returned by the
device capability. For front-end devices, the default method is uploading binary pictures; for back-end devices, the
default method is by device local URL-->
  <ChangedUploadSub><!--message subscription-->
    <interval/><!--opt, xs:integer, lifecycle of arming GUID, the default value is 5 minutes, unit: second. The device will
generate new GUID for the arming connection after it is disconnected for the set lifecycle-->
  <StatusSub>
    <all/><!-- opt, xs:boolean, whether to subscribe all events-->
    <channel/><!--opt, xs:boolean, whether to subscribe channel status. This node is not required when <all> is
"true"-->
    <hd/><!--opt, xs:boolean, whether to subscribe disk status. This node is not required when <all> is "true"-->
    <capability/><!--opt, xs:boolean, whether to subscribe capability change status. This node is not required when
<all> is "true"-->
  </StatusSub>
</ChangedUploadSub>
<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-->
</SubscribeEventCap>

```

C.1.69 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.1.70 XML_ThermalBlackBody

ThermalBlackBody message in XML format.

```
<ThermalBlackBody version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <emissivity><!--req, xs:float, black body emissivity is between 0.01 and 1, which corrects to two decimal places. The
default value is 0.97 --></emissivity>
  <distance><!--req, xs: float, the distance between lens and black body, ranges from 0.0 m to 10.0 m, which corrects
to one decimal place. The default value is 2.0 m, unit:m--></distance>
  <temperature><!--req, xs:float, black body temperature is between 30.0 °C and 50.0 °C, which corrects to one
decimal place. The default value is 35.0 °C, unit: °C--></temperature>
  <CentrePoint><!--req, center position of black body. After clicking this position, the normalized coordinate
information (between 0 and 1000) will be applied-->
    <CalibratingCoordinates><!--dep-->
      <positionX><!--req, xs:integer; coordinate--></positionX>
      <positionY><!--req, xs:integer; coordinate--></positionY>
    </CalibratingCoordinates>
  </CentrePoint>
  <normalizedScreenSize><!--req, ro, read-only, it is the multiples of normalized coordinates returned by device-->
    <normalizedScreenWidth><!--req, ro,xs:integer--></normalizedScreenWidth>
    <normalizedScreenHeight><!--req, ro,xs:integer--></normalizedScreenHeight>
  </normalizedScreenSize>
  <enabled opt="true,false"><!--optional, xs:boolean, whether to enable black body--></enabled>
  <BlackBodyReigon><!--optional, black body detection area-->
    <type opt="point,region"><!--required, xs:string, "point", "region"--></type>
    <Point><!--dependent, point coordinate, it is valid when value of type is "point"-->
      <CalibratingCoordinates><!--dependent, point coordinate-->
        <positionX><!--required, xs:integer; coordinate--></positionX>
        <positionY><!--required, xs:integer; coordinate--></positionY>
      </CalibratingCoordinates>
    </Point>
    <Region><!--dependent, frame coordinate, it is valid when value of type is "region"-->
      <RegionCoordinatesList size="4"><!--dependent, list of region coordinates-->
        <RegionCoordinates><!--list, optional-->
          <positionX><!--required, xs:integer; X-coordinate--></positionX>
          <positionY><!--required, xs:integer; Y-coordinate--></positionY>
        </RegionCoordinates>
      </RegionCoordinatesList>
    </Region>
  </BlackBodyReigon>
</ThermalBlackBody>
```

C.1.71 XML_ThermalCap

XML message about thermal capability

```

<ThermalCap version="2.0" xmlns="http://www.std-cgi.org/ver20/XMLSchema">
  <isSupportFireDetection><!--optional, xs:boolean, whether supports fire detection--></isSupportFireDetection>
  <isSupportThermometry><!--optional, xs:boolean, whether supports temperature measurement--></
isSupportThermometry>
  <isSupportRealtimeThermometry><!--optional, xs:boolean, whether supports uploading real-time temperature
measurement data--></isSupportRealtimeThermometry>
  <isFireFocusZoomSupport><!--optional, xs:boolean, whether supports visible light lens zooming--></
isFireFocusZoomSupport>
  <isSupportManualRanging>
    <!--optional, xs:boolean, this node will be returned if one or more channels of device supports this function, see
detailed channel capability in the response information"-->
  </isSupportManualRanging>
  <isSupportPower><!--optional, xs:boolean, whether supports power on/off capability--></isSupportPower>
  <isSupportRealtimeTempHumi><!--optional, xs:boolean, whether supports real-time detection of temperature and
humidity--></isSupportRealtimeTempHumi>
  <ManualThermCap>
    <manualThermRuleNum>
      <!--optional, xs:integer, the max. number of supported rules for manual temperature measurement. If this node is
not returned, it indicates manual temperature measurement is not supported-->
    </manualThermRuleNum>
  </ManualThermCap>
  <isSupportManualThermBasic>
    <!--optional, xs:boolean, whether supports basic configuration of manual temperature measurement-->
  </isSupportManualThermBasic>
  <isSupportFireShieldMask><!--optional, xs:boolean--></isSupportFireShieldMask>
  <isSupportsmokeShieldMask><!--optional, xs:boolean--></isSupportsmokeShieldMask>
  <isSupportThermometryMode>
    <!--optional, xs:boolean, whether the device supports the configuration of temperature measurement mode-->
  </isSupportThermometryMode>
  <isSupportThermalPip>
    <!--optional, xs:boolean, whether the device supports the PIP configuration-->
  </isSupportThermalPip>
  <isSupportThermalIntelRuleDisplay><!--optional, xs:boolean, whether supports VCA rule configuration--></
isSupportThermalIntelRuleDisplay>
  <AlgVersionInfo><!--opt, whether supports getting the version information of thermal algorithms library-->
    <thermometryAlgName min = "1" max = "128">
      <!--read-only, xs:string, version information of temperature measurement algorithms library-->
    </thermometryAlgName>
    <shipsAlgName min = "1" max = "128"><!--read-only, xs:string, version name of ship detection algorithms library--
></shipsAlgName>
  </AlgVersionInfo>
  <isSupportFaceThermometry><!--optional, xs:boolean, whether supports temperature screening configuration--></
isSupportFaceThermometry>
  <isSupportThermalBlackBody><!--optional, xs:boolean, whether supports black body configuration--></
isSupportThermalBlackBody>
  <isSupportThermalStreamParam><!--optional, xs:boolean, whether supports stream configuration--></
isSupportThermalStreamParam>
  <isSupportBodyTemperatureCompensation>
    <!--optional, xs:boolean, whether supports temperature compensation configuration-->
  </isSupportBodyTemperatureCompensation>
  <isSupportTemperatureCorrection><!--optional, xs:boolean, whether device supports temperature measurement
correction--></isSupportTemperatureCorrection>

```



```
<isSupportClickToThermometry><!--optional, xs:boolean, whether device supports clicking to detect temperature--></isSupportClickToThermometry>
<isSupportThermometryHistorySearch><!--optional, xs:boolean--></isSupportThermometryHistorySearch>
<isSupportBurningPrevention><!--optional, xs:boolean, whether device supports burning prevention--></isSupportBurningPrevention>
<isSupportTemperatureCollection><!--optional, xs:boolean, whether device supports temperature ANR--></isSupportTemperatureCollection>
<isSupportJpegPicWithAppendData>
  <!--optional, xs:boolean, whether device supports getting JPEG picture with pixel-to-pixel temperature measurement data. If supports, it is returned and values true, if not support, it is not returned-->
</isSupportJpegPicWithAppendData>
<isSupportRealTimethermometryForHTTP>
  <!--optional, xs:boolean, whether device supports real-time temperature measurement. If supports, it is returned and its value is true, if not support, it is not returned-->
</isSupportRealTimethermometryForHTTP>
<isSupportShipsDetectionWithScene>
  <!--optional, xs:boolean, whether device supports ship detection by scene, this node and isSupportShipsDetection in XML_SmartCap are mutually exclusive-->
</isSupportShipsDetectionWithScene>
<isSupportthermometryOffLineCapture>
  <!--optional, xs:boolean, whether device supports offline capture. If supports, this node returned and its value is true; if not, it is not returned-->
</isSupportthermometryOffLineCapture>
<isSupportThermalTemperatureCorrect>
  <!--optional, xs:boolean, whether device supports temperature calibration (related URI: /ISAPI/Thermal/channels/<ID>/temperatureCorrect?format=json)-->
</isSupportThermalTemperatureCorrect>
<isSupportGreyScaleAlarm>
  <!--optional, xs:boolean, whether device supports grayscale alarm. If supports, this node returned and its value is true; if not, it is not returned-->
</isSupportGreyScaleAlarm>
<isSupportFaceSnapThermometry>
  <!--optional, xs:boolean, whether device supports uploading captured face picture with temperature information: true-support, no return-not support-->
</isSupportFaceSnapThermometry>
<isSupportTemperatureIntervalMeasurement>
  <!--optional, xs:boolean, whether device supports interval temperature measurement. If supports, this node returned and its value is true; if not, it is not returned-->
</isSupportTemperatureIntervalMeasurement>
</ThermalCap>
```

Remarks

When getting thermal product capabilities, **isSupportShipsDetectionWithScene** has a higher priority than **isSupportShipsDetection**. That is, firstly check if the node **isSupportShipsDetectionWithScene** exists and its value is "true", that indicates ship detection according to scene is supported, otherwise, check the if the node **isSupportShipsDetection** exists.

C.1.72 XML_ThermIntell

ThermIntell message in XML format.

```
<ThermIntell version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id><!--required, xs:integer--></id>
  <intellType>
    <!--required, xs:string, intelligent function types, each two types are mutually exclusive, "thermometryAndSmart"-
    temperature measurement+behavior analysis (default), "shipsDetection"-ship detection, "fireDetection"-fire source
    detection, "pip"-picture in picture function, "faceThermometry"-temperature screening,
    "thermometryAndSmokeFireDetection"-temperature measurement+smoke and fire source detection,
    "thermometryAndFireDetection"-temperature measurement+fire source detection, "basicBehavior"-behavior
    analysis, "thermometry"-temperature measurement-->
  </intellType>
</ThermIntell>
```

C.1.73 XML_ThermometryAlarmRule

ThermometryAlarmRule message in XML format.

```
<ThermometryAlarmRule version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <ThermometryAlarmModeList size="">
    <ThermometryAlarmMode>
      <id min="" max=""><!--req, xs:inter--></id>
      <enabled><!--req, xs:boolean--></enabled>
      <name><!--req, xs:string ro--></name>
      <rule opt="highestGreater, highestLess, lowestGreater, lowestLess, averageGreater, averageLess, diffTempGreater,
diffTempLess">
        <!--req, xs:string,-->
      </rule>
      <alert><!--req, xs: float--></alert>
      <alarm><!--req, xs: float--></alarm>
      <threshold><!--req, xs: float--></threshold>
      <AlertOutputIOPortList><!--opt, temperature pre-alarm I/O port list-->
        <OutputIOPort>
          <portID><!--req, xs:string--></portID>
          <enabled><!--req, xs:bool, "true,false"--></enabled>
        </OutputIOPort>
      </AlertOutputIOPortList>
      <AlarmOutputIOPortList><!--opt, temperature alarm I/O port list-->
        <OutputIOPort>
          <portID><!--req, xs:string--></portID>
          <enabled><!--req, xs:bool, "true,false"--></enabled>
        </OutputIOPort>
      </AlarmOutputIOPortList>
      <alertFilteringTime><!-- opt ,xs:integer, temperature pre-alarm dwell time, unit: second--></alertFilteringTime>
      <alarmFilteringTime><!-- opt ,xs:integer, temperature alarm dwell time, unit: second--></alarmFilteringTime>
    </ThermometryAlarmMode>
  </ThermometryAlarmModeList>
```

```
<TemperatureDifferenceComparisonList size="">
  <TemperatureDifferenceComparison>
    <id min="" max=""><!--req, xs:integer--></id>
    <enabled><!--req, xs:boolean--></enabled>
    <ruleID1 min="" max=""><!--req, xs:string--></ruleID1>
    <ruleID2 min="" max=""><!--req, xs:string--></ruleID2>
    <rule opt="highestGreater, highestLess, lowestGreater, lowestLess, averageGreater, averageLess, diffTempGreater,
diffTempLess">
      <!--req, xs:string,-->
    </rule>
    <temperatureDifference><!--req, xs: float--></temperatureDifference>
  </TemperatureDifferenceComparison>
</TemperatureDifferenceComparisonList>
</ThermometryAlarmRule>
```

C.1.74 XML_ThermometryBasicParam

XML message about temperature measurement basic parameters

```
<ThermometryBasicParam version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id><!--required, xs:integer, channel number--></id>
  <enabled><!--required, xs:boolean, enable or not: false-no, true-yes--></enabled>
  <streamOverlay><!--required, xs:boolean, whether displays temperature information on the stream: false-no, true-yes--></streamOverlay>
  <pictureOverlay><!--required, xs:boolean, whether displays temperature information on the captured picture: false-no, true-yes--></pictureOverlay>
  <temperatureRange>
    <!--required, xs:string, temperature range: "-20-150", "0-550", "0-650", "-4-302", "32-1022", "32-1200", "20-650",
"-20-1500", "automatic", "-20-120", "20-350", "20-45", "20-350", "30-45", "100-550"-->
  </temperatureRange>
  <temperatureUnit>
    <!--required, xs:string, temperature unit: degreeCentigrade-Celsius (°C), degreeFahrenheit-Fahrenheit (°F),
degreeKelvin-Kelvin(K)-->
  </temperatureUnit>
  <temperatureCurve>
    <!--optional, xs:string, temperature curve modes: close-closed, transverseTemperatureTrend-transverse
temperature mode, longitudinalTemperatureTrend-longitudinal temperature mode-->
  </temperatureCurve>
  <fireImageMode>
    <!--optional, xs:string, fire detection modes: blackWhite-black and white mode, thermalProbe-thermal detection
mode, fireGroud-fire scene mode-->
  </fireImageMode>
  <emissivity>
    <!--optional, xs:float, the emissivity is between 0.01 and 1.00, which corrects to two decimal places. This parameter
is used by the device (i.e., DS-2TF03-260V/GLT, DS-2TF03-167V/GLT) that does not support regular frame and preset-->
  </emissivity>
  <distanceUnit>
    <!--required, xs:string, opt="meter,feet,centimeter"--></distanceUnit>
  <TemperatureColor>
    <!--optional, set the alarm information color for the temperature measurement-->
  <type>
```

```

<!-- required, xs:string, "highTemperature, lowTemperature, rangeTemperature, heatPreservation", set temperature
alarm types: highTemperature field-high temperature alarm, when the measured temperature is higher than the
configured value of highTemperature field, the measured temperature will be marked by color; lowTemperature-low
temperature alarm, when the measured temperature is lower than the configured value of lowTemperature field, the
measured temperature will be marked by color; rangeTemperature-range temperature alarm, when the measured
temperature is between the configured values of highTemperature field and lowTemperature field, the measured
temperature will be marked by color; heatPreservation-insulation alarm, when the measured temperature is not
between the configured values of highTemperature field and lowTemperature field, the measured temperature will be
marked by color-->
</type>
<highTemperature><!--dep, xs:integer--></highTemperature>
<lowTemperature><!--dep, xs:integer--></lowTemperature>
</TemperatureColor>
<enviroTemperature><!--optional, xs:integer, environment temperature, unit: °C--></enviroTemperature>
<enviroHumidity><!--optional, xs:integer, environment humidity, unit:%--></enviroHumidity>
<correctionVolume><!--optional, xs:integer, temperature correction--></correctionVolume>
<specialPointThermType>
  <!--required, xs:string, "centerPoint, highestPoint, lowestPoint", display the special point temperature, central point
temperature, highest temperature, lowest temperature. Supports multiple selections-->
</specialPointThermType>
<distance><!--required, xs:integer; unit:m, the distance range is [0, 10000]--></distance>
<reflectiveEnable><!--required, xs:boolean, whether enables temperature reflection--></reflectiveEnable>
<reflectiveTemperature><!--optional, xs:float, reflective temperature, which corrects to one decimal place--></
reflectiveTemperature>
<alert><!--optional, xs: float, pre-alarm threshold--></alert>
<alarm><!--optional, xs: float, alarm threshold--></alarm>
<showTempStripEnable><!--optional, xs:boolean, whether enables displaying temperature bar--></
showTempStripEnable>
<thermalOpticalTransmittance><!--optional, xs: float, optical transmissivity is between 0.001 and 1.000, which
corrects to three decimal places. The default value is 1.000--></thermalOpticalTransmittance>
<externalOpticsWindowCorrection><!--optional, xs: float, external optical temperature is between -40.0 °C and
80.0 °C. The default value is 20 °C--></externalOpticsWindowCorrection>
<AlertOutputIOPortList><!--optional, list of temperature pre-alarm output ports-->
  <OutputIOPort>
    <portID><!--required, xs:string --></portID>
    <enabled><!--required, xs:boolean, "true,false"--></enabled>
  </OutputIOPort>
</AlertOutputIOPortList>
<AlarmOutputIOPortList><!--optional, list of temperature alarm output ports-->
  <OutputIOPort>
    <portID><!--required, xs:string--></portID>
    <enabled><!--required, xs:boolean, "true,false"--></enabled>
  </OutputIOPort>
</AlarmOutputIOPortList>
<alertFilteringTime><!--optional, xs:integer, temperature pre-alarm dwell time, unit:s--></alertFilteringTime>
<alarmFilteringTime><!--optional, xs:integer, temperature alarm dwell time, unit:s--></alarmFilteringTime>
<displayMaxTemperatureEnabled><!--optional, xs:boolean, whether displays the maximum temperature--></
displayMaxTemperatureEnabled>
<displayMinTemperatureEnabled><!--optional, xs:boolean, whether displays the minimum temperature--></
displayMinTemperatureEnabled>
<displayAverageTemperatureEnabled><!--optional, xs:boolean, whether displays the average temperature--></
displayAverageTemperatureEnabled>

```

```

<thermometryInfoDisplayposition><!--optional, xs:string, position of temperature measurement information overlay,
"rules_around,top_left_of_screen"--></thermometryInfoDisplayposition>
<calibrationCoefficientEnabled><!--optional, xs:boolean, whether enables calibration coefficient--></
calibrationCoefficientEnabled>
<calibrationCoefficient><!--dep, xs:float, calibration coefficient, ranges from 0 to 30, corrects to two decimal places--
></calibrationCoefficient>
<emissivityMode>
  <!--optional, xs:string, emissivity type: "rougher"-rougher 0.95, "rough"-rough 0.80, "smooth"-smooth 0.60,
"smoother"-smoother 0.30, "customsettings"-customized setting, values from 0.01 to 1.00, the larger the value, the
higher the roughness-->
</emissivityMode>
<displayTemperatureInOpticalChannelEnabled>
  <!--optional, xs:boolean, display the temperature information of optical channel-->
</displayTemperatureInOpticalChannelEnabled>
<distanceMode><!--optional, xs:string, distance mode: "selfAdapt"-self-adaption, "fixed"-fixed distance--></
distanceMode>
<faceTemperatureInfoUploadEnabled>
  <!--optional, xs:boolean, whether to enable uploading face temperature information: true-enable, false-disable-->
</faceTemperatureInfoUploadEnabled>
<calibrationFileVersion><!--optional, xs:string, read-only, calibration file version information--></
calibrationFileVersion>
<alarmInterval><!--optional, xs:integer, temperature measurement interval, unit: second--></alarmInterval>
<rulesOverlayMode><!--optional, xs:string, rule overlay mode: all (all rules), alarm (triggered alarm rule)--></
rulesOverlayMode>
<toleranceTemperature><!--optional, xs:float, tolerance temperature, value range: [1,5], unit: Celsius--></
toleranceTemperature>
<alarmMode><!--optional, xs:string, alarm mode: "temperatureIntervalMeasurement" (temperature range
measurement), "alarm_alert" (pre-alarm/alarm)--></alarmMode>
<NormalRulesColor><!--optional, normal rule color; this node is valid when the value of alarmMode is
"temperatureIntervalMeasurement"-->
  <R><!--required, xs:integer--></R>
  <G><!--required, xs:integer--></G>
  <B><!--required, xs:integer--></B>
</NormalRulesColor>
<NormalTemperatureIntervalMeasurement><!--optional, normal temperature range measurement-->
  <alarmType><!--optional, xs:string, alarm type: "highestTemp" (the highest temperature), "lowestTemp" (the lowest
temperature)--></alarmType>
  <TemperatureIntervalList><!--optional, temperature range list; up to 4 temperature ranges are allowed-->
    <TemperatureInterval>
      <id><!--optional, xs:integer, No.--></id>
      <enabled><!--optional, xs:boolean, whether to enable--></enabled>
      <name><!--optional, xs:string, range name--></name>
      <minTemperature><!--optional, xs:float, the lowest temperature; value range: [-20,550], unit: Celsius; the value
should be accurate to one decimal place--></minTemperature>
      <maxTemperature><!--optional, xs:float, the highest temperature; value range: [-20,550], unit: Celsius; the value
should be accurate to one decimal place--></maxTemperature>
      <AlarmColor><!--optional, alarm color of temperature range-->
        <R><!--required, xs:integer--></R>
        <G><!--required, xs:integer--></G>
        <B><!--required, xs:integer--></B>
      </AlarmColor>
      <AlarmOutputIOPortList><!--optional, alarm output port list-->

```

```
<OutputIOPort><!--list-->
  <portID><!--required, xs:string, port No.--></portID>
  <enabled><!--required, xs:boolean, whether to enable--></enabled>
</OutputIOPort>
</AlarmOutputIOPortList>
</TemperatureInterval>
</TemperatureIntervalList>
</NormalTemperatureIntervalMeasurement>
</ThermometryBasicParam>
```

Remarks

The following nodes are not supported by the thermographic automation thermal camera (DS-2TA03-15SVI, DS2TA06-25SVI): **<TemperatureColor>**, **<specialPointThermType>**, and **<reflectiveEnable>**.

C.1.75 XML_ThermometryMode

XML message about temperature measurement mode information

```
<ThermometryMode version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <mode><!--required, xs:string, temperature measurement: "normal"-normal mode (default), "expert"-expert mode-->
</mode>
  <thermometryROIEnabled><!--optional, xs:boolean, whether to enable ROI temperature measurement, "true"-yes,
"false"-no--></thermometryROIEnabled>
</ThermometryMode>
```

Remarks

The ROI temperature measurement and mode configuration is mutually exclusive, ROI temperature measurement is mainly applied to temperature measurement.

C.1.76 XML_ThermometryRegion

ThermometryRegion message in XML format

```
<ThermometryRegion version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id><!--req, xs:integer--></id>
  <enabled><!--req, xs:boolean--></enabled>
  <name><!--req, xs:string--></name>
  <emissivity><!--req, xs:float, corrects to two decimal places--></emissivity>
  <distance><!--req, xs:integer--></distance>
  <reflectiveEnable><!--req, xs:boolean--></reflectiveEnable>
  <reflectiveTemperature><!--opt, xs:float, reflective temperature, corrects to one decimal place--></
reflectiveTemperature>
  <sensitivity> <!--req, xs:integer, sensitivity, normalized value, from 1 to 5--></sensitivity>
  <PupilParam><!--req, pupil parameters-->
    <Region><!--req, pupil region coordinates, normalized value, from 0 to 1000-->
      <RegionCoordinatesList> <!--dep-->
        <RegionCoordinates> <!--opt -->
```

```

    <positionX><!-- req, xs:integer;coordinate--></positionX>
    <positionY><!-- req, xs:integer;coordinate--></positionY>
  </RegionCoordinates>
</RegionCoordinatesList>
</Region>
</PupilParam>
<MaxPupilParam><!--optional, the maximum pupil distance-->
  <pupilDistance min="10" max="625">
    <!--optional, xs:integer, pupil distance range, normalized value, range: [0,1000], valid range: [10,625], default value:
625-->
  </pupilDistance>
  <Region>
    <RegionCoordinatesList size=""><!--req-->
      <RegionCoordinates><!--list, pupil region coordinates-->
        <positionX><!--required, xs:integer, coordinate, X-coordinate--></positionX>
        <positionY><!--required, xs:integer, coordinate, Y-coordinate--></positionY>
      </RegionCoordinates>
    </RegionCoordinatesList>
  </Region>
</MaxPupilParam>
<targetSpeed><!--req, xs:integer--></targetSpeed>
<alarmTemperature><!--opt, xs:float, unit:°C, alarm triggered temperature, corrects to one decimal place, ranges
from 0.0°C to 60.0°C--></alarmTemperature>
<type><!--opt, xs:string, "point,region,line"--></type>
<Point>
  <TempValue><!--dep, xs:float "-40.0...1000.0" ro--></TempValue>
  <CalibratingCoordinates><!--dep-->
    <positionX><!--req, xs:integer;coordinate--></positionX>
    <positionY><!--req, xs:integer;coordinate--></positionY>
  </CalibratingCoordinates>
</Point>
<Region>
  <highestTempValue><!--dep, xs: float "-273.0...10000.0" ro--></highestTempValue>
  <lowestTempValue><!--dep, xs: float "-273.0...10000.0" ro--></lowestTempValue>
  <averageTempValue><!--dep, xs: float "-273.0...10000.0" ro--></averageTempValue>
  <diffTempValue><!--dep, xs: float "-273.0...10000.0" ro--></diffTempValue>
  <RegionCoordinatesList><!--dep-->
    <RegionCoordinates><!--opt-->
      <positionX><!--req, xs:integer;coordinate--></positionX>
      <positionY><!--req, xs:integer;coordinate--></positionY>
    </RegionCoordinates>
  </RegionCoordinatesList>
</Region>
<distanceUnit><!--opt, xs:string, opt="meter,feet,centimeter"--></distanceUnit>
<emissivityMode>
  <!--opt,xs:string, emissivity type: "rougner"-rougner 0.95, "rough"-rough 0.80, "smooth"-smooth 0.60, "smoother"-
smoother 0.30, "customsettings"-customized setting, values from 0.01 to 1.00, the larger the value, the higher the
roughness-->
</emissivityMode>
<alarmRule><!--opt, xs:string, alarm rule, "highestGreater,highestLess"--></alarmRule>
</ThermometryRegion>

```

C.1.77 XML_TMPAScheduleList

XML message about temperature measurement pre-alarm arming schedules

```
<?xml version="1.0" encoding="utf-8"?>
<TMPAScheduleList version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <Schedule/><!--optional, see XML_Schedule for details-->
</TMPAScheduleList>
```

See Also

XML_Schedule

C.1.78 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>
```



Note

See *Response Codes of Text Protocol* for details about sub status codes and corresponding error codes.

C.2 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.
NET_DVR_ERRORALARMPORT	16	Alarm port No. error. The alarm input or output port of the specified device does not exist.

Error Name	Error Code	Error Description
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.
NET_DVR_OPERNOTFINISH	36	Operation conflicted.
NET_DVR_GETPLAYTIMEFAIL	37	Failed to get the current played time.
NET_DVR_PLAYFAIL	38	Failed to play.

Error Name	Error Code	Error Description
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.
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.

Error Name	Error Code	Error Description
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.
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.

Error Name	Error Code	Error Description
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.
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.

Error Name	Error Code	Error Description
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.
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.

Error Name	Error Code	Error Description
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.
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

Error Name	Error Code	Error Description
		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.
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 wirting permission.
NET_DVR_IP_CONFLICT	174	IP conflicted.

Error Name	Error Code	Error Description
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.
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.

Error Name	Error Code	Error Description
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.
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.

Error Name	Error Code	Error Description
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.
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.

Error Name	Error Code	Error Description
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.
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.

Error Name	Error Code	Error Description
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.
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.

Error Name	Error Code	Error Description
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
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.

Error Name	Error Code	Error Description
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.
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.

Error Name	Error Code	Error Description
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.
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.

Error Name	Error Code	Error Description
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.
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..

Error Name	Error Code	Error Description
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.
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.

Error Name	Error Code	Error Description
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.
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).

Error Name	Error Code	Error Description
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
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

Error Name	Error Code	Error Description
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)
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_PROGRESSIONG	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.

Error Name	Error Code	Error Description
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
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.

Error Name	Error Code	Error Description
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
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

Error Name	Error Code	Error Description
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
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.

Error Name	Error Code	Error Description
NET_DVR_ERR_LIGHT_COLOR_INVALID	1411	The color of the video detection of traffic light input signal lamp color is not legitimate.
NET_DVR_ERR_LIGHT_DIRECTION_INVALID	1412	The direction property of the video detection of traffic light input light is not valid.
NET_DVR_ERR_LACK_IOABLITY	1413	Lack of IO ablity.
NET_DVR_ERR_FTP_PORT	1414	FTP port error.
NET_DVR_ERR_FTP_CATALOGUE	1415	FTP catalogue error.
NET_DVR_ERR_FTP_UPLOAD_TYPE	1416	FTP upload type error.
NET_DVR_ERR_FLASH_PARAM_WRITE	1417	Setting param flash write error.
NET_DVR_ERR_FLASH_PARAM_READ	1418	Getting param flash read error.
NET_DVR_ERR_PICNAME_DELIMITER	1419	Pic name delimiter error.
NET_DVR_ERR_PICNAME_ITEM	1420	Pic name item error.
NET_DVR_ERR_PLATE_RECOGNIZE_TYPE	1421	Plate recognize type error.
NET_DVR_ERR_CAPTURE_TIMES	1422	Capture times error.
NET_DVR_ERR_LOOP_DISTANCE	1423	Loop distance error.
NET_DVR_ERR_LOOP_INPUT_STATUS	1424	Loop input status error.
NET_DVR_ERR_RELATE_IO_CONFLICT	1425	Related IO conflict.
NET_DVR_ERR_INTERVAL_TIME	1426	Interval time error.
NET_DVR_ERR_SIGN_SPEED	1427	Sign speed error.
NET_DVR_ERR_PIC_FLIP	1428	Flip is used.
NET_DVR_ERR_RELATE_LANE_NUMBER	1429	Related lane number error.
NET_DVR_ERR_TRIGGER_MODE	1430	Trigger mode error.
NET_DVR_ERR_DELAY_TIME	1431	Delay time error.
NET_DVR_ERR_EXCEED_RS485_COUNT	1432	Exceed RS485 count.
NET_DVR_ERR_RADAR_TYPE	1433	Radar type error.

Error Name	Error Code	Error Description
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.
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	Slave 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.

Error Name	Error Code	Error Description
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.
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.

Error Name	Error Code	Error Description
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.
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_PLATE_RECOGNIZE_AREA_PARAM	1496	Plate Recognize Area Param error.
NET_DVR_ERR_PORT_CONFLICT	1497	Port Conflict.

Error Name	Error Code	Error Description
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.
NET_ERR_VQD_CHAN_MAX	1503	The total number of VQD plans exceeds the max limit.
NET_ERR_VQD_TASK_MAX	1504	The total number of VQD tasks exceeds the max limit.
NET_DVR_ERR_EXCEED_MAX_CAPTURE_TIMES	1600	Capture times exceed 2 in flash mode.
NET_DVR_ERR_REDAR_TYPE_CONFLICT	1601	Radar type conflict.
NET_DVR_ERR_LICENSE_PLATE_NULL	1602	The license plate is null.
NET_DVR_ERR_WRITE_DATABASE	1603	Failed to write data into the database.
NET_DVR_ERR_LICENSE_EFFECTIVE_TIME	1604	The effective time of license plate error.
NET_DVR_ERR_PRERECORDED_STARTTIME_LONG	1605	The pre recorded start time is greater than the number of illegal capture.
NET_DVR_ERR_TRIGGER_RULE_LINE	1606	Trigger rule line error.
NET_DVR_ERR_LEFTRIGHT_TRIGGERLINE_NOTVERTICAL	1607	Left and right trigger line is not vertical.
NET_DVR_ERR_FLASH_LAMP_MODE	1608	Flash lamp mode error.
NET_DVR_ERR_ILLEGAL_SNAPSHOT_NUM	1609	Illegal capture number error.
NET_DVR_ERR_ILLEGAL_DETECTION_TYPE	1610	Illegal detection type error.
NET_DVR_ERR_POSITIVEBACK_TRIGGERLINE_HIGH	1611	Positive back to trigger line height error.
NET_DVR_ERR_MIXEDMODE_CAPTYPE_ALLTARGETS	1612	Mixed mode only supports capture type all targets.

Error Name	Error Code	Error Description
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.
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

Error Name	Error Code	Error Description
NET_ERR_LOCAL_CONTROL_DISADD	1911	Distributed access controller is not added
NET_ERR_LOCAL_CONTROL_HASADD	1912	Distributed access controller is added
NET_ERR_LOCAL_CONTROL_DOORNO_CONFLICT	1913	Conflict with added distributed access controller
NET_ERR_LOCAL_CONTROL_COMMUNICATION_FAIL	1914	Distributed access controller communication failed
NET_ERR_OPERAND_INEXISTENCE	1915	Operation object does not exist (operation to door, alarm output, alarm input, return when the object is not added)
NET_ERR_LOCAL_CONTROL_OVER_LIMIT	1916	Distributed access controller exceeded device capability upper limit
NET_ERR_DOOR_OVER_LIMIT	1917	Door exceeded device capability upper limit
NET_ERR_ALARM_OVER_LIMIT	1918	Alarm input and output exceeded device capability upper limit
NET_ERR_LOCAL_CONTROL_ADDRESS_INCONFORMITY_TYPE	1919	Distributed access controller address does not match with type
NET_ERR_NOT_SUPPORT_ONE_MORE_CARD	1920	not support one person multi-card
NET_ERR_DELETE_NO_EXISTENCE_FACE	1921	The face picture does not exist.
NET_ERR_DOOR_SPECIAL_PASSWORD_REPEAT	1922	Repeated door door duress code, the super password, or the dismiss code.
NET_ERR_AUTH_CODE_REPEAT	1923	Repeated device authentication code
NET_ERR_DEPLOY_EXCEED_MAX	1924	No more devices can be armed.
NET_ERR_NOT_SUPPORT_DEL_FP_BY_ID	1925	The fingerprint module does not support deleting fingerprint by finger ID.
NET_ERR_TIME_RANGE	1926	Invalid range of the effective period.
NET_ERR_CAPTURE_TIMEOUT	1927	Collection timed out.
NET_ERR_LOW_SCORE	1928	Low quality of collected data.

Error Name	Error Code	Error Description
NET_ERR_OFFLINE_CAPTURING	1929	The device is collecting data offline and cannot respond.
NET_DVR_ERR_OUTDOOR_COMMUNICATION	1950	Communication exception with outdoor terminal
NET_DVR_ERR_ROOMNO_UNDEFINED	1951	Room number is not set
NET_DVR_ERR_NO_CALLING	1952	No call
NET_DVR_ERR_RINGING	1953	Ringling
NET_DVR_ERR_IS_CALLING_NOW	1954	Call in progress
NET_DVR_ERR_LOCK_PASSWORD_WRONG	1955	Incorrect smart lock password
NET_DVR_ERR_CONTROL_LOCK_FAILURE	1956	Lock control failure
NET_DVR_ERR_CONTROL_LOCK_OVERTIME	1957	Lock control timed out
NET_DVR_ERR_LOCK_DEVICE_BUSY	1958	Smart lock device busy
NET_DVR_ERR_UNOPEN_REMOTE_LOCK_FUNCTION	1959	Remote lock control not enabled
NET_DVR_ERR_FILE_NOT_COMPLETE	2100	Downloaded file is incomplete
NET_DVR_ERR_IPC_EXIST	2101	The camera already exists
NET_DVR_ERR_ADD_IPC	2102	Camera has been added to the channel
NET_DVR_ERR_OUT_OF_RES	2103	Not enough network bandwidth
NET_DVR_ERR_CONFLICT_TO_LOCALIP	2104	IP address of camera conflicts with that of DVR
NET_DVR_ERR_IP_SET	2105	Invalid IP address
NET_DVR_ERR_PORT_SET	2106	Invalid port number
NET_ERR_WAN_NOTSUPPORT	2107	Not in the same LAN, cannot set security question or export GUID file
NET_ERR_MUTEX_FUNCTION	2108	Mutually exclusive function
NET_ERR_QUESTION_CONFIGNUM	2109	Error in number of security question configurations

Error Name	Error Code	Error Description
NET_ERR_FACECHAN_NORESOURCE	2110	All the face VCA channels are occupied.
NET_ERR_DATA_CALLBACK	2111	Data is calling back.
NET_ERR_ATM_VCA_CHAN_IS_RELATED	2112	The VCA channel is already linked.
NET_ERR_ATM_VCA_CHAN_IS_OVERLAPED	2113	The VCA channel is already overlaid.
NET_ERR_FACE_CHAN_UNOVERLAP_EACH_OTHER	2114	The face channels cannot be overlaid.
NET_DVR_SMD_ENCODING_NORESOURCE	2116	Insufficient SMD encoding resource
NET_DVR_SMD_DECODING_NORESOURCE	2117	Insufficient SMD decoding resource
NET_DVR_FACELIB_DATA_PROCESSING	2118	Face picture library data is in processing
NET_DVR_ERR_LARGE_TIME_DIFFERENCE	2119	There is a great time difference between device and server.
NET_DVR_NO_SUPPORT_WITH_PLAYBACK	2120	It is not supported. Playback is enabled.
NET_DVR_CHANNEL_NO_SUPPORT_WITH_SMD	2121	It is not supported. SMD of channel is enabled.
NET_DVR_CHANNEL_NO_SUPPORT_WITH_FD	2122	It is not supported. Face capture of channel is enabled.
NET_DVR_ILLEGAL_PHONE_NUMBER	2123	Invalid telephone number
NET_DVR_ILLEGAL_CERTIFICATE_NUMBER	2124	Invalid ID No.
NET_DVR_ERR_CHANNEL_RESOLUTION_NO_SUPPORT	2125	The channel resolution is not supported
NET_DVR_ERR_CHANNEL_COMPRESSION_NO_SUPPORT	2126	The channel encoding format is not supported
NET_DVR_ERR_CLUSTER_DEVICE_TOO_LESS	2127	Deleting is not allowed. The number of devices is not enough

Error Name	Error Code	Error Description
NET_DVR_ERR_CLUSTER_DEL_DEVICE_CM_PAYLOAD	2128	Deleting is not allowed. The device is cluster host.
NET_DVR_ERR_CLUSTER_DEVNUM_OVER_UPPER_LIMIT	2129	No more devices can be added.
NET_DVR_ERR_CLUSTER_DEVICE_TYPE_INCONFORMITY	2130	Device type mismatched.
NET_DVR_ERR_CLUSTER_DEVICE_VERSION_INCONFORMITY	2131	Device version mismatched.
NET_DVR_ERR_CLUSTER_IP_CONFLICT	2132	Cluster system IP address conflict: ipv4 address conflict, invalid ipv6.
NET_DVR_ERR_CLUSTER_IP_INVALID	2133	Invalid cluster system IP address: invalid ipv4, invalid ipv6.
NET_DVR_ERR_CLUSTER_PORT_CONFLICT	2134	Cluster system port conflict
NET_DVR_ERR_CLUSTER_PORT_INVALID	2135	Invalid cluster system port
NET_DVR_ERR_CLUSTER_USERNAEM_OR_PASSWORD_INVALID	2136	Invalid user name or password
NET_DVR_ERR_CLUSTER_DEVICE_ALREADY_EXIST	2137	The device already exists.
NET_DVR_ERR_CLUSTER_DEVICE_NOT_EXIST	2138	The device does not exist.
NET_DVR_ERR_CLUSTER_NON_CLUSTER_MODE	2139	The device working mode is not the cluster mode .
NET_DVR_ERR_CLUSTER_IP_NOT_SAME_LAN	2140	IP addresses are in different LAN. Building cluster or extending capacity for NVRs in different LAN is not allowed.
NET_DVR_ERR_IDENTITY_KEY	2147	Incorrect interaction password
NET_DVR_MISSING_IDENTITY_KEY	2148	Interaction password is missing
NET_DVR_ERR_CAPTURE_PACKAGE_FAILED	2141	Capturing packets failed.
NET_DVR_ERR_CAPTURE_PACKAGE_PROCESSING	2142	Capturing packet.

Error Name	Error Code	Error Description
NET_DVR_ERR_SAFETY_HELMET_NO_RESOURCE	2143	No enough hard hat detection resource.
NET_DVR_NO_SUPPORT_WITH_ABSTRACT	2144	This function is not supported. Video synopsis is already enabled.
NET_DVR_INSUFFICIENT_DEEP_LEARNING_RESOURCES	2146	No more deep learning resources can be added.
NET_DVR_NO_SUPPORT_WITH_PERSON_DENSITY_DETECT	2149	People gathering density is enabled, it is not supported
NET_DVR_IPC_RESOLUTION_OVERFLOW	2150	The network camera resolution is too large
NET_DVR_IPC_BITRATE_OVERFLOW	2151	The network camera bitrate is too large
NET_DVR_ERR_INVALID_TASKID	2152	Invalid taskID
NET_DVR_PANEL_MODE_NOT_CONFIG	2153	The ATM panel mode is not configured.
NET_DVR_NO_HUMAN_ENGINES_RESOURCE	2154	No enough engine resource
NET_DVR_ERR_TASK_NUMBER_OVERFLOW	2155	No more task data is allowed
NET_DVR_ERR_COLLISION_TIME_OVERFLOW	2156	Collision time is over the limit
NET_DVR_ERR_EVENT_NOTSUPPORT	2159	Subscribing alarm/event is not supported.
NET_DVR_IPC_NUM_REACHES_LIMIT	2184	The max. number of network camera channels reached.
NET_DVR_IOT_NUM_REACHES_LIMIT	2185	The max. number of IoT channels reached
NET_DVR_IOT_CHANNEL_DEVICE_EXIST	2186	Device of the IoT channel already exists.
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

Error Name	Error Code	Error Description
NET_DVR_INVALID_EZVIZ_SECRET_KEY	2189	Invalid verification code
NET_DVR_DUPLICATE_IOT_DEVICE	2190	Duplicated IoT device
NET_DVR_ERROR_NEED_DOUBLE_VERIFICATION	2206	Double verification is required
NET_DVR_NO_DOUBLE_VERIFICATION_USER	2207	No double verification user
NET_DVR_TIMESPAN_NUM_OVER_LIMIT	2209	Max. number of time buckets reached
NET_DVR_CHANNEL_NUM_OVER_LIMIT	2210	Max. number of channels reached
NET_DVR_NO_SEARCH_ID_RESOURCE	2211	Insufficient searchID resources
NET_DVR_SWITCH_TIMEDIFF_LESS_LIMIT	2249	Time difference between power on and off should be less than 10 minutes.
NET_DVR_NO_SUPPORT_DELETE_STRANGER_LIB	2262	Deleting stranger library is not supported
NET_DVR_NO_SUPPORT_CREATE_STRANGER_LIB	2263	Creating stranger library is not supported
NET_DVR_SSD_FILE_SYSTEM_ERROR	2266	SSD file system error
NET_DVR_INSUFFICIENT_SSD_FOR_FPD	2267	Insufficient SSD space for person frequency detection
NET_DVR_SMRDISK_NOT_SUPPORT_RAID	2269	SMR disk does not support RAID.
NET_DVR_ERR_NOTSUPPORT_DEICING	3001	Device does not support deicing function under current status.(Deicing function is only supported under the power status of POE+, AC24V, and DC12V).
NET_DVR_ERR_THERMENABLE_CLOSE	3002	Temperature measurement function is not enabled. (The enable function in NET_DVR_THERMOMETRY_BASICPARAM is not turned on)

Error Name	Error Code	Error Description
NET_DVR_ERR_PANORAMIC_LIMIT_OPERATED	3004	Panoramic map and limit cannot be operated at same time
NET_DVR_ERR_SMARTH264_ROI_OPERATED	3005	SmartH264 and ROI cannot be enabled at the same time.
NET_DVR_ERR_RULENUM_LIMIT	3006	No more rules can be added.
NET_DVR_ERR_LASER_DEICING_OPERATED	3007	Laser and deicing function cannot be enabled at the same time.
NET_DVR_ERR_OFFDIGITALZOOM_OR_MINZOOMLIMIT	3008	Please disable the digital zoom function or set the zoom limit to the minimum value. Otherwise, when enabling smoke and fire detection, behavior analysis, ship detection, defective point correction, temperature measurement, smoke and fire shielding function, this error code will be prompted.
NET_DVR_SYNCHRONIZEFOV_ERROR	3010	Field of view synchronization failed.
NET_DVR_RULE_SHIELDMASK_CONFLICT_ERROR	3013	The rule region conflicts with the shielded area.
NET_DVR_ERR_NO_SAFETY_HELMET_REGION	3501	The hard hat detection area is not configured.
NET_DVR_ERR_UNCLOSED_SAFETY_HELMET	3502	The hard hat detection is enabled.
NET_DVR_UPLOAD_HBDLIBID_ERROR	3504	Incorrect ID of human body picture library (incorrect HBDID or customHBDID)

RTSP Communication Library Related Errors

Error Name	Error Code	Error Description
NET_DVR_RTSP_ERROR_NOENOUGHPRI	401	Authentication failed: if server returns 401, it will change to this error code
NET_DVR_RTSP_ERROR_ALLOC_RESOURCE	402	Failed to allocate the resource
NET_DVR_RTSP_ERROR_PARAMETER	403	Parameter error

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_DESCRIPTERROR	410	RTSP DESCRIBE communicate error
NET_DVR_RTSP_DESCRIBESENDTIMEOUT	411	Sending "RTSP DESCRIBE" is timeout.
NET_DVR_RTSP_DESCRIBESENDERROR	412	Failed to send "RTSP DESCRIBE".
NET_DVR_RTSP_DESCRIBERCVTIMEOUT	413	Receiving "RTSP DESCRIBE" is timeout.
NET_DVR_RTSP_DESCRIBERCVDATAHOST	414	Receiving data of "RTSP DESCRIBE" error.
NET_DVR_RTSP_DESCRIBERCVERROR	415	Failed to receive "RTSP DESCRIBE".
NET_DVR_RTSP_DESCRIBESERVERERR	416	"RTSP DESCRIBE, the device returns the error code: 501 (failed to allocate the resource in the device)
NET_DVR_RTSP_SETUPERROR	420	(or 419), RTSP SETUP interaction error. Generally, it is that the address(URL) returned by the device is not accessible, or it is rejected by the server
NET_DVR_RTSP_SETUPSENDTIMEOUT	421	Sending "RTSP SETUP" is timeout.
NET_DVR_RTSP_SETUPSENDERROR	422	Sending "RTSP SETUP" error.
NET_DVR_RTSP_SETUPRCVTIMEOUT	423	Receiving "RTSP SETUP" is timeout.
NET_DVR_RTSP_SETUPRCVDATAHOST	424	Receiving data of "RTSP SETUP" error.
NET_DVR_RTSP_SETUPRCVERROR	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.
NET_PLAYM4_DEC_VIDEO_ERROR	504	Failed to decode video data.

Error Name	Error Code	Error Description
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.
NET_PLAYM4_UPDATE_ERROR	523	Failed to update overlay surface

Error Name	Error Code	Error Description
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
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

Error Name	Error Code	Error Description
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.

Error Name	Error Code	Error Description
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_STEAM_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.

C.3 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.

Sub Status Code	Error Code	Description
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.
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.

Sub Status Code	Error Code	Description
cloudAutoUpgradeException	0x30000016	Downloading upgrade packet from cloud and upgrading failed.
HBPEException	0x30001000	HBP exception.
UDEPEException	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.
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.
fileAlreadyExists	0x40001007	The file already exists.
invalidID	0x40001008	Invalid ID.
backupnodeNotAlloweLog	0x40001009	Accessing to backup node is not allowed.
exportingError	0x4000100A	Exporting failed.
exporting	0x4000100B	Exporting.
exportEnded	0x4000100C	Exporting stopped.
exported	0x4000100D	Exported.

Sub Status Code	Error Code	Description
IPOccupied	0x4000100E	The IP address is already occupied.
IDAlreadyExists	0x4000100F	The ID already exists.
exportItemsExceedLimit	0x40001010	No more items can be exported.
noFiles	0x40001011	The file does not exist.
beingExportedByAnotherUser	0x40001012	Being exported by others.
needReAuthentication	0x40001013	Authentication is needed after upgrade.
unitAddNotOnline	0x40001015	The added data analysis server is offline.
unitControl	0x40001016	The data analysis server is already added.
analysis unitFull	0x40001017	No more data analysis server can be added.
unitIDError	0x40001018	The data analysis server ID does not exist.
unitExit	0x40001019	The data analysis server already exists in the list.
unitSearch	0x4000101A	Searching data analysis server in the list failed.
unitNotOnline	0x4000101B	The data analysis server is offline.
unitInfoError	0x4000101C	Getting data analysis server information failed.
unitGetNodeInfoError	0x4000101D	Getting node information failed.
unitGetNetworkInfoError	0x4000101E	Getting the network information of data analysis server failed
unitSetNetworkInfoError	0x4000101F	Setting the network information of data analysis server failed
setSmartNodeInfoError	0x40001020	Setting node information failed.
setUnitNetworkInfoError	0x40001021	Setting data analysis server network information failed.
unitRestartCloseError	0x40001022	Rebooting or shutting down data analysis server failed.
virtualIPnotAllowed	0x40001023	Adding virtual IP address is not allowed.
unitInstalled	0x40001024	The data analysis server is already installed.
badSubnetMask	0x40001025	Invalid subnet mask.

Sub Status Code	Error Code	Description
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.
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.

Sub Status Code	Error Code	Description
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.
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.

Sub Status Code	Error Code	Description
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.
TDAConnectFailed	0x40001064	Connecting to traffic data access server failed. Checking the server status.
86sdkConnectFailed	0x40001065	Connecting to listening port of iVMS-8600 System failed. Check the parameters.
nameExist	0x40001066	Duplicated server name.
batchProcessFailed	0x40001067	Processing in batch failed.
IDNotExist	0x40001068	The server ID does not exist.
serviceNumberReachesLimit	0x40001069	No more service can be added.
invalidServiceType.	0x4000106A	Invalid service type.
clusterGetInfo	0x4000106B	Getting cluster group information failed.
clusterDelNode	0x4000106C	Deletion node failed.
clusterAddNode	0x4000106D	Adding node failed.
clusterInstalling	0x4000106E	Creating cluster...Do not operate.

Sub Status Code	Error Code	Description
clusterUninstall	0x4000106F	Reseting cluster...Do not operate.
clusterInstall	0x40001070	Creating cluster failed.
clusterIpError	0x40001071	Invalid IP address of task scheduler server.
clusterNotSameSeg	0x40001072	The master node and slave node must be in the same network segment.
clusterVirIpError	0x40001073	Automatically getting virtual IP address failed. Enter manually.
clusterNodeUnadd	0x40001074	The specified master(slave) 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 master and slave nodes.
addNodeNetFailed	0x40001077	Adding node failed. The network disconnected.
needTwoMgmtNode	0x40001078	Two management nodes are required when adding master and slave 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&mainIPnotSame NetSegment	0x4000107C	The virtual IP address is not in the same network segment with the IP address of master/slave node.
getNodeDispatchInfoFa iled	0x4000107D	Getting node scheduler information failed.
unableModifyManage mentNetworkIP	0x4000107E	Editing management network interface failed. The analysis board is in the cluster.
notSpecifyVirtualIP	0x4000107F	Virtual IP address should be specified for master and slave 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.

Sub Status Code	Error Code	Description
ArmingPlateSame	0x40001086	The license plate number is repeatedly armed.
ArmingParseXLSError	0x40001087	Parsing arming information file failed.
ArmingTimeError	0x40001088	Invalid arming time period.
ArmingSearchTimeError	0x40001089	Invalid search time period.
armingRelationshipReachesLimit	0x4000108A	No more relation can be created.
duplicateArmingName	0x4000108B	The relation name already exists.
noMoreArmingListAdded	0x4000108C	No more blocklist library can be armed.
noMoreCamerasAdded	0x4000108D	No more camera can be armed.
noMoreArmingListAddedWithCamera	0x4000108E	No more library can be linked to the camera.
noMoreArmingPeriodAdded	0x4000108F	No more time period can be added to the arming schedule.
armingPeriodsOverlapped	0x40001090	The time periods in the arming schedule are overlapped.
noArmingAlarmInfo	0x40001091	The alarm information does not exist.
armingAlarmUnRead	0x40001092	Getting number of unread alarms failed.
getArmingAlarmError	0x40001093	Getting alarm information failed.
searchByPictureTimeout	0x40001094	Searching picture by picture timeout. Search again.
comparisonTimeRangeError	0x40001095	Comparison time period error.
selectMonitorNumberUpperLimit	0x40001096	No more monitoring point ID can be filtered.
noMoreComparisonTasksAdded	0x40001097	No more comparison task can be executed at the same time.
GetComparisonResultFailed	0x40001098	Getting comparison result failed.
comparisonTypeError	0x40001099	Comparison type error.

Sub Status Code	Error Code	Description
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.
getProgressFaild	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.

Sub Status Code	Error Code	Description
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 slave database failed. Try again.
syncBackupNTPTimeFailed	0x400010BF	Synchronizing NTP time of slave server failed.
clusterNotSelectLoopbackAddress	0x400010C0	Loopback address is not supported by the master or slave cluster.
addFaceRecordFailed	0x400010C1	Adding face record failed.
deleteFaceRecordFailed	0x400010C2	Deleting face record failed.
modifyFaceRecordFailed	0x400010C3	Editing face record failed.
queryFaceRecordFailed	0x400010C4	Searching face record failed.
faceDetectFailed	0x400010C5	Detecting face failed.
libraryNotExist	0x400010C6	The library does not exist.
blackListQueryExporting	0x400010C7	Exporting matched blocklists.

Sub Status Code	Error Code	Description
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.
deleteGuardFailed	0x400010D4	Deleting arming configuration failed.
modifyGuardFailed	0x400010D5	Editing arming configuration failed.
queryGuardFailed	0x400010D6	Searching arming configurations failed.
uploadUserSuperCaps	0x400010D7	No more user information can be uploaded.
bigDataServerConnectFailed	0x400010D8	Connecting to big data server failed.
microVideoCloudRequestInfoBuildFailed	0x400010D9	Adding response information of micro video cloud failed.
microVideoCloudResponseInfoBuildFailed	0x400010DA	Parsing response information of micro video cloud failed.

Sub Status Code	Error Code	Description
transcodingServerRequestInfoBuildFailed	0x400010DB	Adding response information of transcoding server failed.
transcodingServerResponseInfoParseFailed	0x400010DC	Parsing response information of transcoding server failed.
transcodingServerOffline	0x400010DD	Transcoding server is offline.
microVideoCloudOffline	0x400010DE	Micro video cloud is offline.
UPSServerOffline	0x400010DF	UPS monitor server is offline.
statisticReportRequestInfoBuildFailed	0x400010E0	Adding response information of statistics report failed.
statisticReportResponseInfoParseFailed	0x400010E1	Parsing response information of statistics report failed.
DisplayConfigInfoBuildFailed	0x400010E2	Adding display configuration information failed.
DisplayConfigInfoParseFailed	0x400010E3	Parsing display configuration information failed.
DisplayConfigInfoSaveFailed	0x400010E4	Saving display configuration information failed.
notSupportDisplayConfigType	0x400010E5	The display configuration type is not supported.
passError	0x400010E7	Incorrect password.
upgradePackageLarge	0x400010EB	Too large upgrade package.
sessionUserReachesLimit	0x400010EC	No more user can log in via session.
ISO8601TimeFormatError	0x400010ED	Invalid ISO8601 time format.
clusterDissolutionFailed	0x400010EE	Deleting cluster failed.
getServiceNodeInfoFailed	0x400010EF	Getting service node information failed.
getUPSInfoFailed	0x400010F0	Getting UPS configuration information failed.

Sub Status Code	Error Code	Description
getDataStatisticsReportFailed	0x400010F1	Getting data statistic report failed.
getDisplayConfigInfoFailed	0x400010F2	Getting display configuration failed.
renameAnalysisBoardNotAllowed	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.
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 master 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.

Sub Status Code	Error Code	Description
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.
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.

Sub Status Code	Error Code	Description
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 master and backup cluster requires at least two devices.
updatePackageFileNotUploaded	0x4000111E	Upgrade package is not uploaded.
highPerformanceTasksNotSupportDrawingDetectionRegions	0x4000111F	Drawing detection area is not allowed under high-performance mode.
controlCenterIDDoesNotExist	0x40001120	The control center ID does not exist.
regionIDDoesNotExist	0x40001121	The area ID does not exist.
licensePlateFormatError	0x40001122	Invalid license plate format.
managementNodesNotSupportThisOperation	0x40001123	The operation is not supported.
searchByPictureResourceNotConfiged	0x40001124	The conditions for searching picture by picture are not configured.
videoFileEncapsulationFormatNotSupported	0x40001125	The video container format is not supported.
videoPackageFailure	0x40001126	Converting video container format failed.
videoCodingFormatNotSupported	0x40001127	Video coding format is not supported.
monitorOfDeviceArmingdeleteArmingInfo	0x40001129	The camera is armed. Disarm it and try again.

Sub Status Code	Error Code	Description
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 master node and backup node failed. Check the node status.
theSearchIdDoesNotExist	0x40001133	The search ID does not exist.
theSynchronizationIdDoesNotExist	0x40001134	The synchronization ID does not exist.
theUserIdDoesNotExist	0x40001136	The user ID does not exist.
theIndexCodeDoesNotExist	0x40001138	The index code does not exist.
theControlCenterIdDoesNotExist	0x40001139	The control center ID does not exist.
theAreaIdDoesNotExist	0x4000113A	The area ID does not exist.
theArmingLinkageIdDoesNotExist	0x4000113C	The arming relationship ID does not exist.
theListLibraryIdDoesNotExist	0x4000113D	The list library ID does not exist.
invalidCityCode	0x4000113E	Invalid city code.
synchronizingThePasswordOfSpareServerFailed	0x4000113F	Synchronizing backup system password failed.

Sub Status Code	Error Code	Description
editingStreamingTypesNotSupported	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.
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.

Sub Status Code	Error Code	Description
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.
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.

Sub Status Code	Error Code	Description
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.
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.

Sub Status Code	Error Code	Description
gettingDetailsOfSearch edHigh- frequencyPeopleDetect ionLogsFailed.	0x40001174	Getting the search result details of high frequency alarms failed.
theArmedCamerasAlre adyExistInTheControlC enter	0x40001175	Some cameras in control center are already armed.
disarmingFailedTheCa meralsNotArmed	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.
analysisServicesDoNot Support	0x4000117B	Not supported.
commandAndDispatch ServiceError	0x4000117C	Scheduling service of cloud analytic serice error.
engineModuleError	0x4000117D	Engine module of cloud analytic serice error.
streamingServiceError	0x4000117E	Streaming component of cloud analytic serice error.
faceAnalysisModuleErr or	0x4000117F	Face analysis module of cloud analytic serice error.
vehicleAnalysisModule Error	0x40001180	Vehicle pictures analytic module of cloud analytic serice error.
videoStructuralAnalysis ModuleError	0x40001181	Video structuring module of cloud analytic serice error.
postprocessingModule Error	0x40001182	Post-processing module of cloud analytic serice error.
frequentlyAppearedPe rsonAlarmsIsAlreadyCo nfiguredForListLibrary	0x40001183	High frequency alarm is already armed for blocklist library.
creatingListLibraryFaile d	0x40001184	Creating list library failed.

Sub Status Code	Error Code	Description
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 high frequency alarm failed.
invalidSearchConfiton	0x40001189	Invalid result.
theTaskIsNotCompleted	0x4000118B	The task is not completed.
resourceOverRemainLimit	0x4000118C	No more resource can be pre-allocated.
frequentlyAppearedPersonAlarmsAlreadyConfiguredForTheCameraDisarmFirstAndTryAgain	0x4000118D	The high frequency alarm of this camera is configured. Delete the arming information and try again.
switchtimedifflesslimit	0x4000123b	Time difference between power on and off should be less than 10 minutes.
associatedFaceLibNumOverLimit	0x40001279	Maximum number of linked face picture libraries reached.
noMorePeopleNumChangeRulesAdded	0x4000128A	Maximum number of people number changing rules reached.
noMoreViolentMotionRulesAdded	0x4000128D	Maximum number of violent motion rules reached.
noMoreLeavePositionRulesAdded	0x4000128E	Maximum number of leaving position rules reached.
SMRDiskNotSupportRaid	0x40001291	SMR disk does not support RAID.
OnlySupportHikAndCustomProtocol	0x400012A3	IPv6 camera can only be added via Device Network SDK or custom protocols.

Sub Status Code	Error Code	Description
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.
polygonNotAllowedCrossing	0x400015C2	Polygons are not allowed to cross.
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.
StandbyMmachineCannotBeDeleted	0x400019F2	The hot spare cannot be deleted.
alreadyRunning	0x40002026	The application program is running.

Sub Status Code	Error Code	Description
notRunning	0x40002027	The application program is stopped.
packNotFound	0x40002028	The software packet does not exist.
alreadyExist	0x40002029	The application program already exists.
noMemory	0x4000202A	Insufficient memory.
invalidLicense	0x4000202B	Invalid License.
noClientCertificate	0x40002036	The client certificate is not installed.
noCACertificate	0x40002037	The CA certificate is not installed.
authenticationFailed	0x40002038	Authenticating certificate failed. Check the certificate.
clientCertificateExpired	0x40002039	The client certificate is expired.
clientCertificateRevocation	0x4000203A	The client certificate is revoked.
CACertificateExpired	0x4000203B	The CA certificate is expired.
CACertificateRevocation	0x4000203C	The CA certificate is revoked.
connectFail	0x4000203D	Connection failed.
loginNumExceedLimit	0x4000203F	No more user can log in.
HDMIResolutionIllegal	0x40002040	The HDMI video resolution cannot be larger than that of main and sub stream.
hdFormatFail	0x40002049	Formatting HDD failed.
formattingFailed	0x40002056	Formatting HDD failed.
encryptedFormattingFailed	0x40002057	Formatting encrypted HDD failed.
wrongPassword	0x40002058	Verifying password of SD card failed. Incorrect password.
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.

Sub Status Code	Error Code	Description
completeTheLevelCalibrationFirst	0x4000206A	The level calibration is not set.
completeTheRadarCameraCalibrationFirst	0x4000206B	The radar-camera calibration is not set.
pointsOnStraightLine	0x4000209C	Calibrating failed. The calibration points cannot be one the same line.
TValueLessThanOrEqualZero	0x4000209D	Calibration failed. The T value of the calibration points should be larger than 0.
HBDLibNumOverLimit	0x40002092	The number of human body picture libraries reaches the upper limit
theShieldRegionError	0x40002093	Saving failed. The shielded area should be the ground area where the shielded object is located.
theDetectionAreaError	0x40002094	Saving failed. The detection area should only cover the ground area.
invalidLaneLine	0x40002096	Saving failed. Invalid lane line.
enableITSFunctionOfThisChannelFirst	0x400020A2	Enable ITS function of this channel first.
noCloudStorageServer	0x400020C5	No cloud storage server
NotSupportWithVideoTask	0x400020F3	This function is not supported.
noDetectionArea	0x400050df	No detection area
armingFailed	0x40008000	Arming failed.
disarmingFailed	0x40008001	Disarming failed.
clearAlarmFailed	0x40008002	Clearing alarm failed.
bypassFailed	0x40008003	Bypass failed.
bypassRecoverFailed	0x40008004	Bypass recovery failed.
outputsOpenFailed	0x40008005	Opening relay failed.
outputsCloseFailed	0x40008006	Closing relay failed.
registerTimeOut	0x40008007	Registering timed out.
registerFailed	0x40008008	Registering failed.

Sub Status Code	Error Code	Description
addedByOtherHost	0x40008009	The peripheral is already added by other security control panel.
alreadyAdded	0x4000800A	The peripheral is already added.
armedStatus	0x4000800B	The partition is armed.
bypassStatus	0x4000800C	Bypassed.
zoneNotSupport	0x4000800D	This operation is not supported by the zone.
zoneFault	0x4000800E	The zone is in fault status.
pwdConflict	0x4000800F	Password conflicted.
audioTestEntryFailed	0x40008010	Enabling audio test mode failed.
audioTestRecoveryFailed	0x40008011	Disabling audio test mode failed.
addCardMode	0x40008012	Adding card mode.
searchMode	0x40008013	Search mode.
addRemoterMode	0x40008014	Adding keyfob mode.
registerMode	0x40008015	Registration mode.
exDevNotExist	0x40008016	The peripheral does not exist.
theNumberOfExDevLimited	0x40008017	No peripheral can be added.
sirenConfigFailed	0x40008018	Setting siren failed.
chanCannotRepeatedBinded	0x40008019	This channel is already linked by the zone.
inProgramMode	0x4000801B	The keypad is in programming mode.
inPaceTest	0x4000801C	In pacing mode.
arming	0x4000801D	Arming.
masterSlavesEnable	0x4000802c	The master-slave relationship has taken effect, the slave 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.

Sub Status Code	Error Code	Description
zoneDrawingOutOfRange	0x40008030	The drawn zone is out of detection range.
alarmLineDrawingOutOfRange	0x40008031	The drawn alarm trigger line is out of detection range.
hasTargetInWarningArea	0x40008032	The warning zone already contains targets. Whether to enable mandatory arming?
radarModuleConnectFail	0x40008033	Radar module communication failed.
importCfgFilePasswordErr	0x40008034	Incorrect password for importing configuration files.
overAudioFileNumLimit	0x40008038	The number of audio files exceeds the limit.
audioFileNameIsLong	0x40008039	The audio file name is too long.
audioFormatIsWrong	0x4000803a	The audio file format is invalid.
audioFileIsLarge	0x4000803b	The size of the audio file exceeds the limit.
pircamCapTimeOut	0x4000803c	Capturing of pircam timed out.
pircamCapFail	0x4000803d	Capturing of pircam failed.
pircamIsCaping	0x4000803e	The pircam is capturing.
audioFileHasExisted	0x4000803f	The audio file already exists.
subscribeTypeErr	0x4000a016	This metadata type is not supported to be subscribed.
startAppFail	/	Starting running application program failed.
yuvconflict	/	The raw video stream conflicted.
overMaxAppNum	/	No more application program can be uploaded.
noFlash	/	Insufficient flash.
noFlash	/	The platform mismatches.

StatusCode=5

Sub Status Code	Error Code	Description
badXmlFormat	0x50000001	Invalid XML format.

StatusCode=6

Sub Status Code	Error Code	Description
badParameters	0x60000001	Invalid parameter.
badHostAddress	0x60000002	Invalid host IP address.
badXmlContent	0x60000003	Invalid XML content.
badIPv4Address	0x60000004	Invalid IPv4 address.
badIPv6Address	0x60000005	Invalid IPv6 address.
conflictIPv4Address	0x60000006	IPv4 address conflicted.
conflictIPv6Address	0x60000007	IPv6 address conflicted.
badDomainName	0x60000008	Invalid domain name.
connectSreverFail	0x60000009	Connecting to server failed.
conflictDomainName	0x6000000A	Domain name conflicted.
badPort	0x6000000B	Port number conflicted.
portError	0x6000000C	Port error.
exportErrorData	0x6000000D	Importing data failed.
badNetMask	0x6000000E	Invalid sub-net mask.
badVersion	0x6000000F	Version mismatches.
badDevType	0x60000010	Device type mismatches.
badLanguage	0x60000011	Language mismatches.
incorrentUserNameOrPasswor d	0x60000012	Incorrect user name or password.
invalidStoragePoolOfCloudServ er	0x60000013	Invalid storage pool. The storage pool is not configured or incorrect ID.
noFreeSpaceOfStoragePool	0x60000014	Storage pool is full.
riskPassword	0x60000015	Risky password.
UnSupportCapture	0x60000016	Capturing in 4096*2160 or 3072*2048 resolution is not supported when H.264+ is enabled.

Sub Status Code	Error Code	Description
userPwdLenUnder8	0x60000023	At least two kinds of characters, including digits, letters, and symbols, should be contained in the password.
userPwdNameSame	0x60000025	Duplicated password.
userPwdNameMirror	0x60000026	The password cannot be the reverse order of user name.
beyondARGSRangeLimit	0x60000027	The parameter value is out of limit.
DetectionLineOutOfDetectionRegion	0x60000085	The rule line is out of region.
DetectionRegionError	0x60000086	Rule region error. Make sure the rule region is convex polygon.
DetectionRegionOutOfCountingRegion	0x60000087	The rule region must be marked as red frame.
PedalAreaError	0x60000088	The pedal area must be in the rule region.
DetectionAreaABError	0x60000089	The detection region A and B must be in the a rule frame.
ABRegionCannotIntersect	0x6000008a	Region A and B cannot be overlapped.
customHBPIDError	0x6000008b	Incorrect ID of custom human body picture library
customHBPIDRepeat	0x6000008c	Duplicated ID of custom human body picture library
dataVersionsInHBDLibMismatches	0x6000008d	Database versions mismatches of human body picture library
invalidHBPID	0x6000008e	Invalid human body picture PID
invalidHBDID	0x6000008f	Invalid ID of human body picture library
humanLibraryError	0x60000090	Error of human body picture library

Sub Status Code	Error Code	Description
humanLibraryNumError	0x60000091	No more human body picture library can be added
humanImagesNumError	0x60000092	No more human body picture can be added
noHumanInThePicture	0x60000093	Modeling failed, no human body in the picture
analysisEnginesNoResourceError	0x60001000	No analysis engine.
analysisEnginesUsageExcced	0x60001001	The engine usage is overloaded.
PicAnalysisNoResourceError	0x60001002	No analysis engine provided for picture secondary recognition.
analysisEnginesLoadingError	0x60001003	Initializing analysis engine.
analysisEnginesAbnormaError	0x60001004	Analysis engine exception.
analysisEnginesFacelibImporting	0x60001005	Importing pictures to face picture library. Failed to edit analysis engine parameters.
analysisEnginesAssociatedChannel	0x60001006	The analysis engine is linked to channel.
smdEncodingNoResource	0x60001007	Insufficient motion detection encoding resources.
smdDecodingNoResource	0x60001008	Insufficient motion detection decoding resources.
diskError	0x60001009	HDD error.
diskFull	0x6000100a	HDD full.
facelibDataProcessing	0x6000100b	Handling face picture library data.
capturePackageFailed	0x6000100c	Capturing packet failed.
capturePackageProcessing	0x6000100d	Capturing packet.
noSupportWithPlaybackAbstract	0x6000100e	This function is not supported. Playback by video synopsis is enabled.

Sub Status Code	Error Code	Description
insufficientNetworkBandwidth	0x6000100f	Insufficient network bandwidth.
tapeLibNeedStopArchive	0x60001010	Stop the filing operation of tape library first.
identityKeyError	0x60001011	Incorrect interaction command.
identityKeyMissing	0x60001012	The interaction command is lost.
noSupportWithPersonDensityDetect	0x60001013	This function is not supported. The people density detection is enabled.
ipcResolutionOverflow	0x60001014	The configured resolution of network camera is invalid.
ipcBitrateOverflow	0x60001015	The configured bit rate of network camera is invalid.
tooGreatTimeDifference	0x60001016	Too large time difference between device and server.
noSupportWithPlayback	0x60001017	This function is not supported. Playback is enabled.
channelNoSupportWithSMD	0x60001018	This function is not supported. Motion detection is enabled.
channelNoSupportWithFD	0x60001019	This function is not supported. Face capture is enabled.
illegalPhoneNumber	0x6000101a	Invalid phone number.
illegalCertificateNumber	0x6000101b	Invalid certificate No.
linkedCameraOutLimit	0x6000101c	Connecting camera timed out.
achieveMaxChannelLimit	0x6000101e	No more channels are allowed.
humanMisInfoFilterEnabledChannelNumError	0x6000101f	No more channels are allowed to enable preventing false alarm.
humanEnginesNoResource	0x60001020	Insufficient human body analysis engine resources.
taskNumberOverflow	0x60001021	No more tasks can be added.

Sub Status Code	Error Code	Description
collisionTimeOverflow	0x60001022	No more comparison duration can be configured.
invalidTaskID	0x60001023	Invalid task ID.
eventNotSupport	0x60001024	Event subscription is not supported.
invalidEZVIZSecretKey	0x60001034	Invalid verification code for Hik-Connect.
needDoubleVerification	0x60001042	Double verification required
noDoubleVerificationUser	0x60001043	No double verification user
timeSpanNumOverLimit	0x60001044	Max. number of time buckets reached
channelNumOverLimit	0x60001045	Max. number of channels reached
noSearchIDResource	0x60001046	Insufficient searchID resources
noSupportDeleteStrangerLib	0x60001051	Deleting stranger library is not supported
noSupportCreateStrangerLib	0x60001052	Creating stranger library is not supported
behaviorAnalysisRuleInfoError	0x60001053	Behavior analysis rule parameters error.
safetyHelmetParamError	0x60001054	Hard hat parameters error.
OneChannelOnlyCanBindOneEngine	0x60001077	No more engines can be bound.
engineTypeMismatch	0x60001079	Engine type mismatched.
badUpgradePackage	0x6000107A	Invalid upgrade package.
AudioFileNameDuplicate	0x60001135	Duplicated audio file name.
CurrentAudioFileAIRuleInUseAlreadyDelete	0x60001136	The AI rule linkage related to current audio file has been deleted.
TransitionUseEmmc	0x60002000	Starting device failed. The EMMC is overused.
AdaptiveStreamNotEnabled	0x60002001	The stream self-adaptive function is not enabled.

Sub Status Code	Error Code	Description
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).
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.

Sub Status Code	Error Code	Description
laneAndRegionOverlap	/	The lanes are overlapped.
unitConfigurationNotInEffect	/	Invalid unit parameter.
ruleAndShieldingMaskConflict	/	The line-rule region overlaps with the shielded area.
wholeRuleInShieldingMask	/	There are complete temperature measurement rules in the shielded area.
LogDiskNotSetReadOnlyInGroupMode	0x60001100	The log HDD in the HDD group cannot be set to read-only.
LogDiskNotSetRedundancyInGroupMode	0x60001101	The log HDD in the HDD group cannot be set to redundancy.

StatusCode=7

SubStatusCode	Error Code	Description
rebootRequired	0x70000001	Reboot to take effect.

