



Axxon Next Software Package

Integration SDK (HTTP API)

Version 1.0



EXPERIENCE THE NEXT™

AxxonSoft

Moscow 2012



Contents

1	GENERAL AGREEMENTS	3
2	INFRASTRUCTURE	3
2.1	Get unique identifier	3
2.2	Get server list	3
2.2.1	List of domain servers	3
2.2.2	Server info	3
2.3	Get list of video sources (cameras)	4
2.3.1	Get all available sources	4
2.3.2	Get all available original sources of server	4
2.3.3	Get source info	4
2.3.4	Get all sources info	5
2.4	Get camera live stream	6
2.5	Get camera screenshot	6
2.6	Get archive stream	6
2.6.1	Get archive stream info	7
2.6.2	Control archive stream	7
2.7	Review video footage by frame	7
2.7.1	Get frame by timestamp	7
2.7.2	Get frame registration time	7
2.8	Get MM archive contents	8
2.9	Get info about triggering of detection tools and alarms	9
2.9.1	Get list of alarms	9
2.9.2	Get list of detection tools events	10
2.10	Telemetry control	11
2.10.1	Get list of telemetry devices for specified video source	11
2.10.2	Control degrees of freedom	11
2.10.2.1	Get info of degrees of freedom	11
2.10.2.2	Edit tilt and pan	12
2.10.2.3	Edit degree of freedom	12
2.10.2.4	Capture screen point	13
2.10.2.5	Zoom image area	13
2.10.2.6	Auto focus and iris	13
2.10.3	Preset control	13
2.10.3.1	Get list of presets	13
2.10.3.2	Create and edit preset	14
2.10.3.3	Go to preset and delete preset	14
2.11	Get statistics	14

1 General agreements

HTTP server NGP responds to method calls in the form of JSON.

2 Infrastructure

2.1 Get unique identifier

(UUID) is generated for every GET request to `http://server/prefix/uuid`.

Unique identifier is used to get in last frame info from archive video or to control archived stream.

Response sample:

```
"f03c6ccf-b181-4844-b09c-9a19e6920fd3"
```

2.2 Get server list

2.2.1 List of domain servers

GET `http://server/prefix/hosts/` - gets the list of all domain hosts.

Sample response:

```
[ "SERVER1", "SERVER2" ]
```

2.2.2 Server info

GET `http://server/prefix/hosts/HOSTNAME` - gets host info.

Sample response:

```
{  
  "hostname" : "SERVER2",  
  "domainInfo" :  
  {  
    "domainName" : "DomainName",  
    "domainFriendlyName" : "Custom domain name, if available"  
  },  
  "platformInfo" :  
  {  
    "machine" : "ARM9",  
    "os" : "Linux"  
  }  
}
```

```
},  
"licenseStatus" : "Expired",  
"timeZone" : "+180" // GMT+3  
}
```

2.3 Get list of video sources (cameras)

2.3.1 Get all available sources

GET `http://server/prefix/video-origins/` - gets all available original sources (cameras). The requested identifiers will have the format as follows "HOSTNAME/ObjectType.Id/Endpoint.Name". Friendly name and other related meta data will be received.

Sample response:

```
{  
  "SERVER1/DeviceIpint.3/VideoSource.0:0" :  
    {  
      "origin" : "SERVER1/DeviceIpint.3/VideoSource.0:0",  
      "friendlyNameLong" : "Camera 3",  
      "friendlyNameShort" : "3"  
    },  
  "SERVER2/DeviceIpint.5/VideoSource.0:0" :  
    {  
      "origin" : "SERVER2/DeviceIpint.5/VideoSource.0:0",  
      "friendlyNameLong" : "Camera 5",  
      "friendlyNameShort" : "5"  
    }  
}
```

2.3.2 Get all available original sources of server

GET `http://server/prefix/video-origins/HOSTNAME/` gets all available original sources (cameras) of the specified server.

2.3.3 Get source info

GET `http://server/prefix/video-origins/VIDEOSOURCEID` - gets source information. VIDEOSOURCEID – identifier of endpoint source consisting of 3 components (HOSTNAME/ObjectType.Id/Endpoint.Name).

Sample request:

GET http://server/prefix/video-origins/SERVER1

Sample response:

```
{
  "SERVER1/DeviceIpint.3/VideoSource.0:0" :
  {
    "origin" : "SERVER1/DeviceIpint.3/VideoSource.0:0",
    "friendlyNameLong" : "Camera 3",
    "friendlyNameShort" : "3"
  }
}
```

2.3.4 Get all sources info

GET http://server/prefix/video-sources/* - gets the list of all available sources not limited by original ones.

Sample request:

GET http://server/prefix/video-sources/SERVER2

Sample response:

```
{
  "SERVER2/DeviceIpint.5/VideoSource.0:0" :
  {
    "origin" : "SERVER2/DeviceIpint.5/VideoSource.0:0",
    "friendlyNameLong" : " Camera 5",
    "friendlyNameShort" : "5"
  },
  "SERVER2/VideoDecoder.0/VideoSource" :
  {
    "origin" : "SERVER2/DeviceIpint.5/VideoSource.0:0",
    "friendlyNameLong" : "SERVER2/Videodecoder 0",
    "friendlyNameShort" : "Videodecoder 0"
  }
}
```

}

}

2.4 Get camera live stream

GET <http://server/prefix/live/media/VIDEOSOURCEID?parameters>.

Parameters:

format – parameter values are "mjpeg", "webm" or "h264". The native format is selected by server to prevent additional encoding in case no format is specified or recognized. If a native format is not supported by client, server selects WebM.

w – frame width.

h – frame height.

Sample request:

GET <http://server/prefix/live/media/TEST-1/DeviceIpint.23/VideoSource.0:0?format=mjpeg&w=640&h=480>

2.5 Get camera screenshot

GET <http://server/prefix/live/media/snapshot/VIDEOSOURCEID?parameters>.

Parameters:

w – frame width.

h – frame height.

Sample request:

GET <http://server/prefix/live/media/snapshot/TEST-1/DeviceIpint.23/VideoSource.0:0>

GET <http://server/prefix/live/media/snapshot/TEST-1/DeviceIpint.23/VideoSource.0:0?w=640&h=480>

2.6 Get archive stream

GET <http://server/prefix/archive/media/VIDEOSOURCEID/STARTTIME?parameters>,

where **STARTTIME** – time in ISO format.

Parameters:

speed – playback speed, can be negative.

format - parameter values are "mjpeg", "webm" or "h264". The native format is selected by server to prevent additional encoding in case no format is specified or recognized. If a native format is not supported by client, server selects WebM.

If both parameters are not specified, the speed is equal to 0, JPEG format is selected and the request is handled as a request to review video footage by frames.

id – unique identifier of archive stream (optional). Used to get stream info or control the stream.

w – frame width.

h – frame height.

Sample request:

```
GET http://server/prefix/archive/media/TEST-1/DeviceIpint.23/VideoSource.0:0/20110608T060141.375?speed=1&w=640&h=480
```

2.6.1 Get archive stream info

```
GET http://server/prefix/archive/media/rendered-info/UUID -gets info of the frame last displayed.
```

where UUID is a unique identifier of the requested archive stream.

The following frame info is available:

timestamp – frame time token.

Sample request:

```
GET http://localhost:8000/asip-api/archive/media/rendered-info/%22996cea31-91c4-9a46-9269-48b998fd2f29%22
```

Sample response:

```
{
  "timestamp": "20110408T103627.048"
}
```

2.6.2 Control archive stream

```
GET http://server/prefix/archive/media/stop/UUID - stops archive stream that has the specified UUID.
```

When completed, the last frame info is received.

2.7 Review video footage by frame

2.7.1 Get frame by timestamp

```
GET http://server/prefix/archive/media/VIDEOSOURCEID/STARTTIME - gets frame by its STARTTIME.
Frame is returned in JPEG format.
```

2.7.2 Get frame registration time

```
GET http://server/prefix/archive/contents/frames/VIDEOSOURCEID/ENDTIME/BEGINTIME?limit=COUNT
– gets the time of frame registration in MM archive. Parameter semantics is described in section Get MM archive contents. The default value of limit parameter is 250. This parameter is optional for server and it can return fewer search results.
```

The **frames** property will contain an array of frame timestamps returned as json object.

The returned json response will contain the **more** Boolean property, which is used to specify if complete time interval is selected (false) or some frames were not returned because their timestamps maxed out.

Sample request:

GET

http://server/prefix/archive/contents/frames/SERVER1/DeviceIpint.2/VideoSource.0:0/20101230T103943.000/20101230T103952.000?limit=3

Sample response:

```
{
  "frames" :
  [ "20101230T103951.800", "20101230T103951.760", "20101230T103951.720" ],
  "more" : false
}
```

2.8 Get MM archive contents

GET

http://server/prefix/archive/contents/intervals/VIDEOSOURCEID/ENDTIME/BEGINTIME?limit=COUNT&scale=SIZE - gets archive contents starting from BEGINTIME and ending up with ENDTIME.

If BEGINTIME is not specified, infinite future is considered. If ENDTIME is not specified, infinite past is considered. Words “past” and “future” can be used to set infinite past and infinite future as well.

Optional *limit* parameter is used to specify the limit of frames. The default value is **100**.

Optional *scale* parameter is used to specify the minimum time interval between the frames when they are treated as separate ones (not combined). The default value is **0**.

Interval sequence corresponds to the ratio between specified BEGINTIME and ENDTIME (in ascending order if BEGINTIME<ENDTIME, and in descending order if ENDTIME<BEGINTIME). Start and end points of interval are returned in its common order, i.e.the interval start time is less than the interval end time or equal to it.

The **intervals** property contains array of frame intervals returned as json object.

The returned json response contains the Boolean **more** property which is used to specify if complete time interval is selected (false) or some frames were not returned because their timestamps maxed out (true).

Sample request:

GET

http://server/prefix/archive/contents/intervals/SERVER1/DeviceIpint.2/VideoSource.0:0/20101230T103904.000/20101230T103959.000?limit=3

Sample response:

```
{
  "intervals" :
```

```
[
  { begin: "20101230T103950.000", end: "20101230T103955.230" },
  { begin: "20101230T103923.110", end: "20101230T103941.870" }
],
"more" : true
}
```

2.9 Get info about triggering of detection tools and alarms

2.9.1 Get list of alarms

GET

<http://server/prefix/archive/events/alerts/VIDEOSOURCEID/ENDTIME/BEGINTIME?limit=COUNT&offset=COUNT> – gets the list of alarms. If *limit* is not specified, it is equal to 100. Field **raisedAt** is not unique so passing of previously received alarms starting from the search interval can be requested.

Sample response:

```
{
  "events" :
  [
    {
      "type": "alert",
      "id": "42D43A79-90D6-4ba7-BD23-1714996A2F88",
      "raisedAt": "20101230T103950.000",
      "zone": "SERVER1/DeviceIpint.3/VideoSource.0:0",
      "reasons": ["ruleAlert", "videoDetector"],
      "initiator": "4359EC93-EF31-4de0-9EDE-AA5C5803D6F8",
      "reaction":
      {
        "user": "root",
        "reactedAt": "20101230T103958.000",
        "severity": "alarm"
      }
    },
  ],
}
```

```

...
],
"more": true
}

```

Possible values of **reasons** array: armed, disarmed, userAlert, ruleAlert, videoDetector, audioDetector, ray.

Possible values of field **severity**: unclassified, false, notice, warning, alarm.

2.9.2 Get list of detection tools events

GET

<http://server/prefix/archive/events/detectors/VIDEOSOURCEID/ENDTIME/BEGINTIME?limit=COUNT&ofset=COUNT> – gets the list of detection tool events. If *limit* is not specified, it is equal to 100. Field **timestamp** is not unique so passing of previously received alarms starting from the search interval can be requested.

Sample response:

```

{
  "events":
  [
    {
      "id": "433d45ec-0b7f-aa43-8491-c8acb7d0ac56"
      ,"source": "hosts/SERVER1/SituationDetector.0"
      ,"origin": "hosts/SERVER1/DeviceIpint.0/SourceEndpoint.video:0:0"
      ,"detectorId": "1"
      ,"type": "CrossOneLine"
      ,"alertState": "ended"
      ,"timestamp": "20120314T121512.597"
      ,"rectangles":
      [
        {
          "index": "1"
          ,"left": "0.622086710929871"
          ,"top": "0.68798337459564196"

```

```

    , "right": "0.65736908435821495"
    , "bottom": "0.79889315128326399"
  }
]
},
...
],
"more": true
}

```

2.10 Telemetry control

2.10.1 Get list of telemetry devices for specified video source

GET <http://server/prefix/control/telemetry/list/OBJECTID> - Gets the list of telemetry devices for video sources where OBJECTID is object identifier based on 2 components (HOSTNAME/ObjectType.Id).

Sample response:

```

[
  "SERVER1/DeviceIpint.2/TelemetryControl.0"
]

```

TELEMETRYCONTROLID template will be used to define telemetry devices of theHOSTNAME/ObjectType.Id/TelemetryContol.n type hereinafter.

2.10.2 Control degrees of freedom

2.10.2.1 Get info of degrees of freedom

GET <http://server/prefix/control/telemetry/info/TELEMETRYCONTROLID> - Gets info about supported controllable degrees of freedom and the ways of controlling them (continuous, relative, discrete) and their max values.

Sample response:

```

{
  "degrees":
  {
    "tilt":
    {

```

```

    "relative": {"min": "-45", "max": "45"},
    "continuous": {"min": "-10", "max": "10"}
  },
  "pan":
  {
    "absolute": {"min": "-170", "max": "170"},
    "continuous": {"min": "-10", "max": "10"}
  },
  "zoom":
  {
    "absolute": {"min": "0", "max": "20"}
  }
},
"feature": ["autoFocus", "areaZoom", "pointMove"]
}

```

degrees – information about degrees of freedom. (tilt, pan, zoom, focus, iris). Every degree of freedom contains the list of supported ways of control (absolute, relative, continuous).

feature – list of supported functions (autoFocus, autoIris, areaZoom, pointMove).

2.10.2.2 Edit tilt and pan

GET <http://server/prefix/control/telemetry/move/TELEMETRYCONTROLID?parameters> – changes tilt, pan.

Parameters:

mode – way of control (absolute, relative, continuous);

pan, tilt – values for corresponding degrees of freedom.

Sample request:

GET <http://server/prefix/control/telemetry/move/TEST-1/DeviceIrpint.25/TelemetryControl.0?mode=absolute&pan=-99&tilt=10>

2.10.2.3 Edit degree of freedom

GET <http://server/prefix/control/telemetry/{степень}/TELEMETRYCONTROLID?parameters> – changes one of degrees (zoom, focus, iris).

Parameters:

{degree} – degree of freedom to be updated (zoom, focus, iris);

mode – way of control (absolute, relative, continuous);

value - value.

Sample request:

GET http://server/prefix/control/telemetry/zoom/TEST-1/DeviceIpint.25/TelemetryControl.0?mode=absolute&value=6 - zoom change;
GET http://server/prefix/control/telemetry/focus/TEST-1/DeviceIpint.25/TelemetryControl.0?mode=relative&value=3 - focus change;
GET http://server/prefix/control/telemetry/iris/TEST-1/DeviceIpint.25/TelemetryControl.0?mode=continuous&value=1 - iris change.

2.10.2.4 Capture screen point

GET http://server/prefix/control/telemetry/move/point/TELEMETRYCONTROLID?parameters – captures the screen point.

Parameter:

x,y – values of vertical and horizontal coordinates, specified in relation to image size;

Sample request:

GET http://server/prefix/control/telemetry/move/point/TEST-1/DeviceIpint.23/TelemetryControl.0?x=0.14&y=0.32

2.10.2.5 Zoom image area

GET http://server/prefix/control/telemetry/zoom/area/TELEMETRYCONTROLID?parameters – zooms selected image area.

Parameters:

x,y – left upper corner of selected area;

w,h – width and height of area.

Coordinates and dimensions are specified in relation to image size.

Sample request:

GET http://server/prefix/control/telemetry/zoom/area/TEST-1/DeviceIpint.24/TelemetryControl.0?x=0.23&y=0.089&w=0.25&h=0.25

2.10.2.6 Auto focus and iris

GET http://server/prefix/control/telemetry/auto/TELEMETRYCONTROLID?parameters – auto focus/iris.

Parameters:

degree – is focus or iris.

Sample request:

GET http://server/prefix/control/telemetry/auto/TEST-1/DeviceIpint.24/TelemetryControl.0?degree=iris

2.10.3 Preset control**2.10.3.1 Get list of presets**

GET http://server/prefix/control/telemetry/preset/info/TELEMETRYCONTROLID- gets the list of existing presets.

Sample request:

GET http://server/prefix/control/telemetry/preset/info/TEST-1/DeviceIpint.23/TelemetryControl.0

Sample response:

```
{  
  "0": "Corridor",  
  "1": "Entrance",  
  "4": "Hole in fence"  
}
```

2.10.3.2 Create and edit preset

GET <http://server/prefix/control/telemetry/preset/set/TELEMETRYCONTROLID?parameters> - create/edit preset.

Parameters:

pos - position;

label – preset name.

If a preset with specified position already exists, its label will be deleted.

Sample request:

GET <http://server/prefix/control/telemetry/preset/set/TEST-1/DeviceIpint.23/TelemetryControl.0?pos=0&label=Exit>

2.10.3.3 Go to preset and delete preset

GET <http://server/prefix/control/telemetry/preset/{action}/TELEMETRYCONTROLID?parameters> - go to preset or delete it.

Parameters:

{action} – can be **go** or **remove** and is used to go to preset or delete it;

pos – preset position.

Sample request:

Going to preset with pos 1:

GET <http://server/prefix/control/telemetry/preset/go/TEST-1/DeviceIpint.23/TelemetryControl.0?pos=1>
– this request deletes preset with position 2:

GET <http://server/prefix/control/telemetry/preset/remove/TEST-1/DeviceIpint.23/TelemetryControl.0?pos=2>

2.11 Get statistics

GET <http://server/prefix/statistics/VIDEOSOURCEID> - gets statistics for the specified video source.

GET <http://server/prefix/statistics/webserver> - gets statistics for server.