

Recording

SUNAPI

v2.6.2

2023-04-07



Copyright

© 2023 Hanwha Vision Co., Ltd. All rights reserved.

Restriction

Do not copy, distribute, or reproduce any part of this document without written approval from Hanwha Vision Co., Ltd.

Disclaimer

Hanwha Vision Co., Ltd. has made every effort to ensure the completeness and accuracy of this document, but makes no guarantee as to the information contained herein. All responsibility for proper and safe use of the information in this document lies with users. Hanwha Vision Co., Ltd. may revise or update this document without prior notice.

Contact Information

Hanwha Vision Co., Ltd.

Hanwha Vision 6, Pangyo-ro 319beon-gil, Bundang-gu, Seongnam-si, Gyeonggi-do, 13488, KOREA
www.hanwhavision.com

Hanwha Vision America

500 Frank W. Burr Blvd. Suite 43 Teaneck, NJ 07666
hanwhavisionamerica.com

Hanwha Vision Europe

Heriot House, Heriot Road, Chertsey, Surrey, KT16 9DT, United Kingdom
hanwhavision.eu

Hanwha Vision Middle East FZE

Jafza View 18, Office 2001-2003, Po Box 263572, Jebel Ali Free Zone, Dubai, United Arab Emirates
www.hanwhavision.com/ar

Table of Contents

| | |
|---|----|
| 1. Overview | 6 |
| 1.1. Description | 6 |
| 2. Storage | 7 |
| 2.1. Description | 7 |
| 2.2. Syntax | 7 |
| 2.3. Parameters | 7 |
| 2.4. Examples | 7 |
| 2.4.1. Getting the current information | 8 |
| 2.4.2. Enabling video storage | 10 |
| 2.4.3. Enabling overwriting | 10 |
| 3. General | 11 |
| 3.1. Description | 11 |
| 3.2. Syntax | 11 |
| 3.3. Parameters | 11 |
| 3.4. Examples | 13 |
| 3.4.1. Getting the current information | 13 |
| 3.4.2. Setting the normal record mode to Full | 18 |
| 3.4.3. Setting the event record mode to I-Frame | 18 |
| 3.4.4. Setting the duration of pre-event recording to 3 seconds | 18 |
| 3.4.5. Setting the recording video file format to AVI | 18 |
| 3.4.6. Setting the source profile | 18 |
| 4. Recording Schedule | 19 |
| 4.1. Description | 19 |
| 4.2. Syntax | 19 |
| 4.3. Parameters | 19 |
| 4.4. Examples | 23 |
| 4.4.1. Getting the current information | 23 |
| 4.4.2. Setting the device to always record | 27 |
| 4.4.3. Setting the recording schedule | 27 |
| 5. Overlapped Recording | 29 |
| 5.1. Description | 29 |
| 5.2. Syntax | 29 |
| 5.3. Parameters | 29 |
| 5.4. Examples | 30 |
| 5.4.1. Getting the overlapped recording between 12:00 AM and 12:00 PM on Aug. 1, 2014 | 30 |
| 6. Manual Recording | 32 |

| | |
|--|----|
| 6.1. Description | 32 |
| 6.2. Syntax | 32 |
| 6.3. Parameters | 32 |
| 6.4. Examples | 32 |
| 6.4.1. Getting the current manual recording status | 32 |
| 6.4.2. Starting the manual recording | 33 |
| 6.4.3. Stopping the manual recording | 33 |
| 7. Calendar Search | 34 |
| 7.1. Description | 34 |
| 7.2. Syntax | 34 |
| 7.3. Parameters | 34 |
| 7.4. Examples | 35 |
| 7.4.1. Searching for recordings from February 2013 | 35 |
| 8. Timeline | 37 |
| 8.1. Description | 37 |
| 8.2. Syntax | 37 |
| 8.3. Parameters | 37 |
| 8.4. Examples | 39 |
| 8.4.1. Getting the timeline | 39 |
| 9. Recording Period | 44 |
| 9.1. Description | 44 |
| 9.2. Syntax | 44 |
| 9.3. Parameters | 44 |
| 9.4. Examples | 44 |
| 9.4.1. Searching the recording period | 44 |
| 10. Heat Map Search | 46 |
| 10.1. Description | 46 |
| 10.2. Syntax | 46 |
| 10.3. Parameters | 46 |
| 10.4. Examples | 50 |
| 10.4.1. Heat map search color | 50 |
| 10.4.2. Heat map search | 51 |
| 10.4.3. Getting the status of searching | 53 |
| 10.4.4. Getting the result | 54 |
| 10.4.5. Cancelling the search | 55 |
| 11. People Count Search | 56 |
| 11.1. Description | 56 |
| 11.2. Syntax | 56 |
| 11.3. Parameters | 56 |
| 11.4. Examples | 58 |

| | |
|--|----|
| 11.4.1. People Count Search | 58 |
| 11.4.2. Getting the status of searching | 59 |
| 11.4.3. Getting the result | 60 |
| 11.4.4. Cancelling the search | 61 |
| 12. POS Configuration | 62 |
| 12.1. Description | 62 |
| 12.2. Syntax | 62 |
| 12.3. Parameters | 62 |
| 12.4. Examples | 64 |
| 13. POS Event Configuration | 68 |
| 13.1. Description | 68 |
| 13.2. Syntax | 68 |
| 13.3. Parameters | 68 |
| 13.4. Examples | 69 |
| 14. POS Data | 70 |
| 14.1. Description | 70 |
| 14.2. Syntax | 70 |
| 14.3. Parameters | 70 |
| 14.4. Examples | 70 |
| 15. POS Calendar | 73 |
| 15.1. Description | 73 |
| 15.2. Syntax | 73 |
| 15.3. Parameters | 73 |
| 15.4. Examples | 73 |
| 16. Meta Data Search | 75 |
| 16.1. Description | 75 |
| 16.2. Syntax | 75 |
| 16.3. Parameters | 75 |
| 16.4. Examples | 78 |
| 16.4.1. Start search for POS data | 78 |
| 16.4.2. Cancel Search | 80 |
| 16.4.3. To get search status | 81 |
| 16.4.4. To renew search token (For 1 minute) | 81 |
| 16.4.5. To get the results of search (First 100 results) | 82 |
| 16.4.6. To get the results of search (Next 100 results) | 85 |
| 17. Smart Search | 86 |
| 17.1. Description | 86 |
| 17.2. Syntax | 86 |
| 17.3. Parameters | 86 |
| 17.4. Examples | 88 |

| | |
|--|-----|
| 17.4.1. Start smart search for selected area | 88 |
| 17.4.2. Check the status of smart search | 89 |
| 17.4.3. To get results of smart search | 89 |
| 17.4.4. Start Smart search for the lines | 91 |
| 18. Queue Search | 92 |
| 18.1. Description | 92 |
| 18.2. Syntax | 92 |
| 18.3. Parameters | 92 |
| 18.4. Examples | 95 |
| 18.4.1. Queue Search | 95 |
| 18.4.2. Getting the status of Queue search | 97 |
| 18.4.3. Getting Queue search result | 97 |
| 18.4.4. Cancelling Queue search | 105 |
| 19. Disk Utility | 106 |
| 19.1. Description | 106 |
| 19.2. Syntax | 106 |
| 19.3. Parameters | 106 |
| 19.4. Examples | 106 |
| 19.4.1. Getting the disk information | 106 |
| 20. Bookmark | 110 |
| 20.1. Description | 110 |
| 20.2. Syntax | 110 |
| 20.3. Parameters | 110 |
| 20.4. Examples | 112 |
| 20.4.1. Adding a bookmark | 112 |
| 20.4.2. Removing a bookmark | 113 |
| 20.4.3. Updating a bookmark | 113 |
| 20.4.4. Viewing a bookmark | 114 |
| 21. Event Search | 116 |
| 21.1. Description | 116 |
| 21.2. Syntax | 116 |
| 21.3. Parameters | 116 |
| 21.4. Examples | 119 |
| 21.4.1. Event search | 119 |
| 21.4.2. Viewing search result status | 119 |
| 21.4.3. Viewing search result | 120 |

Chapter 1. Overview

1.1. Description

This document explains recording.cgi.

recording.cgi configures video recording settings so that video can be recorded at a scheduled time or when an event occurs. It also provides submenus for configuring storage and searching recorded videos.

The following submenus of recording.cgi are used:

- **storage**: Sets the storage configuration, including whether to overwrite or auto-delete.
- **general**: Sets the recording mode and the duration of pre/post-event recording.
- **recordingschedule**: Sets the specific schedule for recording.
- **overlapped**: Requests the recordings are overlapped within the given time range
- **manualrecording**: Starts or stops the manual recording.
- **calendarsearch**: Requests for recording information from a month.
- **timeline**: Requests for recording information from a given time period.
- **searchrecordingperiod**: Requests a given recording period.
- **heatmapsearch**: Requests recording information using heat map search function and controls the heat map search settings.
- **peoplecountsearch**: Requests recording information using people count search function, and controls the people count search settings.
- **posconf**: Used for configuring POS device.
- **posevntconf**: Used to configure POS event in the device.
- **posdata**: Used to receive the live POS data from the device.
- **poscalendar**: Used to get the availability of POS data in a given month.
- **metadata**: Used to search POS data for some given search criteria.
- **smartsearch**: Used to search video analytics information for given search criteria.
- **queuesearch**: Requests statistical analysis and measurement of average dwell time and number of people in queues based on given search criteria.
- **diskutility**: Requests information on the NVR's disk array and smart attributes of HDD.
- **bookmark**: Used to manage bookmarks in NVR.
- **eventsearch**: Used to search event data for some given search criteria.

Chapter 2. Storage

2.1. Description

The **storage** submenu requests and configures storage settings.

Access level

| Action | Camera | NVR |
|--------|--------|------|
| view | Admin | User |
| set | Admin | User |

2.2. Syntax

```
http://<Device IP>/stw-cgi/recording.cgi?msubmenu=  
storage&action=<value> [&<parameter>=<value>]
```

2.3. Parameters

| Action | Parameters | Request/ Response | Type/ Value | Description |
|--------|------------------|----------------------|-----------------------|--|
| view | | | | Reads the storage settings. |
| set | Channel | REQ, RES | <int> | Channel ID NVR ONLY |
| | Enable | REQ, RES | <bool> True, False | Enables or disables video storage |
| | OverWrite | REQ, RES | <bool> True, False | Enables or disables overwriting |
| | DiskEndBeep | REQ, RES | <bool> True, False | Whether to beep when the disk ends NVR ONLY |
| | AutoDeleteEnable | REQ, RES | <bool> True, False | Whether to delete videos older than the specified number of days |
| | AutoDeleteDays | REQ, RES | <int> | Days before auto deletion This parameter is valid only when AutoDeleteEnable is set to True. |

2.4. Examples

2.4.1. Getting the current information

REQUEST

```
http://<Device IP>/stw-cgi/recording.cgi?msubmenu=storage&action=view
```

TEXT RESPONSE

```
HTTP/1.0 200 OK  
Content-type: text/plain  
<Body>
```

```
Enable=True  
OverWrite=False  
AutoDeleteEnable=False  
AutoDeleteDays=180
```

JSON RESPONSE

```
HTTP/1.0 200 OK  
Content-type: application/json  
<Body>
```

```
{  
  "Enable": true,  
  "OverWrite": false,  
  "AutoDeleteEnable": false,  
  "AutoDeleteDays": 180  
}
```

The following request example is for NVR only.

REQUEST

```
http://<Device IP>/stw-cgi/recording.cgi?msubmenu=storage&action=view
```

TEXT RESPONSE

```
HTTP/1.0 200 OK  
Content-type: text/plain
```

<Body>

```
Enable=True
OverWrite=True
DiskEndBeep=False
AutoDeleteEnable=True
Channel.0.AutoDeleteDays=400
Channel.1.AutoDeleteDays=400
Channel.2.AutoDeleteDays=400
Channel.3.AutoDeleteDays=400
Channel.4.AutoDeleteDays=400
...
```

JSON RESPONSE

HTTP/1.0 200 OK
Content-type: application/json
<Body>

```
{
  "Enable": true,
  "OverWrite": true,
  "DiskEndBeep": false,
  "AutoDeleteEnable": true,
  "ChannelwiseAutoDeleteDays": [
    {
      "Channel": 0,
      "AutoDeleteDays": 400
    },
    {
      "Channel": 1,
      "AutoDeleteDays": 400
    },
    {
      "Channel": 2,
      "AutoDeleteDays": 400
    },
    {
      "Channel": 3,
      "AutoDeleteDays": 400
    }
  ]
}
```

```
    },  
    {  
        "Channel": 4,  
        "AutoDeleteDays": 400  
    }  
]  
}
```

2.4.2. Enabling video storage

REQUEST

```
http://<Device IP>/stw-  
cgi/recording.cgi?submenu=storage&action=set&Enable=True
```

2.4.3. Enabling overwriting

REQUEST

```
http://<Device IP>/stw-  
cgi/recording.cgi?submenu=storage&action=set&OverWrite=True
```

Chapter 3. General

3.1. Description

The **general** submenu requests and configures general recording settings.

NOTE | Attribute to check for Recording Support: "attributes/Recording/Support/Recording"

Access level

| Action | Camera | NVR |
|--------|--------|------|
| view | Admin | User |
| set | Admin | User |

3.2. Syntax

```
http://<Device IP>/stw-cgi/recording.cgi?msubmenu=  
general&action=<value> [&<parameter>=<value>]
```

3.3. Parameters

| Action | Parameters | Request/ Response | Type/ Value | Description |
|--------|--------------------|----------------------|---------------------------------------|---|
| view | | | | Reads the general recording settings. |
| | Channel | REQ, RES | <csv> | Channel ID |
| | FullFrameBandwidth | RES | <float> | Full frame bandwidth NVR ONLY |
| | FullFrameRate | RES | <float> | Full frame rate NVR ONLY |
| | KeyFrameBandWidth | RES | <float> | Key frame bandwidth NVR ONLY |
| | KeyFrameRate | RES | <float> | Key frame rate NVR ONLY |
| | Codec | RES | <enum> MJPEG, MPEG4, H264, H265 | Codec NVR ONLY |

| Action | Parameters | Request/ Response | Type/ Value | Description |
|--------|---------------------|----------------------|--|--|
| | RecordOverlap | RES | <csv> AudioDetection, VideoAnalysis, AlarmInput, Normal, MotionDetection, Manual, DefocusDetection, Tracking, FogDetection, AudioAnalysis, EmergencyTrigger, GSensorEvent, MaskDetection | Record overlap NVR ONLY |
| | Resolution | RES | <string> | Resolution NVR ONLY |
| | FrameRate | RES | <int> | Frame rate NVR ONLY |
| | CompressionLevel | RES | <int> | Compression level NVR ONLY |
| | SubStreamCodec | RES | <enum> MJPEG, MPEG4, H264, H265 | Substream codec NVR ONLY |
| | SubStreamResolution | RES | <string> | Substream resolution NVR ONLY |
| | SubStreamFrameRate | RES | <int> | SubStream framerate NVR ONLY |
| set | Channel | REQ, RES | <int> | Channel ID |
| | SourceProfile | REQ, RES | <string> | Source profile NVR ONLY |
| | NormalMode | REQ, RES | <enum> I-Frame, Full, Off | Recording type for normal mode (continuous recording) |
| | EventMode | REQ, RES | <enum> I-Frame, Full, Off | Recording type for the occurrence of an event |
| | PreEventDuration | REQ, RES | <enum> | Duration of pre-event recording |
| | PostEventDuration | REQ, RES | <enum> | Duration of post-event recording |

| Action | Parameters | Request/Response | Type/Value | Description |
|--------|------------------------|------------------|-----------------------|---|
| | RecordedVideoFileType | REQ, RES | <enum> STW, AVI | Recording file format <ul style="list-style-type: none"> STW: Hanwha Vision's proprietary file format played with Web viewer and SD memory player AVI: General AVI video file format played with Web viewer and Windows Media Player. CAMERA ONLY |
| | AudioEnable | REQ, RES | <bool> True, False | Enables or disables audio NVR ONLY |
| | BitrateLimit | REQ, RES | <float> | Bitrate limit NVR ONLY |
| | SubStreamEnable | REQ, RES | <bool> True, False | Enables or disables substream recording on a channel. NVR ONLY |
| | SubStreamSourceProfile | REQ, RES | <string> | Substream sourceprofile NVR ONLY |

3.4. Examples

3.4.1. Getting the current information

REQUEST

```
http://<Device IP>/stw-cgi/recording.cgi?submenu=general&action=view
```

TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
<Body>
```

```
Channel=0
NormalMode=Off
EventMode=Full
```

```
PreEventDuration=3s
PostEventDuration=5s
RecordedVideoFileType=AVI
```

JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
  "RecordSetup": [
    {
      "Channel": 0,
      "NormalMode": "Off",
      "EventMode": "Full",
      "PreEventDuration": "3s",
      "PostEventDuration": "5s",
      "RecordedVideoFileType": "AVI"
    }
  ]
}
```

The following request example is for NVR only.

REQUEST

```
http://<Device IP>/stw-cgi/recording.cgi?msubmenu=general&action=view
```

TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
<Body>
```

```
Channel.0.FullFrameBandWidth=3.965550
Channel.0.FullFrameRate=23.500000
Channel.0.KeyFrameBandWidth=0.850044
Channel.0.KeyFrameRate=0.500000
Channel.0.Codec=H264
```

```
Channel.0.RecordOverlap=Normal
Channel.0.SourceProfile=H.264
Channel.0.NormalMode=Full
Channel.0.EventMode=Full
Channel.0.PreEventDuration=5s
Channel.0.PostEventDuration=30s
Channel.0.Resolution=2560x1920
Channel.0.FrameRate=30
Channel.0.CompressionLevel=10
Channel.0.AudioEnable=False
Channel.0.BitrateLimit=2.300000
Channel.0.SubStreamEnable=True
Channel.0.SubStreamSourceProfile=Live4NVR
Channel.0.SubStreamCodec=H264
Channel.0.SubStreamResolution=800x600
Channel.0.SubStreamFrameRate=30
Channel.1.FullFrameBandWidth=1.157300
Channel.1.FullFrameRate=19.960000
Channel.1.KeyFrameBandWidth=0.359871
Channel.1.KeyFrameRate=0.500000
Channel.1.Codec=H264
Channel.1.RecordOverlap=Normal
Channel.1.SourceProfile=Rec4NVR1
Channel.1.NormalMode=Full
Channel.1.EventMode=Full
Channel.1.PreEventDuration=5s
Channel.1.PostEventDuration=30s
Channel.1.Resolution=1920x1080
Channel.1.FrameRate=20
Channel.1.CompressionLevel=10
Channel.1.AudioEnable=False
Channel.1.BitrateLimit=2.300000
Channel.1.SubStreamEnable=True
Channel.1.SubStreamSourceProfile=Live4NVR1
Channel.1.SubStreamCodec=H264
Channel.1.SubStreamResolution=800x600
Channel.1.SubStreamFrameRate=20
Channel.2.FullFrameBandWidth=0.780083
Channel.2.FullFrameRate=25.000000
Channel.2.KeyFrameBandWidth=0.671478
Channel.2.KeyFrameRate=1.910000
```



```
Channel.2.Codec=H264
Channel.2.RecordOverlap=Normal
Channel.2.SourceProfile=H.264
Channel.2.NormalMode=Full
Channel.2.EventMode=Full
Channel.2.PreEventDuration=5s
Channel.2.PostEventDuration=30s
Channel.2.Resolution=320x240
Channel.2.FrameRate=25
Channel.2.CompressionLevel=16
Channel.2.AudioEnable=False
Channel.2.Bitratelimit=2.300000
Channel.2.SubStreamEnable=True
Channel.2.SubStreamSourceProfile=VideoProfile5
Channel.2.SubStreamCodec=H265
Channel.2.SubStreamResolution=3840x2160
Channel.2.SubStreamFrameRate=25
...
```

JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
  "RecordSetup": [
    {
      "Channel": 0,
      "FullFrameBandWidth": 3.416850,
      "FullFrameRate": 17.530000,
      "KeyFrameBandWidth": 1.058580,
      "KeyFrameRate": 0.350000,
      "Codec": "H264",
      "RecordOverlap": [
        "Normal"
      ],
      "SourceProfile": "H.264",
      "NormalMode": "Full",
      "EventMode": "Full",
      "PreEventDuration": "5s",
```

```

        "PostEventDuration": "30s",
        "Resolution": "2560x1920",
        "FrameRate": 30,
        "CompressionLevel": 10,
        "AudioEnable": false,
        "BitrateLimit": 2.300000,
        "SubStreamEnable": true,
        "SubStreamSourceProfile": "Live4NVR",
        "SubStreamCodec": "H264",
        "SubStreamResolution": "800x600",
        "SubStreamFrameRate": "30"
    },
    {
        "Channel": 1,
        "FullFrameBandWidth": 1.157300,
        "FullFrameRate": 19.960000,
        "KeyFrameBandWidth": 0.359871,
        "KeyFrameRate": 0.500000,
        "Codec": "H264",
        "RecordOverlap": [
            "Normal"
        ],
        "SourceProfile": "Rec4NVR1",
        "NormalMode": "Full",
        "EventMode": "Full",
        "PreEventDuration": "5s",
        "PostEventDuration": "30s",
        "Resolution": "1920x1080",
        "FrameRate": 20,
        "CompressionLevel": 10,
        "AudioEnable": false,
        "BitrateLimit": 2.300000,
        "SubStreamEnable": true,
        "SubStreamSourceProfile": "Live4NVR1",
        "SubStreamCodec": "H264",
        "SubStreamResolution": "800x600",
        "SubStreamFrameRate": "20"
    },
    ...
]
}

```

3.4.2. Setting the normal record mode to Full

REQUEST

```
http://<Device IP>/stw-  
cgi/recording.cgi?msubmenu=general&action=set&NormalMode=Full
```

3.4.3. Setting the event record mode to I-Frame

REQUEST

```
http://<Device IP>/stw-  
cgi/recording.cgi?msubmenu=general&action=set&EventMode=I-Frame
```

The following request example is for NVR only.

REQUEST

```
http://<Device IP>/stw-  
cgi/recording.cgi?msubmenu=general&action=set&Channel=1&EventMode=I-Frame
```

3.4.4. Setting the duration of pre-event recording to 3 seconds

REQUEST

```
http://<Device IP>/stw-  
cgi/recording.cgi?msubmenu=general&action=set&PreEventDuration=3s
```

3.4.5. Setting the recording video file format to AVI

REQUEST

```
http://<Device IP>/stw-  
cgi/recording.cgi?msubmenu=general&action=set&RecordedVideoFileType=AVI
```

3.4.6. Setting the source profile

The following request example is for NVR only.

REQUEST

```
http://<Device IP>/stw-  
cgi/recording.cgi?msubmenu=general&action=set&Channel=7&SourceProfile=Profile2
```

Chapter 4. Recording Schedule

4.1. Description

The **recordingschedule** submenu requests and configures recording schedule settings.

This submenu is only for normal recording, not for event recording. To set the recording schedule for events, please refer to the 'Event' document.

Access level

| Action | Camera | NVR |
|--------|--------|------|
| view | Admin | User |
| set | Admin | User |

4.2. Syntax

```
http://<Device IP>/stw-cgi/recording.cgi?msubmenu=  
recordingschedule&action=<value>[&<parameter>=<value>]
```

4.3. Parameters

| Action | Parameters | Request/ Response | Type/ Value | Description |
|--------|------------|----------------------|--------------------------------|---|
| view | | | | Reads the recording schedule settings. |
| | Channel | REQ | <csv> | Channel ID |
| set | Channel | REQ, RES | <int> | Channel ID |
| | Activate | REQ, RES | <enum> Always, Scheduled | Recording type. <ul style="list-style-type: none">• Always: Always records the video• Scheduled: Records only at a specific time on a specific day of the week <div>Note Activate must be sent together with the set action.</div> <div>CAMERA ONLY</div> |

| Action | Parameters | Request/Response | Type/Value | Description |
|--------|------------|------------------|---|---|
| | <ddd> | REQ, RES | (For Cameras) <bool> 0, 1 (For NVR) <enum> 0, 1, E, B | Enables or disables recording for the selected day of week. <ul style="list-style-type: none"> • 0: Disabled • 1: Enabled • E: Events • B: Both <p><ddd> stands for week of the day and should be specified in the short form such as SUN, MON, TUE, WED, THU, FRI, and SAT in uppercase.</p> <p>e.g.) 'SUN=1' indicates that recording is activated every Sunday from 12:00 AM to 11:59 PM, unless the specific time is set using the <dddh> parameter such like SUN1=1, SUN2=1, etc.</p> <p>This parameter is valid only when Activate is set to Scheduled.</p> |
| | EveryDay | REQ, RES | (For Cameras) (For Cameras) <bool> 0, 1 (For NVR) <enum> 0, 1, E, B | Enables or disables recording for every day <ul style="list-style-type: none"> • 0: Disabled • 1: Enabled • E: Events • B: Both <p>'EveryDay=1', indicating that the recording is activated every day, has the same effect as setting the ScheduleType parameter to Always.</p> <p>EveryDay is valid only when Activate is set to Scheduled.</p> |

| Action | Parameters | Request/ Response | Type/ Value | Description |
|--------|-------------|----------------------|--|--|
| | <dddh> | REQ, RES | (For Cameras) <bool> 0, 1 (For NVR) <enum> 0, 1, E, B | Enables or disables recording for the selected hour and day. <ul style="list-style-type: none"> • 0: Disabled • 1: Enabled • E: Events • B: Both <p><dddh> stands for the week of day and time in hour. e.g. SUN1 means 1:00 AM on Sunday. MON2 means 2:00 AM on Monday.</p> <p>This parameter is available if <corresponding weekday> = 1. 'SUN=1' is required for SUN0 ... SUN23.</p> <p>e.g.) 'SUN=1&SUN18=1' indicates that recording is enabled from 6:00 PM to 6:59 PM on every Sunday</p> <p>This parameter is valid only when Activate is set to Scheduled.</p> |
| | EveryDay<h> | REQ, RES | (For Cameras) <bool> 0, 1 (For NVR) <enum> 0, 1, E, B | Enables or disables recording for every day and hour <ul style="list-style-type: none"> • 0: Disabled • 1: Enabled • E: Events • B: Both <p>This parameter is available if EveryDay =1. 'EveryDay=1' is required for EveryDay0 ... EveryDay23.</p> <p>e.g.) 'EveryDay=1&EveryDay18=1' indicates that recording is enabled from 6:00 PM to 6:59 PM every day.</p> <p>This parameter is valid only when Activate is set to Scheduled.</p> |

| Action | Parameters | Request/ Response | Type/ Value | Description |
|--------|--------------------|----------------------|----------------|--|
| | <dddh>.FromTo | REQ, RES | <string> | <p>Start time for recording for the specific time and day of week</p> <p>This parameter should be specified in the format of <mm-mm> and the first 'mm' must be smaller than or equal to the second 'mm'.</p> <p>This is available if <corresponding weekday><hour>=1. 'SUN0=1' is required for SUN0.FromTo.</p> <p>e.g.) 'SUN=1&SUN18=1&SUN18.FromTo=12-20' indicates that recording is activated from 6:12 PM to 6:20 PM on every Sunday.</p> <p>This parameter is valid only whenwhen Activate is set to Scheduled.</p> <p>CAMERA ONLY</p> |
| | EveryDay<h>.FromTo | REQ, RES | <string> | <p>Start time for recording every day</p> <p>This parameter should be specified in the format of <mm-mm> and the first 'mm' must be smaller than or equal to the second 'mm'.</p> <p>This parameter is available if EveryDay<hour>=1. 'EveryDay0=1' is required for EveryDay0.FromTo.</p> <p>e.g.) 'EveryDay=1&EveryDay18=1&EveryDay18.FromTo=12-20' indicates the recording is activated 6:12 PM to 6:20 PM every day.</p> <p>This parameter is valid only when Activate is set to Scheduled.</p> <p>CAMERA ONLY</p> |

4.4. Examples

4.4.1. Getting the current information

REQUEST

```
http://<Device IP>/stw-cgi/recording.cgi?msubmenu=  
recordingschedule&action=view
```

When Activate is set to Always

TEXT RESPONSE

```
HTTP/1.0 200 OK  
Content-type: text/plain  
<Body>
```

```
Channel=0  
Activate=Always  
SUN:0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
MON:0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
TUE:0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
WED:0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
THU:0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
FRI:0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
SAT:0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

JSON RESPONSE

```
HTTP/1.0 200 OK  
Content-type: application/json  
<Body>
```

```
{  
  "RecordSchedule": [  
    {  
      "Channel": 0,  
      "Activate": "Always",  
      "Schedule": {  
        "SUN": [  
          "0", "0", "0", "0", "0", "0", "0", "0", "0", "0",
```



```

"0",
    "0", "0", "0", "0", "0", "0", "0", "0", "0", "0",
"0"
],
"MON": [
    "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0",
"0",
    "0", "0", "0", "0", "0", "0", "0", "0", "0", "0",
"0"
],
"TUE": [
    "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0",
"0",
    "0", "0", "0", "0", "0", "0", "0", "0", "0", "0",
"0"
],
"WED": [
    "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0",
"0",
    "0", "0", "0", "0", "0", "0", "0", "0", "0", "0",
"0"
],
"THU": [
    "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0",
"0",
    "0", "0", "0", "0", "0", "0", "0", "0", "0", "0",
"0"
],
"FRI": [
    "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0",
"0",
    "0", "0", "0", "0", "0", "0", "0", "0", "0", "0",
"0"
],
"SAT": [
    "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0",
"0",
    "0", "0", "0", "0", "0", "0", "0", "0", "0", "0",
"0"
]
}

```

```
}  
]  
}
```

When Activate is set to Scheduled

TEXT RESPONSE

```
HTTP/1.0 200 OK  
Content-type: text/plain  
<Body>
```

```
Channel=0  
Activate=Scheduled  
SUN:1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
MON:1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
TUE:0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
WED:0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
THU:0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
FRI:0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
SAT:1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

JSON RESPONSE

```
HTTP/1.0 200 OK  
Content-type: application/json  
<Body>
```

```
{  
  "RecordSchedule": [  
    {  
      "Channel": 0,  
      "Activate": "Scheduled",  
      "Schedule": {  
        "SUN": [  
          "1", "1", "1", "1", "1", "1", "1", "1", "1", "1", "1", "1",  
          "1",  
          "1", "1", "1", "1", "1", "1", "1", "1", "1", "1",  
          "1"  
        ],  
      },  
    },  
  ],  
}
```

```

"MON": [
    "1", "1", "1", "1", "1", "1", "1", "1", "1", "1", "1",
"1",
    "1", "1", "1", "1", "1", "1", "1", "1", "1", "1",
"1"
],
"TUE": [
    "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0",
"0",
    "0", "0", "0", "0", "0", "0", "0", "0", "0", "0",
"0"
],
"WED": [
    "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0",
"0",
    "0", "0", "0", "0", "0", "0", "0", "0", "0", "0",
"0"
],
"THU": [
    "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0",
"0",
    "0", "0", "0", "0", "0", "0", "0", "0", "0", "0",
"0"
],
"FRI": [
    "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0",
"0",
    "0", "0", "0", "0", "0", "0", "0", "0", "0", "0",
"0"
],
"SAT": [
    "1", "1", "1", "1", "1", "1", "1", "1", "1", "1", "1",
"1",
    "1", "1", "1", "1", "1", "1", "1", "1", "1", "1",
"1"
]
}
}
]
}

```

4.4.2. Setting the device to always record

REQUEST

```
http://<Device IP>/stw-  
cgi/recording.cgi?submenu=recordingschedule&action=set&Activate=Always
```

4.4.3. Setting the recording schedule

To set the schedule (day and time), the **Activate** parameter must be set to Scheduled.

To record every Saturday and Sunday

REQUEST

```
http://<Device IP>/stw-  
cgi/recording.cgi?submenu=recordingschedule&action=set&Activate=Scheduled&S  
AT=1&SUN=1
```

To not record every Saturday and Sunday

REQUEST

```
http://<Device IP>/stw-  
cgi/recording.cgi?submenu=recordingschedule&action=set&Activate=Scheduled&S  
AT=0&SUN=0
```

To record at 1:00 AM every day and disable the recording at 4:00 AM every day

REQUEST

```
http://<Device IP>/stw-  
cgi/recording.cgi?submenu=recordingschedule&action=set&Activate=Scheduled&E  
veryDay=1&EveryDay1=1&EveryDay4=0
```

To record from 3:58 AM to 3:59 AM on Sunday

REQUEST

```
http://<Device IP>/stw-  
cgi/recording.cgi?submenu=recordingschedule&action=set&Activate=Scheduled&S  
UN=1&SUN3=1&SUN3.FromTo=58-59
```

To record every day from 3:58 AM to 3:59 AM

REQUEST

```
http://<Device IP>/stw-  
cgi/recording.cgi?submenu=recordingschedule&action=set&Activate=Scheduled&E  
veryDay=1&EveryDay3=1&EveryDay3.FromTo=58-59
```

Chapter 5. Overlapped Recording

5.1. Description

The **overlapped** submenu requests that recordings are overlapped within a given time range.

In a case where the system time settings change, or DST is applied while recording the video, the recording is overlapped for a certain period of time. A single digit number is assigned to the ID of the overlapped recording. When it returns 'OverlappedIDList=0,1' this means that there are two overlapped recordings.

NOTE | Attribute to check for Feature Support: "attributes/Recording/Support/Overlapped"

Access level

| Action | Camera | NVR |
|--------|--------|------|
| view | Admin | User |

5.2. Syntax

```
http://<Device IP>/stw-cgi/recording.cgi?msubmenu=  
overlapped&action=<value> [&<parameter>=<value>]
```

5.3. Parameters

| Action | Parameters | Request/ Response | Type/ Value | Description |
|--------|--------------------------------|----------------------|----------------|---|
| view | OverlappedIDList | RES | <csv> | Overlapped recording ID list |
| | Channel.#.OverlappedID List | RES | <csv> | Overlapped recording ID list for the given Channel MULTI DIRECTIONAL CAMERA ONLY |
| | ChannelIDList | REQ | <csv> | Channel ID list |
| | FromDate | REQ | <string> | The start date and time for when the recording occurred The date must be specified in the format of <YYYY-MM- DD hh:mm:ss>. Note FromDate and ToDate must be sent together for the view action. |

| Action | Parameters | Request/Response | Type/Value | Description |
|--------|------------|------------------|------------|---|
| | ToDate | REQ | <string> | <p>The end date and time for when the recording occurred The date must be specified in the format of <YYYY-MM-DD hh:mm:ss>.</p> <p>Note FromDate and ToDate must be sent together for the view action.</p> |

5.4. Examples

5.4.1. Getting the overlapped recording between 12:00 AM and 12:00 PM on Aug. 1, 2014

REQUEST

```
http://<Device IP>/stw-
cgi/recording.cgi?submenu=overlapped&action=view&FromDate=2014-08-01
00:00:00&ToDate=2014-08-01 23:59:59
```

TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
<Body>
```

```
OverlappedIDList=2,1,0
```

JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
  "OverlappedIDList": [
    2,
    1,
    0
  ]
}
```

]

}

Chapter 6. Manual Recording

6.1. Description

The **manualrecording** submenu controls the manual recording status (start/stop).

NOTE

This chapter applies to NVR only.

Attribute to check for Manual Recording start:

"attributes/Recording/Support/ManualRecordingStart"

Attribute to check for Manual Recording stop:

"attributes/Recording/Support/ManualRecordingStop"

Access level

| Action | NVR |
|---------|------|
| view | User |
| control | User |

6.2. Syntax

```
http://<Device IP>/stw-cgi/recording.cgi?msubmenu=  
manualrecording&action=<value>[&<parameter>=<value>]
```

6.3. Parameters

| Action | Parameters | Request/ Response | Type/ Value | Description |
|---------|------------|----------------------|-----------------------|---|
| view | | | | Reads manual recording status. |
| control | Mode | REQ | <enum> Start, Stop | Manual recording mode Note Mode must be sent together with the control action. |

6.4. Examples

6.4.1. Getting the current manual recording status

REQUEST

```
http://<Device IP>/stw-  
cgi/recording.cgi?msubmenu>manualrecording&action=view
```

TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
<Body>
```

```
Mode=Start
```

JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
  "Mode": "Start"
}
```

6.4.2. Starting the manual recording

REQUEST

```
http://<Device IP>/stw-
cgi/recording.cgi?submenu=manualrecording&action=control&Mode=Start
```

6.4.3. Stopping the manual recording

REQUEST

```
http://<Device IP>/stw-
cgi/recording.cgi?submenu=manualrecording&action=control&Mode=Stop
```

Chapter 7. Calendar Search

7.1. Description

The **calendarsearch** submenu requests the recording information from a given month.

NOTE

Attribute to check for Feature Support: "attributes/Recording/Support/SearchCalendar"
Attribute to check for Failover feature support: "/stw-cgi/attributes.cgi/attributes/recording/Support/FailOverRecording"

Access level

| Action | Camera | NVR |
|--------|--------|------|
| view | Admin | User |

7.2. Syntax

```
http://<Device IP>/stw-cgi/recording.cgi?msubmenu=  
calendarsearch&action=<value>[&<parameter>=<value>]
```

7.3. Parameters

| Action | Parameters | Request/ Response | Type/ Value | Description |
|--------|------------------------|----------------------|--|--|
| view | Month | REQ | <string> | Target month for searching Month must be specified in the <YYYY-MM> format. Note Month must be sent together with the view action. |
| | ChannelIDList | REQ | <csv> | List of channels in which recordings to be searched. |
| | Channel.#.Result | RES | <string> | Search results |
| | PrimaryDeviceIPAddress | REQ | <string> FormatInfo =IPv4Address or IPv6Addresses | IP address of primary device to which recording to be searched Applicable only if FailOver feature is supported. NVR ONLY |

| Action | Parameters | Request/Response | Type/Value | Description |
|--------|---------------------------|------------------|-----------------------|---|
| | IgnoreChannelBasedResults | REQ | <bool> True, False | If true, consolidated results will be given for all channels NVR ONLY |
| | Result | RES | <string> | Search results <div> Note This parameter provides a response only when IgnoreChannelBasedResults is set to True in the request. </div> NVR ONLY |

NOTE | represents the channel ID.

7.4. Examples

7.4.1. Searching for recordings from February 2013

The response will be a string of 31 digits, each representing a day of the month. If a digit is '0', it indicates that there are no recordings for that day. If the digit is '1', it indicates that a recording exists for that day.

The response code below means that there are recordings for the 6th, 7th and 13th of February 2013.

REQUEST

```
http://<Device IP>/stw-
cgi/recording.cgi?msubmenu=calendarsearch&action=view&Month=2013-
02&ChannelIDList=0
```

TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
<Body>
```

```
Channel.0.Result=00000110000010000000000000000000
```

JSON RESPONSE

```
HTTP/1.0 200 OK
```

Content-type: application/json
<Body>

```
{
  "CalenderSearchResults": [
    {
      "Channel": 0,
      "Result": "00000110000010000000000000000000"
    }
  ]
}
```

REQUEST

```
http://<Device IP>/stw-
cgi/recording.cgi?msubmenu=calendarsearch&action=view&Month=2013-
02&ChannelIDList=0,1,2,3& &IgnoreChannelBasedResults=true
```

TEXT RESPONSE

HTTP/1.0 200 OK
Content-type: text/plain
<Body>

Result=00000110000010000000000000000000

JSON RESPONSE

HTTP/1.0 200 OK
Content-type: application/json
<Body>

```
{
  "Result": "00000110000010000000000000000000"
}
```

Chapter 8. Timeline

8.1. Description

The **timeline** submenu requests recording information from a given time period.

NOTE

Attribute to check for Feature Support: "attributes/Recording/Support/SearchTimeline"

Access level

| Action | Camera | NVR |
|--------|--------|------|
| view | Admin | User |

8.2. Syntax

```
http://<Device IP>/stw-cgi/recording.cgi?msubmenu=  
timeline&action=<value>[&<parameter>=<value>]
```

8.3. Parameters

| Action | Parameters | Request/ Response | Type/ Value | Description |
|--------|------------|----------------------|---|--|
| view | Type | REQ | <enum> Refer to Recording Types for supported values | Recording type to search Common Types <ul style="list-style-type: none">• All: All video recordings including normal and event recordings.• Normal: Continuous video recordings.• Event: Video recording for all events. Note Type , FromDate , and ToDate must be sent together with the view action for the camera. |

| Action | Parameters | Request/ Response | Type/ Value | Description |
|--------|-------------------------------|----------------------|----------------|--|
| | FromDate | REQ | <string> | <p>The start date and time for when the recording occurred.</p> <p>The time is specified in the format of <YYYY-MM-DD hh:mm:ss>.</p> <p>Note Type, FromDate, and ToDate must be sent together with the view action.</p> |
| | ToDate | REQ | <string> | <p>The end date and time for when the recording occurred.</p> <p>The time is specified in the format of <YYYY-MM-DD hh:mm:ss>.</p> <p>Note Type, FromDate, and ToDate must be sent together with the view action.</p> |
| | OverlappedID | REQ | <int> | <p>Overlapped recording ID</p> <p>For more information about the overlapped recording, please refer to '5 Overlapped Recording' (page 22).</p> |
| | ChannelIDList | REQ | <csv> | <p>Channel ID list</p> <p>Note ChannelIDList, FromDate, and ToDate must be sent together with the view action for NVR.</p> |
| | TotalCount | RES | <int> | Total number of results |
| | Channel.#.Result.#.Start Time | RES | <string> | <p>Requested start date and time</p> <p>The time is specified in the format of <YYYY-MM-DD hh:mm:ss DST> (DST displays when supported).</p> |
| | Channel.#.Result.#.EndTime | RES | <string> | <p>Requested end date and time</p> <p>The time is specified in the format of <YYYY-MM-DD hh:mm:ss DST> (DST displays when supported).</p> |

| Action | Parameters | Request/Response | Type/Value | Description |
|--------|-------------------------|------------------|---|---|
| | Channel.#.Result.#.Type | RES | <enum> Refer to Recording Types for supported values | Recording type for the requested period |
| | Channel.#.Result.#.BkID | RES | <string> | Bookmark ID NVR ONLY |
| | PrimaryDeviceIPAddress | REQ | <string> FormatInfo =IPv4Address or IPv6Addresses | IP address of primary device in which recording data to be searched. Applicable only if FailOver feature is supported NVR ONLY |

Recording Types

| | |
|-------------------------|---|
| Type | All, Normal, Event, AlarmInput, VideoAnalysis, MotionDetection, NetworkDisconnect, FaceDetection, TamperingDetection, AudioDetection, Tracking, Manual, UserInput, DefocusDetection, FogDetection, AudioAnalysis, QueueEvent, videoloss, EmergencyTrigger, InternalHDDWarmup, GSensorEvent, ShockDetection, TemperatureChangeDetection, BoxTemperatureDetection, BodyTemperatureDetection, MaskDetection, CallRequest, TamperingSwitch, DTMFReceived, ProximitySensor |
| Channel.#.Result.#.Type | Normal, AlarmInput, VideoAnalysis, MotionDetection, NetworkDisconnect, FaceDetection, TamperingDetection, AudioDetection, Tracking, ManualRecording, UserInput, DefocusDetection, FogDetection, AudioAnalysis, ShockDetection, TemperatureChangeDetection, BoxTemperatureDetection, BodyTemperatureDetection, MaskDetection, CallRequest, TamperingSwitch, DTMFReceived, ProximitySensor |

8.4. Examples

8.4.1. Getting the timeline

Search for normal recording between 9:00 AM and 10:00 AM on Mar. 3, 2013

Type, **FromDate**, and **ToDate** must be sent together with the **view** action for the camera.

REQUEST

```
http://<Device IP>/stw-cgi/recording.cgi?submenu=timeline&action=view&Type=Normal&FromDate=2013-
```



```
03-03 09:00:00&ToDate=2013-03-03 10:00:00
```

The following request example is for NVR only. **ChannelIDList**, **FromDate**, and **ToDate** must be sent together with the **view** action for NVR

REQUEST

```
http://<Device IP>/stw-  
cgi/recording.cgi?submenu=timeline&action=view&ChannelIDList=0&FromDate=201  
3-03-03 09:00:00&ToDate=2013-03-03 10:00:00
```

TEXT RESPONSE

```
HTTP/1.0 200 OK  
Content-type: text/plain  
<Body>
```

```
TotalCount=3  
Channel.0.Result.0.StartTime=2013-03-03 09:15:52  
Channel.0.Result.0.EndTime=2013-03-03 09:18:19  
Channel.0.Result.0.Type=Normal  
Channel.0.Result.1.StartTime=2013-03-03 09:10:56  
Channel.0.Result.1.EndTime=2013-03-03 09:15:51  
Channel.0.Result.1.Type=Normal  
Channel.0.Result.2.StartTime=2013-03-03 00:10:52  
Channel.0.Result.2.EndTime=2013-03-03 00:10:56  
Channel.0.Result.2.Type=Normal
```

JSON RESPONSE

```
HTTP/1.0 200 OK  
Content-type: application/json  
<Body>
```

```
{  
  "TimeLineSearchResults": [  
    {  
      "Channel": 0,  
      "Results": [  
        {
```

```

        "Result": 0,
        "StartTime": "2013-03-03 09:15:52",
        "EndTime": "2013-03-03 09:18:19",
        "Type": "Normal"
    },
    {
        "Result": 1,
        "StartTime": "2013-03-03 09:10:56",
        "EndTime": "2013-03-03 09:15:51",
        "Type": "Normal"
    },
    {
        "Result": 2,
        "StartTime": "2013-03-03 00:10:52",
        "EndTime": "2013-03-03 00:10:56",
        "Type": "FaceDetection"
    }
]
}
]
}

```

Search for motion detection events between 9:00 AM and 10:00 AM on Mar. 3, 2013

REQUEST

```

http://<Device IP>/stw-
cgi/recording.cgi?submenu=timeline&action=view&Type=MotionDetection&FromDate=2013-03-03 09:00:00&ToDate=2013-03-03 10:00:00

```

The following request example is for NVR only.

REQUEST

```

http://<Device IP>/stw-
cgi/recording.cgi?submenu=timeline&action=view&ChannelIDList=0&FromDate=2014-02-10 00:00:01&ToDate=2014-02-10 23:59:59&Type=MotionDetection

```

TEXT RESPONSE

```

HTTP/1.0 200 OK
Content-type: text/plain

```

<Body>

```
TotalCount=2
Channel.0.Result.0.StartTime=2013-03-03 18:16:33
Channel.0.Result.0.EndTime=2013-03-03 18:16:44
Channel.0.Result.0.Type=MotionDetection
Channel.0.Result.1.StartTime=2013-03-03 18:08:36
Channel.0.Result.1.EndTime=2013-03-03 18:11:36
Channel.0.Result.1.Type=MotionDetection
```

JSON RESPONSE

HTTP/1.0 200 OK
Content-type: application/json
<Body>

```
{
  "TimeLineSearchResults": [
    {
      "Channel": 0,
      "Results": [
        {
          "Result": 0,
          "StartTime": "2013-03-03 18:16:33",
          "EndTime": "2013-03-03 18:16:44",
          "Type": "MotionDetection"
        },
        {
          "Result": 1,
          "StartTime": "2013-03-03 18:08:36",
          "EndTime": "2013-03-03 18:11:36",
          "Type": "MotionDetection"
        }
      ]
    }
  ]
}
```

NOTE

Please check if the video storage is enabled (recording.cgi > storage > Enable=True) and

physically connected, if you receive the error code (STATUS_UNKNOWN_ERROR) in the response message.

Chapter 9. Recording Period

9.1. Description

The **searchrecordingperiod** submenu requests the recording period.

NOTE

This chapter applies to NVR only.

Attribute to check for Feature Support: "attributes/Recording/Support/SearchPeriod"

Access level

| Action | NVR |
|--------|------|
| view | User |

9.2. Syntax

```
http://<Device IP>/stw-cgi/recording.cgi?msubmenu=
searchrecordingperiod&action=<value> [&<parameter>=<value>]
```

9.3. Parameters

| Action | Parameters | Request/ Response | Type/ Value | Description |
|--------|--------------|----------------------|-----------------------|--|
| view | | | | Reads recording period |
| | StartTime | RES | <string> | First recording start time |
| | EndTime | RES | <string> | Last recording end time |
| | ResultsInUTC | REQ | <bool> True, False | Enable or disable search result in UTC |

9.4. Examples

9.4.1. Searching the recording period

REQUEST

```
http://<Device IP>/stw-
cgi/recording.cgi?msubmenu=searchrecordingperiod&action=view
```

TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
```

<Body>

StartTime=2014-09-22 16:05:34

EndTime=2014-10-02 11:47:11

JSON RESPONSE

HTTP/1.0 200 OK

Content-type: application/json

<Body>

```
{
  "StartTime": "2014-09-22 16:05:34",
  "EndTime": "2014-10-02 11:47:11"
}
```

Chapter 10. Heat Map Search

10.1. Description

The **heatmapsearch** submenu requests recording information using the heat map search function and controls the heat map search settings.

NOTE

Attribute to check for Feature Support in NVR:

"attributes/Recording/Support/SearchHeatMap"

Attribute to check for Feature Support in Camera: "recording/heatmapsearch".

Access level

| Action | Camera | NVR |
|---------|--------|------|
| view | Admin | User |
| control | Admin | User |

10.2. Syntax

```
http://<Device IP>/stw-cgi/recording.cgi?msubmenu=  
heatmapsearch&action=<value> [&<parameter>=<value>]
```

10.3. Parameters

| Action | Parameters | Request/ Response | Type/ Value | Description |
|--------|------------|----------------------|--|---|
| view | Type | REQ | <enum> GridColor, Results, Status | Type If Type is set to GridColor, the view action must be sent together with ChannelIDList , FromDate , ToDate , OverlappedID , MotionType , and GridRegion ; and if Type is set to Results or Status, the view action must be sent together with SearchToken . Note GridColor Type is supported only for NVR. |

| Action | Parameters | Request/ Response | Type/ Value | Description |
|--------|---------------|----------------------|---|--|
| | ChannelIDList | REQ | <csv> | <p>List of channels in which recordings will be searched</p> <p>ChannelIDList is valid only when Type is set to Gridcolor.</p> <p>NVR ONLY</p> |
| | FromDate | REQ | <string> | <p>The start date and time for when the recording occurred.</p> <p>FromDate is valid only when Type is set to Gridcolor.</p> <p>NVR ONLY</p> |
| | ToDate | REQ | <string> | <p>The end date and time for when the recording occurred.</p> <p>ToDate is valid only when Type is set to Gridcolor.</p> <p>NVR ONLY</p> |
| | OverlappedID | REQ | <int> | <p>Overlapped recording ID</p> <p>For more information about the overlapped recording, please refer to '5 Overlapped Recording'(page 22).</p> <p>OverlappedID is valid only when Type is set to Gridcolor.</p> <p>NVR ONLY</p> |
| | MotionType | REQ | <enum> Person, Vehicle, Anything | <p>Motion type</p> <p>MotionType is valid only when Type is set to Gridcolor.</p> <p>NVR ONLY</p> |
| | GridRegion | REQ | <string> | <p>Grid area</p> <p>GridRegion is valid only when Type is set to Gridcolor.</p> <p>NVR ONLY</p> |

| Action | Parameters | Request/Response | Type/Value | Description |
|--------|--------------------------------|------------------|---|---|
| | SearchToken | REQ | <string> | Search Session token SearchToken is valid only when Type is set to Results or Status. |
| | Channel.#.ResultGridColorLevel | RES | <string> | Grid color level result Channel.#.ResultGridColorLevel is valid only when Type is set to Gridcolor. NVR ONLY |
| | TotalCount | RES | <int> | Total count TotalCount is valid only when Type is set to Results. NVR ONLY |
| | Channel.#.Result.#.StartTime | RES | <string> | Start time for a search in the corresponding results and channel Channel.#.Result.#.StartTime is valid only when Type is set to Results. DST is shown when supported. NVR ONLY |
| | Channel.#.Result.#.EndTime | RES | <string> | End time for search in the corresponding results and channel Channel.#.Result.#.EndTime is valid only when Type is set to Results. DST is shown when supported. NVR ONLY |
| | Channel.#.Result.#.Type | RES | <enum> Person, Vehicle, Anything | Search type in the corresponding results and channel Channel.#.Result.#.Type is valid only when Type is set to Results. NVR ONLY |
| | Status | RES | <enum> Completed, NotCompleted | Search status Status is valid only when Type is set to Status. |

| Action | Parameters | Request/ Response | Type/ Value | Description |
|---------|-----------------|----------------------|---|---|
| control | Mode | REQ | <enum> Start, Cancel | Mode |
| | ChannelIDList | REQ | <csv> | Channel ID list |
| | FromDate | REQ | <string> <format=YY YY-MM- DDTHH:MM :SSZ> | The start date and time for search |
| | ToDate | REQ | <string> <format=YY YY-MM- DDTHH:MM :SSZ> | The end date and time for search |
| | OverlappedID | REQ | <int> | Overlapped recording ID NVR ONLY |
| | MotionType | REQ | <enum> Person, Vehicle, Anything | Type of motion to search NVR ONLY |
| | GridRegion | REQ | <string> | Grid region for search NVR ONLY |
| | SearchToken | REQ, RES | <string> | Search token SearchToken is a request-only parameter when Mode is set to Cancel, but it will return data when Mode is set to Start. |
| | ResultImageType | REQ | <enum> WithBackgr ound, WithoutBac kground | Type of Heat Map Result Image Note This parameter is valid only when ResultAsImage is set to True. CAMERA ONLY |

| Action | Parameters | Request/Response | Type/Value | Description |
|--------|---------------|------------------|-----------------------|---|
| | ResultAsImage | REQ | <bool> True, False | HeatMap result as image <div>CAMERA ONLY</div> <div>Note Currently, the camera supports heat map results only in image format. For this reason, the ResultAsImage parameter should be set to True for camera, and this parameter should send along with the Mode parameter when Mode is set to Start.</div> |

10.4. Examples

10.4.1. Heat map search color

Type must be set to GridColor, and **ChannelIDList**, **FromDate**, **ToDate**, **OverlappedID**, **MotionType** must be sent together with **view** action.

Heat map search color is supported only for NVR. The following example is for NVR only.

REQUEST

```
http://<Device IP>/stw-  
cgi/recording.cgi?msubmenu=heatmapsearch&action=view&Type=GridColor&ChannelI  
DList=0,1&FromDate=2013-12-01 10:11:12&ToDate=2014-06-04  
08:09:10&OverlappedId=52&MotionType=Person&GridRegion=0000011111100000000001  
111110000000000000000000000000000000000000000000000000000000000000000000  
000000000000000000000000000000000000000000000000000000000000000000000000  
000000000000000000000000000000000000000000000000000000000000000000000000  
000000
```

TEXT RESPONSE

```
HTTP/1.0 200 OK  
Content-type: text/plain  
<Body>
```

```
Channel.0.ResultGridColorLevel=000000000000000000000000000000000000000000000000000000000000000000000000  
000000000000000000000000000000000000000000000000000000000000000000000000
```

[illegible]

JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

[illegible]

10.4.2. Heat map search

REQUEST

http://<Device IP>/stw-cgi/recording.cgi?submenu=heatmapsearch&action=control&ChannelIDList=0&Mode=Start&FromDate=2016-09-24T00:00:00Z&ToDate=2016-09-

24T23:59:59Z&ResultAsImage=True&ResultImageType=WithBackground

TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
<Body>
```

SearchToken=HeatMap-2016-09-24T04:32:17-824

JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
  "SearchToken": "HeatMap-2016-09-24T04:32:17-824"
}
```

The following example is for NVR only.

REQUEST

[illegible]

TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
<Body>
```

```
SearchToken=0926800821074333
```

JSON RESPONSE

```
HTTP/1.0 200 OK  
Content-type: application/json  
<Body>
```

```
{  
  "SearchToken": "0104700429062444"  
}
```

10.4.3. Getting the status of searching

Type must be set to Status, and **SearchToken** must be sent with **view** action.

REQUEST

```
http://<Device IP>/stw-  
cgi/recording.cgi?msubmenu=heatmapsearch&action=view&Type=Status&SearchToken  
=0926800821074333
```

TEXT RESPONSE

```
HTTP/1.0 200 OK  
Content-type: text/plain  
<Body>
```

```
Status=Completed
```

JSON RESPONSE

```
HTTP/1.0 200 OK  
Content-type: application/json  
<Body>
```

```
{  
  "Status": "Completed"
```

```
}
```

10.4.4. Getting the result

Type must be set to Results, and **SearchToken** must be sent with **view** action.

REQUEST

```
http://<Device IP>/stw-  
cgi/recording.cgi?msubmenu=heatmapsearch&action=view&Type=Results&SearchToken=0926800821074333
```

RESPONSE

```
HTTP/1.0 200 OK  
Content-type: text/plain  
<Body>
```

```
<PNG Image>
```

The following response example is for NVR only.

TEXT RESPONSE

```
HTTP/1.0 200 OK  
Content-type: text/plain  
<Body>
```

```
TotalCount=2  
Channel.0.Result.1.StartTime=2014-11-02T05:16:12Z  
Channel.0.Result.1.EndTime=2014-11-02T05:16:42Z  
Channel.0.Result.1.Type=Person  
Channel.0.Result.2.StartTime=2014-11-02T06:25:53Z  
Channel.0.Result.2.EndTime=2014-11-02T06:26:23Z  
Channel.0.Result.2.Type=Person
```

JSON RESPONSE

```
HTTP/1.0 200 OK  
Content-type: application/json
```

<Body>

```
{
  "TotalCount": 2,
  "HeatmapSearchResults": [
    {
      "Channel": 0,
      "Results": [
        {
          "Result": 1,
          "StartTime": "2014-11-02T05:16:12Z",
          "EndTime": "2014-11-02T05:16:42Z",
          "Type": "Person"
        },
        {
          "Result": 2,
          "StartTime": "2014-11-02T06:25:53Z",
          "EndTime": "2014-11-02T06:26:23Z",
          "Type": "Person"
        }
      ]
    }
  ]
}
```

10.4.5. Cancelling the search

REQUEST

```
http://<Device IP>/stw-
cgi/recording.cgi?msubmenu=heatmapsearch&action=control&Mode=Cancel&SearchTo
ken=0926800821074333
```


Chapter 11. People Count Search

11.1. Description

The **peoplecountsearch** submenu requests recording information using the people count search function, and controls the people count search settings.

NOTE

This chapter applies to Camera only.

Attribute to check for Feature Support: "recording/peoplecountsearch"

Access level

| Action | Camera |
|---------|--------|
| view | Admin |
| control | Admin |

11.2. Syntax

```
http://<Device IP>/stw-cgi/recording.cgi?msubmenu=  
peoplecountsearch&action=<value>[&<parameter>=<value>]
```

11.3. Parameters

| Action | Parameters | Request/ Response | Type/ Value | Description |
|--------|----------------|----------------------|---|--|
| view | Type | REQ | <enum> Results, Status | Type If Type is set to Results or Status, the view action must be sent together with SearchToken . |
| | SearchToken | REQ | <string> | Search token |
| | ResultInterval | RES | <enum> Hourly, Daily, Weekly, Monthly | Search result interval |

| Action | Parameters | Request/ Response | Type/ Value | Description |
|---------|------------------------------------|----------------------|--------------------------------------|--|
| | Camera.#.Line.#.Direction.#.Result | RES | <csv> | <p>People Count Search Results.</p> <p>If ResultInterval is Hourly, search results are in terms of hours and the number of results is fixed, i.e. 24. Here, the first result in the array represents the 0th hour of the day, and the last result in the array represents the 23rd hour of the day.</p> <p>If ResultInterval is Daily, search results are in terms of days and the first result in the array represents the first day of the month while the last result in the array represents the last day of the month.</p> <p>If ResultInterval is Weekly, search results are in terms of weeks.</p> <p>If ResultInterval is Monthly, search results are in terms of months and the first result in the array represents the first month of the year, while the last result in the array represents the last month of the year.</p> <div> Note ResultInterval is fixed by the camera based on FromDate and ToDate of the search </div> |
| | Status | RES | <enum> Completed, NotCompleted | <p>Search status</p> <p>Status is valid only when Type is set to Status.</p> |
| control | Mode | REQ | <enum> Start, Cancel | Mode |
| | Channel | REQ | <int> | Channel ID |

| Action | Parameters | Request/ Response | Type/ Value | Description |
|--------|---------------------------|----------------------|---|---|
| | FromDate | REQ | <string> <format=YY YY-MM- DDTHH:MM :SSZ> | The start date and time for search |
| | ToDate | REQ | <string> <format=YY YY-MM- DDTHH:MM :SSZ> | The end date and time for search |
| | SearchToken | REQ, RES | <string> | Search token SearchToken is a request-only parameter when Mode is set to Cancel, but it will return data when Mode is set to Start. |
| | Camera.#.Line.#.Direction | REQ | <csv> In,Out | People Count Line Direction to search |

11.4. Examples

11.4.1. People Count Search

REQUEST

```
http://<Device IP>/stw-
cgi/recording.cgi?submenu=peoplecountsearch&action=control&Channel=0&Mode=Start&FromDate=2016-09-24T00:00:00Z&ToDate=2016-09-
24T23:59:59Z&Camera.PeopleCount-
Master.Line.Gate1.Direction=In,Out&Camera.PeopleCount-
Master.Line.Gate2.Direction=In,Out
```

TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
<Body>
```

```
SearchToken=PeopleCount-2016-09-24T02:32:51-614
```

JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
  "SearchToken": " PeopleCount-2016-09-24T02:32:51-614"
}
```

11.4.2. Getting the status of searching

Type must be set to Status, and **SearchToken** must be sent with **view** action.

REQUEST

```
http://<Device IP>/stw-cgi/recording.cgi?msubmenu=peoplecountsearch
&action=view&Type=Status&SearchToken=PeopleCount-2016-09-24T02:32:51-614
```

TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
<Body>
```

```
Status=Completed
```

JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
  "Status": "Completed"
}
```

11.4.3. Getting the result

Type must be set to Results, and **SearchToken** must be sent with **view** action.

REQUEST

```
http://<Device IP>/stw-cgi/recording.cgi?msubmenu=peoplecountsearch
&action=view&Type=Results&SearchToken=PeopleCount-2016-09-24T02:32:51-614
```

TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
<Body>
```

```
ResultInterval = Hourly
Camera.PeopleCount-Master.Line.Gate1.Direction.In.Result =
0,0,0,0,0,0,2,0,0,0,0,0,0,0,6,0,0,0,0,0,3,0,2,2
Camera.PeopleCount-Master.Line.Gate1.Direction.Out.Result =
0,0,0,0,0,0,1,0,0,0,0,0,0,0,2,0,0,0,0,0,2,0,5,3
Camera.PeopleCount-Master.Line.Gate1.Direction.In.Result =
0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,11,0,0,0
Camera.PeopleCount-Master.Line.Gate2.Direction.Out.Result =
0,0,0,0,0,0,2,0,0,1,1,0,0,1,6,0,0,0,0,0,11,0,3,2
```

JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
  "ResultInterval": "Hourly",
  "PeopleCountSearchResults": [
    {
      "Camera": "PeopleCount-Master",
      "LineResults": [
        {
          "Line": "Gate1",
          "DirectionResults": [
            {
```

```

        "Direction": "In",
        "Result":
"0,0,0,0,0,0,2,0,0,0,0,0,0,0,6,0,0,0,0,0,3,0,2,2"
        },
        {
            "Direction": "Out",
            "Result":
"0,0,0,0,0,0,1,0,0,0,0,0,0,0,2,0,0,0,0,0,2,0,5,3"
        }
    ]
},
{
    "Line": "Gate2",
    "DirectionResults": [
        {
            "Direction": "In",
            "Result":
"0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,11,0,0,0"
        },
        {
            "Direction": "Out",
            "Result":
"0,0,0,0,0,0,2,0,0,1,1,0,0,1,6,0,0,0,0,0,11,0,3,2"
        }
    ]
}
]
}
]
}
}

```

11.4.4. Cancelling the search

REQUEST

```

http://<Device IP>/stw-cgi/recording.cgi?submenu=peoplecountsearch
&action=control&Mode=Cancel&SearchToken=PeopleCount-2016-09-24T02:32:51-614

```

Chapter 12. POS Configuration

12.1. Description

The **posconf** submenu is used for configuring a POS device.

NOTE

This chapter applies to NVR only.

Attribute to check for Feature Support: "attributes/System/Limit/MaxPOS"

Access level

| Action | NVR |
|--------|------|
| view | User |
| set | User |

12.2. Syntax

```
http://<Device IP>/stw-cgi/recording.cgi?msubmenu=  
posconf&action=<value>[&<parameter>=<value>]
```

12.3. Parameters

| Action | Parameters | Request/ Response | Type/ Value | Description |
|--------|---------------|----------------------|-------------------------------|--|
| view | DeviceIDList | REQ | <csv> | Optional request parameter for sending a specific device ID. Note DeviceID starts from 0. |
| set | DeviceID | REQ, RES | <int> Start, Cancel | Mode |
| | DeviceName | REQ, RES | <string> | Name of the device |
| | Port | REQ, RES | <int> | Port number to which POS is configured |
| | ChannelIDList | REQ, RES | <csv> | Channels to which POs is mapped |

| Action | Parameters | Request/ Response | Type/ Value | Description |
|--------|------------------|----------------------|---|-------------------------------|
| | EncodingType | REQ, RES | <enum> US-ASCII, UTF-8, UTF- 16, EUC-KR, ISO-2022- KR, EUC-JP, SHIFT-JIS, ISO-2022- JP, EUC-CN, ISO-2022- CN, BIG5, GB2312, ISO-8859-1, ISO-8859-2, ISO-8859-3, WINDOWS- 1250, WINDOWS- 1251, WINDOWS- 1252, WINDOWS- 1253, WINDOWS- 1254, CP850, CP866, CP932, CP949, CP950, CP1250, CP1251, CP1252, CP1253, CP1254, CP1257 | POS encoding type setting |
| | ReceiptStart | REQ, RES | <string> | Receipt start identifier |
| | ReceiptStartType | REQ, RES | <enum> Text,HexCo de,RegExpr ession | Receipt start identifier type |
| | ReceiptEnd | REQ, RES | <string> | Receipt end identifier |

| Action | Parameters | Request/ Response | Type/ Value | Description |
|--------|---------------------------------|----------------------|--|---------------------------------|
| | ReceiptEndType | REQ, RES | <enum> Text,HexCo de,RegExpr ession | Receipt end identifier type |
| | EventPlaybackStartTime | REQ, RES | <int> | Playback start time |
| | EventPlaybackStartTime Units | REQ, RES | <enum> Seconds | Time units |
| | Enable | REQ, RES | <bool> True, False | Enabling and Disabling a device |
| | DeviceType | REQ, RES | <enum> User Defined, EPSON,WIN COR,AXIHO N,RADIANT SYSTEM,IB M,ANPR | Pos Device Types |

12.4. Examples

REQUEST

```
http://<Device IP>/stw-  
cgi/recording.cgi?submenu=posconf&action=view&DeviceIDList=0,1
```

TEXT RESPONSE

```
HTTP/1.0 200 OK  
Content-type: text/plain  
<Body>
```

```
DeviceID.0.DeviceName=TEXT 01  
DeviceID.0.Enable=True  
DeviceID.0.Port=7001  
DeviceID.0.EventPlaybackStartTime=0  
DeviceID.0.EventPlaybackStartTimeUnits=Seconds  
DeviceID.0.ReceiptStart=(1)  
DeviceID.0.ReceiptEnd=(2)  
DeviceID.0.EncodingType=US-ASCII
```

```
DeviceID.0.ChannelIDList=0,1,2,3,4,5,6,7,16,17,18,19,20,21,22,23,32,33,34,35
,36,37,38,39,48,49,50,51,52,53,54,55
DeviceID.1.DeviceName=TEXT 02
DeviceID.1.Enable=True
DeviceID.1.Port=7002
DeviceID.1.EventPlaybackStartTime=0
DeviceID.1.EventPlaybackStartTimeUnits=Seconds
DeviceID.1.ReceiptStart=(1)
DeviceID.1.ReceiptEnd=(2)
DeviceID.1.EncodingType=US-ASCII
DeviceID.1.ChannelIDList=8,9,10,11,12,13,14,15,24,25,26,27,28,29,30,31,40,41
,42,43,44,45,46,47,56,57,58,59,60,61,62,63
```

JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
  "POSDevices": [
    {
      "DeviceID": 0,
      "DeviceName": "TEXT 01",
      "Enable": true,
      "Port": 7001,
      "EventPlaybackStartTime": 0,
      "EventPlaybackStartTimeUnits": "Seconds",
      "ReceiptStart": "(1)",
      "ReceiptEnd": "(2)",
      "EncodingType": "US-ASCII",
      "ChannelIDList": [
        "0",
        "1",
        "2",
        "3",
        "4",
        "5",
        "6",
        "7",
        "16",
```

```

        "17",
        "18",
        "19",
        "20",
        "21",
        "22",
        "23",
        "32",
        "33",
        "34",
        "35",
        "36",
        "37",
        "38",
        "39",
        "48",
        "49",
        "50",
        "51",
        "52",
        "53",
        "54",
        "55"
    ]
},
{
    "DeviceID": 1,
    "DeviceName": "TEXT 02",
    "Enable": true,
    "Port": 7002,
    "EventPlaybackStartTime": 0,
    "EventPlaybackStartTimeUnits": "Seconds",
    "ReceiptStart": "(1)",
    "ReceiptEnd": "(2)",
    "EncodingType": "US-ASCII",
    "ChannelIDList": [
        "8",
        "9",
        "10",
        "11",
        "12",

```

```
        "13" ,  
        "14" ,  
        "15" ,  
        "24" ,  
        "25" ,  
        "26" ,  
        "27" ,  
        "28" ,  
        "29" ,  
        "30" ,  
        "31" ,  
        "40" ,  
        "41" ,  
        "42" ,  
        "43" ,  
        "44" ,  
        "45" ,  
        "46" ,  
        "47" ,  
        "56" ,  
        "57" ,  
        "58" ,  
        "59" ,  
        "60" ,  
        "61" ,  
        "62" ,  
        "63"  
    ]  
}  
]
```

Chapter 13. POS Event Configuration

13.1. Description

The **poseventconf** submenu is used to configure a POS event in the device.

NOTE

This chapter applies to NVR only.

Attribute to check for Feature Support: "attributes/System/Limit/MaxPOS"

Access level

| Action | NVR |
|------------|------|
| view | User |
| set | User |
| add/update | User |
| remove | User |

13.2. Syntax

```
http://<Device IP>/stw-cgi/recording.cgi?msubmenu=  
poseventconf&action=<value> [&<parameter>=<value> ...]
```

13.3. Parameters

| Action | Parameters | Request/ Response | Type/ Value | Description |
|------------|-------------------|----------------------|-------------------------------------|---|
| view | | | | |
| set | AmountEventEnable | REQ, RES | <bool> True, False | Enables or disables the event based on the amount |
| | TotalType | REQ, RES | <enum> Equal, Above, Below | Total amount condition |
| | TotalAmount | REQ, RES | <float> | Total amount |
| add/update | KeywordIndex | REQ, RES | <int> | Index of keyword to be added/updated |
| | KeywordCondition | REQ, RES | <string> | Keyword string |
| remove | KeywordIndex | REQ | <int> | Valid keyword index that needs to be removed. |

13.4. Examples

REQUEST

```
http://<Device IP>/stw-cgi/recording.cgi?msubmenu=poseventconf&action=view
```

TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
<Body>
```

```
AmountEventEnable=True
TotalAmount=100.000000
TotalType=Above
KeywordIndex.1.KeywordCondition=Apple
KeywordIndex.2.KeywordCondition=banana
```

JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
  "AmountEventEnable": true,
  "TotalAmount": 100,
  "TotalType": "Above",
  "Keywords": [
    {
      "KeywordIndex": 1,
      "KeywordCondition": "Apple"
    },
    {
      "KeywordIndex": 2,
      "KeywordCondition": "banana"
    }
  ]
}
```

Chapter 14. POS Data

14.1. Description

The **posdata** submenu is used to receive the live POS data from the device.

NOTE

This chapter applies to NVR only.

Attribute to check for Feature Support: "attributes/System/Limit/MaxPOS"

Access level

| Action | NVR |
|-------------|------|
| monitordiff | User |

14.2. Syntax

```
http://<Device IP>/stw-cgi/recording.cgi?submenu=
posdata&action=<value> [&<parameter>=<value>...]
```

14.3. Parameters

| Action | Parameter | Request/ Response | Type/ Value | Description |
|-------------|--------------|----------------------|----------------|---|
| monitordiff | DeviceIDList | REQ, RES | <csv> | Device ID list for which POS data needs to be monitored |
| | ReceivedDate | RES | <string> | Date and time |
| | DeviceID | RES | <int> | Device ID Note DeviceID starts from 0. |
| | Receipt | RES | <string> | Receipt information |

14.4. Examples

REQUEST

```
http://<Device IP>/stw-cgi/recording.cgi?submenu=posdata&action=monitordiff
```

TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
```

<Body>

--SamsungTechwin

Content-type:text/plain

ReceivedDate=2016-07-28T05:06:55Z

DeviceID=1

Receipt=

03-06-16 2:43P

<keyword>APPLE</keyword> 9.00

BERRY 3.50

MELON 10.50

PLUM 3.00

SUBTOTAL 26.00

TAX 03.00

TOTAL 29.00

CASH 30.00

CHANGE 01.00

--SamsungTechwin

Content-type:text/plain

ReceivedDate=2016-07-28T05:06:55Z

DeviceID=0

Receipt=

02-06-16 2:43P

OKRA 5.00

OIL 9.50

LEMON 2.50

GREEN BANANNAS 3.00

YELLOW BANANNAS 3.00

JSON RESPONSE

HTTP/1.0 200 OK

Content-type: application/json

<Body>

--SamsungTechwin

Content-type:application/json

```
{
  "ReceivedDate": "2016-07-28T05:06:55Z",
  "DeviceID": 1,
  "Receipt": "\r\n03-06-16 2:43P\r\n<keyword>APPLE</keyword>
\t\t9.00\r\nBERRY\t\t3.50\r\nMELON\t\t10.50\r\nPLUM\t
\t3.00\r\n\r\nSUBTOTAL\t26.00\r\nTAX\t\t03.00\r\nTOTAL\t\t29.00\r\nCASH
\t\t30.00\r\nCHANGE\t\t01.00\r\n"
}
```

--SamsungTechwin

Content-type:application/json

```
{
  "ReceivedDate": "2016-07-28T05:06:55Z",
  "DeviceID": 0,
  "Receipt": "\r\n02-06-16 2:43P\r\nOKRA\t\t5.00\r\nOIL\t\t9.50\r\nLEMON
\t\t2.50\r\nGREEN BANANNAS\t3.00\r\nYELLOW
BANANNAS\t3.00\r\n\r\n\r\nSUBTOTAL\t23.00\r\nTAX\t\t02.70\r\nTOTAL
\t\t25.70\r\nCASH\t\t30.00\r\nCHANGE\t\t04.30\r\n"
}
```

Chapter 15. POS Calendar

15.1. Description

The **poscalendar** submenu is used to get the availability of POS data in a month.

NOTE

This chapter applies to NVR only.

Attribute to check for Feature Support: "attributes/System/Limit/MaxPOS"

Access level

| Action | NVR |
|--------|------|
| view | User |

15.2. Syntax

```
http://<Device IP>/stw-cgi/recording.cgi?msubmenu=  
poscalendar&action=<value>[&<parameter>=<value>...]
```

15.3. Parameters

| Action | Parameter | Request/ Response | Type/ Value | Description |
|--------|-----------|----------------------|----------------|--|
| view | Month | REQ | <string> | Target month for searching Month must be specified in <YYYY-MM> format. |
| | Calendar | RES | <string> | String of 31 characters consisting of 0s and 1s to represent each day of the month; if data is available it is set to 1 and set to 0 otherwise |

15.4. Examples

REQUEST

```
http://<Device IP>/stw-  
cgi/recording.cgi?msubmenu=poscalendar&action=view&Month=2016-09
```

TEXT RESPONSE

```
HTTP/1.0 200 OK  
Content-type: text/plain
```

<Body>

Calendar=0000011110000000001100101010101

JSON RESPONSE

HTTP/1.0 200 OK
Content-type: application/json
<Body>

```
{  
  "Calendar": "0000011110000000001100101010101"  
}
```

Chapter 16. Meta Data Search

16.1. Description

The **metadata** submenu searches POS, GPS, video, audio, and image information for a given search criteria.

NOTE

This chapter applies to NVR only.

Attribute to check for Feature Support in NVR: "attributes/Recording/Support/SearchMetadata"

Attribute to check for MaxResults supported in NVR: "attributes/Recording/metadata/view/MaxResults"

Attribute to check for Maximum Allowed time gap between search from date and search to date: "attributes/Recording/Limit/MaxMetadataSearchDays"

Access level

| Action | NVR |
|---------|------|
| view | User |
| control | User |

16.2. Syntax

```
http://<Device IP>/stw-cgi/recording.cgi?msubmenu=  
metadata&action=<value>[&<parameter>=<value>]
```

16.3. Parameters

| Action | Parameters | Request/ Response | Type/ Value | Description |
|--------|-----------------|----------------------|------------------------------|---|
| view | Type | REQ | <enum> Results, Status | Search Type |
| | SearchToken | REQ | <sring> | Search Session token |
| | ResultFromIndex | REQ | <int> | Index from which search results to be fetched |
| | ResultFromTime | REQ | <string> | Time from which search results are fetched. Time in UTC format. YYYY-MM-DDTHH:MM::SSZ |

| Action | Parameters | Request/ Response | Type/ Value | Description |
|--------------------------|-------------------|--|------------------------------------|---|
| | ResultToTime | REQ | <string> | Time to which search results are fetched. Time in UTC format. YYYY-MM-DDTHH:MM::SSZ |
| | MaxResults | REQ | <int> | Maximum number of search results to return |
| | Status | RES | <enum> Results, Status | Search status |
| | TotalResultsFound | RES | <int> | Total results |
| | TotalCount | RES | <int> | Total result count |
| | TimedOut | RES | <bool> True, False | |
| SearchTokenExpiryTime | RES | <string> UTCFormat =YYYY-MM-DDTHH:MM:SSZ | Time at which search token expires | |
| Result.#.Date | RES | <string> UTCFormat =YYYY-MM-DDTHH:MM:SSZ | Result date | |
| Result.#.PlayTime | RES | <string> UTCFormat =YYYY-MM-DDTHH:MM:SSZ | Result play time | |
| Result.#.DeviceID | RES | <int> | POS ID | |
| Result.#.TextData | RES | <string> | POS receipt | |
| Result.#.KeywordsMatched | RES | <csv> | Keywords found in POS receipt | |

| Action | Parameters | Request/ Response | Type/ Value | Description |
|------------------------|------------|---|---|-------------|
| Result.#.ChannelIDList | RES | <csv> | Result channel ID List | |
| Result.#.BKID | RES | <string> | Bookmark ID | control |
| Mode | REQ | <enum> Start, Cancel, Renew, Stop | Search Mode | |
| MetadataType | REQ | <enum> POS | Type of metadata to be searched | |
| DeviceIDList | REQ | <csv> | POS ID list | |
| Overlapped ID | REQ | <int> | Overlapped number | |
| Keyword | REQ | <string> | Search keyword | |
| IsWholeWord | REQ | <bool> True, False | To match whole word or not for search | |
| IsCaseSensitive | REQ | <bool> True, False | Whether search is case-sensitive or not | |
| FromDate | REQ | <string> UTCFormat =YYYY-MM-DDTHH:MM:SSZ | From date of the search | |

| Action | Parameters | Request/ Response | Type/ Value | Description |
|--------|------------|--|----------------------|-------------|
| ToDate | REQ | <string> UTCFormat =YYYY-MM- DDTHH:MM :SSZ | To date of search | |

16.4. Examples

16.4.1. Start search for POS data

Request without any filters

REQUEST

```
http://<Device IP>/stw-
cgi/recording.cgi?submenu=metadata&action=control&Mode=Start&MetadataType=P
OS&FromDate=2016-07-13T00:00:00Z&ToDate=2016-07-16T23:59:59Z
```

Request with Overlapped ID

REQUEST

```
http://<Device IP>/stw-
cgi/recording.cgi?submenu=metadata&action=control&Mode=Start&MetadataType=P
OS&FromDate=2016-07-15T00:00:00Z&ToDate=2016-07-16T23:59:59Z&OverlappedID=11
```

Request with Overlapped ID and Single Keyword

REQUEST

```
http://<Device IP>/stw-
cgi/recording.cgi?submenu=metadata&action=control&Mode=Start&MetadataType=P
OS&FromDate=2016-07-15T00:00:00Z&ToDate=2016-07-
16T23:59:59Z&OverlappedID=11&Keyword=Apple
```

Request with Overlapped ID and Keyword Green or Apple

REQUEST

```
http://<Device IP>/stw-
cgi/recording.cgi?submenu=metadata&action=control&Mode=Start&MetadataType=P
OS&FromDate=2016-07-15T00:00:00Z&ToDate=2016-07-
```

```
16T23:59:59Z&OverlappedID=11&IsWholeWord=false&Keyword=Green%20Apple
```

Request with Overlapped ID and Keyword "Green,Apple"sss

REQUEST

```
http://<Device IP>/stw-  
cgi/recording.cgi?submenu=metadata&action=control&Mode=Start&MetadataType=P  
OS&FromDate=2016-07-15T00:00:00Z&ToDate=2016-07-  
16T23:59:59Z&OverlappedID=11&Keyword=Green,Apple
```

Request with Overlapped ID, Keyword and IsCaseSensitive

REQUEST

```
http://<Device IP>/stw-  
cgi/recording.cgi?submenu=metadata&action=control&Mode=Start&MetadataType=P  
OS&FromDate=2016-07-15T00:00:00Z&ToDate=2016-07-  
16T23:59:59Z&OverlappedID=11&Keyword=APPLE&IsCaseSensitive=true
```

Request with Overlapped ID, Keyword, IsCaseSensitive and Single DeviceID

REQUEST

```
http://<Device IP>/stw-  
cgi/recording.cgi?submenu=metadata&action=control&Mode=Start&MetadataType=P  
OS&FromDate=2016-07-15T00:00:00Z&ToDate=2016-07-  
16T23:59:59Z&OverlappedID=11&Keyword=OKRA&IsCaseSensitive=true&DeviceIDList=  
0
```

Request with Overlapped ID, Keyword, IsCaseSensitive and Multiple DeviceIDs

REQUEST

```
http://<Device IP>/stw-  
cgi/recording.cgi?submenu=metadata&action=control&Mode=Start&MetadataType=P  
OS&FromDate=2016-07-15T00:00:00Z&ToDate=2016-07-  
16T23:59:59Z&OverlappedID=11&Keyword=OKRA&IsCaseSensitive=true&DeviceIDList=  
1,2
```

TEXT RESPONSE

```
HTTP/1.0 200 OK
```



```
Content-type: text/plain
<Body>
```

```
SearchToken=7475
```

JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
  "SearchToken": "7475"
}
```

16.4.2. Cancel Search

REQUEST

```
http://<Device IP>/stw-
cgi/recording.cgi?submenu=metadata&action=control&Mode=Cancel&SearchToken=7
475
```

TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
<Body>
```

```
OK
```

JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
```

```
"Response": "Success"
}
```

16.4.3. To get search status

REQUEST

```
http://<Device-IP>/stw-
cgi/recording.cgi?submenu=metadata&action=view&Type=Status&SearchToken=7475
```

TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
<Body>
```

```
Status=Completed
```

JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
  "Status": "Completed"
}
```

16.4.4. To renew search token (For 1 minute)

REQUEST

```
http://<Device-IP>/stw-
cgi/recording.cgi?submenu=metadata&action=control&Mode=Renew&SearchToken=74
75
```

TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
```

<Body>

OK

JSON RESPONSE

HTTP/1.0 200 OK
Content-type: application/json
<Body>

```
{  
  "Response": "Success"  
}
```

16.4.5. To get the results of search (First 100 results)

REQUEST

http://<Device-IP>/stw-
cgi/recording.cgi?submenu=metadata&action=view&Type=Results&ResultFromIndex
=1&MaxResults=100&SearchToken=6619

TEXT RESPONSE

HTTP/1.0 200 OK
Content-type: text/plain
<Body>

SearchTokenExpiryTime=2016-07-19T07:22:47Z
TotalResultsFound=399
TotalCount=100

Result.1.DeviceID=1
Result.1.Date=2016-07-18T07:28:01Z
Result.1.ChannelIDList=0,1,2,3,4,5,6,7
Result.1.KeywordsMatched=
Result.1.TextData=
02-06-16 2:43P
OKRA 5.00

OIL 9.50
LEMON 2.50
GREEN BANANNAS 3.00
YELLOW BANANNAS 3.00

SUBTOTAL 23.00
TAX 02.70
TOTAL 25.70
CASH 30.00
CHANGE 04.30

Result.2.DeviceID=2
Result.2.Date=2016-07-18T07:28:00Z
Result.2.ChannelIDList=8,9,10,11,12,13,14,15
Result.2.KeywordsMatched=
Result.2.TextData=
03-06-16 2:43P
APPLE 9.00
BERRY 3.50
MELON 10.50
PLUM 3.00

SUBTOTAL 26.00
TAX 03.00
TOTAL 29.00
CASH 30.00
CHANGE 01.00

Result.3.DeviceID=1
Result.3.Date=2016-07-18T07:27:56Z
Result.3.ChannelIDList=0,1,2,3,4,5,6,7
Result.3.KeywordsMatched=
Result.3.TextData=
02-06-16 2:43P
OKRA 5.00
OIL 9.50
LEMON 2.50
GREEN BANANNAS 3.00
YELLOW BANANNAS 3.00

JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
  "SearchTokenExpiryTime": "2016-07-19T07:22:47Z",
  "ToralResultsFound": 399,
  "TotalCount": 100,
  "MetaDataSearchResults": [
    {
      "Result": 1,
      "DeviceID": 2,
      "Date": "2016-07-18T07:22:47Z",
      "ChannelIDList": [
        8,
        9,
        10,
        11,
        12,
        13,
        14,
        15
      ],
      "KeywordsMatched": [],
      "TextData": "\r\n03-06-16 2:43P\r\nAPPLE \t\t9.00\r\nBERRY\t\t3.50\r\nMELON \t\t10.50\r\nPLUM\t\t3.00\r\n\r\nSUBTOTAL \t26.00\r\nTAX \t\t03.00\r\nTOTAL \t\t29.00\r\nCASH \t\t30.00\r\nCHANGE \t\t01.00\r\n"
    },
    {
      "Result": 2,
      "DeviceID": 1,
      "Date": "2016-07-18T07:22:47Z",
      "ChannelIDList": [
        0,
        1,
        2,
        3,
        4,
        5,

```

```

        6,
        7
    ],
    "KeywordsMatched": [],
    "TextData": "\r\n02-06-16 2:43P\r\nOKRA \t\t5.00\r\nOIL\t
\t9.50\r\nLEMON \t\t2.50\r\nGREEN BANANNAS\t3.00\r\nYELLOW
BANANNAS\t3.00\r\n\r\n\r\nSUBTOTAL \t23.00\r\nTAX \t\t02.70\r\nTOTAL
\t\t25.70\r\nCASH \t\t30.00\r\nCHANGE \t\t04.30\r\n"
    },
    {
        "Result": 3,
        "DeviceID": 2,
        "Date": "2016-07-18T07:22:43Z",
        "ChannelIDList": [
            8,
            9,
            10,
            11,
            12,
            13,
            14,
            15
        ],
        "KeywordsMatched": [],
        "TextData": "\r\n03-06-16 2:43P\r\nAPPLE \t\t9.00\r\nBERRY\t
\t3.50\r\nMELON \t\t10.50\r\nPLUM\t \t3.00\r\n\r\nSUBTOTAL \t26.00\r\nTAX
\t\t03.00\r\nTOTAL \t\t29.00\r\nCASH \t\t30.00\r\nCHANGE \t\t01.00\r\n"
    }
]
}

```

16.4.6. To get the results of search (Next 100 results)

REQUEST

```

http://<Device-IP>/stw-
cgi/recording.cgi?msubmenu=metadata&action=view&Type=Results&ResultFromIndex
=101&MaxResults=100&SearchToken=6619

```

Chapter 17. Smart Search

17.1. Description

The **smartsearch** submenu is used to search video analytics information for a given search criteria.

NOTE

This chapter applies to NVR only.

Attribute to check for Feature Support in NVR for each channel:

"attributes.cgi/attributes/recording/Support/1/SmartSearch"

Attribute to check for maximum number of include areas supported in NVR:

"attributes/recording/Limit/MaxSmartSearchIncludeAreas"

Attribute to check for maximum number of exclude areas supported in NVR:

"attributes/recording/Limit/MaxSmartSearchExcludeAreas"

Attribute to check for maximum number of lines supported in NVR:

"attributes/recording/Limit/MaxSmartSearchlines"

Access level

| Action | NVR |
|---------|------|
| view | User |
| control | User |

17.2. Syntax

```
http://<Device IP>/stw-cgi/recording.cgi?msubmenu=  
smartsearch&action=<value> [&<parameter>=<value>]
```

17.3. Parameters

| Action | Parameters | Request/ Response | Type/ Value | Description |
|--------|----------------------------------|----------------------|------------------------------|------------------------------|
| view | Type | REQ | <enum> Results, Status | Search Type |
| | SearchToken | REQ | <string> | Search session token |
| | TotalCount | RES | <int> | Result count |
| | TimedOut | RES | <bool> True, False | Asynchronous search timeout. |
| | Channel.#.Result.#.Event Time | RES | <string> | Time at which event happened |

| Action | Parameters | Request/ Response | Type/ Value | Description |
|-------------------|----------------------------------|--|---|--|
| | Channel.#.Result.#.Event Type | RES | <enum> Motion, Enter, Exit, Pass | Type of event |
| | Status | RES | <enum> Completed, NotComple ted | Search status This parameter is returned only when Type is set to Status in the request. |
| control | Mode | REQ | <enum> Start, Cancel, Renew, Stop | Search mode |
| | OverlappedID | REQ | <int> | Overlapped ID number |
| | Channel | REQ | <int> | Channel ID |
| | FromDate | REQ | <string> | From date of the search |
| | ToDate | REQ | <string> | To date of search |
| | Area.#.EventType | REQ | <csv> Motion, Enter, Exit | Type of area event |
| | Line.#.EventType | REQ | <enum> BothDirecti ons, Right, Left | Type of line event |
| | Area.#.Type | REQ | <enum> Inside, Outside | Area type |
| | Area.#.Coordinates | REQ | <string> Format=x1, y1,x2,y2 | |
| Area.#.Filt er | REQ | <enum>+ Person, Vehicle, Unknown Object filter | Coordinate s of the area | |

| Action | Parameters | Request/ Response | Type/ Value | Description |
|--------------------|------------|--|-------------------------|-------------|
| Line.#.Coordinates | REQ | <string> Format=x1, y1,x2,y2 | Coordinates of the line | |

17.4. Examples

17.4.1. Start smart search for selected area

Request without any filters

REQUEST

```
http://<Device IP>/stw-
cgi/recording.cgi?submenu=smartsearch&action=control&Mode=Start&channel=5&F
romDate=2016-06-15T00:00:00Z&ToDate=2016-06-
15T23:59:59Z&Area.1.EventType=Motion,Enter,Exit&Area.1.Type=Inside&Area.1.Co
ordinates=-0.903226,0.870504,-0.903226,-0.877698,0.903226,-
0.877698,0.903226,0.870504
```

TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
<Body>
```

```
SearchToken=4174
```

JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
  "SearchToken": "4174"
}
```

17.4.2. Check the status of smart search

REQUEST

```
http://<Device IP>/stw-  
cgi/recording.cgi?msubmenu=smartsearch&action=view&Type=Status&SearchToken=4  
174
```

TEXT RESPONSE

```
HTTP/1.0 200 OK  
Content-type: text/plain  
<Body>
```

```
Status=Completed
```

JSON RESPONSE

```
HTTP/1.0 200 OK  
Content-type: application/json  
<Body>
```

```
{  
  "Status": "Completed"  
}
```

17.4.3. To get results of smart search

REQUEST

```
http://<Device IP>/stw-  
cgi/recording.cgi?msubmenu=smartsearch&action=view&Type=Results&SearchToken=  
4174
```

TEXT RESPONSE

```
HTTP/1.0 200 OK  
Content-type: text/plain  
<Body>
```

```
TotalCount=5
Channel.5.Result.1.EventTime=2016-06-15T00:08:35Z
Channel.5.Result.1.EventType=Motion
Channel.5.Result.2.EventTime=2016-06-15T00:32:34Z
Channel.5.Result.2.EventType=Motion
Channel.5.Result.3.EventTime=2016-06-15T00:32:35Z
Channel.5.Result.3.EventType=Motion
Channel.5.Result.4.EventTime=2016-06-15T01:32:02Z
Channel.5.Result.4.EventType=Motion
Channel.5.Result.5.EventTime=2016-06-15T01:32:03Z
Channel.5.Result.5.EventType=Motion
```

JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
  "TotalCount": 5,
  "SmartSearchResults": [
    {
      "Channel": 5,
      "Results": [
        {
          "Result": 1,
          "EventTime": "2016-06-15T00:08:35Z",
          "EventType": "Motion"
        },
        {
          "Result": 2,
          "EventTime": "2016-06-15T00:32:34Z",
          "EventType": "Motion"
        },
        {
          "Result": 3,
          "EventTime": "2016-06-15T00:32:35Z",
          "EventType": "Motion"
        },
        {

```

```

        "Result": 4,
        "EventTime": "2016-06-15T01:32:02Z",
        "EventType": "Motion"
    },
    {
        "Result": 5,
        "EventTime": "2016-06-15T01:32:03Z",
        "EventType": "Motion"
    }
]
}

```

17.4.4. Start Smart search for the lines

REQUEST

```

http://<Device IP>/stw-
cgi/recording.cgi?submenu=smartsearch&action=control&Mode=Start&channel=5&F
romDate=2016-06-15T00:00:00Z&ToDate=2016-06-
15T23:59:59Z&Line.1.EventType=Right&Line.1.Coordinates=-
0.903226,0.870504,0.903226,0.870504

```

Chapter 18. Queue Search

18.1. Description

The **queuesearch** submenu provides statistical analysis and measurement of average dwell time and number of people in queues based on a given search criteria.

NOTE

This chapter applies to camera only.

Attribute to check for Feature Support: "Recording/Support/QueueManagement"

Attribute to check for Max Queues Supported: "Eventsource/Limit/MaxQueues"

Access level

| Action | Camera |
|---------|--------|
| view | Admin |
| control | Admin |

18.2. Syntax

```
http://<Device IP>/stw-cgi/recording.cgi?msubmenu=
queuesearch&action=<value>[&<parameter>=<value>]
```

18.3. Parameters

| Action | Parameters | Request/ Response | Type/ Value | Description |
|--------|----------------|----------------------|---|--|
| view | Type | REQ | <enum> Results, Status | Type If Type is set to Results or Status, the view action must be sent together with SearchToken . |
| | SearchToken | REQ | <string> | Search token |
| | ResultInterval | RES | <enum> Hourly, Daily, Weekly, Monthly | Search result interval |

| Action | Parameters | Request/ Response | Type/ Value | Description |
|--------|---------------------------------|----------------------|----------------|--|
| | Queue.#.AveragePeople Result | RES | <csv> | <p>Queue Search Average People Results.</p> <p>If ResultInterval is Hourly, search results are returned in terms of hours and the number of results is fixed, i.e. 24. Here, the first result in the array represents the 0th hour of the day, and the last result in the array represents the 23rd hour of the day.</p> <p>If ResultInterval is Daily, search results are returned in terms of days and the first result in the array represents the first day of the month while the last result in the array represents the last day of the month.</p> <p>If ResultInterval is Weekly, search results are returned in terms of weeks.</p> <p>If ResultInterval is Monthly, search results are returned in terms of months and the first result in the array represents the first month of the year, while the last result in the array represents the last month of the year.</p> <div> Note ResultInterval is fixed by the camera based on FromDate and ToDate of the search </div> |

| Action | Parameters | Request/ Response | Type/ Value | Description |
|---------|--------------------------------------|----------------------|--------------------------------------|---|
| | Queue.#.Level.#.CumulativeTimeResult | RES | <csv> | <p>Queue Search Cumulative Time Result</p> <p>If ResultInterval is Hourly, search results are returned in terms of hours and the number of results is fixed, i.e. 24. Here, the first result in the array represents the 0th hour of the day, and the last result in the array represents the 23rd hour of the day.</p> <p>If ResultInterval is Daily, search results are returned in terms of days and the first result in the array represents the first day of the month while the last result in the array represents the last day of the month.</p> <p>If ResultInterval is Weekly, search results are returned in terms of weeks.</p> <p>If ResultInterval is Monthly, search results are returned in terms of months and the first result in the array represents the first month of the year, while the last result in the array represents the last month of the year.</p> <div> Note ResultInterval is fixed by the camera based on FromDate and ToDate of the search </div> |
| | Status | RES | <enum> Completed, NotCompleted | <p>Search status</p> <p>Status is valid only when Type is set to Status.</p> |
| control | Mode | REQ | <enum> Start, Cancel | Mode |
| | Channel | REQ | <int> | Channel ID |

| Action | Parameters | Request/ Response | Type/ Value | Description |
|--------|--------------------------------|----------------------|---|---|
| | FromDate | REQ | <string> <format=YY YY-MM- DDTHH:MM :SSZ> | The start date and time for search |
| | ToDate | REQ | <string> <format=YY YY-MM- DDTHH:MM :SSZ> | The end date and time for search |
| | SearchToken | REQ, RES | <string> | Search token SearchToken is a request-only parameter when Mode is set to Cancel, but it will return data when Mode is set to Start. |
| | Queue.#.AveragePeople | REQ | <bool> True, False | Enables or disables Queue Average People search |
| | Queue.#.Level.#.CumulativeTime | REQ | <bool> True, False | Enables or disables Queue Cumulative Time search |

18.4. Examples

18.4.1. Queue Search

Queue search for average people result

REQUEST

```
http://<Device IP>/stw-
cgi/recording.cgi?submenu=queuesearch&action=control&Channel=0&Mode=Start&F
romDate=2017-01-17T00:00:00Z&ToDate=2017-01-
17T23:59:59Z&Queue.1.AveragePeople=True&Queue.2.AveragePeople=True&Queue.3.A
veragePeople=True
```

JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```



```
{
  "SearchToken": "QueueManagement-2017-09-15T04:3"
}
```

Queue search for cumulative time result

REQUEST

```
http://<Device IP>/stw-
cgi/recording.cgi?submenu=queuesearch&action=control&Channel=0&Mode=Start&F
romDate=2017-01-17T00:00:00Z&ToDate=2017-01-
17T23:59:59Z&Queue.1.Level.High.CumulativeTime=True&Queue.1.Level.Medium.Cum
ulativeTime=True&Queue.2.Level.High.CumulativeTime=True&Queue.2.Level.Medium
.CumulativeTime=True&Queue.3.Level.High.CumulativeTime=True&Queue.3.Level.Me
dium.CumulativeTime=True
```

JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
  "SearchToken": "QueueManagement-2017-09-15T04:4"
}
```

Queue search for average people and cumulative time result

REQUEST

```
http://<Device IP>/stw-
cgi/recording.cgi?submenu=queuesearch&action=control&Channel=0&Mode=Start&F
romDate=2017-01-17T00:00:00Z&ToDate=2017-01-
17T23:59:59Z&Queue.1.AveragePeople=True&Queue.2.AveragePeople=True&Queue.3.A
veragePeople=True&Queue.1.Level.High.CumulativeTime=True&Queue.1.Level.Medium
.CumulativeTime=True&Queue.2.Level.High.CumulativeTime=True&Queue.2.Level.M
edium.CumulativeTime=True&Queue.3.Level.High.CumulativeTime=True&Queue.3.Lev
el.Medium.CumulativeTime=True
```

JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
  "SearchToken": "QueueManagement-2017-09-15T04:5"
}
```

18.4.2. Getting the status of Queue search

Type must be set to Status, and **SearchToken** must be sent with **view** action.

REQUEST

```
http://<Device IP>/stw-
cgi/recording.cgi?submenu=queuesearch&action=view&Type=Status&SearchToken=Q
ueueManagement-2017-09-15T04:4
```

JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
  "Status": "Completed"
}
```

18.4.3. Getting Queue search result

Type must be set to Results, and **SearchToken** must be sent with **view** action.

Getting Average People search result

REQUEST

```
http://<Device IP>/ stw-
cgi/recording.cgi?submenu=queuesearch&action=view&Type=Results&SearchToken=
QueueManagement-2017-09-15T04:3
```

JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
  "ResultInterval": "Hourly",
  "QueueResults": [
    {
      "Queue": 1,
      "AveragePeopleResult": [
        "0",
        "1",
        "2",
        "3",
        "4",
        "5",
        "6",
        "7",
        "8",
        "9",
        "10",
        "11",
        "12",
        "13",
        "14",
        "15",
        "16",
        "17",
        "18",
        "19",
        "20",
        "21",
        "22",
        "23"
      ]
    },
    {
      "Queue": 2,
      "AveragePeopleResult": [
```

```

        "0",
        "1",
        "2",
        "3",
        "4",
        "5",
        "6",
        "7",
        "8",
        "9",
        "10",
        "11",
        "12",
        "13",
        "14",
        "15",
        "16",
        "17",
        "18",
        "19",
        "20",
        "21",
        "22",
        "23"
    ]
},
{
    "Queue": 3,
    "AveragePeopleResult": [
        "0",
        "1",
        "2",
        "3",
        "4",
        "5",
        "6",
        "7",
        "8",
        "9",
        "10",
        "11",

```

```

        "12",
        "13",
        "14",
        "15",
        "16",
        "17",
        "18",
        "19",
        "20",
        "21",
        "22",
        "23"
    ]
}
]
}

```

Getting Cumulative Time search result

REQUEST

```

http://<Device IP>/ stw-
cgi/recording.cgi?submenu=queuesearch&action=view&Type=Results&SearchToken=
QueueManagement-2017-09-15T04:4

```

JSON RESPONSE

```

HTTP/1.0 200 OK
Content-type: application/json
<Body>

```

```

{
  "ResultInterval": "Hourly",
  "QueueResults": [
    {
      "Queue": 1,
      "QueueLevels": [
        {
          "Level": "High",
          "CumulativeTimeResult": [
            "0", "1", "2", "3", "4", "5", "6", "7", "8", "9",

```

```

"10", "11", "12",
                                "13", "14", "15", "16", "17", "18", "19", "20",
"21", "22", "23"
    ]
    },
    {
        "Level": "Medium",
        "CumulativeTimeResult": [
            "0", "1", "2", "3", "4", "5", "6", "7", "8", "9",
"10", "11", "12",
                                "13", "14", "15", "16", "17", "18", "19", "20",
"21", "22", "23"
        ]
    }
]
},
{
    "Queue": 2,
    "QueueLevels": [
        {
            "Level": "High",
            "CumulativeTimeResult": [
                "0", "1", "2", "3", "4", "5", "6", "7", "8", "9",
"10", "11", "12",
                                "13", "14", "15", "16", "17", "18", "19", "20",
"21", "22", "23"
            ]
        },
        {
            "Level": "Medium",
            "CumulativeTimeResult": [
                "0", "1", "2", "3", "4", "5", "6", "7", "8", "9",
"10", "11", "12",
                                "13", "14", "15", "16", "17", "18", "19", "20",
"21", "22", "23"
            ]
        }
    ]
},
{
    "Queue": 3,

```

```

    "QueueLevels": [
      {
        "Level": "High",
        "CumulativeTimeResult": [
          "0", "1", "2", "3", "4", "5", "6", "7", "8", "9",
"10", "11", "12",
          "13", "14", "15", "16", "17", "18", "19", "20",
"21", "22", "23"
        ]
      },
      {
        "Level": "Medium",
        "CumulativeTimeResult": [
          "0", "1", "2", "3", "4", "5", "6", "7", "8", "9",
"10", "11", "12",
          "13", "14", "15", "16", "17", "18", "19", "20",
"21", "22", "23"
        ]
      }
    ]
  }
}

```

Getting Average People and Cumulative Time search result

REQUEST

```

http://<Device IP>/ stw-
cgi/recording.cgi?submenu=queuesearch&action=view&Type=Results&SearchToken=
QueueManagement-2017-09-15T04:5

```

JSON RESPONSE

```

HTTP/1.0 200 OK
Content-type: application/json
<Body>

```

```

{
  "ResultInterval": "Hourly",
  "QueueResults": [

```

```

{
  "Queue": 1,
  "AveragePeopleResult": [
    "0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "10",
"11", "12",
    "13", "14", "15", "16", "17", "18", "19", "20", "21", "22",
"23"
  ],
  "QueueLevels": [
    {
      "Level": "High",
      "CumulativeTimeResult": [
        "0", "1", "2", "3", "4", "5", "6", "7", "8", "9",
"10", "11", "12",
        "13", "14", "15", "16", "17", "18", "19", "20",
"21", "22", "23"
      ]
    },
    {
      "Level": "Medium",
      "CumulativeTimeResult": [
        "0", "1", "2", "3", "4", "5", "6", "7", "8", "9",
"10", "11", "12",
        "13", "14", "15", "16", "17", "18", "19", "20",
"21", "22", "23"
      ]
    }
  ],
  {
    "Queue": 2,
    "AveragePeopleResult": [
      "0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "10",
"11", "12",
      "13", "14", "15", "16", "17", "18", "19", "20", "21", "22",
"23"
    ],
    "QueueLevels": [
      {
        "Level": "High",
        "CumulativeTimeResult": [

```



```

        "0", "1", "2", "3", "4", "5", "6", "7", "8", "9",
"10", "11", "12",
        "13", "14", "15", "16", "17", "18", "19", "20",
"21", "22", "23"
    ]
},
{
    "Level": "Medium",
    "CumulativeTimeResult": [
        "0", "1", "2", "3", "4", "5", "6", "7", "8", "9",
"10", "11", "12",
        "13", "14", "15", "16", "17", "18", "19", "20",
"21", "22", "23"
    ]
}
]
},
{
    "Queue": 3,
    "AveragePeopleResult": [
        "0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "10",
"11", "12",
        "13", "14", "15", "16", "17", "18", "19", "20", "21", "22",
"23"
    ],
    "QueueLevels": [
        {
            "Level": "High",
            "CumulativeTimeResult": [
                "0", "1", "2", "3", "4", "5", "6", "7", "8", "9",
"10", "11", "12",
                "13", "14", "15", "16", "17", "18", "19", "20",
"21", "22", "23"
            ]
        },
        {
            "Level": "Medium",
            "CumulativeTimeResult": [
                "0", "1", "2", "3", "4", "5", "6", "7", "8", "9",
"10", "11", "12",
                "13", "14", "15", "16", "17", "18", "19", "20",

```

```
"21", "22", "23"  
    ]  
  }  
] }  
}
```

18.4.4. Cancelling Queue search

REQUEST

```
http://<Device IP>/stw-cgi/recording.cgi?msubmenu=queuesearch  
&action=control&Mode=Cancel&SearchToken=QueueManagement-2017-09-15T04:3
```

Chapter 19. Disk Utility

19.1. Description

The **diskutility** submenu gets the details of HDD array from NVR

NOTE

This submenu is only available for NVR

Access level

| Action | NVR |
|--------|------|
| view | User |

19.2. Syntax

```
http://<Device IP>/stw-cgi/recording.cgi?msubmenu=  
diskutility&action=<value> [&<parameter>=<value>]
```

19.3. Parameters

| Action | Parameters | Request/ Response | Type/ Value | Description |
|--------|--------------|----------------------|----------------|-------------------|
| view | Index | REQ | <int> | Disk index |
| | Disk.#.Index | RES | <int> | Disk index |
| | Disk.#.Name | RES | <string> | Disk model name |
| | Disk.#.SMART | RES | <string> | Smart HDD details |

19.4. Examples

19.4.1. Getting the disk information

REQUEST

```
http://<Device IP>/stw-cgi/recording.cgi?msubmenu=diskutility&action=view
```

TEXT RESPONSE

```
HTTP/1.0 200 OK  
Content-type: text/plain  
<Body>
```

```
Disk.0.Index=14
Disk.0.Name=ST1000VM002-1CT162
Disk.1.Index=15
Disk.1.Name=WDC WD60PURX-64T0ZY1
Disk.2.Index=16
Disk.2.Name=WDC WD40PURX-64N96Y0
```

JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
  "Disks": [
    {
      "Index": 14,
      "Name": "ST1000VM002-1CT162 "
    },
    {
      "Index": 15,
      "Name": "WDC WD60PURX-64T0ZY1 "
    },
    {
      "Index": 16,
      "Name": "WDC WD40PURX-64N96Y0 "
    }
  ]
}
```

REQUEST

```
http://<Device IP>/stw-
cgi/recording.cgi?msubmenu=diskutility&action=view&Index=14
```

TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
```

<Body>

```
Enable=True
Disk.0.Index=14
Disk.0.Name=ST1000VM002-1CT162
Disk.0.SMART=<html><body><h3 style="color:rgb(60,179,133);"> Status :
GOOD</h3>
<pre>Model Name : ST1000VM002-1CT162
Serial : W1G0AJYW
Firmware Version : SC23
Capacity : 1 TB
Temperature : 35&#8451; / 95&#8457;

</pre><pre>ID Attribute Name Current Worst Threshold RawValue Status
001 Read Error Rate 117 099 006 0000000057888 GOOD
003 Spin-Up Time 098 097 000 0000000000000 GOOD
004 Start/Stop Count 096 096 020 0000000004413 GOOD
005 Reallocated Sectors 100 100 036 0000000000000 GOOD
007 Seek Error Rate 080 060 030 0000000024466 GOOD
009 Power-On Hours Count 076 076 000 0000000021334 GOOD
010 Spin Retry Count 100 100 097 0000000000000 GOOD
012 Power Cycle Count 099 099 020 0000000001131 GOOD
184 End-to-End error 100 100 099 0000000000000 GOOD
187 Reported Uncorrectable 088 088 000 0000000000012 GOOD
188 Command Timeout 100 099 000 0000000000002 GOOD
189 High Fly Writes 001 001 000 0000000000363 GOOD
190 Temperature Diff 065 051 045 0000000000035 GOOD
191 G-sense error rate 100 100 000 0000000000000 GOOD
192 Power-off retract 100 100 000 0000000001125 GOOD
193 Load/Unload cycle 098 098 000 0000000004413 GOOD
194 HDA temperature 035 049 000 0000000000035 GOOD
197 Current pending 100 100 000 0000000000000 GOOD
198 Offline scan wrong 100 100 000 0000000000000 GOOD
199 UDMA CRC error rate 200 200 000 0000000000000 GOOD
</pre></body></html>
```

JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
```

<Body>

```
{
  "Disks": [
    {
      "Index": 14,
      "Name": "ST1000VM002-1CT162 ",
      "SMART": "<html><body><h3 style=\"color:rgb(60,179,133);\">
Status : GOOD</h3>\n<pre>Model Name : ST1000VM002-1CT162\nSerial :
W1G0AJYW\nFirmware Version : SC23\nCapacity : 1 TB\nTemperature : 35&#8451;
/ 95&#8457; \n\n</pre><pre>ID Attribute Name Current Worst Threshold
RawValue Status\n001 Read Error Rate 117 099 006 0000000057888 GOOD \n003
Spin-Up Time 098 097 000 0000000000000 GOOD \n004 Start/Stop Count 096 096
020 0000000004413 GOOD \n005 Reallocated Sectors 100 100 036 0000000000000
GOOD \n007 Seek Error Rate 080 060 030 0000000024466 GOOD \n009 Power-On
Hours Count 076 076 000 0000000021334 GOOD \n010 Spin Retry Count 100 100 097
0000000000000 GOOD \n012 Power Cycle Count 099 099 020 0000000001131 GOOD
\n184 End-to-End error 100 100 099 0000000000000 GOOD \n187 Reported
Uncorrectable 088 088 000 0000000000012 GOOD \n188 Command Timeout 100 099
000 0000000000002 GOOD \n189 High Fly Writes 001 001 000 0000000000363 GOOD
\n190 Temperature Diff 065 051 045 0000000000035 GOOD \n191 G-sense error
rate 100 100 000 0000000000000 GOOD \n192 Power-off retract 100 100 000
0000000001125 GOOD \n193 Load/Unload cycle 098 098 000 0000000004413 GOOD
\n194 HDA temperature 035 049 000 0000000000035 GOOD \n197 Current pending
100 100 000 0000000000000 GOOD \n198 Offline scan wrong 100 100 000
0000000000000 GOOD \n199 UDMA CRC error rate 200 200 000 0000000000000 GOOD
\n</pre></body></html>"
    }
  ]
}
```

Chapter 20. Bookmark

20.1. Description

The **bookmark** submenu can be used to bookmark a video clip in NVR.

NOTE

This submenu is available only for NVR that supports System/Support/AIFeatures

Access level

| Action | NVR |
|--------|------|
| view | User |
| add | User |
| update | User |
| remove | User |

20.2. Syntax

```
http://<Device IP>/stw-cgi/recording.cgi?submenu=  
bookmark&action=<value>[&<parameter>=<value>]
```

20.3. Parameters

| Action | Parameters | Request/ Response | Type/ Value | Description |
|--------|-------------------|----------------------|----------------|--|
| view | ChannelIDList | REQ | <csv> | Channel id list |
| | FromDate | REQ | <string> | Time in UTC format YYYY-MM-DDTHH:MM:SSZ |
| | ToDate | REQ | <string> | Time in UTC format YYYY-MM-DDTHH:MM:SSZ |
| | OverlappedID | REQ | <int> | Overlapped id |
| | TotalCount | RES | <int> | Total result count |
| | Result.#.FromDate | RES | <string> | Time in UTC format YYYY-MM-DDTHH:MM:SSZ |
| | Result.#.ToDate | RES | <string> | Time in UTC format YYYY-MM-DDTHH:MM:SSZ |

| Action | Parameters | Request/ Response | Type/ Value | Description |
|--------|-----------------------|----------------------|--|--|
| | Result#.ChannelIDList | RES | <csv> | Channel id list |
| | Result#.Name | RES | <string> | Name of bookmark |
| | Result#.Category | RES | <enum> TIME, EVENT, SMART, TEXT, AI_PERSON, AI_FACE, AI_FACE_RECOGNITION, AI_VEHICLE | Bookmark category |
| | Result#.SubCategory | RES | <enum> AlarmInput, MotionDetection, VideoLoss, Passing, Entering, Exiting, Appearing, Disappearing, Tampering, FaceDetection, Loitering, Tracking, DefocusDetection, FogDetection, AudioDetection, Scream, Gunshot, Explosion, GlassBreak, GSensorEvent, EmergencyTrigger, Intrusion | Subcategory of bookmark |
| | Result#.OverlappedID | RES | <int> | Recording overlapped id |
| | Result#.ObjectID | RES | <int> | Object ID |
| | Result#.BkID | RES | <string> | Unique bookmark ID (UUID) |
| add | FromDate | REQ | <string> | Time in UTC format YYYY-MM-DDTHH:MM:SSZ |
| | ToDate | REQ | <string> | Time in UTC format YYYY-MM-DDTHH:MM:SSZ |
| | ChannelIDList | REQ | <csv> | Channel id list |
| | ObjectID | REQ | <int> | Object ID |
| | Name | REQ | <string> | Bookmark Name |

| Action | Parameters | Request/Response | Type/Value | Description |
|--------|--------------|------------------|---|----------------------------------|
| | Category | REQ | <enum> TIME, EVENT, SMART, TEXT, AI_PERSON, AI_FACE, AI_FACE_RECOGNITION, AI_VEHICLE | Bookmark category |
| | OverlappedID | REQ | <int> | Recording overlapped id |
| update | BkID | REQ | <string> | Bookmark unique id (UUID) |
| | Name | REQ | <string> | Bookmark name |
| remove | BkID | REQ | <string> | Unique bookmark id to be removed |

20.4. Examples

20.4.1. Adding a bookmark

REQUEST

```
http://<Device IP>/stw-
cgi/recording.cgi?submenu=bookmark&action=add&FromDate=2020-04-
12T09:00:00Z&ToDate=2020-04-
13T09:03:00Z&ChannelIDList=0&Category=TIME&Name=TestBookMark&OverlappedID=-
1&objectID=214
```

TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
<Body>
```

```
BkID=3757c60d27b8484cab29468d0c800a23
```

JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
  "BkID": "3757c60d27b8484cab29468d0c800a23"
}
```

20.4.2. Removing a bookmark

REQUEST

```
http://<Device IP>/stw-
cgi/recording.cgi?submenu=bookmark&action=remove&BkID=3757c60d27b8484cab294
68d0c800a23
```

TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
<Body>
```

```
OK
```

JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
  "Response": "Success"
}
```

20.4.3. Updating a bookmark

REQUEST

```
http://<Device
IP>/stwcgi/recording.cgi?submenu=bookmark&action=update&BkID=3757c60d27b848
4cab29468d0c800a23&Name=TestBookMark2
```

TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
<Body>
```

```
OK
```

JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
  "Response": "Success"
}
```

20.4.4. Viewing a bookmark

REQUEST

```
http://<Device IP>/stw-
cgi/recording.cgi?submenu=bookmark&action=view&Category=TEXT
```

TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
<Body>
```

```
TotalResultsFound=1
TotalCount=1

Result.0.FromDate=2020-03-31T09:00:00Z
Result.0.ToDate=2020-03-31T09:03:00Z
Result.0.ChannelIDList=0
Result.0.Name=TestBookMark
Result.0.Category=TIME
```

```
Result.0.SubCategory=  
Result.0.OverlappedID=100  
Result.0.ObjectID=0  
Result.0.BkID=3757c60d27b8484cab29468d0c800a23
```

JSON RESPONSE

```
HTTP/1.0 200 OK  
Content-type: application/json  
<Body>
```

```
{  
  "TotalResultsFound": 1,  
  "TotalCount": 1,  
  "BookmarkResults": [  
    {  
      "Result": 0,  
      "FromDate": "2020-03-31T09:00:00Z",  
      "ToDate": "2020-03-31T09:03:00Z",  
      "ChannelIDList": [  
        "0"  
      ],  
      "Name": "TestBookMark",  
      "Category": "TIME",  
      "SubCategory": "",  
      "OverlappedID": 100,  
      "ObjectID": 0,  
      "BkID": "3757c60d27b8484cab29468d0c800a23"  
    }  
  ]  
}
```

Chapter 21. Event Search

21.1. Description

The **eventsearch** submenu used to search event information for a given time period.

NOTE

This submenu is supported only in NVR.

Attribute to check for Feature Support: "attributes/Recording/Support/SearchEvent"

Access level

| Action | NVR |
|---------|------|
| view | User |
| control | User |

21.2. Syntax

```
http://<Device IP>/stw-cgi/recording.cgi?submenu=  
eventsearch&action=<value>[&<parameter>=<value>]
```

21.3. Parameters

| Action | Parameters | Request/ Response | Type/ Value | Description |
|--------|-----------------|----------------------|------------------------------|--|
| view | Type | REQ | <enum> Results, Status | If Type is passed as Status , search status is informed. If Type is passed as Result , search result is provided. |
| | SearchToken | REQ | <string> | Search session token |
| | ResultFromIndex | REQ | <int> | Index from which search results are fetched |
| | ResultFromTime | REQ | <string> | Time from which search results are fetched. Time in UTC format. YYYY-MM-DDTHH:MM::SSZ |

| Action | Parameters | Request/ Response | Type/ Value | Description |
|--------|------------------------|----------------------|--|---|
| | ResultToTime | REQ | <string> | Time to which search results are fetched. Time in UTC format. YYYY-MM-DDTHH:MM::SSZ |
| | MaxResults | REQ | <int> | Maximum number of search results to return |
| | Status | RES | <enum> Completed, NotComple ted | Search status |
| | TotalResultsFound | RES | <int> | Total results |
| | TotalCount | RES | <int> | Total count of result |
| | TimedOut | RES | <bool> True, False | Search timeout. |
| | IntervalFrom | RES | <string> | Start time of search result. Time in UTC format. YYYY-MM-DDTHH:MM::SSZ |
| | IntervalTo | RES | <string> | End time of search result. Time in UTC format. YYYY-MM-DDTHH:MM::SSZ |
| | SearchTokenExpiryTime | RES | <string> | Time when the search token expires Time in UTC format YYYY-MM-DDTHH:MM:SSZ |
| | Result.#.StartDateTime | RES | <string> | Time in UTC format YYYY-MM-DDTHH:MM:SSZ |
| | Result.#.EndDateTime | RES | <string> | Time in UTC format YYYY-MM-DDTHH:MM:SSZ |
| | Result.#.Channel | RES | <int> | Result channel ID |
| | Result.#.OverlapId | RES | <int> | Recording overlapped id |

| Action | Parameters | Request/Response | Type/Value | Description |
|---------|--------------------|------------------|---|--|
| | Result.#.EventType | RES | <enum> Refer to Recording Types for supported values | EventType |
| | Result.#.BkID | RES | <string> | Bookmark ID |
| control | Mode | REQ | <enum> Start, Cancel, Renew, Stop | Used to start, cancel, renew or stop search |
| | FromDate | REQ | <string> | Time in UTC format YYYY-MM-DDTHH:MM:SSZ |
| | ToDate | REQ | <string> | Time in UTC format YYYY-MM-DDTHH:MM:SSZ |
| | ChannelIDList | REQ | <csv> | On which channel search has to be performed |
| | OverlappedID | REQ | <int> | Recording overlapped id |
| | Type | REQ | <enum> Refer to Recording Types for supported values | Recording type to search Common Types <ul style="list-style-type: none"> • All: All video recordings including normal and event recordings. • Normal: Continuous video recordings. • Event: Video recording for all events. |
| | WaitTime | REQ | <int> | Timeout second.(Default:60 sec.) |

Recording Types

| | |
|------|---|
| Type | All, Normal, Event, AlarmInput, VideoAnalysis, MotionDetection, NetworkDisconnect, FaceDetection, TamperingDetection, AudioDetection, Tracking, Manual, UserInput, DefocusDetection, FogDetection, AudioAnalysis, QueueEvent, videoloss, EmergencyTrigger, InternalHDDWarmup, GSensorEvent, ShockDetection, TemperatureChangeDetection, BoxTemperatureDetection, BodyTemperatureDetection, MaskDetection, CallRequest, TamperingSwitch, DTMFReceived, ProximitySensor |
|------|---|

| | |
|-----------------------------|--|
| Channel.#.Result.#. Type | Normal, AlarmInput, VideoAnalysis, MotionDetection, NetworkDisconnect, FaceDetection, TamperingDetection, AudioDetection, Tracking, ManualRecording, UserInput, DefocusDetection, FogDetection, AudioAnalysis, ShockDetection, TemperatureChangeDetection, BoxTemperatureDetection, BodyTemperatureDetection, MaskDetection, CallRequest, TamperingSwitch, DTMFReceived, ProximitySensor |
|-----------------------------|--|

21.4. Examples

21.4.1. Event search

REQUEST

```
http://<Device IP>/stw-
cgi/recording.cgi?submenu=eventSearch&action=control&Mode=Start&OverlappedI
D=100&FromDate=2023-04-04T00:00:00Z&ToDate=2023-04-
04T01:00:00Z&ChannelIDList=0
```

TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
<Body>
```

```
SearchToken=22775
```

JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
  "SearchToken": "22775"
}
```

21.4.2. Viewing search result status

REQUEST

```
http://<Device IP>/stw-
cgi/recording.cgi?submenu=eventsearch&action=view&Type=Status&SearchToken=2
```


2775

TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
<Body>
```

```
SearchTokenExpiryTime=2023-04-04T01:36:35Z
Status=Completed
TotalResultsFound=0
TotalCount=4
TimedOut=False
IntervalFrom=2023-04-04T00:01:54Z
IntervalTo=2023-04-04T00:25:15Z
```

JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
  "SearchTokenExpiryTime":    "2023-04-04T01:36:35Z",
  "Status":    "Completed",
  "TotalResultsFound":    0,
  "TotalCount":    4,
  "TimedOut":    "False",
  "IntervalFrom":    "2023-04-04T00:01:54Z",
  "IntervalTo":    "2023-04-04T00:25:15Z",
  "Results":    []
}
```

21.4.3. Viewing search result

REQUEST

```
http://<Device IP>/stw-
cgi/recording.cgi?submenu=eventsearch&action=view&Type=Results&SearchToken=
```

```
22775&ResultFromIndex=1&MaxResults=100
```

TEXT RESPONSE

```
HTTP/1.0 200 OK
Content-type: text/plain
<Body>
```

```
SearchTokenExpiryTime=2023-04-04T01:38:05Z
Status=Completed
TotalResultsFound=4
TotalCount=4
TimedOut=False
Result.0.StartDateTime=2023-04-04T00:24:50Z
Result.0.EndDateTime=2023-04-04T00:25:15Z
Result.0.Channel=0
Result.0.OverlapId=100
Result.0.EventType=Normal
Result.0.BkID=00000000000000000000000000000000
...
Result.3.StartDateTime=2023-04-04T00:01:54Z
Result.3.EndDateTime=2023-04-04T00:23:38Z
Result.3.Channel=0
Result.3.OverlapId=100
Result.3.EventType=Normal
Result.3.BkID=00000000000000000000000000000000
```

JSON RESPONSE

```
HTTP/1.0 200 OK
Content-type: application/json
<Body>
```

```
{
  "SearchTokenExpiryTime":    "2023-04-04T01:38:05Z",
  "Status":    "Completed",
  "TotalResultsFound":    4,
  "TotalCount":    4,
  "TimedOut": "False",
  "Results":  [{
```

```

    "Result": 0,
    "StartDateTime": "2023-04-04T00:24:50Z",
    "EndDateTime": "2023-04-04T00:25:15Z",
    "Channel": 0,
    "OverlapId": 100,
    "EventType": "Normal",
    "BkID": "00000000000000000000000000000000"
  },
  ...
  {
    "Result": 3,
    "StartDateTime": "2023-04-04T00:01:54Z",
    "EndDateTime": "2023-04-04T00:23:38Z",
    "Channel": 0,
    "OverlapId": 100,
    "EventType": "Normal",
    "BkID": "00000000000000000000000000000000"
  }
]
}

```