ACS API

AXIS Camera Station 5 API

Please note that

- the aim is to maintain stability for the AXIS Camera Station API over long periods, but
- we do regular improvements, replacements, and expansions and
- on occasion we are forced to do breaking changes.

May 8, 2024
Rev. 2.30

Contents

1. Versions

The versions of this API were released with the following AXIS Camera Station versions:

API Version	AXIS Camera Station Version	Changes
1.3	5.16	
1.4	5.19	
1.5	5.20	• Added GetCameraCapabilities
1.6	5.21	• Added support for Action Buttons and On-Screen Controls.
1.7	5.23	• Added RecordingSnapshotFacade with endpoint GetSnapshots • Added hardware and vendor information to GetSystem • Added GetRecordedMedia endpoint to RecordingFacade • Added EventFacade with endpoint GetEventList.
1.8	5.25	• Added audio transmission API and capabilities for it.
2.0	5.26	• Changed ID-numbers to be format “3615_0566a4da-7ff4-4a72-bc4c-5fe609a46030”. • Methods in RecordingFacade now returns seconds with 7 decimals. • GetCameraList now includes user camera privileges for AudioInAccess and AudioOutAccess. • Removed undocumented methods LiveViewFacade.GetLiveViewStream and PlaybackFacade.GetPlaybackStream.
2.1	5.27	• Added GetOnScreenControls and TriggerOnScreenControl.
2.2	5.28	• Added H.265 support for live and playback streaming.
2.3	5.29	• Internal release.
2.4	5.30	• Added Client Commands feature.
2.5	5.31	• Added mechanical PTZ priority feature. • Added MPEG-4. • Changed validation of IDs in all calls. Incorrect IDs will now throw an ApiException.
2.6	5.32	• Added DeviceListFacade endpoint for retrieving the Devices.
2.7	5.33	• Internal release.
2.8	5.34	• Deprecated GetDeviceList endpoint. • Included TriggerFacade documentation on activation and deactivation of triggers.
2.9	5.35	• Added section with supported http methods for endpoints. • Improved input data validation and error messaging for API calls.
2.10	5.35.271	• Changed the format of the parameter section in OnScreenControlsFacade.
2.11	5.36	• GetServerConfiguration now returns only existing media profiles. Field type corrected to String for VideoViewToken in documentation. • ActionButtonFacade checks if user has access to cameras and throws UnauthorizedException otherwise. • ActionButtonFacade.SetActionButtons validates buttons state and throws CommunicationException otherwise. • The returned Name in GetPtzPresets is now UTF8 encoded instead of HTML encoded. • Removed deprecated endpoint PtzFacade/GetPresets.

		• Removed deprecated endpoint PtzFacade/GotoPreset. • Removed deprecated option to start playback using Sequence ID.
2.12	5.37	• Added TimeZone parameter to GetSystem response. • Prerecorded videos will now include a streaming profile, this will result in them correctly returning MediaProfiles when calling GetCameraList.
2.13	5.38	• Internal release.
2.14	5.39	• Top level input parameters can now start with an upper-case letter. • Introduced Deep Links for launching the AXIS Camera Station client with certain parameters set.
2.16	5.41	• Added lowestavailable as a quality level. • Introduced new RecordingControlFacade for setting and getting recording status for cameras. • Introduced new Export streaming endpoint for exporting recording to a seek-able file. • Introduced new DeviceId and VideoViewToken fields for GetCameraList response.
2.17	5.42	• Introduced new BookmarkFacade for managing bookmarks. • Added new optional parameters for controlling quality when requesting recording snapshots. • Extended deep link support for recording tab, export tab, and configuration pages. • Added AddCamera endpoint to CameraFacade.
2.18	5.43	• Internal release.
2.19	5.44	• Time can now be omitted in AddBookmark in BookmarkFacade. If omitted, the bookmark time will be set to the current server time.
2.20	5.45	• Internal release.
2.21	5.46	• Introduced camera ID field to relevant event log types in GetEventLogList response.
2.22	5.47	• Internal release.
2.23	5.49	• Internal release.
2.24	5.50	• Internal release.
2.25	5.51	• Internal release. • Corrected BookmarkFacade documentation.
2.26	5.52	• Solved an issue in AddCamera that prevented adding of cameras if the provided address contained capital characters.
2.27	5.53	• Internal release
2.28	5.54 (and 5.55)	• Internal release
2.29	5.56	• Introduced new RecordedEventFacade for retrieving recorded events and event types.
2.30	5.57	• GetDeviceList endpoint deprecation revoked. • Added new optional parameter for preferred resolution when requesting recording snapshots. • GetCameraList response includes additional camera privileges for recording export and snapshot. • Introduced the ability to include dynamic trigger data in HTTP notification action, to send event information from an action rule to a web server.

Table: ACS API versions and AXIS Camera Station versions.

2. URI

To access the ACS API the URI is constructed in the following way:

`https://<Server-address>:<port>/Acs/Api/<Facade>/<Operation>`

- <Server-address> - The IP address or hostname to the AXIS Camera Station server.
- <Port> - The port for the AXIS Camera Station API on the server (default 55756).
- <Facade> - The facade name, e.g., “CameraListFacade”.
- <Operation> - The name of the operation to perform, e.g., “GetCameraList”

An example URI could thus be:

```
https://localhost:55756/Acs/Api/CameraListFacade/GetCameraList
```

With exception for Playback and Streaming where the URIs are on the form of

```
https://<Server-address>:55756/Acs/Streaming/Video/<StreamingType>/<Format>/?
<UriParameter1>=<UriParameter1Value>&<UriParameter2>=<UriParameterValue2>
```

- <StreamingType> - The type of streaming, live or playback.
- <Format> - The video format, Matroska or MP4
- <UriParameter> - The URI parameters of the request
- <UriParameterValue> - The value of that URI parameter.

3. Default Ports

API: 55756 (HTTPS)

The TCP port for HTTPS on the ACS server (default 55756). This port is also known as the mobile API port. What port to use can be modified in the AXIS Camera Station Service Control.

4. Authentication

Supported authentication methods are HTTPS basic, Kerberos or NTLM. Username and password is required when authenticating.
The server certificate is self-signed.

Future authentication support

Support for authentication against the AXIS Camera Station server using client certificates is planned but not yet implemented. A client certificate would mean that the client uses Windows account credentials for first login on the AXIS Camera Station server. The client would then submit its public certificate key to the server, which adds the key to a list of trusted client certificates. Subsequent requests would use mutual certificate authentication with HTTPS, without the need to involve Windows account credentials.

5. HTTP POST/GET

HTTP **POST** should be used by clients when making requests and sending parameters JSON-formatted in the request body. An example body for a HTTP **POST** request for `https://localhost:55756/Acs/Api/CameraListFacade/GetCameraList` could look like:

```
{
   "Range": {
      “StartIndex”: 0,
      "NumberOfElements": 5
   }
}
```

Where not stated otherwise it is also possible to use HTTP **GET** and sending the JSON parameters without newlines as parameters in the URI when e.g., using a web browser. An example HTTP **GET** request could look like:

```
https://localhost:55756/Acs/Api/CameraListFacade/GetCameraList?{"Range":{"Sta
rtIndex":0,"NumberOfElements":5}}
```

6. HTTP Status Codes

The following HTTP status codes are possible responses from the ACS API:

- **200 OK** – The operation was performed successfully.
- **400 Bad Request** – There was some error with the request. The “Bad Request” reason phrase is replaced with the error type. These errors are described in detail in Section 8 Error Handling.
- **401 Unauthorized** – The user has provided invalid credentials and can therefore not access the API.
- **404 Not Found** – The requested operation could not be found.
- **500 Internal Server Error** – An internal server error has occurred.

7. Format

7.1. Encoding

All requests and responses are encoded using UTF-8.

7.2. Parameter letter case

Input parameter names can start with an upper- or lower-case letter. They are parsed case insensitively. Output parameters are in Pascal case: each word starts with an upper-case letter.

7.3. ID format

Input data validation and error messaging for API calls is improved from version 2.9. The IDs consists of a string containing numbers, letters, underscore, and dashes. For backwards compatibility some of the older formats containing numbers and # are still accepted on some

APIs. *Note:* The old format could potentially impose duplicates or non-unique IDs if used in a multi-server environment or in future versions.

8. Error Handling

For HTTP status code 400 the ACS API response body is on the form:

```
{
    "Type" : "ExceptionType"
    "Message" : "Exception message"
}
```

For each operation in the ACS API a set of generic errors can occur. These generic errors are described here. Besides these generic errors, there may be more specific errors for each operation. Such specific errors are documented in section Facades and operations below for each operation.

ApiException

The most generic exception thrown by the API when an operation has failed.

Response example:

HTTP Response: 400 ApiException

```
{
    "Type" : "ApiException",
    "Message" : "Some error"
}
```

UnauthorizedException

Error due to that the user is authenticated, but unauthorized to perform the operation. This exception is due to authorization problems with the server, i.e., an operation that fails due to authorization problems between server and camera will not generate this exception.

Response example:

HTTP Response: 400 UnauthorizedException

```
{
    "Type" : "UnauthorizedException",
    "Message" : "Some error"
}
```

CommunicationException

Error for communication-related issues like JSON format error or wrong parameters. Also, if e.g., a camera ID is in the wrong format, or if it is not corresponding to the current server or a camera installed on the server. This exception is due to communication problems with the server, i.e., an operation that fails due to communication problems between server and camera will not generate this exception.

Response example: HTTP Response: 400 CommunicationException

```
{
    "Type" : "CommunicationException",
    "Message" : "Some error"
}
```

ConcurrencyException

Error due to some concurrency problem, for example when trying to access a camera that does not exist anymore (most likely since some other client has removed it).

Response example: HTTP Response: 400 ConcurrencyException

```
{
    "Type" : "ConcurrencyException",
    "Message" : "Some error"
}
```

9. Recordings

9.1. Triggers

In AXIS Camera Station a recording is started by a trigger. There may be several simultaneous recordings started by different triggers. A trigger can be one of the following:

- **Motion detection** – The video is only recorded when motion has been detected.
- **Manual** – A user has manually started a recording from a client. The API endpoint StartRecording is also considered a manually started recording.
- **Continuous or scheduled** – The video from the camera is recorded continuously or according to a schedule.
- **Failover** – If the connection between the server and the camera has been lost, video can be saved to the camera SD card. When the connection is restored, the video is imported to the server.
- **Rule** – A custom rule that has been set up by a user to start recording when a certain trigger has occurred.

9.2. Sequences and Recordings

When recording a video in AXIS Camera Station a *sequence* is created. The sequence describes a time interval during which recording was made for a specific trigger. During a sequence there may be interrupts in the recording due to e.g., network errors. Therefore, a sequence can consist of several parts, called *recordings*. The figure below shows the relation between a sequence and recordings:

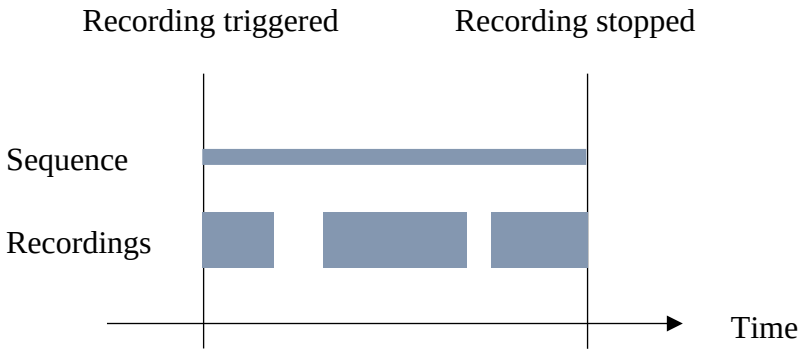

Figure: The relation between a sequence and recordings.

10. Streaming

Introduction

The AXIS Camera Station server can deliver media streams in an MP4 or Matroska container through HTTPS. For authentication, Basic HTTP authentication, Kerberos and NTLM is supported. To request live and playback streams, the ACS API can be used to retrieve available cameras and recordings.

For live streams fragmented MP4 is used. Each fragment contains a single frame to minimize latency. For playback, each fragment contains one or more complete GOPs. MP4 streaming is available on AXIS Camera Station servers with API version 1.3 or higher.

For Matroska, segments of unknown size are used (see Matroska Live Streaming). Matroska streaming is available with API version 1.4 or higher.

Export of recording

Since API version 2.16, there are two streaming endpoints which can be used to export recorded video.

The playback endpoint supports both the MP4 and Matroska container format but has the downside of not being seek-able in media players.

The export endpoint adds support for downloading a file that is seek-able. This endpoint cannot stream recording that can be viewed progressively but must be fully downloaded first before playing back.

Limitations

- The MP4 container format does not support the G.711 and G.726 audio encodings and will filter out the audio if those encodings are encountered.
- The export endpoint only supports the MP4 container format.
- Currently, the server certificate for HTTPS communication is self-signed. This means that web browsers will complain about unsafe source if an MP4 or Matroska stream is requested from the server to the browser.

Reference Documents

- ISO/IEC 14496-12:2015 - Information technology — Coding of audio-visual objects — Part 12: ISO base media file format
- ISO/IEC 14496-14:2003 - Information technology — Coding of audio-visual objects — Part 14: MP4 file container
- ISO/IEC 14496-15:2017 - Information technology — Coding of audio-visual objects — Part 15: Carriage of network abstraction layer (NAL) unit structured video in the ISO base media file format
- https://www.matroska.org/technical/specs/index.html – Matroska Documentation

10.1. Live streaming

A stream is requested using the following URI syntax:

URI: `/Acs/Streaming/Video/Live/<Container>/?camera=<cameraId>&quality=<qualityLevel>&audio=<isRequestingAudio>&prerolling=<isPrerolling>`

Request:

Field	Description
<Container>	The video container. Could be either MP4 (API version 1.3) or Matroska (API version 1.4).
<cameraId>	The ID of the camera in AXIS Camera Station to receive the live stream from.
<qualityLevel>	The requested quality level of the stream. Available values are the ones supported by the camera, normally **high**, **medium,** and **low**.
<isRequestingAudio>	Boolean – Optional. (Default value 1). Value 1 means that the stream contains audio if audio is enabled for the camera. Value 0 means that no audio is delivered in the stream. For MP4, the value must be 0.
<isPrerolling>	Boolean – Optional. (Default value 0). Value 1 means that the first frames of the stream may be old frames from a cache. This may lower the delay until from sending the streaming request to receiving the first frame in the response. Value 0 means that no frames will be pre-rolled, that is, all frames delivered in the response stream are fetched from the source camera after the server receives the streaming request.

Example:
`https://172.25.127.28:55756/Acs/Streaming/Video/Live/Mp4/?camera=284_54dd4102-96c7-4cdb-a8d2-85c2187a5d47&quality=medium&audio=0`

If no camera or quality is available according to the requested parameters, the HTTP response will be 404 Not Found.

This operation is available from API version 1.3 with MP4 and 1.4 with Matroska.

10.2. Playback

A stream is requested using the following URI syntax:

URI: `/Acs/Streaming/Video/Playback/<Container>/?camera=<cameraId>&quality=<qualityLevel>&start=<startTime>&end=<endTime>&audio=<isRequestingAudio>`

Request:

Field	Description
<Container>	The video container. Could be either MP4 (API version 1.3) or Matroska (API version 1.4).
<cameraId>	The ID of the camera to receive the live stream from.

<qualityLevel>	The requested quality level of the stream. Available values are the ones supported by the camera, normally **high**, **medium, low, highestavailable, lowestavailable**
<startTime>	The requested start time of the stream in UTC on the format "YYYY-MM-DD-hhmmss-fffffffZ". The actual time of the first frame in the stream may be lower depending on GOP structure in the recording.
<endTime>	Optional. The requested end time of the stream in UTC on the format "YYYY-MM-DD-hhmmss-fffffffZ".
<isRequestingAudio>	Boolean – Optional. (Default value 1). Value 1 means that the stream contains audio if audio is enabled for the camera. Value 0 means that no audio is delivered in the stream. For MP4, the value must be 0.

Example:
`https://172.25.127.28:55756/Acs/Streaming/Video/Playback/Mp4/?camera=284_54dd4102-96c7-4cdb-a8d2-85c2187a5d47&quality=medium&start=2017-11-29-175333-9529723Z&end=2017-11-29-175455-9934500Z&audio=0`

If no camera or quality is available according to the requested parameters, the HTTP response will be 404 Not Found.

This operation is available from API version 1.3 with MP4 and 1.4 with Matroska.

10.3. Export

Provides a MP4 file that is seek-able in media players. A stream is requested using the following URI syntax:

URI: /`Acs/Streaming/Video/Export/<Container>/?camera=<cameraId>&quality=<qualityLevel>&start=<startTime>&end=<endTime>&audio=<isRequestingAudio>`

Request:

Field	Description
<Container>	The video container. Must be Mp4.
<cameraId>	The ID of the camera to receive the live stream from.
<qualityLevel>	The requested quality level of the stream. Available values are the ones supported by the camera, normally **high**, **medium, low, highestavailable**. **lowestavailable**
<startTime>	The requested start time of the stream in UTC on the format "YYYY-MM-DD-hhmmss-fffffffZ". The actual time of the first frame in the stream may be lower depending on GOP structure in the recording.
<endTime>	Optional. The requested end time of the stream in UTC on the format "YYYY-MM-DD-hhmmss-fffffffZ".
<isRequestingAudio>	Boolean – Optional. (Default value 1). Value 1 means that the stream contains audio if audio is enabled for the camera. Value 0 means that no audio is delivered in the stream. For MP4, the value must be 0.

Example:
```
https://172.25.127.28:55756/Acs/Streaming/Video/Export/Mp4/?camera=284_54dd410
2-96c7-4cdb-a8d2-85c2187a5d47&quality=medium&start=2017-11-29-175333-
9529723Z&end=2017-11-29-175455-9934500Z&audio=0
```

If no camera or quality is available according to the requested parameters, the HTTP response will be 404 Not Found.

This operation is available from API version 2.16.

10.4. Audio transmission

An audio transmission is sent using the following URI syntax:

URI: `/Acs/Streaming/TransmitAudio/<Container>/?camera=<cameraId>`

Request:

Field	Description
<Container>	The audio container. Only Raw is supported at this time, which is a continuous stream of bytes being sent by the client.
<cameraId>	The AXIS Camera Station ID of the camera to transmit the audio to.

Example:
```
https://172.25.127.28:55756/Acs/Streaming/TransmitAudio/Raw/?camera=2542_54dd4
102-96c7-4cdb-a8d2-85c2187a5d47
```

If no camera with the specified ID is available or the camera has no speaker attached to it, the HTTP response will be 404 Not Found.

The encoding of the audio must be Axis µ-law 128 kbps. (G.711 µ-law PCM 16 kHz 128 kbps)

This operation is available from API version 1.8.

11. Facades and operations

Facades (same as an interface) contains operations for a specific area, e.g., snapshot, or live view handling. The following facades and operations are available in the ACS API. For each operation there is an example URI, request, and response.

11.1. VersionFacade

Handles version information about the API.

GetApiVersion

Gets the version of the API. The major version is increased when backward compatibility cannot be assumed. The minor version is increased at changes which are backward compatible.

URI: `/Acs/Api/VersionFacade/GetApiVersion`

Request: No parameters are used in the request body.

Example request: `https://172.25.127.28:55756/Acs/Api/VersionFacade/GetApiVersion`

Response:

Field	Description
Major	Integer - The major version of the API
Minor	Integer - The minor version of the API

Table: GetApiVersion response

Example response:

HTTP Response: 200 OK

```
{
    "Major": 1,
    "Minor": 7
}
```

Available from API version 1.3 and later.

11.2. SystemFacade

Handles information about the system.

GetSystem

Gets basic system information.

URI: `/Acs/Api/SystemFacade/GetSystem`

Request: No parameters are used in the request body.

Example request: `https://172.25.127.28:55756/Acs/Api/SystemFacade/GetSystem`

Response:

Field	Description
Id	String – The global unique identifier of the server.
Name	String – The name of the server. This should be considered as a friendly name and cannot be used to distinguish different systems.
ServerDisplayVersion	String – The server version. (API version 1.5)
Hardware	The system hardware (API version 1.7)
ModelName	String – The name of the hardware model. (API version 1.7)
Vendor	String – The name of the hardware vendor. (API version 1.7)
TimeZone	String – The time zone of the server in IANA format. (API version 2.12)

Table: GetSystem response

Example response:
HTTP Response: 200 OK


```
{
    "Id": "018b7214-cd17-4eab-bc0a-83ed4753feee",
    "Name": "Test and Demo system",
    "ServerDisplayVersion": "5.46.400",
    "Hardware": {
        "ModelName": "HP ZBook 15 G6",
        "Vendor": "HP"
    },
    "TimeZone": "Europe/Berlin"
}
```


The operation is available from API version 1.3 and later.
ServerDisplayVersion is available in API version 1.5 and later.
Hardware, ModelName, and Vendor are available in API version 1.7 and later.
TimeZone is available in API version 2.12 and later.

11.3. ServerConfigurationFacade

Provides system configuration information.

GetServerConfiguration

Gets the complete server configuration.

URI: `/Acs/Api/ServerConfigurationFacade/GetServerConfiguration`

Request: The request lacks body.

Example request:
`https://172.25.127.28:55756/Acs/Api/ServerConfigurationFacade/GetServerConfiguration`

Response:

Field	Description
CameraSettings	The list of camera settings
CameraName	String – The user chosen camera name.
Manufacturer	String – The Manufacturer of the camera.
Model	String – The camera Model
FirmwareVersion	String – The firmware version
Address	String – The IP Address of the camera.
HttpPort	Integer – The HTTP port to use for communication.
HostName	Integer –The desired framerate for the quality level.
MacAddress	String – The MAC address of the camera.
VideoViewToken	String – The video view token of the camera.
CameraId	The camera Id.
Id	String – The unique identifier of the camera.
IsEnabled	Boolean – If the camera is enabled.
DisconnectSinceServerStart	Integer – Number of disconnects since server start.

HasPtz	Boolean – If the camera has PTZ.
VideoAudioSettings	The list of video audio settings.
Id	String – The unique identifier of the camera.
HasAudio	Boolean – If the camera has audio.
LiveViewAudio	Boolean – If the camera uses audio for live view.
RecordingAudio	Boolean – If the camera uses audio for recordings.
High	The media profile high. *See table: Media profiles*. In version 2.10 and later this field will be missing on some cameras depending on the number of media profiles they have.
Medium	The media profile medium. *See table: Media profiles*. In version 2.10 and later this field will be missing on some cameras depending on the number of media profiles they have.
Low	The media profile low. *See table: Media profiles*. In version 2.10 and later this field will be missing on some cameras depending on the number of media profiles they have.

Table: GetServerConfiguration response

Field	Description
QualityLevel	Integer –The quality level of the media profile. *See table QualityLevels*.
Framerate	Integer –The desired framerate for the quality level.
Resolution	String – The Media profile resolution. Can be either numeric (e.g., “1280x720”) or textual (e.g., “4CIF”).
VideoEncoding	Integer – The Media profile video encoding. *See table VideoEncodings*.

Table: Media profiles

QualityLevel	QualityLevel corresponding value
1	High
2	Medium
3	Low

Table: QualityLevels

VideoEncoding	VideoEncoding corresponding value
1	H.264
2	Mjpeg
3	H.265
4	MPEG-4

Table: VideoEncodings

Example response:

HTTP Response: 200 OK

```
{
    "CameraSettings": [
        {
            "CameraName": "AXIS M3005 (2)",
            "Manufacturer": "Axis",
            "Model": "AXIS M3005",
            "FirmwareVersion": "5.50.5.13",
            "Address": "m3005.axis.example.com",
            "HttpPort": 80,
            "HostName": " m3005.axis.example.com",
```

```
            "MacAddress": "00408CDC09C3",
            "VideoViewToken": "1",
            "CameraId": {
                "Id": "2438_54dd4102-96c7-4cdb-a8d2-85c2187a5d47"
            },
            "IsEnabled": true,
            "DisconnectSinceServerStart": 0,
            "HasPtz": false
        }
    ],
    "VideoAudioSettings": [
        {
            "Id": {
                "Id": "2438_54dd4102-96c7-4cdb-a8d2-85c2187a5d47"
            },
            "HasAudio": false,
            "LiveViewAudio": false,
            "RecordingAudio": false,
            "High": {
                "QualityLevel": 1,
                "Framerate": 25,
                "Resolution": "1920x1080",
                "VideoEncoding": 1
            },
            "Medium": {
                "QualityLevel": 2,
                "Framerate": 15,
                "Resolution": "1280x720",
                "VideoEncoding": 1
            },
            "Low": {
                "QualityLevel": 3,
                "Framerate": 5,
                "Resolution": "640x360",
                "VideoEncoding": 1
            }
        }
    ]
}
```


The operation is available from API version 1.6 and later.

11.4. CameraListFacade

Handles listing of cameras.

GetCameraList

Gets cameras sorted by name, limited to the specified range.

URI: `/Acs/Api/CameraListFacade/GetCameraList`

Request:

Field	Description
Range	The range of elements to be collected from the server
StartIndex	Integer - The starting index of the first camera in the reply.
NumberOfElements	Integer - The number of cameras to be in the reply.

Table: GetCameraList request

Example request:

```
https://172.25.127.28:55756/Acs/Api/CameraListFacade/GetCameraList
{
        "range" : {
                "StartIndex": 0,
                "NumberOfElements": 20
        }
}
```

Response:

Field	Description
Cameras	The list of cameras.
CameraId	The camera Id.
Id	String – The unique identifier of the camera.
DeviceId	The device Id for the device associated with the camera.
Id	String – The unique identifier of the device.
Name	String – The name of the camera. This should be considered as a friendly name and cannot be used to distinguish different systems.
Model	String – The camera model. (API version 1.5)
VideoViewToken	String – The camera view token. Image index for VAPIX devices and VideoSourceConfiguration token for ONVIF cameras.
MediaProfiles	The list of media profiles for the camera.
QualityLevel	Integer –The media profile quality level. *See table QualityLevels*.
Framerate	Integer –The desired framerate for the quality level.
Resolution	String – The media profile resolution. Can be either numeric (e.g., “1280x720”) or textual (e.g., “4CIF”).
VideoEncoding	Integer – Media profile video encoding. *See table VideoEncodings*.
Status	Integer – The status of the camera. *See table CameraStatus*.
CameraPrivileges	List of Integers – The user privileges granted for this camera. Empty if the user has no privileges. *See table CameraPrivileges*.
DeviceSerialNumber	String – The serial number of the camera.
Manufacturer	String – The Manufacturer of the camera. (API version 1.3)
ContainsLastCamera	Boolean – If the last camera is in this reply.

Table: GetCameraList response

QualityLevel	QualityLevel corresponding value
1	High
2	Medium
3	Low

Table: QualityLevels

VideoEncoding	VideoEncoding corresponding value
1	H.264
2	Mjpeg

3	H.265
4	MPEG-4

Table: VideoEncodings

Status	Status corresponding value
1	OK
2	Unauthenticated
3	Not Accessible

Table: CameraStatus

CameraPrivileges	CameraPrivileges corresponding value
1	Live view Access
2	Playback Access
3	Ptz Access
4	AudioInAccess
5	AudioOutAccess
6	ExportAccess
7	SnapshotAccess

Table: CameraPrivileges

Example response:
HTTP Response: 200 OK


```
{
   "Cameras": [
        {
            "CameraId": {
                "Id": "2542_54dd4102-96c7-4cdb-a8d2-85c2187a5d47"
            },
            "DeviceId": {
                "Id": "1240_54dd4102-96c7-4cdb-a8d2-85c2187a5d47"
            },
            "Name": "Reception",
            "Model": "AXIS Q1765-LE",
            "VideoViewToken": "0",
            "MediaProfiles": [
                {
                    "QualityLevel": 1,
                    "Framerate": 25,
                    "Resolution": "1920x1080",
                    "VideoEncoding": 1
                },
                {
                    "QualityLevel": 2,
                    "Framerate": 15,
                    "Resolution": "1280x720",
                    "VideoEncoding": 1
                },
                {
                    "QualityLevel": 3,
                    "Framerate": 5,
                    "Resolution": "640x360",
                    "VideoEncoding": 1
                }
            ],
```

```
            "Status": 1,
            "CameraPrivileges": [
                1,
                2,
                3,
                4,
                5,
                7
            ],
            "DeviceSerialNumber": "ACCC8E05DD5D",
            "Manufacturer": "Axis"
        }
    ],
    "ContainsLastCamera": true
}
```


The operation is available from API version 1.3 and later. The Manufacturer field is available in API version 1.3 and later. The Model field is available in API version 1.5 and later. The CameraPrivileges, HasAudioIn and HasAudioOut fields are available from version 2.0 and later. The DeviceId and VideoViewToken fields are available from version 2.16 and later. CameraPrivileges, ExportAccess, and SnapshotAccess are available from version 2.30.

GetNumberOfCameras

Gets the total number of cameras.

URI: `/Acs/Api/CameraListFacade/GetNumberOfCameras`

Request: No parameters are used in the request body.

Example request:
`https://172.25.127.28:55756/Acs/Api/CameraListFacade/GetNumberOfCameras`

Response:

Field	Description
NumberOfCameras	Integer – The number of cameras.

Table: GetNumberOfCameras response

Example response: HTTP Response: 200 OK


```
{
    "NumberOfCameras": 7
}
```


The operation is available from API version 1.3 and later.

11.5. CameraFacade

Handles general camera functionality.

GetCameraCapabilities

Gets capabilities for cameras sorted by name, limited to the specified range.

URI: `/Acs/Api/CameraFacade/GetCameraCapabilities`

Request:

Field	Description
range	The range of elements to be collected from the server
StartIndex	Integer - The starting index of the first camera in the reply.
NumberOfElements	Integer - The number of cameras to be in the reply.

Table: GetCameraCapabilities request

Example request:

```
https://172.25.127.28:55756/Acs/Api/CameraFacade/GetCameraCapabilities
{
   "range" : {
        "StartIndex": 0,
        "NumberOfElements": 5
    }
}
```

Response:

Field	Description
Capabilities	The list of capabilities.
Camera	The camera.
Id	String – The unique identifier of the camera.
HasPtz	Boolean – If the camera has PTZ capabilities.
HasSpeaker	Boolean – If the camera has a speaker for audio transmission. Can either be camera internal or connected externally through Axis Camera Station.
DewarpCapability	Present in response only if the camera has dewarping capabilities. Tilt orientation and lens parameters are provided.
LensParameters	The lens parameter used for dewarping.
RadialDistortionX	Double – The radial distortion coefficient X value.
RadialDistortionY	Double – The radial distortion coefficient Y value.
RadialDistortionZ	Double – The radial distortion coefficient Z value.
OpticalCenterX	Double – The X coordinate of the optical center.
OpticalCenterY	Double – The Y coordinate of the optical center.
TiltOrientation	Integer – The tilt orientation. *See TiltOrientations table*.
ContainsLastCapability	Boolean – If the last camera is in this reply.

Table: GetCameraCapabilities response

TiltOrientations	TiltOrientations corresponding value
0	Not Applicable, e.g., regular camera.
1	Ceiling
2	Desk
3	Wall

Table: TiltOrientations

Example response: HTTP Response: 200 OK


```
{
    "Capabilities": [
        {
            "Camera": {
                "Id": "1821_6d83ad9b-1d27-409b-85f8-2656485bdb40"
            },
            "HasPtz": false,
            "HasSpeaker": true
        },
        {
            "Camera": {
                "Id": "3896_6d83ad9b-1d27-409b-85f8-2656485bdb40"
            },
            "HasPtz": false,
            "DewarpCapability": {
                "LensParameters": {
                    "RadialDistortionX": -43.970703,
                    "RadialDistortionY": 29.148499,
                    "RadialDistortionZ": 715.732193,
                    "OpticalCenterX": 1296.0,
                    "OpticalCenterY": 972.0
                },
                "TiltOrientation": 1
            },
            "HasSpeaker": false
        }
    ],
    "ContainsLastCapability": true
}
```


The operation is available from API version 1.5 and later. HasSpeaker is included from API version 1.8 and later.

AddCamera

Adds a camera where a camera is a device with video source(s).

URI: `/Acs/Api/CameraFacade/AddCamera`

Supported Http Methods

- POST

Request:

Field	Subfield	Description
ConnectionInfo	**Address**	String - The IP address or hostname of the camera.
	Port	Integer - The port to use when communicating with the camera.
AuthenticationInfo	**Username**	String - The username to use for authentication. Default name is the same name as when using the discovery function in the AXIS Camera Station client.
	Password	String - The password to use for authentication.

	SecurityMode	String - Optional. The scheme to use during communication with the camera. Default is HttpDigest. Valid values are: HttpBasic, HttpsBasic, HttpDigest, HttpsDigest.
Options - Optional	**Name**	String – Optional. The name used for the camera. A default name based on the camera model will be used if not provided.
	Description	String – Optional. A description to add to the camera.
	RetentionTime	Integer – Optional. The retention time of the recording in days. Value between 0-9999. A value of 0 will prevent deletion of any recordings. Default retention time is 90 days.
	ViewToken	The token used to identify a view (camera) on a device with multiple views. E.g., "0", "1" on a camera implementing VAPIX. First available view will be used if not provided.

Table: AddCamera request

Example request:

```
https://172.25.127.28:55756/Acs/Api/CameraFacade/AddCamera
{
    "connectionInfo": {
        "address": "m3005.axis.example.com",
        "port": "443"
    },
    "AuthenticationInfo": {
        "username": "testuser",
        "password": "testpassword",
        "securityMode": "HttpsDigest"
    },
    "Options": {
        "name": "MyCamera",
        "description": "My Description",
        "retentionTime": 0,
        "ViewToken": "0"
    }
}
```

Response: The response includes camera ID for the added camera. An exception may be thrown.
Example response: HTTP Response: 200 OK

```
{
    "CameraId": {
        "Id": "17820_98b5f40a-1d27-4e86-b72e-33def46e74eb"
    }
}
```

The operation is available from API version 2.17 and later.

11.6. DeviceListFacade

GetDeviceList

Gets devices sorted by device id, limited to the specified range.

URI: `/Acs/Api/DeviceListFacade/GetDeviceList`

Request:

Field	Description
range	The range of elements to be collected from the server
StartIndex	Integer - The starting index of the first device in the reply.
NumberOfElements	Integer - The number of devices to be in the reply.

Table: GetDeviceList request

Example request:

```
https://172.25.127.28:55756/Acs/Api/DeviceListFacade/GetDeviceList
{
        "range" : {
                "StartIndex": 0,
                "NumberOfElements": 10
        }
}
```

Response:

Field	Description
Devices	The list of devices.
DeviceId	The device Id.
Id	String – The unique identifier of the device.
Name	String – The name of the device. This should be considered as a friendly name and cannot be used to distinguish different systems.
MacAddress	String – The mac address of the device.
ProductType	String – The type of the device.
Model	String – The device model.
FirmwareVersion	The version of the device firmware.
Address	String –The IP address or the hostname of the device.
Port	Integer –The device port.
AuthenticationMethod	String – The method used for authentication: basic or digest.
Protocol	String – Protocol used for device communication: HTTP or HTTPS.

Table: GetDeviceList response

ProductType
3D People Counter
Media Uri Device
Network Camera
Network Dome Camera
Network Door Controller
Network IO Audio Module
Network IO Module
Network PACS prototype
Network Radar Detector
Network Speaker
Network Switch
Network Video Decoder
Network Video Door Station
Network Video Encoder
Network Video Recorder
Networked Door Controller
Onvif Network Camera
PTZ Dome Network Camera
PTZ Network Camera
Security Radar
Temperature Alarm Camera
Thermal Network Camera
Video Server
Virtual Camera

Table: ProductType values

Example response:

HTTP Response: 200 OK


```
{
  "Devices": [
    {
      "DeviceId": {
        "Id": "10583_7743fe02-c360-4de7-8420-929190d15712"
      },
      "Name": "AXIS P5635-E",
      "MacAddress": "00408CCD3333",
      "ProductType": "PTZ Dome Network Camera",
      "Model": "AXIS P5635-E",
      "FirmwareVersion": "6.50.4.1",
      "Address": "172.25.126.212",
      "Port": 80,
      "AuthenticationMethod": "digest",
      "Protocol": "http"
    }
  ],
  "ContainsLastDevice": true
}
```


11.7. SnapshotFacade

Handles camera live stream snapshot.

GetSnapshot

Get snapshots for specified cameras. Cameras where the user has no access to live view will be excluded. A client should check permissions before requesting snapshots (e.g., by the info provided by the camera list) and avoid requesting snapshots for such cameras. Cameras where status != OK will also be excluded as well as cameras with no snapshot available.

URI: `/Acs/Api/SnapshotFacade/GetSnapshot`

Request:

Field	Description
cameraIds	List of the IDs of the cameras to get snapshots from.
Id	String – The unique identifier of the camera.
PreferredResolution	The preferred resolutions of the snapshot, there are no guarantees that this is the actual resolution of the snapshot in the reply.
Width	Integer – Width of the snapshot
Height	Integer – Height of the snapshot

Table: GetSnapshot request

Example request:

```
https://172.25.127.28:55756/Acs/Api/SnapshotFacade/GetSnapshot
{
        "cameraIds" : [
                {
                        "Id" : "2502_54dd4102-96c7-4cdb-a8d2-85c2187a5d47"
                },
                {
                        "Id" : "2542_54dd4102-96c7-4cdb-a8d2-85c2187a5d47"
                }
        ],
        "preferredResolution" : {
                "Width" : 640,
                "Height" : 480
        }
}
```

Response:

Field	Description
Snapshots	The list of snapshots.
CameraId	The camera Id.
Id	String – The unique identifier of the camera.
ImageData	String – The JPEG image data.

Table: GetSnapshot response

Example response. All image data is not shown below:

HTTP Response: 200 OK

```
{
    "Snapshots": [
        {
```

```
            "CameraId": {
                "Id": "2502_54dd4102-96c7-4cdb-a8d2-85c2187a5d47"
            },
            "ImageData": "/9j/4AAQSkZJRgABAgAAAQABAAD//gAPCgF"
        },
        {
            "CameraId": {
                "Id": "2542_54dd4102-96c7-4cdb-a8d2-85c2187a5d47"
            },
            "ImageData": "/9j/4AAQSkZJRgABAgAAAQABAAD//gAPCgF"
        }
    ]
}
```


The operation is available from API version 1.3 and later.

11.8. RecordingSnapshotFacade

Handles snapshots from recordings.

GetSnapshots

Gets snapshots for specified cameras and time. An ApiException will be thrown if requesting snapshots from cameras without access privilege. If no recording is available, no snapshot will be returned.
The time returned in the response is the time specified in the request. This requested time can differ from the actual time the snapshot was taken (especially when useDeltaFrames is false).

URI: `/Acs/Api/RecordingSnapshotFacade/GetSnapshots`

Request:

Field	Description
snapshotRequests	List of snapshot requests.
cameraId	The camera identification information.
Id	String – The unique identifier of the camera
Time	String – The requested time in UTC and “YYYY-MM-DD hh:mm:ss.fffffff” format. The second fraction digits can be omitted.
useDeltaFrames	String – Optional. Boolean value, true if delta frames should be used. False otherwise.
requestedQualityLevel	String – Optional. Requested quality level for the snapshots. Should be one of high, medium, low, lowestavailable, or highestavailable. (Default value is lowestavailable.)
jpegEncodingQualityPercent	Integer – Optional. Quality level in percent to be used by the JPEG encoder. (Default value is 75 percent.)
preferredResolution	Optional. The preferred resolutions of the snapshot. There is no guarantee that this is the actual resolution of the snapshot in the reply.
Width	Integer – Width of the snapshot
Height	Integer – Height of the snapshot

Table: GetSnapshots request

Example request:

```
https://172.25.127.28:55756/Acs/Api/RecordingSnapshotFacade/GetSnapshots
{
      "snapshotRequests": [
            {
                  "cameraId": {
                        "Id": "2542_54dd4102-96c7-4cdb-a8d2-85c2187a5d47"
                  },
                  "time": "2019-04-02 11:59:30"
            },
            {
                  "cameraId": {
                        "Id": "2542_54dd4102-96c7-4cdb-a8d2-85c2187a5d47"
                  },
                  "time": "2019-04-02 12:00:00"
            }
      ],
        "useDeltaFrames": "true" ,
        "requestedQualityLevel": "highestavailable",
        "jpegEncodingQualityPercent": 90,
        "preferredResolution" : {
               "Width" : 640,
               "Height" : 480
        }
}
```

Response:

Field	Description
CameraId	The camera identification information.
Id	String – The unique identifier of the camera
Time	String – Requested time in UTC and in “YYYY-MM-DD hh:mm:ss.fffffff” format.
ImageData	String – The Base64 encoded JPEG image data

Table: GetSnapshots response

Example response. HTTP Response: 200 OK. All image data is not shown below:

```
[
    {
        "CameraId": {
            "Id": "2542_54dd4102-96c7-4cdb-a8d2-85c2187a5d47"
        },
        "Time": "2019-04-02 11:59:30.0000000",
        "ImageData": "/9j/4AAQSkZJRgABAQEAYABgAAD/2wBDA"
    },
    {
        "CameraId": {
            "Id": "2542_54dd4102-96c7-4cdb-a8d2-85c2187a5d47"
        },
        "Time": "2019-04-02 12:00:00.0000000",
        "ImageData": "/9j/4AAQSkZJRgABAQEAYABgAAD/2wBDA"
    }
]
```

Available from API version 1.7 and later. Optional quality parameters available from API version 2.17. Optional preferredResolution parameter available from API version 2.30.

11.9. RecordingFacade

Handles the recordings.

GetRecordedMedia

Gets recordings limited to the specified cameras, interval, and range. All recordings that have some data in the interval will be included, even if they start or end outside the interval. The start time and end time will be excluded in the response if they are outside the search interval.

URI: `/Acs/Api/RecordingFacade/GetRecordedMedia`

Request:

Field	Description
cameraIds	List of the IDs of the cameras to get recordings from.
Id	String – The unique identifier of the camera.
interval	The time interval to get recordings from.
StartTime	String – The start time in UTC and “YYYY-MM-DD hh:mm:ss” format.
StopTime	String – The stop time in UTC and “YYYY-MM-DD hh:mm:ss” format.
Range	The range of element to be collected from the server
StartIndex	Integer - The starting index of the first camera in the reply.
NumberOfElements	Integer - The number of cameras to be in the reply.

Table: GetRecordedMedia request

Example request:

```
https://172.25.127.28:55756/Acs/Api/RecordingFacade/GetRecordedMedia
{
    "cameraIds": [
        {
            "Id": "2250_e9f9497f-31a0-4e58-9d13-bb44ed78f294"
        }
    ],
    "interval": {
        "StartTime": "2018-10-25 12:00:00",
        "StopTime": "2018-10-25 13:00:00"
    },
    "range": {
        "StartIndex": 0,
        "NumberOfElements": 5
    }
}
```

Response:

Field	Description
RecordedMedia	The list of Recordings.
CameraId	The camera Id.
Id	String – The unique identifier of the camera.

StartTime	String – Recording start time in UTC, in "YYYY-MM-DD hh:mm:ss.fffffff" format. Excluded if it is outside the search interval.
EndTime	String – Recording end time in UTC, in "YYYY-MM-DD hh:mm:ss.fffffff" format. Excluded if it is outside the search interval.
QualityLevel	Integer –The quality level of the recording. *See table QualityLevels*.
VideoTrack	Contains information about video in a recording.
Encoding	Integer –The video encoding. *See table VideoEncodings*.
AudioTrack	Contains information about audio in a recording.
AudioFormat	Integer –The audio format. *See table AudioFormats*.
ContainsLastResult	Boolean – If the last recording in the interval is in this response.

Table: GetRecordedMedia response

QualityLevels	QualityLevels corresponding value
1	High
2	Medium
3	Low

Table: QualityLevels

VideoEncodings	VideoEncodings corresponding value
1	H.264
2	Mjpeg
3	H.265
4	MPEG-4

Table: VideoEncodings

AudioFormats	AudioFormats corresponding value
1	None, recording lacks audio
2	G.711
3	G.726
4	AAC

Table: AudioFormats

Example response.

HTTP Response: 200 OK

```
{
  "RecordedMedia": [
    {
      "CameraId": {
        "Id": "2250_e9f9497f-31a0-4e58-9d13-bb44ed78f294"
      },
      "StartTime": "2018-10-25 12:15:49.3228688",
      "EndTime": "2018-10-25 12:32:44.3218022",
      "QualityLevel": 2,
      "VideoTrack": {
        "Encoding": 1
      },
      "AudioTrack": {
        "AudioFormat": 4
      }
    },
    {
```

```
      "CameraId": {
        "Id": "2250_e9f9497f-31a0-4e58-9d13-bb44ed78f294"
      },
      "StartTime": "2018-10-25 12:47:46.6418022",
      "QualityLevel": 2,
      "VideoTrack": {
        "Encoding": 1
      },
      "AudioTrack": {
        "AudioFormat": 4
      }
    }
  ],
  "ContainsLastResult": true
}
```


The operation is available from API version 1.7 and later.

GetRecordings

The recommended way of retrieving recordings is using **GetRecordedMedia**. **GetRecordings** will be removed in a future version of AXIS Camera Station API.

Gets recordings limited to the specified cameras, interval, and range. All recordings that have some data in the interval will be included, even if they start or end outside the interval. A recording belongs to a sequence, and a sequence can contain one or several recordings. If a sequence consists of several recordings, only the recordings contained in the specified interval will be included in the result. The start and end time of the recordings will be that of the recordings, i.e., they will not be altered according to the search interval or sequence times. Recordings starting at the interval stop time or ending at the interval start time will not be included. When limiting the result to a specific range this will be done according to recording start time in ascending order.

URI: `/Acs/Api/RecordingFacade/GetRecordings`

Request:

Field	Description
cameraIds	List of the IDs of the cameras to get recordings from.
Id	String – The unique identifier of the camera.
interval	The time interval to get recordings from.
StartTime	String – The start time in UTC, in “YYYY-MM-DD hh:mm:ss” format.
StopTime	String – The stop time in UTC, in “YYYY-MM-DD hh:mm:ss” format.
range	The range of elements to be collected from the server
StartIndex	Integer - The starting index of the first camera in the response.
NumberOfElements	Integer - The number of cameras to be in the response.

Table: GetRecordings request

Example request: `https://172.25.127.28:55756/Acs/Api/RecordingFacade/GetRecordings`

```
{
        "cameraIds": [
                {
                        "Id": "2542_54dd4102-96c7-4cdb-a8d2-85c2187a5d47"
                },
                {
                        "Id": "2438_54dd4102-96c7-4cdb-a8d2-85c2187a5d47"
                }
        ],
        "interval": {
                "StartTime": "2019-04-02 12:00:00",
                "StopTime": "2019-05-02 13:00:00"
        },
        "range" : {
                "StartIndex": 0,
                "NumberOfElements": 20
        }
}
```

Response:

Field	Description
Recordings	The list of Recordings.
SequenceId	This id is for internal use only.
CameraId	The camera Id.
Id	String – The unique identifier of the camera.
IsOngoing	Boolean – If the recording is still ongoing
SequenceStartTime	String – Sequence start time in UTC, in "YYYY-MM-DD hh:mm:ss" format.
TriggerTime	String – Trigger time of the recording in UTC, in "YYYY-MM-DD hh:mm:ss" format.
StartTime	String – Recording start time in UTC, in "YYYY-MM-DD hh:mm:ss" format.
EndTime	String – Recording end time in UTC, in "YYYY-MM-DD hh:mm:ss" format. EndTime is set even if the recording is ongoing, i.e., there is a playback limit even for ongoing recordings.
RecordingTrigger	Integer –What triggered the recording. *See table RecordingTriggers*.
VideoEncoding	Integer –The video encoding. *See table VideoEncodings*.
Resolution	String – The recording resolution can either be numeric (e.g., "1280x720") or textual (e.g., "4CIF").
FrameRate	Integer – The number of frames per second.
AudioFormat	Integer –The audio format. *See table AudioFormats*.
ContainsLastRecording	Boolean – If the last recording in the interval is in this reply.

Table: GetRecordings response

RecordingTriggers	RecordingTriggers corresponding value
1	Motion detection
2	Manual
3	Continuous
4	Failover
5	Rule

Table: RecordingTriggers

VideoEncodings	VideoEncodings corresponding value
1	H.264
2	Mjpeg
3	H.265
4	MPEG-4

Table: VideoEncodings

AudioFormats	AudioFormats corresponding value
1	None, recording lacks audio
2	G.711
3	G.726
4	AAC

Table: AudioFormats

Example response. HTTP Response: 200 OK


```
{
    "Recordings": [
        {
            "SequenceId": {
                "Id": "11_7490bc51-7990-460d-b14e-4e5bcd668b6e"
            },
            "CameraId": {
                "Id": "2288_7490bc51-7990-460d-b14e-4e5bcd668b6e"
            },
            "IsOngoing": true,
            "SequenceStartTime": "2019-05-02 09:01:13.9179539",
            "TriggerTime": "2019-05-02 09:01:13.9179539",
            "StartTime": "2019-05-02 09:01:14.9116500",
            "EndTime": "2019-05-02 09:05:06.6184820",
            "RecordingTrigger": 3,
            "VideoEncoding": 1,
            "Resolution": "1280x720",
            "FrameRate": 15,
            "AudioFormat": 1
        }
    ],
    "ContainsLastRecording": true
}
```


The operation is available from API version 1.3 and later.
The operation returns seconds decimal precision from API version 2.0 and later.

11.10. RecordedEventFacade

Handles the recorded events. These events can be mapped with the recordings and, for example, used for presenting what triggered a recording.

GetRecordedEvents

Retrieves a list of all events recorded during a specified time interval.

URI: `/Acs/Api/RecordedEventFacade/GetRecordedEvents`

Request:

Field	Description
CameraIds	List of the IDs of the cameras to get recorded events from.
Id	String – The unique identifier of the camera.
Interval	The time interval to get recorded events from.
StartTime	String – The start time in UTC and “YYYY-MM-DD hh:mm:ss” format.
StopTime	String – The stop time in UTC and “YYYY-MM-DD hh:mm:ss” format.
Range	The range of elements to be collected from the server.
StartIndex	Integer - The starting index of the first recorded event in the reply.
NumberOfElements	Integer - The number of recorded events to be in the reply.

Table: GetRecordedEvents request

Example request:

```
https://172.25.127.28:55756/Acs/Api/RecordedEventFacade/GetRecordedEvent
{
    "CameraIds": [
        {
            "Id": "2250_e9f9497f-31a0-4e58-9d13-bb44ed78f294"
        }
    ],
    "Interval": {
        "StartTime": "2018-10-25 12:00:00",
        "StopTime": "2018-10-25 13:00:00"
    },
    "Range": {
        "StartIndex": 0,
        "NumberOfElements": 5
    }
}
```

Response:

Field	Description
RecordedEvents	The list of recorded events.
CameraId	The camera Id.
Id	String – The unique identifier of the camera.
StartTime	String – Event start time in UTC, in “YYYY-MM-DD hh:mm:ss.ffffff” format. Excluded if it is outside the search interval.
EndTime	String – Event end time in UTC, in “YYYY-MM-DD hh:mm:ss.ffffff” format. Excluded if it is outside the search interval.
Type	The Type can be matched with an ID of a RecordedEventType from GetRecordedEventTypes,
ContainsLastResult	Boolean – If the last event in the interval is in this response.

Table: GetRecordedEvents response

Example response: HTTP Response: 200 OK

```
{
  "RecordedEvents": [
    {
      "CameraId": {
        "Id": "2250_e9f9497f-31a0-4e58-9d13-bb44ed78f294"
      },
```

```
      "StartTime": "2018-10-25 12:15:49.3228688",
      "EndTime": "2018-10-25 12:32:44.3218022",
      "Type": "Motion"
    },
    {
      "CameraId": {
        "Id": "2250_e9f9497f-31a0-4e58-9d13-bb44ed78f294"
      },
      "StartTime": "2018-10-25 12:47:46.6418022",
      "Type": "Manual"
    }
  ],
  "ContainsLastResult": true
}
```


GetRecordedEventTypes

Used to retrieve a list of all event types.

URI: `/Acs/Api/RecordedEventFacade/GetRecordedEventTypes`

Response:

Field	Description
RecordedEventTypes	The list of event types
Name	The name of the event type. Used as Id.
Title	Name in server language.
Description	Description of the event type in the server language.
Color	Dictionary with the different color themes.

Table: GetRecordedEventTypes response

Example response: HTTP Response: 200 OK


```
{
    "RecordedEventTypes": [
        {
            "Name": "Motion",
            "Title": "Motion",
            "Description": "",
            "Color": {
                lightTheme: "#FF0000",
                darkTheme: "#FF0000"
            }
        },
        {
            "Name": "Manual",
            "Title": "Manual",
            "Description": "",
            "Color": {
                lightTheme: "#FFFF00",
                darkTheme: "#FFFF00"
            }
        },
        ....
```

```
    ]
}
```

11.11. RecordingControlFacade

Handles the recording status for cameras.

StartRecording

Sets the manual recording status for the specified camera.

Note that a successful response does not mean that recording for the camera is ongoing, only that it will attempt to record. Cases such as a full disk or network issues would prevent recording, but not the setting of the camera's recording state.

URI: `/Acs/Api/RecordingControlFacade/StartRecording`

Request:

Field	Description
cameraId	The camera identification information.
Id	String – The unique identifier of the camera

Table: StartRecording request

Example request:

```
https://172.25.127.28:55756/Acs/Api/RecordingControlFacade/StartRecording
{
        "cameraId": {
                "Id": "2542_54dd4102-96c7-4cdb-a8d2-85c2187a5d47"
        }
}
```

Response: The operation lacks response body. An exception may be thrown.

Example response:

HTTP Response: 200 OK

The operation is available from API version 2.16 and later.

StopRecording

Sets the manual recording status for the specified camera as stopped.

URI: `/Acs/Api/RecordingControlFacade/StopRecording`

Request:

Field	Description
cameraId	The camera identification information.
Id	String – The unique identifier of the camera

Table: StopRecording request

Example request:

```
https://172.25.127.28:55756/Acs/Api/RecordingControlFacade/StopRecording
{
        "cameraId": {
                "Id": "2542_54dd4102-96c7-4cdb-a8d2-85c2187a5d47"
        }
}
```

Response: The operation lacks response body. An exception may be thrown.

Example response: HTTP Response: 200 OK

The operation is available from API version 2.16 and later.

GetRecordingStatus

Gets the recording status of the specified camera's manual recording.

URI: `/Acs/Api/RecordingControlFacade/GetRecordingStatus`

Request:

Field	Description
cameraId	The camera identification information.
Id	String – The unique identifier of the camera

Table: GetRecordingStatus request

Example request:

```
https://172.25.127.28:55756/Acs/Api/RecordingControlFacade/GetRecordingStatus
{
        "cameraId": {
                "Id": "2542_54dd4102-96c7-4cdb-a8d2-85c2187a5d47"
        }
}
```

Response:

Field	Description
IsRecording	Whether the camera's manual recording profile is currently recording.

Table: GetRecordingStatus response

Example response: HTTP Response: 200 OK

```
{
        "IsRecording": true
}
```

The operation is available from API version 2.16 and later.

11.12. PtzFacade

Handles PTZ functionality.

Exceptions

ApiException
The operations in this facade will throw ApiException if a camera id to a camera without PTZ capabilities is provided for a PTZ operation.

UnauthorizedException
The operations in this façade will throw UnauthorizedException if the user doesn't have access to the camera being operated on.

PtzPriorityException
This exception can only be thrown from an AXIS Camera Station server where PTZ priority is configured in the user permissions. It is thrown when the operation is blocked because of an ongoing PTZ operation on the same camera by a user who has a higher priority. The block will be lifted after a predefined time from the latest PTZ movement. This time defaults to 10 seconds. This exception is available from API version 2.5 and later.

GetPtzCapabilities

Lists the PTZ capabilities for a range of cameras.

URI: `/Acs/Api/PtzFacade/GetPtzCapabilities`

Request:

Field	Description
Range	The range of elements to be collected from the server
StartIndex	Integer - The starting index of the first camera in the response.
NumberOfElements	Integer - The number of cameras to be in the resopnse.

Table: GetPtzCapabilities request

Example request:

```
https://172.25.127.28:55756/Acs/Api/PtzFacade/GetPtzCapabilities
{
        "range" : {
                "StartIndex": 0,
                "NumberOfElements": 20
        }
}
```

Response:

Field	Description
Capabilities	The list of capabilities.
Camera	The camera.
Id	String – The unique identifier of the camera.
HasPtz	Boolean – If the camera has PTZ capabilities.
ContainsLastCapability	Boolean – If the response contains the last capability

Table: GetPtzCapabilities response

Example response: HTTP Response: 200 OK

```
{
    "Capabilities": [
        {
```

```
            "Camera": {
                "Id": "2465_54dd4102-96c7-4cdb-a8d2-85c2187a5d47"
            },
            "HasPtz": false
        },
        {
            "Camera": {
                "Id": "2542_54dd4102-96c7-4cdb-a8d2-85c2187a5d47"
            },
            "HasPtz": true
        }
    ],
    "ContainsLastCapability": true
}
```

The operation is available from API version 1.1 and later.

GetCameraPtzCapabilities

Gets PTZ capabilities for the specified camera.

URI: `/Acs/Api/PtzFacade/GetCameraPtzCapabilities`

Request:

Field	Description
cameraId	The camera Id.
Id	String – The unique identifier of the camera.

Table: GetCameraPtzCapabilities request

Example request:

```
https://127.0.0.1:55756/Acs/Api/PtzFacade/GetCameraPtzCapabilities
{
        "cameraId" : {
                "Id" : "2502_54dd4102-96c7-4cdb-a8d2-85c2187a5d47"
        }
}
```

Response:

Field	Description
Camera	The camera.
Id	String – The unique identifier of the camera.
HasPtz	Boolean – If the camera has PTZ capabilities.

Table: GetCameraPtzCapabilities response

Example response: HTTP Response: 200 OK

```
{
    "Camera": {
        "Id": "2502_54dd4102-96c7-4cdb-a8d2-85c2187a5d47"
    },
    "HasPtz": false
}
```

The operation is available from API version 1.1 and later.

Move

Moves a PTZ camera in a specified direction

URI: `/Acs/Api/PtzFacade/Move`

Request:

Field	Description
cameraId	The camera Id.
Id	String – The unique identifier of the camera.
direction	Integer – The direction to move the camera in. *See table Directions*.

Table: Move request

Direction	Direction corresponding value
0	Left
1	Right
2	Up
3	Down

Table: Directions

Example request:

```
https://127.0.0.1:55756/Acs/Api/PtzFacade/Move
{
        "cameraId" : {
                "Id" : "2502_54dd4102-96c7-4cdb-a8d2-85c2187a5d47"
        },
        "direction" : 1
}
```

Response: The operation lacks response body. A PtzPriorityException may be thrown.

Example response: HTTP Response: 200 OK

The operation is available from API version 1.1 and later.

Zoom

Zooms a PTZ camera a specified number of steps.

URI: `/Acs/Api/PtzFacade/Zoom`

Request:

Field	Description
cameraId	The camera Id.
Id	String – The unique identifier of the camera.
Steps	Integer – The number of steps to zoom.

Table: Zoom request

Example request:

```
https://127.0.0.1:55756/Acs/Api/PtzFacade/Zoom
{
        "cameraId" : {
                "Id" : "2448_54dd4102-96c7-4cdb-a8d2-85c2187a5d47"
        },
        "steps" : 100
```

```
}
```

Response: The operation lacks response body. A PtzPriorityException may be thrown.

Example response: HTTP Response: 200 OK

The operation is available from API version 1.1 and later.

Center

Centers a PTZ camera on the provided coordinates.

URI: `/Acs/Api/PtzFacade/Center`

Request:

Field	Description
cameraId	The camera Id.
Id	String – The unique identifier of the camera.
Coordinates	The coordinates to center the camera around.
X	The X component of the coordinates. Must be between 0 and ImageWidth.
Y	The Y component of the coordinates. Must be between 0 and ImageHeight.
ImageWidth	The image width.
ImageHeight	The image height.

Table: Center request

Example request:

```
https://172.25.127.28:55756/Acs/Api/PtzFacade/Center
{
        "cameraId" : {
                "Id" : "2448_54dd4102-96c7-4cdb-a8d2-85c2187a5d47"
        },
        "coordinates" : {
                "X" : 50,
                "Y" : 50,
                "ImageWidth" : 640,
                "ImageHeight" : 480
        }
}
```

Response: The operation lacks response body. A PtzPriorityException may be thrown.

Example response: HTTP Response: 200 OK

The operation is available from API version 1.1 and later.

ContinuousPanAndTilt

Moves a PTZ camera with the specified speed.

URI: `/Acs/Api/PtzFacade/ContinuousPanAndTilt`

Request:

Field	Description
cameraId	The camera Id.
Id	String – The unique identifier of the camera.
panSpeed	Integer – The Pan Speed
tiltSpeed	Integer – The tilt speed

Table: ContinuousPanAndTilt request

Example request:

```
https://172.25.127.28:55756/Acs/Api/PtzFacade/ContinuousPanAndTilt
{
        "cameraId" : {
                "Id" : "2448_54dd4102-96c7-4cdb-a8d2-85c2187a5d47"
        },
        "panSpeed" : 0,
        "tiltSpeed" : 0
}
```

Response: The operation lacks response body. A PtzPriorityException may be thrown.

Example response: HTTP Response: 200 OK

The operation is available from API version 1.1 and later.

ContinuousZoom

Zooms a PTZ capable camera at the specified speed.

URI: `/Acs/Api/PtzFacade/ContinuousZoom`

Request:

Field	Description
cameraId	The camera Id.
Id	String – The unique identifier of the camera.
Speed	Integer – The zoom speed.

Table: ContinuousZoom request

Example request:

```
https://172.25.127.28:55756/Acs/Api/PtzFacade/ContinuousZoom
{
        "cameraId" : {
                "Id" : "2448_54dd4102-96c7-4cdb-a8d2-85c2187a5d47"
        },
        "speed" : 0
}
```

Response: The operation lacks response body.

A PtzPriorityException may be thrown.

Example response: HTTP Response: 200 OK

The operation is available from API version 1.1 and later.

GotoPresetToken

Moves a PTZ camera to the specified preset.

URI: `/Acs/Api/PtzFacade/GotoPresetToken`

Request:

Field	Description
cameraId	The camera Id.
Id	String – The unique identifier of the camera.
presetToken	String – The token of the preset to go to.

Table: GotoPresetToken request

Example request:

```
https://172.25.127.28:55756/Acs/Api/PtzFacade/GotoPresetToken
{
        "cameraId" : {
                "Id" : "2448_54dd4102-96c7-4cdb-a8d2-85c2187a5d47"
        },
        "presetToken" : "Home"
}
```

Response: The operation lacks response body. A PtzPriorityException may be thrown.

Example response: HTTP Response: 200 OK

The operation is available from API version 1.6 and later.

GotoPresetId

Moves a PTZ camera to the specified preset.

URI: `/Acs/Api/PtzFacade/GotoPresetId`

Request:

Field	Description
cameraId	The camera Id.
Id	String – The unique identifier of the camera.
presetId	Integer – The ID of the preset to go to.

Table: GotoPresetId request

Example request:

```
https://172.25.127.28:55756/Acs/Api/PtzFacade/GotoPresetId
{
        "cameraId" : {
                "Id" : "2448_54dd4102-96c7-4cdb-a8d2-85c2187a5d47"
        },
        "presetId" : 1
}
```

Response: The operation lacks response body. A PtzPriorityException may be thrown.

Example response: HTTP Response: 200 OK

The operation is available from API version 1.6 and later.

GetPtzPresets

Gets all defined presets for a PTZ camera.

URI: `/Acs/Api/PtzFacade/GetPtzPresets`

Request:

Field	Description
cameraId	The camera Id.
Id	String – The unique identifier of the camera.

Table: GetPtzPresets request

Example request:

```
https://172.25.127.28:55756/Acs/Api/PtzFacade/GetPtzPresets
{
        "cameraId" : {
                "Id" : "2448_54dd4102-96c7-4cdb-a8d2-85c2187a5d47"
        }
}
```

Response:

Field	Description
Id	Integer – The ID of the PTZ preset.
Name	String – The name of the PTZ preset.
Token	String – The token of the PTZ preset.

Table: GetPtzPresets response

Example response: HTTP Response: 200 OK

```
[
    {
        "Id": 1,
        "Name": "Home",
        "Token": "Home"
    },
    {
        "Id": 2,
        "Name": "Entrance",
        "Token": "Entrance"
    },
    {
        "Id": 3,
        "Name": "Parking",
        "Token": "Parking"
    }
]
```

The operation is available from API version 1.6 and later.

11.13. TriggerFacade

Handles activation and deactivation of triggers configured in the AXIS Camera Station Action rules. Requires rule configuration via AXIS Camera Station client UI.

ActivateTrigger

Triggers the associated rule to perform configured action(s).

URI: `/Acs/Api/TriggerFacade/ActivateTrigger`

Request:

Field	Description
triggerName	String – the name of the trigger to activate. This trigger is configured in the Action rule in AXIS Camera Station client.

Table: ActivateTrigger request

Example request:

```
https://172.25.127.28:55756/Acs/Api/TriggerFacade/ActivateTrigger
{
        "triggerName" : "ConfiguredTriggerName"
}
```

Response: The operation lacks response body. An exception may be thrown.

Example response: HTTP Response: 200 OK

The operation is available from API version 1.1 and later.

DeactivateTrigger

Deactivates the trigger to stop action(s) configured in the associated common rule.

URI: `/Acs/Api/TriggerFacade/DeactivateTrigger`

Request:

Field	Description
triggerName	String – the name of the trigger to deactivate. This trigger is configured in the Action rule in AXIS Camera Station client

Table: DeactivateTrigger request

Example request:

```
https://172.25.127.28:55756/Acs/Api/TriggerFacade/DeactivateTrigger
{
        "triggerName" : "ConfiguredTriggerName"
}
```

Response: The operation lacks response body. An exception may be thrown.

Example response: HTTP Response: 200 OK

The operation is available from API version 1.1 and later.

ActivateDeactivateTrigger

Triggers the associated rule to perform configured action(s) for the specified time.

URI: `/Acs/Api/TriggerFacade/ActivateDeactivateTrigger`

Request:

Field	Description
triggerName	String – the name of the trigger to activate. This trigger is configured in the Action rule in AXIS Camera Station client.
deactivateAfterSeconds	String – integer value > -1. Number of seconds to keep trigger active.

Table: ActivateDeactivateTrigger request

Example request:

```
https://172.25.127.28:55756/Acs/Api/TriggerFacade/ActivateDeactivateTrigger
{
        "triggerName" : "ConfiguredTriggerName",
        "deactivateAfterSeconds" : "30"
}
```

Response: The operation lacks response body. An exception may be thrown.

Example response: HTTP Response: 200 OK

The operation is available from API version 1.1 and later.

PulseTrigger

Triggers the associated rule to perform configured action(s) and then stop them immediately, i.e., trigger activation followed by an immediate deactivation.

URI: `/Acs/Api/TriggerFacade/PulseTrigger`

Request:

Field	Description
triggerName	String – the name of the trigger to activate. This trigger is configured in the Action rule in AXIS Camera Station client

Table: PulseTrigger request

Example request:

```
https://172.25.127.28:55756/Acs/Api/TriggerFacade/PulseTrigger
{
        "triggerName" : "ConfiguredTriggerName"
}
```

Response: The operation lacks response body. An exception may be thrown.

Example response: HTTP Response: 200 OK

The operation is available from API version 1.1 and later.

11.14. Send HTTP Notification action

This is not a façade within ACS API, but as the action can be used to get event information sent from AXIS Camera Station to third party system, information on the trigger data sent is included in this documentation.

Can be used similar to the SMTP option (Send e-mail action) to retrieve event information from AXIS Camera Station. Manages notification configured in the AXIS Camera Station Action rules. To receive event information to a web server requires rule configuration via AXIS Camera Station client UI.

Default HTTP method is GET, without a request body. To provide data in some form to the target host instead use POST or PUT, both enabling the request body field allowing specifying selected data for upload.

For guidance on how to set this up in AXIS Camera Station refer to the Online User Manual.

Content type

You can select one of the default content types: Plain text, JSON, XML, or SOAP.
If needed, you can specify a different content type by typing free text.

Trigger data

You can choose what trigger data parameters,to include within the request body, and where. It is inserted on the parameter form: **`$(TriggerData.<DATA>)`**. The following data types are supported:

Data	Parameter	Description
Type	$(TriggerData.Type)	The type of trigger that activated this action rule. E.g. motion detection, device event, etc.
Source ID	$(TriggerData.SourceId)	The ID of the source origin for what triggered the action rule, if applicable. E.g. a camera ID, a device ID, an action button ID, etc.
Source Name	$(TriggerData.SourceName)	The name of the source that triggered the action rule, if applicable. E.g. a camera name, device name, action button name, etc.
Time (UTC)	$(TriggerData.TimeUtc)	The UTC date and time when the action rule was triggered.
Time (local)	$(TriggerData.TimeLocal)	The date and time of the server when the action rule was triggered.

Authentication methods

Digest is the default and recommended authentication method. Basic can be used but be aware that credentials are sent base 64 encoded in the HTTP header.

Authentication flow:

1. When receiving a response without authentication information you must respond with 401.
2. Add WWW header with information on authentication, such as **Authorization: Digest username=, realm=, nonce=, etc.**

Expected WWW header format:

```
WWW-Authenticate: Basic realm="User Visible Realm", charset="UTF-8"
```

WWW-Authenticate: Digest (or **Basic**) is sufficient, but the web server can specify 'realm' and 'charset'. The server must respond with information on how it expects the response to be composed.

Response format:

```
Authorization: Basic dGVzdDoxMjPCow==
```

The response text is in requested charset if specified.

The operation is available from API version 2.30 and later.

11.15. ActionButtonFacade

Handles camera action button functionality. *Limitation*: These endpoints only work on camera action buttons. Other action buttons, such as buttons in map views, are not supported.

GetActionButtons

Gets the action buttons for a list of cameras.

URI: `/Acs/Api/ActionButtonFacade/GetActionButtons`

Request:

Field	Description
cameraIds	List of the IDs of the cameras to get action buttons for.
Id	String – the unique identifier for the camera.

Table: GetActionButtons request

Example request:

```
https://172.25.127.28:55756/Acs/Api/ActionButtonFacade/GetActionButtons
{
        "cameraIds": [
                {
                        "Id": "2143_0566a4da-7ff4-4a72-bc4c-5fe609a46030"
                },
                {
                        "Id": "2288_7490bc51-7990-460d-b14e-4e5bcd668b6e"
                },
                {
                        "Id": "4625_7490bc51-7990-460d-b14e-4e5bcd668b6e"
                }
        ]
}
```

Response:

Field	Description

ActionButtons	A dictionary with camera Ids as keys and list of action buttons as values.
Id	The Id of the action button.
IsToggleButton	Boolean – If the button is a toggle button
Label	String – The label of the button, should be used as a friendly name not an identifier.
Tooltip	String – The tooltip of the button.
UntoggleLabel	String – The label of the button to use when the button is toggled.
ButtonState	Integer – The state of the button. *See table ButtonStates*.

Table: GetActionButtons response

ButtonState	ButtonState corresponding value
0	Toggled
1	Untoggled
2	Disabled

Table: ButtonStates

Example response:

HTTP Response: 200 OK

```
{
    "ActionButtons": {
        "2143_0566a4da-7ff4-4a72-bc4c-5fe609a46030": [],
        "2288_7490bc51-7990-460d-b14e-4e5bcd668b6e": [
            {
                "Id": {
                    "Id": "4719_7490bc51-7990-460d-b14e-4e5bcd668b6e"
                },
                "IsToggleButton": true,
                "Label": "Turn on",
                "UntoggleLabel": "Turn off",
                "ButtonState": 1
            }
        ],
        "4625_7490bc51-7990-460d-b14e-4e5bcd668b6e": [
            {
                "Id": {
                    "Id": "4702_7490bc51-7990-460d-b14e-4e5bcd668b6e"
                },
                "IsToggleButton": false,
                "Label": "Open Door",
                "ButtonState": 0
            }
        ]
    }
}
```

The operation is available from API version 1.6 and later.

SetActionButton

Sets the status of an action button. *Note:* It is not possible to set ButtonState disabled.

URI: `/Acs/Api/ActionButtonFacade/SetActionButton`

Request:

Field	Description
buttonId	The button Id.
Id	Integer – The unique identifier of the button.
buttonState	Integer – The state to set for the button. *See table ButtonStates*.

Table: SetActionButton request

ButtonState	ButtonState corresponding value
0	Toggled
1	Untoggled

Table: ButtonStates

Example request:

```
https://172.25.127.28:55756/Acs/Api/ActionButtonFacade/SetActionButton
{
        "buttonId":
                {
                        "Id": "3615_0566a4da-7ff4-4a72-bc4c-5fe609a46030"
                },
        "buttonState": 0

}
```

Response:

Field	Description
IsButtonSet	Boolean – If the new button state was set.

Table: SetActionButton response

Example response.

HTTP Response: 200 OK

```
{
    "IsButtonSet": true
}
```

The operation is available from API version 1.6 and later.

11.16. EventLogFacade

Exposes methods to retrieve event logs from the AXIS Camera Station server. It consists of the event timestamp, the event log type, and data that contains relevant information.

GetEventLogList

This method is used to retrieve a list of event logs from the AXIS Camera Station server from a specific date and time until the current time.

URI: `/Acs/Api/EventLogFacade/GetEventLogList`

Request:

Field	Description
Range	The range of elements to be collected from the server.
StartIndex	Integer - The starting index of the oldest event log in the reply.

NumberOfElements	Integer - The maximum number of event logs to be in the reply.
Time	String - The requested time in UTC and in “YYYY-MM-DD hh:mm:ss.fffffff”. The second fraction digits can be omitted.

Table: *GetEventLogList* request body parameters.

Example request (*Gets 10000 event logs from 2017-03-28 until now*):

```
https://172.25.127.28:55756/Acs/Api/EventLogFacade/GetEventLogList
{
    "range" : {
        "StartIndex": "0",
        "NumberOfElements":"10000"
    },
    "time" : "2017-03-28 22:10:15"
}
```

Response

Field	Description
Events	The list of event logs.
Timestamp	The date and time of the event log in Coordinated Universal Time (UTC).
EventLogType	String - specifies the type of the event log (i.e recording started, camera status down, disk error etc).
Data	Additional information regarding the event. *See table EventLogDataField.*
Name	The camera name to which the Event occurred.
DiskPath	The path of the disk to which the event occurred.
RecordingExportDirectory	The path of the directory to which the recording export is attempted.
RecordingDirectoryPath	The path of the directory to which a recording is saved.
ContainsLastResult	Specifies whether there are more event logs to be retrieved.

Table: *GetEventLogList* response.

Example response (*not all possible results are included in the example)*:

HTTP Response: 200 OK

```
{
    "Events": [
        {
            "Timestamp": "2022-04-27 14:11:37.0803000",
            "EventLogType": "RecordingFailure",
            "Data": {
                "Name": "AXIS 223M",
                "CameraId": "89_070011fc-e4fb-4290-81fb-4e1140cae2f3"
            }
        },
        {
            "Timestamp": "2022-04-27 14:10:44.7977000",
            "EventLogType": "RecordingStarted",
            "Data": {
                "Name": "AXIS A8207-VE - Custom View",
                "CameraId": "156_070011fc-e4fb-4290-81fb-4e1140cae2f3"
            }
        }
    ]
```

```
}
```

EventLogType cases explained

The event log type returned by the AXIS Camera Station server is in string format (as seen in the "*EventLogType*" field in the example response). The following table explains which event cases are included on each EventLogType.

EventLogType	Happens When
RecordingStarted	• A camera has started recording.
RecordingStopped	• A camera has stopped recording.
DiskError	• Disk access denied. • Disk not found. • Recording export disk space shortage. • Recording directory inaccessible. • Recording export failed.
DiskWarning	• Intruding disk data. • Disk full.
RecordingFailCameraAccess	• Camera is inaccessible for recording.
RecordingFailure	• Failed recording on camera.
CameraStatusDown	• Camera status is down.
CameraStatusUp	• Camera status is up.

Table: *EventLogType*.

EventLogType and data fields

Each event contains event log type specific information in a data structure called *Data*. The following table displays the various information that can be included inside the *Data* object alongside a description of its context.

EventLogType	Data	Description
RecordingStarted	Name	The name of the camera.
	CameraId	The ID of the camera. (2.21)
RecordingStopped	Name	The name of the camera.
	CameraId	The ID of the camera. (2.21)
DiskError	RecordingExportDirectory	The recording export directory path.
	RecordingDirectoryPath	The recording directory path.
	DiskPath	The path of the disk.
DiskWarning	DiskPath	The path of the disk.
RecordingFailCameraAccess	Name	The name of the camera.
	CameraId	The ID of the camera. (2.21)
RecordingFailure	Name	The name of the camera.
	CameraId	The ID of the camera. (2.21)
CameraStatusUp	Name	The name of the camera.
	CameraId	The ID of the camera. (2.21)
CameraStatusDown	Name	The name of the camera.
	CameraId	The ID of the camera. (2.21)

Table: *EventLogType Data Field*.

Available from API version 1.7 and later. The CameraId field within the Data structure was introduced in API version 2.21.

11.17. BookmarkFacade

Bookmarks in this context is an event indicator that is visible in the timeline in the AXIS Camera Station client. *Note:* Bookmarks are removed by Axis Camera Station following the same pattern as recordings, adhering to the retention time set for the camera. The bookmark identifier, or bookmark ID, is returned when adding or retrieved when listing bookmarks using GetBookmarks.

AddBookmark

Adds a bookmark for the camera at a specified time.
The ID of the newly created bookmark is returned as response and can be used in other endpoints.

URI: `Acs/Api/BookmarkFacade/AddBookmark`

Supported Http Methods

- POST

Request:

Field	Description
CameraId	The camera identifier.
Id	String – The unique camera identifier.
Time	String – Optional. Time for the bookmark in UTC in format “YYYY-MM-DD hh:mm:ss.ffffff”. The second fraction digits can be omitted. If not specified, the bookmark time will be set to the server time it was added.
Name	String - Name of the bookmark. Maximum 255 characters.
Description	String - Optional. Description of the bookmark. Maximum 255 characters.

Example request: `https://localhost:55756/Acs/Api/BookmarkFacade/AddBookmark`

```
{
    "CameraId": {
        "Id": "2542_54dd4102-96c7-4cdb-a8d2-85c2187a5d47"
    },
    Time: "2019-08-01 19:20:19",
    Name: "Special event",
    Description: "This was a special event"
}
```

Response:

Field	Description
Id	String – Bookmark identifier. The ID of the newly created bookmark.

Example response: HTTP Response: 200 OK

```
{
      "Id": "144_e9f9497f-31a0-4e58-9d13-bb44ed78f294"
}
```

GetBookmark

Retrieves a bookmark specified by its ID.

URI: `Acs/Api/BookmarkFacade/GetBookmark`

Supported Http Methods

- POST

Request

Field	Description
BookmarkId	String – Bookmark identifier.
Id	String – Unique bookmark identifier.

Example request: `https://localhost:55756/Acs/Api/BookmarkFacade/GetBookmark`

```
{
      "BookmarkId": {
          "Id": "77016_018b7214-cd17-4eab-bc0a-83ed4753feee"
       }
}
```

Response:

Field	Description
CameraId	String – The camera identifier.
Id	String – The camera identifier.
BookmarkId	String – Bookmark identifier.
Id	String – Unique bookmark identifier.
Time	String –Time for the bookmark, in UTC, in format “YYYY-MM-DD hh:mm:ss.ffffff”.
Name	String - Name of the bookmark.
Description	String – If set, description of the bookmark.

Example response: HTTP Response: 200 OK

```
{
    "CameraId": {
        "Id": "2250_e9f9497f-31a0-4e58-9d13-bb44ed78f294"
    },
    "BookmarkId": {
       "Id": "77016_018b7214-cd17-4eab-bc0a-83ed4753feee"
    },
    "Time": "2018-10-25 12:15:49.3228688",
    "Name": "Special event",
    "Description": "A very special event"

}
```

UpdateBookmark

Updates name and description of a bookmark specified by its ID.
Note that the *time* and *camera* for the bookmark cannot be changed. In such cases a new bookmark should be added instead and the old one removed. If either name or description is omitted, their old values are retained.

URI: `Acs/Api/BookmarkFacade/UpdateBookmark`

Supported Http Methods
- POST

Request:

Field	Description
BookmarkId	Bookmark identifier for the bookmark to be updated.
Id	String – The unique bookmark identifier.
Name	String - Optional. Name of the bookmark. Maximum 255 characters.
Description	String - Optional. Description of the bookmark. Maximum 255 characters.

Example request:
```
https://localhost:55756/Acs/Api/BookmarkFacade/UpdateBookmark
{
        "BookmarkId": {
                "Id": "144_e9f9497f-31a0-4e58-9d13-bb44ed78f294"
        },
        "Name": "Changed event",
        "Description": "Updated information about this event"
}
```

Response: The operation lacks response body.

Example response: HTTP Response: 200 OK

RemoveBookmark

Removes a bookmark specified by its ID.

URI: `Acs/Api/BookmarkFacade/RemoveBookmark`

Supported Http Methods
- POST

Request:

Field	Description
BookmarkId	Bookmark identifier for the bookmark to be removed.
Id	String – The unique bookmark identifier.

Example request: `https://localhost:55756/Acs/Api/BookmarkFacade/RemoveBookmark`

```
{
    "BookmarkId":
    {
        "Id": "17_5d5eabf6-49f7-449d-955a-839f8adb894b"
    }
}
```

Response: The operation lacks response body.

Example response: HTTP Response: 200 OK

GetBookmarks

Lists all bookmarks for a single camera or for multiple cameras within a certain time interval.

URI: `Acs/Api/BookmarkFacade/GetBookmarks`

Supported Http Methods

- POST

Request:

Field	Description
CameraIds	List of the IDs of the cameras to get bookmarks for.
Id	String - The unique identifier of each camera.
Interval	The time interval to get bookmarks from.
StartTime	The start time, in UTC, in “YYYY-MM-DD hh:mm:ss” format.
StopTime	The stop time, in UTC, in “YYYY-MM-DD hh:mm:ss” format.
Range	The range of element to be collected from the server
StartIndex	Integer - The starting index of the first bookmark in the response.
NumberOfElements	Integer - Number of bookmarks to include in the response.

Example request:

`https://localhost:55756/Acs/Api/BookmarkFacade/GetBookmarks`

```
{
    "CameraIds": [
    {
        "Id": "2250_e9f9497f-31a0-4e58-9d13-bb44ed78f294"
    }
    ],
    "Interval": {
        "StartTime": "2018-10-25 12:00:00",
        "StopTime": "2018-10-25 13:00:00"
    },
    "Range": {
        "StartIndex": 0,
        "NumberOfElements": 5
    }
 }
```

Response:

Field	Description
Bookmarks	The list of bookmarks.
CameraId	String – The camera identifier.
Id	String – The unique camera ID.
BookmarkId	String – Bookmark identifier.
Id	String – The unique bookmark ID.
Time	String –Time for the bookmark, in UTC, in format “YYYY-MM-DD hh:mm:ss.fffffff”.
Name	String – Name of the bookmark.
Description	String – If set, description of the bookmark.
ContainsLastResult	Specifies whether there are more bookmarks available for fetching within this interval.

Table: *GetBookmarks* response.

Example response: HTTP Response: 200 OK


```
{
 "Bookmarks": [
    {
      "CameraId": {
        "Id": "2250_e9f9497f-31a0-4e58-9d13-bb44ed78f294"
      },
      "BookmarkId": {
        "Id": "Id": "144_e9f9497f-31a0-4e58-9d13-bb44ed78f294",
      },
      "Time": "2018-10-25 12:15:49.3228688",
      "Name": "Special event",
      "Description": "A very special event"
    },
    {
      "CameraId": {
        "Id": "2250_e9f9497f-31a0-4e58-9d13-bb44ed78f294"
      },
      "BookmarkId": {
        "Id": ""123_e9f9497f-31a0-4e58-9d13-bb44ed78f294",
      },
      "Time": "2018-10-25 12:43:29.3128688",
      "Name": "Person detected",
      "Description": ""
    }
 ],
 "ContainsLastResult": true
}
```


11.18. OnScreenControlsFacade

Handles On-Screen Controls functionality. An extensive explanation of some properties may be required to get the most out of these functions. Contact Axis Communications for more details.

GetOnScreenControls

Gets the On-Screen Controls for a list of cameras.

URI: `/Acs/Api/OnScreenControlsFacade/GetOnScreenControls`

Request:

Field	Description
cameraIds	List of the IDs of the cameras to get On-Screen Controls for.
Id	String – The unique camera identifier.
Locale	String – Optional. Preferred language for the description. Follow RFC4646 or ISO639-1. Availability of specified language is not guaranteed. Default is “en-us”.

Table: GetOnScreenControlsRequest request

Example request:

```
https://172.25.127.28:55756/Acs/Api/OnScreenControlsFacade/GetOnScreenControls
{
	"cameraIds" : [
		{
			"Id" : "3609_54dd4102-96c7-4cdb-a8d2-85c2187a5d47"
		},
		{
			"Id" : "2438_54dd4102-96c7-4cdb-a8d2-85c2187a5d47"
		}
	],
	"locale" : "en-us"
}
```

Response: There is no guarantee that all listed properties are included in the response.

Field	Description
OnScreenControls	The list of On-Screen Controls.
CameraId	The camera id.
Id	String – The unique identifier of the camera.
Controls	A group of features for the device. *See table: FeatureGroup.*

Table: GetOnScreenControls response

Field	Description
Feature Group properties	Feature Group Properties explanation
Name	String – The name of the feature group in English used as an identifier.
NiceName	String – The name of the feature group to be displayed for the user according to the requested locale.
Features	The list of features in this feature group. See table: Feature.
FeatureGroups	The list of sub-feature groups for this feature group. Used for grouping features together, like PTZ features or similar.

Table: FeatureGroup

Field	Description
Name	String – The name of the feature in English used as an identifier.
NiceName	String – The name of the feature group to be displayed for the user according to the requested locale.
Users	List of integers – Describes the users allowed to trigger this On-Screen Control. *See table Users*.
Info	String – Information about the feature according to the requested locale.
RequestType	String – The type of request to be done to the camera by AXIS Camera Station. Valid values are GET and POST.

RequestPayload	String – The request Payload of the request AXIS Camera Station will send to the camera when triggering the feature. Valid values are JSON. Reserved for future use is BINARY, SOAP and XML,
RequestUrl	String – The request URL AXIS Camera Station will use when triggering this On-Screen Control. {<param>} should be replaced by parameters values. *See Table ParametersInRequestUrlStructure.*
Response	Boolean – Indicates whether this feature returns a response from the camera. *See table RequestUrlResponseStructure*.
vmsHint	String – Client-side actions that are suitable to do when the On-Screen Controls is triggered. *See table vmsHint*.
vmsReq	String – The VMS requirements to fulfill for the On-Screen Control to be useful for an operator. *See table vmsReq*.
jsonTemplate	Used as a template for the JSON payload of a POST request, this field is only present when RequestPayload is JSON. The String that shows how a JSON Request should be formatted. The template string could for example look like this: "{1,2,3,4:[4,5],6}" or "{1}", "{1,2,3,4:{5}}". 1,2,3,4.. is the name of the parameters in the parameters JSON Object.
parameters	Description of the parameters to be used in the requestUrl. *See table ParametersInRequestUrlStructure*.

Table: Feature

User	User corresponding value
0	Admin
1	Operator
2	Viewer

Table: Users

Field	Description
Type	Describes the type of the parameter. *See table Types*.
NiceName	String – Short description of the parameter
Info	String - Optional information about the parameter
Key	String – String to use as a key for parameters in a JSON Payload.
jsonVariableFormat	Format to use for the JSON parameter. *See table JsonVariableFormat*.

Table: ParametersInRequestUrlStructure

Type	Type corresponding value
0	JsonContext. *See table JsonOnlySpecificParameters*.
1	JsonFixed. *See table JsonOnlySpecificParameters*.
2	JsonKeyOnly. *See table JsonOnlySpecificParameters*.
3	MultiOptions. *See table MultiOptionSpecificParameters*.
4	Range. *See table RangeSpecificParameters*.
5	Shape. *See table ShapeTypeSpecificParameters*.
6	String. *See table StringSpecificParameters*.

Table: Types

Range specific fields	Description
SelectedValue	Double – The selected value.
Unit	String – Type of range value. Valid values are integer or fraction.
Min	Double - Minimum value for range.
Max	Double – Maximum value for range.
Step	Double – Step size for range.

Table: RangeSpecificParameters

ShapeType specific fields	Description
ShapeTypeDto	String - The shape of a shape parameter. Valid values are Coordinate, Line, Box, and Polygon.
Coordinates	String – The coordinates of the shape.

Table: ShapeTypeSpecificParameters

StringSpecificParameters specific fields	Description
SelectedValue	String – The selected value.

Table: StringSpecificParameters

MultiOptionSpecificParameters specific fields	Description
SelectedValue	String – The selected value.
ValueNiceNames	Dictionary with the Values and their corresponding NiceNames.

Table: MultiOptionSpecificParameters

JsonOnlySpecificParameters specific fields	Description
fixed	The string in "value" should be used as the value in the key:value JSON pair.
context	The value in the value field in the key:value pair will be returned by the camera.
keyOnly	Used when only a key is needed. Such as header for JSON objects and lists.

Table: JsonOnlySpecificParameters

JsonVariableFormat valid values	Description
string	The key:value pair, formatted like this "key":"variable string"
integer	The key:value pair, formatted like this "key":5 or "key":-5
fraction	The key:value pair, formatted like this "key":5.0 or "key":-5.0
exponent	The key:value pair, formatted like this "key":5.0e+2 or "key":-5.0e-3
boolean	The key:value pair, formatted like this "key":true or "key":false
null	The key:value pair, formatted like this "key":null

Table: JsonVariableFormat

vmsHint	vmsHint explanation
windowClose	If the camera is in a popup window, close the window.
audioOutEnable	Mute the operator microphone, usually used as a mute button for cameras with audio communication devices
audioOutToggle	Open the operator microphone to the camera, usually used as an unmute button for cameras with audio communication devices.
AudioOutDisable	Toggle the operator microphone between mute and unmute state
audioInDisable	Mute the camera microphone, usually used as a mute button for cameras with audio communication devices.
audioInEnable	Start listening on the microphone for the camera.
audioInToggle	Toggle the camera microphone between mute and unmute state
hidden	Features of lower importance which can be hidden from the operator.

Table: vmsHint

vmsReq	vmsReq explanation
eventStream	This On-Screen Control produces an event in the Dynamic Event Stream. The VMS must understand these events to benefit from this button.

Table: vmsReq

ParametersInRequestUrlStructure	ParametersInRequestUrlStructure explanation
vmsHint	Integer – How to treat the return value. *See table vmsHintInResponse.*
context	String – Variable containing the context value, from the request.
type	String – Type of response, integer, string, boolean, array, object.
value	Defined by type – Return value of the request.

Table: ParametersInRequestUrlStructure

vmsHintInResponse valid values	vmsHintInResponse explanation — What the VMS should do with the return value
0	Display - Display the returned value to the user.
1	Store – Store the returned value as searchable metadata for an operator.
2	Unknown

Table: vmsHintInResponse

Example response: HTTP Response: 200 OK


```
{
  "OnScreenControls": [
    {
      "CameraId": {
        "Id": "8461_0e12a4e3-ef64-4db1-b2c1-8a9124adc66f"
      },
      "Controls": {
        "Name": "root",
        "Features": [],
        "FeatureGroups": [
          {
            "Name": "IR",
            "NiceName": "IR Illumination",
            "Features": [
              {
                "Users": [
                  1
                ],
                "Name": "activateLight",
                "NiceName": "Turn on IR Illumination",
```

```
                "Info": "Use the built-in IR Illumination in low-light environments",
                "RequestType": "POST",
                "RequestPayload": "JSON",
                "RequestUrl": "/axis-cgi/lightcontrol.cgi",
                "JsonTemplate": "{ 1, 2, 3, 4: {5}}",
                "Response": false,
                "Parameters": [
                  {
                    "Type": 1,
                    "ParameterName": "1",
                    "NiceName": "",
                    "DefaultValue": "",
                    "JsonKey": "apiVersion",
                    "JsonValue": "1.0",
                    "JsonVariableFormat": 0
                  },
                  {
                    "Type": 0,
                    "ParameterName": "2",
                    "NiceName": "",
                    "DefaultValue": "",
                    "JsonValue": "",
                    "JsonVariableFormat": 0
                  },
                  {
                    "Type": 1,
                    "ParameterName": "3",
                    "NiceName": "",
                    "DefaultValue": "",
                    "JsonKey": "method",
                    "JsonValue": "activateLight",
                    "JsonVariableFormat": 0
                  },
                  {
                    "Type": 2,
                    "ParameterName": "4",
                    "NiceName": "",
                    "DefaultValue": "",
                    "JsonKey": "params",
                    "JsonValue": "",
                    "JsonVariableFormat": 0
                  },
                  {
                    "Type": 1,
                    "ParameterName": "5",
                    "NiceName": "",
                    "DefaultValue": "",
                    "JsonKey": "lightID",
                    "JsonValue": "led0",
                    "JsonVariableFormat": 0
                  }
                ]
              }
            ],
            "FeatureGroups": []
          }
        ]
      }
    }
  ]
}
```


The operation is available from API version 2.1 and later.

TriggerOnScreenControl

Triggers an On-Screen Control.

URI: `/Acs/Api/OnScreenControlsFacade/TriggerOnScreenControl`

Request:

Field	Description
cameraId	Object – The camera Id.
Id	String – The unique identifier of the camera.
TriggeredFeature	The feature to trigger, as described in the tables from the GetOnScreenControls section above.
Users	List of integers – Describes the users allowed to trigger this On-Screen Control. *See table Users*. *(List can be empty)*
Name	String – The name of the feature in English, used as an identifier.
NiceName	String – Optional. The name of the feature group to be displayed for the user according to the requested locale.
Info	String – Optional. Information about the feature according to the requested locale.
RequestType	String – The type of request to be done to the camera by AXIS Camera Station. Valid values are GET and POST.
RequestUrl	String – The request URL AXIS Camera Station will use when triggering this On-Screen Control. {<param>} should be replaced by parameters values. *See Table ParametersInRequestUrlStructure*.
Response	Boolean – Indicates whether this feature returns a response from the camera. *See table RequestUrlResponseStructure*.
parameters	List of string – Description of the parameters to be used in the requestUrl. *See table ParametersInRequestUrlStructure*. (List can be empty).

Table: TriggerOnScreenControl request

Example request:

```
https://172.25.127.28:55756/Acs/Api/OnScreenControlsFacade/TriggerOnScreenCont
rol
{
        "cameraId" : {
                        "Id" : "3609_54dd4102-96c7-4cdb-a8d2-85c2187a5d47"
        },
        "triggeredFeature" :
        {
                "Users": [
                        1
                ],
                "Name": "SpeedDry",
                "NiceName": "Speed dry",
                "Info": "Vibrate the camera to remove water from dome",
                "RequestType": "GET",
                "RequestUrl": "/axis-cgi/com/ptz.cgi?auxiliary=speeddry",
                "Response": false,
                "Parameters": []
        }
}
```

Response:

Field	Description
IsResponse	Boolean – If the triggered feature has a response from the camera.
VmsHint	Integer – As described in the table vmsHintInResponse above.
Type	Integer – The type of the response value. *See table ResponseValueTypes.*
Value	String – The response value.

Table: TriggerOnScreenControl response

ResponseValueTypes valid values	ResponseValueTypes explanations
0	String
1	Integer
2	Boolean
3	Array
4	Object
5	Unknown

Table: ResponseValueTypes response

Example response.

HTTP Response: 200 OK

```
{
    "IsResponse": false,
    "VmsHint": 0,
    "Type": 0
}
```

The operation is available from API version 2.1 and later.

11.19. Deep Links for launching the AXIS Camera Station client

A 'Deep Link' in this API context is an URI (resource address) that links to a specific location within the AXIS Camera Station client application.
Using such a 'Deep Link' it is possible to launch and perform multiple actions within the client. The interaction can be performed internally within the client as well as from external sources such as browsers.
Note: Server IDs used in deep links must be known to the client to be triggered. Server IDs of AXIS Camera Station services not connected to the client will not have any effect.

The AXIS Camera Station deep links follow the URI syntax, using the custom scheme '**`axacs`**':

`axacs://axis.com?query`

The query component of the URL is what ties parameters to the performed action. Query components are always initiated with a '?'-sign and multiple parameters are separated by '&'-signs. Example: **`axacs://axis.com?queryParameterA&queryParameterB`**

To use deep links the AXIS Camera client must be installed. During setup, the custom **`axacs`** scheme is automatically registered with windows and associated with the client.
To use a deep link to navigate to a camera stream even if camera streams are hidden from the navigation bar check the 'Show in navigation' client setting.

System specific IDs such as Camera ID are retrieved with the appropriate API method.

Deep Link Actions

All deep link actions are governed by the mode parameter where mode identifies the action.

Basic

The most basic form of deep link is the pure link without any parameters. This launches the client but perform no further actions.

Link: `axacs://axis.com`

Live

Navigates to the AXIS Camera Station live view tab and opens a live stream for the selected camera.

Link: `axacs://axis.com?mode=live&cameraId=<CameraID>&newTab=<Boolean>`
Request:

Parameter	Value or Description	Value Type
mode	`live`	String
cameraId	The ID of the camera in AXIS Camera Station to receive the live stream from. *Note*: This requires the long camera ID format.	Integer_GUID

Table: Launch live view tab URL

Example URLs:

- `axacs://axis.com?mode=live&cameraId=13111_4ea7e955-3169-4ae1-a433-a2d87e55e93b`

Recording

Navigates to the AXIS Camera Station client recordings tab for the selected camera, optionally to a pre-selected point in the recording's timeline.

Link:
`axacs://axis.com?mode=recording&cameraId=<CameraID>&timestamp=<yyyy-MM-ddHHmmss-fffffffZ>&exportMarkerStart=<yyyy-MM-ddHHmmss-fffffffZ>&exportMarkerStop=<yyyy-MM-ddHHmmss-fffffffZ>`

Request:

Parameter	Value or Description	Value Type
mode	`recording` or `playback (alias)`	String
cameraId	The ID of the camera in AXIS Camera Station to receive the live stream from. *Note*: This requires the long camera ID format.	Integer_GUID
timestamp	Date and time for position in the recordings timeline. Expected to be in the UTC+0 time zone (Z)	DateTime – Optional Format:"yyyy-MM-dd-HHmmss-ffffffZ"
exportMarkerStart and **exportMarkerStop**	Date and time to place export markers designating a timeline area for export. Expected to be in the UTC+0 time zone (Z)	DateTime – Optional Format:"yyyy-MM-dd-HHmmss-ffffffZ"

Table: Launch recordings tab URL

Example URLs:

- Camera recordings
  `axacs://axis.com?mode=recording&cameraId=12252&serverId=a2595185-8057-44b4-88e0-d228aaa5ea81`
- Camera recordings with timestamp
  `axacs://axis.com?mode=recording&cameraId=12252&serverId=a2595185-8057-44b4-88e0-d228aaa5ea81&timestamp=2021-08-05-134253-9529723Z`
- Camera recordings with export markers
  `axacs://axis.com?mode=recording&cameraId=12951&serverId=3c1b4421-dc7b-4b11-b201-3215256353e9&exportMarkerStart=2021-10-13-090255-8665680Z&exportMarkerStop=2021-10-13-090315-8665680Z`

Export

Adds all recordings in a time span to the export tab for a selected camera. Silent mode, i.e., adding selected recordings without bringing the export tab into focus is supported.
An export mode link is available for 24 hours.

Link: `axacs://axis.com?mode=export&cameraId=<CameraID>&exportStart=<yyyy-MM-ddHHmmss-fffffffZ>&exportStop=<yyyy-MM-ddHHmmss-fffffffZ>&exportNotes=<URL encoded UTF-8 text>&silentExport=true>`

Request:

Parameter	Value or Description	Value Type
mode	`export`	String
cameraId	The ID of the camera to receive the live stream from. *Note*: This requires the long camera ID format.	Integer_GUID
exportStart and **exportStop**	Optional. Date and time to place export markers designating a timeline area for export. Expected to be in the UTC+0 time zone (Z)	DateTime on format:"yyyy-MM-dd-HHmmss-fffffffZ"
exportNotes	Notes that can be attached to all recordings added to the export. *Note*: This text must be URL-encoded using the UTF-8 character set and will be cropped if more than 1300 characters long.	String
silentExport	Recordings exported silently will be added to the export tab without bringing the tab into focus, allowing more recordings to be added without interrupting the user workflow. Default value: false.	Boolean

Table: Launch export tab URL

Example URLs:

- With export notes
  `axacs://axis.com?mode=export&cameraId=12951&serverId=3c1b4421-dc7b-4b11-b201-3215256353e9&exportStart=2021-10-13-090255-8665680Z&exportStop=2021-10-13-090315-8665680Z&exportNotes=Export%20Notes.`
- Using silent export
  `axacs://axis.com?mode=export&cameraId=12951&serverId=3c1b4421-dc7b-4b11-b201-3215256353e9&exportStart=2021-10-13-090255-8665680Z&exportStop=2021-10-13-090315-8665680Z&silentExport=true`

Configuration

Navigates to the configuration tab at optional specific configuration page.
If AXIS Camera Station has no added devices and a deep link with `mode=configuration` is triggered during start-up or login, `DeviceDiscovery` ("Add devices") will be selected.
The scroll bar will not be adjusted for configuration links.

Link: `axacs://axis.com?mode=configuration&page=<ID String>`

Request:

Parameter	Value or Description	Value Type
mode	`configuration`	String
page	ID string of the page to show in the configuration tab. *See table Configuration page ID string*.	String

Table: Launch configuration tab URL

Configuration Category	Configuration Page	ID String
Devices	Add devices	`DeviceDiscovery`
	Cameras	`Cameras`
	Other devices	`Devices`
	Streaming profiles	`StreamingProfiles`
	Image configuration	`ImageConfiguration`
	PTZ presets	`PtzPresets`
	Management	`DeviceManagement`
	External data sources	`ExternalDataSources`
Storage	Management	`StorageManagement`
	Selection	`StorageSelection`
Recording and events	Schedules	`Schedules`
	Recording method	`RecordingMethod`
	I/O ports	`IOSettings`
	Action rules	`ActionRules`
Client	Settings	`ClientSettings`
	Streaming	`ClientStreamingSettings`
Connected services	Firmware upgrade settings	`UpgradeFirmware`
	Axis Secure Remote Access	`SecureRemoteAccess`
	AXIS Camera Station Update	`CameraStationUpdate`
Server	Scheduled export	`ScheduledExport`
	Incident report	`IncidentReport`
	Settings	`ServerSettings`
Switch	Management	`Switch`
Licenses	Management	`LicenseManagement`
	Device status	`DeviceLicenseStatus`
	Keys	`LicenseKeys`
Security	User permissions	`UserPermission`
	Certificates	`Certificates`

Table: Configuration page ID strings

Example URL:
`axacs://axis.com?mode=configuration&page=DeviceDiscovery`

Limitations

An AXIS Camera Station client running in Administrator mode requires a link sent from a client also running in Administrator mode. Multi-screen and AXIS Camera Station client windows focus loss might cause an external deep link to fail. Live view action is not available for a disabled camera. Recording tabs cannot be reused by timestamped links once the client is restarted.

11.20. ClientCommandsFacade

NOTE: It is strongly recommended to instead use the Deep Links feature described above.

Handles Client Commands making it possible to remote control the AXIS Camera Station Windows client. It is possible to select cameras, start/stop playback, change playback speed and move the playback marker in the timeline.
The client is identified by the name of the computer the client is running on. This can be found, both in the basic system information view in Windows or in the Client Configuration Sheet in AXIS Camera Station.
The commands are executed with best effort. If no connected client is connected to the server or if the client is in a state where it will not accept commands, the command is discarded.

GoToCameras

Opens a new tab in the client with a dynamic split with the provided cameras. Consecutive calls to GoToCameras navigates in the same tab. If the tab is closed a new tab will be opened on the next call.

URI: `/Acs/Api/ClientCommandsFacade/GoToCameras`

Request:

Field	Description
machineName	The name of the client
cameraIds	List of the IDs of the cameras to include in the opened view.

Table: GoToCameras request

Example request: `https://172.25.127.28:55756/Acs/Api/ClientCommandsFacade/GoToCameras`

```
{
	"machineName" : "ACSClientPC",
	"cameraIds" : [
		{
			"Id" : "3609_54dd4102-96c7-4cdb-a8d2-85c2187a5d47"
		}
	]
}
```

Response: The operation lacks response body.

Example response: HTTP Response: 200 OK

StartPlayback

Starts playback in the tab opened by calling GoToCameras.

URI: `/Acs/Api/ClientCommandsFacade/StartPlayback`

Request:

Field	Description
serverAddress	The IP address or host name of the AXIS Camera Station server.
machineName	The name of the client

Table: StartPlayback request

Example request: `https://172.25.127.28:55756/Acs/Api/ClientCommandsFacade/StartPlayback`

```
{
       "machineName" : "ACSClientPC"
}
```

Response: The operation lacks response body.

Example response: HTTP Response: 200 OK

PausePlayback

Pauses playback in the tab opened by calling GoToCameras.

URI: `/Acs/Api/ClientCommandsFacade/PausePlayback`

Request:

Field	Description
serverAddress	The IP address or host name of the ACS server.
machineName	The name of the client

Table: PausePlayback request

Example request:

`https://172.25.127.28:55756/Acs/Api/ClientCommandsFacade/PausePlayback`

```
{
        "machineName" : "ACSClientPC"
}
```

Response: The operation lacks response body.

Example response: HTTP Response: 200 OK

GoToLiveView

Switches in the tab opened by calling GoToCameras to live view mode.

URI: `/Acs/Api/ClientCommandsFacade/GoToLiveView`

Request:

Field	Description
serverAddress	The IP address or host name of the AXIS Camera Station server.
machineName	The name of the client

Table: GoToLiveView request

Example request:

`https://172.25.127.28:55756/Acs/Api/ClientCommandsFacade/GoToLiveView`

```
{
```

```
        "machineName" : "ACSClientPC"
}
```

Response: The operation lacks response body.

Example response: HTTP Response: 200 OK

SetPlaybackSpeed

Sets the playback speed on the client. The speed can be a value between]0.0-100.0]

URI: `/Acs/Api/ClientCommandsFacade/SetPlaybackSpeed`

Request:

Field	Description
serverAddress	The IP address or host name of the AXIS Camera Station server.
machineName	The name of the client
speed	The speed factor

Table: SetPlaybackSpeed request

Example request:

```
https://172.25.127.28:55756/Acs/Api/ClientCommandsFacade/SetPlaybackSpeed
{
        "machineName" : "ACSClientPC",
        "speed" : "4.0"
}
```

Response: The operation lacks response body.

Example response: HTTP Response: 200 OK

SetPlaybackPositionUtc

Sets the playback position on the client. The position must be in UTC time.

URI: `/Acs/Api/ClientCommandsFacade/SetPlaybackPositionUtc`

Request:

Field	Description
serverAddress	The IP address or host name of the AXIS Camera Station server.
machineName	The name of the client
position	The position in UTC time

Table: SetPlaybackPositionUtc request

Example request:

```
https://172.25.127.8:55756/Acs/Api/ClientCommandsFacade/SetPlaybackPositionUtc
{
        "machineName" : "ACSClientPC",
        "position" : "2019-11-22T12:06:00Z"
}
```

Response: The operation lacks response body.

Example response: HTTP Response: 200 OK