Chapter 1 - Appendix 2 – ATEAS API

1.1. ATEAS API for camera servers (ATEAS Server)

1.1.1. Communication principle

Camera servers can receive events via the TCP/IP protocol on a specified event port – 8505 (see the Network configuration appendix). This data is in text form and you can set the text encoding for each communication channel. ASCII encoding is used by default. Other encoding types available include UNICODE, UTF8, Windows 1250 and others.

The principle of network communication using the TCP/IP protocol: When forwarding an ATEAS API command, the party generating this command has to establish a TCP/IP connection to the camera server. You can forward several commands within the frame of one TCP connection as well create a new connection for every single command. In order to receive ATEAS API commands, the camera server enables up to 50 simultaneous connections to the input port. Therefore, if the commands are generated from more than 50 sources (for example from all cameras connected to the camera server – up to 999), inactive connections will have to be closed.

General syntax: All ATEAS API commands are enclosed in square brackets. The data outside these brackets is not evaluated. This protocol does not distinguish between upper and lower case.

1.1.2. Receiving external events

The ATEAS Security system includes an expanding number for possible event sources which are capable of synchronizing the entire camera system. For example, either motion detection, camera failure or alarm input (de)activation on one or more camera servers and on one or more devices can simultaneously invoke various reactions according to the event scenario setup (starting a record or enhancing the frame rate, activation of alarm inputs, switching a defined camera view to monitors of defined users, positioning the camera, sending notifications via e-mail etc.).

Development within the branch of intelligent video enables loading applications into end camera units. These applications represent specific event sources. Integration tendencies additionally require the system to react to other systems (input verification system, ARC and many others). You can invoke any camera system reactions from any of these systems using the ATEAS API program interface.

This is the general form of a text message:

[ATEAS EVENT {START,STOP} camera code_name data]

where

(camera) stands for the camera number on a particular camera server (this value ranges from 1 to 999 – the highest number of a camera on a particular server),
(code_name) is a registered codename of a custom camera event defined in the device administration section – custom events,
(data) can contain random text information, stored together with the event and in the metadata database. The length is limited to 200 characters. The data item is optional and does not have to be contained in the message.

Examples:

[ATEAS EVENT START 2 TAMPER]
[ATEAS EVENT STOP 2 TAMPER]
[ATEAS EVENT START 2 AUDIO]
[ATEAS EVENT START 1 CARWEIGHT 1500kg]

NOTE

If the event is not or cannot be ended by the STOP command, it will be handled as in internal event according to the local setup of event parameters (i.e. it can be ended automatically).

Example of use:

Axis cameras enable to connect for example audio detection or camera tampering (Active Tampering Alarm) modules via the web interface of the corresponding camera. In order to do so, you must to create a TCP event server (set an IP address of the camera server and set the event port of the camera server to 8505) in the Event Servers (or Events - Recipients) section and create an Event in the Event Types (or Events - Action Rules) section. This event has to be connected to the TCP server and must contain a message in accordance with ATEAS API requirements.

1.1.3. Metadata injection

For injecting metadata into a camera server, used for video synchronization, you not only can use the event messages, but also direct metadata injection messages. The direct input of metadata will not result in an event.

[ATEAS META camera time code_name data]

where

(camera) specifies the camera number on the given camera server,
(time) is a numerical value, which represents the timestamp, a zero value indicates the time will be determined by the server, a positive value is interpreted as an absolute time in UTC expressed in 100-nanosecond intervals starting from 1 January 1601, a negative value is interpreted as a time shift from the current server time into the past, expressed in milliseconds,
(code_name) is a registered codename of a custom camera event defined in the device administration section– custom events,
(data) can contain random text information stored in the metadata database. The length is limited to 200 characters. The data item is optional and does not have to be contained in the message.

Examples:

[ATEAS META 1 0 SCAN AB512459]
[ATEAS META 2 -1000 SCAN AC548947]
[ATEAS META 3 132125472000000000 SCAN AC548947]

NOTE

Time shifts outside of shifts greater than 30 days in the past and 1 minute into the future will not be accepted.

NOTE

Time shifts are good to use when external data is added to the system offline or cannot be sent accurately in real time.

1.2. ATEAS API for the administration server (ATEAS Administrator)

1.2.1. Communication principle

The administration server can receive events through either a specific event port 8504 (see the network configuration appendix) or serial ports COM1 and COM2. This data is in text form and you

can set the text encoding for each communication channel. ASCII encoding is used by default. Other encoding types available include UNICODE, UTF8, Windows 1250 and others.

The principle of network communication using the TCP/IP protocol: When forwarding an ATEAS API command, the party generating this command has to establish a TCP/IP connection to the camera server. You can forward several commands within the frame of one TCP connection as well create a new connection for every single command. In order to receive ATEAS API commands, the administration server enables up to 50 simultaneous connections to the input port. Therefore, if the commands are generated from more than 50 sources, inactive connections will have to be closed.

Principle of communication when using the serial port: When forwarding a command using the serial connection to port COM1 or COM2, the party generating the command has to have the corresponding port opened with the same parameters as in the ATEAS Security. These parameters especially include bit rate, parity, number of data bits and the number of stop bits.

General syntax: All ATEAS API commands are enclosed in square brackets. The data outside these brackets is not evaluated. This protocol does not distinguish between upper and lower case.

1.2.2. Receiving external events

This is the general form of a text message:

[ATEAS EVENT {START,STOP} object id]

where

(object) is a numeric value ranging from 1 to 10 000. This value is the main code that identifies an event – e.g. a number for the monitored building,
(id) is a numeric value within the range of 1 to 99 999. This value is an additional code identifying an event – e.g. number of a loop in the monitored building.

Examples:

[ATEAS EVENT START 1 1]
[ATEAS EVENT STOP 1 1]
[ATEAS EVENT START 100 2]
[ATEAS EVENT START 150 60]

Example of use:

Alarm receiving centre (ARC), produced by the NAM company, can be extended for SERVICE COM module, which forwards events to the system using a serial connection. The form of information is modifiable and can be set to the expected form in accordance with ATEAS API. You must correctly specify the serial connection parameters and create event scenarios which will then be assigned to individual objects and loops.

1.2.3. Controlling the video wall

This is the general form of a text message:

[ATEAS VIDEOWALL monitor (submonitor = 0) {serverid | 0} {deviceid | urlid} (wallid = 1) (meta = 0)]

where

(monitor) is a numeric value ranging from 1 to 192, which specifies the number of a monitor included in either physical or virtual video wall,
(submonitor) is a numeric value ranging from 0 to 16, which specifies the number of a submonitor if a Quad, Triple, or Sixteen type monitor is used,
(serverid) is a numeric value ranging from 0 to 9 999, which identifies a camera server,
(deviceid) is a numeric value ranging from 0 to 999, which indicates a camera number on a server,
(urlid) is a numeric value ranging from 1 to 9 999, which indicates a URL number (if serverid is 0),
(wallid) is a numeric value ranging from 1 – 1000, which indicates a video wall number in the system,
(meta) is a numeric value 0 or 1 for displaying metadata on the wall.

Examples:

[ATEAS VIDEOWALL 1 1 3]
[ATEAS VIDEOWALL 2 1 4]
[ATEAS VIDEOWALL 3 1 1 4]

NOTE

If both the serverid and deviceid values are equal to zero, the corresponding monitor will be turned off (the video will be turned off and the monitor will be switched to a default state showing the ATEAS logo).

NOTE

To ensure compatibility, the default value of the submonitor is 0. If a monitor is switched to either the Standard or Alarm type, this value must be set to 0 (if specified). When using a Quad type monitor, this value must be set to 1, 2, 3, or 4 and so on.

NOTE

You can also switch monitors automatically (ATEAS API is not used) to the video wall. To do so, an event must occur and the corresponding monitor must be included within the event monitors group. For further information, see chapters regarding the video wall.

NOTE

The wallid value is optional and if it is not provided, a value of one is used, which indicates the use of the predefined main video wall. If you intend to enter a video wall number, you must also enter the value for the submonitor. Otherwise the message cannot be properly parsed.

If serverid is set to zero, a positive deviceid is interpreted as an URL ID which has been entered by an authorized user.

Example of use: Using any external application, you can switch cameras from any server to either the virtual or physical video wall.

1.2.4. Event notifications

CAUTION

Events are sent via the TCP/IP communication channel only. Therefore, it is not possible to receive event information, for example, via a serial connection.

Considering the information from previous subchapters, system event information is sent over this communication channel upon establishing a TCPIP connection. All events invoked by camera activity are sent including motion detection, alarm input activation, camera unavailable event, custom camera

events or LP recognition events. Using this one single connection to the system administration server, events are sent from all camera servers independently of their location.

Individual events are sent containing all significant facts in the form of short XML documents.

Event start

The following is the basic form of the XML document, generated and sent via TCPIP connection, providing an event occurs:

```
<?xml version=”1.0” encoding=”utf-8”?>
<ateas>
      <event>
            <id>ID</id>
            <imageid>Image ID</imageid>
            <level>Level</level>
            <server>
                  <id>Server ID</id>
                  <name>Server name</name>
            </server>
            <camera>
                  <id>Camera ID</id>
                  <name>Camera name</name>
            </camera>
            <source>
                  <id>Source ID</id>
            </source>
            <datetime>
                  <utcstamp>Timestamp</utcstamp>
                  <localvalue>Date and time</localvalue>
            </datetime>
            <data>Data</data>
            <dataex>Data 2</dataex>
            <uuid>UUID</uuid>
            <videoobject>
                  <rectangle>Position</rectangle>
            </videoobject>
      </event>
</ateas>
```

Event stop

```
<?xml version=”1.0” encoding=”utf-8”?>
<ateas>
      <eventstop>
            <id>ID</id>
            <datetime>
                  <utcstamp>Timestamp</utcstamp>
                  <localvalue>Date and time</localvalue>
            </datetime>
            <data>Data</data>
      </eventstop>
</ateas>
```

The meaning of each value within XML tags is as follows:

ID – Unambiguous event identifier (ascending from number 1 since the server started).

Image ID – Identifier which unambiguously links XML notifications to exported images via FTP protocol. This ID is the initial part of the exported file name.

Level – Event significance level, the system distinguishes a common event (1) and alarm (2) according to time plan settings.

Server ID – Server identification, e.g. 1, 2. Always 1 for HOME and PROFESSIONAL editions.

Server name – Server name, e.g. Server 1.

Camera ID – Camera identification, e.g. 1, 2.

Camera name – Camera name, e.g. Camera 1.

Source ID – Event source identification. See list below.

Timestamp – Absolute time when the event occurred expressed as the number of 100 nanosecond intervals elapsed since midnight January 1, 1601 UTC. Therefore, the information is cleaned of time zone and daylight savings influences, e.g. 128989433710312500.

Date and time – Information in d.M.yyyy H:mm:ss format containing the date and time an event occurred considering the time zone and daylight savings, e.g. 9.10.2009 9:05:51.

Data – Event data, can be used for extended event data, e.g. vehicle LP – 2A5 6217, AEP 44-44.

Data 2 – Additional event data according to the event's type.

UUID – Additional identifier of the event source (if available) that may help identifying the source.

Position – Indicates the position of the object in the image in X Y W H format, whereby X Y are coordinates of the object position and W and H indicate the width and height of the object. All data is specified in absolute coordinates for the size of the associated image.

The source ID value is decisive for evaluating events in terms of the action which invoked the event. This value expresses the event source, meaning the actual proceedings that lead to the origin of the event on the relevant camera. ATEAS API currently distinguishes and sends the following event source label values:

1 – motion detection on the camera, data is empty
2 – device unavailable, data is empty
3 – camera alarm input activation, data contains input number (1, 2, 3, 4, …)

10 – vehicle LP recognition – unregistered, data contains the vehicle LP in decorative form
11 – vehicle LP recognition – white listed, data contains the vehicle LP in decorative form
12 – vehicle LP recognition – black listed, data contains the vehicle LP in decorative form
13 – vehicle LP recognition – user defined 1, data contains the vehicle LP in decorative form
14 – vehicle LP recognition – user defined 2, data contains the vehicle LP in decorative form

20 – analytical event – intrusion, data is empty
21 – analytical event – tripwire detection, data is empty
22 – analytical event – fence detection, data is empty
23 – analytical event – unattended object, data is empty
24 – analytical event – removed object, data is empty
25 – analytical event – stopped vehicle, data is empty
26 – analytical event – loitering, data is empty
27 – analytical event – camera shift, data is empty
28 – analytical event – no video signal, data is empty
29 – analytical event – object tracking, data is empty
30 – analytical event – bad signal, data is empty

31 – analytical event – face detection, data is empty

32 – video quality event, data contains the frame rate threshold value

40 – motion detection on the server, data is empty

51 – 100 – custom camera events defined by the administrator can be used for random events, for example, camera tampering, audio detection and others, data may contain custom event source data

110 – manual camera recording event, data is empty

111 – 130 – Onvif event source, data may contain Onvif event source data

131 – 150 – complex event sources created by the administrator consisting of other elementary event sources, data is empty

151 – 200 – custom camera events defined by the administrator can be used for random events, for example, camera tampering, audio detection and others, data may contain custom event source data

201 – 250 – analytical events defined by the administrator evaluated by neural network modules on the camera server

NOTE

Establishing camera events and sending them in terms of the ATEAS API to all established TCPIP connections is only possible providing the relevant event sources are statically or dynamically time mapped. The event scenario can be defined optionally. See the Event management chapter for further information.

NOTE

In order to establish a TCP connection for communication via ATEAS API, the availability of the relevant network port shall be secured under ATEAS Administrator described in the document appendix.

NOTE

Using the IP filtering tool integrated in the ATEAS Security system, you can specify the addresses from which connection will be permitted to the ATEAS API communication channel.

NOTE

The XML document can also include characters with diacritical marks or specific symbols from other languages (e.g. in the camera name), in this case, attention is required when configuring the text coding when opening the TCP channel using the ATEAS Observer client. Besides XML documents, the server can also send keep-alive bytes with a value of 0, characters appearing outside the XML documents, however, do not require any special attention.

CAUTION

The event ID is a unique identifier used to incrementally number the events starting from 1 since the start of the administration server. Restarting the administration server therefore resets the number back to 1 and can also result in some events not being closed.

1.2.5. User notifications

By using the user management notifications, any third party system can receive real time information about users logging into the system or leaving it. A different message type is capable of inducing a forced user logout of any logged user.

NOTE

With the help of these notifications it is possible to easily integrate with an access control system to perform real time checks, whether any particular user is permitted to enter the camera system at any given time or if somebody tries to misuse an account.

After a user has been logged on or off, a message will be sent in the form of a simple XML document with the following structure.

```
<?xml version=”1.0” encoding=”utf-8”?>
```

```
<ateas>
      <user>
            <id>ID</id>
            <name>Username</level>
            <action>Action</action>
            <datetime>
                  <utcstamp>Timestamp</utcstamp>
                  <localvalue>Date and time</localvalue>
            </datetime>
      </user>
</ateas>
```

The inner values of the XML document are interpreted as follows:

ID – A unique identifier of the user, e.g. 20.

Username – The name of the user as registered in the system.

Action – The value ‘logon’ (when logging on) or ‘logout’ (when logging out).

Timestamp – Absolute time when the event occurred expressed as the number of 100 nanosecond intervals elapsed since midnight January 1, 1601 UTC. Therefore, the information is cleaned of time zone and daylight savings influences, e.g. 128989433710312500.

Date and time – Information in d.M.yyyy H:mm:ss format containing the date and time an event occurred considering the time zone and daylight savings, e.g. 9.10.2009 9:05:51.

1.3. Parameterized application launch

ATEAS Security applications can be launched with additional parameters that are passed to the application executable while starting. In Windows these parameters can be added under the service settings. All existing parameters are described below.

1.3.1. Administration server

Parameter	Values	Meaning	Note
-ssl	password	Certificate password	Necessary to use when the PFX certificate is password protected.

1.3.2. Camera server

Parameter	Values	Meaning	Note
-ssl	password	Certificate password	Necessary to use when the PFX certificate is password protected.
-loglevel	0 - 1	Log level setting	Using a positive value activates logging of the record buffer level in the log subfolder.