ONVIF™

Action Engine Service Specification

Version 22.12

December, 2022
### CONTENTS

1. **Scope** ......................................................... 4
2. **Normative references** ........................................ 4
3. **Terms and Definitions** ....................................... 4
   - 3.1 Definitions .................................................. 4
   - 3.2 Abbreviations ............................................... 4
   - 3.3 Namespaces .................................................. 4
4. **Overview** .................................................... 5
5. **Service** ....................................................... 6
   - 5.1 Action representation ...................................... 6
   - 5.2 Action description language ............................. 6
   - 5.3 Operations on actions ..................................... 6
     - 5.3.1 GetSupportedActions .................................. 6
     - 5.3.2 GetActions ............................................... 7
     - 5.3.3 CreateActions .......................................... 7
     - 5.3.4 ModifyActions .......................................... 7
     - 5.3.5 DeleteActions ........................................... 8
     - 5.3.6 GetServiceCapabilities .............................. 8
     - 5.3.7 GetActionTriggers ..................................... 9
     - 5.3.8 CreateActionTriggers ................................. 9
     - 5.3.9 ModifyActionTriggers ............................... 9
     - 5.3.10 DeleteActionTriggers ............................... 10
   - 5.4 Standard actions .......................................... 10
     - 5.4.1 CommandAction ........................................ 10
     - 5.4.2 E-Mail Action ............................................ 11
     - 5.4.3 HTTP POST Action .................................... 11
     - 5.4.4 FTP Action .............................................. 12
     - 5.4.5 SMS Action .............................................. 12
     - 5.4.6 Camera Local Recording Action .................... 13
   - 5.5 Capabilities ................................................ 13
   - 5.6 Events ...................................................... 13
     - 5.6.1 Action completed ...................................... 13
     - 5.6.2 Action failed ............................................ 14
5. **Revision History** ........................................... 18

**Annex A** Examples (informative) ......................... 15
   - A.1 Command Action configuration .......................... 15
   - A.2 E-Mail Action configuration .............................. 15
   - A.3 HTTP POST Action configuration ....................... 16
   - A.4 FTP Action configuration ................................ 16
   - A.5 GetActions Response ...................................... 17
1 Scope

This document defines the web service interface for configuration of the Actions and Action Trigger conditions based on events.

2 Normative references


3 Terms and Definitions

3.1 Definitions

**Action Trigger** Condition. Topic and message content filter based condition

**Action** Executed when one of its conditions (Action Trigger) is satisfied

3.2 Abbreviations

- **FTP** File Transfer Protocol
- **E-Mail** Electronic Mail
- **SMTP** Simple Mail Transfer Protocol
- **POP** Post Office Protocol
- **SMS** Short Message Service

3.3 Namespaces

Table 1 lists the prefix and namespaces used in this specification. Listed prefixes are not part of the standard and an implementation can use any prefix.

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Namespace URI</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tae</td>
<td><a href="http://www.onvif.org/ver10/actionengine/wsd1">http://www.onvif.org/ver10/actionengine/wsd1</a></td>
<td>The namespace for the WSDL action engine service.</td>
</tr>
<tr>
<td>ter</td>
<td><a href="http://www.onvif.org/ver10/error">http://www.onvif.org/ver10/error</a></td>
<td>The namespace for ONVIF defined faults.</td>
</tr>
<tr>
<td>tns1</td>
<td><a href="http://www.onvif.org/ver10/topics">http://www.onvif.org/ver10/topics</a></td>
<td>The namespace for the ONVIF topic namespace</td>
</tr>
</tbody>
</table>

This specification references to the following namespaces (listed in Table 2) by specified prefix.

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Namespace URI</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>soapenv</td>
<td><a href="http://www.w3.org/2003/05/soap-envelope">http://www.w3.org/2003/05/soap-envelope</a></td>
<td>Envelope namespace as defined by SOAP 1.2 [SOAP 1.2, Part 1]</td>
</tr>
<tr>
<td>xs</td>
<td><a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a></td>
<td>Instance namespace as defined by XS [XML-Schema, Part1] and [XML-Schema, Part 2]</td>
</tr>
</tbody>
</table>
4 Overview

This document describes Action Engine web service depicted also in video analytics architecture (Figure 6) in [Video Analytics Service Specification]. The Action Engine service enables new extensible device capabilities by providing further integration opportunities with other networked devices through actions that are triggered by events.

![Figure 1: Video Analytics Architecture](image)

Action Engine service interface allows service requester to list available action types (defined in action description language), creates new actions, modify the action configurations, and delete actions.

![Figure 2: Action creation and configuration sequence](image)

Action Engine interface also allows service requester to define the action triggers that are defined based on Topic expression (Section Topic Filter in [Core Specification]) and Message Content expression (Section Message Content Filter in [Core Specification]). Service requester can list, create, modify, and delete the action triggers. Action Engine service provider passes the triggering event information to triggered action instances.
5 Service

5.1 Action representation

The configuration of an action contains the action name and the other Action type specific parameter configurations. The Parameters element of the Action element contains the configurations of action parameters. Each Parameter is defined by either a SimpleItem or an ElementItem data type. The Name attribute of each parameter shall be unique within the parameter list. The Value attribute of SimpleItem contains the parameter configuration. The child element of an ElementItem contains the configured value of an ElementItem. It is RECOMMENDED to represent as many configuration parameters as possible by SimpleItems.

5.2 Action description language

The description of an Action contains the action type name (Name) and the type information of all parameters belonging to the defined Action type. The parameters of a certain Action Type are listed below the ParameterDescription element. All parameters are either Simple or ElementItems and can be described by either a SimpleItemDescription or an ElementItemDescription. Both ItemDescriptions contain a Name attribute to identify the parameter and a Type attribute to reference a specific XML schema type. The Type attribute of the SimpleItemDescription shall reference to XML schema simple type definition. The Type attribute of the ElementItemDescription shall reference a global element declaration of an XML schema.

Section 5.4 demonstrates the usage of the Action Description Language. The following definitions are included for convenience:

```xml
<xs:element name="ActionDescription" type="tt:ActionConfigDescription"/>
<xs:complexType name="ActionConfigDescription">
  <xs:sequence>
    <xs:element name="ParameterDescription" type="tt:ItemListDescription"/>
  </xs:sequence>
  <xs:attribute name="Name" type="xs:QName" use="required"/>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
```

5.3 Operations on actions

If the device supports an Action Engine service as defined by ONVIF, then it shall implement the following operations to manage actions. The Create/Delete/Modify operations are atomic, meaning that either all modifications can be processed or the complete operation shall fail.

5.3.1 GetSupportedActions

The service provider returns the supported action types. The response returns a list of Action Descriptions according to the Action Description Language described in Section 5.2. The response also contains a list of URLs that provide the location of the schema files. These schema files describe the types and elements used in the Action Descriptions. If action descriptions reference types or elements of the ONVIF schema file, the ONVIF schema file shall be explicitly listed.

REQUEST:

This is an empty message.

RESPONSE:

- **SupportedActions [tt:SupportedActions]**
  The response contains the supported actions.

FAULTS:

- There are no operation specific faults.
ONVIF™ – 7 – ActionEngine – Ver. 22.12

ACCESS CLASS:

READ_SYSTEM

5.3.2 GetActions

The get actions operation retrieves currently installed Actions.

REQUEST:

This is an empty message.

RESPONSE:

- Action - optional, unbounded [tt:Action]
  The response is a list of installed actions for the specified configuration.

FAULTS:

- There are no operation specific faults.

ACCESS CLASS:

READ_SYSTEM

5.3.3 CreateActions

The create action operation adds actions to configuration. The create action operation is atomic. If a service provider can not create all of requested actions, then, the service provider responds with a fault message.

REQUEST:

- Action - unbounded [tt:ActionConfiguration]
  The request message specifies list of Actions to be added.

RESPONSE:

- Action - optional, unbounded [tt:Action]
  The response is a list of created actions.

FAULTS:

  The requested actions configuration is not valid.
  The requested action type is not known.
- enc:Receiver - ter:Action - ter:TooManyActions
  There is not enough space to add actions.

ACCESS CLASS:

WRITE_SYSTEM

5.3.4 ModifyActions

The modify action operation modifies action configurations. The modify action operation is atomic. If a service provider can not modify all of requested action configurations, then, the service provider responds with a fault message.

REQUEST:

- Action - unbounded [tt:Action]
  The request message specifies list of Actions to be modified.
RESPONSE:

This is an empty message.

FAULTS:

  The requested action configuration is not valid.

  The actions do not exist.

ACCESS CLASS:

**WRITE_SYSTEM**

All action parameters, except the action type, can be modified. The service provider shall return `InvalidAction` error if the request attempts to change the action type with modify action request.

### 5.3.5 DeleteActions

The delete operation deletes actions. The delete action operation is atomic. If a service provider cannot delete all of requested actions, then, the service provider responds with a fault message.

REQUEST:

- `Token - unbounded [xs:ReferenceToken]`
  The request message specifies list of Actions to be removed.

RESPONSE:

This is an empty message.

FAULTS:

  The actions do not exist.

- `env:Receiver - ter:Action - ter:ConfigurationConflict`
  The service cannot delete the actions without creating a conflicting configuration.

ACCESS CLASS:

**WRITE_SYSTEM**

### 5.3.6 GetServiceCapabilities

The get capabilities operation returns the Action Engine capabilities.

REQUEST:

This is an empty message.

RESPONSE:

- `Capabilities [tt:ActionEngineCapabilities]`
  The capability response message contains Action Engine capabilities information.

FAULTS:

- `There are no operation specific faults.`

ACCESS CLASS:

**PRE_AUTH**
The capabilities data structure indicates the maximum number of action and action trigger instances.

5.3.7 GetActionTriggers

Returns configured action triggers.

REQUEST:

This is an empty message.

RESPONSE:

- **ActionTrigger** - optional, unbounded [tt:ActionTrigger]
  The request message specifies list of action triggers.

FAULTS:

- There are no operation specific faults.

ACCESS CLASS:

  **READ_SYSTEM**

5.3.8 CreateActionTriggers

Creates action triggers. The create action triggers operation is atomic. If a service provider can not create all of requested action triggers, then, the service provider responds with a fault message.

REQUEST:

- **ActionTrigger** - unbounded [tt:ActionTriggerConfiguration]
  The request message specifies list of action triggers to be created.

RESPONSE:

- **ActionTrigger** - optional, unbounded [tt:ActionTrigger]
  The request message specifies list of action triggers.

FAULTS:

- **env:Sender** - **ter:InvalidArgVal** - **ter:InvalidActionTrigger**
  The requested action trigger configuration is not valid.
- **env:Receiver** - **ter:Action** - **ter:TooManyActionTriggers**
  There is not enough space to add action triggers.

ACCESS CLASS:

  **WRITE_SYSTEM**

5.3.9 ModifyActionTriggers

Modifies existing action triggers. The modify action triggers operation is atomic. If a service provider can not modify all of requested action trigger configurations, then, the service provider responds with a fault message.

REQUEST:

- **ActionTrigger** - unbounded [tt:ActionTrigger]
  The request message specifies list of action triggers to be modified.

RESPONSE:

This is an empty message.
FAULTS:

  The requested action trigger configuration is not valid.
  The action triggers do not exist.

ACCESS CLASS:

**WRITE_SYSTEM**

### 5.3.10 DeleteActionTriggers

Deletes action triggers. The delete action triggers operation is atomic. If a service provider cannot delete all of requested action triggers, then, the service provider responds with a fault message.

REQUEST:

- **Token - unbounded [tt:ReferenceToken]**
  The request message specifies list of action triggers to be removed.

RESPONSE:

This is an empty message.

FAULTS:

  The action triggers do not exist.

ACCESS CLASS:

**WRITE_SYSTEM**

### 5.4 Standard actions

#### 5.4.1 CommandAction

The command action executes one or more ONVIF commands on the device itself or on a remote device. This mechanism is applicable to a wide range of ONVIF commands.

```
<tt:ActionDescription Name="tt:CommandAction">
  <tt:ParameterDescription>
    <tt:SimpleItemDescription Name="XAddr" Type="xs:anyURI"/>
    <tt:SimpleItemDescription Name="Operation" Type="xs:QName"/>
    <tt:ElementItemDescription Name="Parameters" Type="tt:AnyHolder"/>
  </tt:ParameterDescription>
</tt:ActionDescription>
```

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xaddr</td>
<td>Endpoint where the command shall be executed. Use localhost for local commands instead to avoid that device IP address changes break the commands.</td>
</tr>
<tr>
<td>Operation</td>
<td>Operation name including namespace</td>
</tr>
<tr>
<td>Parameters</td>
<td>Input parameter of the operation as defined in the corresponding ONVIF WSDL file.</td>
</tr>
</tbody>
</table>
5.4.2 E-Mail Action

E-mail action definition allows application to send an e-mail to the configured addresses. The e-mail content is provided during the execution.

```xml
<tt:ActionDescription Name="tt:EMailAction">
  <tt:ParameterDescription>
    <tt:ElementItemDescription Name="Destinations" Type="tae:EMailServerConfiguration"/>
    <tt:ElementItemDescription Name="Receivers" Type="tae:EMailReceiverConfiguration"/>
    <tt:SimpleItemDescription Name="Sender" Type="xs:string"/>
    <tt:SimpleItemDescription Name="Subject" Type="xs:string"/>
    <tt:ElementItemDescription Name="Body" Type="tae:EMailBodyTextConfiguration"/>
    <tt:ElementItemDescription Name="Attachment" Type="tae:EMailAttachmentConfiguration"/>
    <tt:ElementItemDescription Name="MediaReference" Type="tt:MediaSource"/>
  </tt:ParameterDescription>
</tt:ActionDescription>
```

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMaiServerConfiguration</td>
<td>Contains Email Server configuration</td>
</tr>
<tr>
<td>SMTP Configuration</td>
<td>SMTP Server IP address and port number</td>
</tr>
<tr>
<td>POP Configuration</td>
<td>POP Server IP address and port number</td>
</tr>
<tr>
<td>Authentication</td>
<td>Configure username and password</td>
</tr>
<tr>
<td>Receivers</td>
<td>Contains list of receivers (TO and CC fields)</td>
</tr>
<tr>
<td>Sender (From)</td>
<td>Sender information</td>
</tr>
<tr>
<td>Subject</td>
<td>E-Mail subject Line configuration</td>
</tr>
<tr>
<td>Body</td>
<td>E-Mail body text configuration</td>
</tr>
<tr>
<td>Attachment</td>
<td>E-Mail Attachment file name configuration</td>
</tr>
<tr>
<td>File Name</td>
<td>Attachment file name</td>
</tr>
<tr>
<td>File Name Suffix</td>
<td>Attachment file name suffix configuration</td>
</tr>
<tr>
<td>Media Reference</td>
<td>Media Profile Token</td>
</tr>
</tbody>
</table>

5.4.3 HTTP POST Action

HTTP POST action definition allows application to send data to the configured address via HTTP POST mechanism. The content is provided during the execution.

```xml
<tt:ActionDescription Name="tt:HttpPOSTAction">
  <tt:ParameterDescription>
    <tt:ElementItemDescription Name="Destinations" Type="tae:HttpHostConfigurations"/>
    <tt:ElementItemDescription Name="POSTContent" Type="tae:PostContentConfiguration"/>
  </tt:ParameterDescription>
</tt:ActionDescription>
```

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Destinations</td>
<td>Contains HTTP server configurations</td>
</tr>
<tr>
<td>Server Address</td>
<td>Server IP address and port number</td>
</tr>
<tr>
<td>Authentication</td>
<td>Server Authentication configuration (username, password, authentication mechanism)</td>
</tr>
</tbody>
</table>
### Parameters

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>URI</td>
<td>Request-Uri</td>
</tr>
<tr>
<td>Protocol</td>
<td>Select HTTP or HTTPS</td>
</tr>
<tr>
<td>POST Content</td>
<td>POST content configuration for form data, triggering event data, and media</td>
</tr>
<tr>
<td>Media Reference</td>
<td>Media Profile Token</td>
</tr>
</tbody>
</table>

### 5.4.4 FTP Action

FTP action definition allows application to send data to the configured address via FTP mechanism. The content is provided during the execution.

```
<tt:ActionDescription Name="tt:FtpAction">
  <tt:ParameterDescription>
    <tt:ElementItemDescription Name="Destinations" Type="tae:FtpHostConfigurations"/>
    <tt:ElementItemDescription Name="FtpContent" Type="tae:FtpContentConfiguration"/>
    <tt:ElementItemDescription Name="MediaReference" Type="tt:MediaSource"/>
  </tt:ParameterDescription>
</tt:ActionDescription>
```

Table 6: FTP Action configuration parameters

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Destinations</td>
<td>Server Address Server IP and Port number configuration</td>
</tr>
<tr>
<td></td>
<td>Upload Path Upload path on FTP server</td>
</tr>
<tr>
<td></td>
<td>Authentication Authentication configuration</td>
</tr>
<tr>
<td>FTP Content</td>
<td>FTP Content Configuration Includes configurations for the upload of sequence of images and upload of a file from local storage</td>
</tr>
<tr>
<td>Media Reference</td>
<td>Media Profile Token</td>
</tr>
</tbody>
</table>

### 5.4.5 SMS Action

SMS action definition allows application to send data to the configured address via SMS Text Messaging mechanism. The content is provided during the execution.

```
<tt:ActionDescription Name="tt:SMSNotificationAction">
  <tt:ParameterDescription>
    <tt:ElementItemDescription Name="SMSProvider" Type="tae:SMSProviderConfiguration"/>
    <tt:ElementItemDescription Name="SMSSenderInfo" Type="tae:SMSSenderConfiguration"/>
    <tt:SimpleItemDescription Name="Destination" Type="xs:string"/>
    <tt:SimpleItemDescription Name="Message" Type="tae:SMSMessage"/>
  </tt:ParameterDescription>
</tt:ActionDescription>
```

Table 7: SMS Action configuration parameters

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMSProvider</td>
<td>SMS Provider Information</td>
</tr>
<tr>
<td>SMSSenderInfo</td>
<td>SMS Sender Information</td>
</tr>
</tbody>
</table>
5.4.6 Camera Local Recording Action

Note, that this interface has been deprecated since the same functionality can be configured directly via the recording control service.

Camera Local Recording Action definition allows application to initiate recording of data to the camera local storage. The content is provided during the execution.

<tt:ActionDescription Name="tt:RecordingAction">
    <tt:ParameterDescription>
        <tt:ElementItemDescription Name="RecordingConfiguration" Type="tae:RecordingActionConfiguration"/>
    </tt:ParameterDescription>
</tt:ActionDescription>

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Destination</td>
<td>Destinations that will receive the message</td>
</tr>
<tr>
<td>Message</td>
<td>Text Message</td>
</tr>
</tbody>
</table>

**Table 8: Camera Action configuration parameters**

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recording configuration</td>
<td>Local recording configuration</td>
</tr>
<tr>
<td>Pre recording duration</td>
<td>Recording before the triggering event</td>
</tr>
<tr>
<td>Post recording duration</td>
<td>Recording after alarm recording duration</td>
</tr>
<tr>
<td>Record duration</td>
<td>Record duration</td>
</tr>
<tr>
<td>Recording frame rate</td>
<td>Recording frame rate</td>
</tr>
<tr>
<td>Audio Recording on/off</td>
<td>Whether Audio recording on/off</td>
</tr>
</tbody>
</table>

5.5 Capabilities

The capabilities reflect optional functions and functionality of a service. The following capabilities are available:

**MaximumActions:** The maximum number of actions that the service provider can concurrently support.

**MaximumTriggers:** The maximum number of trigger configurations that the service provider can concurrently support.

5.6 Events

The action engine events allow controlling of action execution as well as building of action chains.

The action service shall dispatch events through the event service.

5.6.1 Action completed

The device shall be capable of generating the following event whenever an action has been completed. This event is triggered whenever an action completes without a fault message. When an event is generated due to execution of CommandAction, the Data element of event message can contain the typed information of the request and response message payloads of corresponding ONVIF message exchange.

**Topic:** tns1:ActionEngine/Completed

<tt:MessageDescription IsProperty="false">
    <tt:Source>
5.6.2 Action failed

The device shall be capable of generating the following event whenever an action has been completed with an error.

This event is triggered whenever an action completes with a fault message or the action could not be executed because of the other error. The response contains the fault codes including any sub codes.

Topic: tns1:ActionEngine/Failed

```xml
<tt:MessageDescription IsProperty="false">
  <tt:Source>
    <tt:SimpleItemDescription Name="Token" Type="tt:ReferenceToken"/>
    <tt:SimpleItemDescription Name="ActionName" Type="xs:string"/>
  </tt:Source>
  <tt:Data>
    <tt:ElementItemDescription Name="RequestResponse" Type="tt:ActionEngineEventPayload"/>
  </tt:Data>
</tt:MessageDescription>
```
Annex A.
Examples (informative)

A.1 Command Action configuration

Starting and stopping of recording job on Recording Control Service by using CommandAction definition is illustrated in the following.

Start a recording on the same device when a relay is switched on.

```xml
<tae:Action Token="453638">
  <tae:Configuration Name="StartRecording" Type="tae:CommandAction">
    <tt:Parameters>
      <tt:SimpleItem Name="XAddr" Value="http://127.0.0.1/Onvif/device_service"/>
      <tt:SimpleItem Name="Operation" Value="tt:SetRecordingJobMode"/>
      <tt:ElementItem Name="Parameters">
        <tt:JobToken>123</tt:JobToken>
        <tt:Mode>Active</tt:Mode>
      </tt:ElementItem>
    </tt:Parameters>
  </tae:Configuration>
</tae:Action>

Stop the recording one minute after the relay is switched off.

```xml
<tae:Action Token="5674749">
  <tae:Configuration Name="StopRecording" Type="tae:CommandAction">
    <tt:Parameters>
      <tt:SimpleItem Name="XAddr" Value="http://127.0.0.1/Onvif/device_service"/>
      <tt:SimpleItem Name="Operation" Value="tt:SetRecordingJobMode"/>
      <tt:ElementItem Name="Parameters">
        <tt:JobToken>123</tt:JobToken>
        <tt:Mode>Idle</tt:Mode>
      </tt:ElementItem>
    </tt:Parameters>
  </tae:Configuration>
</tae:Action>

A.2 E-Mail Action configuration

The following demonstrates the E-Mail action configuration information.

```xml
<tt:Configuration Token="576858" Name="Notify_SGrp_1" Type="tt:EMailAction">
  <tt:Parameters>
    <!-- Server configurations -->
    <tt:ElementItem Name="Destinations">
      <tt:SMTPConfig>
        <tt:HostAddress formatType="ipv4">172.34.123.65</tt:HostAddress>
      </tt:SMTPConfig>
      <tt:POPConfig>
        <tt:HostAddress formatType="ipv4">162.34.123.56</tt:HostAddress>
      </tt:POPConfig>
      <tt:AuthenticationConfig mode="none"/>
    </tt:ElementItem>
    <!-- Email receiver configurations -->
    <tt:ElementItem Name="Receivers">
      <tt:TO>sg1@hq.co</tt:TO>
      <tt:TO>sg2@hq.co</tt:TO>
      <tt:CC>sgm@hq.co</tt:CC>
    </tt:ElementItem>
  </tt:Parameters>
</tt:Configuration>
```
A.3 HTTP POST Action configuration

The following demonstrates the configuration for HTTP POST action. The configuration utilizes the Media Profile Reference to indicate the media source.

```
<tt:Action Token="343234">  
  <tt:Configuration Name="POST2_VMS1" Type="tt:HttpPOSTAction">  
    <tt:Parameters>  
      <!-- HTTP Server Address configuration -->  
      <tt:ElementItem Name="Destinations">  
        <tt:HttpDestination uri="/post_event">  
          <tt:HostAddress formatType="ipv4">192.134.123.214</tt:HostAddress>  
        </tt:HttpDestination>  
      </tt:ElementItem>  
      <!-- HTTP POST Request Body configuration and Media Profile Token -->  
      <tt:ElementItem Name="POSTContent">  
        <tt:MediaReference>  
          <tt:ProfileToken>6565746</tt:ProfileToken>  
        </tt:MediaReference>  
        <tt:PostBody includeMedia="true" includeEvent="true"/>  
      </tt:ElementItem>  
    </tt:Parameters>  
  </tt:Configuration>  
</tt:Action>  
```

A.4 FTP Action configuration

An example configuration for sending images (from Media Profile) into a directory (Destinations/UploadPath) in FTP server (Destinations) for 15 minutes (FtpContent/HowLong) with 10 seconds snapshots (FtpContent/SampleInterval) with the configured file names (FtpContent/FileName).

```
<tt:Action Token="45345" >  
  <tt:Configuration Name="FTP_STORE_BRNCH1" Type="tt:FtpAction">  
    <tt:Parameters>  
      <tt:ElementItem Name="Destinations">  
        <tt:FtpDestination>  
          <tt:HostAddress formatType="ipv4">132.34.13.64</tt:HostAddress>  
          <tt:UploadPath>/home/cam2</tt:UploadPath>  
          <tt:FtpAuthentication>  
            <tt:username> </tt:username>  
            <tt:password> </tt:password>  
          </tt:FtpAuthentication>  
        </tt:FtpDestination>  
      </tt:ElementItem>  
      <tt:ElementItem Name="FtpContent">  
        <tt:FtpContentConfig Type="image_upload">  
          <tt:UploadImages>  
            <tt:HowLong>P0Y0M0DT0H15M</tt:HowLong>  
            <tt:SampleInterval>P0Y0M0DT0H0M10S</tt:SampleInterval>  
          </tt:UploadImages>  
          <tt:FileName file_name="ALM_" suffix="sequence"/>  
        </tt:FtpContentConfig>  
      </tt:ElementItem>  
    </tt:Parameters>  
  </tt:Configuration>  
</tt:Action>  
```
A.5 GetActions Response

For example, GetActionsResponse message contains installed Action information as;

```xml
..<tt:Action Token="191918">
   <tt:Configuration Name="HQMail" Type="tt:EMailAction">
      <tt:Parameters>
         <tt:ElementItem Name="Destinations">
         ...
      </tt:Parameters>
   </tt:Configuration>
</tt:Action>
<tt:Action Token="1913338">
   <tt:Configuration Name="HQMail_2" Type="tt:EMailAction">
      <tt:Parameters>
         <tt:ElementItem Name="Destinations">
         ...
      </tt:Parameters>
   </tt:Configuration>
</tt:Action>
..```

Annex B.
Revision History

<table>
<thead>
<tr>
<th>Rev.</th>
<th>Date</th>
<th>Editor</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>Feb-2012</td>
<td>Hasan T. Ozdemir</td>
<td>First release.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Change Request 1843</td>
</tr>
<tr>
<td>22.12</td>
<td>Dec-2022</td>
<td>Hans Busch</td>
<td>Deprecate camera local recording action</td>
</tr>
</tbody>
</table>