

ONVIF™ Action Engine Service Specification

Version 22.12

December, 2022



Copyright © 2008-2022 ONVIF™ All rights reserved.

Recipients of this document may copy, distribute, publish, or display this document so long as this copyright notice, license and disclaimer are retained with all copies of the document. No license is granted to modify this document.

THIS DOCUMENT IS PROVIDED "AS IS," AND THE CORPORATION AND ITS MEMBERS AND THEIR AFFILIATES, MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR TITLE; THAT THE CONTENTS OF THIS DOCUMENT ARE SUITABLE FOR ANY PURPOSE; OR THAT THE IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

IN NO EVENT WILL THE CORPORATION OR ITS MEMBERS OR THEIR AFFILIATES BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE OR CONSEQUENTIAL DAMAGES, ARISING OUT OF OR RELATING TO ANY USE OR DISTRIBUTION OF THIS DOCUMENT, WHETHER OR NOT (1) THE CORPORATION, MEMBERS OR THEIR AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, OR (2) SUCH DAMAGES WERE REASONABLY FORESEEABLE, AND ARISING OUT OF OR RELATING TO ANY USE OR DISTRIBUTION OF THIS DOCUMENT. THE FOREGOING DISCLAIMER AND LIMITATION ON LIABILITY DO NOT APPLY TO, INVALIDATE, OR LIMIT REPRESENTATIONS AND WARRANTIES MADE BY THE MEMBERS AND THEIR RESPECTIVE AFFILIATES TO THE CORPORATION AND OTHER MEMBERS IN CERTAIN WRITTEN POLICIES OF THE CORPORATION.

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
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
Annex B	Revision History	18

1 Scope

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

2 Normative references

ONVIF™ Core Specification, Version 2.2, May, 2012
<<http://www.onvif.org/specs/core/ONVIF-Core-Specification.pdf>>

ONVIF™ Recording Control Service Specification, Version 2.2, May, 2012.
<<http://www.onvif.org/specs/srv/rec/ONVIF-RecordingControl-Service-Spec.pdf>>

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 1: Namespaces used in this specification

Prefix	Namespace URI	Description
tt	http://www.onvif.org/ver10/schema	XML schema descriptions in this specification.
tae	http://www.onvif.org/ver10/actionengine/wsd	The namespace for the WSDL action engine service.
ter	http://www.onvif.org/ver10/error	The namespace for ONVIF defined faults.
tns1	http://www.onvif.org/ver10/topics	The namespace for the ONVIF topic namespace

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

Table 2: Referenced namespaces

Prefix	Namespace URI	Description
soapenv	http://www.w3.org/2003/05/soap-envelope	Envelope namespace as defined by SOAP 1.2 [SOAP 1.2, Part 1]
xs	http://www.w3.org/2001/XMLSchema	Instance namespace as defined by XS [XML-Schema, Part1] and [XML-Schema, Part 2]

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.

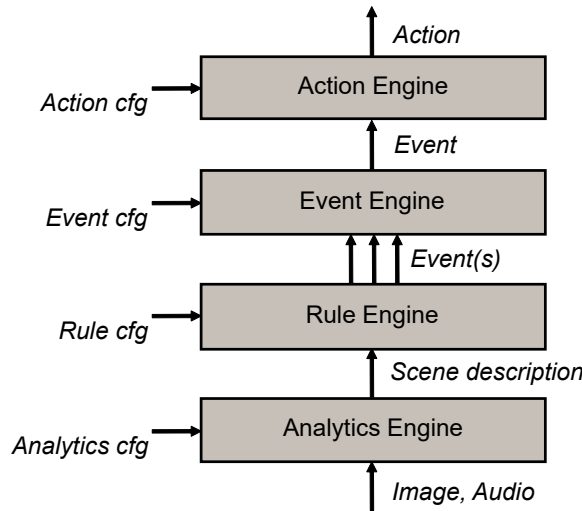


Figure 1: Video Analytics Architecture

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.

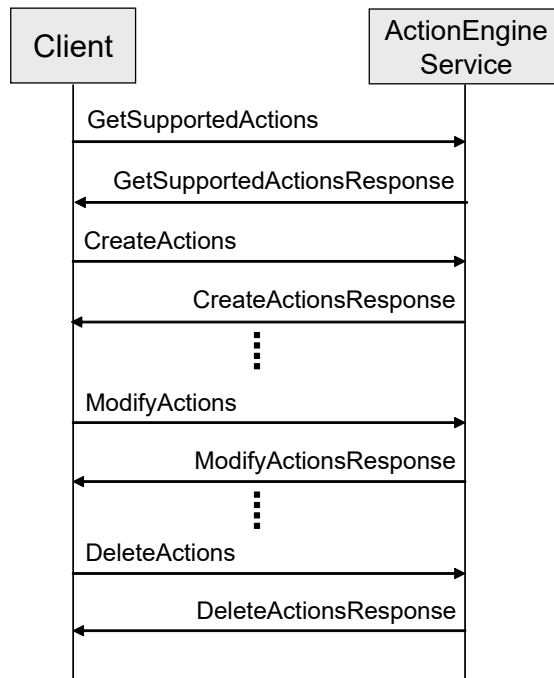


Figure 2: Action creation and configuration sequence

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:

```
<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.

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:

- **env:Sender - ter:InvalidArgVal - ter:InvalidAction**
The requested actions configuration is not valid.
- **env:Sender - ter:InvalidArgVal - ter:UknownAction**
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:

- **env:Sender - ter:InvalidArgVal - ter:InvalidAction**
The requested action configuration is not valid.
- **env:Sender - ter:InvalidArgs - ter:ActionNotExist**
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 can not 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:

- **env:Sender - ter:InvalidArgs - ter:ActionNotExist**
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:

- **env:Sender - ter:InvalidArgVal - ter:InvalidActionTrigger**
The requested action trigger configuration is not valid.
- **env:Sender - ter:InvalidArgs - ter:ActionTriggerNotExist**
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 can not 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:

- **env:Sender - ter:InvalidArgs - ter:ActionTriggerNotExist**
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 3: Standard Action configuration parameters

Parameters	Description
Xaddr	Endpoint where the command shall be executed. Use localhost for local commands instead to avoid that device IP address changes break the commands.
Operation	Operation name including namespace
Parameters	Input parameter of the operation as defined in the corresponding ONVIF WSDL file.

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.

```
<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 4: E-Mail Action configuration parameters

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

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.

```
<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 5: POST Action configuration parameters

Parameters		Description
Destinations		Contains HTTP server configurations
	Server Address	Server IP address and port number
	Authentication	Server Authentication configuration (username, password, authentication mechanism)

Parameters		Description
	URI	Request-Uri
	Protocol	Select HTTP or HTTPS
POST Content		POST content configuration for form data, triggering event data, and media
	Media Reference	Media Profile Token

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

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

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

Parameters	Description
SMSProvider	SMS Provider Information
SMSSenderInfo	SMS Sender Information

Parameters	Description
Destination	Destinations that will receive the message
Message	Text Message

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 8: Camera Action configuration parameters

Parameters	Description
Recording configuration	Local recording configuration
Pre recording duration	Recording before the triggering event
Post recording duration	Recording after alarm recording duration
Record duration	Record duration
Recording frame rate	Recording frame rate
Audio Recording on/off	Whether Audio recording on/off

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>
```

```
<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>
```

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

```
<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.

```
<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.

```
<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.

```
<tt:Action Token="576858">
  <tt:Configuration 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:ElementItem Name="Sender">NVT@123.23.23.22</tt:ElementItem>
<!-- Attached File Name configurations -->
  <tt:ElementItem Name="Attachment">
    <tt:FileName>EVT_</tt:FileName>
    <tt:doSuffix>dateTime</tt:doSuffix>
  </tt:ElementItem>
  <!-- Media Profile configuration -->
    <tt:ElementItem Name="MediaReference">
      <tt:ProfileToken>4757585</tt:ProfileToken>
    </tt:ElementItem>
  </tt:Parameters>
</tt:Configuration>
</tt:Action>

```

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:FileName file_name="ALM_" suffix="sequence"/>
          </tt:UploadImages>
        </tt:FtpContentConfig>
      </tt:ElementItem>
    </tt:Parameters>
  </tt:Configuration>
</tt:Action>

```



```
        </tt:UploadImages>
    </tt:FtpContentConfig>
</tt:ElementItem>
<tt:ElementItem Name="MediaReference">
    <tt:ProfileToken>5657959</tt:ProfileToken>
</tt:ElementItem>
</tt:Parameters>
</tt:Configuration>
</tt:Action>
```

A.5 GetActions Response

For example, GetActionsResponse message contains installed Action information as;

```
..
<tt:Action Token="191918">
  <tt:Configuration Name="HQMail" Type="tt:EmailAction">
    <tt:Parameters>
      <tt:ElementItem Name="Destinations">
        ...
      </tt:ElementItem>
    </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:ElementItem>
    </tt:Parameters>
  </tt:Configuration>
</tt:Action>
..
```

Annex B. Revision History

Rev.	Date	Editor	Changes
1.0	Feb-2012	Hasan T. Ozdemir	First release.
			Change Request 1843
22.12	Dec-2022	Hans Busch	Deprecate camera local recording action