HOW TO

External HTTPS Trigger
AXIS Camera Station 5.06 and above

Created: October 17, 2016
Last updated: November 19, 2016
Rev: 1.2

1 Please note that AXIS does not take any responsibility for how this configuration may affect your system. If the modification fails or if you get other unexpected results, you may have to restore the factory default settings as described in the User's manual.

Introduction

The following document highlights how to configure external HTTPS triggers from AXIS Camera Station. External HTTPS triggers enable external sources to trigger the rule system in AXIS Camera Station by making HTTPS requests to the AXIS Camera Station Server.
It uses the same authentication as the mobile API, i.e. the source needs to be able to authenticate as a valid AXIS Camera Station user. The following requests are possible:

- Activate trigger with id "trigger1":
  https://<address>:55756/Acs/Api/TriggerFacade/ActivateTrigger?{"triggerName":"trigger1"}
- Deactivate trigger with id "trigger1":
  https://<address>:55756/Acs/Api/TriggerFacade/DeactivateTrigger?{"triggerName":"trigger1"}
- Activate trigger with id "trigger1" and then automatically deactivate it after 30 seconds:
  https://<address>:55756/Acs/Api/TriggerFacade/ActivateDeactivateTrigger?{"triggerName":"trigger1","deactivateAfterSeconds":"30"}
- Pulse trigger with id "trigger1" (Activate followed immediately by deactivate):
  https://<address>:55756/Acs/Api/TriggerFacade/PulseTrigger?{"triggerName":"trigger1"}

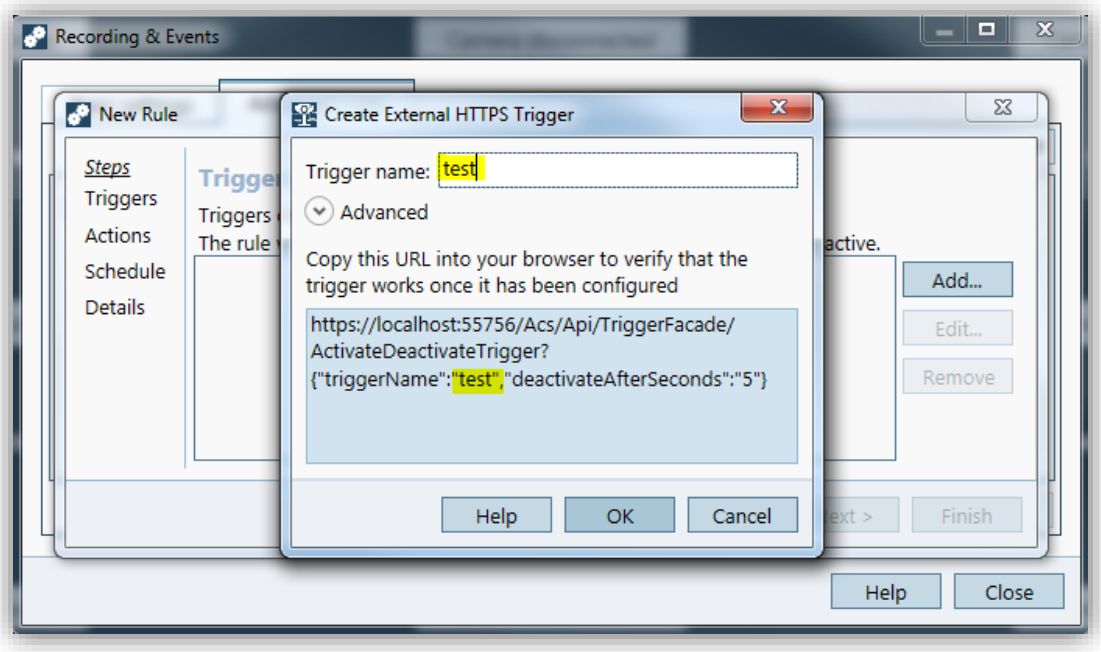
Because of some limitations, it is only possible to test the above commands in Chrome or an external application. It does not work in Edge or Internet Explorer.

Step 1 – Install AXIS Camera Station 5.06 or above

Download and install the latest AXIS Camera Station (5.06 or above) from axis.com.

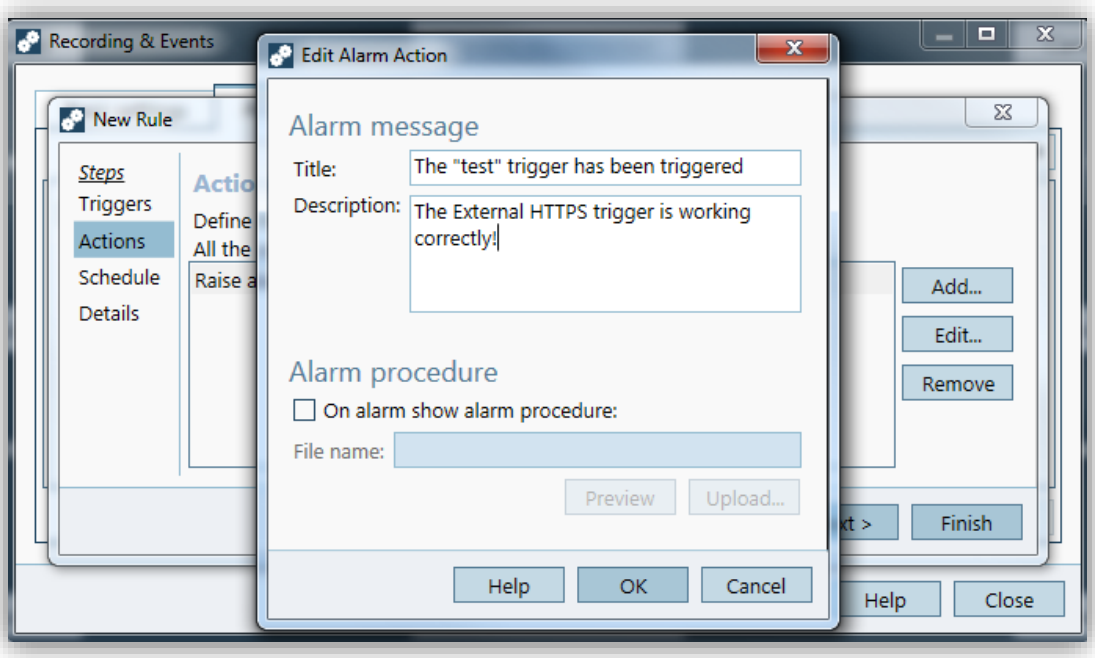
Step 2 – Create a new advanced rule

Go to the **Configuration** menu > **Advanced rules** > **New…**
Add a new trigger: **External HTTPS** and enter any **Trigger name**:



Copy the URL displayed in the lower part of the dialog. It will be needed for testing. In this case:
https://localhost:55756/Acs/Api/TriggerFacade/ActivateDeactivateTrigger?{"triggerName":"test","deactivateAfterSeconds":"5"}

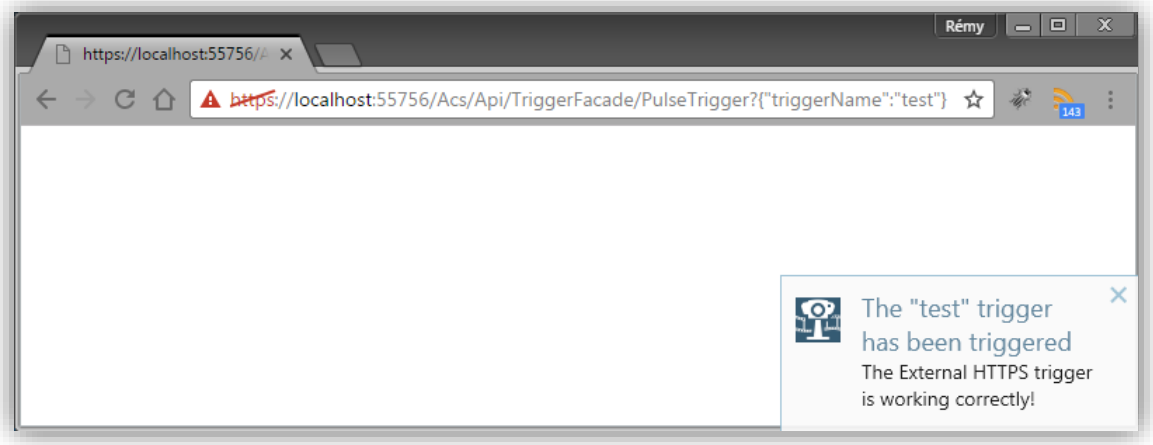
Click **Next** and add a new **Action**. In this case we are raising an **alarm**:



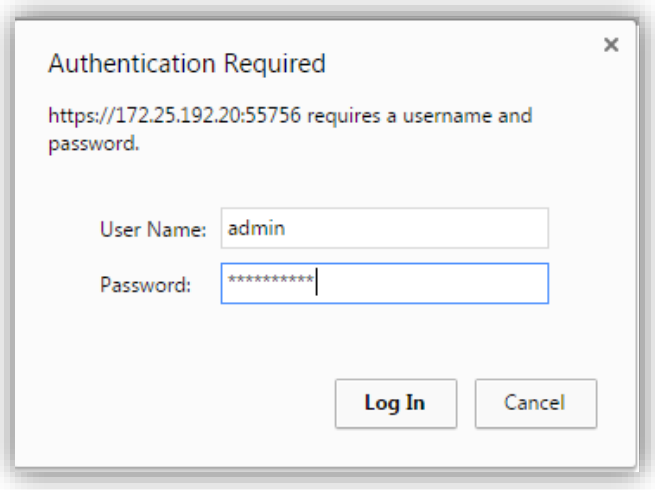
Click **Next** and add a schedule and rule name if you wish. The rule is then available in AXIS Camera Station.

Step 3 – Testing the rule from Google Chrome

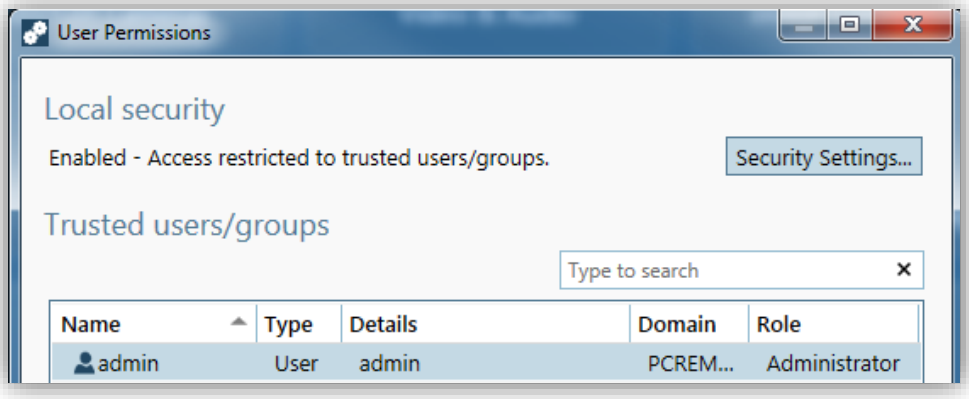
Open Google Chrome and enter the test URL provided by AXIS Camera Station when configuring the rule. The alarm should be displayed:



Note: When testing the command in the web browser on the AXIS Camera Station server itself, the current Windows user will be used and you **may not** be asked to login to send the command. **However, when testing from another computer (or from an external application), an authorized AXIS Camera Station user is required. This is the expected behaviour.**



Such user can be configured in the User Permission dialog of AXIS Camera Station. Any of the user levels (viewer, operator and administrator) are allowed to use External HTTPS triggers.



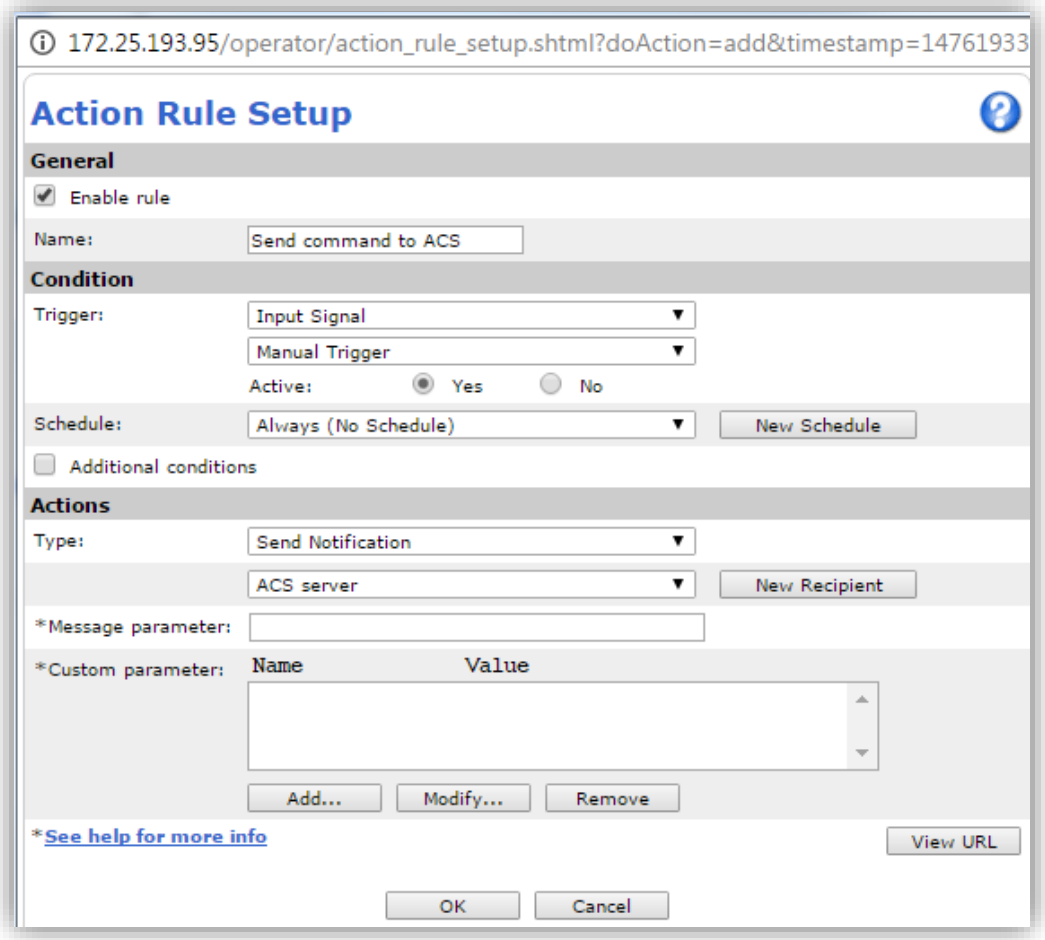
Step 4 – Testing the rule from an Axis Camera

1. Login to the camera's web interface and go to the Events menu to create a new action rule.
2. Add a new Trigger as you wish (Detectors, Input Signal…) and choose action "Send Notification".
3. Create a new recipient with the following values:
   Type: HTTPS
   URL: The complete URL provided by the ACS dialog. The JSON part of the URL needs to be encoded. For instance:

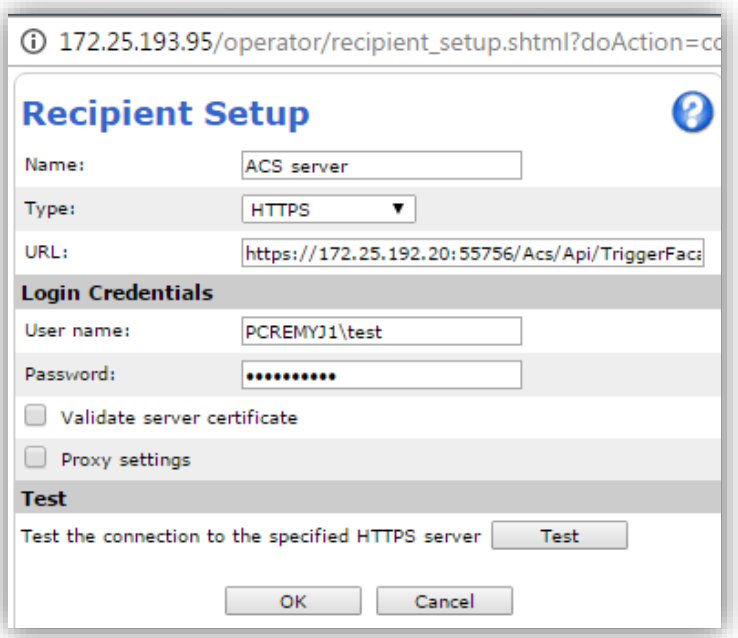
https://172.25.192.20:55756/Acs/Api/TriggerFacade/PulseTrigger?{%22triggerName%22%3A%22test%22}

   Username: Valid ACS username, including domain name.
   Password: Password of the ACS user.
   Validate server certificate: Not selected.

Action Rule configuration:



Recipient configuration:



Step 5 – Testing the rule from an external application

Requirements:

- Support for HTTPS commands (GET or POST method). When using POST the JSON data should be put in the body of the request instead.
- Support for authentication.
- Support for JSON format.