Disclaimer

Legal disclaimer

This guide and its contents (the "Guide") are provided by NXGEN. All rights to the Guide and to the intellectual property embodied in it — including, without limitation, trademarks, trade names, logos and similar marks appearing in or on it — are protected by law. All right, title and interest in the Guide and in any related intellectual property rights belong to and remain with NXGEN and its licensors.

The Guide is provided free of charge, "as is" and for informational purposes only, without warranty of any kind. It is neither intended nor suitable to establish any legal relationship between the reader and NXGEN or its affiliates, including, without limitation, any obligation of NXGEN or its affiliates toward the reader.

The obligations of NXGEN and its affiliates in respect of NXGEN products or services purchased by a customer are governed exclusively by the terms of the agreement under which those products or services were purchased.

For clarification

The reader bears the entire risk as to the use, results and performance of the Guide. To the fullest extent permitted by applicable law, NXGEN disclaims and excludes all warranties, whether statutory, express or implied — including, without limitation, implied warranties of merchantability, fitness for a particular purpose, title and non-infringement of third-party rights — as well as any liability or warranty arising from suggestions, specifications or samples related to the Guide.

Mobotix Alarm Configuration Guide

Updated 3 September 2026

Introduction

This guide describes how to configure a MOBOTIX M15 camera to generate a structured alarm notification and send it to an NXGEN alarm receiver over Raw TCP/IP, using the camera's event system and IP Notify profile. It covers the MOBOTIX web interface configuration, the IP Notify profile (Custom Configuration over Raw TCP/IP), the JSON alarm payload, validating the message at the receiver, and adding the MOBOTIX device to GCXONE.

Integration Overview

The recommended configuration uses the MOBOTIX event engine to trigger an IP Notify action. The IP Notify profile formats the event data as JSON and sends it to an external receiver over a TCP connection initiated by the camera.

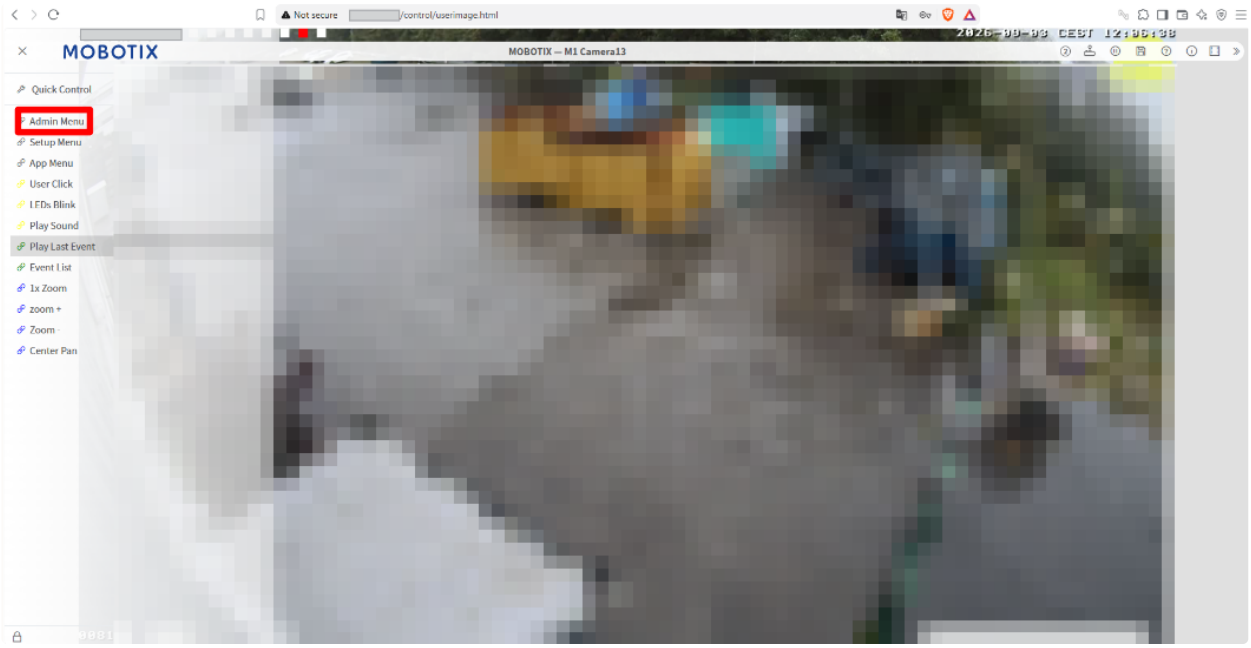
- Event source: a MOBOTIX internal event — sensor, analytics, input, or another configured trigger.
- Event reaction: an Action Group executes the IP Notify action.
- Notification transport: Raw TCP/IP.
- Destination: the external receiver's IP address and TCP port.
- Message format: plain text containing a JSON object populated with MOBOTIX variable expressions.
- Receiver responsibility: accept the TCP message, detect the configured separator, parse the JSON, validate the event, and process it according to the downstream integration.

Flow: MOBOTIX Event → Action Group → IP Notify Profile → Raw TCP/IP → Alarm Receiver → NXGEN / downstream alarm processing.

Open the MOBOTIX Configuration Menus

From the camera's web interface, open theMenu control and enter the administrative configuration area.

Exact menu labels can vary with camera firmware and interface version.



11

Navigate to the event configuration area and locate theAction Group / IP Notify settings. The Action Group determines when a notification is generated; the IP Notify profile determines where and how the notification is transmitted.

Configure the IP Notify Profile

Create or edit an IP Notify profile dedicated to the external alarm receiver.

FieldRecommended valuePurpose / notesIP Notify ProfileA dedicated name, e.g. MultipleNotifyUse a profile dedicated to this integrationIP Notify TypeCustom ConfigurationAllows the alarm payload to be explicitly defined.Destination AddressReceiver IPv4 address and TCP portConfirm the correct endpoint with Customer Support before going live — do not assume a defaultSend OrderParallel send to allUse when multiple destinations are configured and all should receive the messageData ProtocolRaw TCP/IP-Sends the notification as a TCP data stream rather than an HTTP requestSeparator for Raw TCP/IPA unique

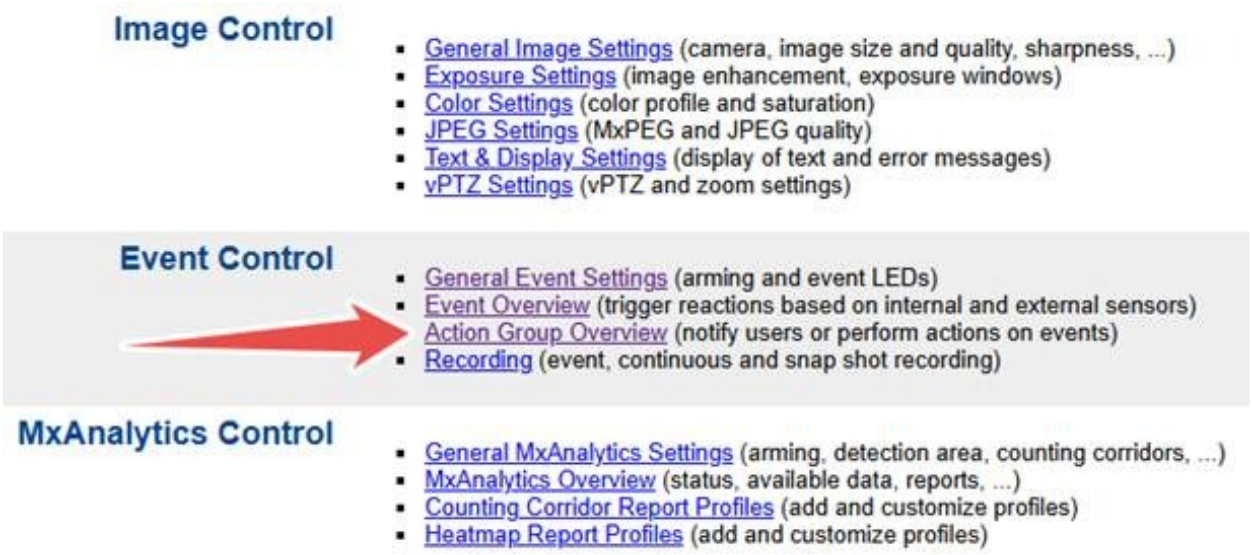
string, e.g. --next-image--The receiver uses this marker to identify message boundaries in the TCP stream. It must not occur inside the JSON body.Data TypePlain textRequired for the JSON payload used in this guide.Send Port0Automatic.

Important: Always confirm the destination IP address and port with Customer Support before configuring a live integration.

14

Configure the MOBOTIX Event / Action Group

The IP Notify profile does not by itself define when an alarm is sent — an event must be connected to an Action Group that invokes it.



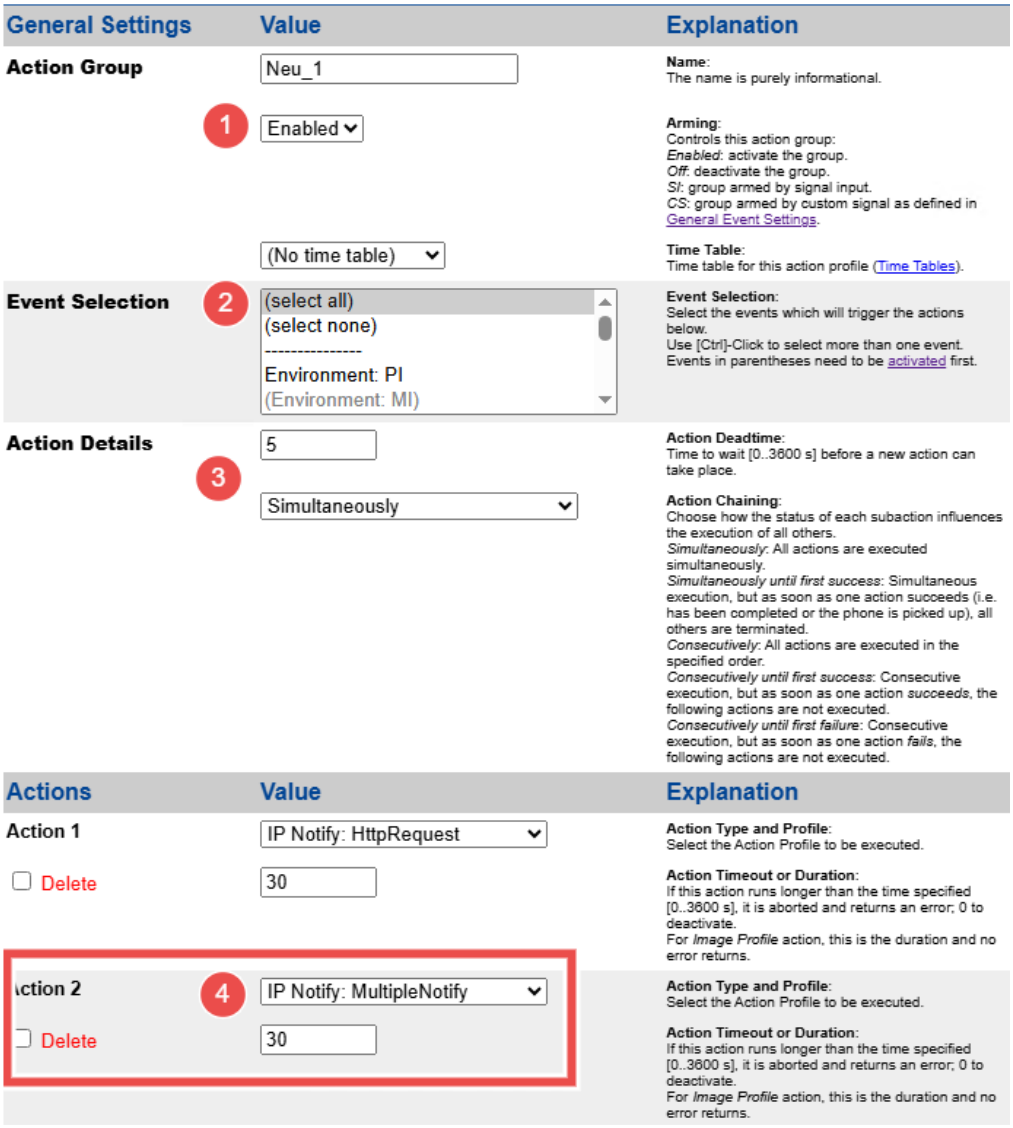
14

Step 1: Enable the new (or existing) Action Group.

Step 2: Select the events that should trigger a TCP alarm notification. Selecting all will notify on every configured camera and system event.

Step 3: Set a deadtime of about 5 seconds between actions, and set actions to trigger simultaneously.

Step 4: Select the IP Notify profile you configured, and set a timeout of about 30 seconds.



45

Recommended configuration sequence:

1. Configure or verify the event source (motion, input activation, an analytics event, or another alarm condition).
2. Create or edit an Action Group for that event.
3. Add the IP Notify action and select the dedicated IP Notify profile.
4. Set any pre-event / post-event behavior the alarm workflow requires.
5. Enable the Action Group and confirm the event is armed and active.
6. Trigger a controlled test event and check the receiver side.

Alarm Message JSON Payload

Use the following plain-text message body in the IP Notify profile. Each `$(...)` expression is a MOBOTIX substitution variable, evaluated when the notification is generated.

```
{
"lasteventTimestamp": "$(LEV.DATE)",
"manufacturer": "$(FPR.PRD)",
"imageNumber": "$(FPR.FRM)",
"eventNumber": "$(FPR.ENO)",
"eventActiveGroups": "$(EVT.AST)",
"eventsEnabled": "$(EVT.EST.SELECTED)",
"eventCode": "$(EVT.EST.ACTIVATED)",
"eventPreList": "$(EVT.SYA)",
"eventPostList": "$(EVT.SYP)",
"eventPreInterval": "$(EVT.SIA)",
"eventPostInterval": "$(EVT.SIP)",
"eventImageType": "$(FPR.IMT)",
"eventTime": "$(FPR.TIMESTAMP)",
"timezone": "$(FPR.TZN)",
"eventTimeUTC": "$(TMS.RFC822)",
"cameraModel": "$(IMG.CTY)",
"videoCodec": "$(IMG.ICC)",
"frameRate": "$(IMG.FRJ)",
"bayerFrameRate": "$(IMG.FRB)",
"imageSizeX": "$(IMG.XTO)",
"imageSizeY": "$(IMG.YTO)",
"imageQuality": "$(IMG.QLT)",
"cameraImageType": "$(IMG.CAM)",
"imageZoomLevel": "$(IMG.ZOM)",
"imageMirrored": "$(IMG.MIR)",
"imageRotated": "$(IMG.ROT)",
"mac": "$(ID.MAC)",
"serial": "$(ID.FIP)",
"hostname": "$(ID.NAM)",
"ipaddress": "$(ID.ET0)",
"cameraSoftwareVersion": "$(ID.SWV)",
"uptime": "$(ID.UPT)",
"timeserverIP": "$(ID.TSI)",
"timeServerConfig": "$(ID.TSP)",
"timeserverOffset": "$(ID.TSO)",
"inputState": "$(SEN.INA)",
"rightCamButton": "$(SEN.BTR)",
"leftCamButton": "$(SEN.BTL)",
"pirValue": "$(SEN.PIR)",
"microphoneValue": "$(SEN.MIC)",
"rightLensBrightness": "$(SEN.ILR)",
"leftLensBrightness": "$(SEN.ILL)",
"triggeredMotionWindows": "$(SEN.VM1)",
"cameraInternalTemp": "$(SEN.TIN.CELSIUS)",
"cameraExternalTemp": "$(SEN.TOU.CELSIUS)",
"externalIOTemp": "$(SEN.TEX.CELSIUS)",
"GPSBoxTemp": "$(SEN.TGP.CELSIUS)",
"rightSensorThermalCenterTemp": "$(SEN.TSR.CELSIUS)",
"leftSensorThermalCenterTemp": "$(SEN.TSL.CELSIUS)",
```

```
"thermalRadiometryTemp": "$(SEN.TTR.CELSIUS)",
"thermalCoordinates": "$(SEN.TCO)",
"ftpDir": "$(TEXT.FTPDIR)",
"ftpEventFile": "$(TEXT.EVENTFILE)",
"storageUtilization": "$(STORAGE.BUFFERFILL.CCURRENT)"
}
```

Receiver-Side Processing

The receiver treats the MOBOTIX connection as a TCP stream. TCP does not preserve application-level message boundaries, so the configured separator must be handled explicitly:

- Accept the incoming TCP connection on the configured port.
- Accumulate bytes until the configured separator (for example `--next-image--`) is detected.
- Extract the message bytes preceding the separator.
- Trim transport whitespace and parse the extracted text as JSON.
- Validate mandatory fields such as `eventTime`, `eventCode`, `ipaddress`, and `serial`, plus any event-related values the downstream system requires.
- Map the normalized event into the alarm model used by the target alarm receiver.
- Persist the original MOBOTIX payload for troubleshooting and audit where practical.

Message framing example:

```
{JSON alarm object}--next-image--{next JSON alarm object}--next-image--
```

Configuration Checklist

- IP Notify profile created and enabled.
- Destination IP and TCP port verified with Customer Support.
- Raw TCP/IP selected as the data protocol.
- A unique separator configured and documented.

- Plain text selected as the data type.
- JSON payload inserted and validated.
- Duplicate JSON property names removed or renamed.
- Action Group configured with the intended event source.
- Action Group invokes the correct IP Notify profile.
- A controlled test event was successfully received and parsed.
- A multiple-event / burst test was completed.
- The alarm is mapped and persisted in the downstream integration.
- MOBOTIX device added to GCXONE and discovered successfully.

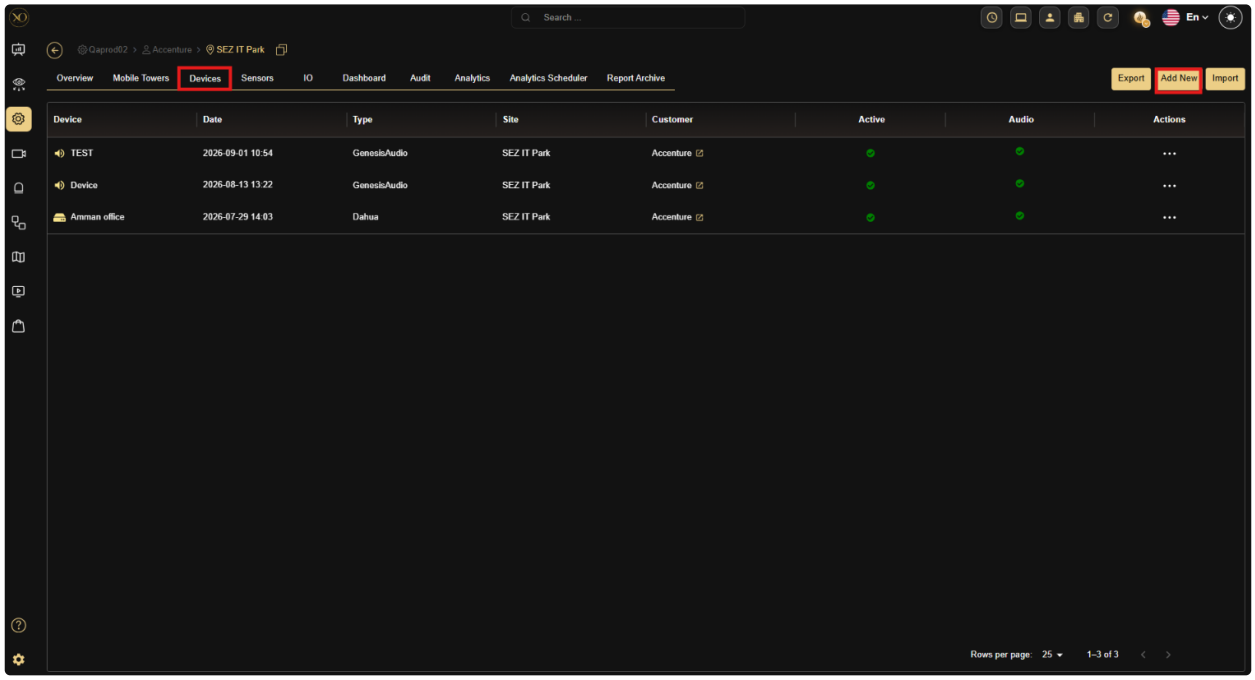
Notes and Firmware Compatibility

- MOBOTIX variable names, and which variables are available, can depend on camera model, firmware version, and event context. Validate the variables on the actual camera firmware before standardizing the payload.
- Configuration menus and labels can vary between firmware versions — the screens referenced in this guide may be reorganized in future MOBOTIX firmware.

Mobotix Config Guide with GCXONE

Step 1: Navigate to the Devices tab.

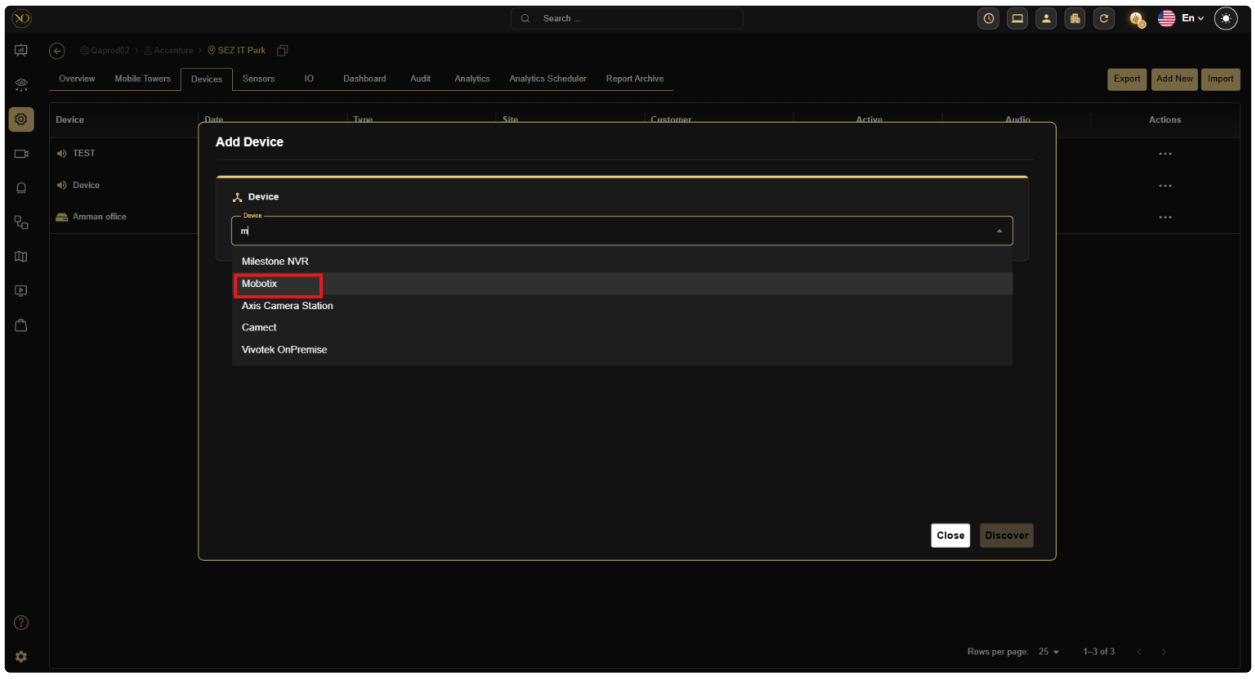
Step 2: Click the Add New button.



14

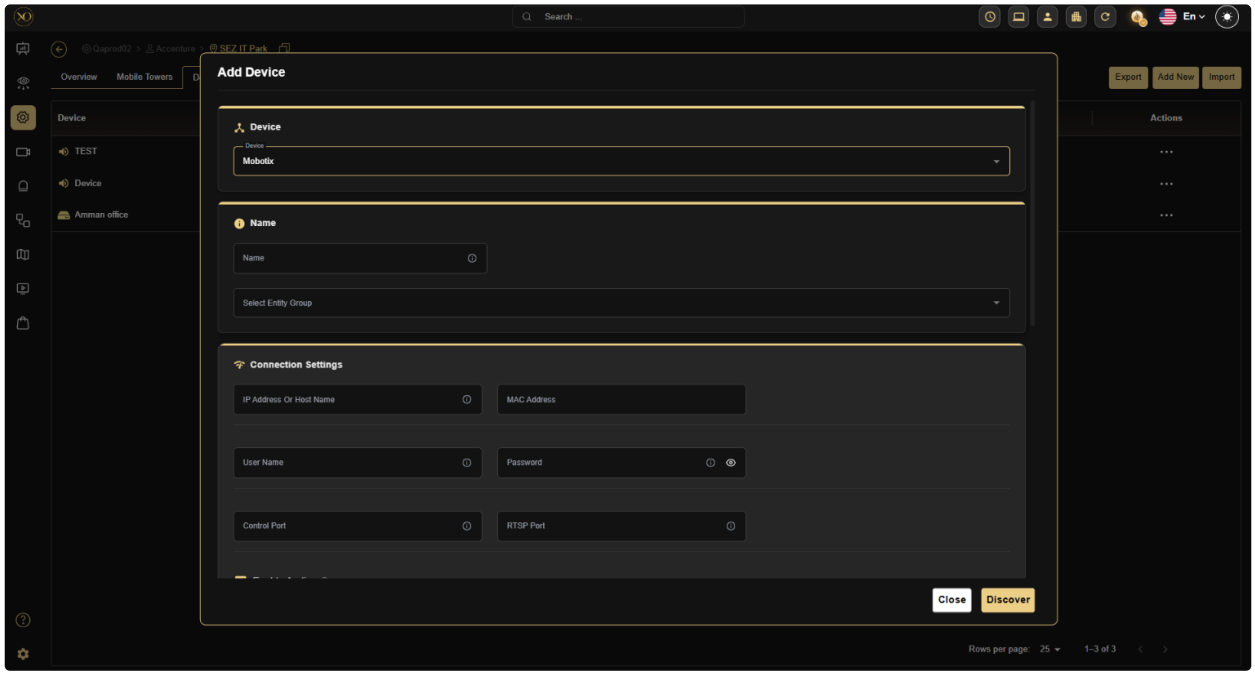
Step 3: The Add Device dialog will be displayed.

Step 4: In the dialog, open the Device dropdown and select Mobotix .



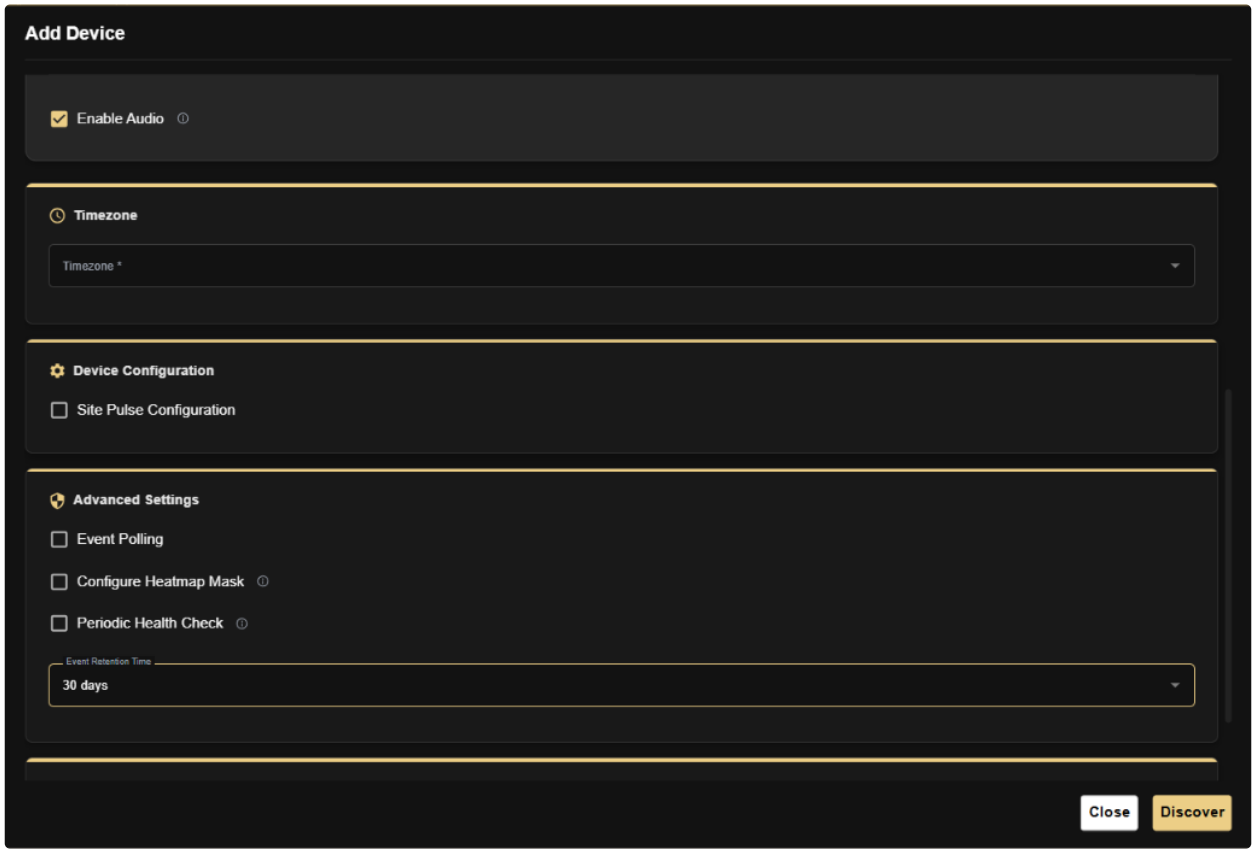
2112

Step 5: Enter all required fields:Name , IP Address or Host Name , MAC Address , User Name , Password , Control Port , RTSP Port .



12112

Step 6: Below the connection fields, the dialog also has a Select Entity Group dropdown, an Enable Audio toggle (on by default), a required Timezone , a Device Configuration section (Site Pulse Configuration), an Advanced Settings section (Event Polling, Configure Heatmap Mask, Periodic Health Check, Event Retention Time — 30 days by default), and an optional Location section (Latitude, Longitude, Plus Code, What3Words).



121

Step 7: Click Discover .

Step 8: Once the device is successfully discovered, click Save.