

AiDANT Markers Client User Guide



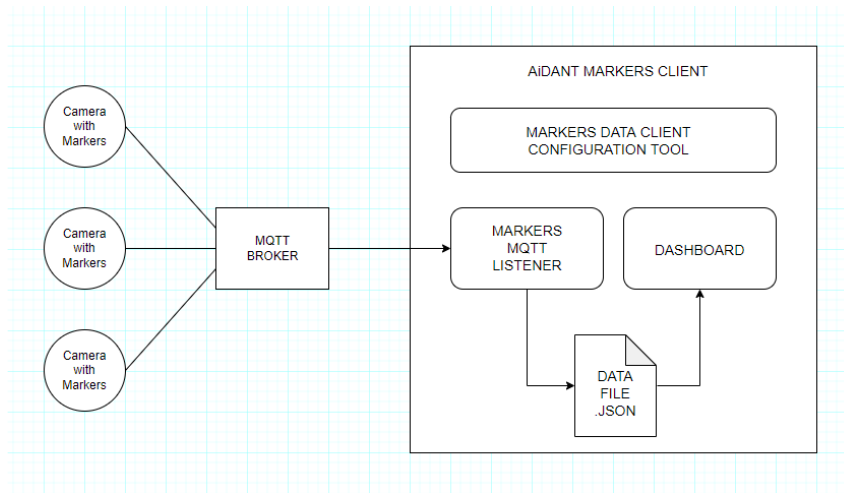
- [1. Overview](#)
- [2. Glossary](#)
- [3. Requirements](#)
- [4. Installation](#)
- [5. Using the Markers Data Client Configuration Tool](#)
 - [Basic Setup](#)
 - [Other Available Controls](#)
- [6. Using the Markers Client Dashboard](#)
 - [Visualizations:](#)
- [7. Appendix](#)
 - [MQTT Broker Host](#)
 - [After installing the Mosquitto broker](#)
 - [Using Mosquitto with Credentials \(optional\)](#)
 - [Troubleshooting](#)

1. Overview [↗](#)

The **AiDANT Markers Client** application developed by [AiDANT.ai](#) is a **Windows** application that can be used to collect data from the **AiDANT Markers (ACAP)** edge application. This application can be installed on any Windows machine.

The Client consists of three components:

1. MQTT Listener.
2. AiDANT Markers Data Client Configuration Tool.
3. Dashboard - with two possible visualizations, plot and/or timeline.



Markers Client Components

2. Glossary [↗](#)

AiDANT MARKERS (ACAP)	An application created by AiDANT Intelligent Technology which is installed inside the AXIS camera and can send MQTT messages when it detects markers.
AiDANT MARKERS CLIENT	A Windows application installed on a computer that can listen to the MQTT events, serialize (save) them and display the information in a dashboard.
ACAP	AXIS Camera Application Platform (ACAP) is the name of applications that are installed inside an AXIS camera.
MQTT	MQTT is a standard messaging protocol for the Internet of Things (IoT). It is designed as an extremely lightweight publish/subscribe messaging transport that is ideal for connecting remote devices with a small code footprint and minimal network bandwidth.
MQTT Broker	The MQTT broker is a central server application that manages all the MQTT messages. It is installed on a central location/server or a desktop computer.
MQTT Client	A client listens to MQTT messages. It is subscribed to the broker.
Status	The status of the Markers ACAP application running on the cameras.
Data File	The JSON data file where the MQTT messages will be written. It is used by the dashboard to display the visualizations.

JSON	JavaScript Object Notation. It's a lightweight format for storing and exchanging data. Think of it as a simple way to organize information that's easy for both humans to read and computers to understand.
Data Events	The information we wish to monitor in our dashboard. A data event has a name, is associated with one camera, and has a payload which can be a numeric value or contain a numeric value. It is represented by a color of your choice.
MQTT Messages	<p>A package of information coming from the camera through the broker. An MQTT message has a topic, a source, a timestamp and a payload.</p> <p>For example:</p> <pre>{ "topic": "axis:CameraApplicationPlatform/AiDANTMarkersApplication/AiDANTMarkersData", "timestamp": 1733266128365, "message": { "source": { "serial": "B8A44F9D7926" }, "key": { }, "data": { "payload": "1" } } }</pre>

3. Requirements

In order to run the AiDANT Markers client application, you will need:

1. A Windows computer.
2. At least one instance of **AiDANT Markers (ACAP)** running on an AXIS camera.
3. The **AiDANT Markers Client** application installer. Download and extract (unzip) the application installer package through the link <https://downloads.aidant.online/>.
4. An MQTT broker running on a host machine. See appendix “**MQTT Broker Host**”.
5. MQTT Messages coming from the camera: Configure the camera to publish the events from an AiDANT Markers (ACAP) as MQTT:
 - a. In the camera user interface, go to *System/MQTT*.
 - b. Select “Connect” to connect to MQTT services.
 - c. Set the IP address of the *Broker Host* as needed. **Save**.
 - d. In the “MQTT Publication” tab Click the + icon to “Add Condition”, selecting **AiDANT Markers Data**. Press **Add**.

Note: If you don't see the condition entry **AiDANT Markers Data** in the list, check that your **AiDANT Markers (ACAP)** is running on the camera.

MQTT client

Connect

Status: Connected

Broker ^

Host

Protocol

Port

Username

Password

Client ID

Clean session

HTTP proxy

HTTPS proxy

Keep alive interval s Timeout s

Reconnect automatically

Device topic prefix

Condition

MQTT topic

Payload topic

Retain

None

Property

All

QoS

0 (at most once)

1 (at least once)

2 (exactly once)

4. Installation [↗](#)

Once all the requirements are satisfied, please do the following:

- Once extracted (unzipped), install the client by running “**setup.exe**”. It will create a shortcut to the configuration tool and the dashboard on your desktop. **Note:** You might need to install the **.NET8.0 Desktop Runtime** framework if you don’t already have it.

5. Using the Markers Data Client Configuration Tool [↗](#)

Basic Setup [↗](#)

Important! This requires an MQTT broker running on a host machine. See appendix “**MQTT Broker Host**”.

- Launch **Markers Client Configuration** from your desktop launcher.



2. Select **File/New Project** to create a new project. Choose a location and a name. The extension will be: *.am1*.
3. Set the location and name of your MQTT data file. Enter the information into the edit box next to the label **MQTT Data File** by pressing the three dots button (...), selecting a folder location and entering the file name.

MQTT Broker

MQTT Broker IP: 192.168.0.40

MQTT Broker Port: 1883

MQTT Data File: D:\Data\data2.json ... Folder

4. Press **Save**. Check your folder to see the *.am1* file is there where you think it is.
5. Set the MQTT broker IP address (if running broker locally, it's the IP of this machine) and port (usually 1883).
6. Add your cameras by pressing **Add**. Enter camera name, IP (eg: **yes** 192.168.0.90, **not** http://192.168.0.90/) and and credentials. Press **Save**.
7. Start the MQTT listener: Press button **Start MQTT Listener**. **Important:** The listener will listen to the configured cameras. **If adding or editing the camera list, the listener must be restarted.**
8. After starting the listener, minimize its command window. This process needs to run in the background for the listener to work.

Other Available Controls [↗](#)

- *Clone*: Clone a selected camera.
- *Edit*: Edit selected camera (you can also double click the selected row).
- *Delete*: Will delete the camera form the project.
- *Collect Marker Status*: Collect the status of the Markers application from any camera.
- *Folder*: Open the selected report folder.
- *Clear*: Clears the log.
- *Open MQTT Data*: Opens the JSON data file.
- *Clear MQTT Data*: Clears the JSON data file.
- *Dashboard*: Launch the dashboard.

6. Using the Markers Client Dashboard [↗](#)

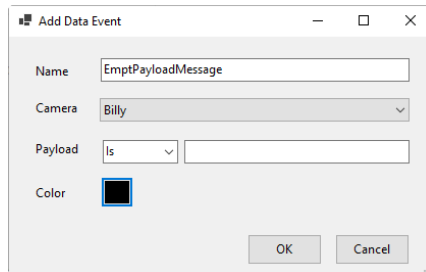
The dashboard can be launched 'in context' from the configuration tool, or launched separately by running the dashboard and opening the project file (.am1) from the **File/Open** menu.

1. In the **Data Client Configuration Tool** Press the **Dashboard** button
2. Create data events. Press the **Add** button to add events to the list.
For example, a data event can be configured like this:
Data Event Name: **Door 1 Open**
Camera: **Cam1**
Logic: **Contains**
Payload: **5**
Color: **Red**

This will add a new data event which will be shown in the visualizations when a payload containing "5" is sent in the MQTT messages.

Important: The MQTT messages **are NOT stateful, they do not have an ON or OFF state.**

In order to mimic statefulness, please remember to always **add the data event with empty payload.** This event can be configured in the data event but hidden in the visualization if needed.



3. The dashboard has the following controls:

File Menu

- Open Project
- Save Project
- Save Project As
- Close Project

View Menu

- Refresh Plot (F5)
- Show/Hide Configuration Panel
- Enable Plot
- Enable Timeline

Data Menu

- Open Data File
- Clear Data File

MQTT Menu

- Start/Stop Listener

Help Menu

- About
- User Guide

4. Once the dashboard is running, you can do the following:

- Set the title.
- Add *Data Events* to monitor.
- Set the colors of the backgrounds
- Set the number of columns in the *Plot*.
- Set the *Timeline* timeframe.
- Set the autorefresh rate.
- View and clear the data file.
- Start and stop the MQTT listener.

5. If you wish to view a **different data file** from the dashboard:

- Open the project file using the **AiDANT Markers Data Client Configuration Tool**.
- Select a **different data file** to associate with the project. **Save**.
- Launch the dashboard in context by pressing the **Dashboard** button in the user interface of the configuration tool.

Visualizations: [↗](#)

1. The *Plot* visualization plots the messages available in the data file one by one.
2. The *Timeline* visualization graphs the messages relative to the timeframe selected.

Choose to view either one or both using the **View** menu item.

7. Appendix [↗](#)

MQTT Broker Host [↗](#)

Deploy an MQTT broker in a known host location accessible to the application. An open source MQTT called “**Mosquitto**” can be found here: <https://mosquitto.org/download/>

After installing the Mosquitto broker [↗](#)

In the installation folder, eg: “C:\Program Files\mosquitto”

1. Create a config file called **myconfig.conf**.
The content should be:
listener 1883
allow_anonymous true
2. Then open a command window.
Start server like this **mosquitto -c myconfig.conf -v**

Using Mosquitto with Credentials (optional) [↗](#)

Setting up Mosquitto with credentials involves a few steps. Here's a breakdown of how to do it:

1. Create a Password File:

- Mosquitto uses a password file to store usernames and hashed passwords. You can create this file using the `mosquitto_passwd` utility.
- Open your terminal and use the following command to create a new password file (e.g., `mosquitto.passwd`):

```
1 mosquitto_passwd -c mosquitto.passwd <username>
```

- Replace `<username>` with the desired username. You'll be prompted to enter and confirm the password.
- To add more users to the same file, omit the `-c` option:

```
1 mosquitto_passwd mosquitto.passwd <username>
```

2. Configure Mosquitto:

- Open the Mosquitto configuration file (usually located at `/etc/mosquitto/mosquitto.conf`), in our case **myconfig.conf**.
- Add the following lines to enable password authentication and specify the password file:

```
1 allow_anonymous false
2 password_file /path/to/mosquitto.passwd
3
```

- Replace `/path/to/mosquitto.passwd` with the actual path to your password file.

3. Restart Mosquitto:

- Restart the Mosquitto service to apply the changes:

```
1 mosquitto -c myconfig.conf -v
```

Explanation:

- `allow_anonymous false` disables anonymous connections, forcing clients to authenticate.
- `password_file /path/to/mosquitto.passwd` tells Mosquitto where to find the password file.

Troubleshooting [↗](#)

Q: Messages don't reach dashboard.

Here are a few things you can try:

- Try to restart the MQTT listener.
- Check if the messages are written to the data file, if so, refresh the dashboard (F5) or turn on auto refresh.
- Restart the MQTT host.
- Check that the camera is sending messages by looking at the host terminal window.
- Reconnect the camera to MQTT
- Check the setting for the MQTT condition in the camera,
- Make sure **Markers ACAP** is running in the camera,
- Make sure the area of interest is set correctly in the Markers ACAP,
- Make sure the markers are detected,
- Make sure the dictionary used is the same as the printed markers.

Copyright [Aidant.ai](#) 2025: This application and documentation, including any attachments, is for the sole use of the intended client. Any unauthorized use or distribution is prohibited. Please contact info@aidant.ai for additional information.