MIKRODEV®
Akıllı Kontrolde Teknoloji Devi

ViewPLUS SCADA Programming Manual

DOCUMENT NAME	DATE	VERSION
MIKRODEV_SM_ViewPLUS_Scada_EN v1.0	06 / 2025	ViewPLUS 1.0.9 (Official Build)

CONTENTS

About Mikrodev

Since 2006, MIKRODEV has been developing and manufacturing industrial control and communication products. MIKRODEV serves the system integrators in the public and private sector, OEM and end users.

Our products are manufactured complying with the quality standards required by the industrial automation industry and the quality of our products are proved on the field for many years

MIKRODEV is one of the few companies in the world that has its own designed IEC 61131-3 compliant library for its programmable logic control devices. In addition, the open, flexible, programmable SCADA solution developed by MIKRODEV is also available to customers.

MIKRODEV products' performance and wide range of applications make them possible for customers to achieve faster, simplified and cost-effective results.

WARNING!

- ✓ Use the programming editor only for Mikrodev Certifed devices
- ✓ When you change your physical hardware configuration, update your development to the appropriate version.
- ✓ The developed program should be tested separately before taking to field service and should be shipped to the field after the tests are successfully completed.
- ✓ Take all accident prevention measures and safety measures identified by local law

Failure to comply with these rules may result in death, serious injury or property damage

1 ViewPLUS SCADA

The term SCADA is an abbreviation formed from the initials of “Supervisory Control and Data Acquisition.” SCADA is a comprehensive and integrated database-based control and monitoring system that enables automatic control and supervision of all electronic units in a facility or plant, and reporting of results. Essentially, SCADA software is expected to perform functions such as monitoring, control, data acquisition, data recording, and storage.

SCADA systems can operate continuously, intermittently, repeatedly, or in discrete modes in industrial processes, manufacturing, production, power generation, and refineries. Infrastructure operations may include water treatment and pumping stations, wastewater treatment, oil and gas pipelines, power transmission and distribution, wind turbines, civil defense siren systems, and large communication systems in both public and private sectors. SCADA systems may also be used in facilities like plant buildings, airports, ships, and space stations. Monitoring and controlling access and energy consumption may be necessary in heating and ventilation systems (HVAC). Due to the benefits, safety, and convenience it provides, SCADA will continue to be one of the most essential needs of industrial facilities.

While developing the ViewPLUS SCADA software, stability, ease of use, and visual appeal were prioritized. With ViewPLUS SCADA, all kinds of automation units in the field can be visually monitored, controlled, and evaluated.

Extensive Visual Library

- In addition to a broad visual library, it allows users to easily create their own SCADA components.

Redundant Working Mode

- Data from the field is read and processed by both SCADA systems simultaneously
- preventing data loss
- Communication status tests (field devices or backup SCADA) can be performed.

Server/Client Architecture

- Supports multiple workstations with assignable authority levels.

Tag Capacity Based on License

- Number of alarms and trends can be defined according to the tag count.

Operating System Support:

- Windows 7/8/10/11
- Windows Server 2008/2012/2016/2019/2022
- Linux (Debian)
- MacOSX.

Security

- 128 different access permissions for tags and pages
- 128 user group assignments, OS-integrated security
- TLS/SSL communication protocols
- salted SHA256 hashes instead of plain text passwords.

Data Transfer

- Compatible with SQL for data import/export.

Alarm

- Define alarms for all tags, 256 different criticality levels, monitor active alarms and alarm history, filter by date and importance, export to Excel, printer, or PDF.

Data Logging and Trend Monitoring

- Logging for all tags, graphical display for trend-defined tags, export to Excel, PDF, or printer.

Communication

- Excellent connection with Mikrodev PLC, REMOTE IO, and Gateway products. Compatible with many industry-standard protocols: MODBUS TCP, MODBUS RTU, DNP3, SNMP, IEC104, BACNET.

Database Support

- PgSQL
- ORACLE
- SQLite.

1.1 ViewPLUS SCADA PC Requirements

For optimal performance of the ViewPLUS SCADA software, the required PC specifications are as follows:

SCADA PC Requirements	
Processor	Intel(R) Xeon(R) CPU E7-4870 @ 2.40 GHz (8 cores)
Installed RAM	32.0 GB
System Type	64-bit OS, x64-based processor
Storage	1 TB SSD

Note: One important point is that the processor must have 8 cores; it does not necessarily need to be a Xeon processor.

In addition to SCADA server requirements, the server must have a Static IP. To enable client connections from devices outside the network, port forwarding must be configured for the default port 3344.

1.2 Architecture

The ViewPLUS SCADA server connects to field devices via intranet or internet, enabling real-time monitoring and control. Collected data is recorded in tag, event, and alarm databases. Using the configured SCADA project, it allows remote users to monitor and control field devices via internet connection to the server. While various configurations are possible depending on the designed network and hardware topology, the basic architecture is as illustrated below.

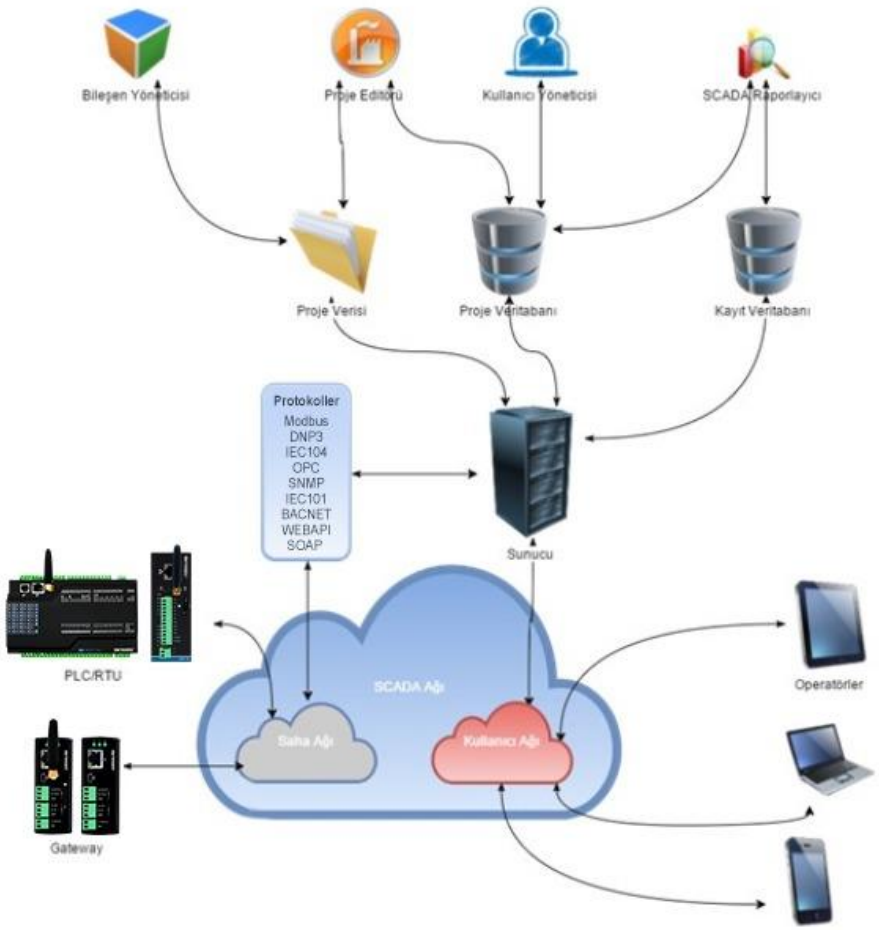

Figure 1 SCADA Architecture

2 SCADA Database

2.1 Installing PostgreSQL 13

PostgreSQL is a powerful, open-source database management system. To install version 13 of PostgreSQL, follow the steps below. This guide includes installation steps for Windows, macOS, and Linux (Ubuntu) operating systems.

2.1.1 PostgreSQL 13 Installation for Windows

- Visit the official PostgreSQL website and download version 13 for Windows.
- Run the downloaded installer and launch the setup wizard.
- Follow the steps in the installation wizard:

Start: Click “Next” to begin.

Installation Directory: Choose the directory for installation (default is C:\Program Files\PostgreSQL\13).

Components: Select all components (PostgreSQL Server, pgAdmin 4, Stack Builder, etc.).

Data Directory: Choose a directory for data storage.

Password: Set a password for the PostgreSQL superuser (postgres).

Port: Default port number is 5432. Proceed without changing.

Locale Settings: Select the default locale settings.

Completion: Click “Next” and then “Finish” to complete installation.

- After installation, verify success by connecting through pgAdmin 4 or command line.

2.1.2 PostgreSQL 13 Installation for macOS

Use Homebrew to install PostgreSQL 13 by executing the following commands in Terminal:

brew install postgresql@13

- Start PostgreSQL services:

brew services start postgresql@13

- Verify installation using the command `psql postgres`. If connection is successful, the installation is complete.

2.1.3 PostgreSQL 13 Installation for Linux (Ubuntu)

- Add the official PostgreSQL repository:

sudo sh -c 'echo "deb http://apt.postgresql.org/pub/repos/apt/ $(lsb_release -cs)-pgdg main" > /etc/apt/sources.list.d/pgdg.list'

- Add the APT key:

wget -qO - https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key add -

- Update package list:

sudo apt-get update

- Install PostgreSQL 13:

sudo apt-get install postgresql-13

- Start PostgreSQL service:

sudo systemctl start postgresql

- Verify installation by connecting to PostgreSQL:

sudo -u postgres psql

2.1.4 PostgreSQL Configuration

After installation, you can configure the database by editing the configuration files. Main configuration files include:

- postgresql.conf: General PostgreSQL settings
- pg_hba.conf: Host-based authentication settings

These files are typically located in `/etc/postgresql/13/main/` on Ubuntu or in the installation directory on Windows/macOS.

2.2 Database Tables

ViewPLUS SCADA uses various database tables to manage, log, and report SCADA projects. For PostgreSQL, tables are organized under the "public" and "logs" schemas.

2.2.1 Public Schema Tables

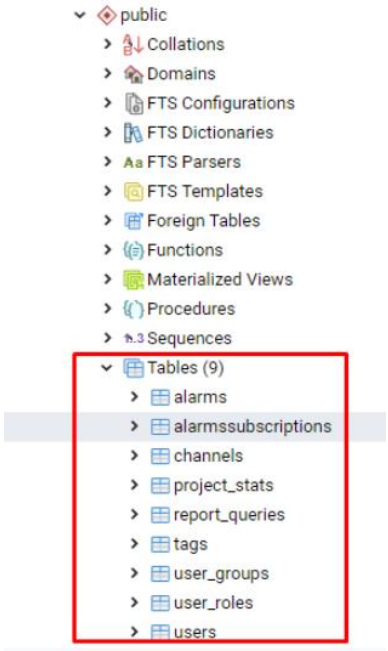

Figure 2 Public Database Tables

- **Alarms:** Contains alarm definitions created via the SCADA editor.
- **Channels:** Contains connection information for field devices.
- **Tags:** Used to define tag details in the system.
- **Users:** List of authorized users for the project.

2.2.2 Logs Schema Tables

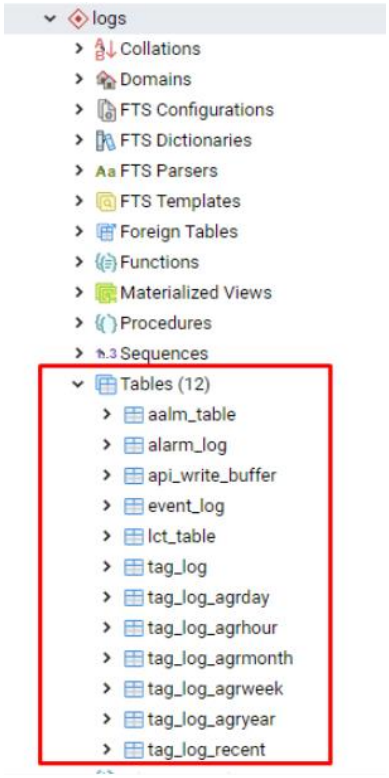

Figure 3 Logs Database Tables

- **alarm_table:** Stores real-time active alarm information.
- **alarm_log:** Stores historical alarm data.
- **event_log:** Contains user login, request, and event data.
- **lcd_table:** Latest value and timestamp for system tags from field devices.
- **tag_log:** Raw data collected from the field; this table can grow rapidly.
- **tag_log_agrhour:** Hourly statistics for system tags, updated every minute.
- **tag_log_agrday:** Daily statistics for system tags, updated every minute.
- **tag_log_agrweek:** Weekly statistics for system tags, updated every minute.
- **tag_log_agrmonth:** Monthly statistics for system tags, updated every minute.
- **tag_log_agryear:** Yearly statistics for system tags, updated every minute.

3 Creating a New Project

Follow the steps below to create a new project:

- Click File > New File or Project > Mikrodev SCA Project.

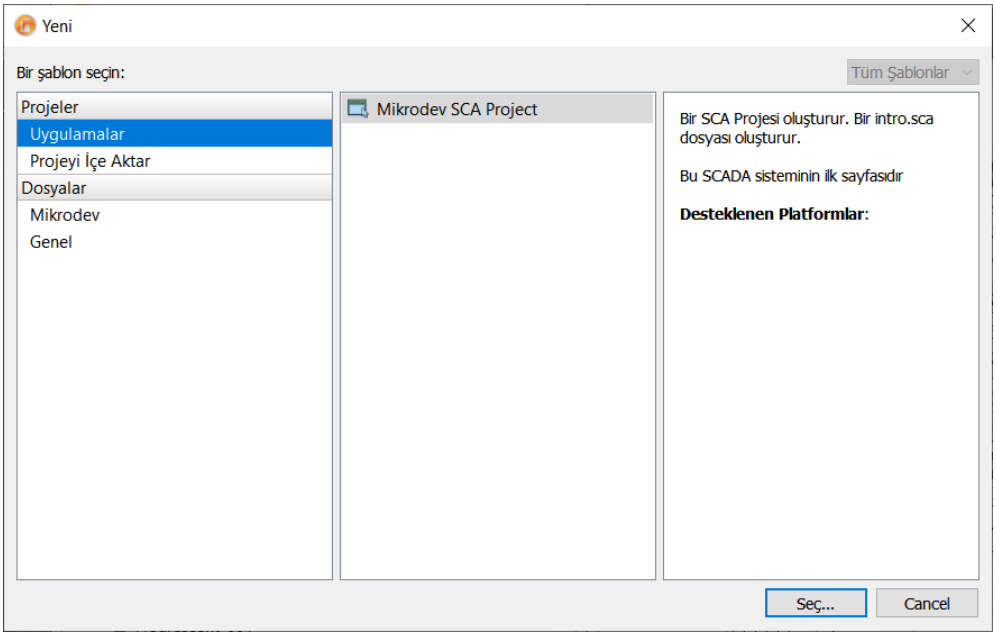

Figure 4 Creating a New Project

- Project Name and Location:

Enter the project name and select the location where it will be saved.

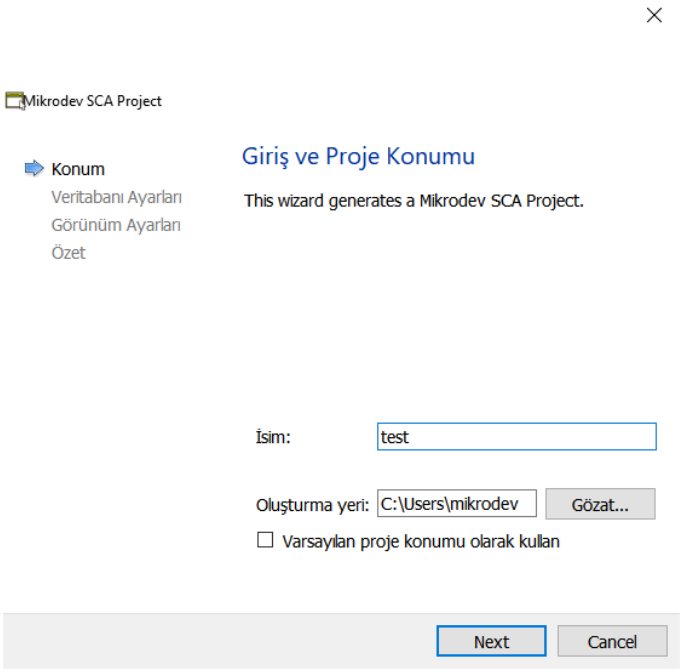

Figure 5 Project Location

- Database Settings:

After entering the database username and password, click the “Test Connection” button to verify the password. If the PostgreSQL database is not installed on your system, click the “Download PgSQL” link to download and install the required version.

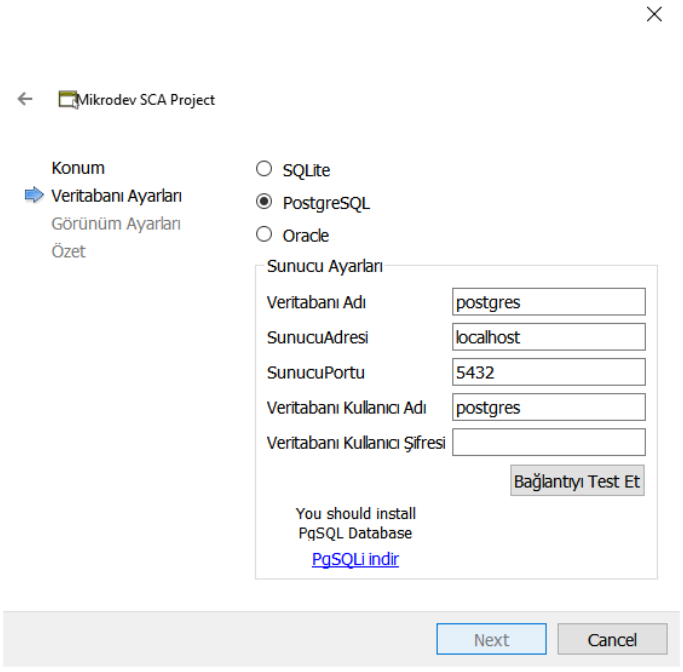

Figure 6 Database Selection

- Visual Settings:

Adjust visual parameters such as screen color and default page size.

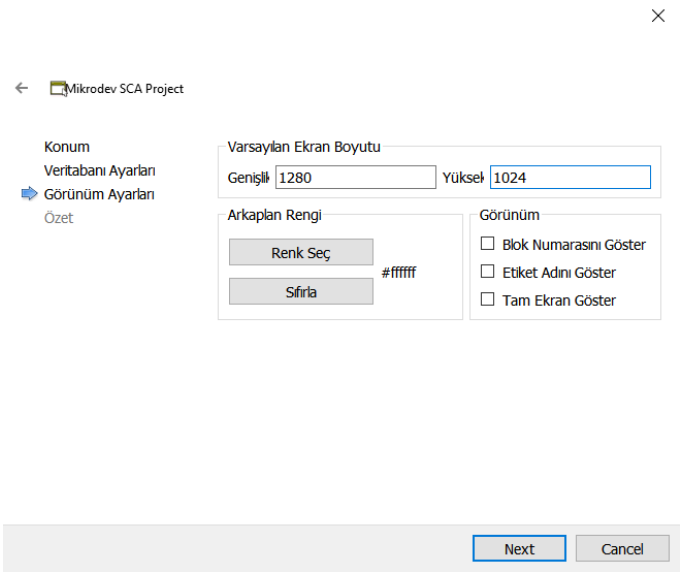

Figure 7 Visual Settings

- Version Control Settings:

On the next page, if you plan to use the SVN version control tool, configure the necessary settings.

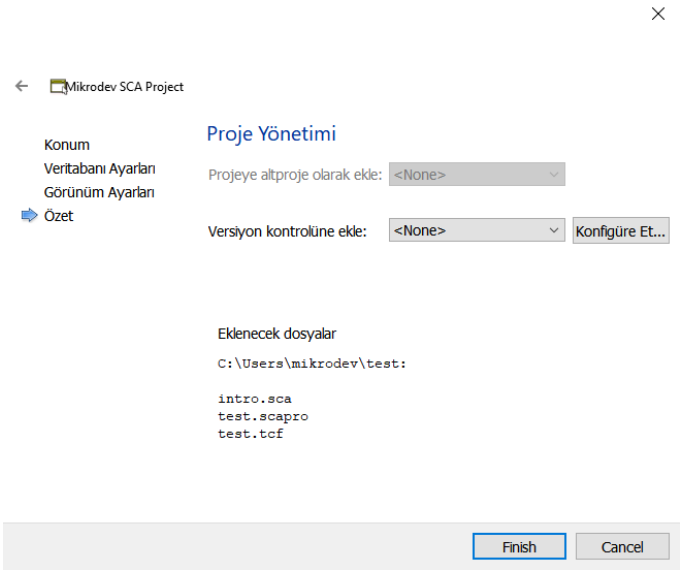

Figure 8 Version Control

- Completing the Project:

Click the “Finish” button to create the new project. The newly created project consists of two files:

- A file with the extension “.scapro” that stores the project configuration parameters. This file generally does not require modification.

- A file named “intro.sca” where you can design the main screen used in the SCADA system.

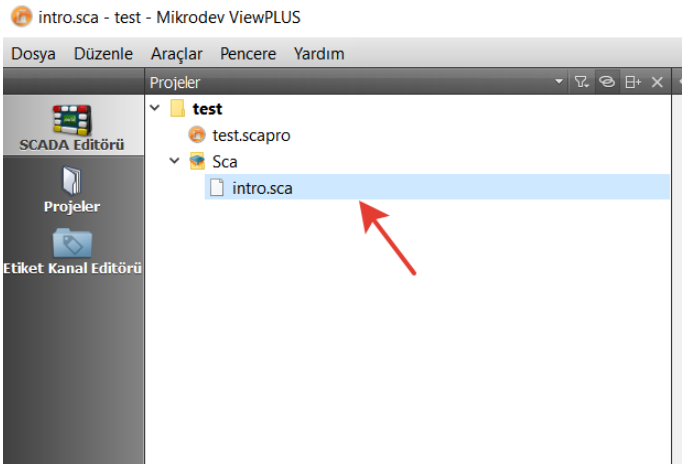

Figure 9 Project Files

By following these steps, you can create a new SCADA project. After creating your project, you can make adjustments and add components based on your SCADA system's needs.

4 Scada Editor Interface

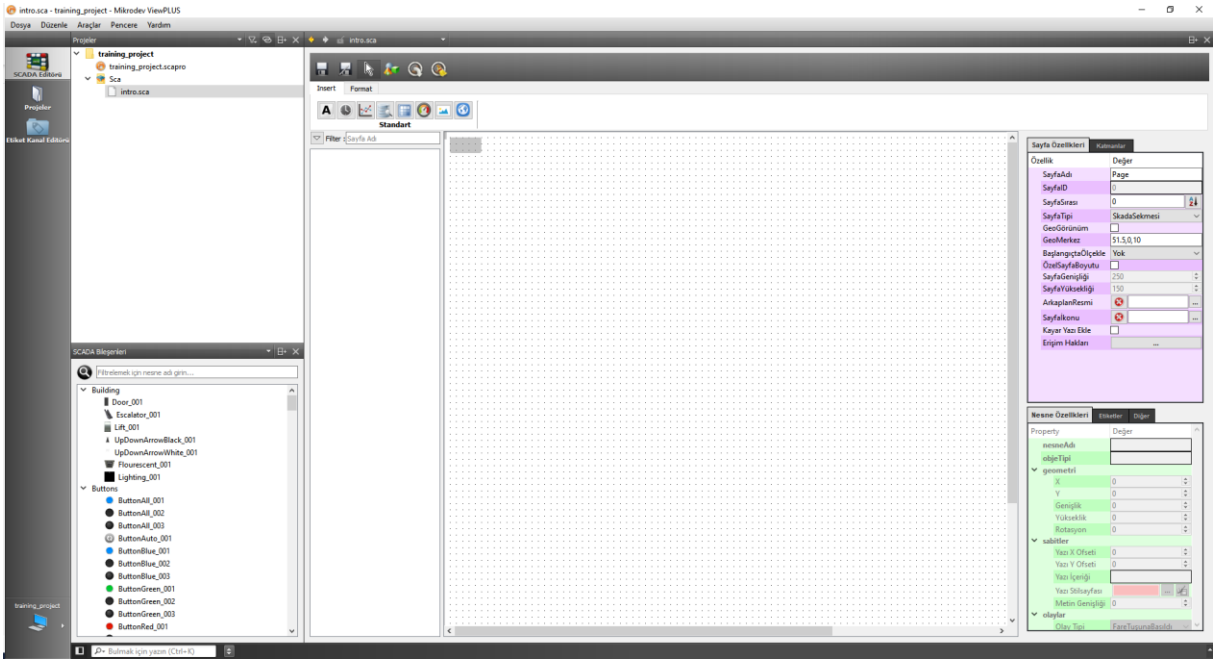

Figure 10 ViewPLUS SCADA Interface

4.1 Sidebar

The sidebar plays an important role in the SCADA Editor view. This panel is used to browse projects, files, and add components.

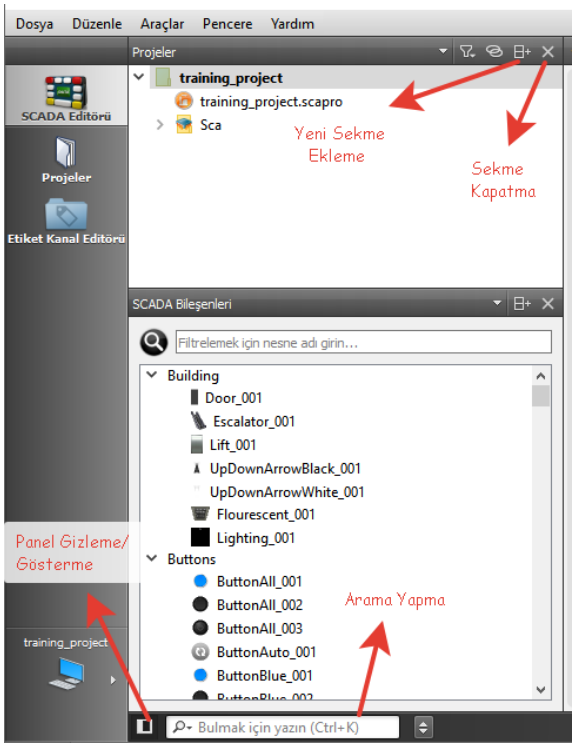

Figure 11 Sidebar Features

- **Add New Pane:** On the left side of the editor, panes allow you to control project files and components. These provide useful tools for managing your projects and components.
- **Close Pane:** You can close unused panes to make the workspace more efficient. This helps clean up the workspace by hiding unnecessary panes.
- **Search:** The search box at the bottom of the sidebar allows you to search current project files, open diagrams, etc. This is useful for quickly locating specific files in large projects.
- **Hide/Show Panels:** A button at the bottom-left of the panel allows hiding or showing all panels, providing flexibility in organizing the workspace.

Note: If the sidebar is closed, it can be reopened via the window menu using the "Show Sidebar" option.

4.1.1 SCADA Components Panel

This section displays the list and categories of components to be used in SCADA projects.

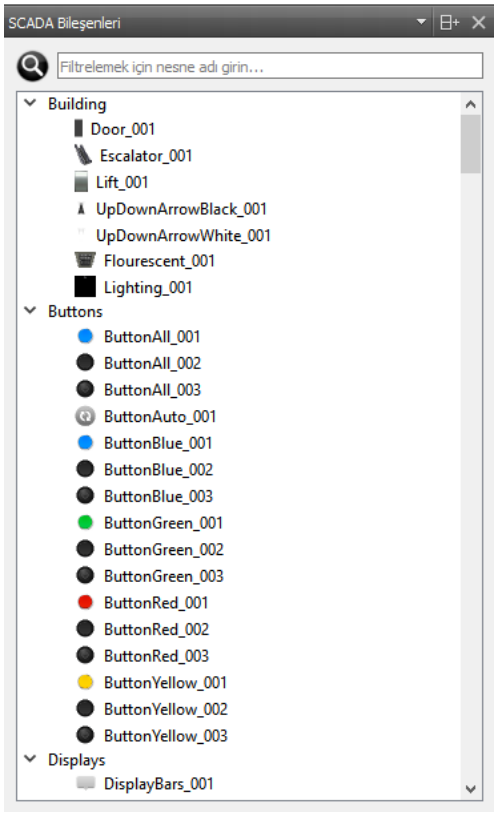

Figure 12 SCADA Components Panel

- **Categories:** SCADA components are grouped under different categories, such as "Building", "Buttons", and "Lighting".
- **Components:** Each category lists related components, which can be added to the project using drag-and-drop.
- **Filtering:** You can use the search box at the top to filter components and quickly find what you're looking for.

4.1.2 Projects Pane

This section displays the existing SCADA projects and the files included in these projects.

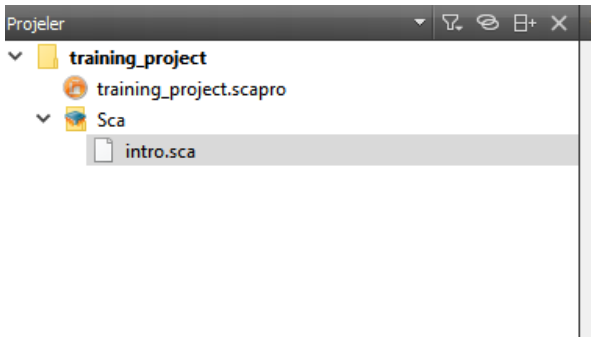

Figure 13 Projects Panel

- **Project List:** Displays a list of your current projects. Each project contains project files and pages.
- **Project Files:** Shows the files under the selected project, such as .scapro and .sca files.
- **File Operations:** Right-click on project files to add a new file, delete, or edit existing files.

4.1.3 Open Diagrams Panel

This section displays the currently open diagrams in your projects.

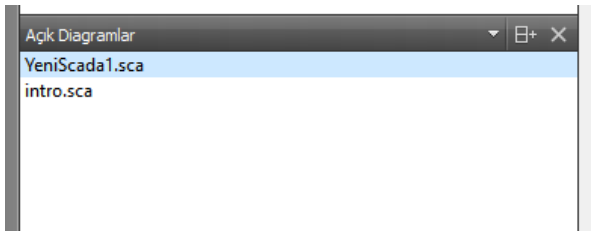

Figure 14 Open Diagrams Panel

- **Open Diagrams List:** You can see which diagrams are currently open and quickly switch between them.
- **Diagram Management:** You can close open diagrams or perform operations on them.

4.1.4 File System Pane

This section provides access to your computer's file system.

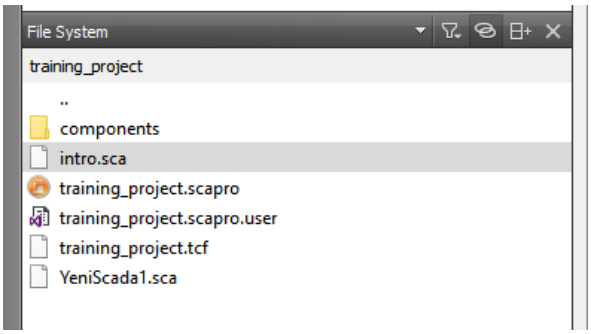

Figure 15 File System Pane

- **File and Folder Access:** Access files and folders on your computer and manage your project files.

4.2 Main Workspace

The main workspace is the area where you can design SCADA pages and place components.

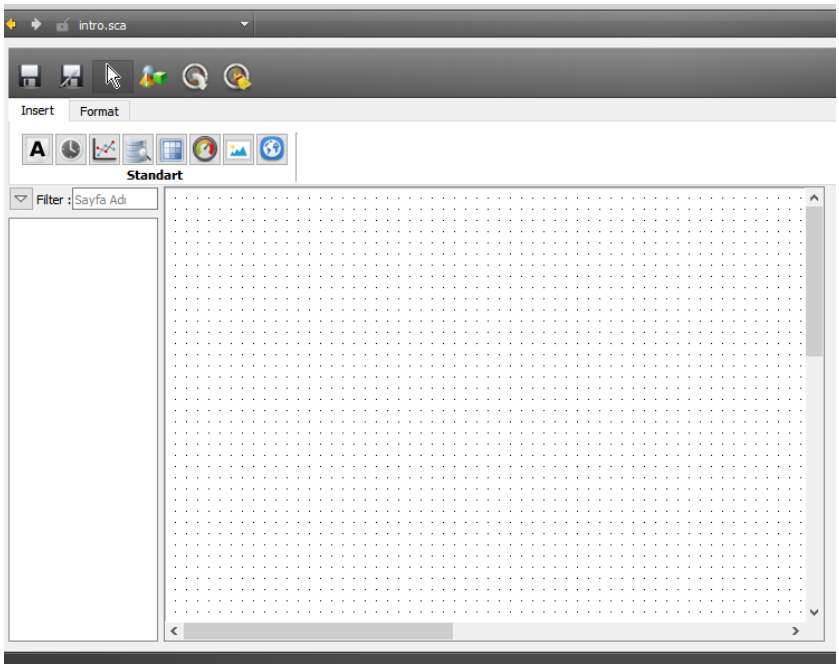

Figure 16 Main Workspace

- **Page Editing:** You can add new components, move existing ones, and make edits.
- **Page Shortcuts:** Use the toolbar at the top to perform page editing actions (e.g., save, save as, auto-select last added item).

4.2.1 Page Shortcuts

Shortcut tools provide essential features to help you use the SCADA editor more efficiently. Their functions and usage are explained below.

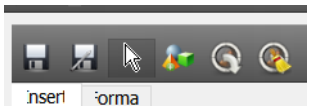

Figure 17 Page Shortcuts

- **Save:** Click this icon to save changes made to the project. This overwrites the current files and stores all changes.
- **Save As:** Click this icon to save changes as a new file, preserving the original files.
- **Pointer Mode:** Click this icon to exit edit mode and enter pointer mode. This allows you to select and move objects in the workspace.
- **Last Added Component:** Click this icon to quickly find the last added component on the page. This is helpful for locating newly added items in large projects.
- **Change Labels:** Click this icon to change the label associations of components with similar label structures on the page. This allows for quick updates of specific labels.

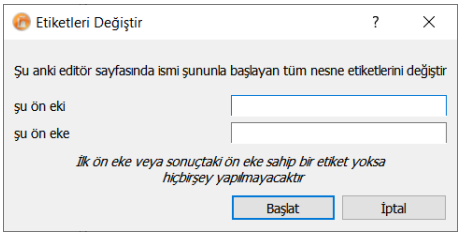

Figure 18 Change Labels Screen

When performing automatic label changes, it is important that labels associated with similar components have a similar structure.
Example for an energy analyzer page:
Cell H01
H01_FazL12VoltageInfo
H01_FazL1CurrentInfo
H01_FazL1VoltageInfo

Cell H02
H02_FazL12VoltageInfo
H02_FazL1CurrentInfo
H02_FazL1VoltageInfo

If H01 page is to be adapted to H02, select the "Change Label" shortcut.

Current Prefix: Enter the variable part of labels currently on the page (e.g., H01).
New Prefix: Enter the variable part of labels to be reassigned (e.g., H02).

- **Clear Invalid Labels:** This option clears invalid labels from the page.

4.3 Page Properties Panel

This panel allows configuring various settings of the selected page.

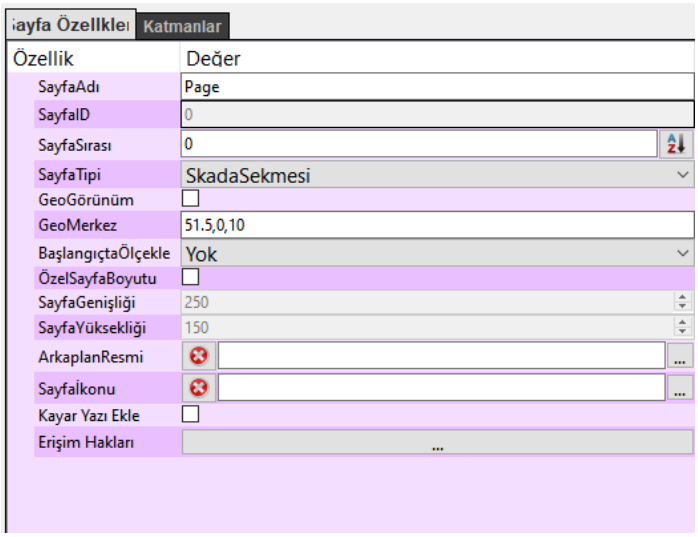

Figure 19 Page Properties Panel

- **Page Name:** Set or edit the name of the page.
- **Page ID:** Each page has a unique, automatically assigned ID number.
- **Page Order:** Determines the order of pages (with type “SCADA Tab”) on the Client screen. The topmost page acts as the homepage. Click the AZ icon in the "Page Order" section to reorder pages using the left arrows in the opened dialog.

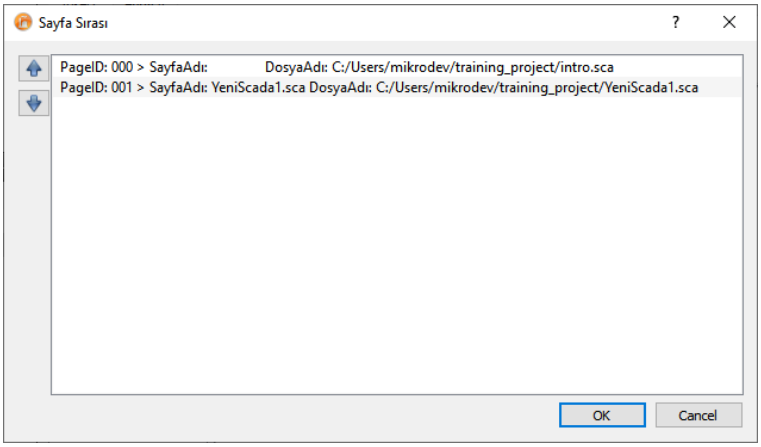

Figure 20 SCADA Pages Ordering

- **Page Type:** There are 3 page types in SCADA software:

SCADA Tab: Visible in the left tab section of the Client screen.

SCADA Dialog: Popup-type pages not visible in the left tab section.

SCADA Linked Page: Not in the tab section; opened via navigation from another page.

- **Geo View:** If checked, the page will be used as a map page.
- **Geo Center:** Enter the coordinates to focus on when the map page is opened.
- **Initial Scaling:** Choose the scaling type to apply when the page is opened.

Visible / Width / Height: Controls how the SCADA page will be shown in the Client screen based on width, height, or full visibility.

- **Custom Page Size:** Enable this option to set custom dimensions for the SCADA page.

Page Width / Height: These become active when "Custom Page Size" is selected.

- **Background Image:** Add a background image to the SCADA page.
- **Page Icon:** Assign an icon for SCADA Tab pages to display in the left sidebar.
- **Scrolling Text:** Enable this option to add scrolling text to the SCADA page.
- **Access Rights:** Restrict page visibility to users with assigned permissions.

4.4 Layers Panel

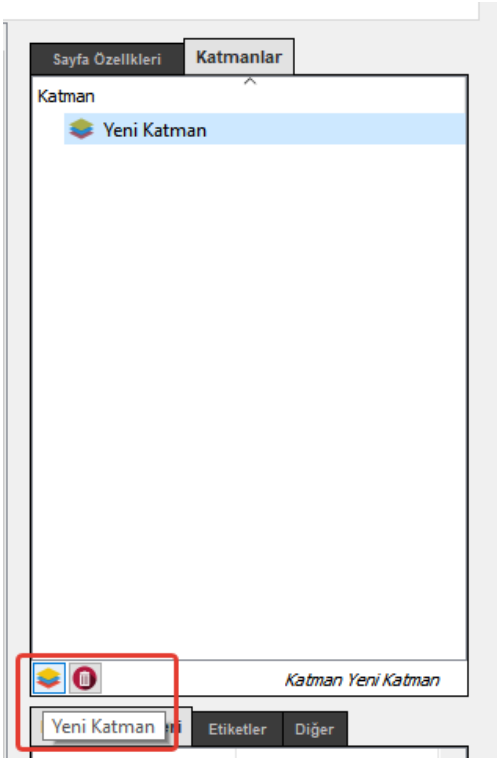

Figure 21 Creating a New Layer

You can create or delete layers on the Layers panel. By placing components on the created layers, you can establish a hierarchical structure on the page. You can hide/show layers to more easily manage complex designs. Another benefit of layers is the ability to change their visibility depending on different zoom levels.

- **Hide/Show Layers:** You can hide or show layers you've added using the "Hide/Show Layer" option from the right-click menu.

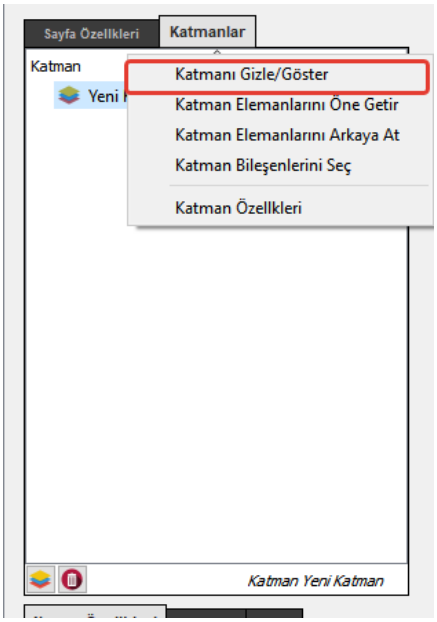

Figure 22 Layer Properties

- **Change Component Order in a Layer:**

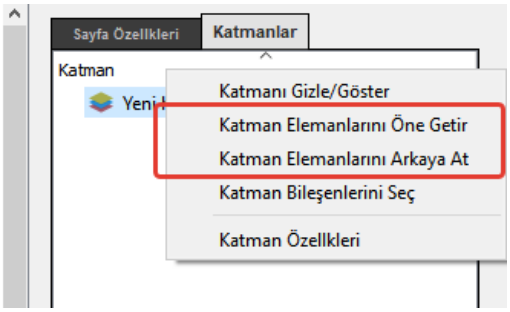

Figure 23 Changing the Order of Layers

You can change the order of the layers using the options "Bring Layer Components Forward" or "Send Layer Components Back" from the right-click menu

- **Hide/Show Layers Based on Zoom Level:**

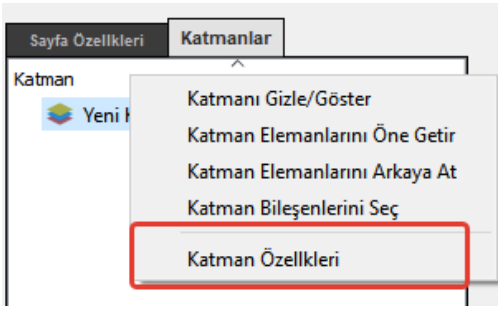

Figure 24 Selecting Layer Properties

You can hide or show defined layers on the page depending on the zoom level of the screen. This allows you to show components with more detail at higher zoom levels and reduce clutter by showing fewer components at lower zoom levels. To do this, first select "Layer Properties" in the Layers Panel.

In the opened dialog, two fields indicate the zoom level parameters.

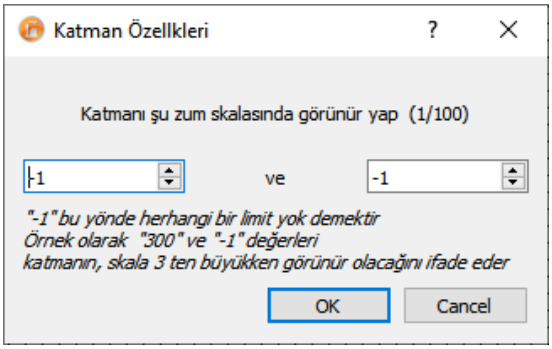

Figure 25 Layer Properties

The layer will be visible between the two zoom values entered here. If one of the values is set to "-1," the layer will remain visible regardless of the zoom level in that direction. For example, setting values to "300" and "-1" means: "Show the layer when the zoom level is higher than 3."

Zoom less than 3:

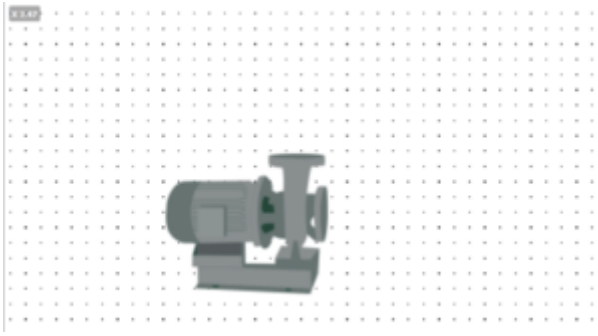

Figure 26 Low Zoom

Zoom greater than 3:

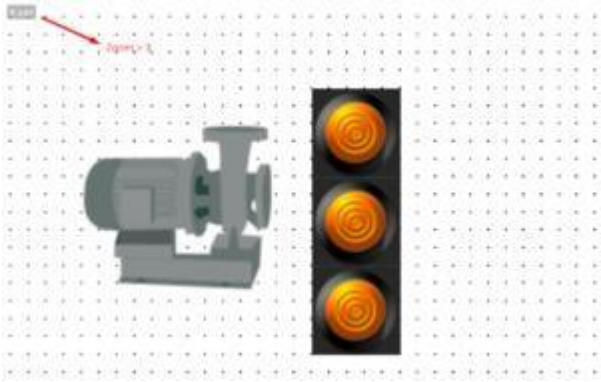

Figure 27 High Zoom

As shown in the image, when the zoom level exceeds 3, the layer containing the buttons becomes visible.

4.5 Object Properties Panel

The Object Properties panel allows you to make detailed settings for the selected component.

4.5.1 Object Properties Tab

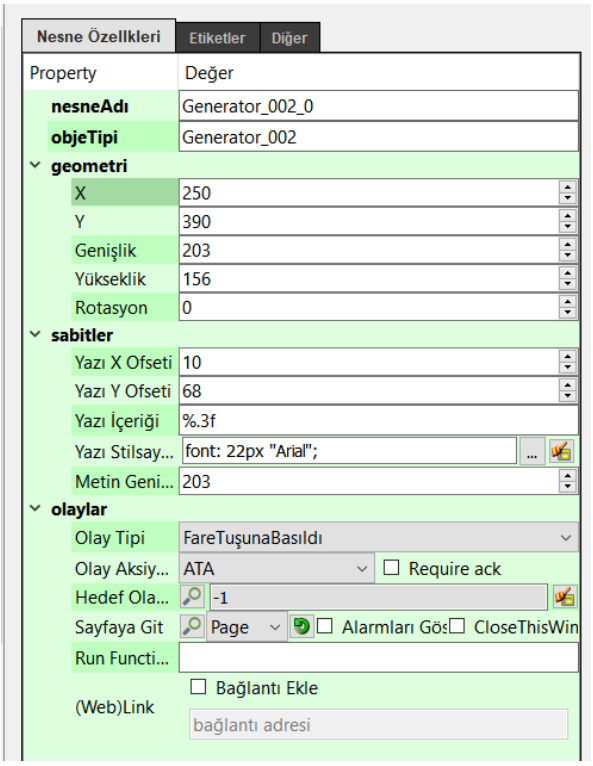

Figure 28 Component Parameters – Properties

- **Object Name:** A user-defined name to distinguish the component from others.
- **Geometry:** Position, size, and rotation values of the component on the screen are defined as integer
- **Constants:**

Text X Offset: If a “Text Tag” is defined and “Text Content” is entered, this parameter defines a horizontal offset for the displayed text.

Text Y Offset: If a “Text Tag” is defined and “Text Content” is entered, this parameter defines a vertical offset for the displayed text.

Text Content: If a “Text Tag” is defined, the formatted text entered here will be displayed on the component. The text must follow printf text formatting syntax.

Examples:

Integers: %d → "1977"

Leading spaces: %10d → " 1977"

Leading zeros: %010d → "0000001977"

Float numbers: %4.2f → "3.14"

- **Text Style Sheet:** You can define a stylesheet for the displayed text content to customize font, size, and color.
- **Text Width:** Specifies the maximum width of the text to be shown on the component.
- **Events:**

Event Type: Specifies under what condition the event is triggered.

Mouse Button Pressed: Triggered when the left mouse button is pressed.

Mouse Button Released: Triggered when the left mouse button is released.

Key Pressed: Triggered when a keyboard key is pressed.

Event Action: Function to be executed when the defined "Event Type" is triggered.

SET: Sets the value of the "Target Event Tag" to 1.

CLEAR: Sets the value of the "Target Event Tag" to 0.

TOGGLE: Changes the value of the "Target Event Tag": If 1 → 0, if 0 → 1.

LOAD VALUE: If selected, right-clicking the component in the client software opens a dialog allowing the user to manually change the "Target Event Tag" value.

Require Ack: If the event action is SET, CLEAR, or TOGGLE, an on-screen confirmation dialog is shown before the value is changed.

GO TO PAGE: Navigates to the client screen defined under "Go to Page".

GO TO LINK: Opens the link defined under "Web Link" in the Web View area of the client screen.

Target Event Tag: Defines the tag affected by the selected "Event Action".

Go to Page: If the event action is "GO TO PAGE", the page defined here opens when the component is clicked.

Show Alarms: When "GO TO PAGE" is selected, if there are active alarms on any tags within that page, an exclamation mark icon appears on the component.

Run Function: Used to create a dynamic page structure. For more information, see the Dynamic Pages section.

Add Link: Enables definition of a web link by checking the “Add Link” option.

Web Link: If the event action is "GO TO LINK", the URL entered here will be opened in the Web View area when the component is clicked.

4.5.2 Tags Tab

Values of the selected tags in this panel constantly update the corresponding parameters based on real-time field data. These tags can also be used in macros. Each tag has predefined variable types: i, o, s, w (e.g., i1, i2, o1, o2).

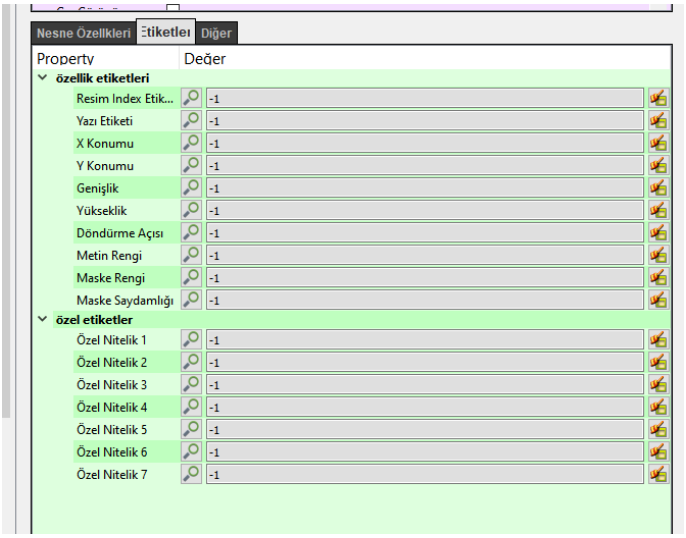

Figure 29 Component Parameters – Tags

- **Property Tags:**

Image Index Tag: Represents the currently displayed image index. The images and their corresponding indices can be viewed in the "Images" section under the "Other" tab or modified via the Component Manager.

Text Tag:The value of this tag is formatted using "Text Content" and displayed on the component.

X Position: Defines the horizontal screen coordinate of the component. The origin (0,0) is the top-left corner. X increases to the right.

Y Position: Defines the vertical screen coordinate of the component. The origin (0,0) is the top-left corner. Y increases downward.

Width: Width of the component.

Height: Height of the component.

Rotation Angle: Rotation angle of the component relative to the X-axis, in degrees.

Text Color: The color of the text shown on the component.*

Mask Color: The color of the mask applied to the component.*

** Color values are defined as hex values in the format `#112233`. These are converted into integers by taking the hex part (e.g., 112233) and interpreting it as a decimal number. You can view sample hex color codes here, or convert them using the tool here.*

Mask Transparency: A value between 0–255 is taken from this tag to define mask transparency.

- **Custom Tags:** You can define 7 custom tags to be used in macros. Each tag supports:

i (in): Raw value read from the field.

o (out): The value to be displayed on the screen after macro processing.

S (set): The value that the user intends to set.

w (write): The final value to be written to the field device after macro processing.

i ---Macro---> o, s ---Macro---> w: > If no macro is applied: i = o, s = w

4.5.3 Other Tab

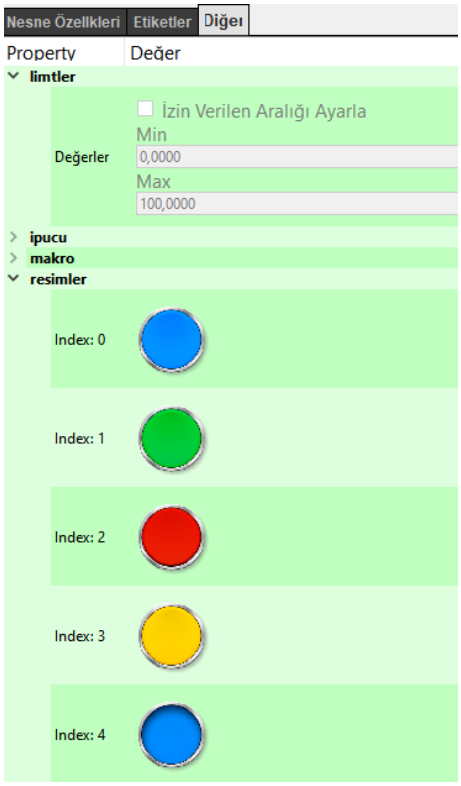

Figure 30 Component Parameters – Other

- **Limits:** Sets minimum and maximum values that users can assign to the "Target Event Tag" through the client software.
- **Tooltip:** The content shown when the user hovers the mouse over the component in the client.
- **Macro:** The macro script for the component is written here. It uses predefined tags from the Tags section.

- **Images:** Displays the images and their index values associated with this component. These can be edited in the “Component Manager” or new ones can be added.

Index 0: The image corresponding to index value 0 from the Image Index Tag

Index 1: The image corresponding to index value 1 from the Image Index Tag

Index 2: The image corresponding to index value 2 from the Image Index Tag

Index 3: The image corresponding to index value 3 from the Image Index Tag

5 Tag and Channel Editor

Channels and tags are special definitions that enable the SCADA software to access data on field devices. Channels contain protocol definitions and specific settings related to those protocols, which allow communication with field devices. Tags, on the other hand, consist of address definitions for registers on connected devices.

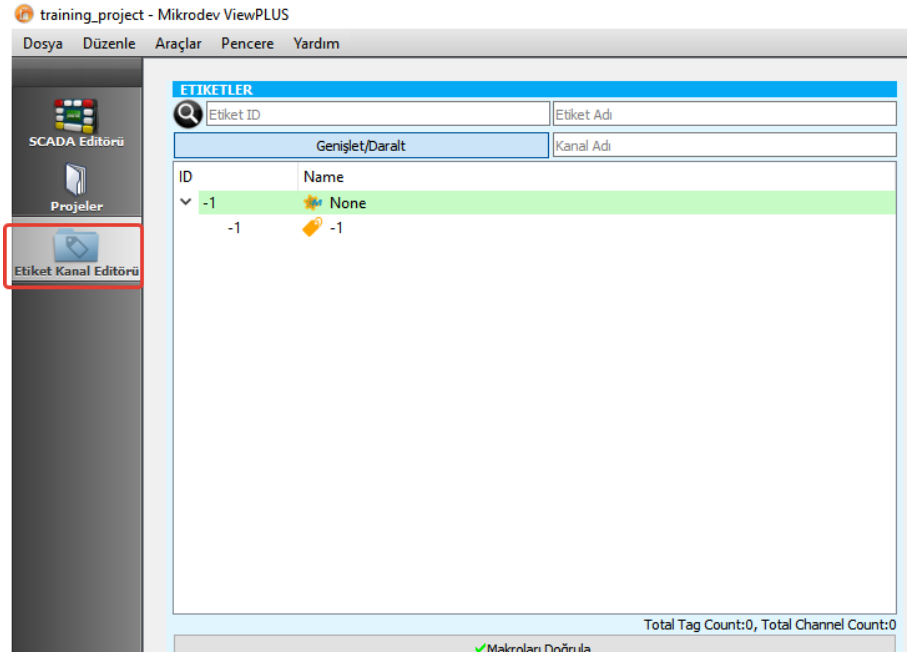

Figure 31 General View of Tag and Channel Editor

In the opened dialog window, necessary configurations related to the new channel can be made.

5.1 Channels

Channels can be created to define communication settings with field devices, or to establish virtual channels such as macros or database queries.

To create a new channel:

1. Open the Tag and Channel Editor tab.
2. Right-click on any existing channel.
3. Select New Channel from the menu.

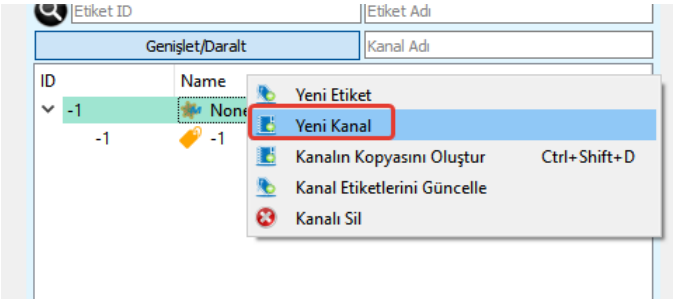

Figure 32 Adding a New Channel

5.1.1 Modbus TCP Channel

To communicate with field devices using the Modbus TCP protocol, a new Modbus channel must be created. In the new channel creation window, select “Modbus TCP” as the Protocol Type.

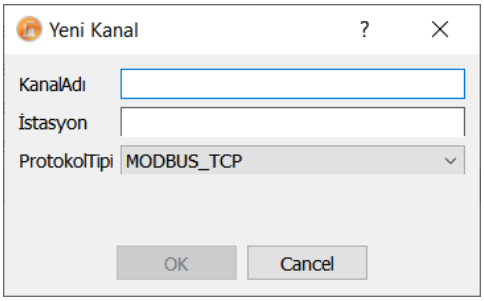

Figure 33 Modbus TCP Channel Definition

5.1.1.1 Channel Parameters

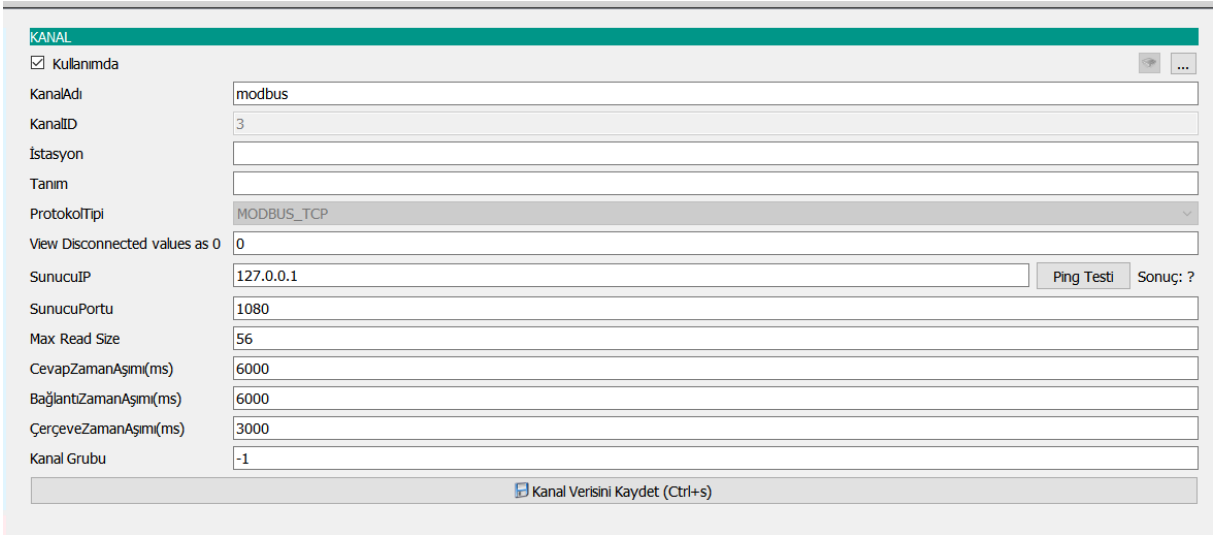

Figure 34 Modbus Channel Parameters

- **View Disconnected values as 0:** This option should be enabled (set to 1). If the connection with the slave device is lost, this parameter ensures that the values of the related Modbus tags are displayed as 0.
- **Server Address**: Enter the IP address of the device to be used for Modbus TCP communication.
- **Server Port:** Enter the port number used by the device for Modbus TCP communication (default is usually 502).
- **Response Timeout:** The time to wait for a response from the slave device after each Modbus query (in milliseconds). If no response is received within this time, the query is retried.
- **Connection Timeout:** The time to wait after sending a connection request to the device before retrying if the connection cannot be established (in milliseconds).
- **Frame Timeout:** The time to wait after receiving a response to a query before sending the next query (in milliseconds).

5.1.1.2 Tag Parameters

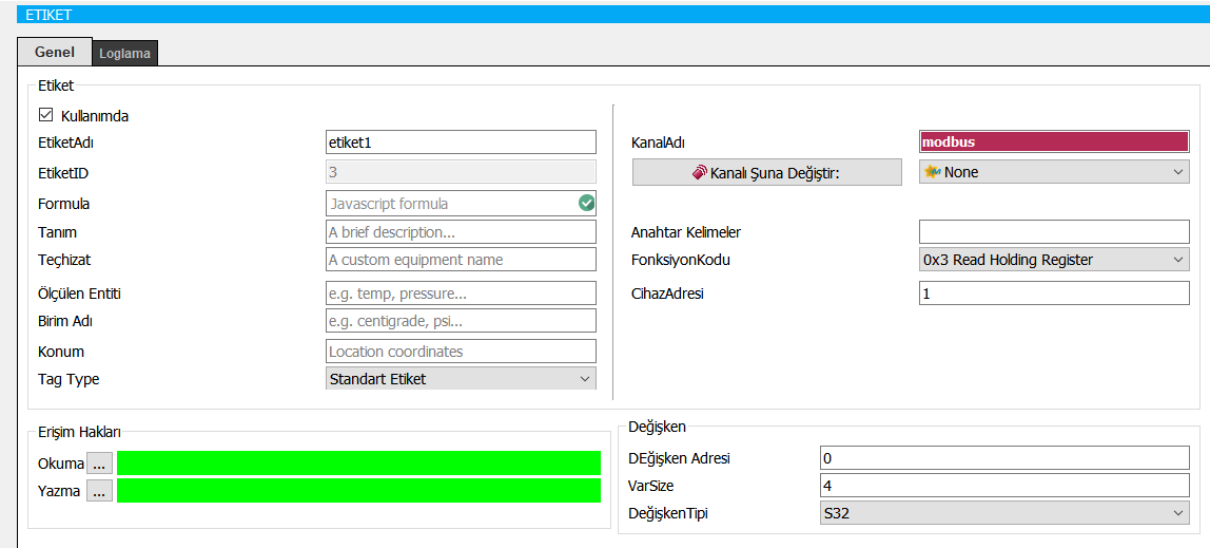

Figure 35 Modbus Tag Parameters

- **Function Code:** The function code used to read or write the tag via Modbus (e.g., 03: Read Holding Registers, 06: Write Single Register).
- **Device Address (Slave ID):** The identifier number of the target slave device in the Modbus protocol.
- **Register Address:** The Modbus address of the variable to be read or written.
- **Data Size:** The size of the data at the specified address. This value is determined automatically based on the selected data type.
- **Data Type:** The type of data at the specified address (e.g., Integer, Float, Boolean).

5.1.2 MQTT Client Channel

To communicate with field devices over a TCP/IP network using the MQTT protocol, a new MQTT Client channel must be created.

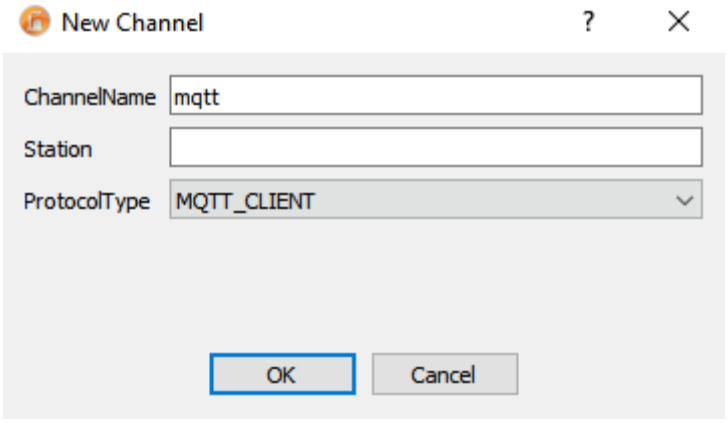

Figure 36 MQTT Client Channel

Two different Payload formats are supported for communication via the MQTT driver.

Selecting the Payload Format

The payload format determines the structure of MQTT messages and how data is configured and transmitted.

- **Payload Format: 0**

Compatible with Mikrodev devices. Uses a simple structure containing the value of a single variable.

Structure: { "deger3": 14.0000 }

deger3: The variable name defined in the MQTT tag. Represents the line tag defined in the PLC project.

Preferred in applications where only a single data field is sent.

- **Payload Format: 1**

This format supports multiple devices and multiple variables.

Structure: { "124": { "test_deger": {"V": 15.0000 } } }

124: Device ID used in MQTT communication.

test_deger: The name of the variable.

V: Field that carries the actual value (e.g., "V" - Value).

This structure is especially useful in large-scale projects for distinguishing data by device.

5.1.2.1 MQTT Channel Parameters

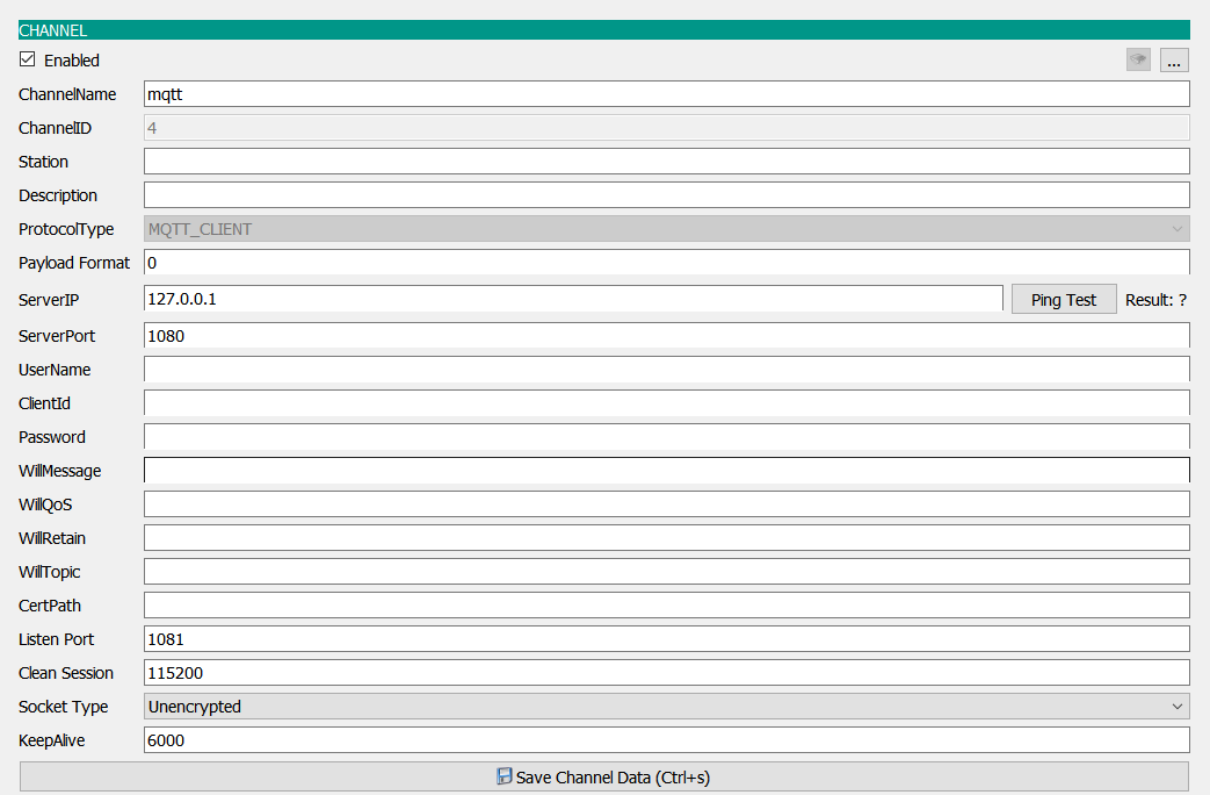

Figure 37 MQTT Channel Parameters

The following settings are used to connect to the MQTT broker:

Server IP: IP address of the MQTT broker.

Server Port: The port number the broker listens on (e.g., 1883).

Username: Enter the username if authenticated connection is required.

Client ID: The client ID to be used when connecting to the broker.

Password: Password for the given username.

Will Message: The message the broker publishes if the connection is lost.

Will QoS: Quality of Service level for the Will message (0, 1, 2).

Will Retain: Whether the Will message should be retained by the broker (true/false).

Will Topic: The MQTT topic to which the Will message is published.

Cer Path: Full path of the certificate file if a TLS encrypted connection is used.

Listen Port: The port on which the MQTT client listens for data. For example: 1081.

Clean Session: true/false. If true, previous session data is cleared upon session termination.

Socket Type: Connection type – Unencrypted or SSL/TLS.

Keep Alive: The interval to send keep-alive packets to maintain the connection (in milliseconds). For example: 6000.

5.1.2.2 Tag Definitions

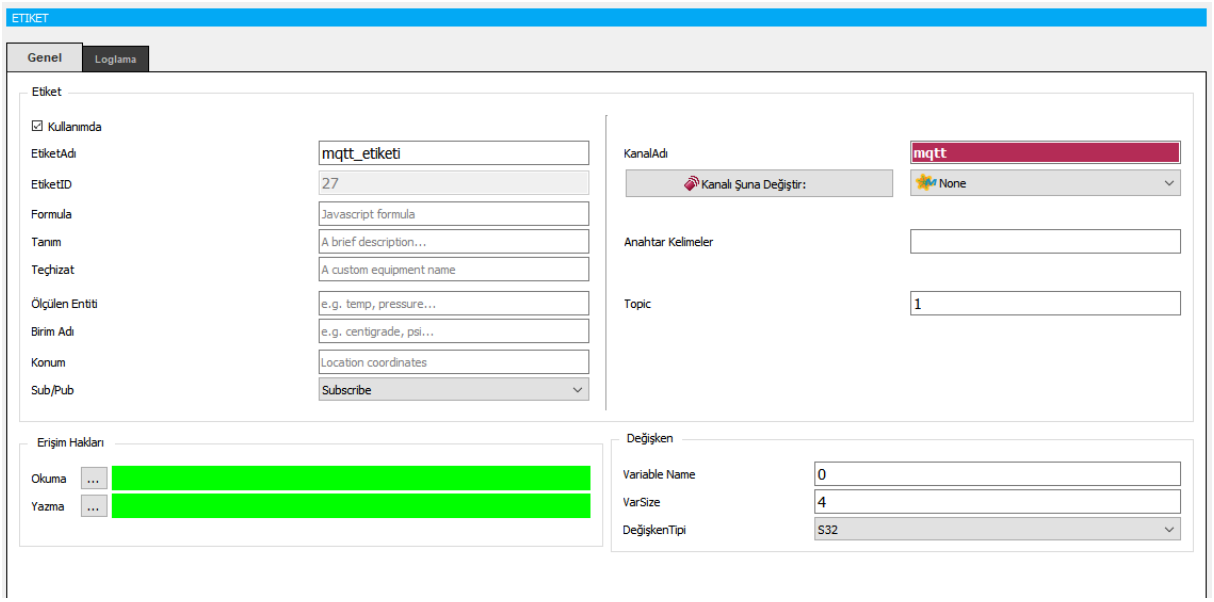

Figure 38 MQTT Tag Definitions

Device (Device ID): Enter the unique ID number of the device.

- This field can be left blank.
- It is mandatory if Payload Format 1 is used.

Topic: The topic name used to send or receive data on the MQTT broker.

- Different topics can be defined for each variable or device.

Sub/Pub: Select whether the channel will perform Publish, Subscribe, or both.

- Subscribe – Receives data
- Publish – Sends data
- Sub / Pub – Both sends and receives

Variable Name: Enter the variable name.

- This name must exactly match the line tag name in the PLC project.
- Data matching in MQTT messages is done using this name.

Variable Type: Select the variable type here.

- Supported types include s32, u32, s16, u16, s64, u64, bool, dbl, and all inverse types.

VarSize: The variable size is automatically assigned by the SCADA software based on the selected type.

- No manual input is required from the user.

5.1.3 Macro Channel

The Macro Channel is used to create virtual tags and perform script-like operations for calculations or logical checks on these tags.

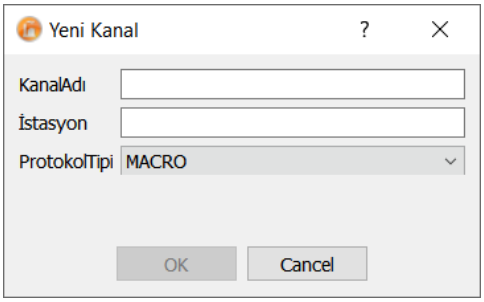

Figure 39 Macro Channel Definition

5.1.3.1 Channel Parameters

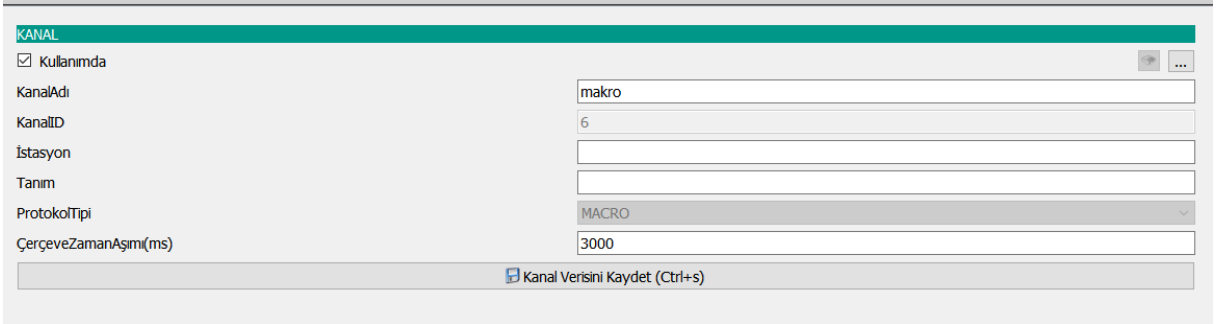

Figure 40 Macro Channel Parameters

- **Frame Timeout:** The time interval that determines how frequently the macro channel is executed. It is defined in milliseconds (e.g., 1000 = executed every 1 second).

5.1.3.2 Tag Parameters

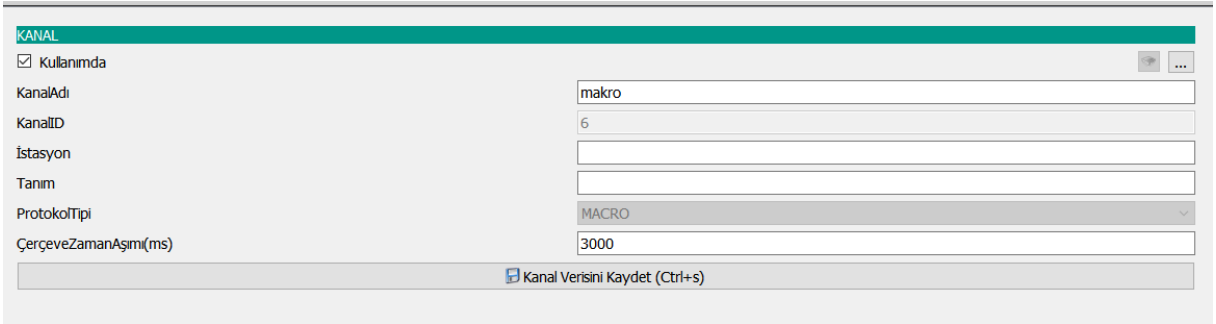

Figure 41 Macro Writing Screen

Virtual variables used in macros can be defined from v0 to v199. A total of 200 virtual variables are supported.

Tag sources:

Real Tags: Called using the actual tag ID, such as $1234.

Constant Values: Direct constants like 234, 12.5 can be used as integers or decimals.

5.1.3.3 Macro Commands

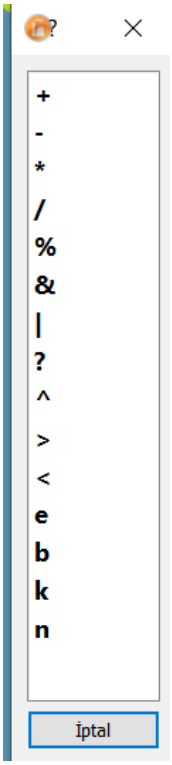

Figure 42 Macro Commands

The operators usable within the macro channel are as follows:

- **+:** Addition
- **-:** Subtraction
- ***:** Multiplication
- **/:** Division
- **%:** Modulus
- **&:** Logical AND
- **|:** Logical OR
- **^:** Logical XOR
- **>:** Greater than
- **<:** Less than
- **e:** Equals
- **n:** Not equal
- **b:** Greater than or equal
- **k:** Less than or equal

- **?:** Special operation operator

5.1.3.4 Example Usages

Arithmetic Operation Example

[v0 = $1234 * 2]

Explanation: The value of the tag with ID $1234 is multiplied by 2 and written into variable v0.

Usage of Special ? Operator

Syntax	Explanation
[v0 = 1234 ? 0]	Gets RX counter value of tag 1234
[v0 = 1234 ? 1]	Gets read time of tag 1234
[v0 = 1234 ? 2]	Checks communication status (0 or 1)
[v0 = 1234 ? 3]	Checks if the value is valid
[v0 = v0 ? 20]	Converts v0 to epoch time (in seconds)
[v1 = v0 ? 21]	Extracts year from v0
[v2 = v0 ? 22]	Extracts month from v0
[v3 = v0 ? 23]	Extracts day from v0
[v4 = v0 ? 24]	Extracts hour from v0
[v5 = v0 ? 25]	Extracts minute from v0
[v6 = v0 ? 26]	Extracts second from v0

Condition Control Commands

- [IF] – Condition Validation

[IF , v0 , 2]

[v1 = 555]

[E]

Explanation: If v0 equals 1, the next line (v1 = 555) is executed. Otherwise, it skips down by the parameter 2 and ends at [E].

- [NI] – Negative Condition (NOT IF)

[NI , v0 , 2]

[v1 = 555]

[E]

Explanation: If v0 equals 0, the next line (v1 = 555) is executed. Otherwise, it skips down by 2 lines and exits with [E].

Macro Termination

Each macro must be ended with the [E] command. This indicates the completion of the macro cycle.

Note: Macro language is line-based; each operation should be written on a single line.

Note: Tag IDs are written with the $ symbol, while constant IDs are used directly in ? operations (e.g., 1234).

5.1.4 IEC104 Channel

To communicate with field devices over the TCP/IP network using the IEC 60870-5-104 (IEC 104) protocol, you need to create a new IEC 104 channel.

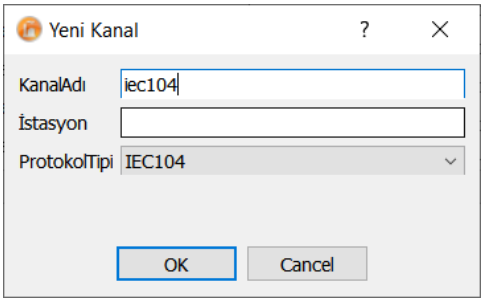

Figure 43 IEC 104 Channel Definition

5.1.4.1 Channel Parameters

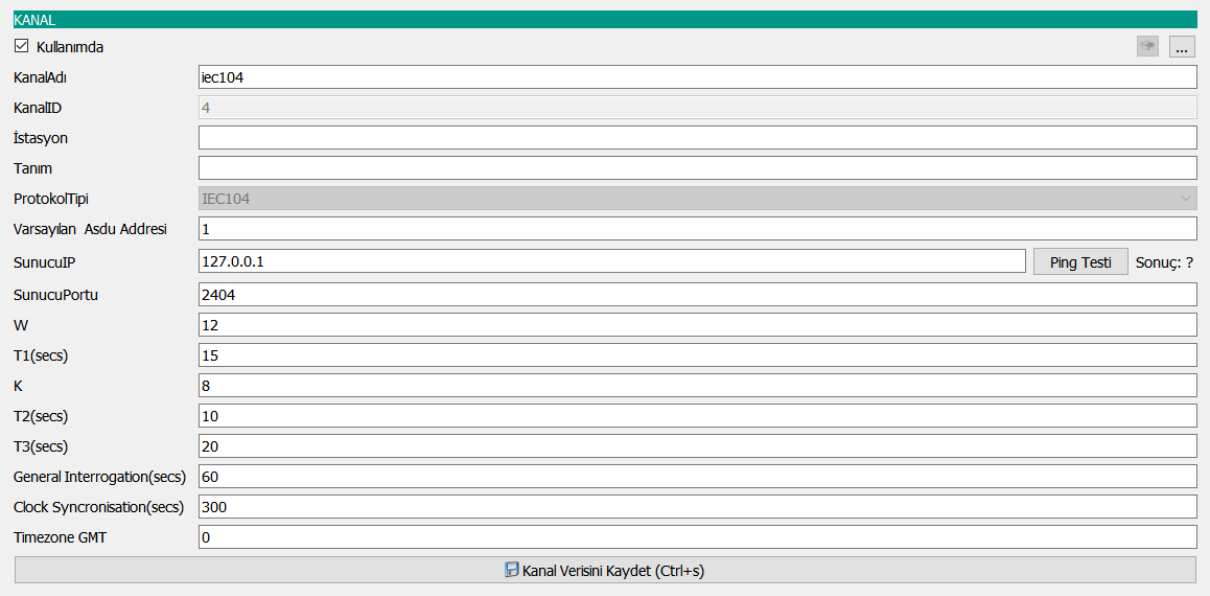

Figure 44 IEC 104 Channel Parameters

- **W (Window Size - Send Acknowledgement Trigger):** This parameter defines after how many data frames an ACK (acknowledgement) message will be sent. An ACK is sent after sending W number of data frames.
- **K (Acknowledgement Window Size):** The maximum number of packets that can be sent without receiving an ACK. If this number is exceeded, the connection may break or need to be re-established.
- **T1 (Acknowledgement Timeout):** The maximum time to wait for an ACK after an ASDU is sent.
- **T2 (Passive ACK Timeout):** If no new data is sent, an ACK is automatically sent to the other party after this time.
- **T3 (Test Frame Timeout):** Used to verify whether the connection is still alive. A test frame is sent at regular intervals based on this parameter.
- **General Interrogation (GI):** Frequency of sending the General Interrogation command. This command retrieves all available data again (in seconds).
- **Clock Synchronisation (CS):** Frequency of sending the Clock Synchronisation (CS) command to synchronize system time (in seconds).
- **Timezone GMT:** The time zone of the server running the SCADA system in GMT. Time synchronizations are adjusted based on this value.

5.1.4.2 Tag Parameters

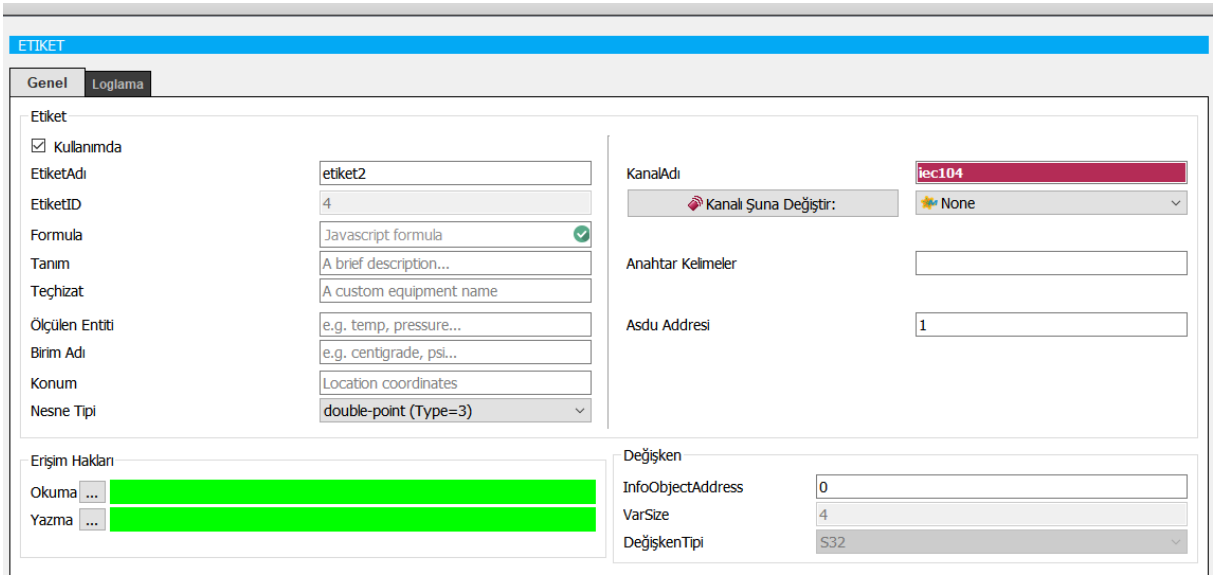

Figure 45 IEC 104 Tag Parameters

- **ASDU Address:** The Application Service Data Unit (ASDU) address to which the tag belongs. This address is used to define data groups in the IEC 104 system.
- **Object Type:** Specifies the IEC 104 object type of the data to be read or written. For example: Single-point information, Measured value, etc.
- **InfoObject Address:** The information object address of the defined tag. SCADA uses this address to perform read and write operations.

5.1.5 DNP3 Channel

A new DNP3 Channel must be created to communicate with field devices over a TCP/IP network using the DNP3 (Distributed Network Protocol).

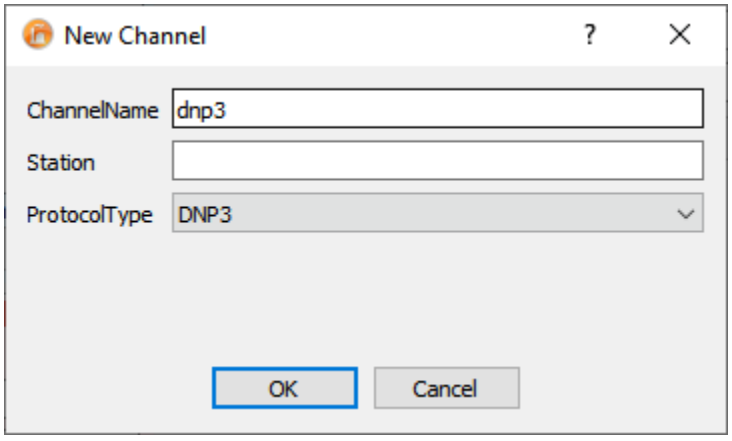

Figure 46 Defining the DNP3 Channel

5.1.5.1 Channel Parameters

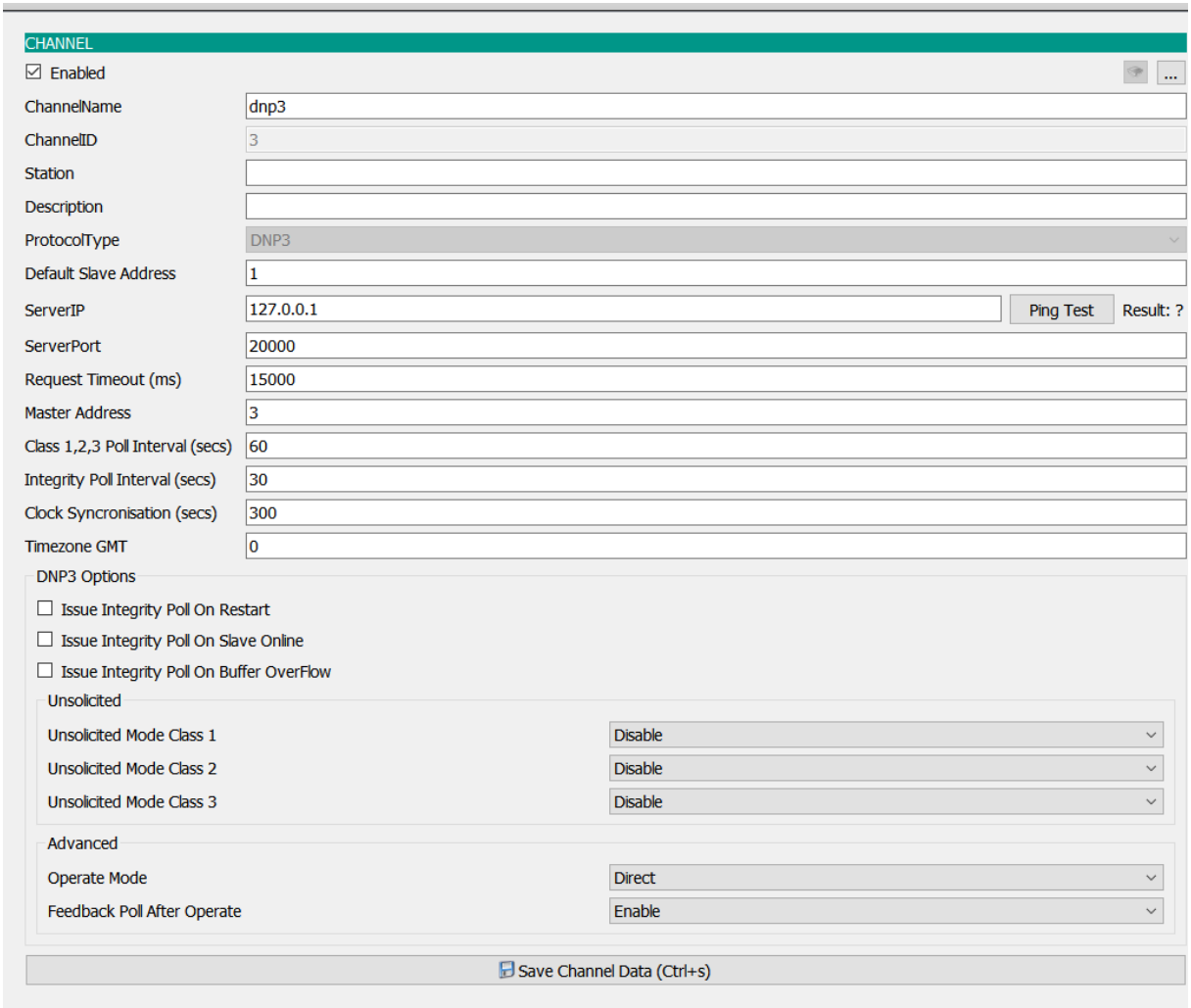

Figure 47 DNP3 Channel Parameters

- **Default Slave ID:** The default slave address used in requests sent to slave devices (DNP3 Slave ID). Each device should have a unique address.
- **Server IP:** The IP address of the slave device for DNP3 communication. The connection is made to this address over TCP/IP.
- **Server Port:** The TCP port used for the DNP3 protocol. The default port number is 20000.
- **Request Timeout:** The maximum time (in milliseconds) the SCADA system will wait for a response from the slave device after sending a request. If no response is received within this time, the request is considered to have timed out.
- **Master Address:** The DNP3 Master ID of the SCADA system. Slave devices use this address to identify incoming requests. It must be unique within the system.
- **Class 1/2/3 Polling Interval:** Specifies polling intervals for DNP3 data classes. These queries allow periodic collection of event-based data (in seconds).

  Class 1: High priority data (e.g., alarms, fault conditions)

Class 2: Medium priority data

Class 3: Low priority data

- **Time Synchronization:** Frequency of sending time information from SCADA to slave devices to prevent clock drift (in seconds).
- **Integrity Polling Interval:** The frequency with which SCADA queries the complete dataset from slave devices, even if no events occur (in seconds).
- **Clock Synchronization (CS):** The frequency with which the SCADA system sends a CS command to synchronize device clocks at the protocol level. This adds an extra layer of security beyond standard time sync.
- **Timezone GMT:** The time zone of the computer running the SCADA system, in GMT. This is used as a reference during time synchronization.
- **DNP3 Options:**

  Issue Integrity Poll on Restart: When the SCADA system restarts, it automatically initiates an integrity poll to retrieve all data.

  Issue Integrity Poll on Slave Online: When a slave device comes online, an integrity poll is automatically performed.

  Issue Integrity Poll on Buffer Overflow: If the slave device's buffer overflows, the system initiates an integrity poll to prevent data loss.

Unsolicited:

Unsolicited Mode Class 1: Indicates whether the slave device is allowed to send Class 1 (high priority) data to SCADA unsolicited.

Unsolicited Mode Class 2: Allows the slave device to send Class 2 (medium priority) data to SCADA unsolicited.

Unsolicited Mode Class 3: Allows the slave device to send Class 3 (low priority) data to SCADA unsolicited.

This mode supports event-based data transmission and can reduce polling frequency while accelerating data delivery.

Advanced:

Operate Mode: Specifies how commands are sent:

Direct: The command is executed immediately.

Select Before Operate: The command is first confirmed with a "select" step before sending the "operate" command. Recommended for security.

Feedback Poll After Operate: After an "operate" command, the corresponding tag’s status is queried again to verify successful execution.

5.1.5.2 Tag Parameters

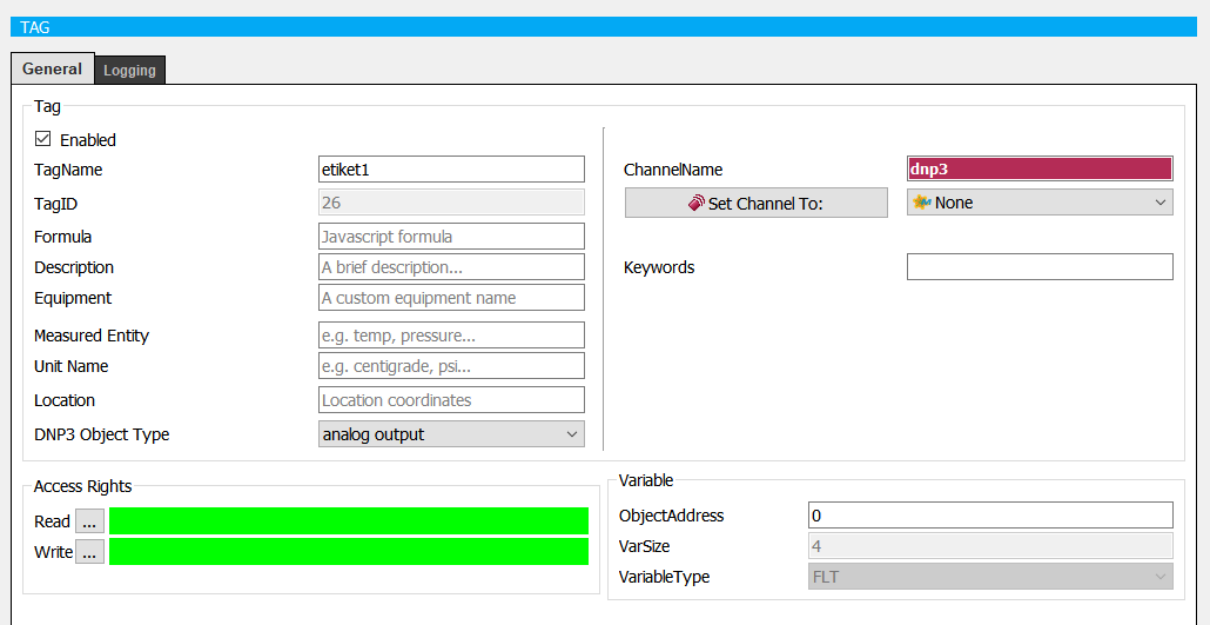

Figure 48 DNP3 Tag Parameters

To define data tags to be used in the SCADA system via DNP3 protocol, the following parameters must be configured:

- **DNP3 Object Type:** Specifies the type of data represented by the tag. One of the following types, compliant with the DNP3 standard, should be selected:

Object Type	Explanation
Binary Input	Digital input (e.g., switch open/closed, alarm state). Read-only.
Binary Output	Digital output (e.g., relay trigger). Writable.
Control Relay	Digital output for control commands. Typically supports “Select Before Operate.”
Analog Input	Analog input data (e.g., temperature, pressure). Read-only.
Analog Output	Analog output value (e.g., adjustable voltage/current). Writable.
Control Analog 32	32-bit analog control command.
Control Analog 16	16-bit analog control command.
Control Analog Float	Analog control command with floating-point (float) data type.

- **Object Address:** The address (index) where the data resides in the DNP3 protocol. SCADA uses this address to read from or write to the device.

5.1.6 Global Database Channel

The Global Database Channel is used to connect to databases running on external systems outside of SCADA, and to transfer specific query results from those databases to SCADA project tags. This channel enables flexible data exchange by allowing access to remote databases like PostgreSQL or SQLite instead of the SCADA's local database.

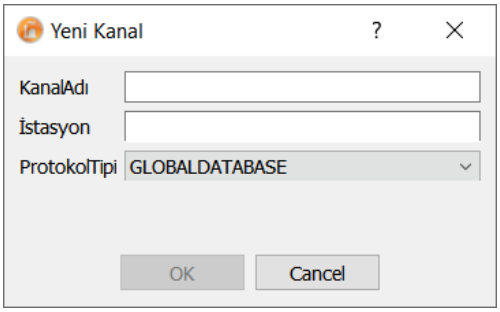

Figure 49 Defining the Global Database Channel

5.1.6.1 Channel Parameters

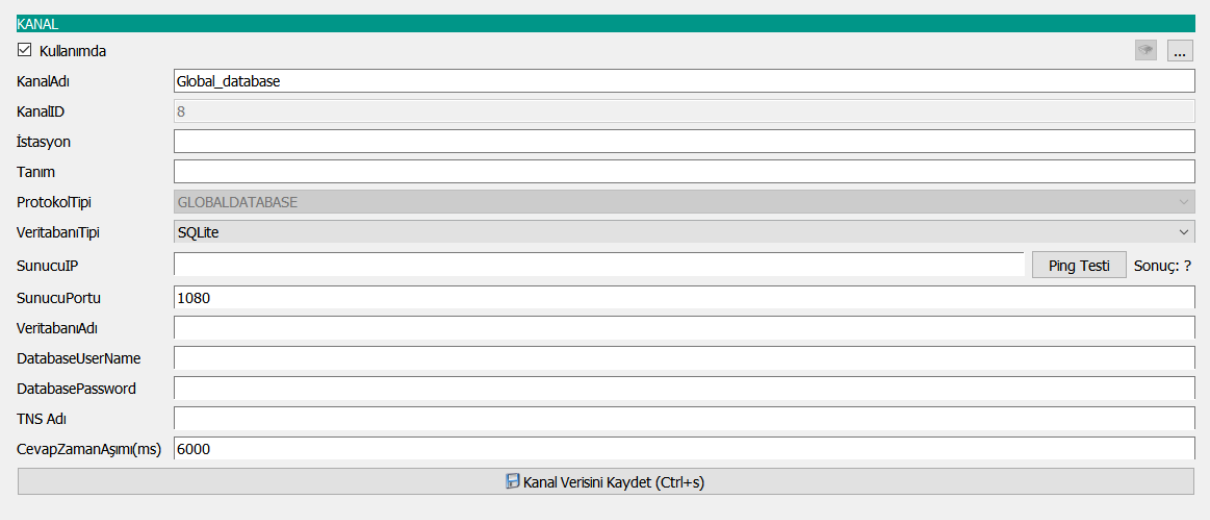

Figure 50 Global Database Channel Parameters

- **Database Type:** The type of the database to connect (e.g., PostgreSQL, MySQL, etc.)
- **Server IP:** IP address of the database server
- **Server Port:** Port number of the database server
- **Database Name:** Name of the target database to connect
- **Database Username:** Username for accessing the database
- **Database Password:** Password for accessing the database
- **Response Timeout:** Timeout duration in milliseconds to wait for a query response if none is received. For example: 10000 → 10 seconds.

5.1.6.2 Tag Parameters

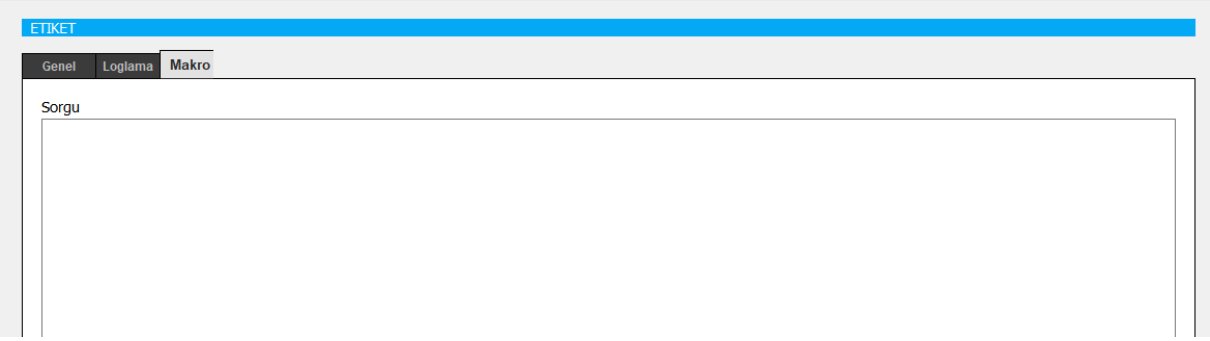

Figure 51 Global Database Tag Parameters

Database queries are written in the macro section of the channel. The obtained query results can be directly written to project tags.

5.1.6.3 Example Database Query:

```
WITH
 t1 AS (
  SELECT data_value FROM logs.tag_log WHERE tag_id=19 AND data_value IS NOT NULL ORDER BY
logtime DESC LIMIT 1),
 t2 AS (
  SELECT data_value FROM logs.tag_log WHERE tag_id=20 AND data_value IS NOT NULL ORDER BY
logtime DESC LIMIT 1),
 t3 AS (
  SELECT data_value FROM logs.tag_log WHERE tag_id=29 AND data_value IS NOT NULL ORDER BY
logtime DESC LIMIT 1),
 t4 AS (
  SELECT data_value FROM logs.tag_log WHERE tag_id=26 AND data_value IS NOT NULL ORDER BY
logtime DESC LIMIT 1)
SELECT t1.data_value AS data1, t2.data_value AS data2, t3.data_value AS data3, t4.data_value AS data4
FROM t1, t2, t3, t4;
```

:{${32}, ${33}, ${34}, ${35}}:

Explanation:

- The latest database values of tags with tag_id 19, 20, 29, and 26 are fetched.
- These values are then transferred to SCADA tags with IDs 32, 33, 34, and 35 respectively.
- Using the :{${...}}: syntax, the columns returned from the query are sequentially mapped to SCADA tags.

5.1.7 Database Channel

The Database Channel is a type of channel used in the SCADA system to execute SQL queries on its own database in order to query specific tag or record information. This channel is especially suitable for historical data analysis or operations with specific conditions.

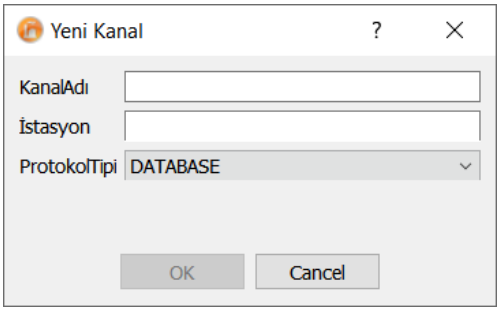

Figure 52 Defining the Database Channel

5.1.7.1 Channel Parameters

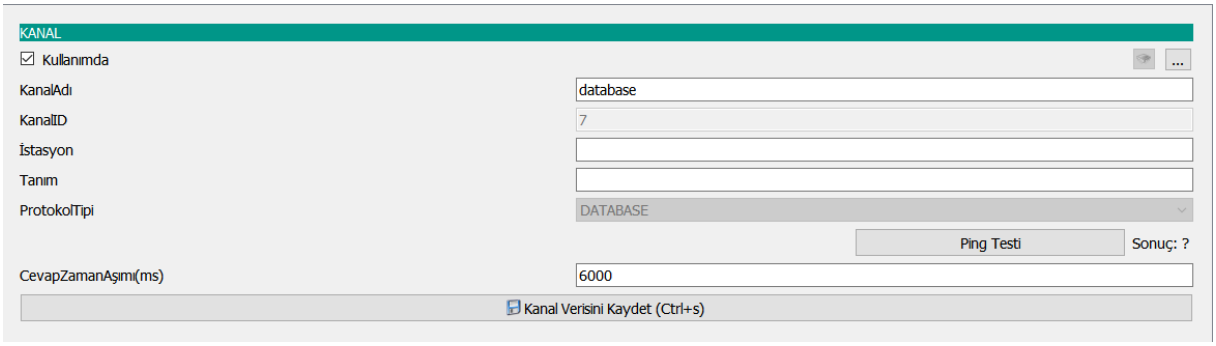

Figure 53 Database Channel Parameters

- **Response Timeout:** The timeout value defines how long the system should wait if no response is received from the database query. This value is in milliseconds. For example: a value of 10000 means the system will wait 10 seconds for a response. If no response is received within this time, the connection is terminated and considered an error.

5.1.7.2 Tag Parameters

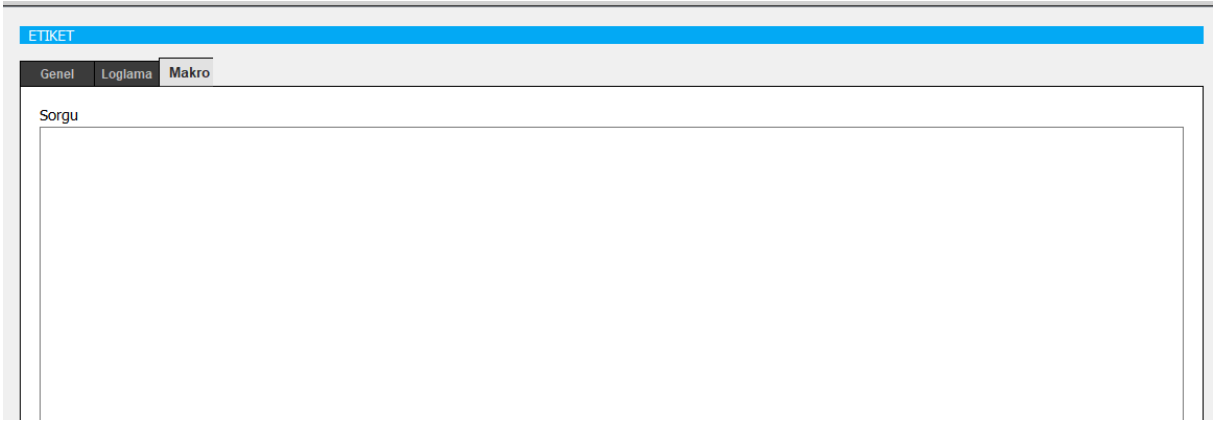

Figure 54 SQL Query Input Screen

SQL queries are defined in the macro section of the channel. Direct access is provided to the internal database of the SCADA system.

Note: The query must return only a single value. Queries that return multiple rows or columns are considered invalid.

5.1.7.3 Example Queries

- Last value of a specific tag:

SELECT data_value FROM logs.tag_log WHERE tag_id=1 ORDER BY logtime DESC LIMIT 1

- First value of a specific tag:

SELECT data_value FROM logs.tag_log WHERE tag_id=1 ORDER BY logtime LIMIT 1

5.1.8 ICCP Channel

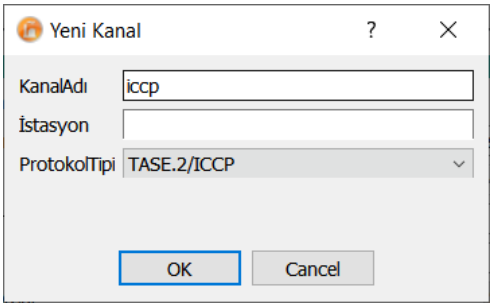

Figure 55 ICCP Channel Definition

Communication Test and Compatibility Requirements

- Mutual Ping Test: Required to verify the connection between the SCADA and TEİAŞ servers.
- Port 102: Must be open on the system running SCADA and should allow incoming connections.
- Firewall & Antivirus: Custom rules may need to be defined for port 102 in these software systems.

5.1.8.1 Channel Parameters

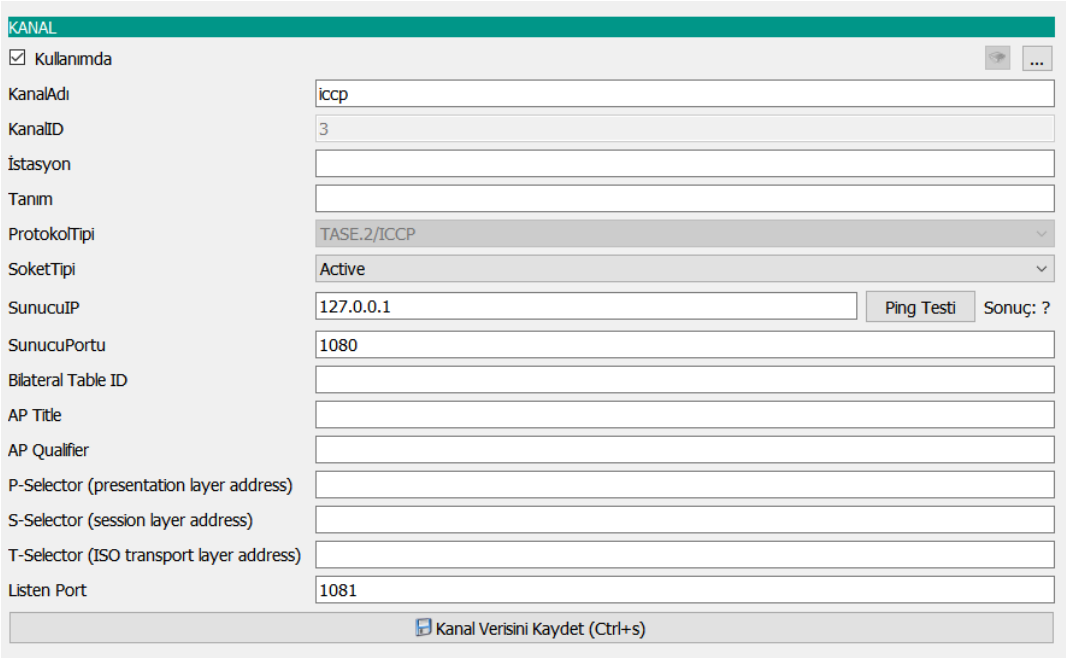

Figure 56 ICCP Channel Parameters

Channel Name: Provided by TEİAŞ. Enter the IP as 127.0.0.1. The term “TEIAS” must be removed.

Protocol Type: Should be selected as TASE.2 / ICCP.

Socket Type: Should be set to Passive.

Server IP: Should be entered as 127.0.0.1.

Server Port: 102 (default ICCP port)

Listen Port: Should be set to 102.

Bilateral Table ID: Provided by TEİAŞ. Enter as per ICCP client configuration information.

AP Title: Provided by TEİAŞ.

AP Qualifier: Provided by TEİAŞ.

P-Selector: Provided by TEİAŞ. Enter the IP as 127.0.0.1.

S-Selector: Provided by TEİAŞ. The term “TEIAS” must be removed.

T-Selector: Provided by TEİAŞ. The term “TEIAS” must be removed.

Note: The ICCP version on the ViewPLUS SCADA side must match the ICCP version provided by TEİAŞ.

5.1.8.2 Tag Parameters

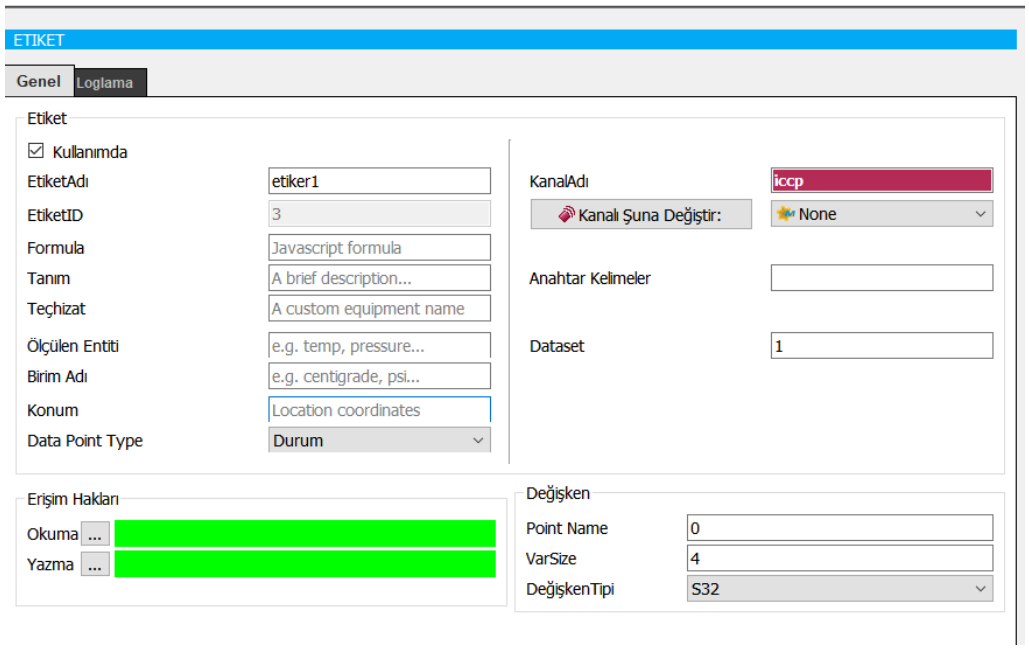

Figure 57 ICCP Tag Parameters

ICCP Object Name: The name specified in the TEİAŞ signal list; used directly as the tag name.

Dataset: Determined based on the signal unit:

- MWh → COUNTER
- MW, MVAR → ANALOG

Data Point Type: Should be set to RealQ for all signals.

Variable Type: Should be set to DBL (double) for all signals.

5.1.9 Server Info Channel

The Server Info Channel provides access to system-level information of the SCADA server. Through this channel, real-time data such as the number of active/defined/acknowledged alarms, system time, and connection status can be read.

Its purpose is to monitor the system status and enhance operational awareness.

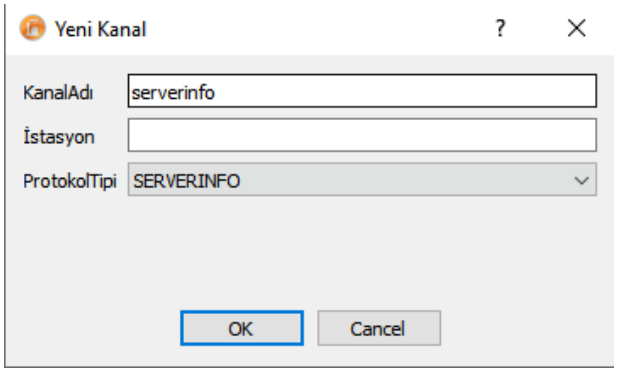

Figure 58 Server Info Channel Definition

5.1.9.1 Channel Parameters

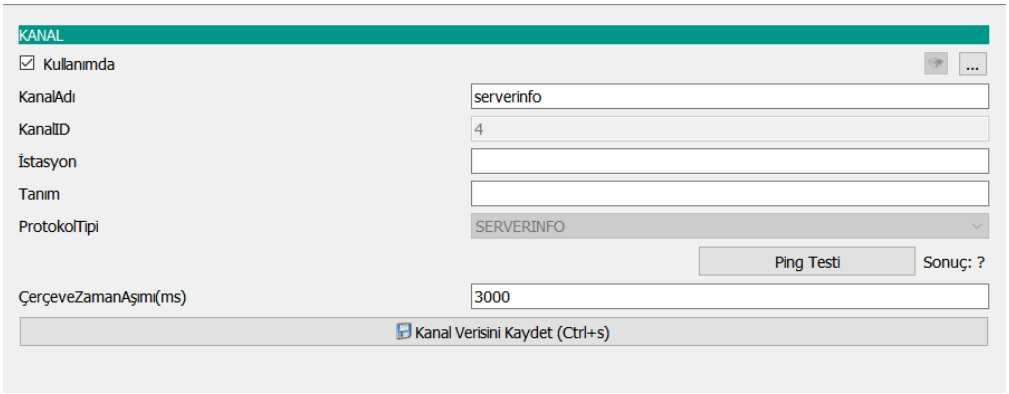

Figure 59 Server Info Channel Parameters

Response Timeout: The interval at which the function defined in the server info channel will be executed. Expressed in milliseconds.

Example: 1000 ⟶ The function will be called every 1 second.

5.1.9.2 Tag Parameters

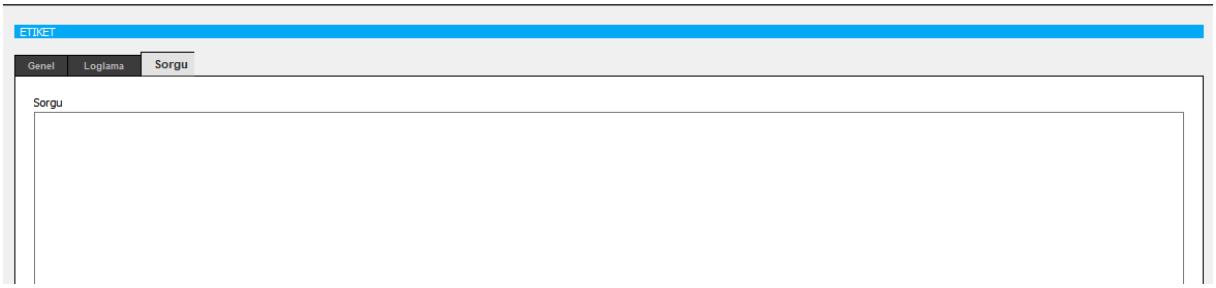

Figure 60 Server Info Tag Parameters

Each tag's Query field is populated with special commands (keywords) that return information related to the server's internal state.

Note: Although written in JavaScript syntax, this channel uses special system-specific keywords.

Note: The return statement is mandatory. Without it, the query result will not be written to the tag.

Note: Values cannot be assigned to other system tags; only read/query operations are allowed.

5.1.9.3 Supported Keywords and Descriptions

Keyword	Description
:ALARMCOUNT()	Returns the total number of defined alarms in the system.
:ALARMCOUNT(ClassName)	Returns the number of alarms defined in the specified alarm class.
:ACTALARMCOUNT()	Returns the number of currently active (triggered) alarms.
:ACTALARMCOUNT(ClassName)	Returns the number of active alarms within the specified class.
:ACKEDALARMCOUNT()	Returns the total number of acknowledged active alarms.
:ACKEDALARMCOUNT(ClassName)	Returns the number of acknowledged active alarms for the specified class.
:SYSTEMTIME()	Returns the server system time in milliseconds since the epoch.
:CONNECTEDCHANNELS(ChannelName)	Returns the number of connected (active) channels in the specified channel group.
:DISCONNECTEDCHANNELS(ChannelName)	Returns the number of disconnected (inactive) channels in the specified channel group.

5.1.9.4 Example Usages

Example 1: Total number of defined alarms in two alarm classes

var1 = :ALARMCOUNT(Panel1);

var2 = :ALARMCOUNT(Panel2);

totalalarms_defined = (var1 + var2);

return totalalarms_defined;

Example 2: Simple tag query to display the number of active alarms

return :ACTALARMCOUNT();

Example 3: Retrieving the server system time

return :SİSTEM ZAMANI();

5.1.10 SNMP Channel

The SNMP (Simple Network Management Protocol) Channel allows the SCADA system to read data from devices that communicate using the SNMP protocol.

Through this channel, systematic data can be collected from network infrastructure devices such as switches, UPS units, routers, etc.

Note: For this channel to work, the Net-SNMP library must be installed on the system. You can access the relevant library here.

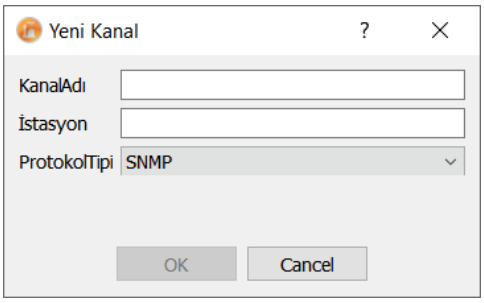

Figure 61 SNMP Channel Definition

5.1.10.1 Channel Parameters

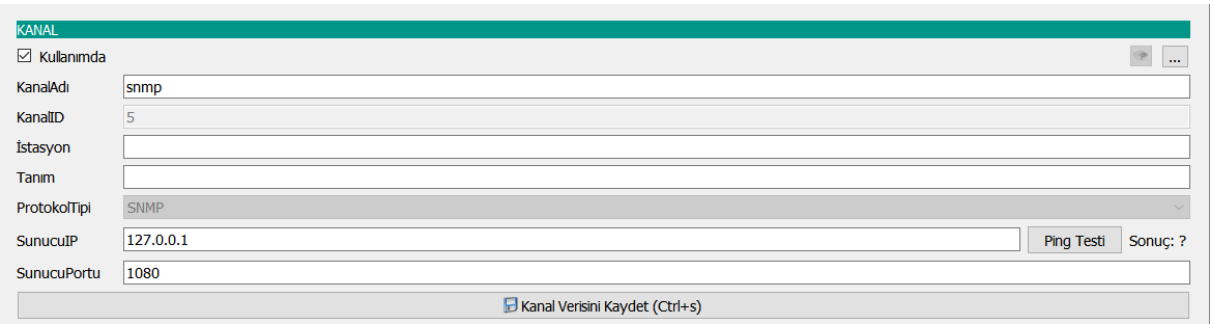

Figure 62 SNMP Channel Parameters

Server Address: The IP address of the device to which SNMP queries will be sent.

Server Port: The SNMP port number.

Default value: 161

5.1.10.2 Tag Parameters

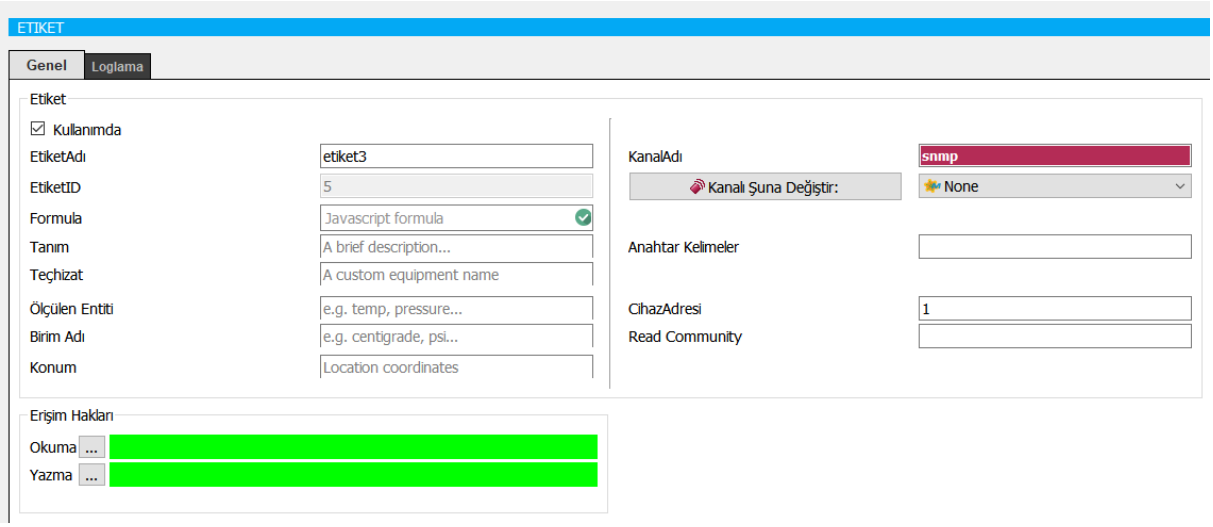

Figure 63 SNMP Tag Parameters

Device Address: The address information of the device defined in the SNMP protocol. It is usually the same as the IP address or may require a specific definition.

Read Community: The access key used to retrieve SNMP data (e.g., public).

Required for SNMP versions 1 and 2.

5.1.11 SOAP API Channel

The SOAP API Channel is a channel type used to connect to a SOAP (Simple Object Access Protocol) Web Service, parse the service responses, and write the retrieved data into tags within the SCADA project. SOAP envelopes are processed using a WSDL-based definition structure, and the response is integrated into the SCADA system using an XML parser.

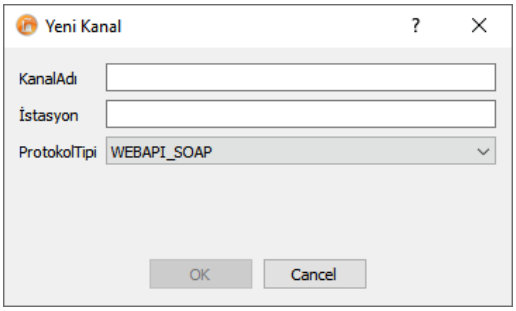

Figure 64 SOAP API Channel Definition

5.1.11.1 Channel Parameters

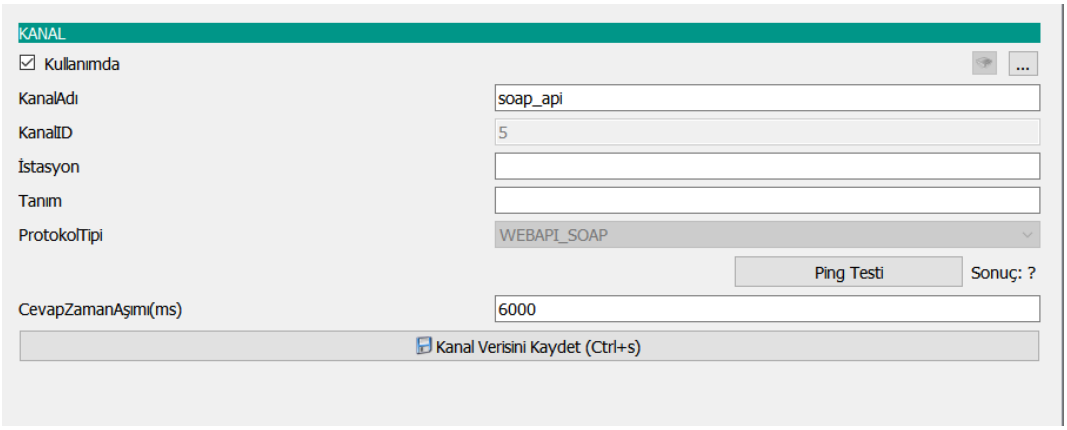

Figure 65 SOAP API Channel Parameters

Response Timeout: The timeout duration (in milliseconds) that defines how long the system should wait for a response after executing a SOAP query. If no response is received within the defined time, the connection is terminated.

5.1.11.2 Tag Parameters

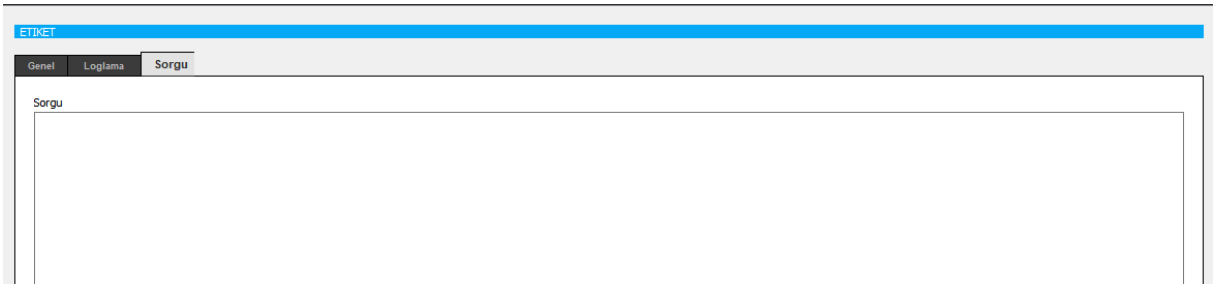

Figure 66 SOAP API Tag Parameters

The queries to be sent to the SOAP service are written into the Query field of the relevant tag. The query consists of three parts:

WSDL Path: The WSDL definition of the SOAP service.

SOAP Envelope: The request prepared in XML format.

Tag Mapping Rule: Defines how the XML response will be mapped to SCADA tags.

5.1.11.3 Example Implementation

Example Query:

WSDL_PATH = https://minosxcloud.umpi.it/ws/wsminos.php?wsdl;

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ns1="http://localhost/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
 <SOAP-ENV:Body>
  <ns1:getStatus>
   <user xsi:type="xsd:string">ws-istanbul</user>
   <password xsi:type="xsd:string">passwordvalue</password>
   <db_name xsi:type="xsd:string">databasename</db_name>
   <id_andros xsi:type="xsd:string">s345dfsad2345asd45fsdfsgds4542345a</id_andros>
  </ns1:getStatus>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

:{${32}, <stato><statopl num="1"><fase1 stato>;

${33}, <stato><statoingresso num=17 stato>}:

Example Response from Server:

```
<SOAP-ENV:Envelope SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
   xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/">
   <SOAP-ENV:Body>
      <ns1:getStatusResponse
         xmlns:ns1="http://localhost/">
         <res xsi:type="xsd:string">
            <armadio identificatore="b3e1f32cb0db36ef0fbfaf047074e4d5">
               <stato>
                  <statoingresso num="16" stato="1">ON Input Andros CMS-EXP 1</statoingresso>
                  <statoingresso num="17" stato="1">ON Input Andros CMS-EXP 2</statoingresso>
                  <statopl num="1">
                     <fase1 stato="0">Line 1 Phase 1 OK</fase1>
                     <fase2 stato="0">Line 1 Phase 2 OK</fase2>
                     <fase3 stato="0">Line 1 Phase 3 OK</fase3>
                  </statopl>
               </stato>
            </armadio>
         </res>
      </ns1:getStatusResponse>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Tag Mapping Rules (Parser Structure)

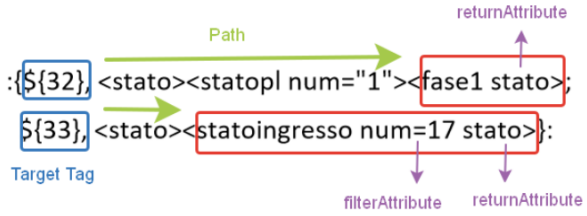

Figure 67 SOAP API Tag Writing Method

:{${TagID}, <child1><child2><child3 filterAttribute returnAttribute>}:

Interpretation Rules:

Condition	Description
Only node path is defined	The node's text value is read and written to the tag.
returnAttribute is defined	The value of the specified attribute in the first matched node is written to the tag.
Both filterAttribute and returnAttribute	The tag receives the attribute value of the node that matches the specified filter.

Examples:

${32}, <stato><statopl num="1"><fase1 stato> →The value of the stato attribute in the fase1 node is written to tag ID 32.

${33}, <stato><statoingresso num=17 stato> →The stato value of the statoingresso node with num="17" is written to tag ID 33.

5.1.12 JavaScript Channel

The JavaScript Channel is a channel type used within the SCADA system to create JavaScript-based calculations, conditional operations, or data processing logic. Using JavaScript functions, values of different tags can be read, mathematical operations can be performed, and the result can be written to a virtual tag.

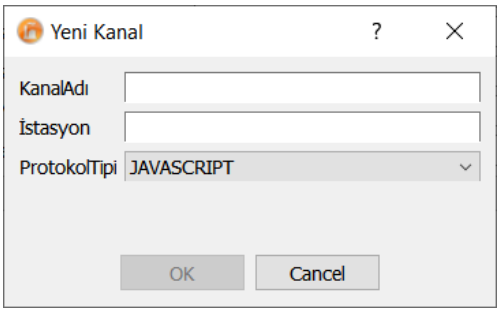

Figure 68 JavaScript Channel Definition

5.1.12.1 Channel Parameters

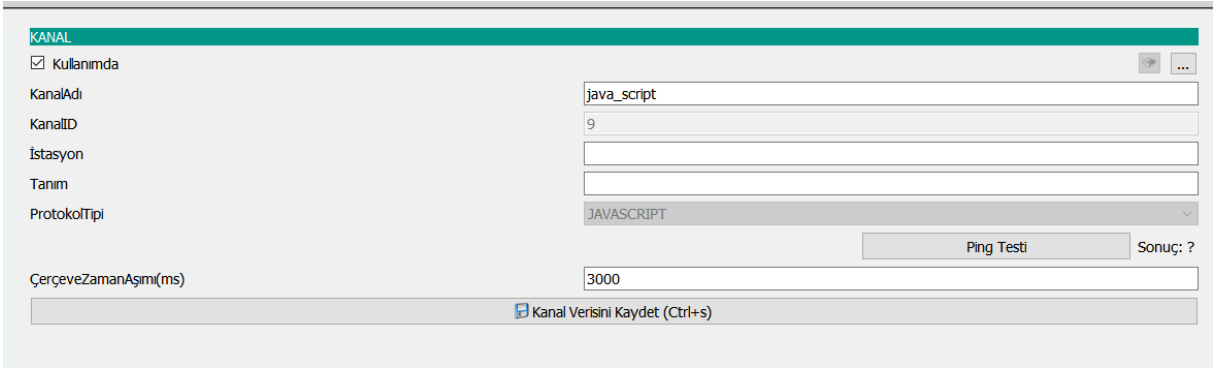

Figure 69 JavaScript Channel Parameters

- **Frame Timeout:** The interval at which the JavaScript function will be executed, in milliseconds. For example: 1000 → the function is called every 1 second.

5.1.12.2 Tag Parameters

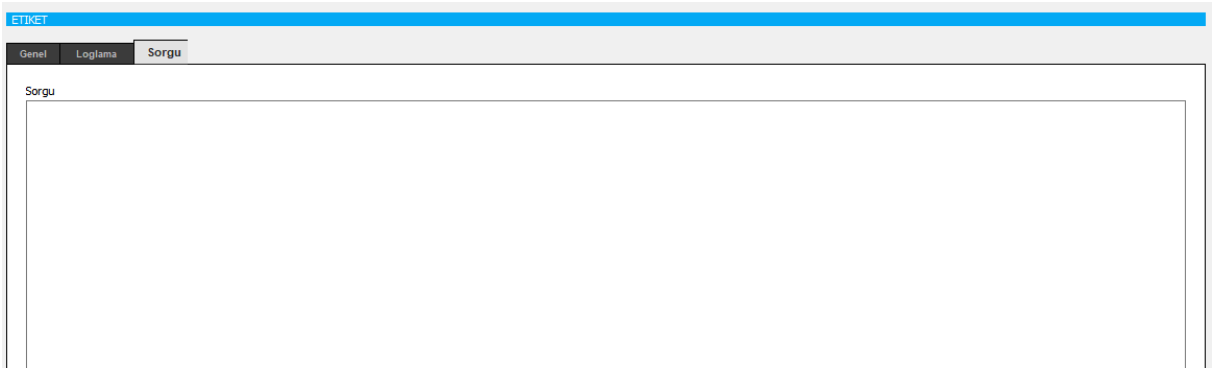

Figure 70 JavaScript Tag Parameters

JavaScript commands are written in the Query field of the relevant tag. Other system tags can be accessed using the format ${tagID}.

Note: Assigning values to other system tags is not allowed within the JavaScript channel. Use the Macro Channel for such operations.

5.1.12.3 Value Assignment Rules for JavaScript Tags

Reading tag values: ${2}, ${3}, etc.

Defining and processing variables: var x = ...

Return: Use the return ...; command to write the result to the SCADA tag.

5.1.12.4 Supported JavaScript Features

- All basic arithmetic operations
- Comparison operators (===, >, <, !==)
- Math object: Math.sqrt(), Math.abs(), Math.pow(), etc.
- return is mandatory; otherwise, the value will not be written to the tag.

5.1.12.5 Example Applications

Addition Example

Adds the values of two tags and returns the result.

```
var3 = ${2} + ${3};
return var3;
```

Sum and Square Root Calculation

```
var1 = 5 + ${2};
var2 = 3 + ${3};
subtotal = var1 + var2;
return Math.sqrt(subtotal);
```

Condition Comparison (Equality Check)

```
var1 = ${2};
var2 = ${3};
var3 = var1 === var2 ? 1 : 0;
return var3;
```

Explanation: Returns 1 if var1 and var2 are equal, otherwise returns 0.

Invalid Example (Will Not Work)

```
${3} = 3 + var1;
```

Note: This example is invalid because value assignment to other tags is not allowed in the JavaScript channel. Use the Macro Channel for this purpose.

6 SCADA Editor – Alarms

Alarms are one of the most critical components of SCADA systems in terms of monitoring and intervention. ViewPLUS SCADA provides alarm generation and management based on field data and user definitions.

6.1 Creating an Alarm

Alarm definitions are created through the Tag/Channel Editor panel.

- Open the Tag/Channel Editor screen.
- Right-click the “Alarms” tab located in the lower-left section.
- Select the “New Alarm” option from the menu.

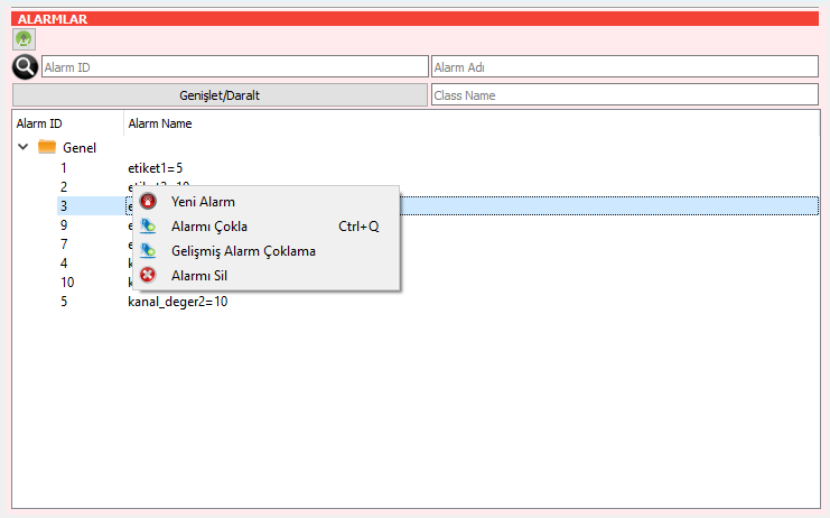

Figure 71 Adding a New Alarm

In the alarm definition screen that appears, enter the alarm name, description, and class information.

- If an existing class name is entered, the alarm is added to that class.
- If a new class name is entered, the system creates a new class and assigns the alarm to it.

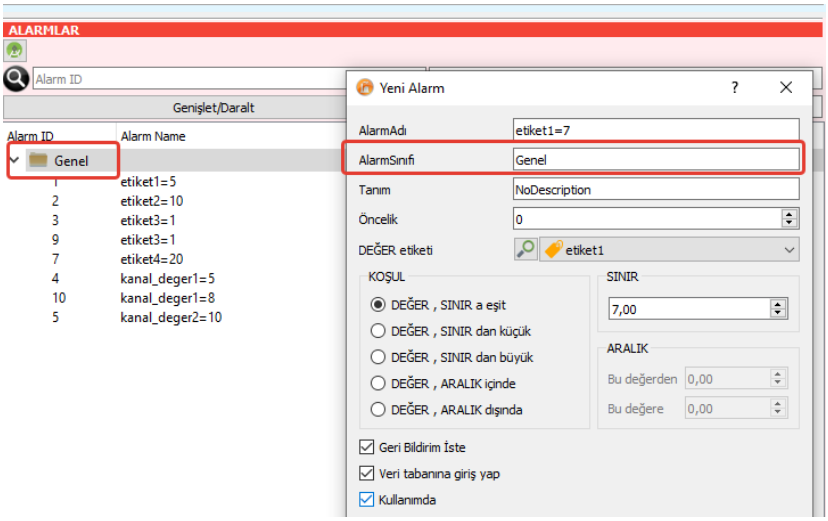

Figure 72 Creating an Alarm Class

6.1.1 General Alarm Parameters

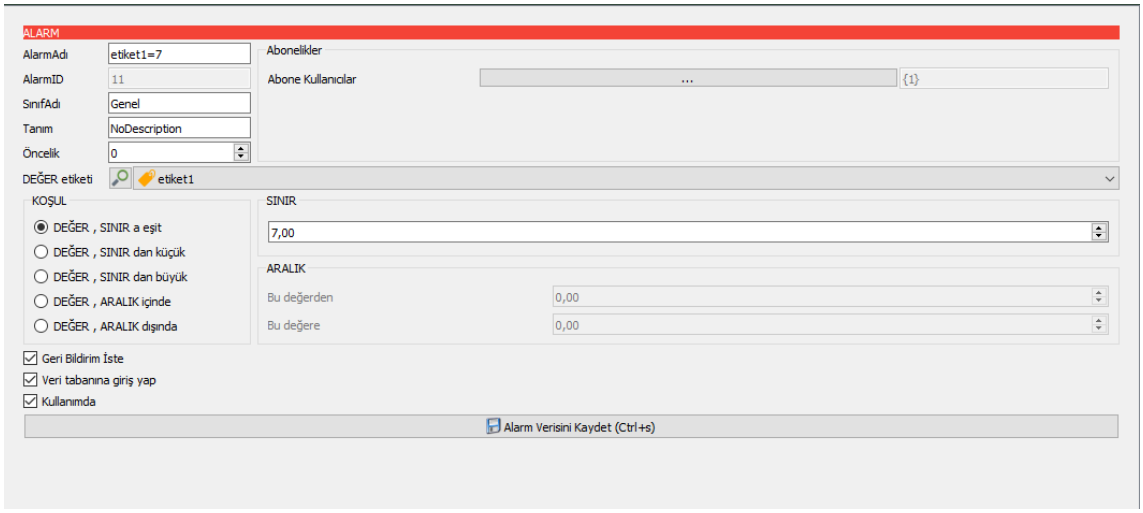

Figure 73 General Alarm Parameters

Alarm Name: The name by which the alarm will be recognized in the system.

Class Name: The name of the class to which the alarm belongs. Allows grouping of alarms.

Description: A descriptive text explaining the function of the alarm.

Priority: Defines the importance level of the alarm. ViewPLUS SCADA (v0.9.154 and later) supports four levels:

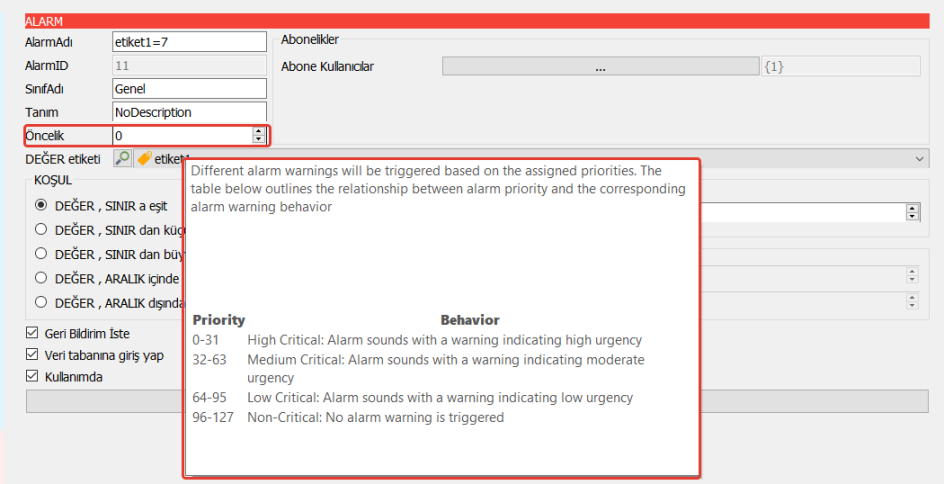

Figure 74 Alarm Priorities

0–31: High
32–63: Medium
64–95: Low
96–127: Warning (silent alarm – no sound)

Value Tag: The tag to be used for triggering the alarm.

Condition: The logical condition that activates the alarm: equal to, less than, greater than, within/outside a range, etc.

Subscriptions: Users who will monitor the alarm are defined here.

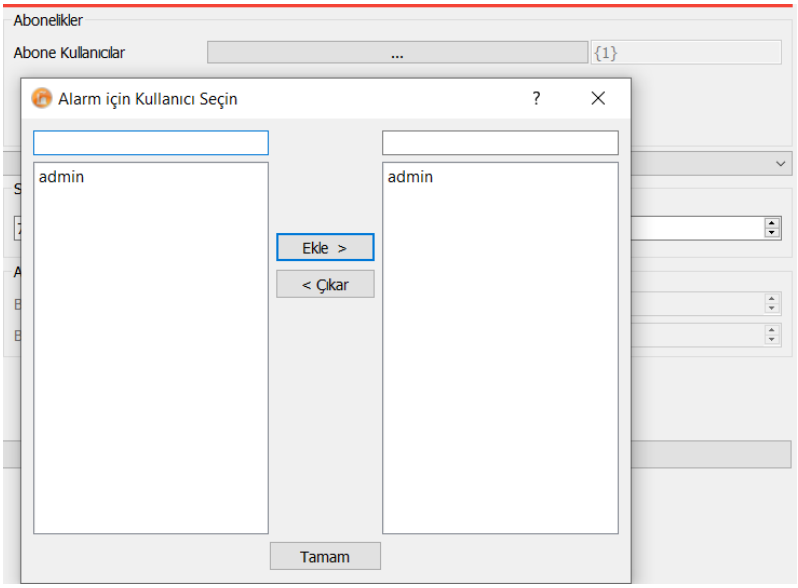

Figure 75 Defining Users Subscribed to the Alarm

Other Settings:

- **Require Acknowledgment:** Requires user confirmation.
- **Log to Database:** Logs the alarm status to the database.
- **Enabled:** Defines whether the alarm is active or passive.

6.2 Monitoring Alarms

Created alarms can be monitored both from the Client interface and SCADA design pages.

6.2.1 Monitoring Alarms on SCADA Design Pages

Using SCADA components, visual monitoring and acknowledgment of alarms can be performed directly on pages.

Steps:

1. Create a new SCADA project and define tags
2. Add alarm conditions to relevant tags.

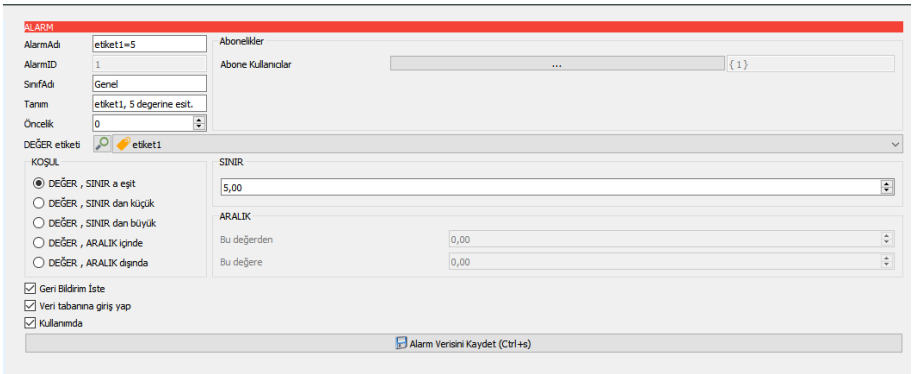

Figure 76 Example of Creating an Alarm Condition

3. Add components like Display and Button to the SCADA page.

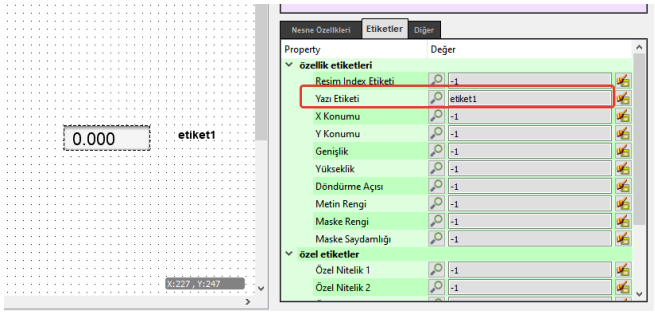

Figure 77 Associating Alarm Tag with a SCADA Component

4. In the component's "Events" tab, check the Show Alarms option.

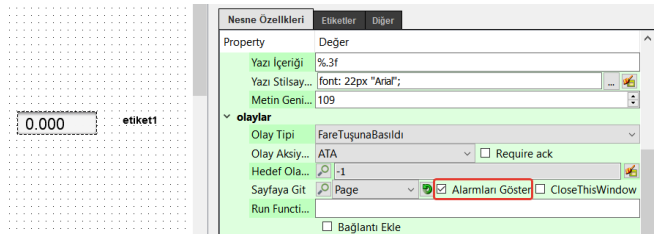

Figure 78 Enabling the Show Alarms Option

5. Start the server and open the Client screen.
6. When an alarm occurs, a bell icon or color change appears on the component.

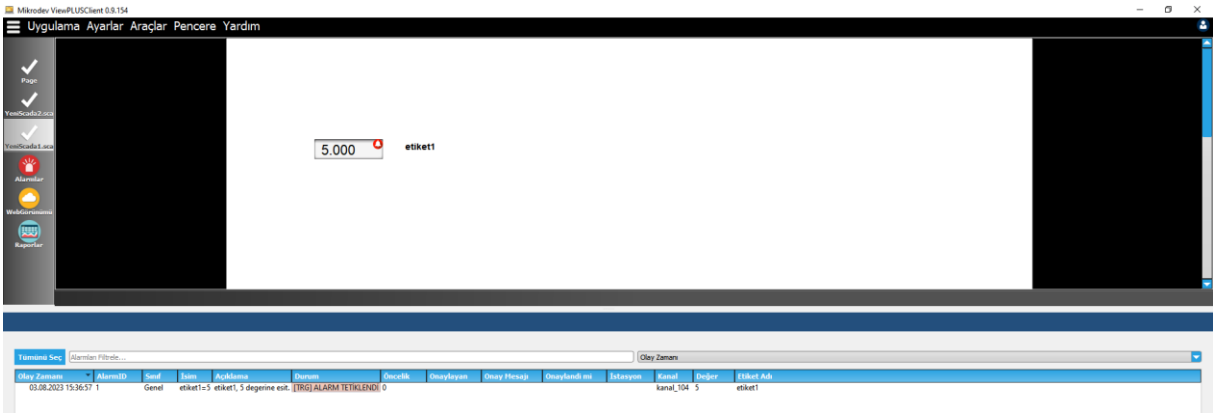

Figure 79 Monitoring Alarm from the Client

7. Right-click the bell icon and select Acknowledge Alarm to confirm it.

Figure 80 Acknowledging the Alarm

8. After entering the acknowledgment message, the alarm icon disappears.

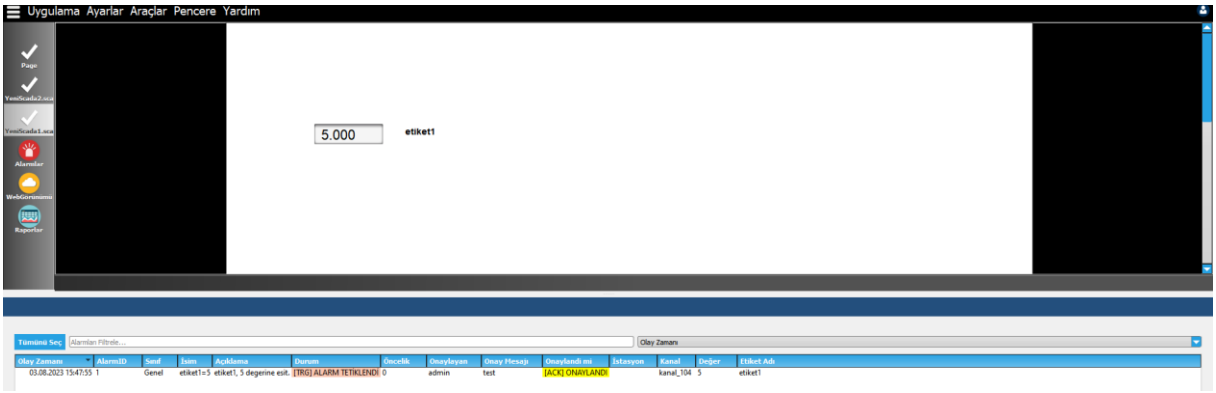

Figure 81 Monitoring the Acknowledged Alarm from the Client

6.2.2 Page-Based Alarm Monitoring

Allows monitoring of alarm information on a separate SCADA page.

Steps:

1. Create two pages: a main page and an alarm page.
2. Define a page transition using a button component. Enable the Show Alarms option on the button.

Figure 82 Defining Go to Page Function on Button

3. Add components to the alarm page to monitor alarms.

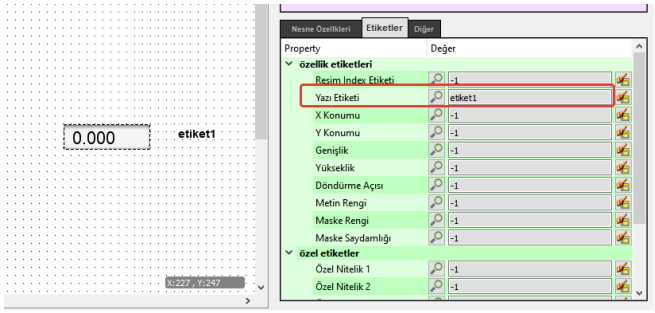

Figure 83 Associating Alarm Tag with a Component

4. When an alarm occurs, an exclamation mark appears on the button.

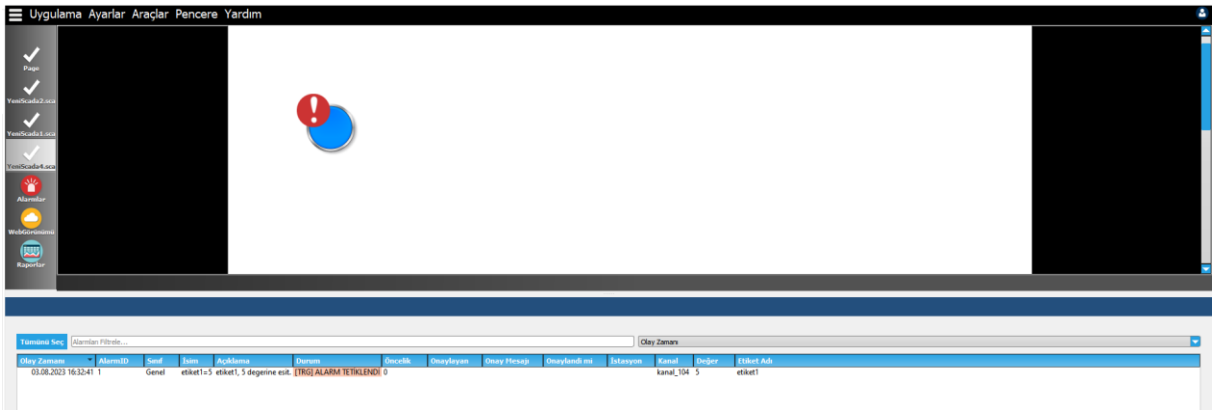

Figure 84 Monitoring Page-Based Alarms via Component

Note: This method only allows monitoring; acknowledgment is not possible.

6.2.3 Monitoring Alarms on the Map Page

Alarms can be monitored on GIS-based map pages using markers.

Steps:

1. Create a map page and enable the Geo View feature.

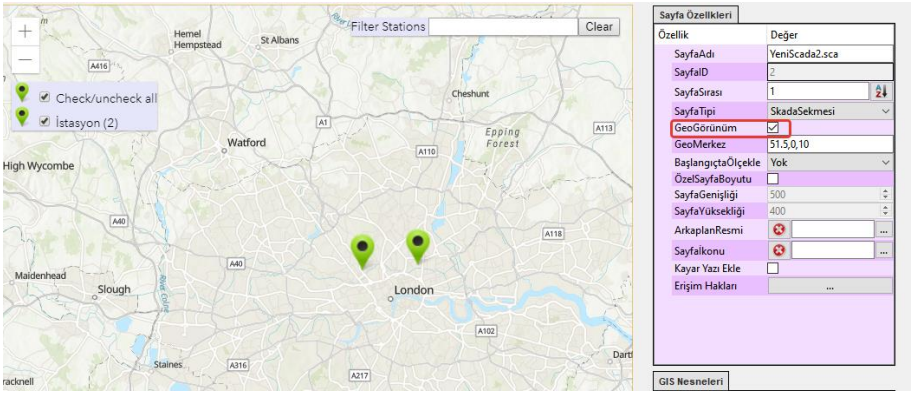

Figure 85 Enabling Geo View for Map Page

2. Add a marker.
3. Assign the alarm-related page to the marker and enable the Show Alarms checkbox.

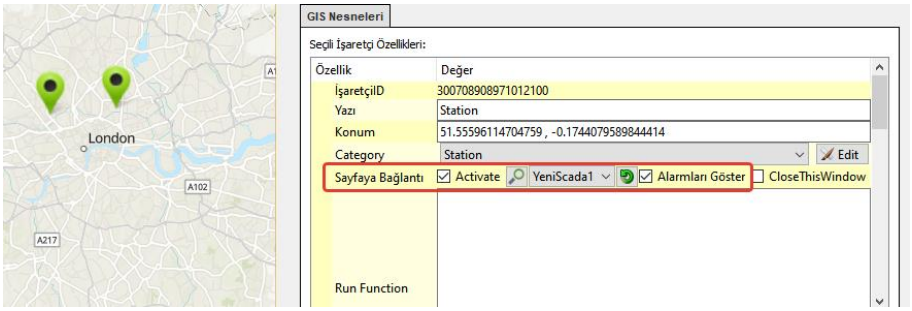

Figure 86 Enabling Show Alarms Option on Marker

4. Add an alarm monitoring component to the relevant SCADA page.
5. When an alarm occurs, the marker will flash with an exclamation icon.
6. The number of alarmed markers is displayed at the top-left of the Client screen.

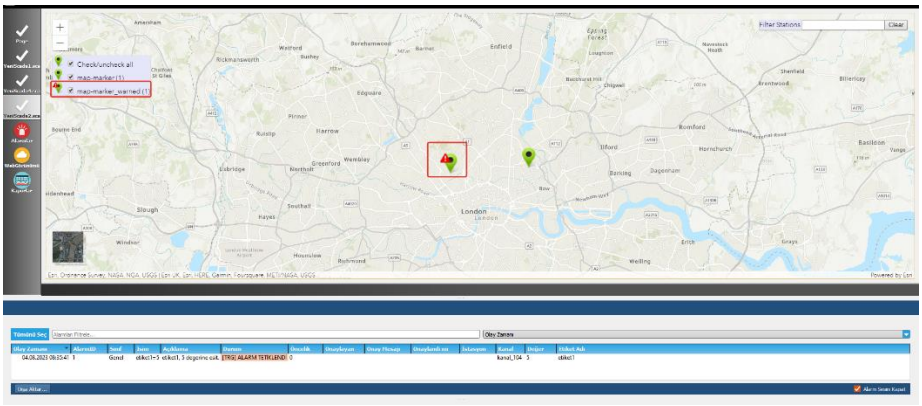

Figure 87 Monitoring Active Alarms from the Map Page

Note: Alarm icons on the map are for monitoring only; they cannot be acknowledged.

7 ViewPLUS SCADA Map Page

7.1 Designing a Map Page

The ViewPLUS SCADA application supports geographic map infrastructure. With this feature, map-based SCADA pages can be created, and the field status can be monitored using shape and color changes based on tag values.

Geo View Feature

By selecting Page Properties > Page Type from the top-right of the ViewPLUS SCADA Editor page, Geo View can be activated to display the page as a map view.

The map infrastructure works integrated with Google Maps or ArcGIS.

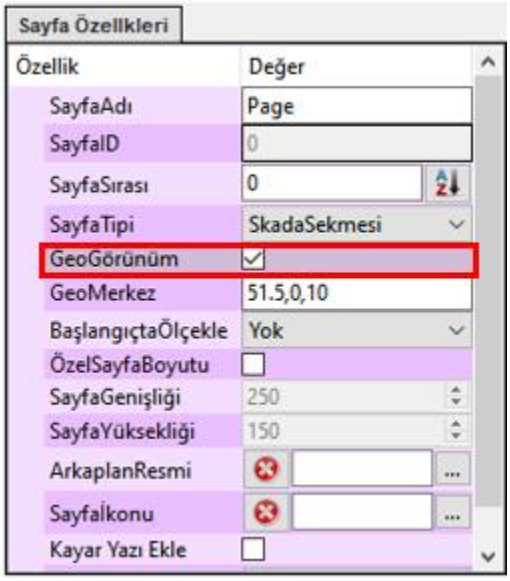

Figure 88 Geo View Option

Using Google Maps

- A Map API key is required.
- Once the key is set, the “For development purposes only” watermark disappears.

Note: The editor must be restarted.

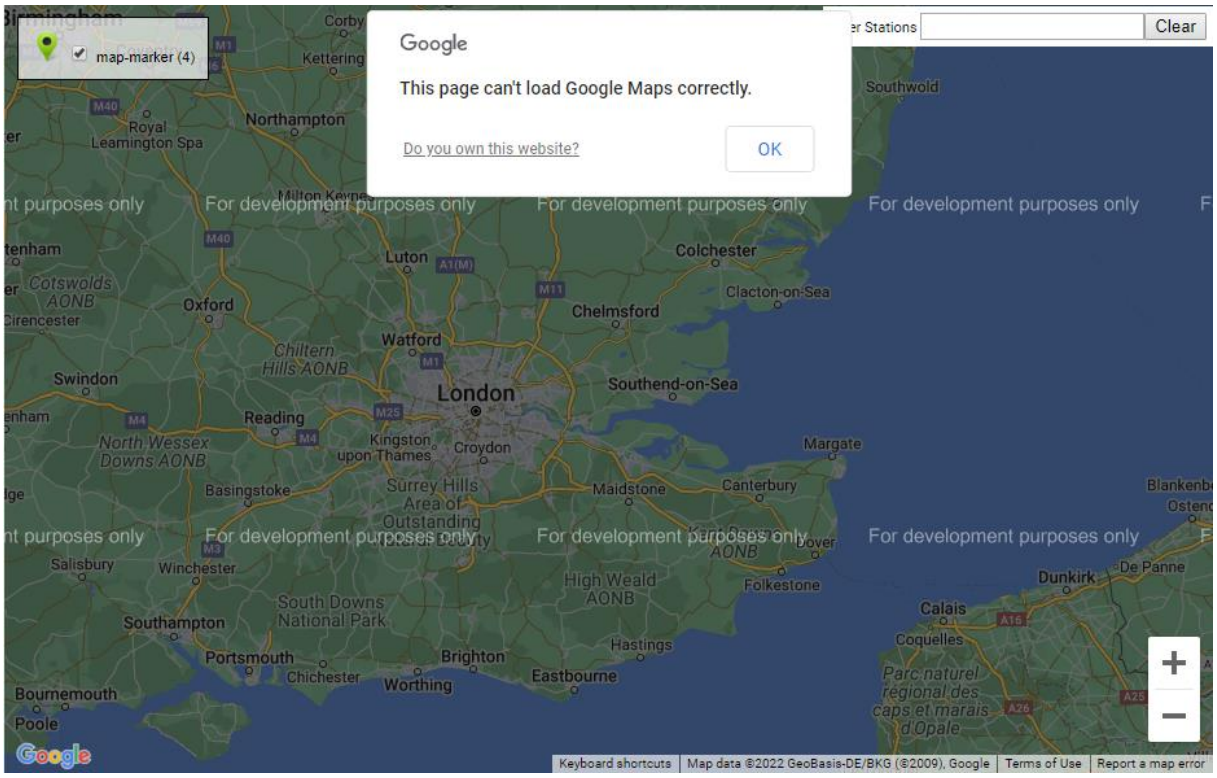

Figure 89 Google Maps View

Definition path:

> Projects > Project Configuration > GIS Settings > GIS Provider: Google

> Projects > Project Configuration > GIS Settings > Map API Key

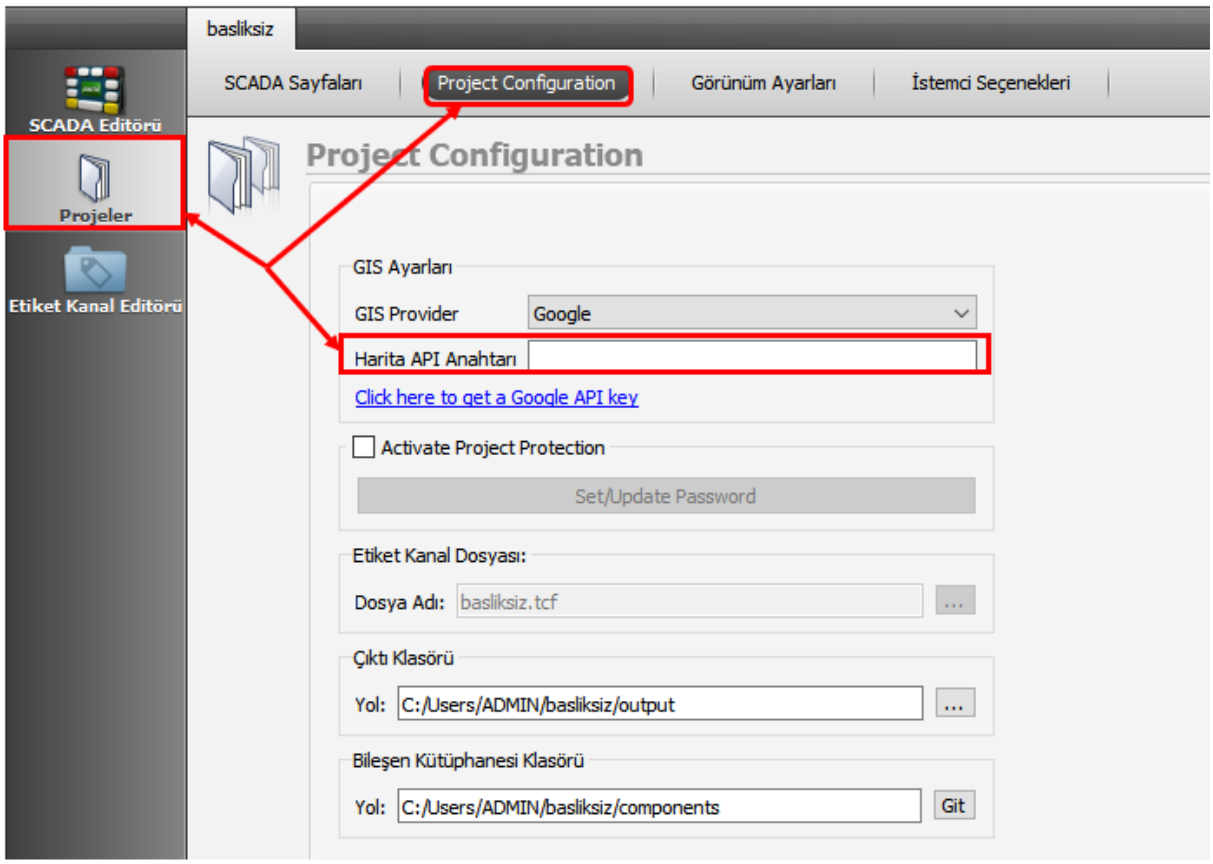

Figure 90 Map API Key Definition

Using ArcGIS

- No API key is required.
- It is free of charge.

The definition is done from the same menu by selecting ArcGIS.

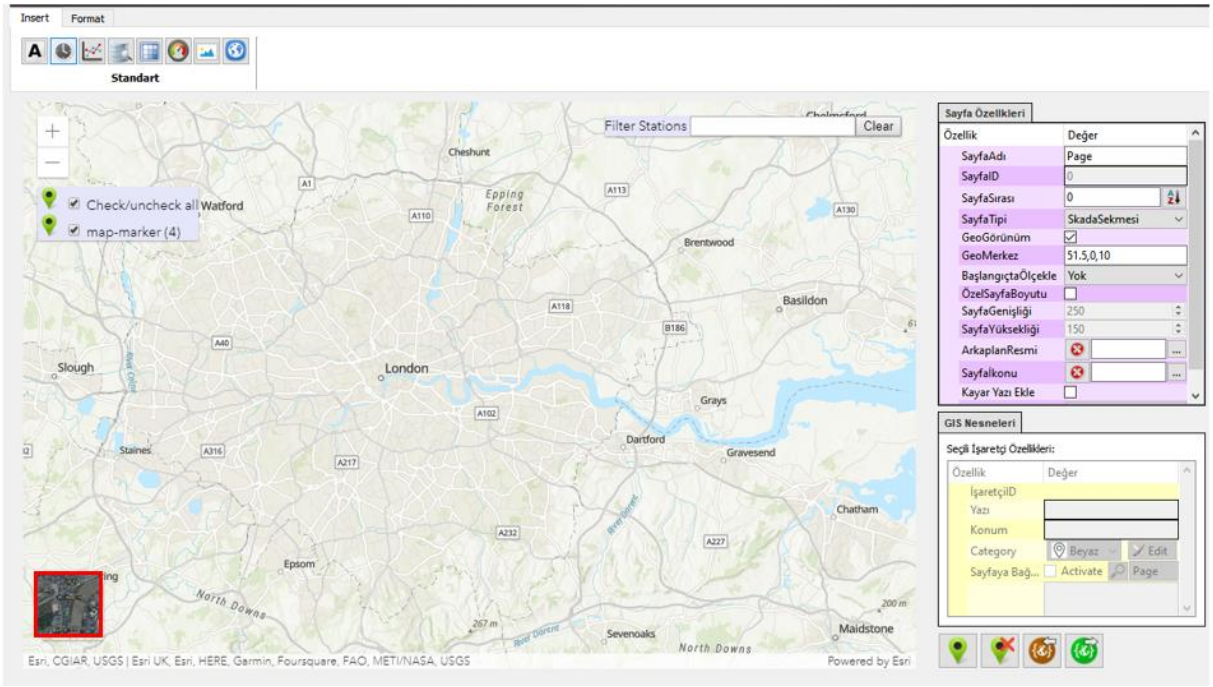

Figure 91 ArcGIS View

7.2 Adding Icons to the Map Page

7.2.1 Adding an Icon

1. Click Add Marker in the SCADA editor.

2. Mark the relevant location on the map.

3. After adding the marker, you can interact with it.

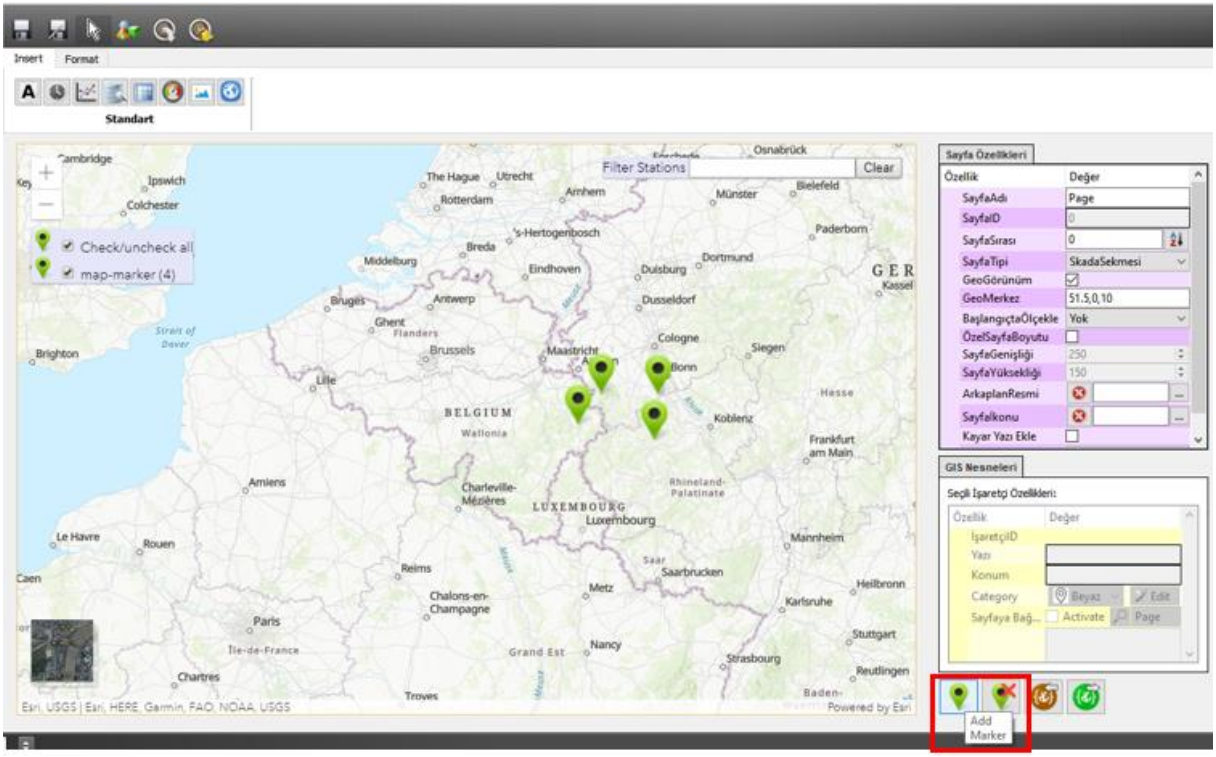

Figure 92 Adding an Icon to the Map Page

7.2.2 Creating a Category

1. Click on the marker.

2. Go to GIS Objects > Selected Marker Properties > Category > Click Edit.

3. Click the Add New Category option to define the category name and icon.

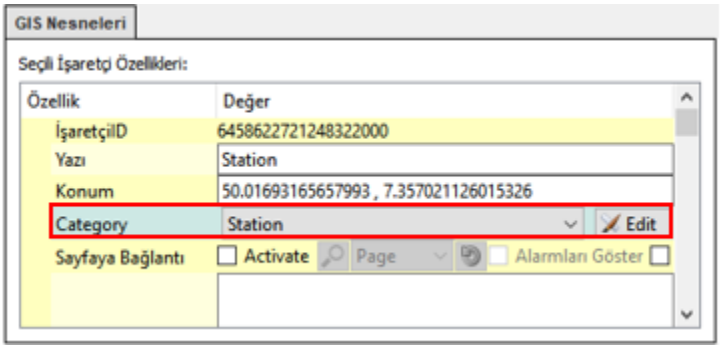

Figure 93 Category Creation Page

Click the Add New Category button, enter the category name, and upload the image. This way, map icons can be used in different colors and shapes as indicators for alarms or other conditions.

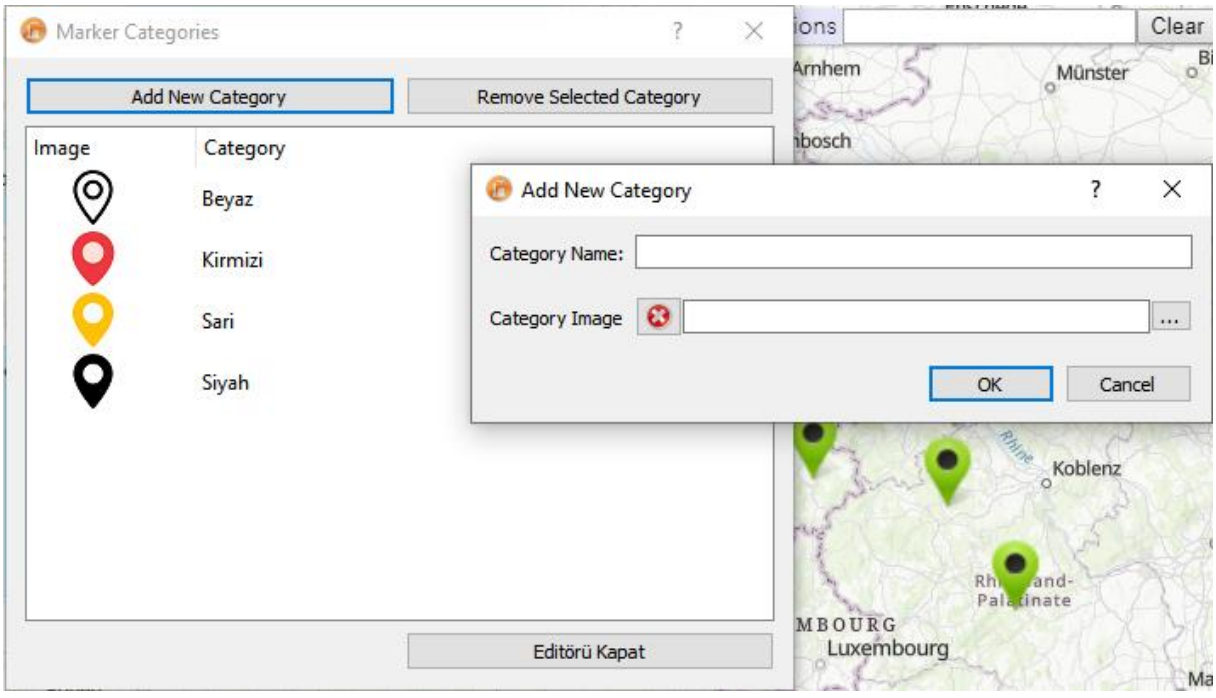

Figure 94 Example Category Creation

7.2.3 Linking Tags with Categories (Run Function)

> GIS Objects > Selected Marker Properties > Run Function

You can define the following example function for each icon:

```
function onCheckMarkerIcon() {
   var $Category ="White";
   if (${1} == 1)
      $Category="Yellow";
   else if(${1} == 2)
      $Category="Black";
   else if(${1} == 3)
      $Category="Red";
   return $Category;
}
```

- ${1}: Represents the value of the tag with ID 1.
- $Category: Represents the category name.

The category is changed conditionally, and the icon is displayed with the corresponding category visual.

7.2.4 Application Results

Based on the defined categories and conditions, the icons on the map dynamically change in color/visuals. This allows centralized monitoring of field device statuses via the map.

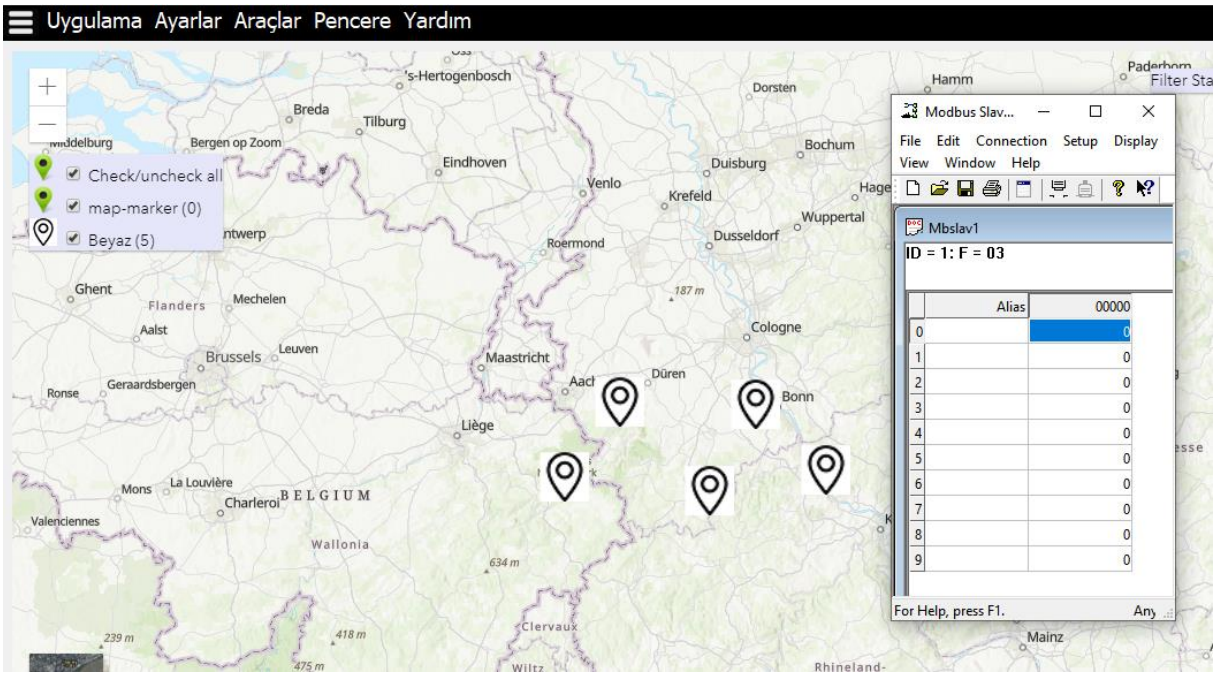

Figure 95 Example of Displaying White Marker When Value is 0

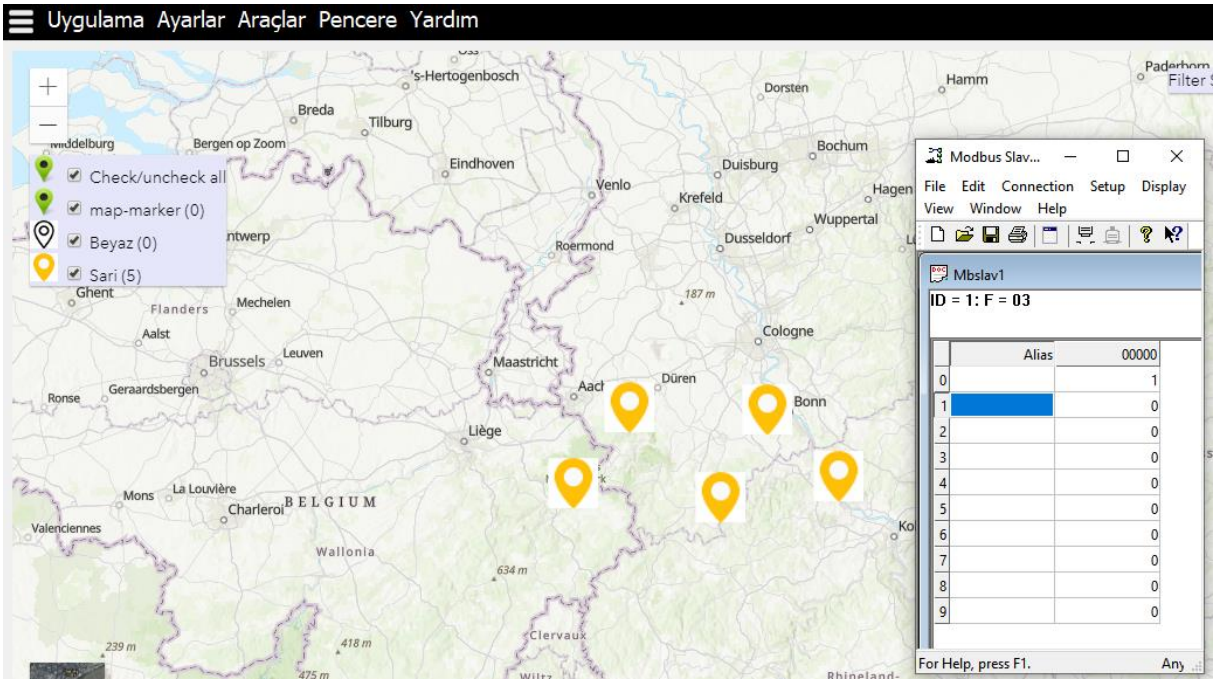

Figure 96 Example of Displaying Yellow Marker When Value is 1

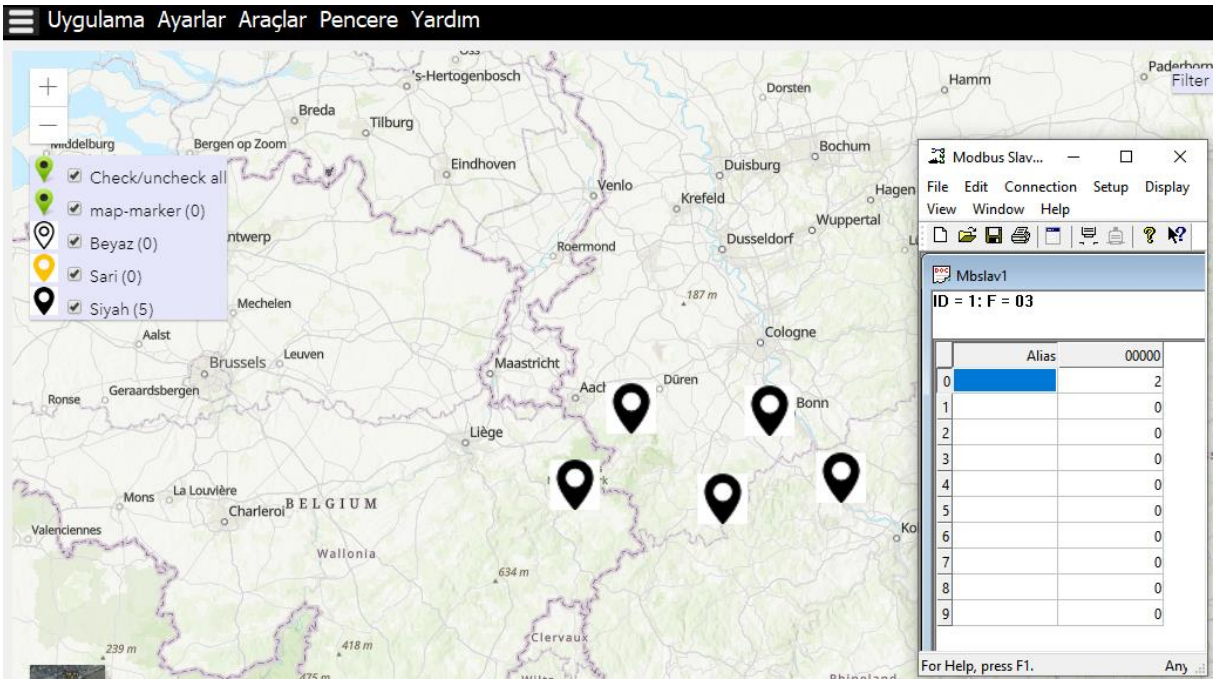

Figure 97 Example of Displaying Black Marker When Value is 2

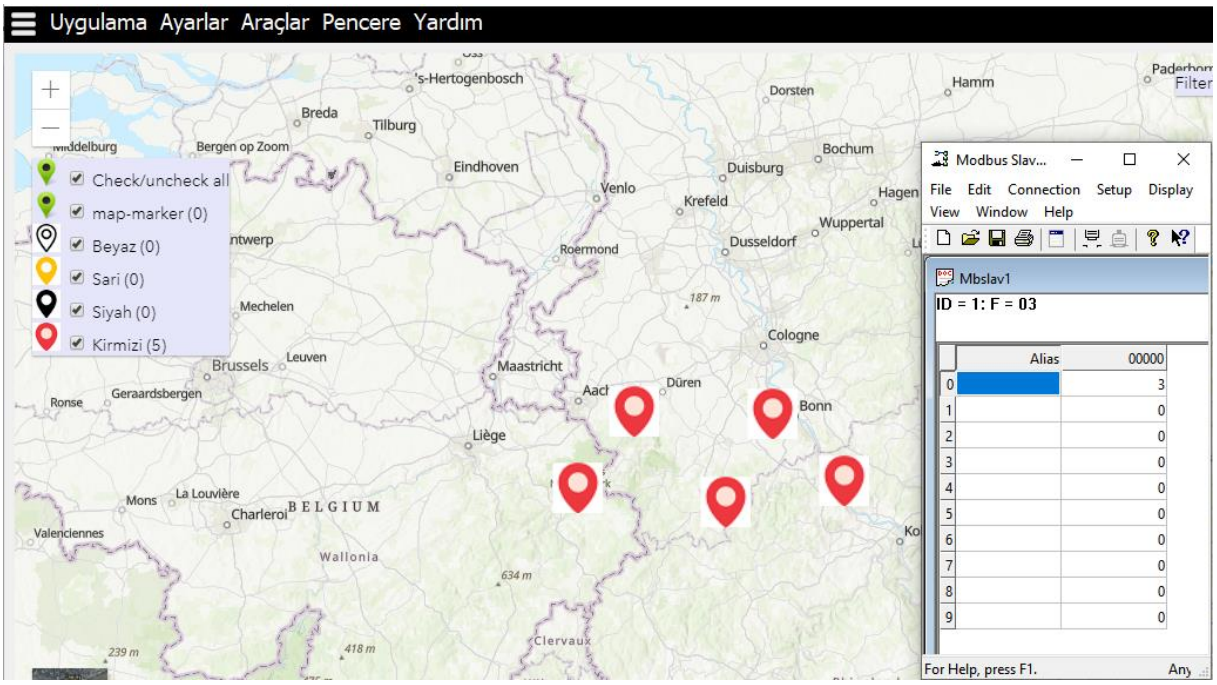

Figure 98 Example of Displaying Red Marker When Value is 3

8 SCADA Server

The server module of ViewPLUS SCADA software is the main component that collects data from field devices and transfers it to clients. To operate, this server must be associated with a SCADA project.

When the server is started, it establishes a database connection, begins receiving data from field devices, and provides a data service through a port that clients can connect to.

8.1 Server Main Screen

When the ViewPLUS SCADA server is started, it presents a graphical interface to the user. This interface provides status information, protocol driver monitoring, alarm control, and simplified connection management.

8.1.1 Top Menu Options

The menu bar at the top of the SCADA server provides quick access to essential operations, system settings, and diagnostic tools.

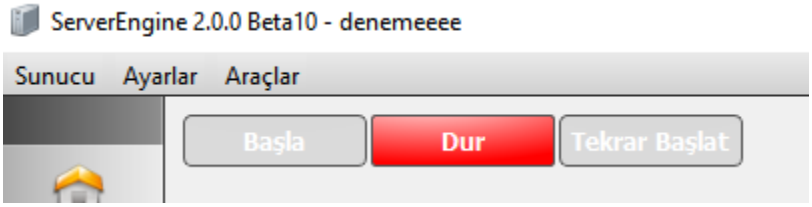

Figure 99 Top Menu Options

8.1.1.1 Server Menu

This menu allows you to directly control the server's operational state:

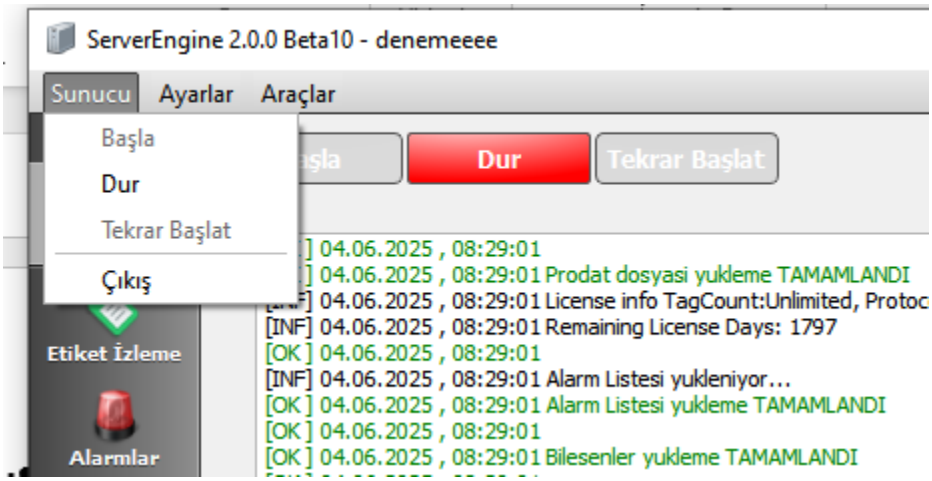

Figure 100 Server Menu

Start: Launches the server and activates data communication.

Stop: Stops the server and terminates all device and client connections.

Restart: Stops and restarts the server to reload system configurations and drivers.

Exit: Closes the server application.

8.1.1.2 Settings Menu

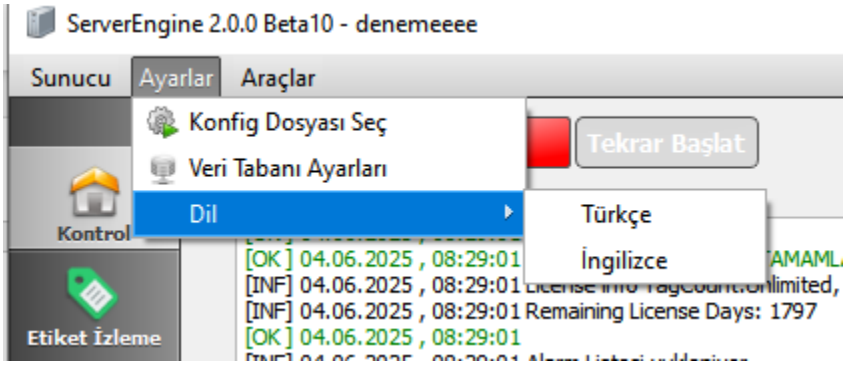

Figure 101 Settings Menu

Select Config File: Allows you to manually choose a .ini configuration file.

Database Settings: Configure database connection details here (e.g., type, username, password).

Language: Allows switching the SCADA server interface language between Turkish and English.

8.1.1.3 Tools Tab

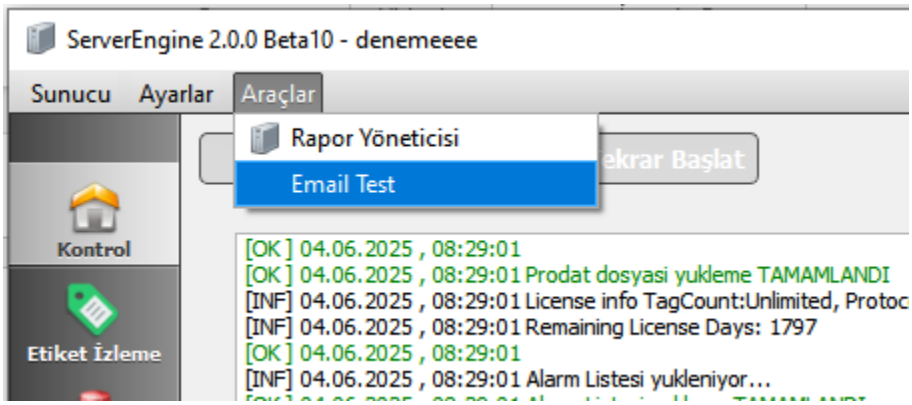

Figure 102 Tools Menu

Report Manager: Opens the integrated SCADA reporting module (e.g., generating reports from recorded data).

Email Test: Allows sending a test message via the SMTP server for email notification setup.

E-mail Test Window

This window is used to test the SMTP configuration to ensure that the server can send emails during alarm conditions.

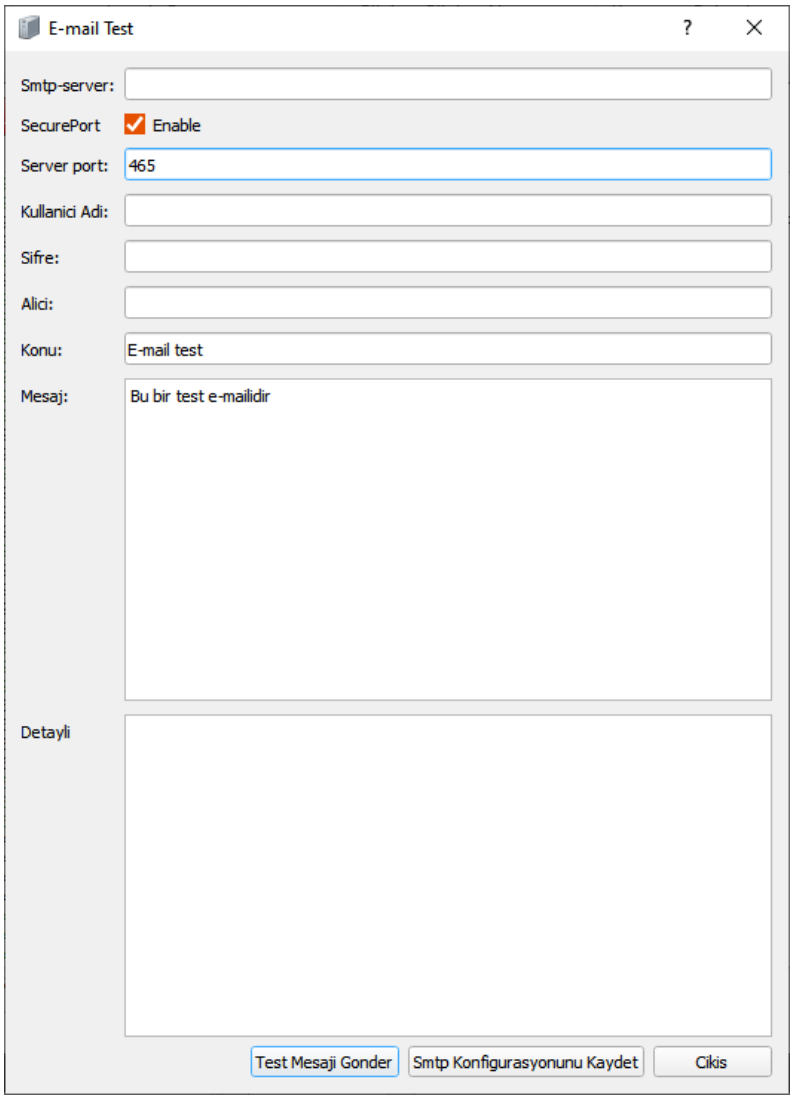

Figure 103 E-mail Test Window

Smtp-server: SMTP server address (e.g., smtp.gmail.com)

SecurePort Enable: Enables the use of a secure port (SSL/TLS)

Server Port: Port number of the SMTP server (e.g., 465)

Username: Email address from which the message will be sent

Password: Application password for the sender email account (an app-specific password should be used for providers like Google)

Recipient: Email address to which the test message will be sent

Subject: Subject of the email

Message: Content of the email

Details: Success or error messages returned by the server are displayed here

Bottom Buttons:

Send Test Message: Sends a test email using the entered settings.

Save SMTP Configuration: Saves the SMTP settings to the configuration file.

Exit: Closes the window.

8.1.2 Control Screen

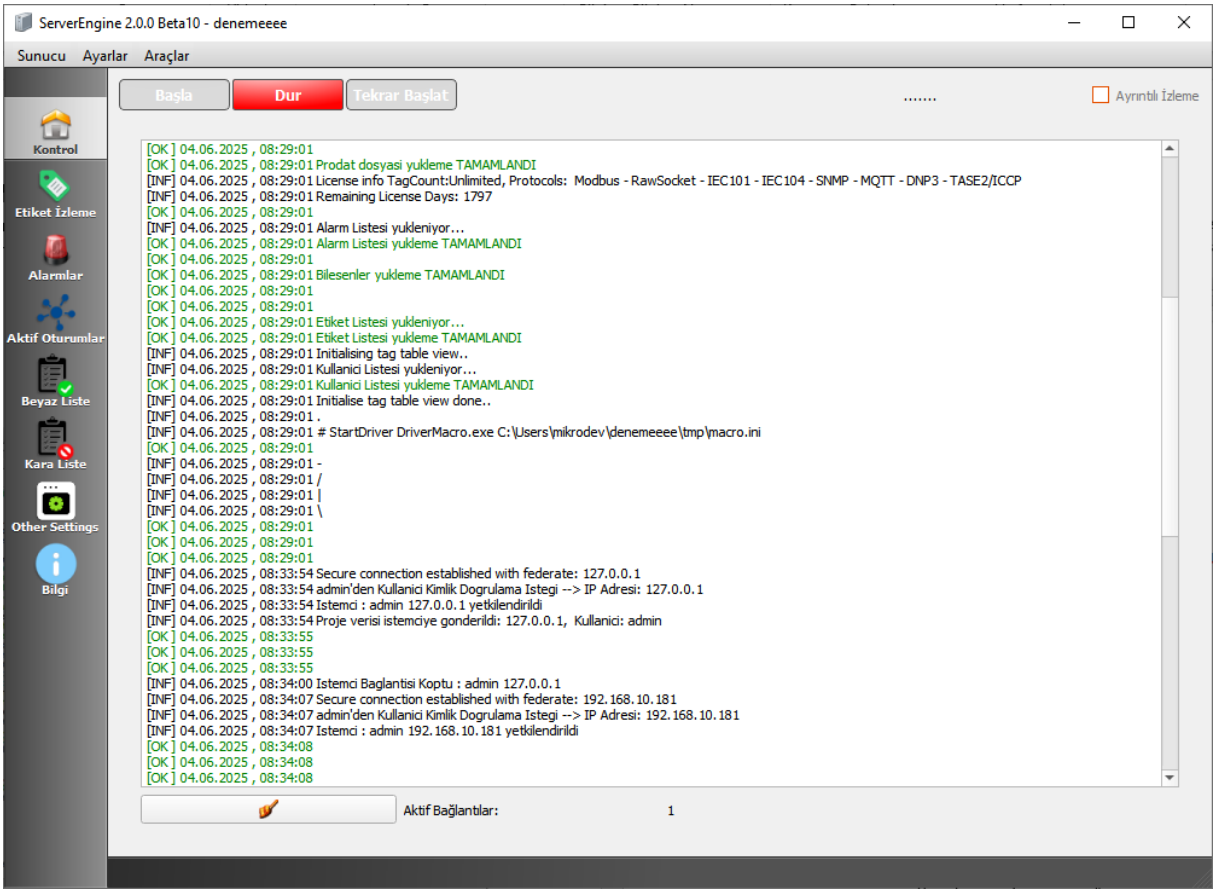

Figure 104 Server Control Screen

- Displays messages related to server startup, shutdown, device connections, database connection, and driver loading.
- Real-time information such as active drivers and data connection statuses is displayed here.
- The server can also be stopped, started, or restarted using the buttons at the top of the page.

8.1.3 Tag Monitoring Screen

This tab is used to monitor real-time values of all tags defined in the SCADA project. The server displays live data read from field devices in this section.

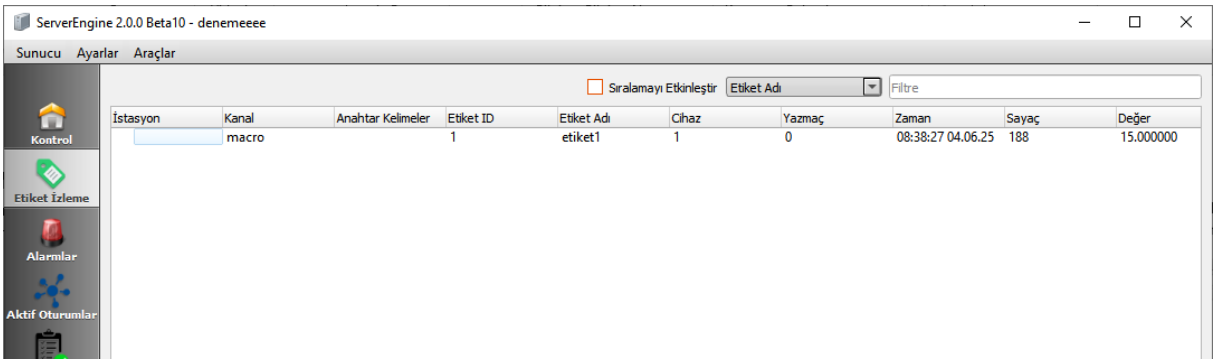

Figure 105 Tag Monitoring Screen

Station: Name of the station the tag is linked to

Channel: Name of the channel the tag is linked to

Keywords: Keywords associated with the tag

Tag ID: System-assigned ID for the tag

Tag Name: Name of the tag (e.g., tag1)

Device: Device address

Register: Address of the data register

Time: Last data read time (date and time)

Counter: Total read count for the tag

Value: Current tag value (example: 15.000000)

8.1.3.1 Tag Monitoring Screen Features:

Real-Time Monitoring: Changes in field data are instantly reflected.

Filtering: The search box at the top right allows filtering by Tag Name, Tag ID, Channel, etc., for quick access in large projects.

Sorting: Enable Sorting option allows sorting by column values (e.g., by value ascending).

8.1.4 Alarms Screen

The Alarms tab allows monitoring of all defined alarm events in the SCADA system. It provides real-time alarm tracking and access to historical alarm records.

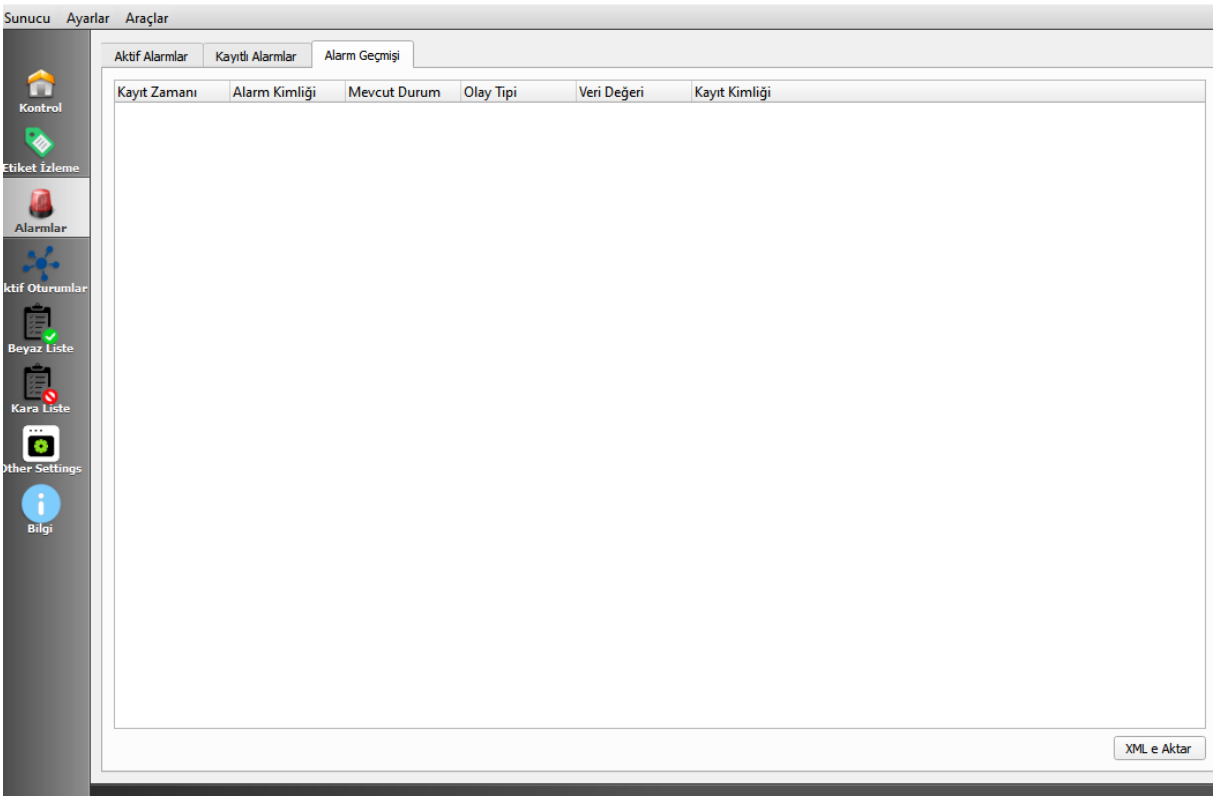

Figure 106 Alarms Screen

Record Time: Date and time the alarm occurred

Alarm ID: Unique system ID of the alarm

Current Status: Shows whether the alarm is active or passive

Event Type: Reason the alarm was triggered (e.g., threshold exceeded, connection lost)

Data Value: Real-time tag value when the alarm occurred

Record ID: System log record number

8.1.4.1 Tabs

Active Alarms: Displays currently active alarms in the system.

Logged Alarms: Displays recorded alarm events.

Alarm History: Lists detailed records of all historical alarm events.

8.1.4.2 Features

Export to XML: Alarm history can be exported using the "Export to XML" button.

Log Analysis: Past logs can be analyzed for identifying system issues.

Time-Based Tracking: Chronological event chains can be analyzed based on date and time information.

8.1.5 Active Sessions Screen

This tab is used to view session information of users connected to the server. It provides visibility into all clients accessing the SCADA system.

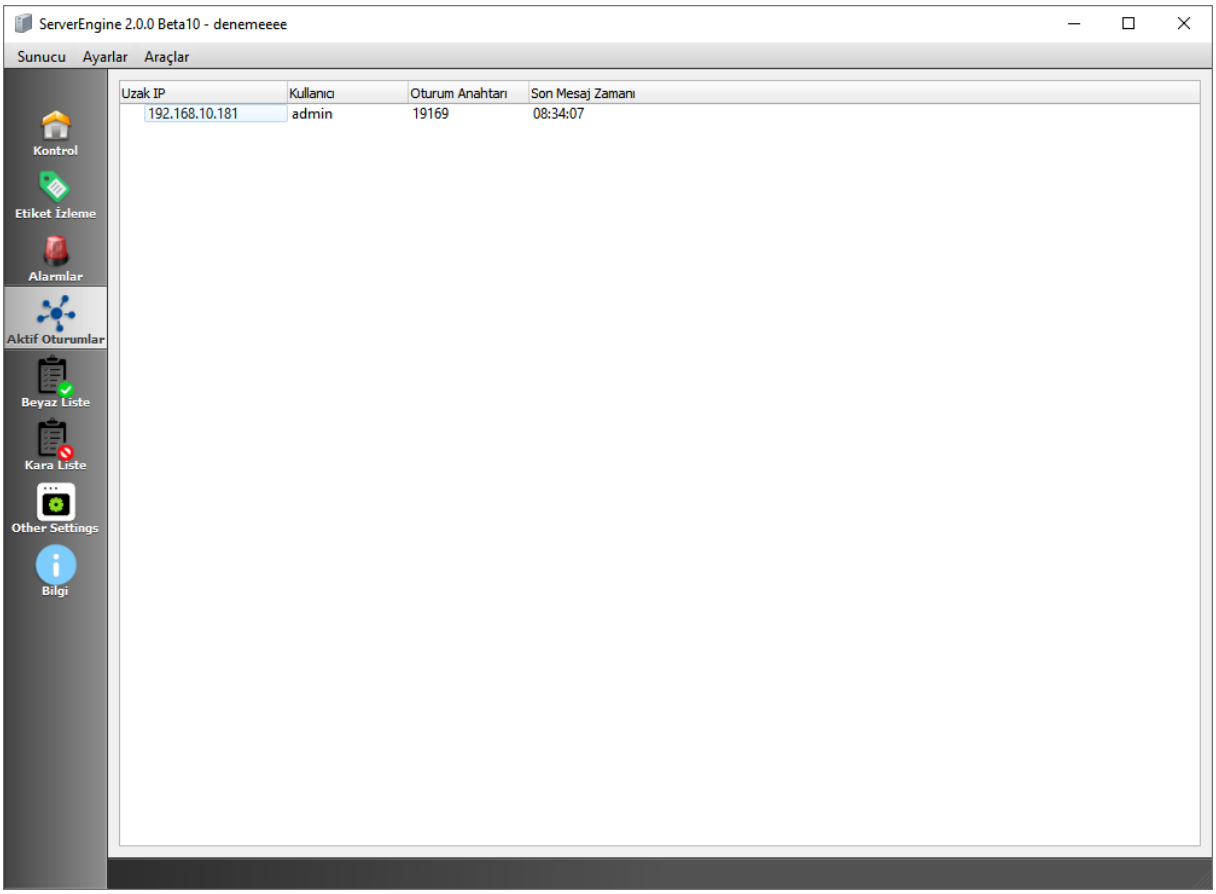

Figure 107 Active Sessions Screen

Remote IP: IP address from which the client is connected

User: Username of the logged-in user

Session Key: Unique session identifier for each connection

Last Message Time: Time when the last message was received from the client

8.1.5.1 Features

Live Monitoring: Connections can be monitored in real time. New connections appear instantly.

Security Tracking: Information such as which users are accessing the system and their IPs can be tracked for security purposes.

Time Information: The last message time indicates whether a session is still active. Sessions inactive for long periods can be tracked by administrators.

8.1.6 Whitelist Screen

The Whitelist tab is used to allow connections to the SCADA server only from predefined IP addresses. This is an important tool to enhance external access security.

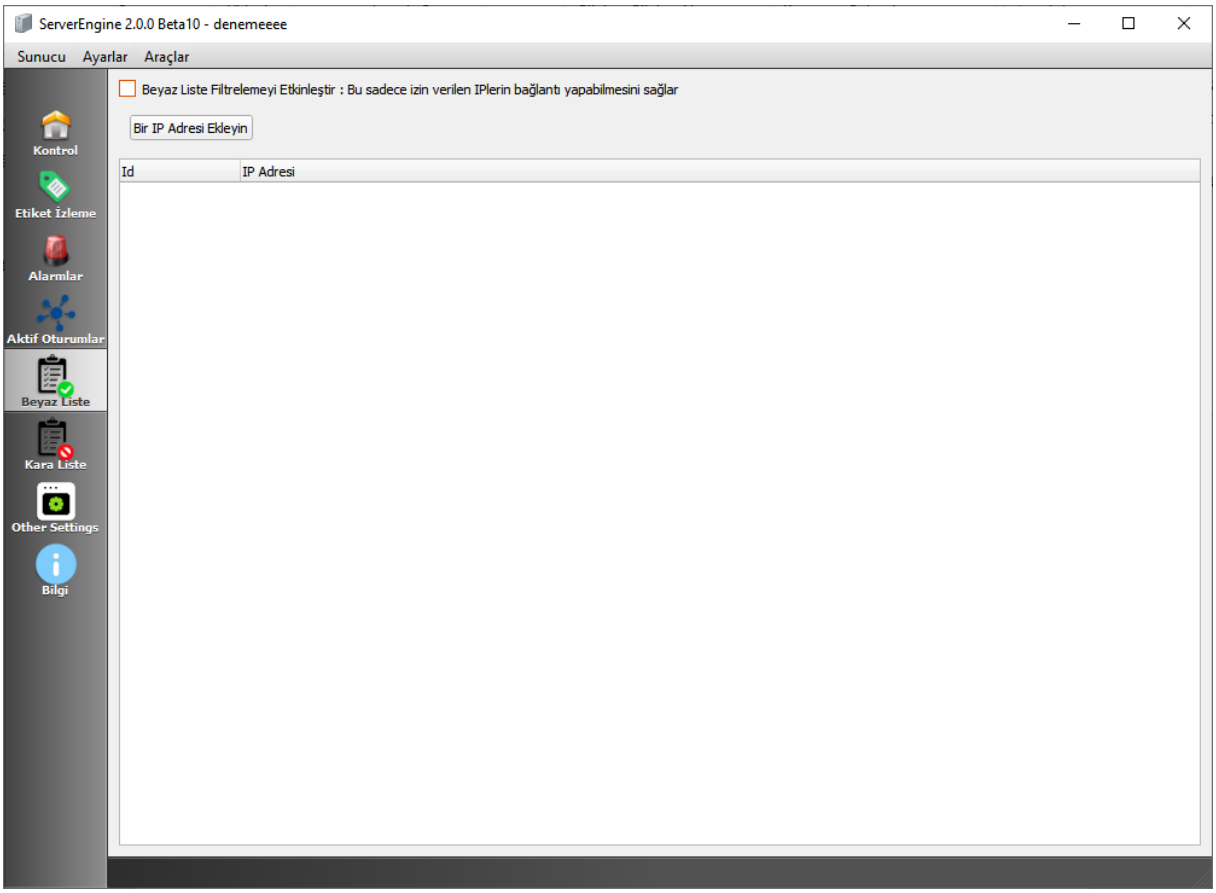

Figure 108 Whitelist Screen

8.1.6.1 Features

Enable Whitelist Filtering: When checked, only IP addresses on the list are allowed to connect. All other IPs are automatically blocked.

Add IP Address Button: Used to add a new IP address. Opens a window where the address can be entered and added to the list.

List View: Added IP addresses are displayed in a table with ID and IP Address information.

8.1.7 Blacklist Screen

The Blacklist tab is used to define IP addresses that should be blocked from accessing the server. This feature helps protect against unauthorized access.

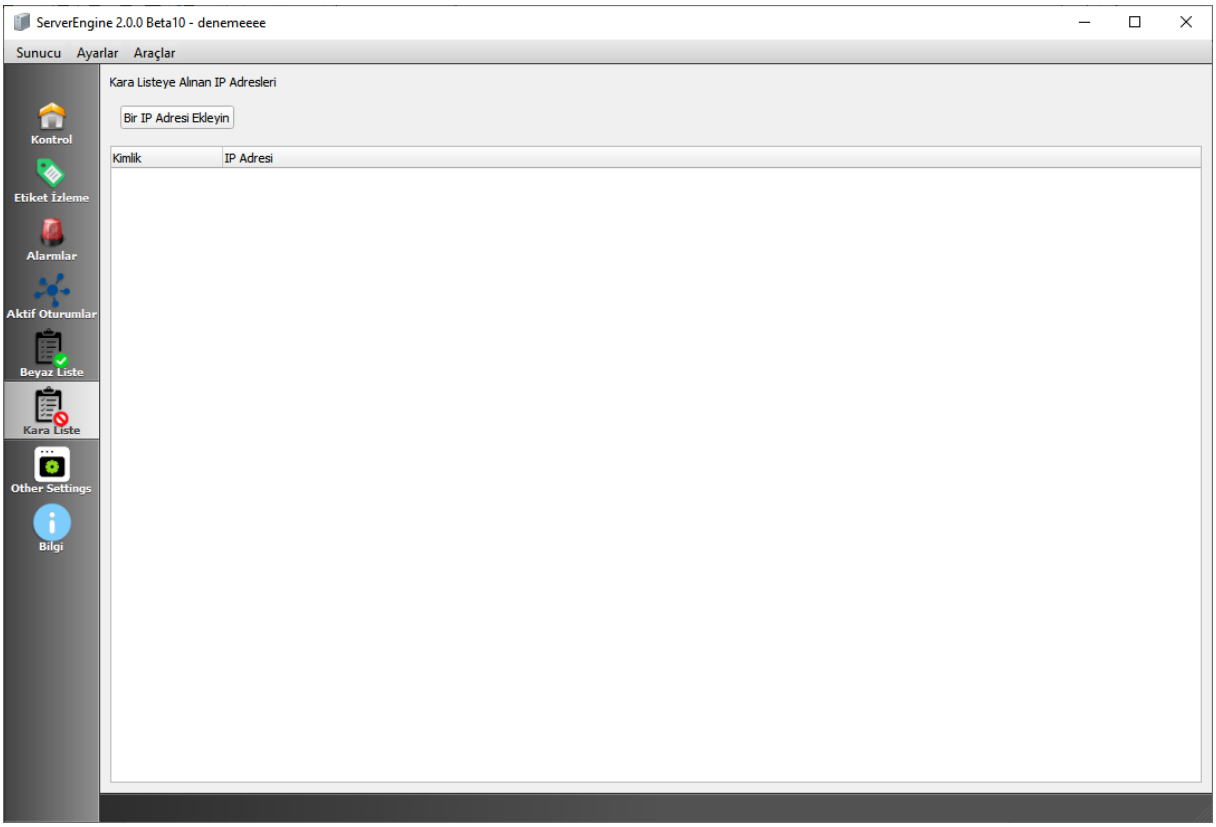

Figure 109 Blacklist Screen

8.1.7.1 Features

Add IP Address Button: Used to add a new IP address to the blacklist. Enter the IP in the window to block it.

Blacklisted IP Addresses: These IPs are completely blocked from accessing the server.

List View: Each IP address is listed with an ID number in the table.

8.1.8 Other Settings Screen

The Other Settings tab includes advanced settings for client security, server redundancy, and MQTT bridging.

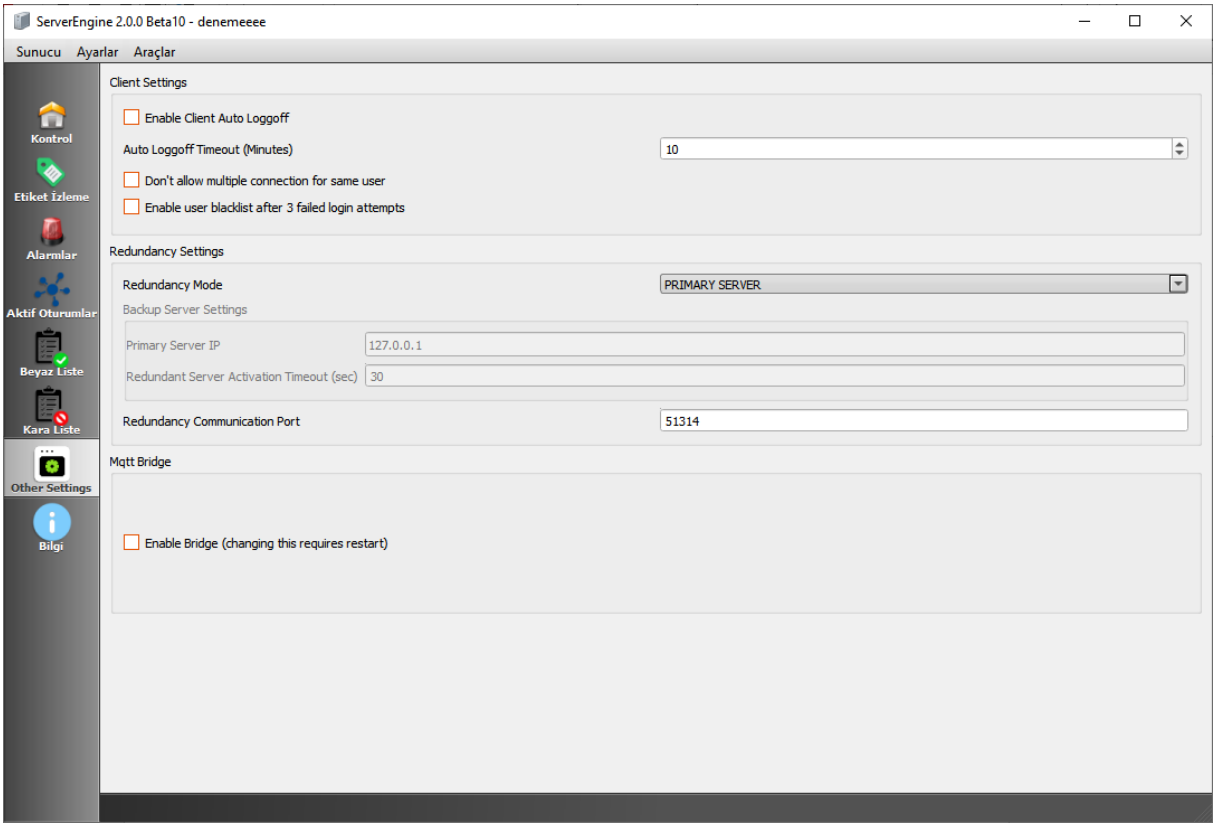

Figure 110 Other Settings Screen

8.1.8.1 Client Settings

This section configures client session and security behavior:

Enable Client Auto Logoff: Ends client sessions that remain idle for a specified period.

Auto Logoff Timeout (Minutes): Sets the timeout period (e.g., "10" = 10 minutes of inactivity).

Don't allow multiple connection for same user: Prevents simultaneous connections using the same username.

Enable user blacklist after 3 failed login attempts: Automatically blacklists the IP after three failed login attempts.

8.1.8.2 Redundancy Settings

This section configures how primary and backup servers operate for uninterrupted SCADA system performance:

Redundancy Mode:

PRIMARY SERVER: The server acts as the main server.

BACKUP SERVER: The server acts as a backup.

Primary Server IP: Specifies the main server IP address if the current one is a backup.

Redundant Server Activation Timeout (sec): Defines the time (in seconds) before the backup takes over after loss of connection to the main server.

Redundancy Communication Port: Port used for communication between main and backup servers (e.g., 51314).

8.1.8.3 MQTT Bridge

Enable Bridge: Enables the MQTT Bridge mode of the SCADA server.

Note: The server must be restarted for this setting to take effect.

This feature allows the SCADA system to exchange data with an external MQTT server.

8.1.9 Information Screen

The Information tab shows the database usage status of the SCADA project running on the server. This helps administrators track how much data has been generated and how much space it occupies over time.

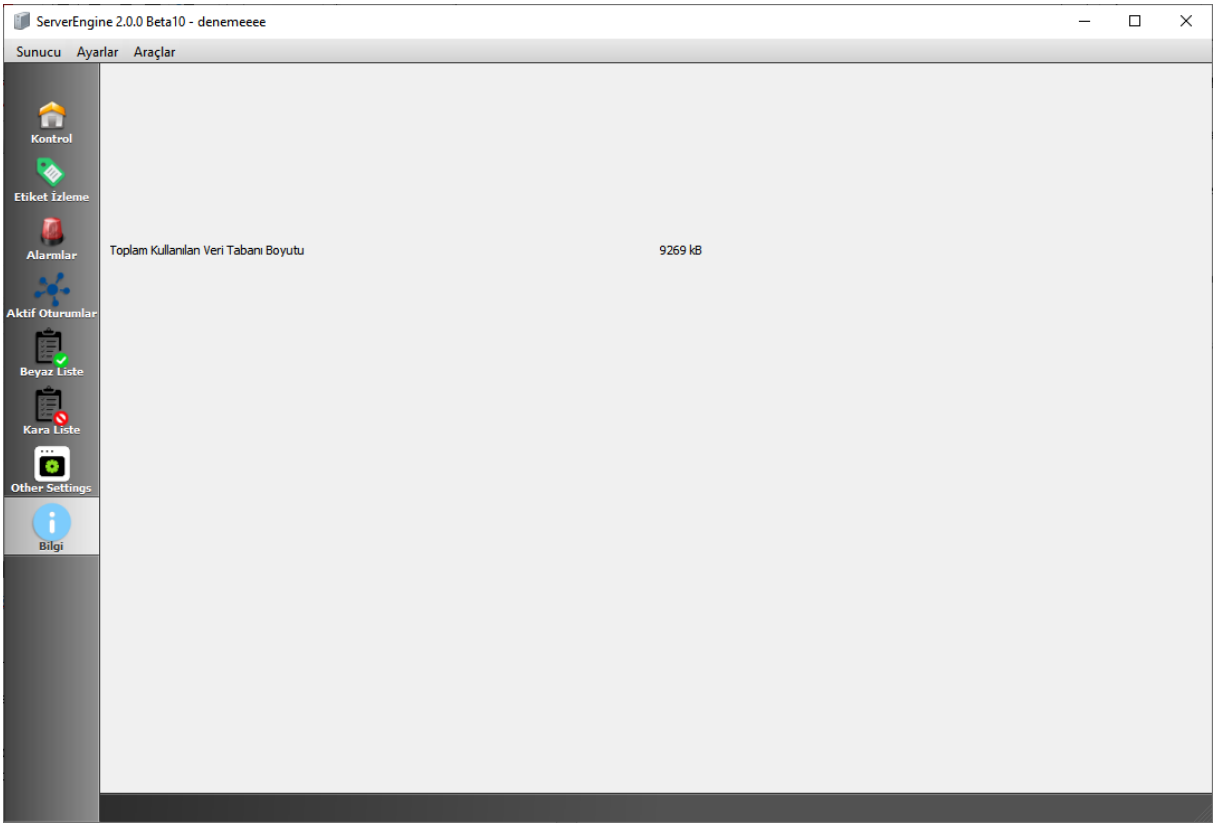

Figure 111 Information Screen

Total Used Database Size: Displays the total size of all logged data. Example: 9269 kB (approximately 9 MB)

8.1.9.1 Features

Database Management: Facilitates tracking of archiving and cleanup needs to prevent database bloat.

Project Monitoring: Useful for long-term monitoring of data growth to plan system resource usage.

8.2 Starting the SCADA Server

To start the ViewPLUS SCADA server, follow these steps:

- Open the SCADA project in the ViewPLUS editor.
- Click Tools > Start Server from the menu.

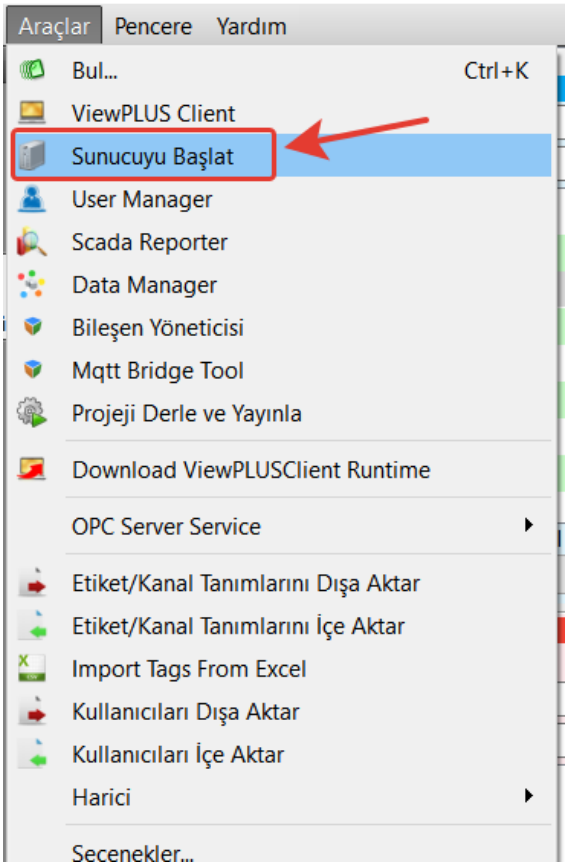

Figure 112 Server Startup Option

The server will open a window to access the database information of the project. You must enter the database username and password here. This step is required only for the first launch and is saved to the configuration file thereafter.

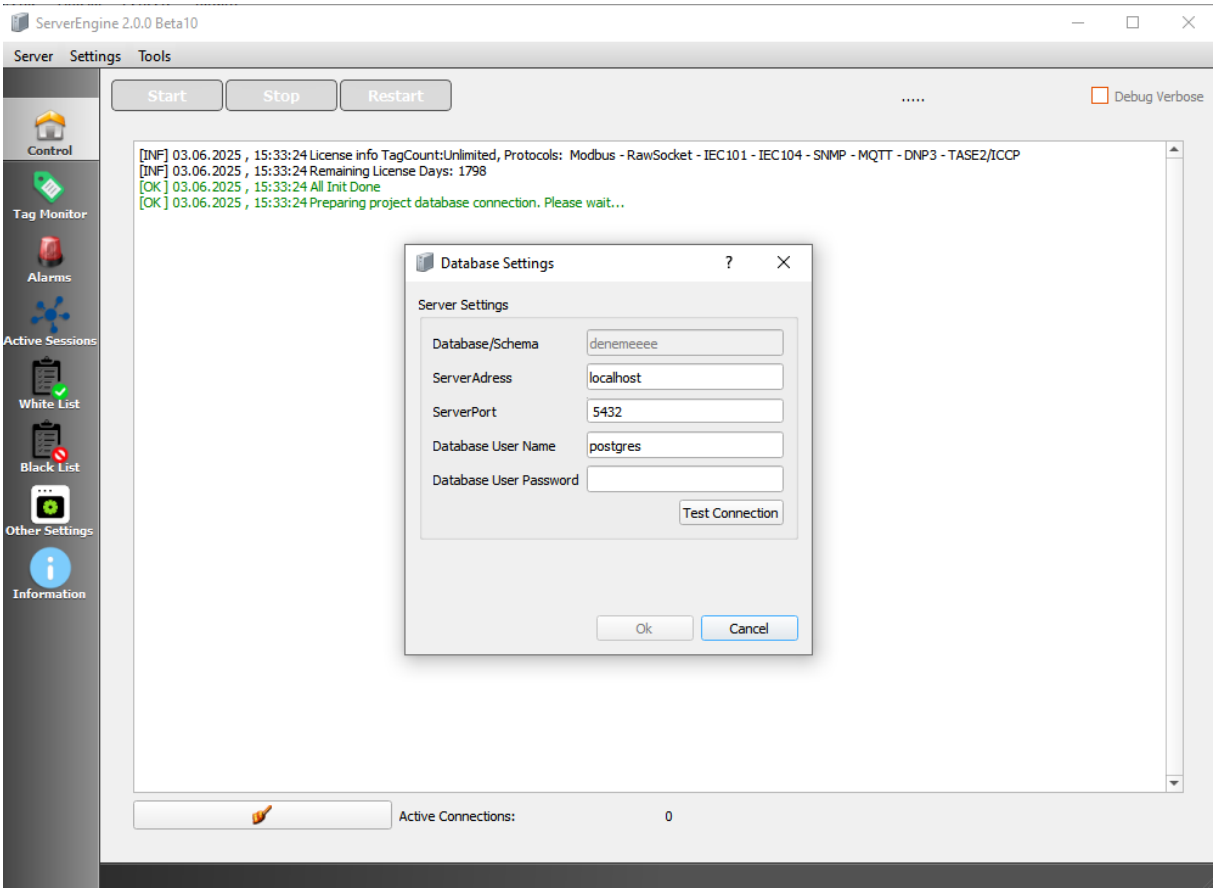

Figure 113 Database Settings Window

- When the server is started, it attempts to connect to field devices and launches a service that clients can connect to.

8.2.1 Starting via Desktop Shortcut

The SCADA server software can also be started independently of the editor using a desktop shortcut.

- First, create a desktop shortcut to the ServerEngine.exe file.

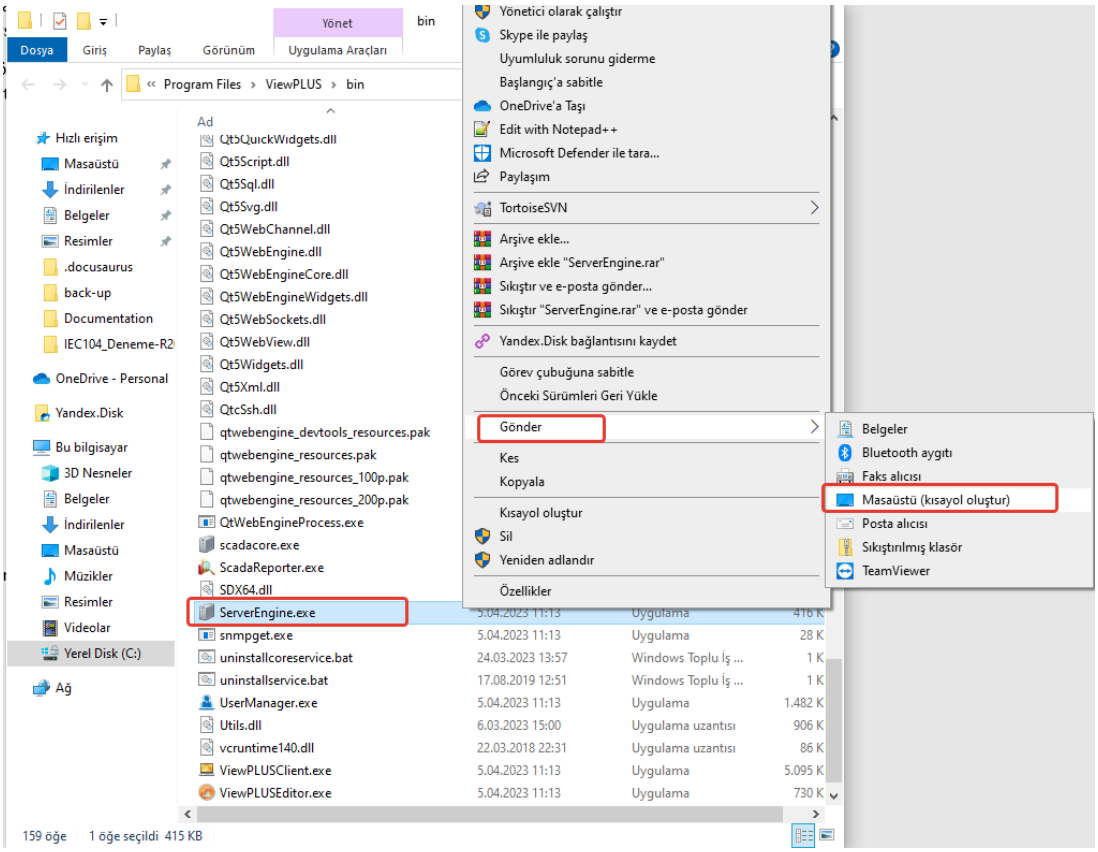

Figure 114 Creating a Desktop Shortcut for ServerEngine.exe

- Then, in the shortcut's properties, add the path of the SCADA project to be run along with the necessary parameters in the “Target” field:
  -dir "C:\<Proje klasör yolu>" -start

  Örnek hedef parametre: "C:\Program Files\ViewPLUS\bin\ServerEngine.exe" -dir “C:\Users\mikrodev\Desktop\zamanlayici_etiketi_test” -start

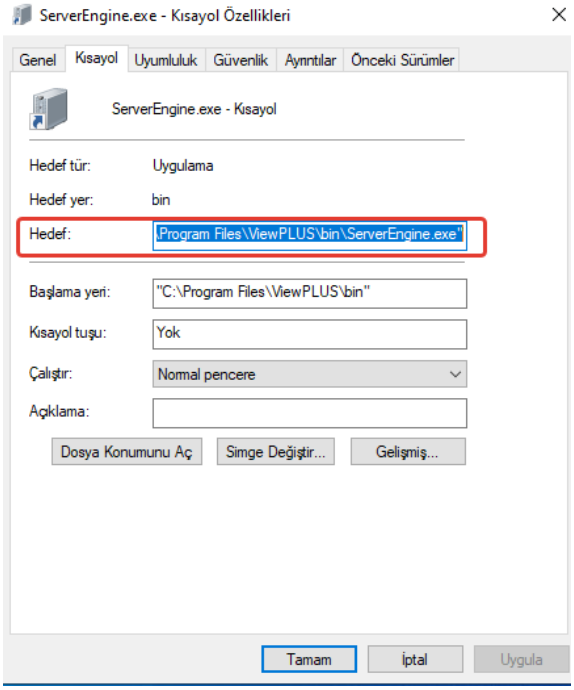

Figure 115 Shortcut Properties Target Parameter

- Double-click the created shortcut to start the server for the specified SCADA project. Once running, the server connects to field devices, opens a port for client connections, and starts logging data to the database.

8.2.2 Automatic Startup (at Windows Boot)

SCADA Server can be configured to start automatically when the computer boots. This is achieved by copying the server shortcut to the Windows Startup folder. In this way, when the user logs into Windows, the SCADA server will automatically launch.

This method is especially ideal for users who want SCADA systems to operate continuously without requiring operator intervention. The same approach can also be used for client applications.

- Locate the previously created ServerEngine.exe shortcut on the desktop.
- Copy this shortcut into the following folder:
  C:\Users\<Username>\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup

This folder contains applications that are automatically launched when Windows starts.

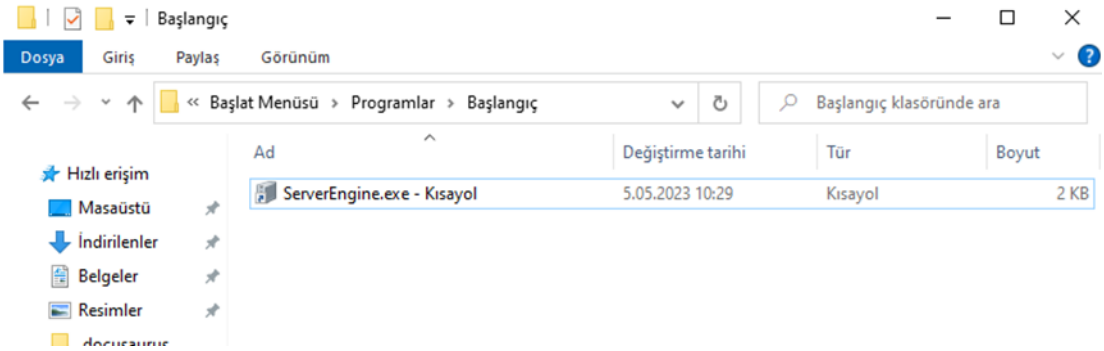

Figure 116 Copying ServerEngine.exe Shortcut to the Startup Folder

8.2.3 Running as a Windows Service

The ViewPLUS SCADA server can be run as a service on the Windows operating system. With this method, the server operates silently in the background and can start automatically even if the system is rebooted without requiring a user to log in. This setup is recommended for systems that demand uninterrupted operation.

8.2.3.1 Installation Steps

Service Installation:

First, the server service component must be installed. Right-click on the installcoreservice.bat file located in the bin folder of ViewPLUS SCADA and select Run as Administrator.

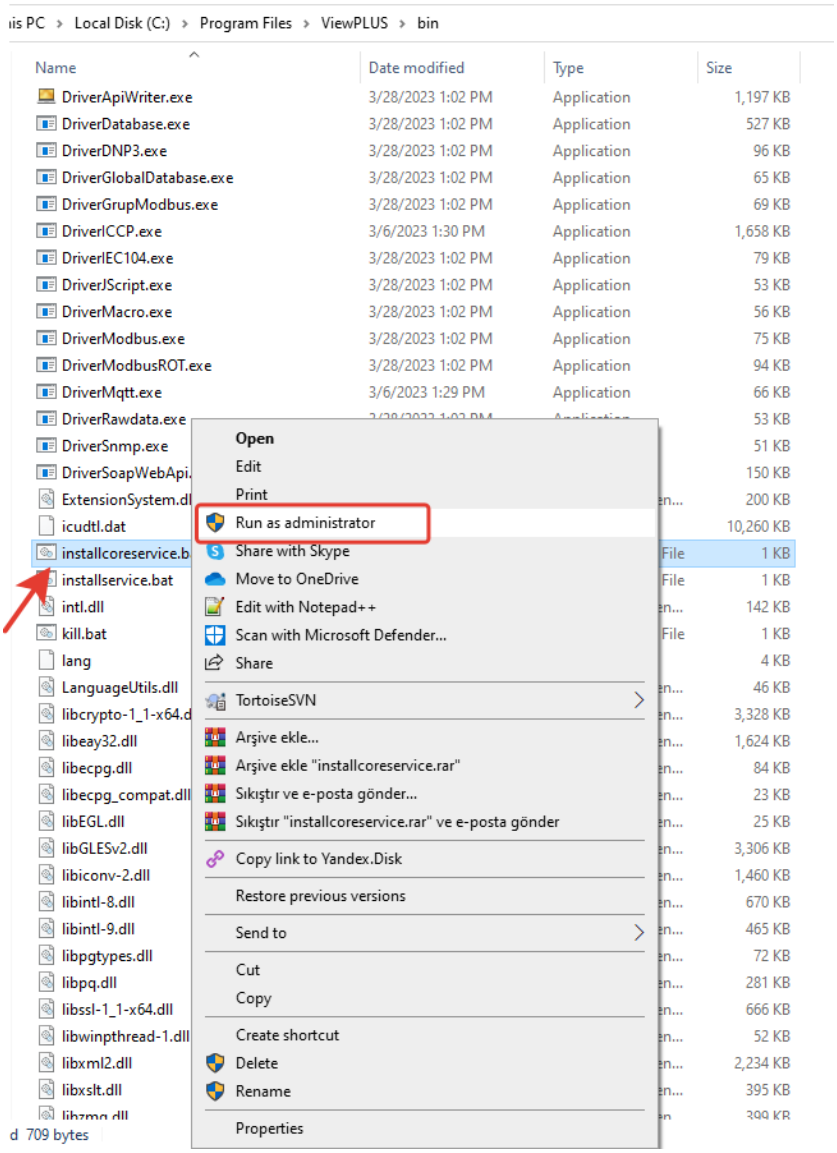

Figure 117 Running installcoreservice.bat File

Defining a System Environment Variable:

To define which project the server will run, a system environment variable must be created:

- Go to the Start menu → open "Edit the system environment variables"
- In the window that opens, click on Advanced > Environment Variables

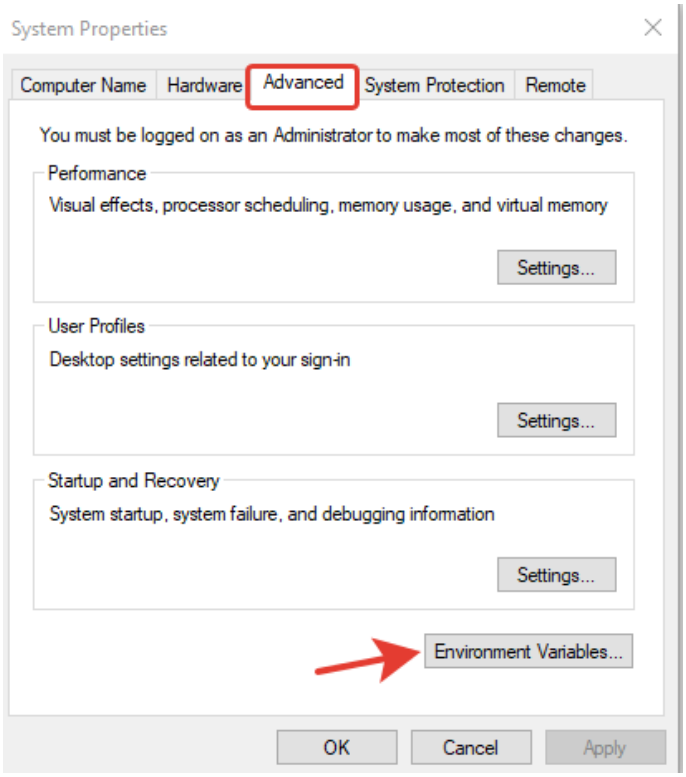

Figure 118 System Environment Variables

- In the "System variables" section, click on New.

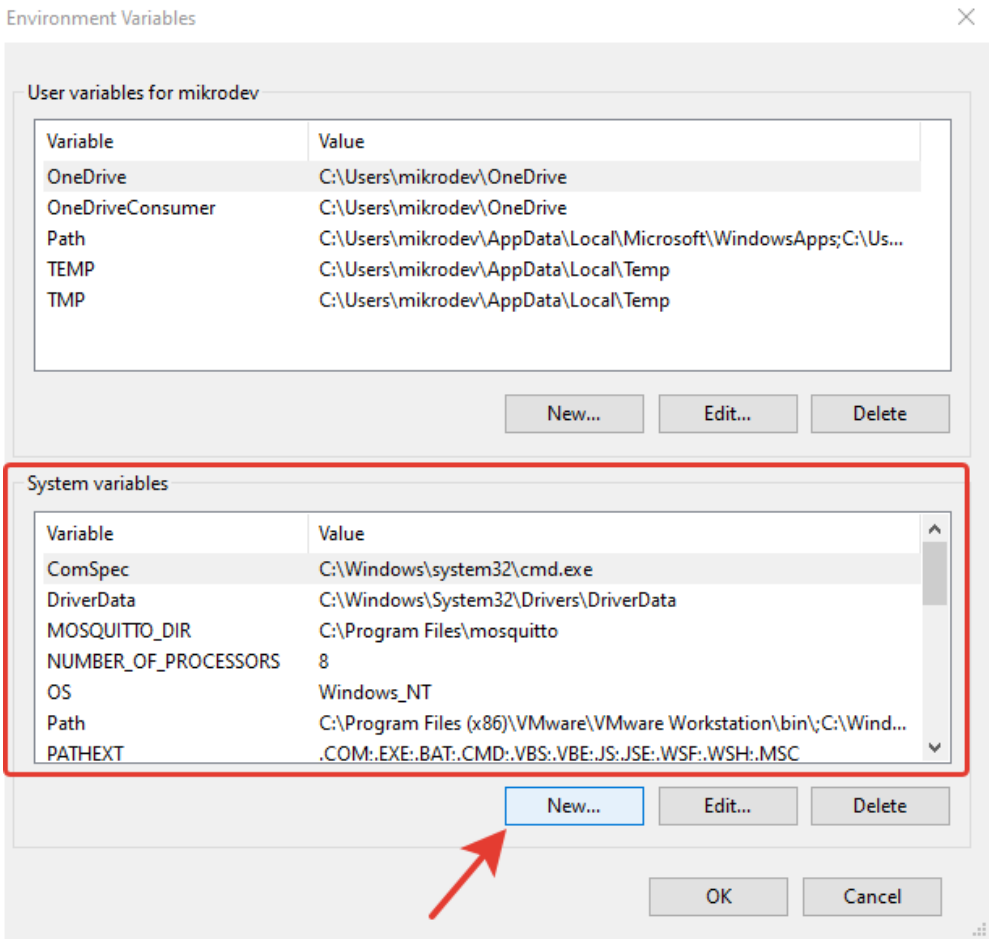

Figure 119 Defining a New System Variable

- Define a new variable as shown below:

Variable name: VPLUS_SCADA_DIR

Variable value: <Full path to the project folder>

Example:

Variable name: VPLUS_SCADA_DIR

Variable value: C:\Users\mikrodev\Desktop\SCADA_Projem

Click OK in all open windows to save the variable.

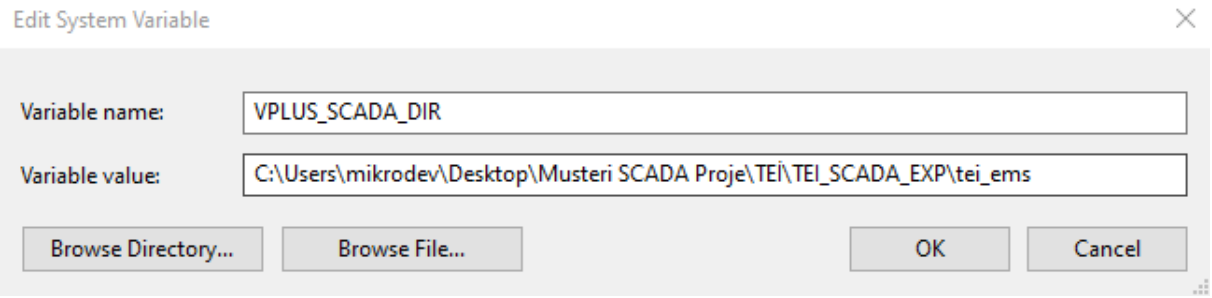

Figure 120 Adding New System Variable

Starting the Service:

Once the installation is complete, go to Task Manager > Services tab.

Find the service named VPlusScadaService, right-click it, and select Start.

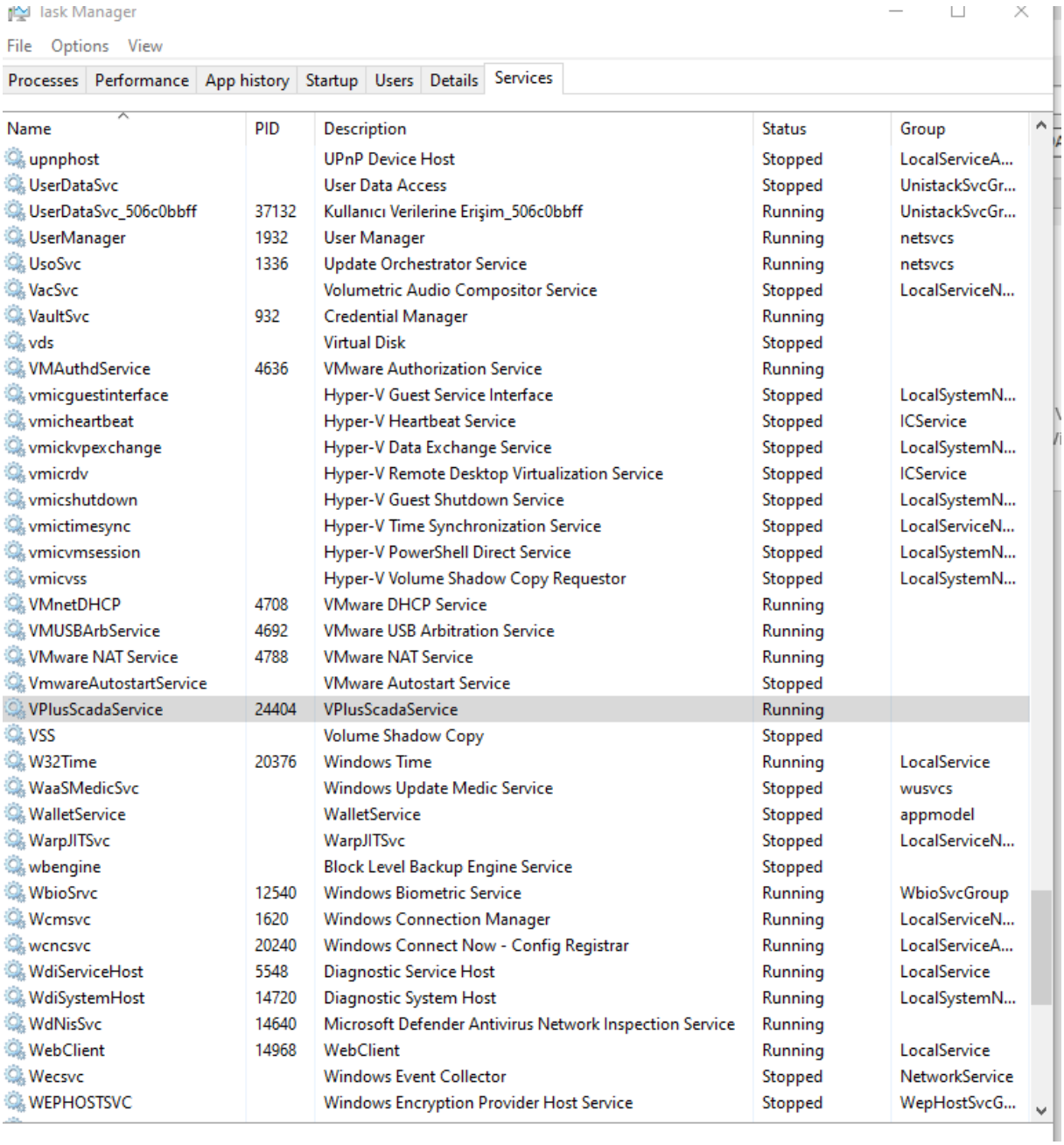

Figure 121 Starting the Windows Service

9 ViewPLUS Scada Client

The ViewPLUS SCADA client allows users to connect to the SCADA server and view and control defined projects.

To establish a connection, a username, password, server IP address, and port number must be entered.

New users can only be defined through the SCADA editor.

Once user authentication is completed, an encrypted connection is established between the client and the server, and the main screen defined by the server is loaded on the client.

The pages and elements displayed on the client are filtered according to the permissions defined for the user.

Navigation to pages for which the user does not have access is not allowed,

and visual components associated with tags that are not within access rights are disabled.

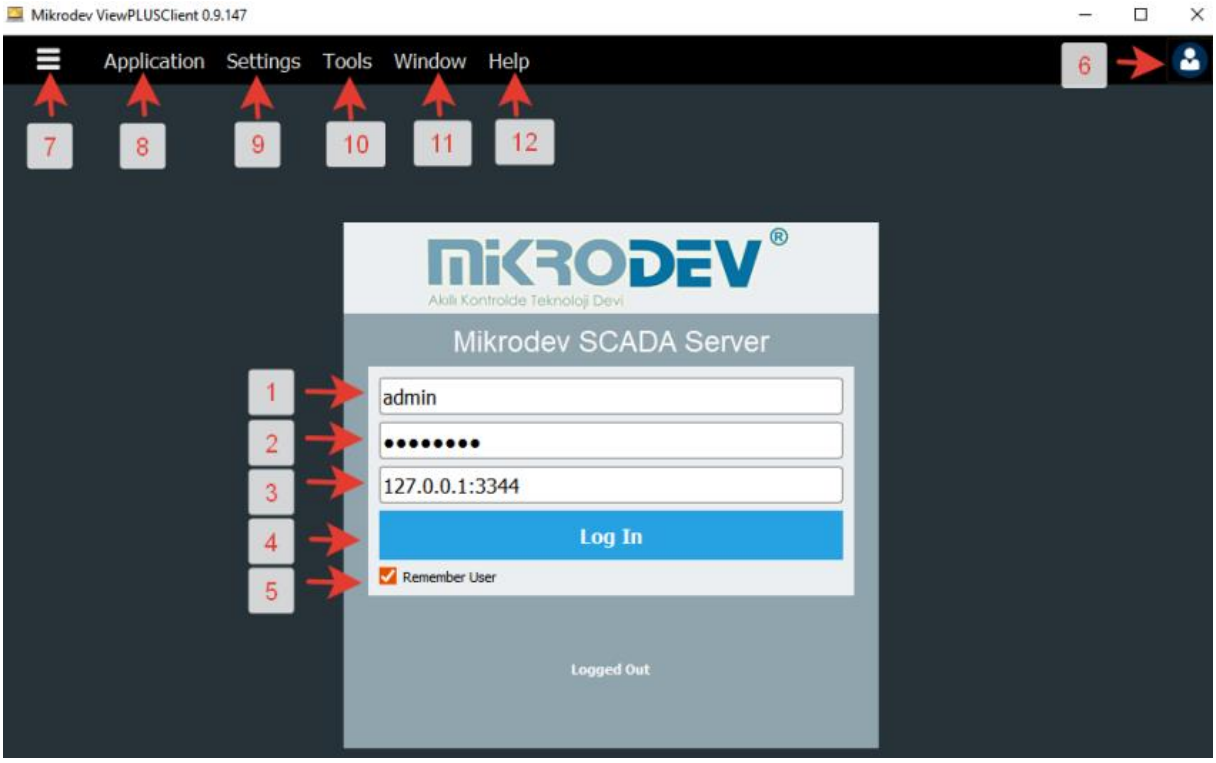

Figure 122 Client Interface

9.1 Basic Interface Elements

No	Component	Description
1	Username	Username defined in the SCADA system
2	Password	Password of the user authorized to log in
3	IP and Port	IP address and port of the server computer
4	Login	Connect to the server using login credentials
5	Remember User	Saves the last used user information
6	Profiles	Previously saved user profiles
7	Toolbar	Manages the visibility of the left panel
8	Application	From the application menu, you can save the current user profile, log out of the session, or exit the application completely.
9	Settings	From the settings menu, you can select the language, manage visual indications of connection interruptions, control cache settings, and disable the alarm sound.
10	Tools	Provides quick access to the report screen or project pages from the tools menu.
11	Window	From the window menu, you can maximize the screen, split the active page, zoom in/out, or arrange the layout of alarm windows.
12	Help	From the help menu, you can enable debug mode and view version information of the client software.

9.2 Top Menu Items

9.2.1 Application Menu

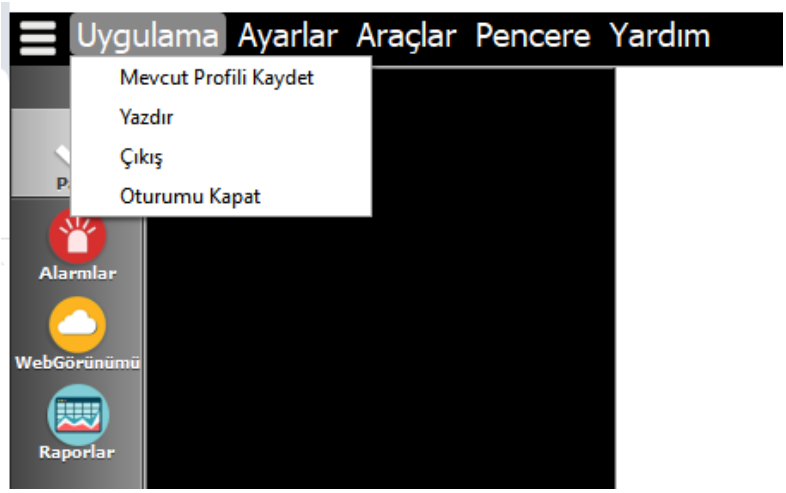

Figure 123 Application Menu

Save Current Profile: Creates a new profile using the session information.

Print: Initiates the print process for the active SCADA page. Offers options to send directly to the printer, save as PDF, or export as a PostScript file.

Exit: Closes the client application.

Log Out: Terminates the active connection with the server.

9.2.2 Settings Menu

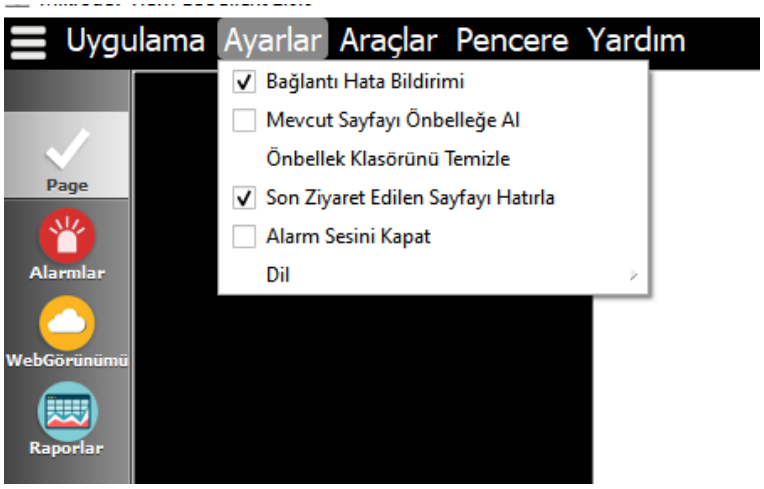

Figure 124 Settings Menu

Language: Select Turkish / English interface

Draw Connection Error Layout: Displays a warning through components when the connection is lost

Cache the Current Page: Enables page caching

Clear Cache Folder: Clears the local cache

Remember Last Visited Page: Automatically redirects to the last visited page after reconnection

Disable Alarm Sound: Disables audible alerts

9.2.3 Settings Menu

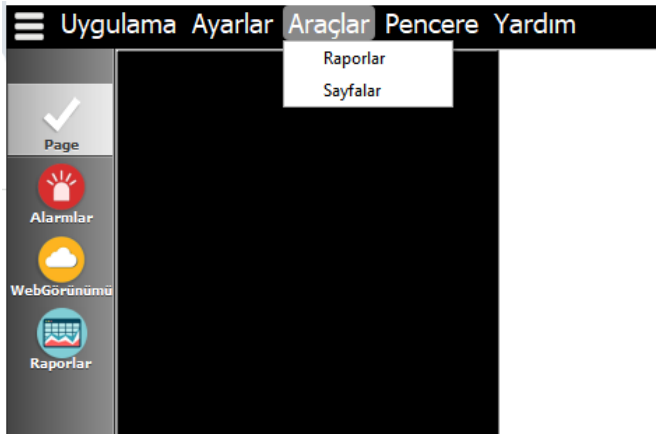

Figure 125 Settings Menu

Reports: Opens the reporting module

Pages: Displays the project pages

9.2.4 Window Menu

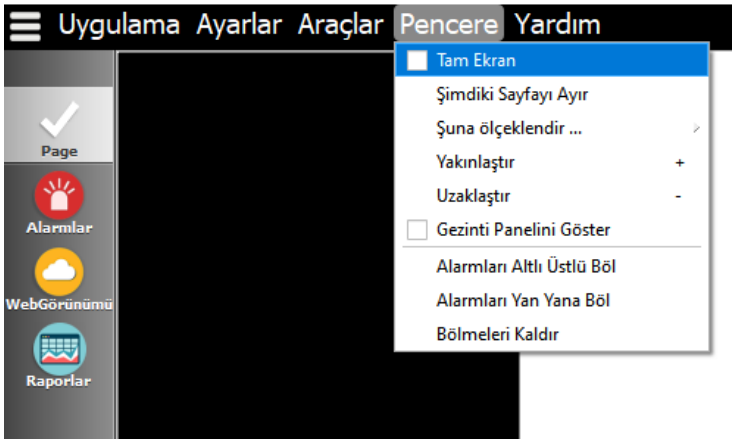

Figure 126 Window Menu

Full Screen: Displays the application in full screen

Detach Current Page: Displays the page in a separate window

Scale / Zoom In / Zoom Out: Adjusts the display settings

Show Navigation Panel: Opens or hides the navigation panel (page selector). This panel allows quick switching between user-defined SCADA pages.

Split Alarms: Displays alarms in a bottom or side panel

Remove Panes: Closes the open alarm windows

9.2.5 Help Menu

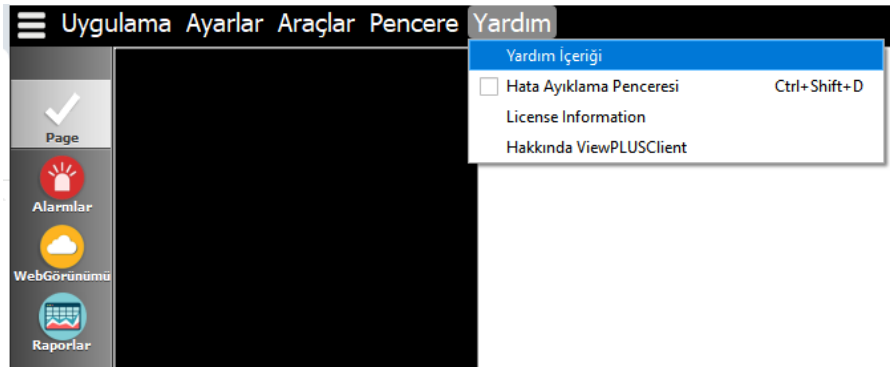

Figure 127 Help Menu

Help Content: Opens a custom help link defined via the SCADA editor. Directs the user to the specified document using the default web browser.

Debug Window: Can be opened using the Ctrl + Shift + D shortcut. Displays the IDs and values of tags on the page. Primarily used for project testing and error identification.

License Information: Displays the current license details, including validity period and granted permissions, if applicable.

About ViewPLUSClient: Contains version information, build number, and company details of the client software. Serves as a reference for technical support requests.

Note: The "Help Content" link is a custom URL defined in the editor. It can be configured via *Project > Client Options > Custom Help Link for Clients*.

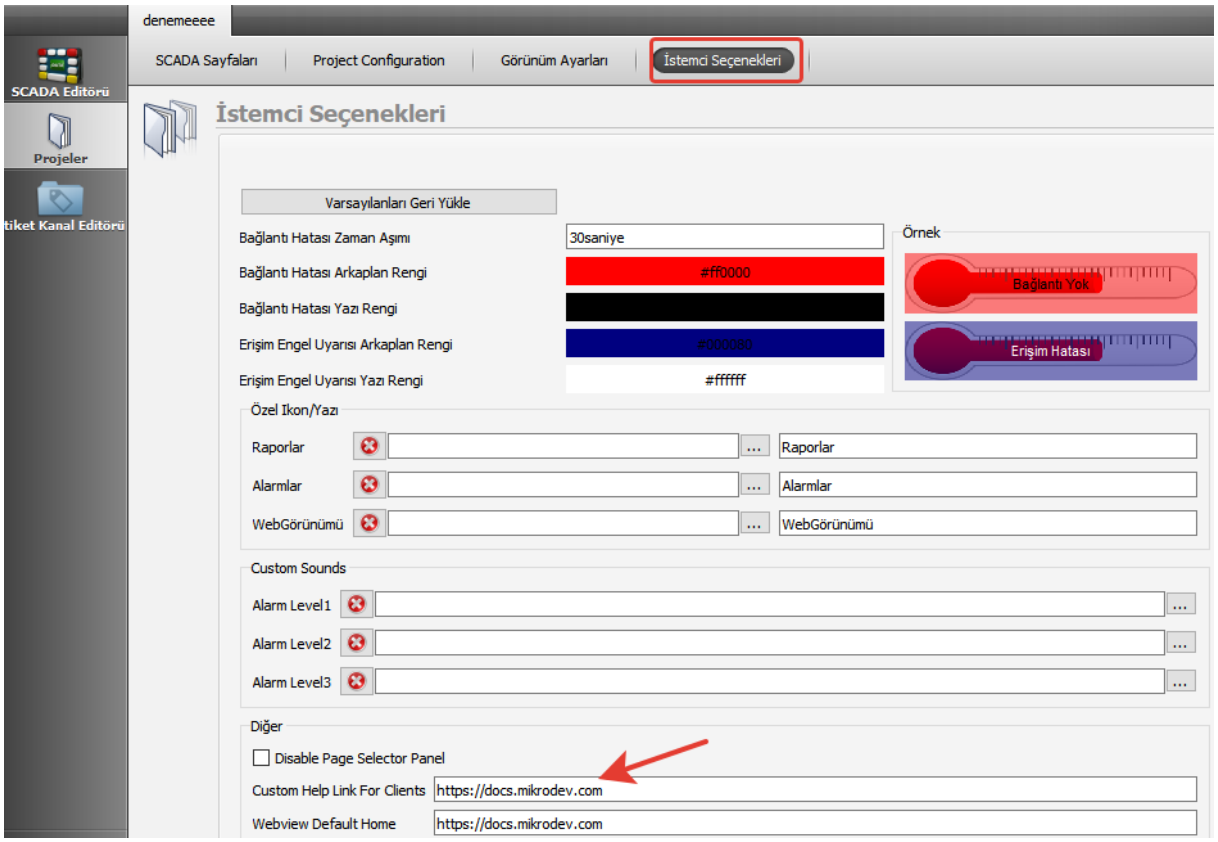

Figure 128 Defining Custom Help Link for Clients

9.3 Left Navigation Panel

When the ViewPLUS SCADA client is launched, this panel appears on the left side of the screen and provides the user with quick access to project pages, alarm management, report screens, and custom links. The options displayed in the panel may change dynamically based on the user's access rights.

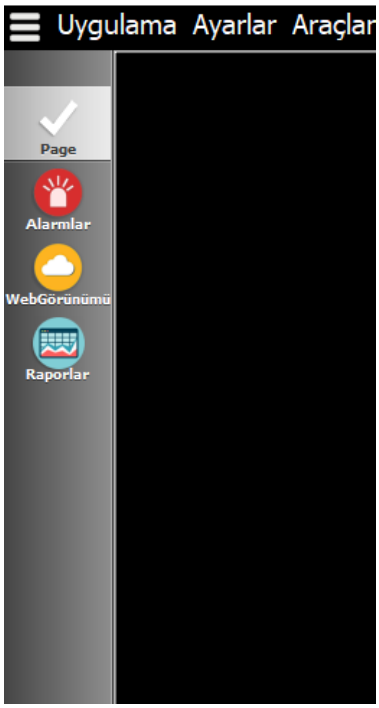

Figure 129 Left Navigation Panel

Tab	Description
Page	Lists all "Page Tab" type pages defined in the SCADA project. The user can view only the pages they have access rights to. In the current example, only one page is defined, so only one option is shown.
Alarmlar	Used to monitor active alarms across the system. Alarm history, active alarms, and alarms assigned to the user can be tracked in detail through this tab. (A more detailed explanation will be added later.)
Web View	This section is used by assigning a custom web link for the client. A web address configured via the editor is displayed here as an icon. When the user clicks on this tab, the specified web page opens in the default browser. It is commonly used for links to technical documents, company websites, or customer portals.
Reports	Displays the reports generated from SCADA system data logs. Defined report templates, records based on time intervals, and export operations are managed through this tab. (A more detailed explanation will be added later.)

Note: The “Web View” link is a custom URL defined in the editor. It can be configured via Project > Client Options > Default Web View Homepage menu.

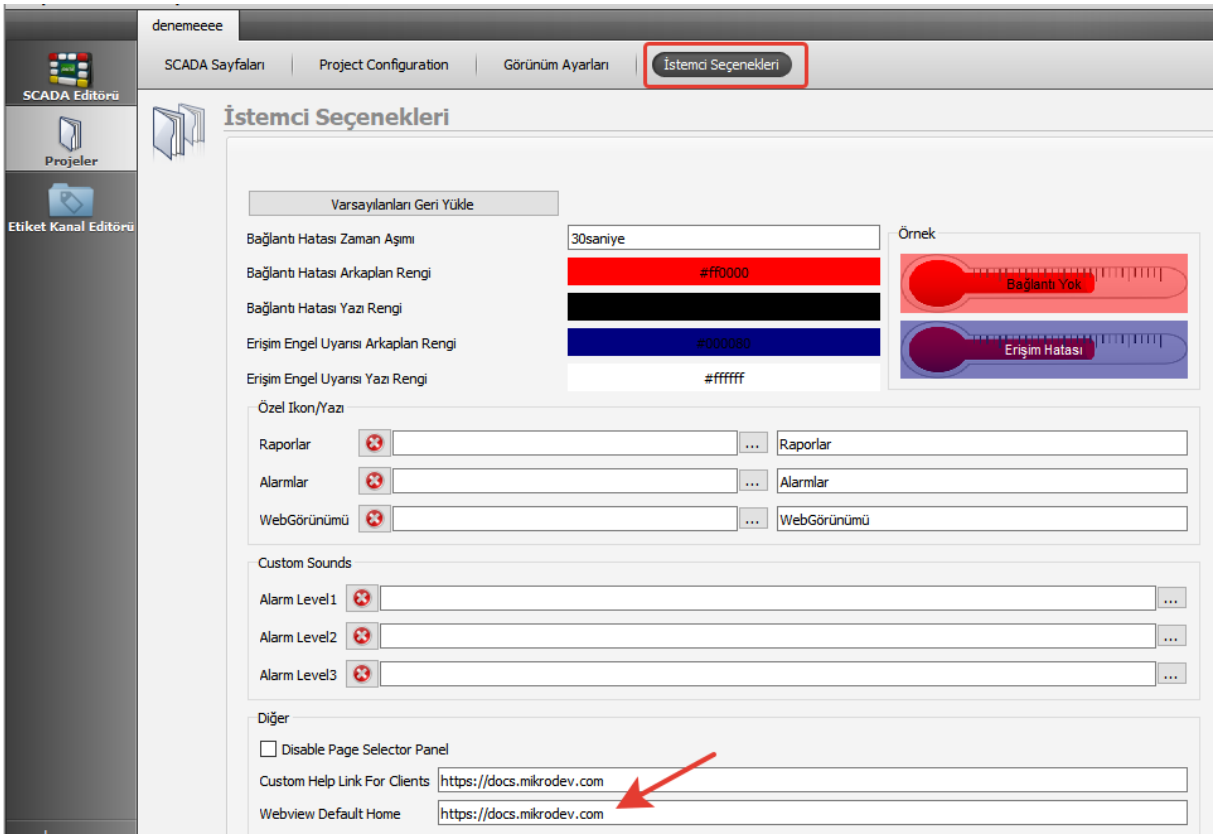

Figure 130 Adding Default Link for Web View

9.4 ViewPLUS Client Alarm Screen

In the ViewPLUS SCADA system, alarm monitoring can be performed from both the client and server (Server Engine) interfaces. The alarm structure consists of three main sections: active alarms, alarm history, and alarm configurations.

9.4.1 Active Alarms Tab

This area displays the alarms triggered in the system in real time

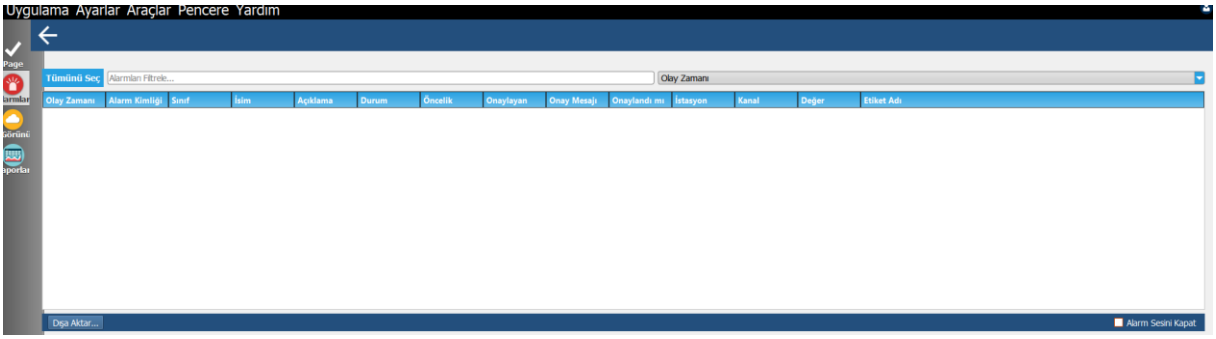

Figure 131 Active Alarms Tab

- Alarm details such as name, description, priority level, acknowledgment status, alarm time, and user information are shown in detail in separate columns.
- Alarms are categorized into 4 sound levels based on their priority. Silent alarms can also be defined.
- Each alarm remains active until it is acknowledged.
- Alarms can be exported (.csv, .html, .xml, etc.).
- Right-clicking an alarm allows direct access to the associated SCADA page.

9.4.2 Alarm Logs

All historical alarms are stored in this tab.

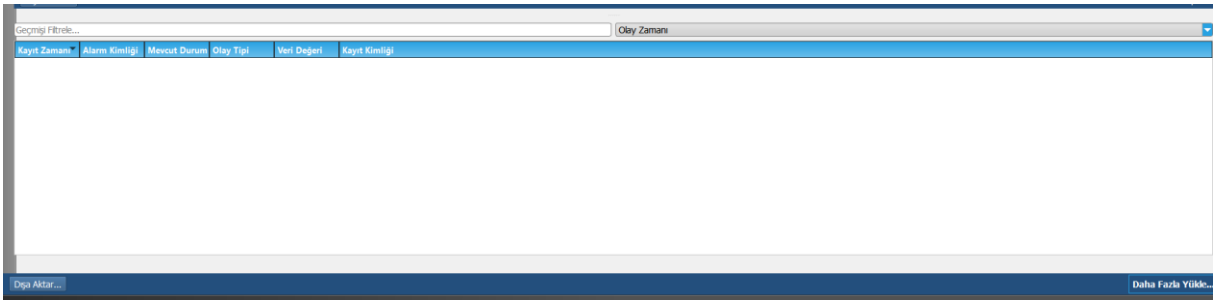

Figure 132 Alarm Logs Tab

- Alarms are categorized in detail by event type (Triggered, Acknowledged, Removed, Recovered).
- Old alarms can be filtered and exported.
- With the "Load More" feature, past database records can be loaded.

9.4.3 Alarms on the SCADA Page

The alarm screen can be pinned below or beside the SCADA page:

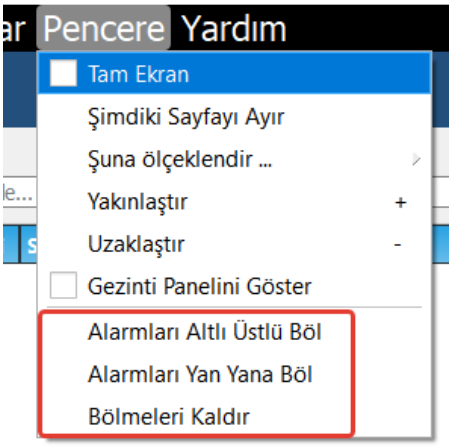

Figure 133 Displaying Alarms on the SCADA Page

- Window > Split Alarms Vertically
- Window > Split Alarms Horizontally
- Window > Close Panes to remove

9.5 ViewPLUS SCADA Reporter Screen

The Reporter tool is used to view and analyze trend data recorded by the SCADA software. Reporting is performed on three main log types: tag logs, alarm logs, and event logs.

To access the Reporter screen, follow these steps:

1. Start the server in your project.
2. Launch the ViewPLUS Client application.
3. Click the "Reporter" tab on the left sidebar.

Note: If the "Reporter" tab does not appear in the left sidebar, ensure that the logged-in user has the necessary permissions to access the reports.

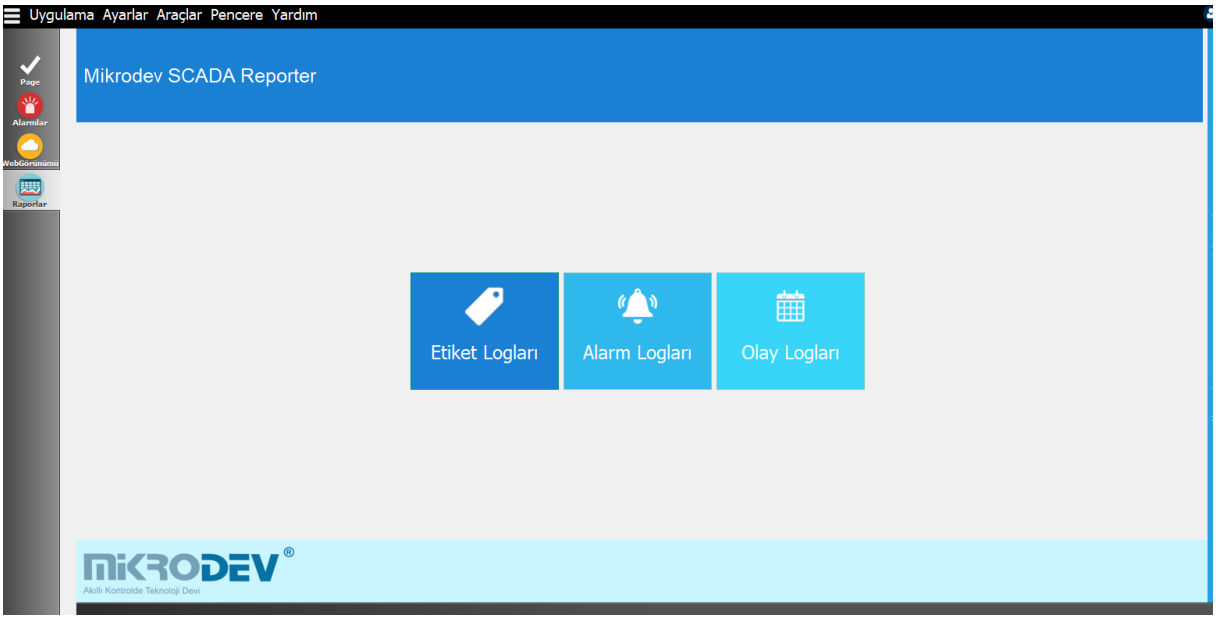

Figure 134 Reporter Screen

9.5.1 Tag Logs

Tag logs represent the time-based data recorded for tags with logging enabled in the Tag Channel Editor. These logs can be filtered by station, time, and statistical parameters and reported in tabular or graphical format.

9.5.1.1 Tag Selection

Tags are grouped by station and listed accordingly. Only tags with logging enabled are displayed.

Double-clicking on a tag adds it to the right panel. Tags can be removed individually or all at once using the "Clear List" option.

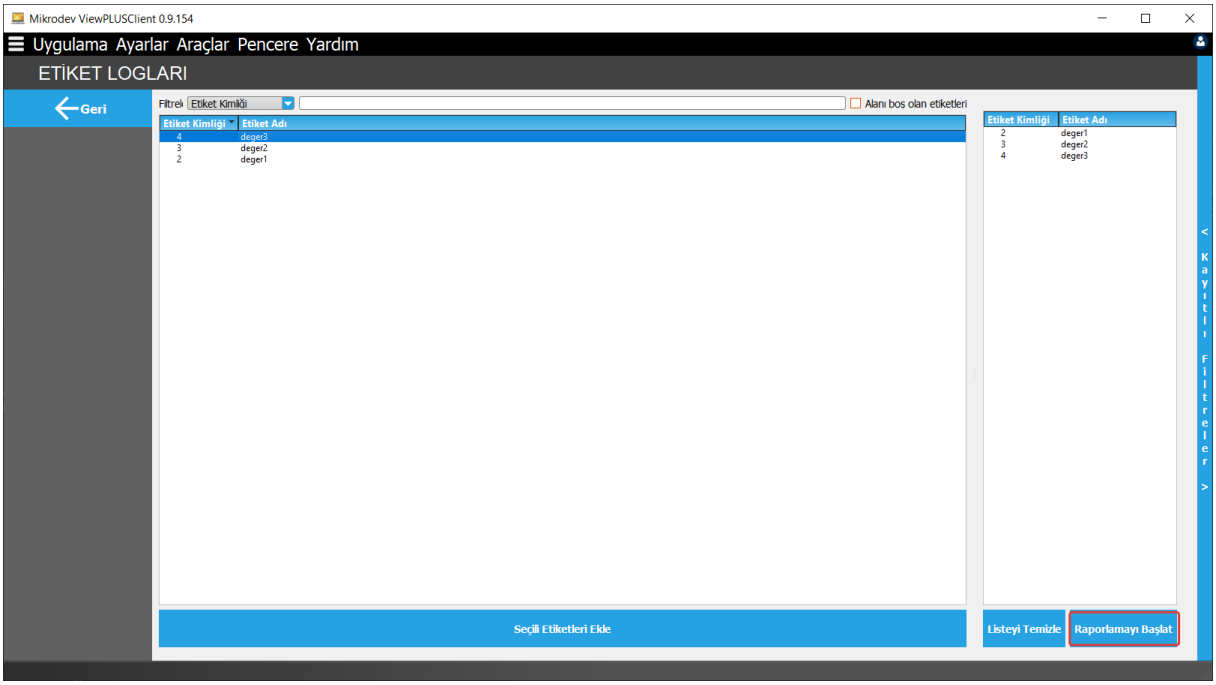

Figure 135 Tag Selection Screen

9.5.1.2 Viewing Tag Logs

Tag values are presented either graphically or in a table view. Two key parameters can be selected:

- **Time Range Selection:**

Between: Select two dates.

Within Last: Specify a time range such as hours/days/weeks retrospectively.

Time/Date: A specific timestamp.

Before Time: From today backwards.

Figure 136 Time Range Selection for Tag Logs

- **Data Type Selection:**

All Data: Displays all recorded values.

Statistical: Displays average, max, min, total, delta, median.

9.5.1.3 Managing Graphs

Show Label: Displays the tag name and its color on the graph.

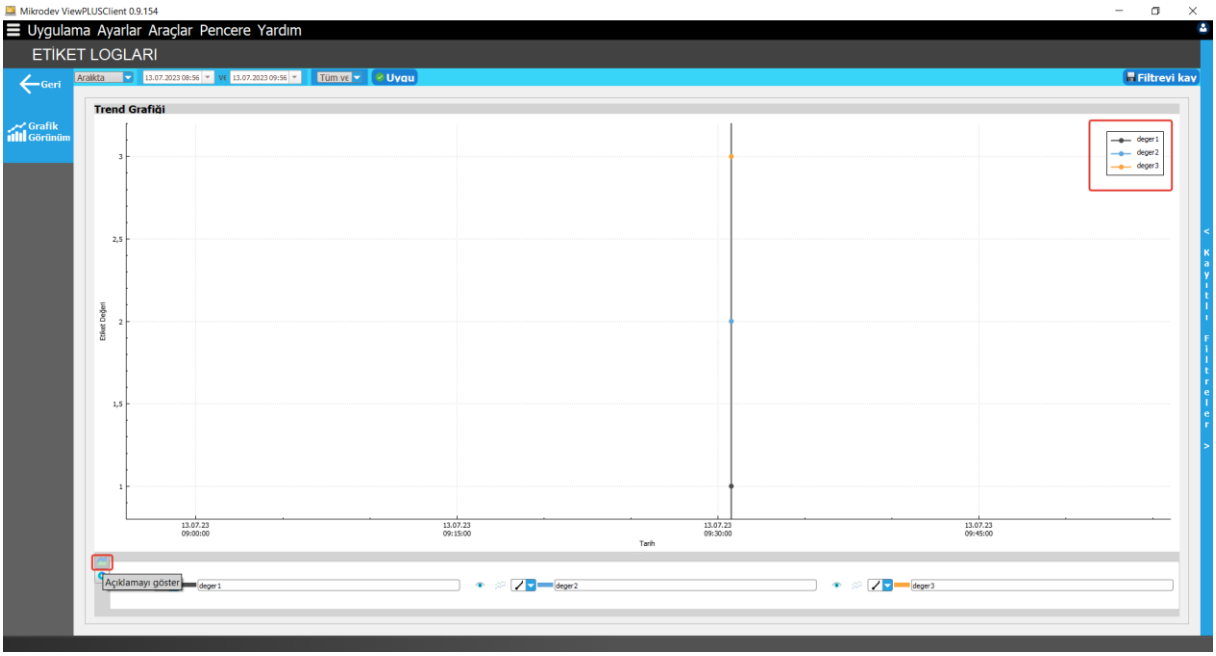

Figure 137 Show Label Option

Add New Trend: Adds a comparison graph.

Figure 138 Add New Trend Option

Show/Hide: Toggles visibility of the graph line.

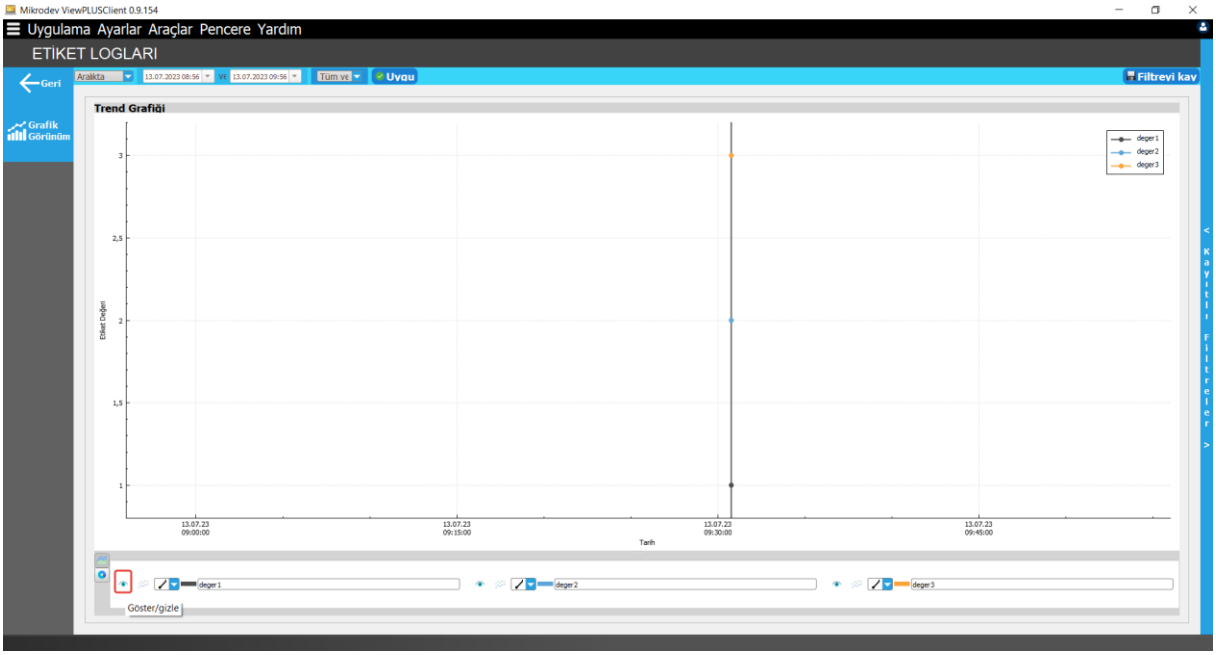

Figure 139 Show/Hide Option

Fill/Clear: Fills or clears the area below the graph line.

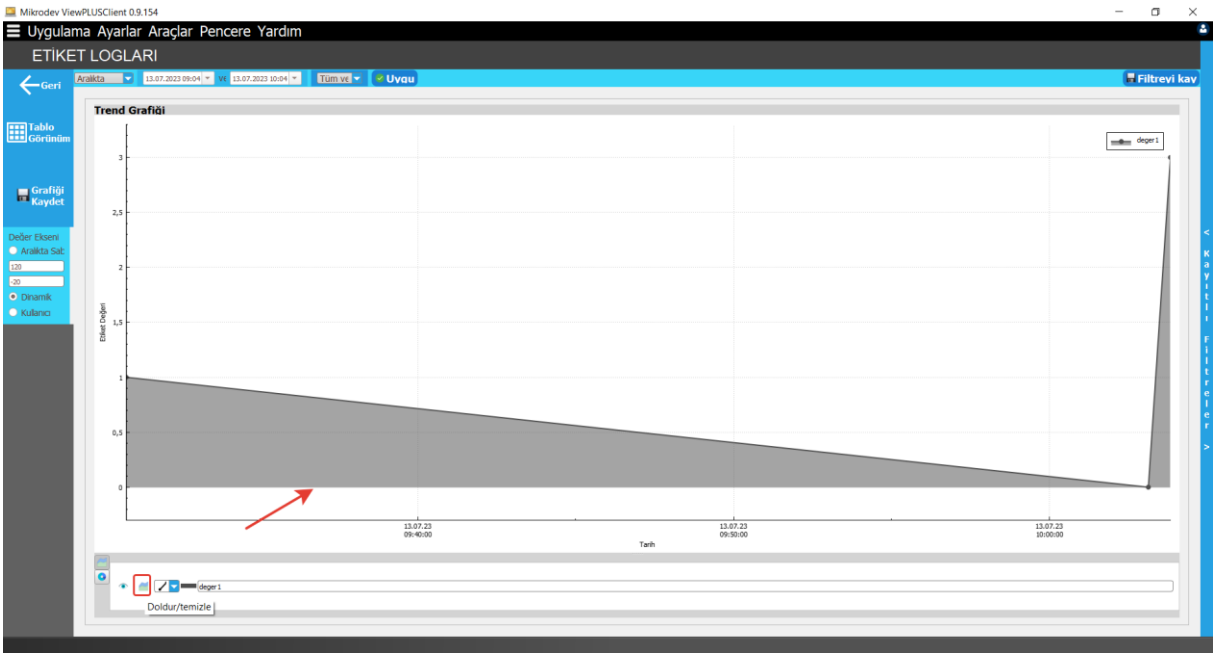

Figure 140 Fill/Clear Option

Join Points: Adjusts the line connection style for data points.

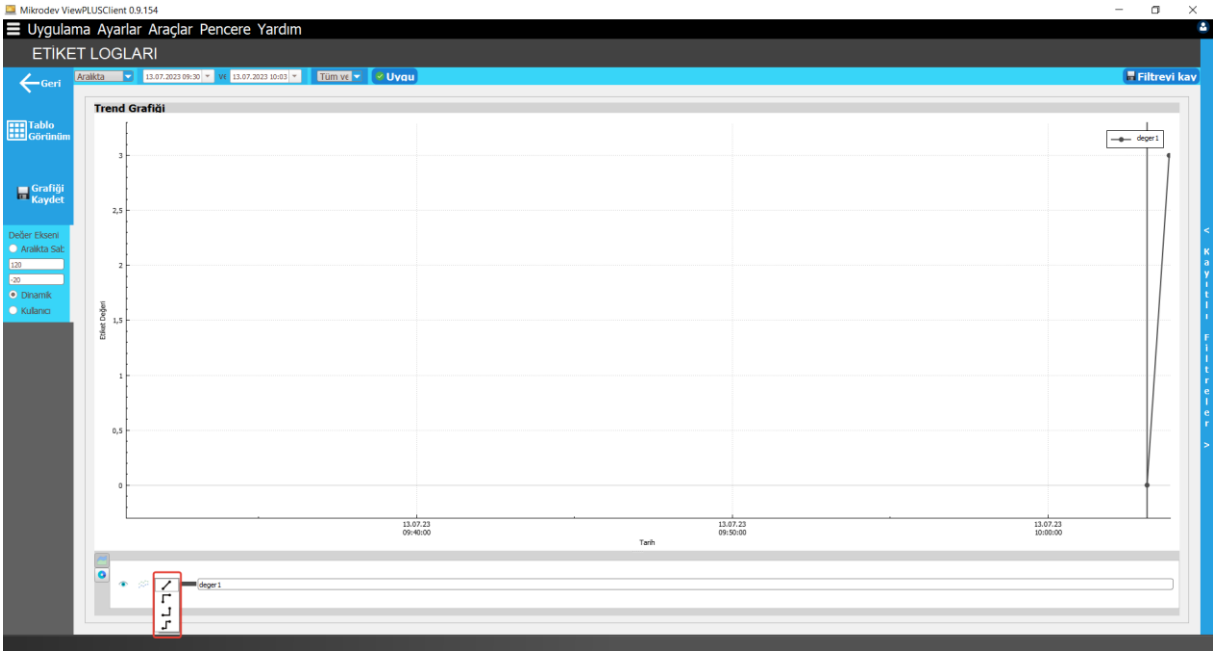

Figure 141 Join Points Option

9.5.1.4 Saving the Graph

Graph can be saved as screenshots in PNG format using the "Save Graph" option.

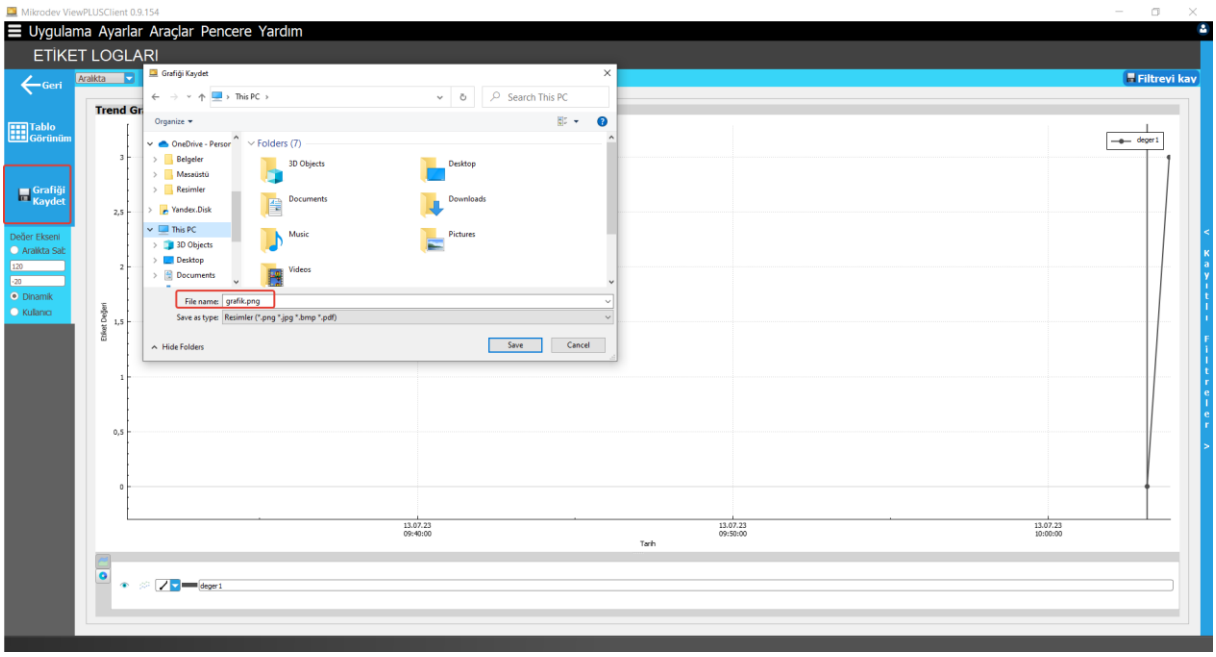

Figure 142 Save Tag Logs as Graph

9.5.1.5 Exporting Data

Switch to table view to export data in HTML, XML, XLSX, or CSV format.

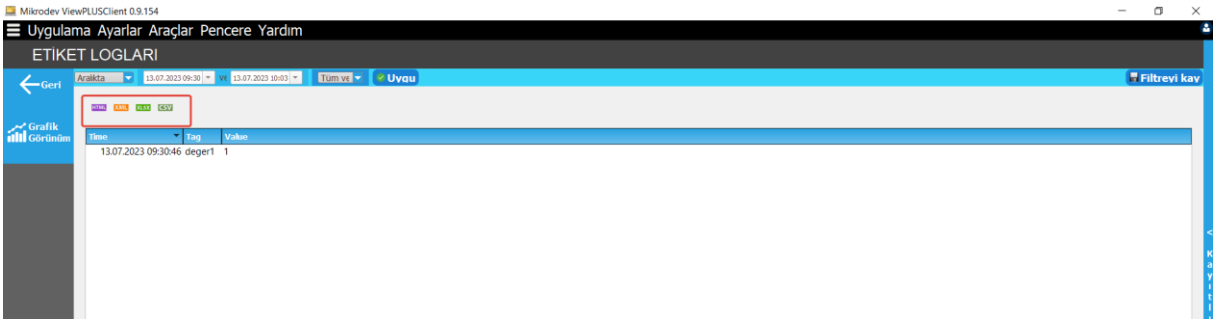

Figure 143 Export Tag Data

9.5.1.6 Saving Filters

Filter parameters can be saved with a custom name using the "Save this filter" option for later reuse.

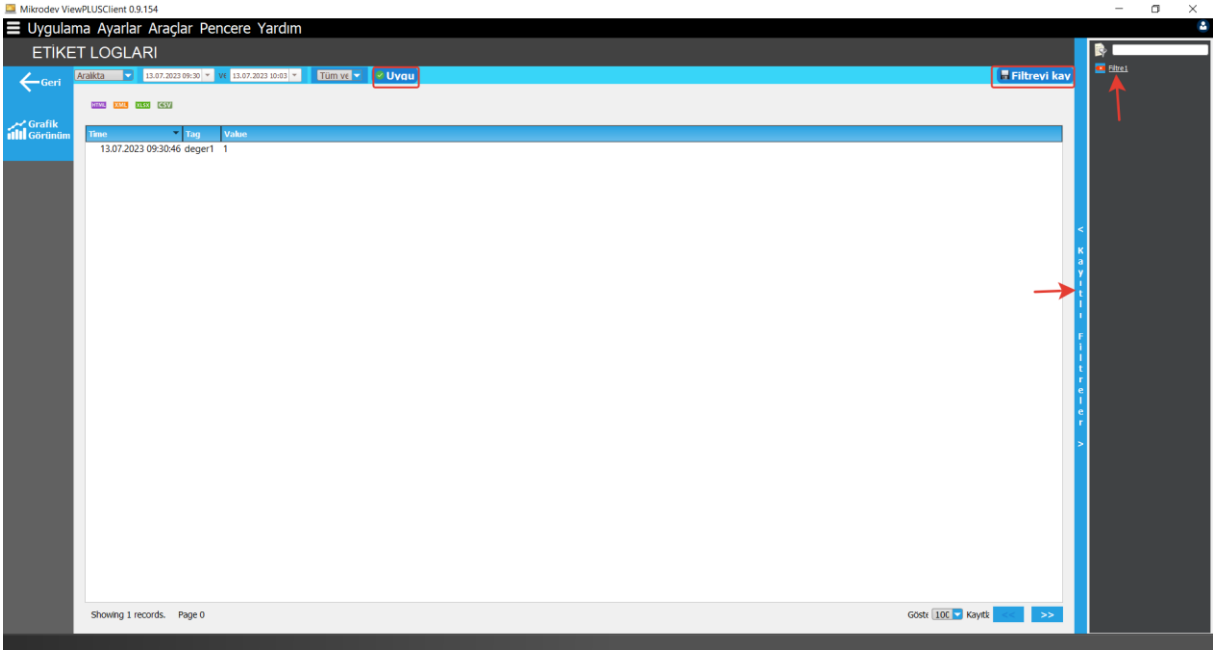

Figure 144 Save Filter for Tag Logs

9.5.1.7 Linking Saved Filter to SCADA Object

A saved filter can be copied via right-click and linked to an object in the editor screen. When the linked object is clicked, the report screen opens with the predefined filter applied.

9.5.2 Alarm Logs

Alarms generated in the system can be monitored via the Alarm Logs section of the Reporter screen.

9.5.2.1 Selecting Alarms to Display on the Reporter Screen

Alarms created in the ViewPLUS SCADA Editor are listed on the alarm selection page. The alarm list can be filtered in various ways according to alarm properties. Alarms to be reported are selected by double-clicking items in the left panel. Selected alarms appear in the right panel.

To open the report screen for the selected alarms, click the "Start Reporting" button in the bottom right corner.

To delete a specific alarm from the right panel, select the alarm and press the Delete key. To remove all alarms, select "Clear List."

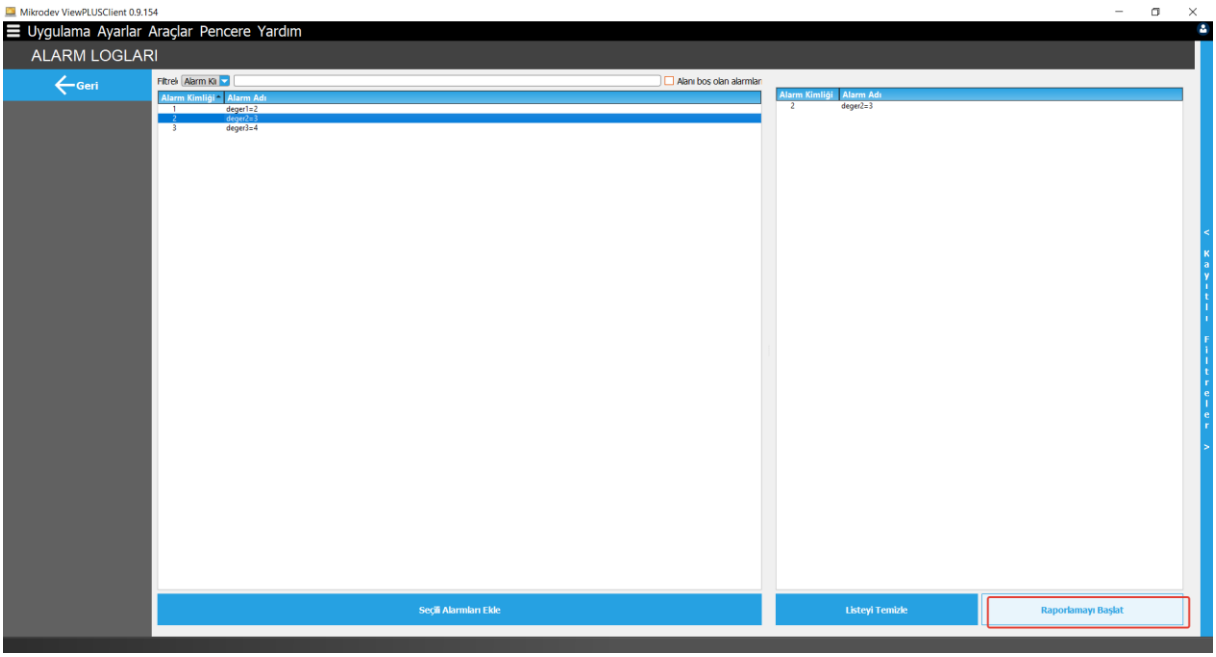

Figure 145 Alarm Selection Screen

9.5.2.2 Viewing Alarm Logs

When the report page is first opened, the selected alarms are displayed in a table format for a specified time range.

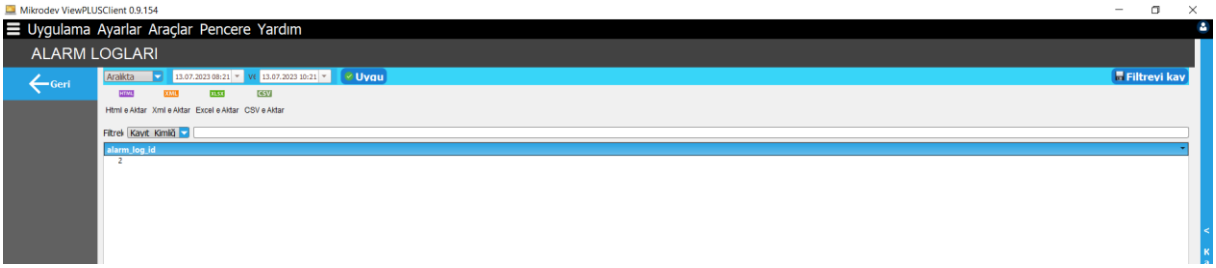

Figure 146 Viewing Alarm Logs

- **Time Range Selection:**

Between: Two different dates are selected.

Within Last: Choose hour, day, week, etc., and calculate time backward from the current date/time.

Date/Time: A specific date and time is selected.

Before Time: Time is calculated retrospectively from today. For example, “Today - 5 days” represents 5 days ago.

Once the appropriate filter option is selected, click Apply on the right.

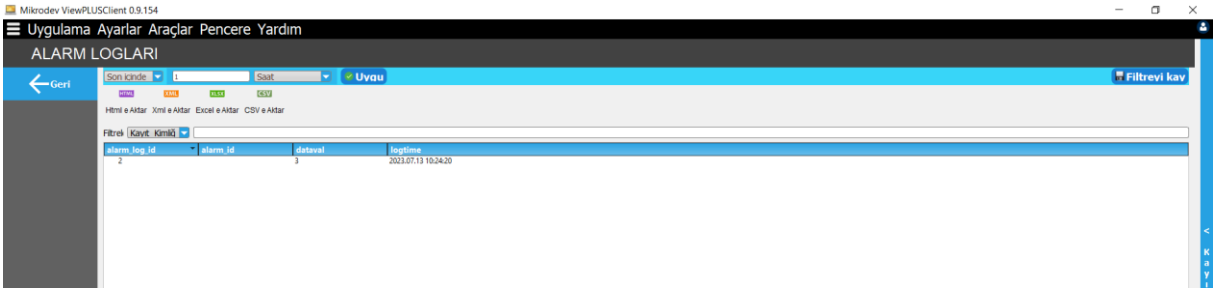

Figure 147 Time Range Selection for Alarm Logs

9.5.2.3 Exporting Data

Based on the selected parameters, data can be exported in various formats. Use the buttons at the top of the report screen—HTML, XML, XLSX, CSV—to export data.

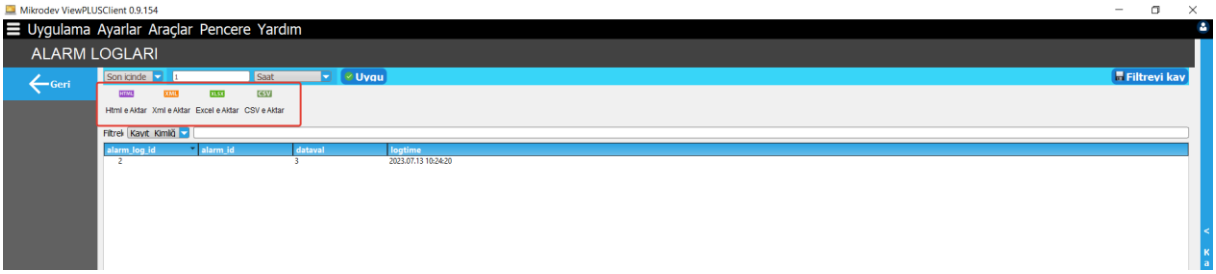

Figure 148 Exporting Alarm Logs

9.5.2.4 Saving Filters

Selected parameters must be applied at least once. Then, use the "Save this filter" option to name and save the filter. The saved filter will appear in the filter menu on the right, specific to the user account.

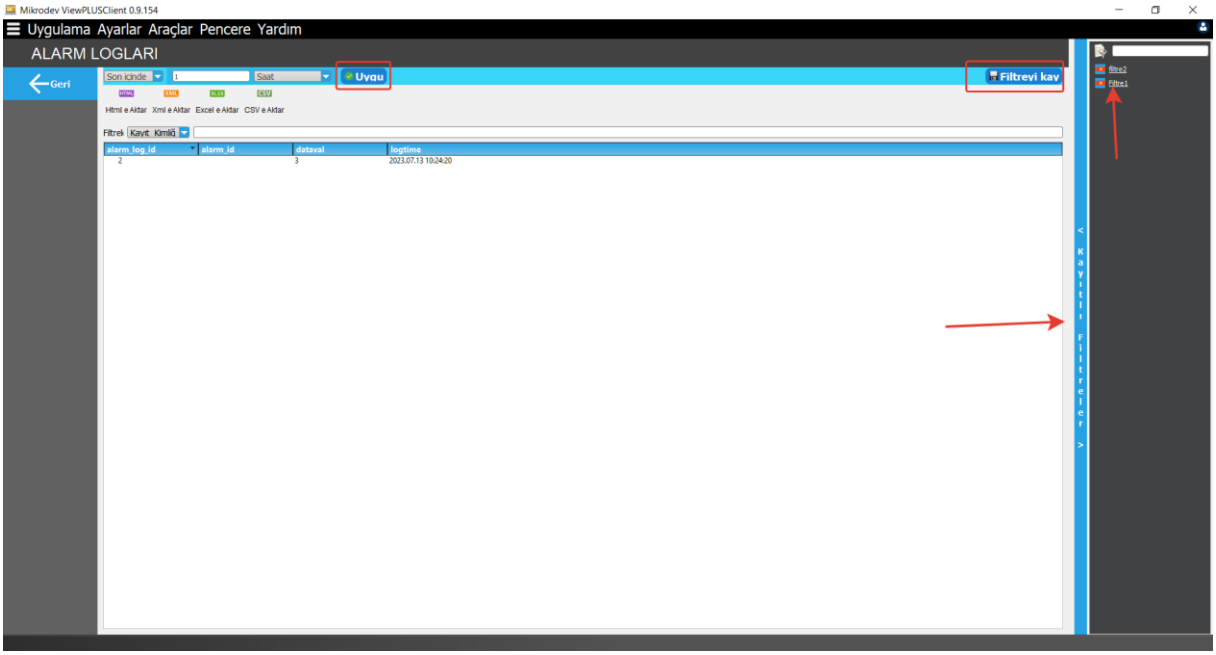

Figure 149 Saving Filters for Alarm Logs

9.5.2.5 Linking Saved Filters to SCADA Objects

A saved filter can be copied via right-click and linked to SCADA objects in the Editor. When the linked object is clicked, the report screen will open with the predefined filter applied.

9.5.3 Event Logs

User actions can be monitored in the Event Logs section of the Reporter screen.

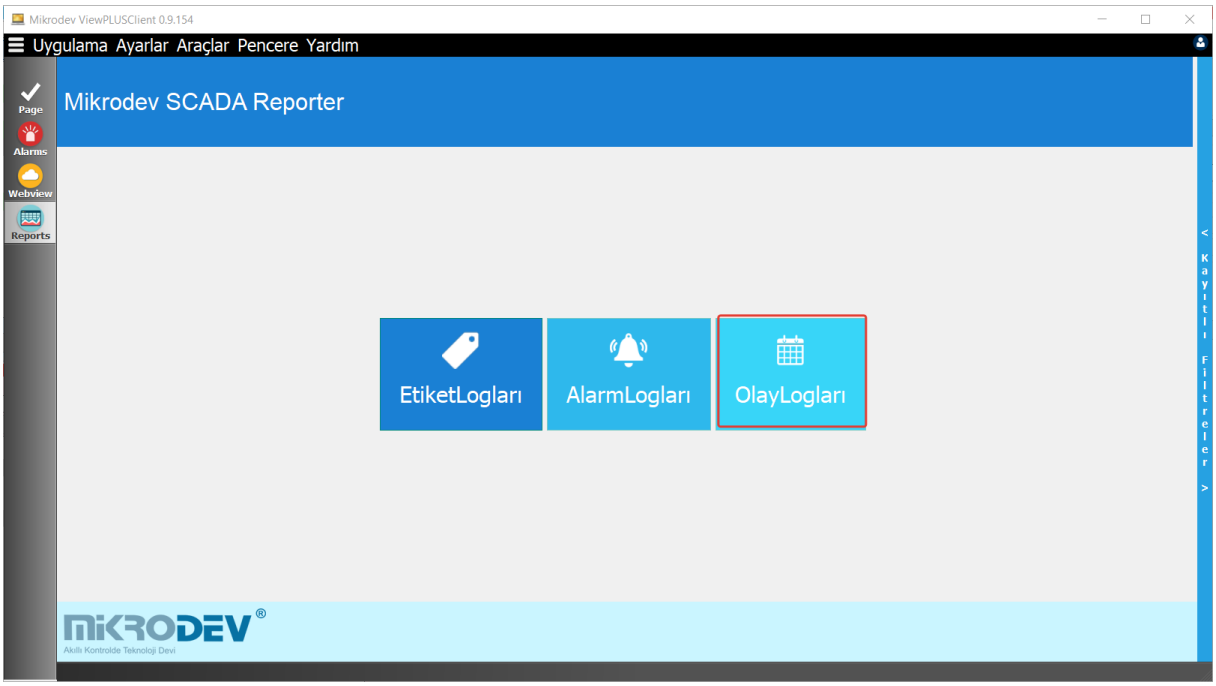

Figure 150 Event Logs

9.5.3.1 Viewing Event Logs in the Reporter Screen

When the report page is first opened, all user actions are displayed in a table. Actions such as login IP addresses, writing values to tags, report page access requests, the names of tags for which write requests were made, and the timestamps of these events can be seen.

- **Selecting a Time Range:**

Interval: Two different dates are selected.

Within Last: Choose hour, day, week, etc., and calculate time backward from the current date/time.

Date/Time: A specific date is selected.

Time Ago: Time is calculated retrospectively from today.

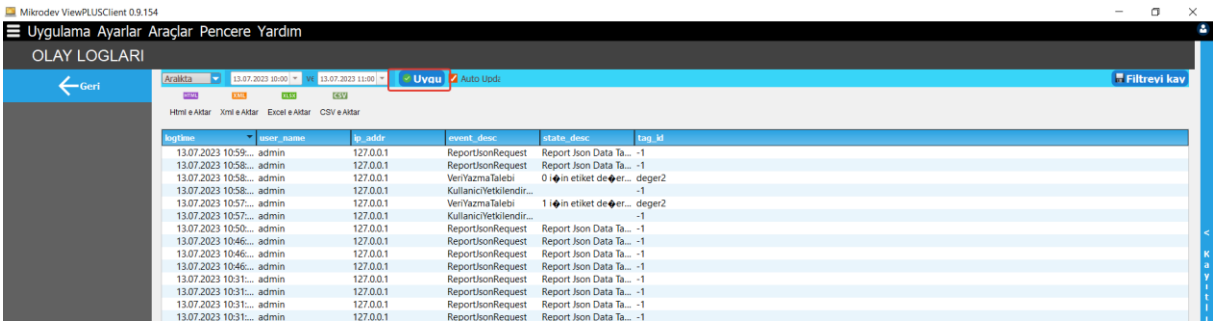

Figure 151 Selecting Time Range for Event Logs

After choosing the appropriate filter, click Apply on the right.

Note: If the "Auto Update" option on the right of the Apply section is checked, the event log screen will refresh automatically.

9.5.3.2 Exporting Data

Data in the table can be exported using the html, xml, xlsx, csv buttons located at the top of the Reporter screen.

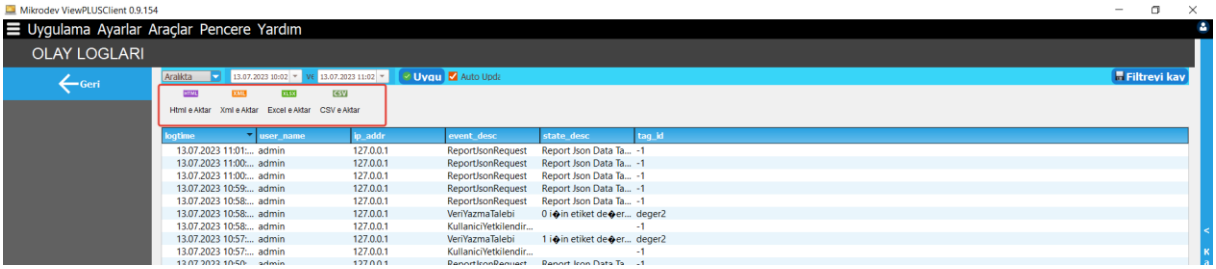

Figure 152 Exporting Event Logs

9.5.3.3 Saving Filters

Used parameters can be saved with the "Save this filter" option for future reuse. Filters are saved per user account.

10 Component Manager

The "Component Manager" is an editor used in ViewPLUS SCADA projects to organize visual elements, define new components, and create project-specific component libraries. With this manager tool, component files with the .comx extension can be defined, and visual animations can be integrated into the SCADA interface.

The application can be launched from the Tools menu on the main SCADA screen by selecting Component Manager.

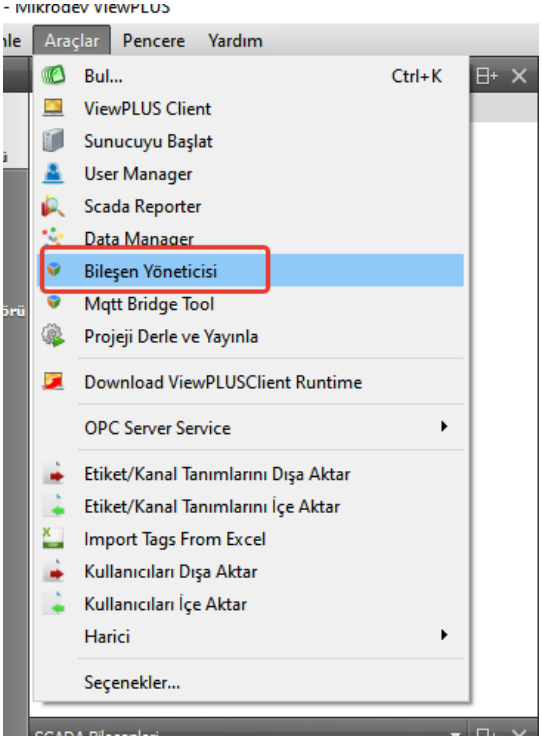

Figure 153 Accessing the Component Manager

When the program is launched, the main screen appears as shown below.

This screen includes the component folder, the list of existing components, and action buttons.

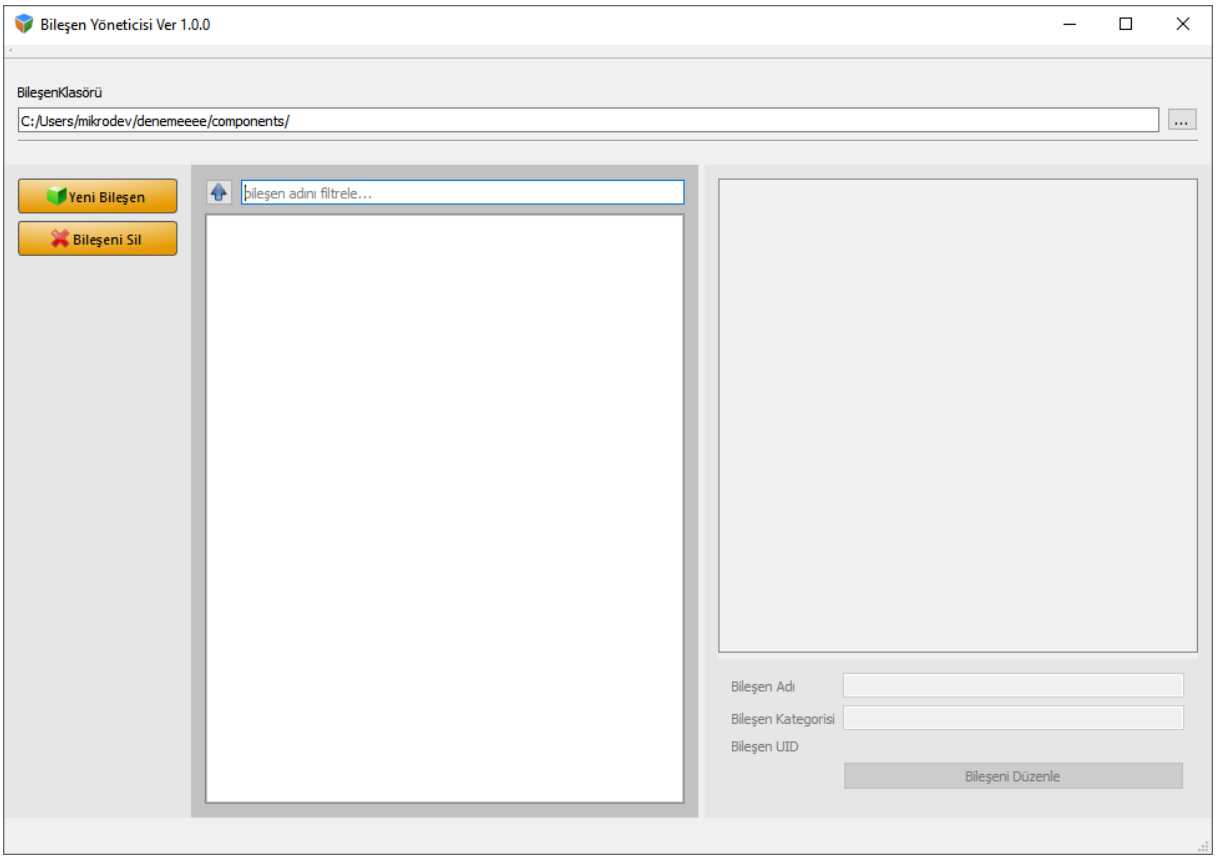

Figure 154 Component Manager Main Interface

- **Component Folder:** Indicates the directory where the project components are located.
- **New Component:** Used to define a new component.
- **Delete Component:** Permanently deletes the selected component.
- **Filtering Field:** Allows filtering among existing components by name.

10.1 Defining a New Component

When the "New Component" button is clicked, the component definition window opens.

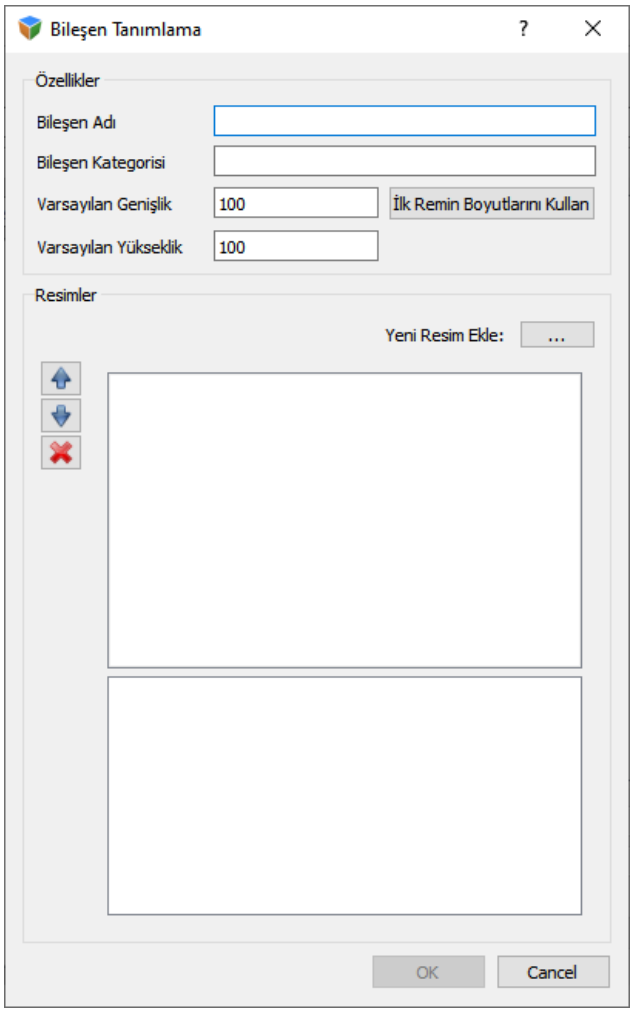

Figure 155 Component Definition Screen

Component Name: The name to be assigned to the new component.

Component Category: This field specifies the group to which the component belongs. Multiple components can be defined under the same category name.

Default Width/Height: The component's size on the interface.

Use First Image's Size: When selected, the component's size is automatically adjusted according to the dimensions of the first added image.

Images Area: This area is where the visuals (.png, .jpeg, .gif, .svg) belonging to the component are uploaded and ordered.

Note: The "index" values corresponding to the images determine which image is displayed on the SCADA screen based on the tag value. For example, if the tag value is "2", the image under index 2 will be displayed.

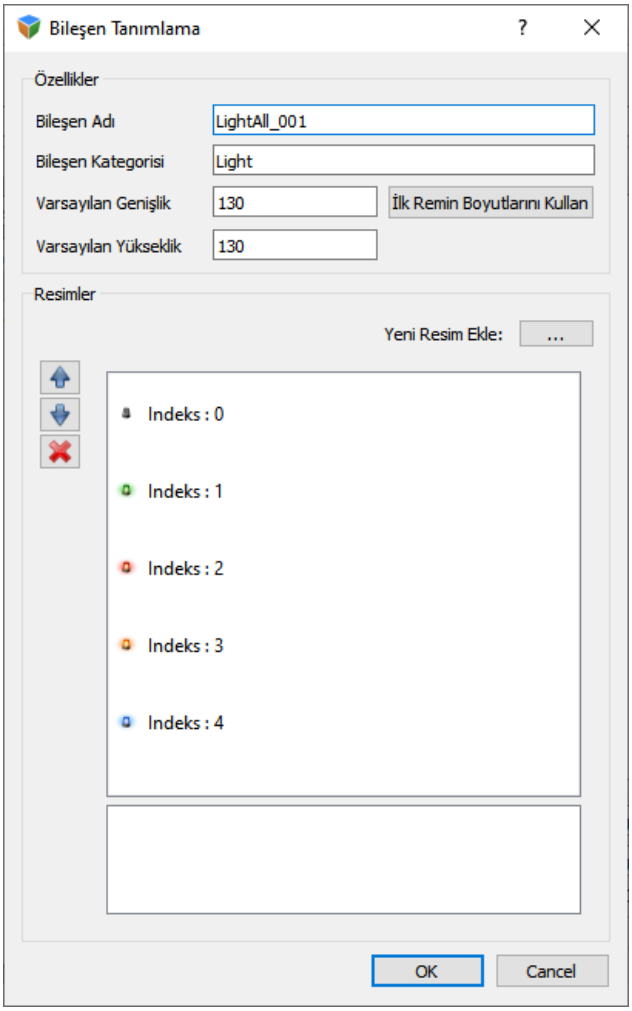

Figure 156 Example Component Definition Screen

Components under the same category are listed as a group in the SCADA editor.

Components can be used as elements in screen design and can be associated with animations.

Existing components can be edited by adding or removing images.

11 SCADA User Manager

In the ViewPLUS SCADA system, user management, access rights control, and group definitions are carried out through the User Manager interface.

To create a user and perform necessary configurations in ViewPLUS, go to the Tools tab and select Launch User Manager.

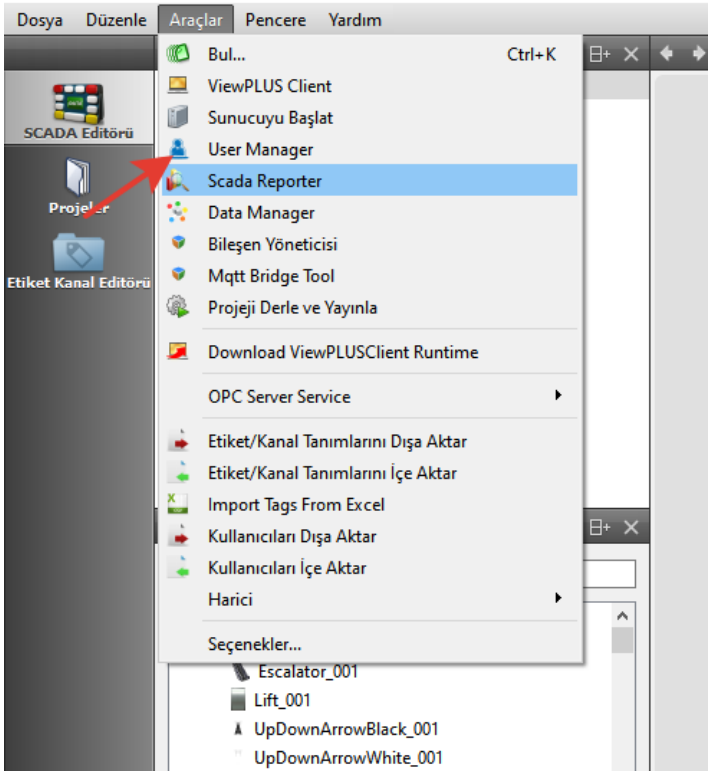

Figure 157 Opening the User Manager

11.1 Creating a New User

Click the New User tab and fill in the required parameters to create a new user. The created user will appear under the username in the Users tab.

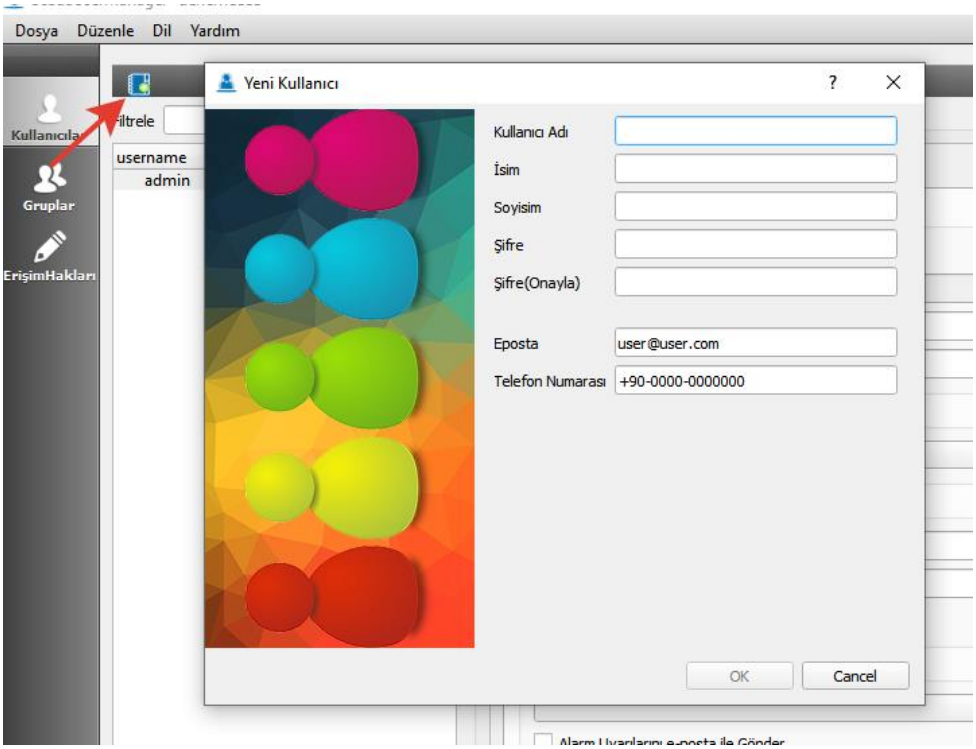

Figure 158 Creating a User

User Creation Interface:

Username: The username used to log into the SCADA system. It must be unique.

First Name: The user's first name. This can appear in visualizations and reports.

Last Name: The user's surname.

Password: The password the user will use to log in. A secure password should be chosen.

Confirm Password: A repeated entry of the above password for verification.

Email: The user's email address. System notifications and alarm alerts can be sent to this address.

Phone Number: The user's phone number. Especially useful for alarm notifications or user identification.

11.2 Users Tab

In ViewPLUS SCADA, the Users tab is used to manage users, control access rights, and define group structures via the User Manager interface.

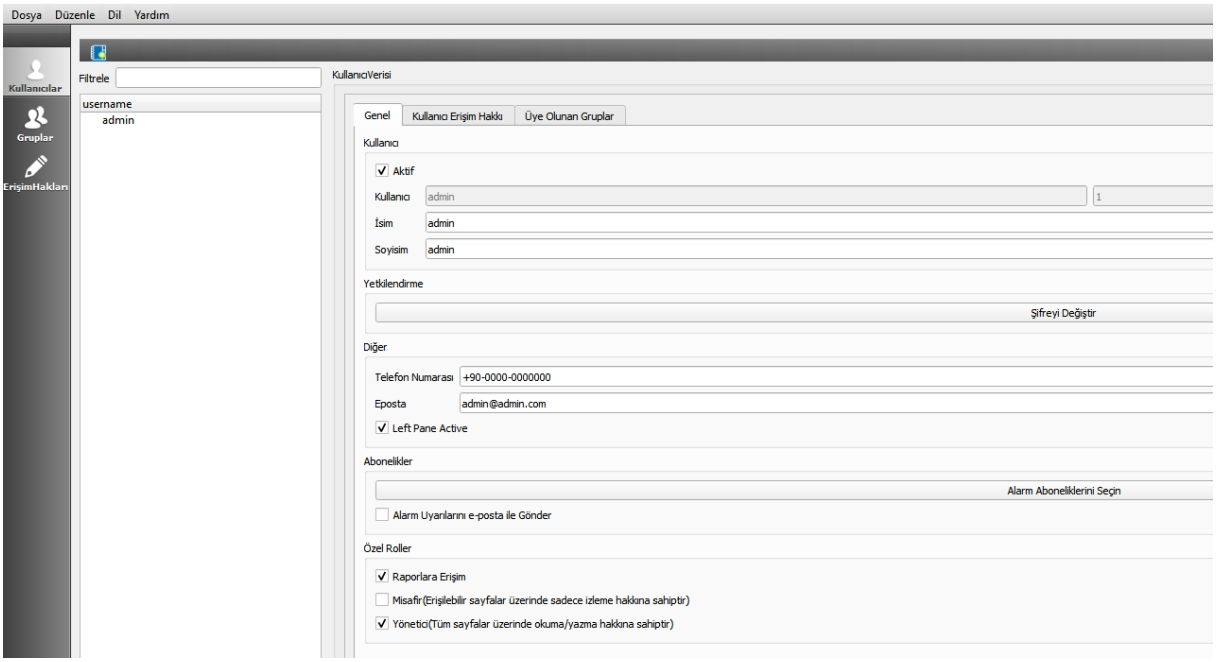

Figure 159 General User Settings

Active: Indicates whether the user is active in the system. If unchecked, the user cannot log in.

User: The username (non-editable).

Name / Surname: The user's full name. Appears in reports and logs.

Authorization: Allows password changes for existing users.

Phone Number / Email: User contact details. Alarm notifications are sent via the defined email address.

Left Pane Active: Determines whether the SCADA Client’s left panel is visible for this user.

Send Alarm Alerts via Email: Sends alarms assigned to the user via email. The email address must be valid.

Select Alarm Subscriptions: Defines which alarm lists the user can monitor. Only selected alarms will be shown.

Access to Reports: Allows access to the Reporter screen in ViewPLUS Client. If not selected, the "Reports" tab will not be visible.

Guest: The user can only view accessible pages without control or write permissions.

Administrator: Has full read/write access to all SCADA pages and tags. This is the most privileged user role.

11.3 User Access Rights

Which pages and tags a user can see or interact with in SCADA screens are determined by access rights. These can be assigned on a page, tag, or object basis and are configured under the User Access Rights tab in the user definition screen.

11.3.1 User Access Rights Tab

This tab lists all predefined access rights in the system (Right1, Right2, ..., Right48).

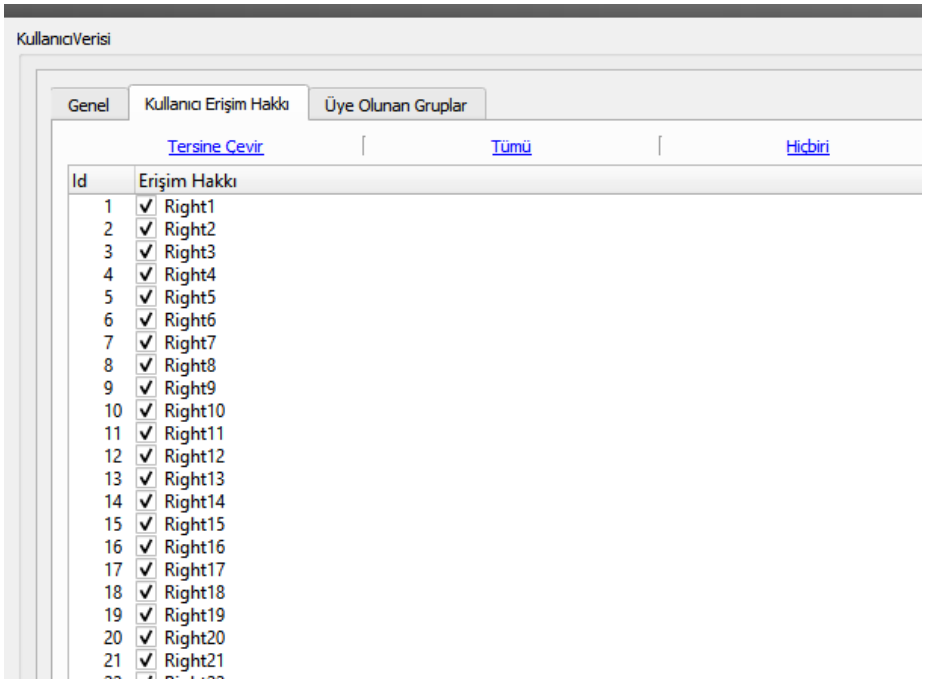

Figure 160 Defining User Access Rights

- **All:** Selects all access rights.
- **None**: Clears all selections.
- **Invert:** Inverts the current selection.

These rights are used in the page and tag access windows explained below.

11.3.2 Tag Access Rights – Tag Channel Editor

In the tag detail window, the Access Rights section allows:

Read Permission: Users can only read the tag

Write Permission: Users can write values to the tag.

Access rights IDs (e.g., Right2, Right5) are assigned to each permission field.

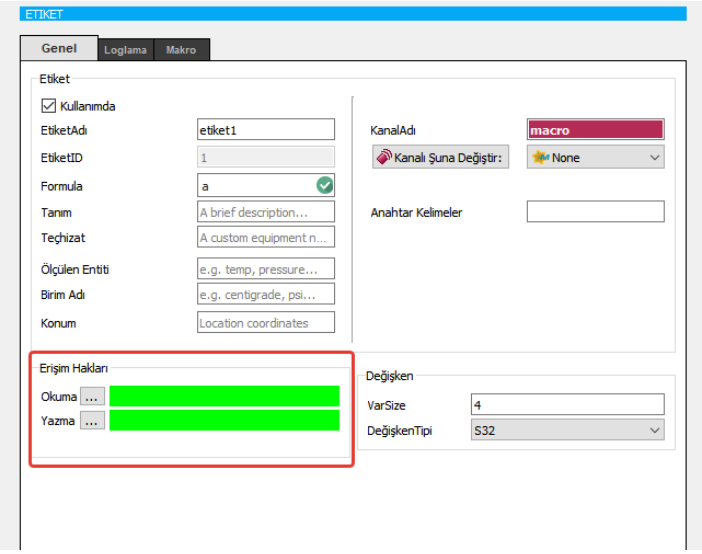

Figure 161 Defining Tag Access Rights

Example: If Tag1 has Read: Right2 and Write: Right5, users without these rights will neither see nor change the tag in SCADA screens.

11.3.3 Page Access Rights – SCADA Design Page

If you want only specific users to access a SCADA page:

- Go to the page properties and click the Access Rights field.
- Select the access rights (e.g., Right1, Right6) required to view the page.

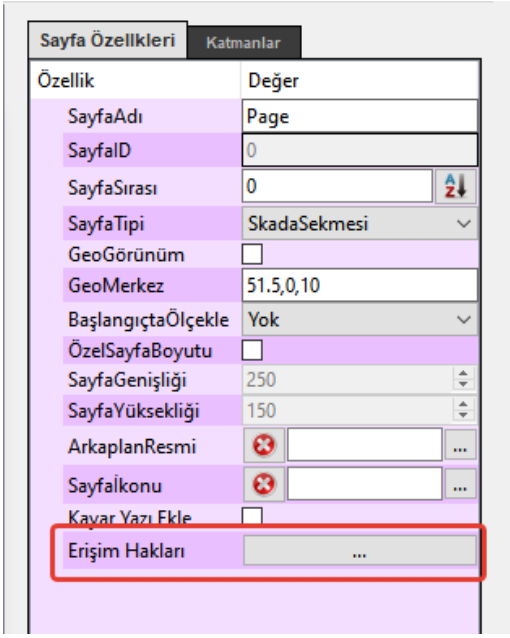

Figure 162 Page Access Rights

This determines visibility in the “page selector” panel. Users without access rights won’t see the page listed, and navigation links will not function.

Note: Assigning an access right (e.g., Right5) to a user only gives that right to the user. To be effective, the same access right must also be defined on each page or tag the user should access.

So, granting access rights alone is not sufficient. The same access rights must also be configured on every relevant object (tag, page, or component).

11.4 Groups Tab

Users can be assigned to groups. Group definitions can be used to centrally manage access rights.

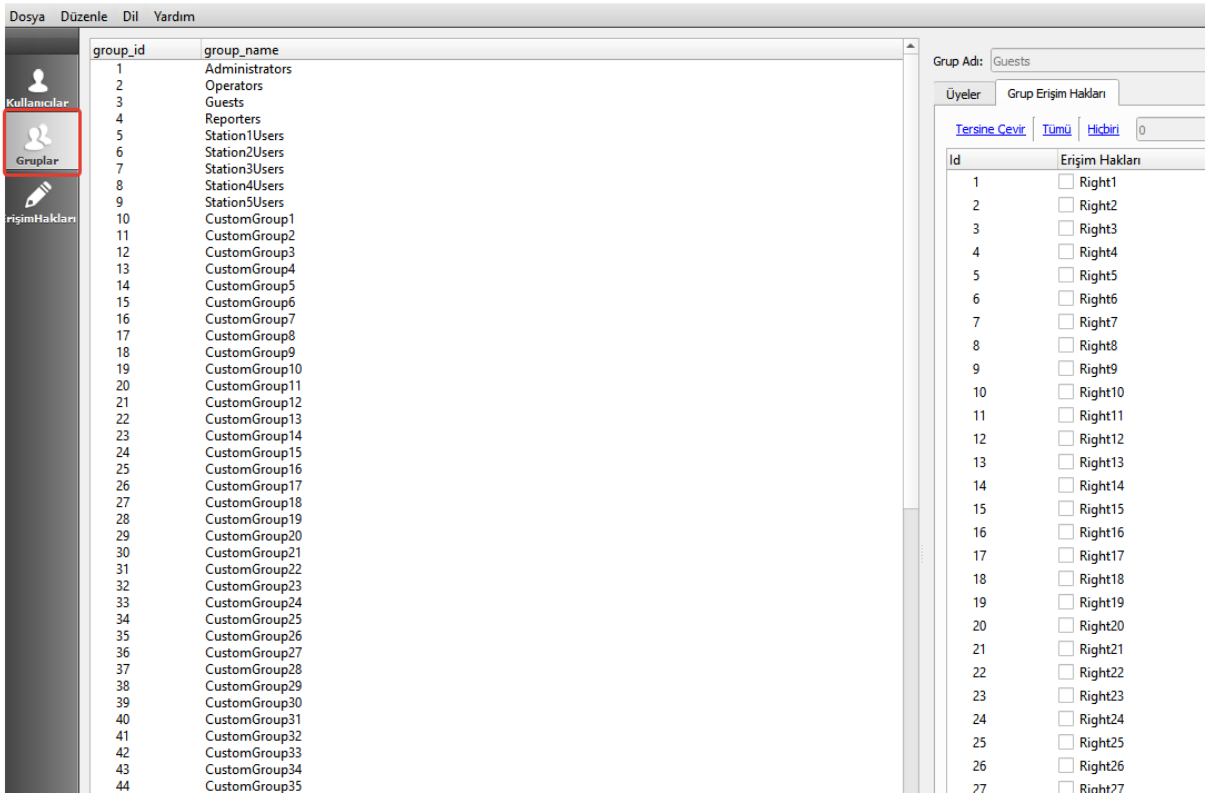

Figure 163 Groups Tab

From the Member Groups section, a user can be added to one or more groups. Access rights assigned to the group will automatically apply to all users in that group.

11.5 Access Rights Tab

The Access Rights tab shows all access rights defined in the system. From here, rights can be named and edited, but they are not assigned to users or groups from this screen.

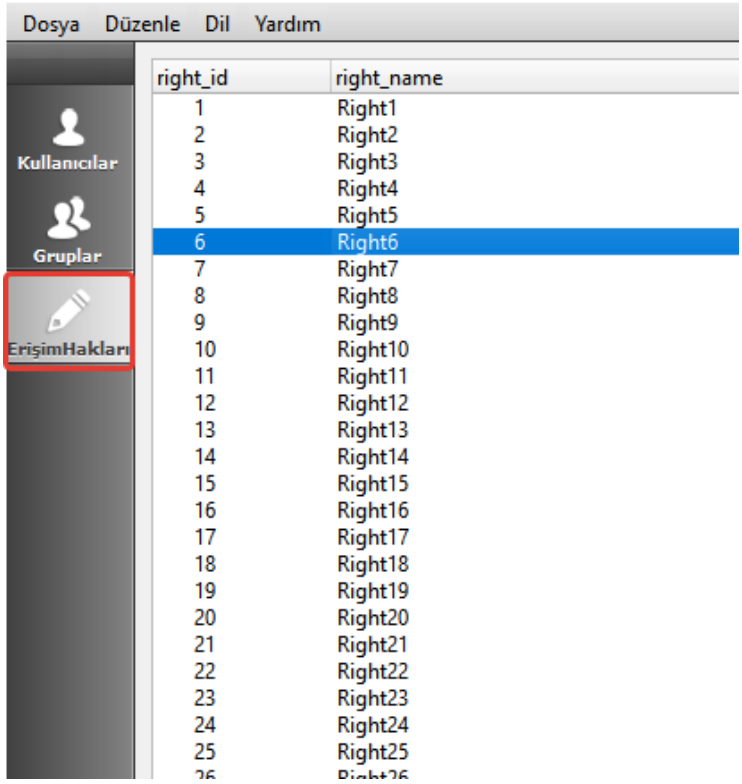

Figure 164 Access Rights Tab

12 SCADA Reporter Tool

The SCADA Reporter tool is used in Mikrodev SCADA projects to analyze, filter, export, and report logged (recorded) data. This application does not monitor live data, it only operates on data previously recorded in the SCADA system.

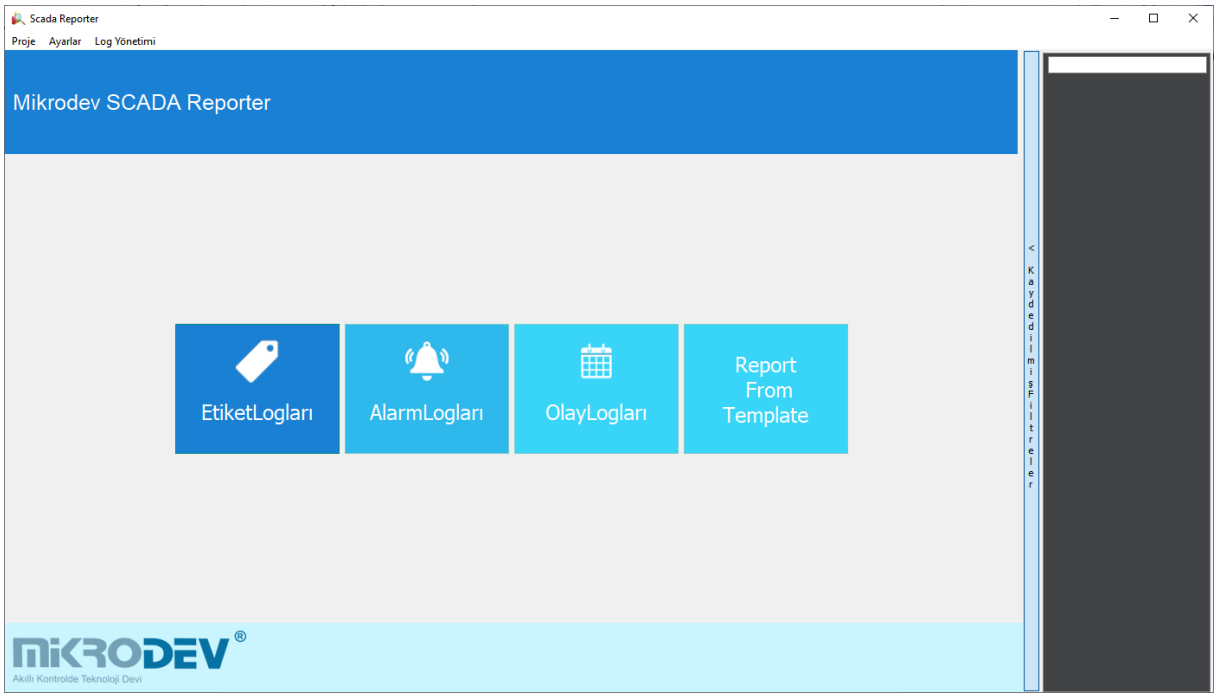

Figure 165 Scada Reporter Tool

12.1 Accessing the SCADA Reporter Application

The application can be launched from the SCADA Editor interface via Tools > Scada Reporter.

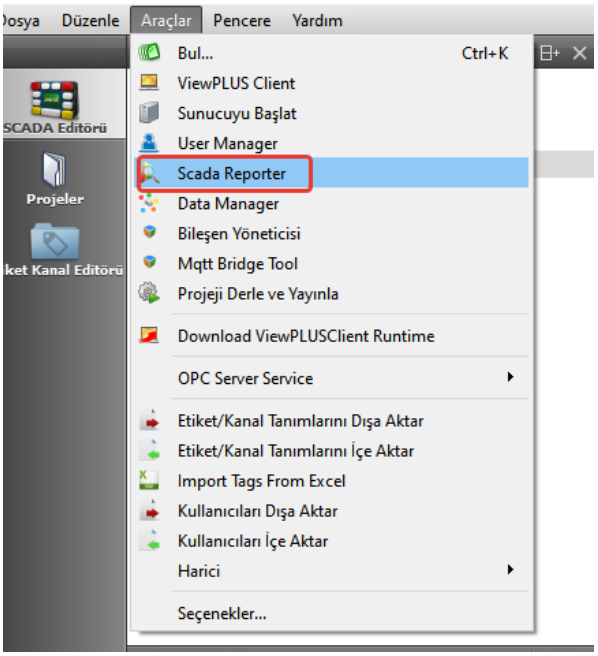

Figure 166 Launching the Reporter Tool

12.2 Main Interface

When the SCADA Reporter application is launched, the user is presented with four main categories:

Tag Logs: Used to view time-based logged tag values in the SCADA system.

Alarm Logs: Used to view records such as the timestamp, status, and type of alarms occurring in the system.

Event Logs: User interactions such as logins, value writings, and page entries can be monitored in this section.

Report From Template: Used to create reports using pre-defined filter templates.

12.3 Top Menu Options

12.3.1 Project Menu

Connect to Project: Connects to the SCADA database to be reported.

Options: Used to configure settings such as connection information and output file path.

Exit: Closes the application.

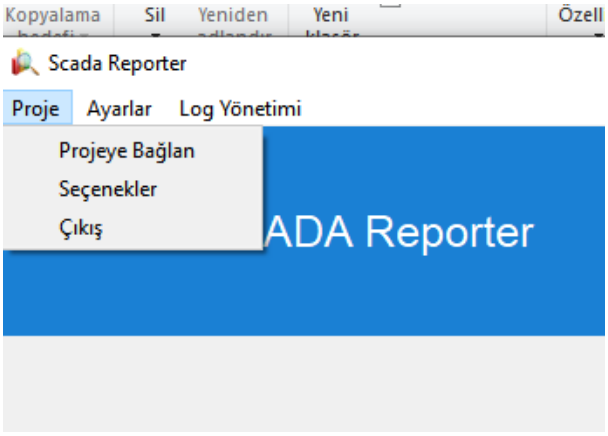

Figure 167 Project Menu

12.3.2 Settings Menu

Language: Changes the application language to Turkish or English.

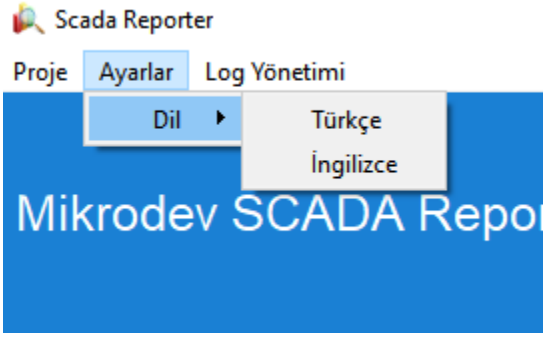

Figure 168 Settings Menu

Note: The reporter tool must be restarted for this selection to take effect.

12.3.3 Log Management Menu

Clear Logs: Used to delete recorded data from the system. This is permanent and should be used carefully.

Backup Logs: Creates a backup of the log data in the database.

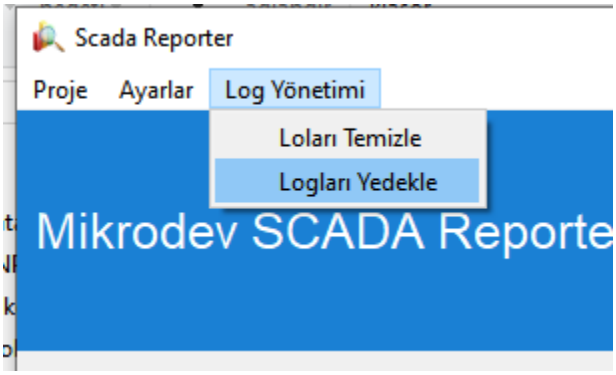

Figure 169 Log Management Menu

12.4 Using Saved Filters via SCADA Buttons

Filters created in the ViewPLUS SCADA Reporter application can be accessed not only from the Reporter interface but also via a button on the SCADA screen. This feature provides user-friendly access to preconfigured report filters.

12.4.1 Saving a Reporting Filter

After applying a filter for any tag, alarm, or event log on the Reporter screen:

- Click the "Save This Filter" button located at the top right.
- In the window that appears, enter a name for the filter and click **OK** to complete the save process.

The saved filter will appear in the "Saved Filters" section on the right panel.

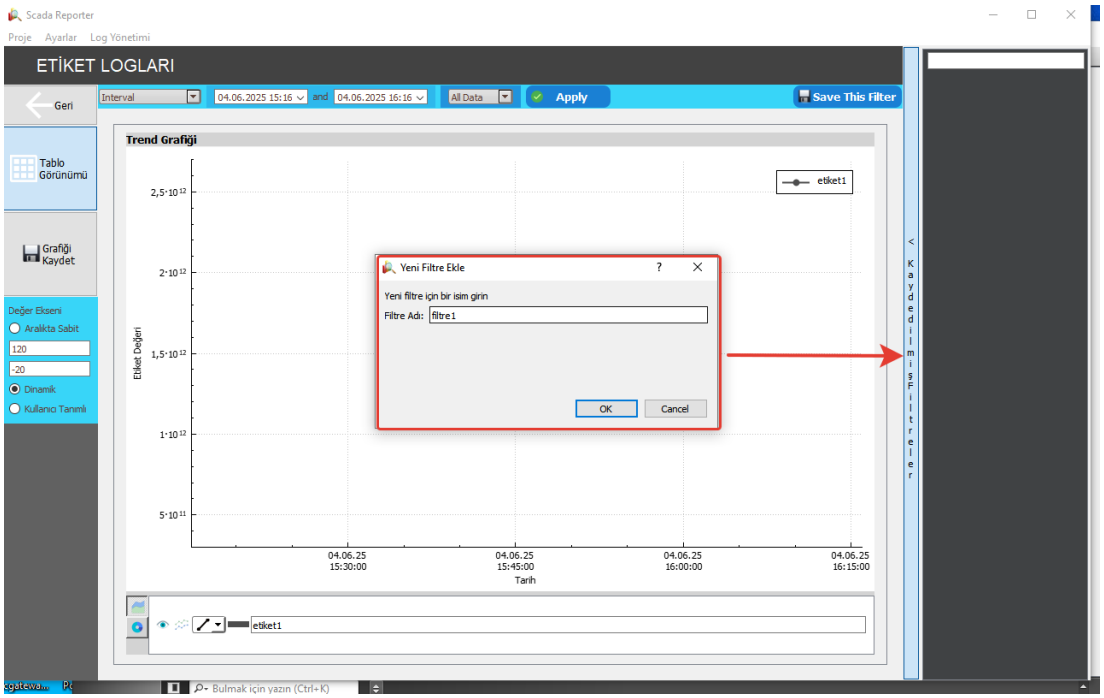

Figure 170 Saving a Reporting Filter

12.4.2 Copying the Filter Link Address

- Right-click on the desired filter in the filter list on the right.
- From the menu that appears, select "Copy Link Location" to copy the link to the clipboard.

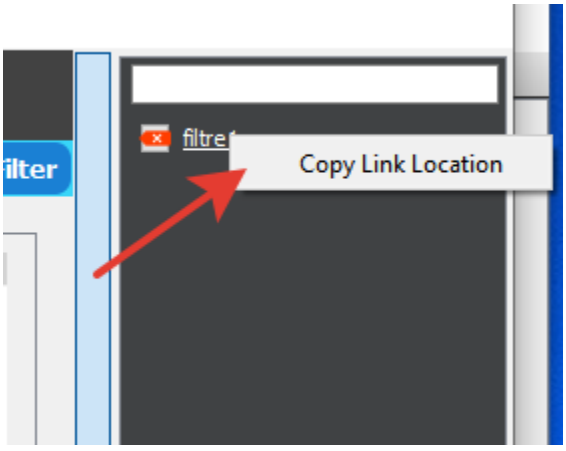

Figure 171 Copying the Filter Link Address

12.4.3 Assigning the Filter Link to a Button in SCADA Editor

- A button component is created in the SCADA Editor.
- After selecting the button, navigate to the "Events" tab in the bottom-right panel.
- Set the "Event Action" to GO_TO_LINK
- In the lower section, check the Link checkbox (under Web), and paste the previously copied filter link into the input field.

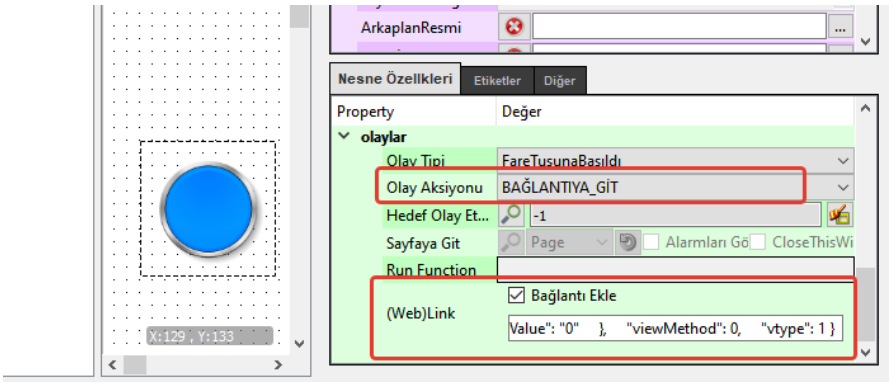

Figure 172 Assigning a Filter Link to a Button

Now, when the user clicks this button on the SCADA screen, the report window defined with the corresponding filter opens directly. This method:

Accelerates operational reporting,

Provides easy access to frequently used reports,

Reduces user errors and complex interface interactions.

13 ViewPLUS SCADA Data Manager

In the ViewPLUS SCADA system, the Data Manager is a management tool that provides archiving, backup, data deletion, and organization functions to maintain long-term system efficiency.

The Data Manager is launched from the Tools tab in the main SCADA interface.

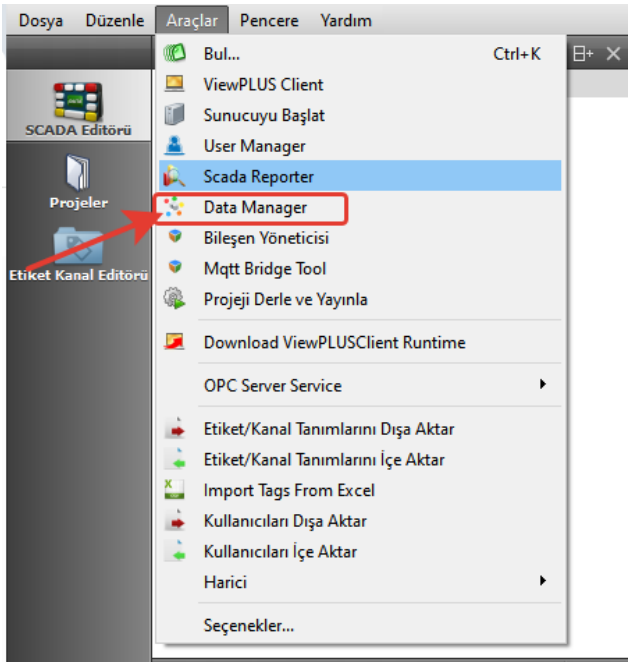

Figure 173 Launching the Data Manager

When the Data Manager window opens, two main functions are displayed on the main screen:

- **Archiving**: Archiving of old data.
- **Backup**: Backing up the entire database.

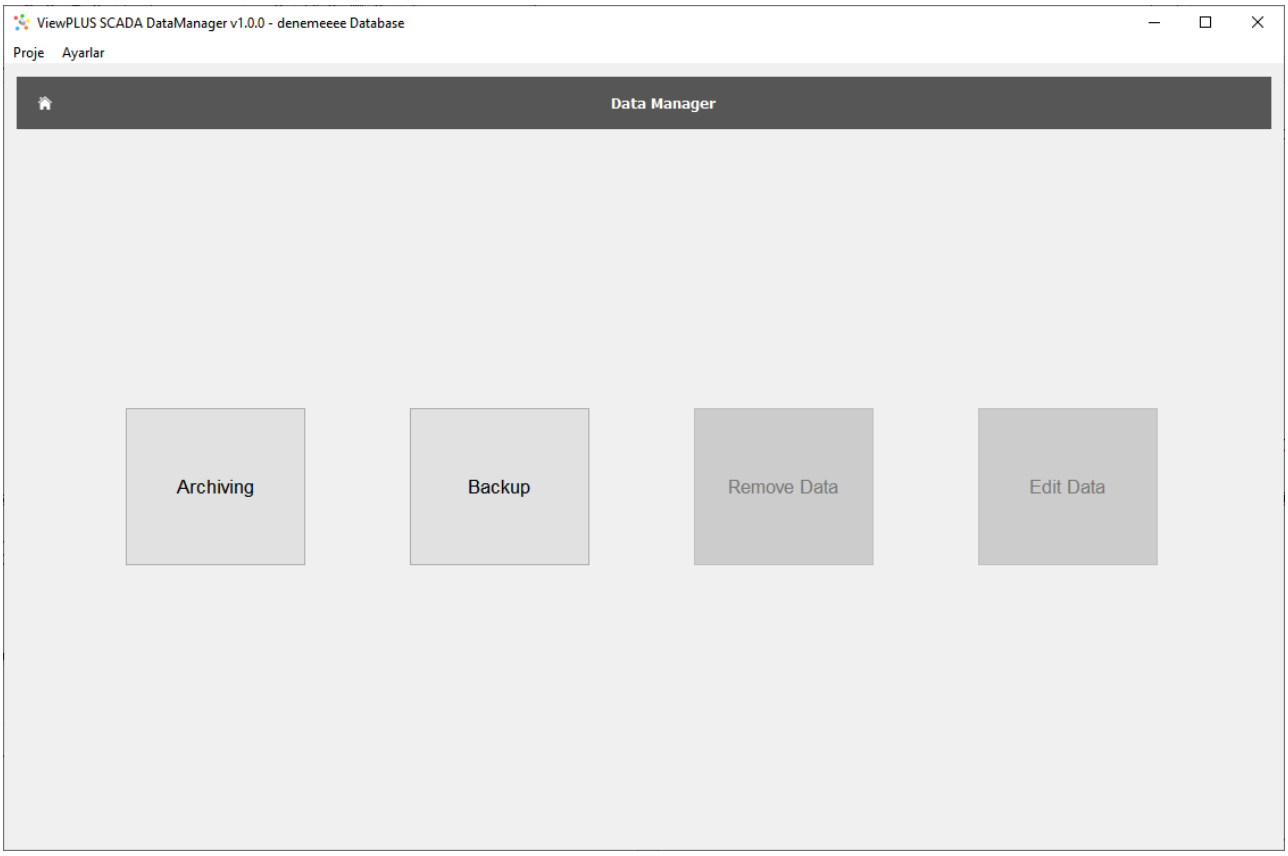

Figure 174 Data Manager Main Screen

13.1 Archiving Process

Clicking the Archiving option opens the archiving screen. On this screen, the temporary folder, database path, and archive date are specified.

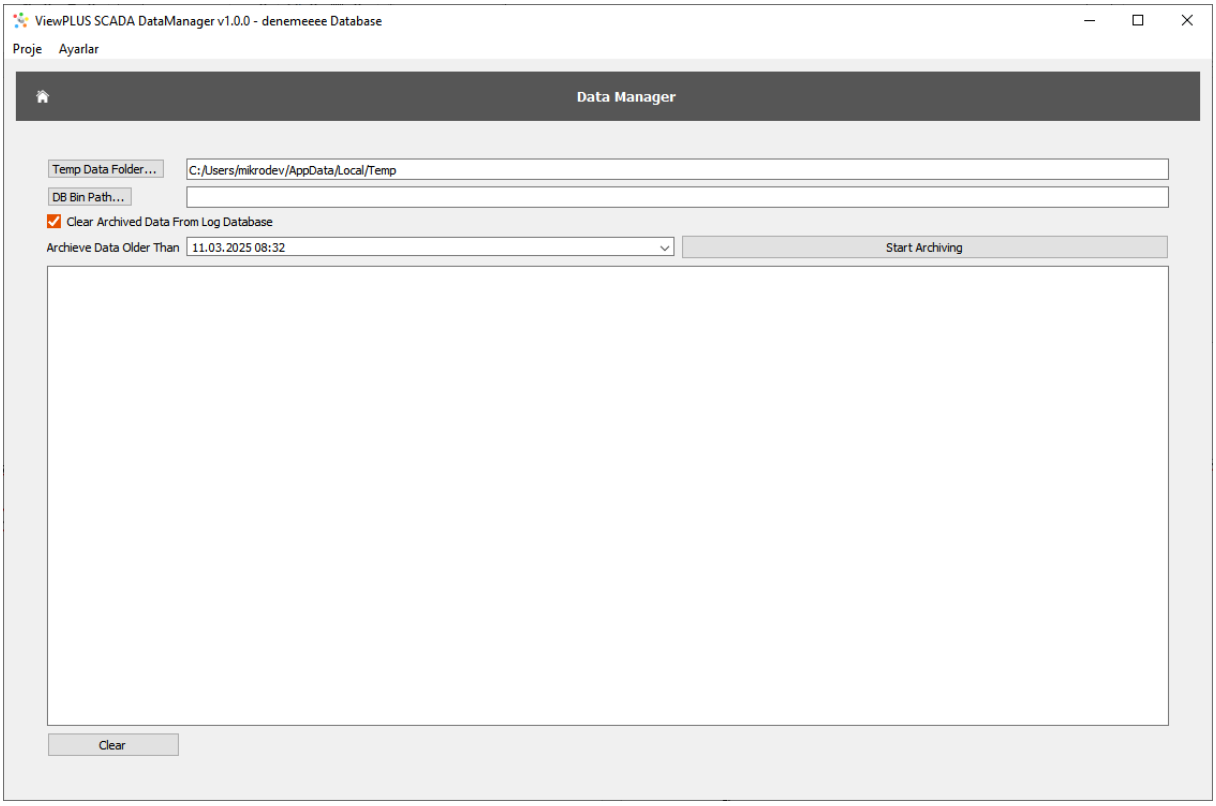

Figure 175 Archiving Screen

Temp Data Folder: The location where temporary data is stored.

DB Bin Path: The path where the database executables are located.

Example Path: C:/Program Files/PostgreSQL/13/bin

Clear Archived Data From Log Database: If checked, archived data will be deleted from the main database.

Archive Data Older Than: All logs prior to this date will be archived.

Start Archiving: Initiates the archiving process.

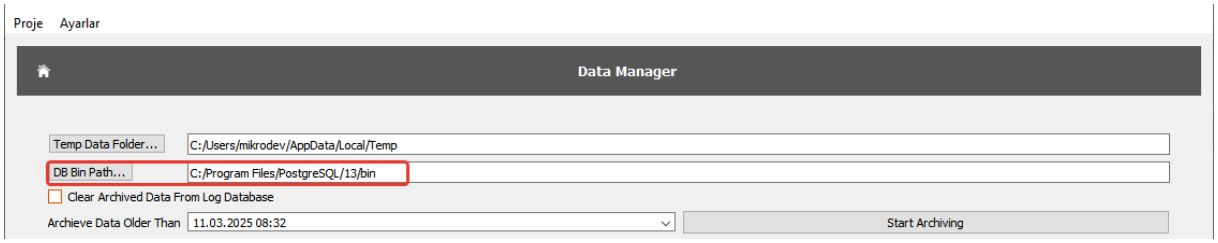

Figure 176 Selecting Bin Folder for Archiving Process

If the Clear Archived Data From Log Database option is checked, a warning will appear asking whether to delete the logs after archiving has started.

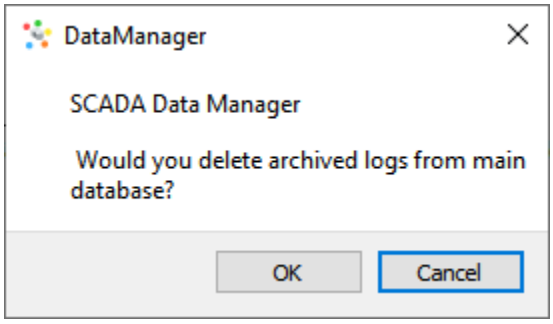

Figure 177 Warning Message for Deleting Logs After Archiving

- **OK**: Deletes the archived logs from the main database.
- **Cancel**: Keeps the logs in the main database.

After the archiving is completed, a message saying **"Archiving completed"** will appear.

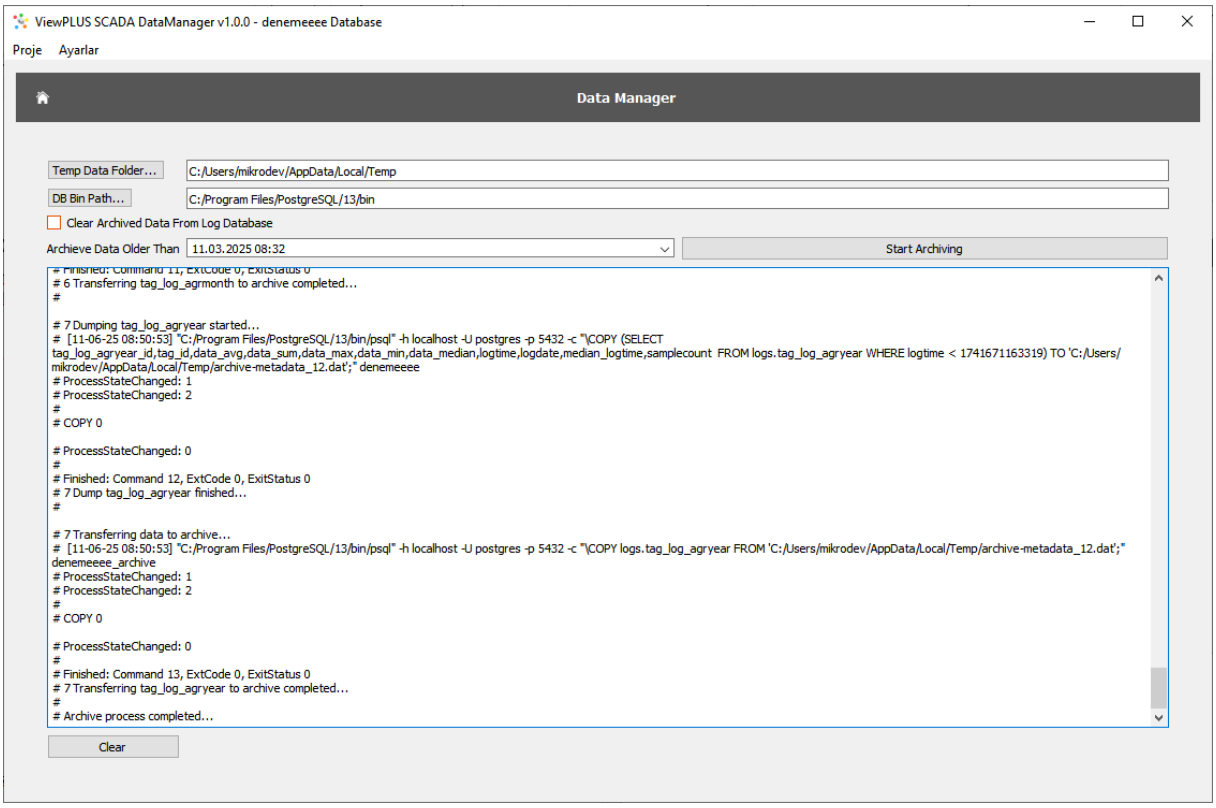

Figure 178 Archiving Completed Message

A new archive database is created, for example: denemeeee_archive

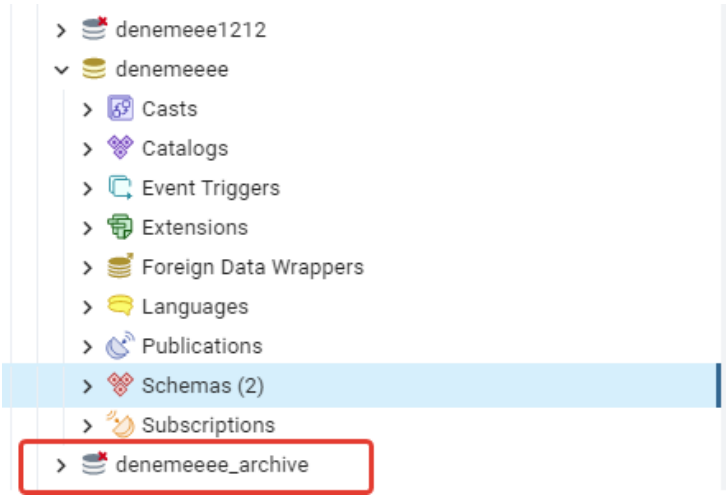

Figure 179 Resulting Database After Archiving

Archived logs can be viewed from this database.

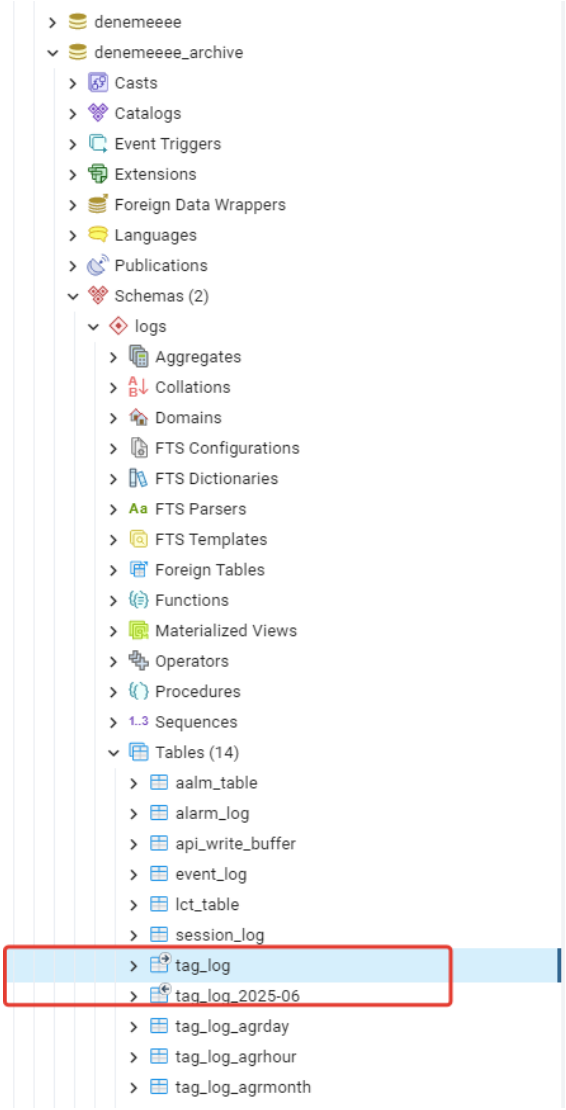

Figure 180 Archive Database

13.2 Backup Process

Clicking the Backup button from the main Data Manager screen opens the backup interface.

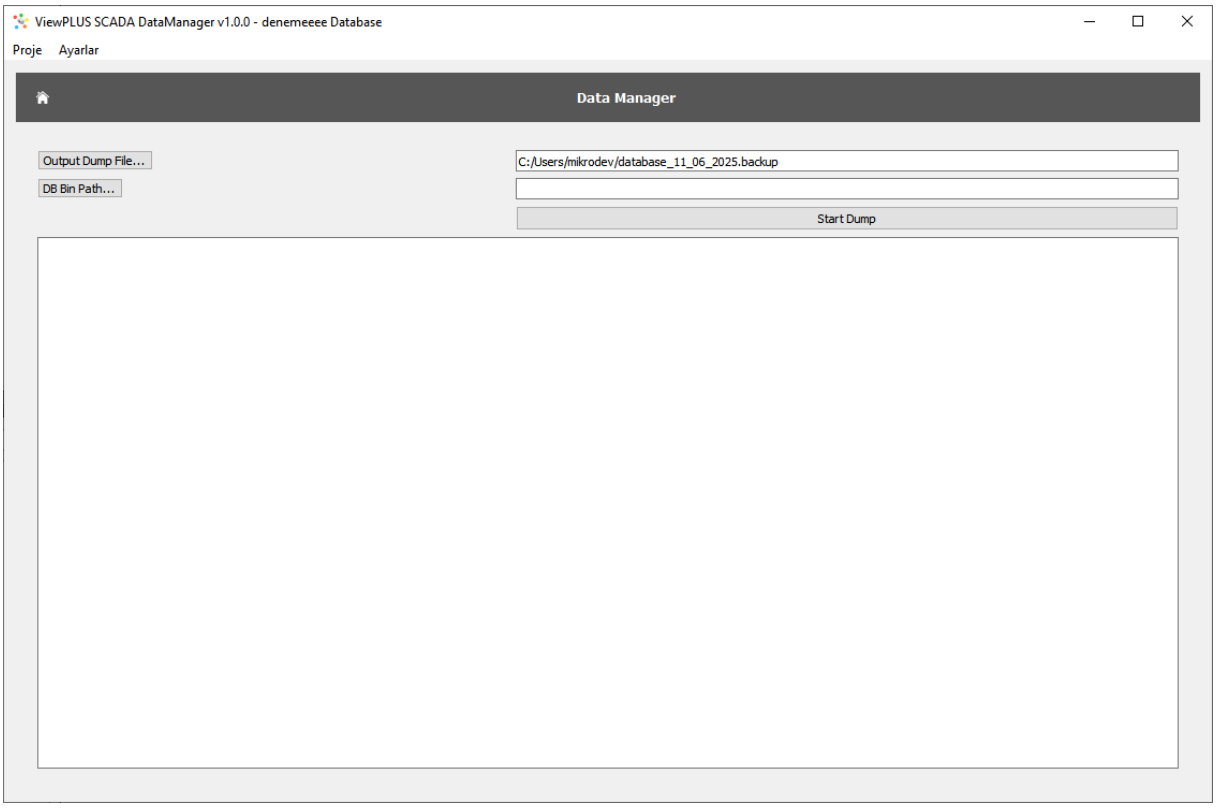

Figure 181 Backup Main Screen

- **Output Dump File:** The path where the backup file will be saved.
- **DB Bin Path:** The folder containing the database executables.
- **Start Dump:** Starts the backup process.

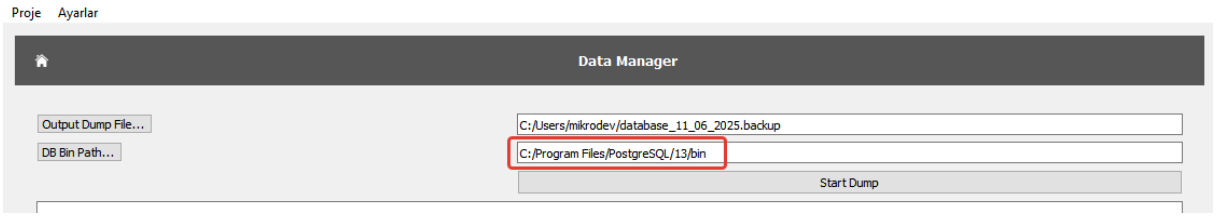

Figure 182 Selecting Bin Folder for Backup Process

Once the backup is complete, a message saying "Backup completed" will be shown.

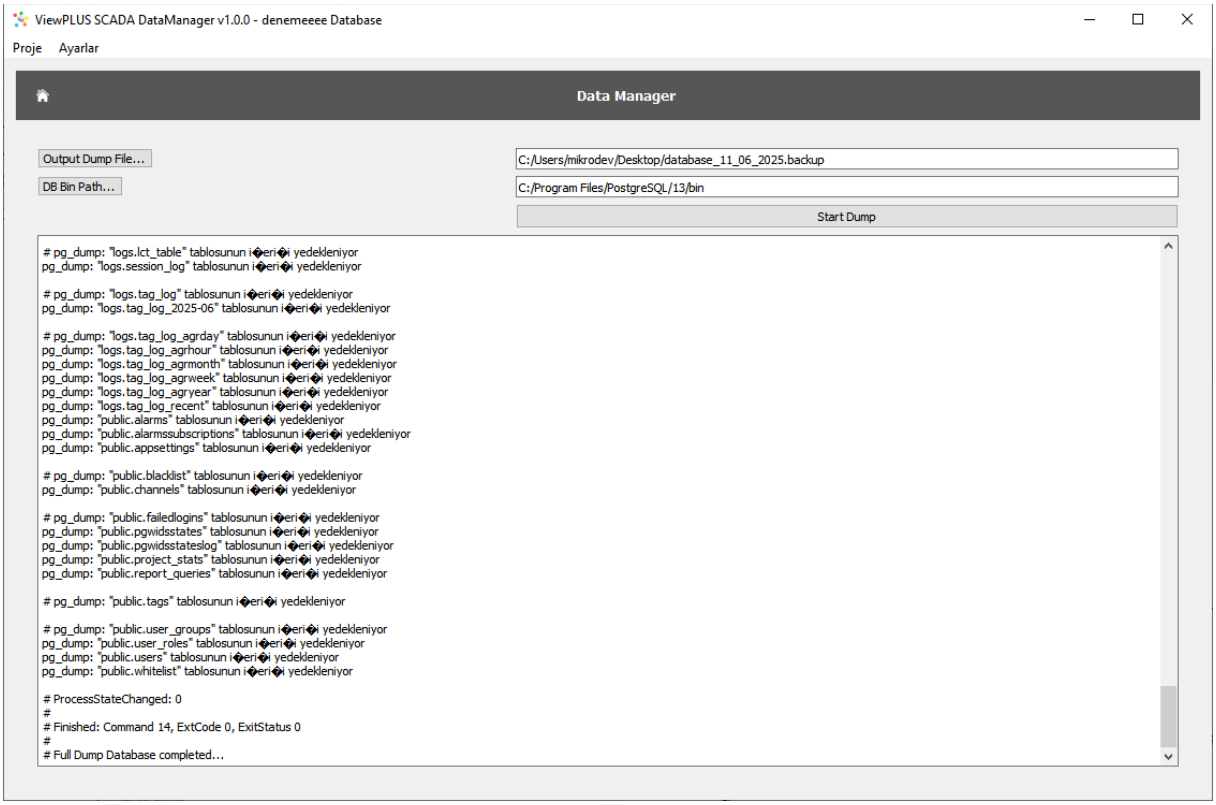

Figure 183 Backup Completion Message

After the backup is completed, a file with a .backup extension will be created in the specified directory. This file can later be used to restore the database or migrate it to another system.

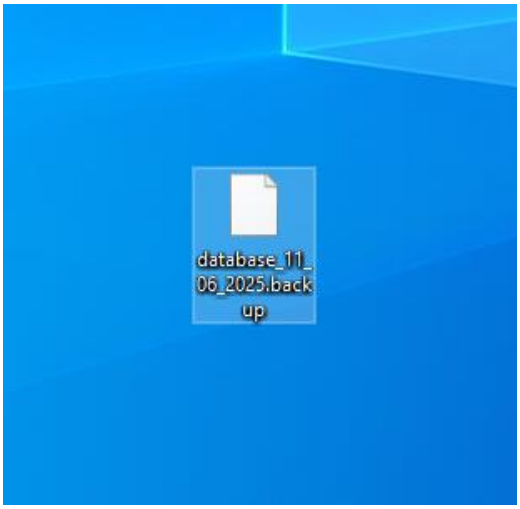

Figure 184 Backup File Generated After Backup Process

14 MQTT Bridge Tool

Devices in the field communicate with ViewPLUS SCADA using protocols such as IEC 60870-5-104 (IEC 104) and Modbus. All data transmitted to the SCADA system via these protocols can be forwarded to the MQTT protocol with the help of the MQTT Bridge Tool. This enables seamless integration of the data into cloud systems, mobile applications, or external analytics platforms.

14.1 Main Interface

MQTT Bridge Tool editör ekranı, ViewPLUS SCADA editörü içinden Araçlar > MQTT Bridge Tool menüsü aracılığıyla açılır.

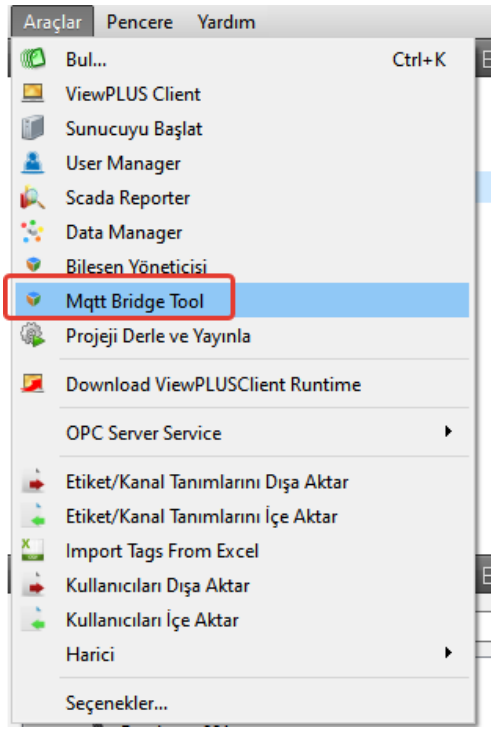

Figure 185 Launching MQTT Bridge Tool

All defined connections are listed in the MQTT Bridge Tool interface. Each connection displays information such as Client ID, Topic, data publishing method (On Change / Periodic), Payload, etc.

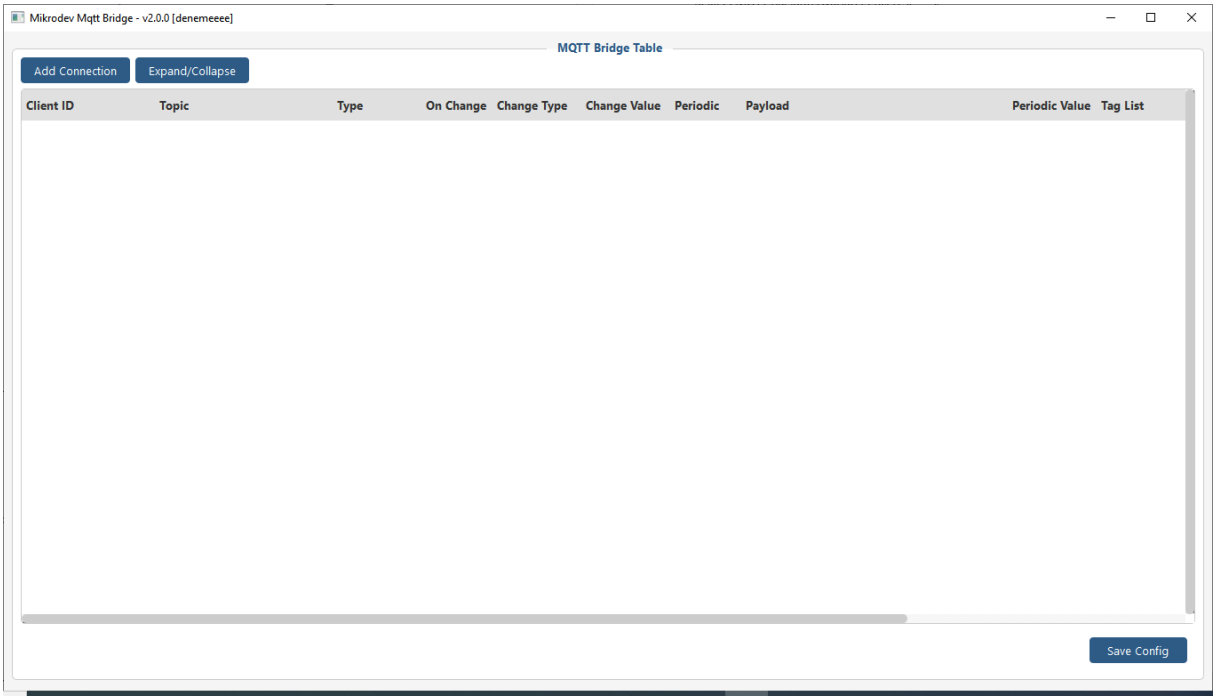

Figure 186 MQTT Bridge Tool Main Screen

14.2 Adding a New MQTT Connection

Clicking the "Add Connection" button opens the screen to input new MQTT connection information.

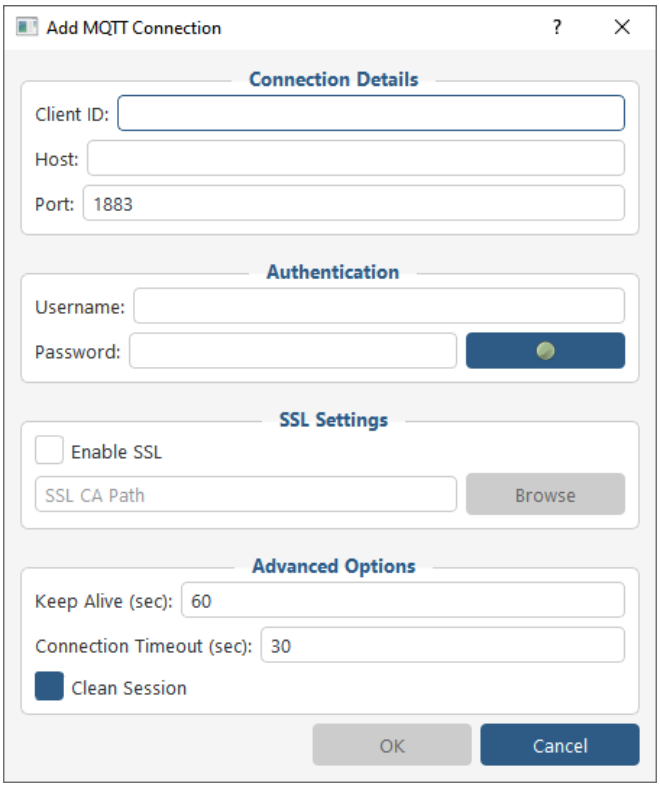

Figure 187 MQTT Connection Settings Window

- Connection Details

Client ID: Unique identifier of the MQTT client.

Host: Broker IP address.

Port: Connection port (default: 1883).

- Authentication

Username / Password: For brokers requiring authentication.

- SSL Settings

Enable SSL: Option to enable secure connection.

SSL CA Path: Path to the certificate.

- Advanced Options

Keep Alive: Duration for keeping the connection alive.

Connection Timeout: Timeout duration for connection.

Clean Session: If enabled, starts a clean session on each connection.

14.3 Defining Publish and Subscribe Topics

Once a connection is defined, right-clicking on it allows the following operations:

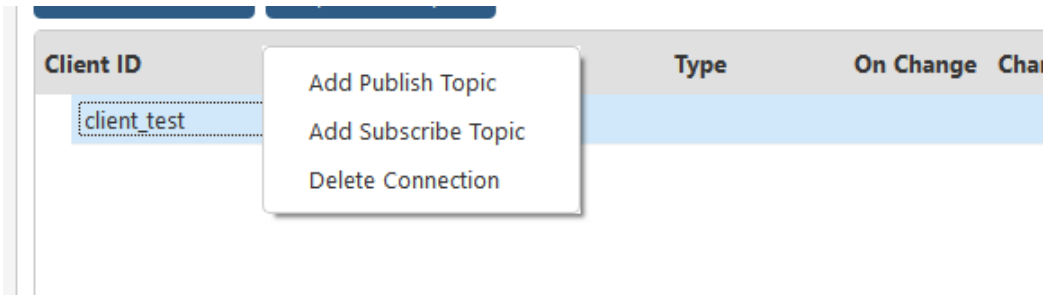

Figure 188 Defining Publish and Subscribe Topics

Add Publish Topic: Defines a new topic for MQTT publishing.

Add Subscribe Topic: Defines a topic to listen to incoming MQTT messages.

Delete Connection: Deletes the connection.

14.3.1 Publish Topic

Using the "Add Publish Topic" option, you can define the topic through which data will be published.

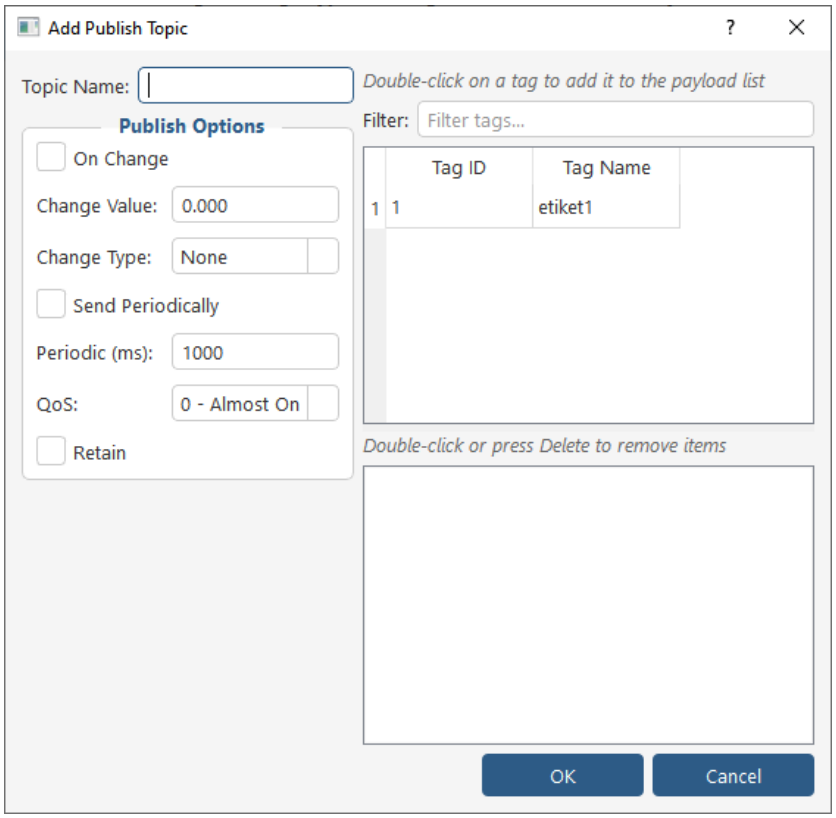

Figure 189 Add Publish Topic Screen

Topic Name: Name of the MQTT topic to which data will be published.

On Change: Enables sending data on tag change.

Change Value: Threshold value for triggering data send.

Change Type: Types such as None, Level, Percent, Integral.

Send Periodically: Enables data to be sent at specified intervals.

Periodic (ms): Interval in milliseconds.

QoS: MQTT quality of service level.

Retain: Retains the last message on the broker for new clients.

Double-clicking a tag from the list on the right inserts it into the payload.

14.3.2 Subscribe Topic

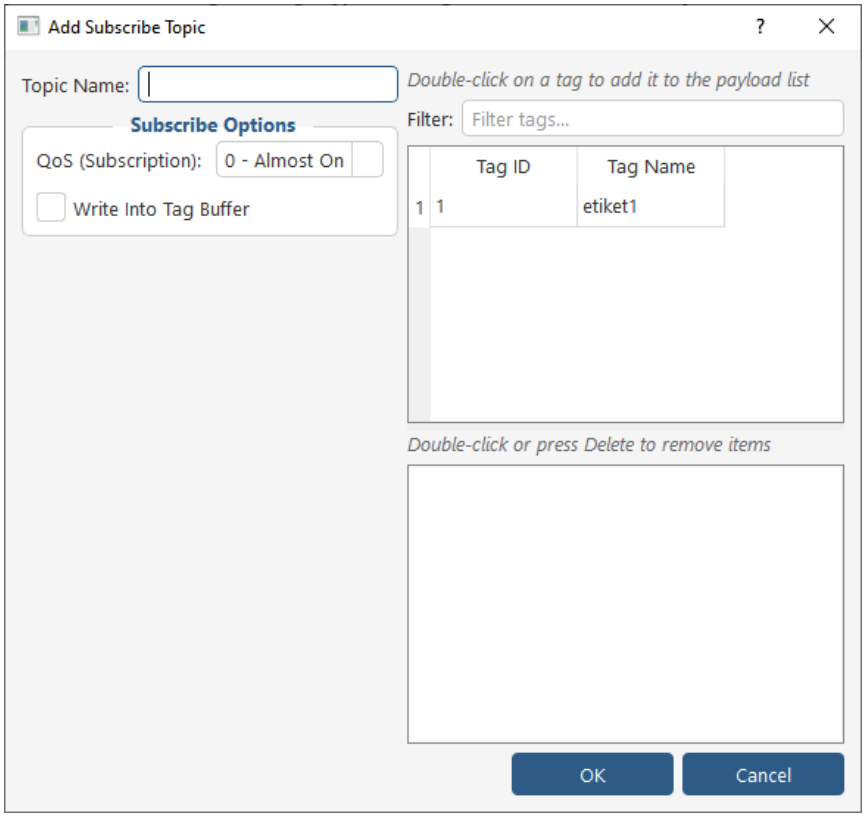

Figure 190 Add Subscribe Topic Screen

Topic Name: Name of the MQTT topic to be listened to.

QoS (Subscription): Quality level for incoming data.

Tags from the list on the right can be double-clicked to map them with the payload.

These settings allow publishing data from SCADA to the MQTT broker.

14.4 Example Payload

[

{"ts":"1733141120","values":{"1":"30"}},

{"ts":"1733141120","values":{"2":"28"}},

{"ts":"1733141120","values":{"3":"32"}},

{"ts":"1733141120","values":{"4":"31"}},

{"ts":"1733141120","values":{"5":"15"}}

]

ts: Timestamp

values: Tag ID and its value

With the help of the MQTT Bridge Tool, SCADA projects gain the ability to publish measurement data to online platforms, enable real-time monitoring, and support IoT integration.

14.5 Starting the Driver for MQTT Bridge Tool

To ensure that the MQTT Bridge Tool is automatically enabled when the server starts, follow the steps below:

- Open the Other Settings tab from the left menu in the ViewPLUS ServerEngine application.
- Locate the Mqtt Bridge section at the bottom of the opened screen.
- Check the Enable Bridge option. When this setting is enabled, the MQTT Bridge driver will be active after the next server restart.

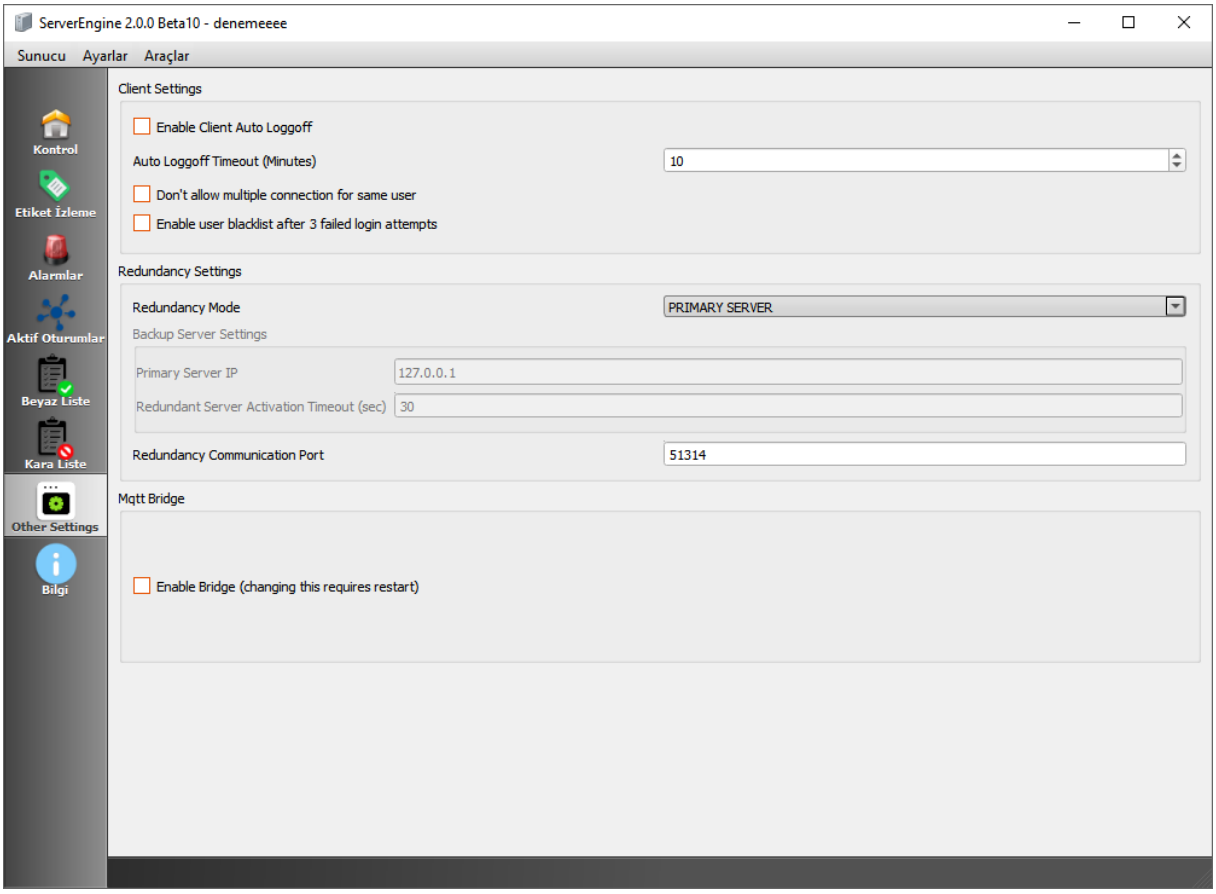

Figure 191 MQTT Bridge Driver Setting (ServerEngine Other Settings)

Note: The server application must be restarted for this setting to take effect.

15 ViewPLUS SCADA Web API

With the ViewPLUS SCADA Node WebAPI, you can access field devices and perform remote control operations. This guide includes all steps from installation to data query examples.

To download the WebAPI folder: SCADA Node Web API

15.1 Database Connection

In order for the WebAPI to connect to the database, system environment variables must be defined:

1. Open the Edit the system environment variables window from the Start menu. Click on Environment Variables under the Advanced tab.

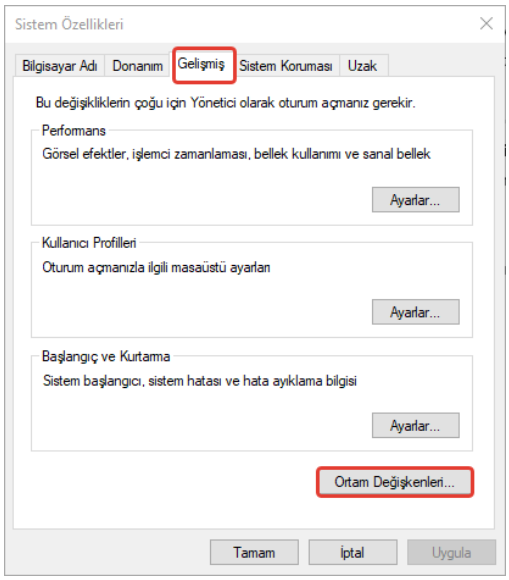

Figure 192 Selecting System Environment Variables

2. Under System Variables, click New and add the following variables:

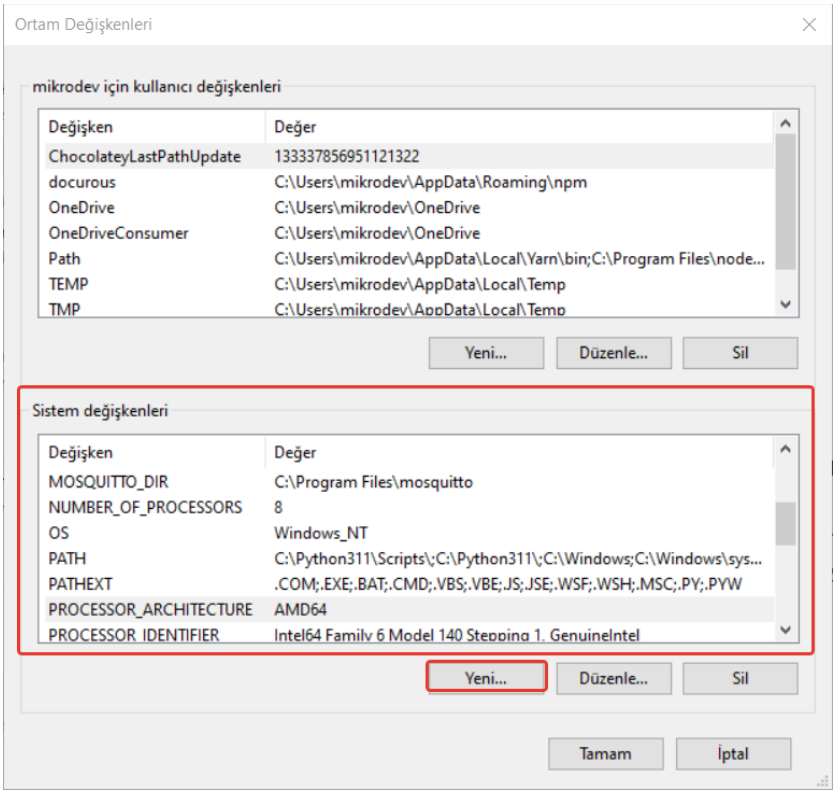

Figure 193 Defining a New System Variable

- For PostgreSQL;

HR_DBTYPE = POSTGRES

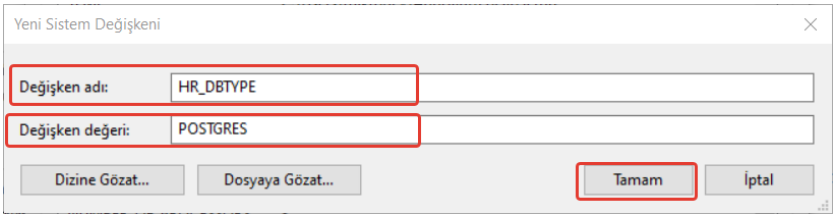

Figure 194 Variable for PostgreSQL (1)

HR_PGCONNECTIONSTRING = postgres://postgres:qwx123@127.0.0.1:5432/basliksiz2

(Database user: postgres, password: qwx123, server address: 127.0.0.1, port: 5432, SCADA project: basliksiz2)

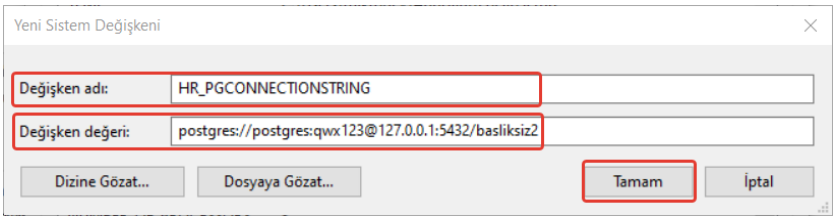

Figure 195 Variable for PostgreSQL (2)

- For ORACLE:

HR_CONNECTIONSTRING = 127.0.0.1/orcl

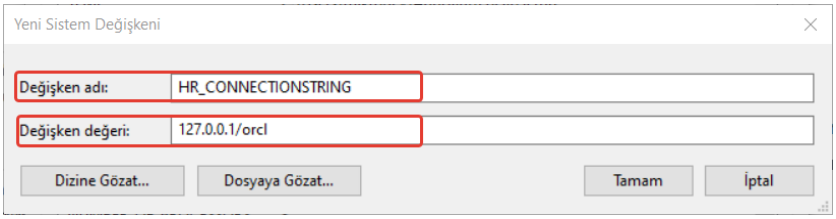

Figure 196 Variable for ORACLE (1)

HR_USER = ORACLETEST

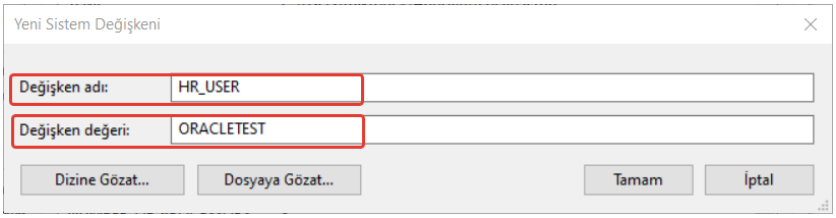

Figure 197 Variable for ORACLE (2)

HR_PASSWORD = qwx123

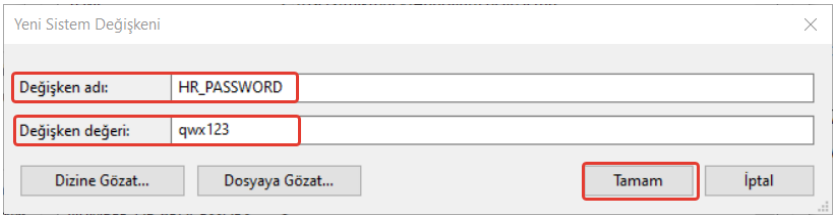

Figure 198 Variable for ORACLE (3)

15.2 Starting the Server

1. Ensure Node.js is installed on your computer. Download Node.js
2. After defining the environment variables, you can change the port number in the config/web-server.js file within the WebAPI folder (default: 3000).

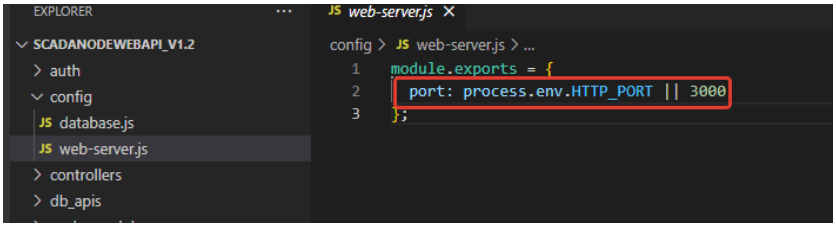

Figure 199 Updating Web Server Port

3. Update the JWT security key in config.js (e.g., changethissecret).

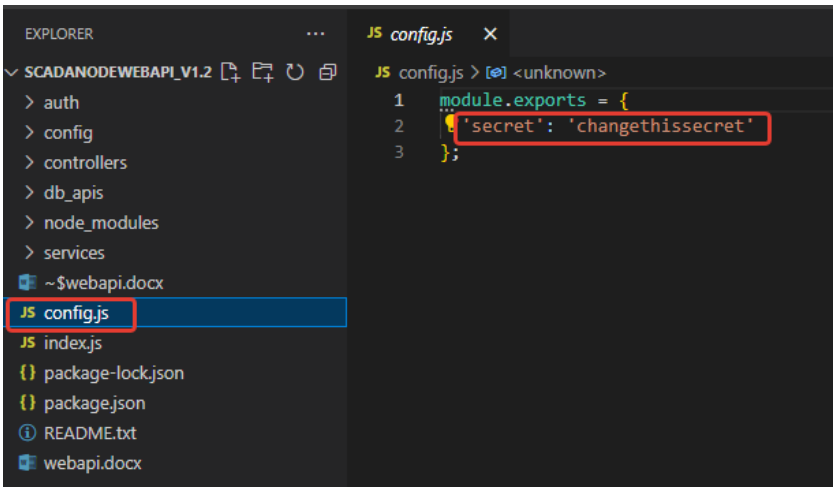

Figure 200 Updating JSON Web Token

4. Check if the correct version of the PostgreSQL library is installed:

npm list

npm install pg@8.11.1 # gerekirse

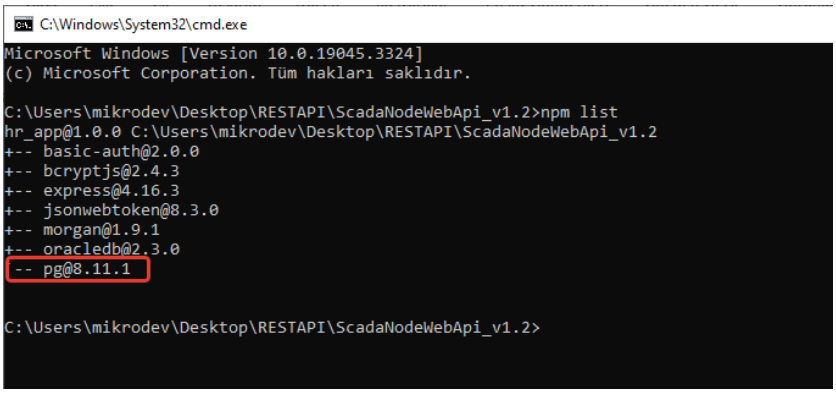

Figure 201 Verifying Database Version

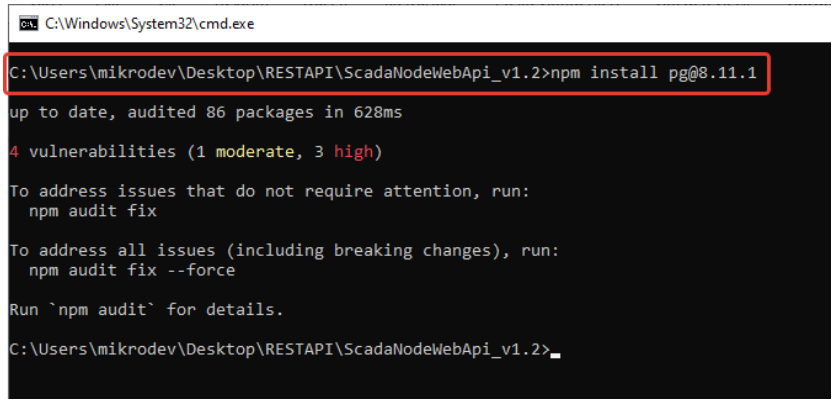

Figure 202 Updating Version via Command Line

5. To start the server:

node .

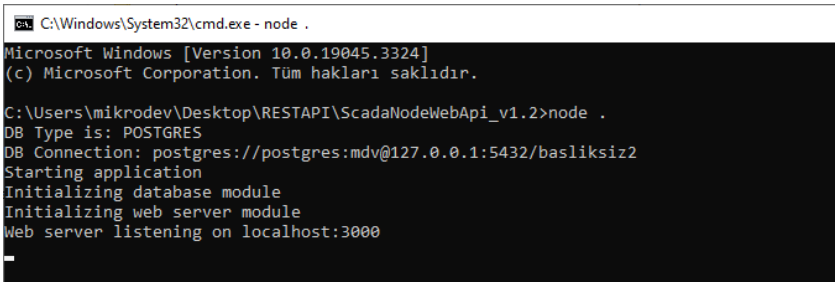

Figure 203 Starting Web API Server

15.3 HTTP Client (Thunder Client)

To test API requests, you can install the Thunder Client extension in Visual Studio Code:

1. Install VS Code.
2. Open the Extensions menu on the left and search for Thunder Client, then install it.
3. Pin the Thunder Client icon to the sidebar.

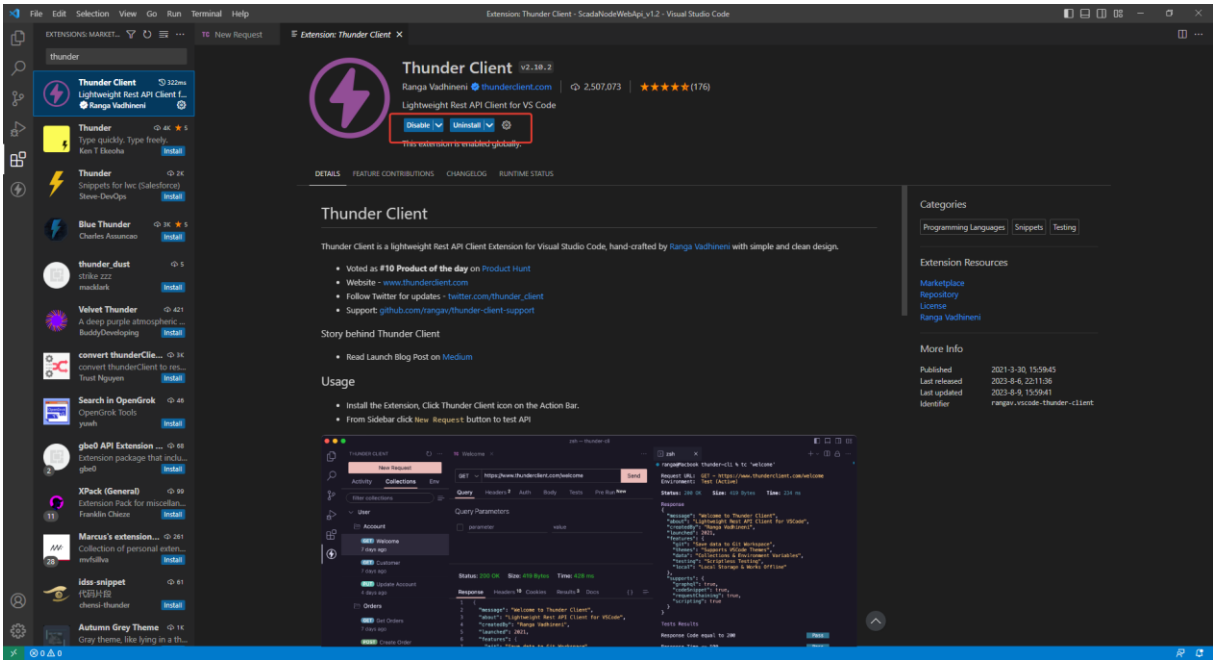

Figure 204 Installing Thunder Client Extension

15.4 Retrieving Access Token

1. Create a user in the SCADA editor (must have report permissions).
2. Send a POST request using Thunder Client:

URL: http://localhost:3000/api/auth/login

Body:

```
{
  "username": "<username>",
  "password": "<password>"
}
```

3. Save the access token returned in the response for future use.

Note: The server must be running (node .) to retrieve the token.

15.5 Fetching Data with Web API

Use the token to send GET requests:

In Thunder Client, go to the Headers tab and remove all existing headers.

Add a new header

Key: x-access-token

Value: your previously received token (without quotes)

- **List All Tags**

GET http://<server_IP>:3000/api/auth/tags

Returns: All tag details (id, tag name, address, etc.)

- **Get a Single Tag**

GET http://<server_IP>:3000/api/auth/tags/<tag_id>

- **Access Alarms**

GET http://<server_IP>:3000/api/auth/alarms

Note: Use the tag_id in the alarm object to access related tag data.

- **List Channels**

GET http://<server_IP>:3000/api/auth/channels

- **Get a Single Channel**

GET http://<server_IP>:3000/api/auth/channels/<channel_id>

- **All Real-Time Tag Values**

GET http://<server_IP>:3000/api/auth/rt_values

- **Single Real-Time Tag Value**

GET http://<server_IP>:3000/api/auth/rt_values/<tag_id>

Notes:

- A token is required for all requests.
- The token must be sent in the x-access-token header.
- The server IP refers to the device where WebAPI is installed.
- The port number can be changed from config/web-server.js.
- Token expiration may vary depending on the application.

16 Installing the OPC UA Server Service

16.1 Activating the OPC Server Service

Within ViewPLUS SCADA, go to the Tools tab and select Install OPC Server Service to perform the installation.

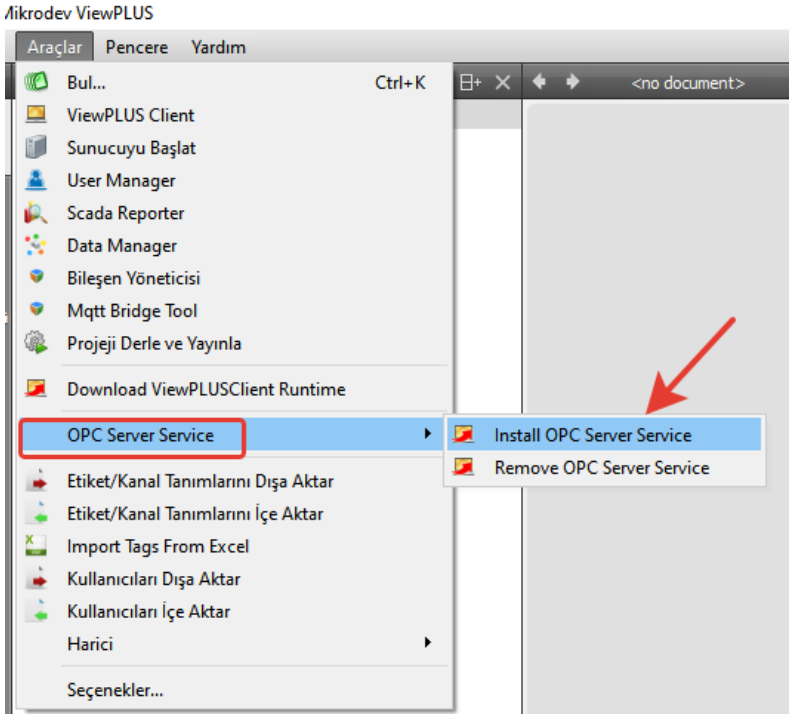

Figure 205 Installing the OPC Service

After installation, the service named VPlusScadaOPCService should be running and visible in Task Manager.

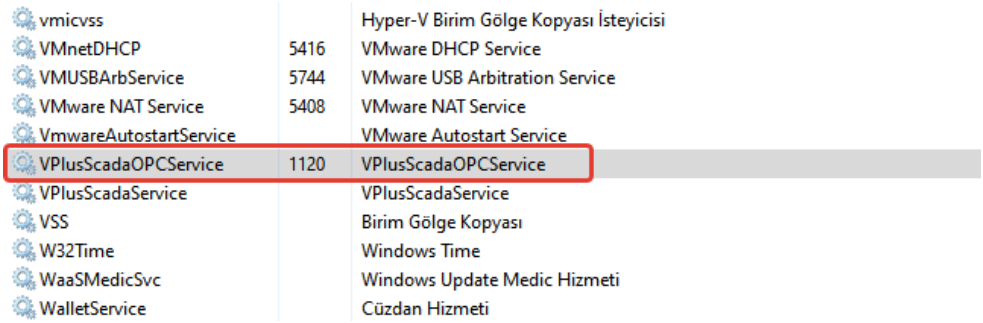

Figure 206 Starting the OPC Service

16.2 Connecting via UAExpert

16.2.1 Installing UAExpert

Download and install UAExpert from:

https://www.unified-automation.com/downloads/opc-ua-clients.html

16.2.2 Adding a Server

1. In the UAExpert interface, go to Server → Add.

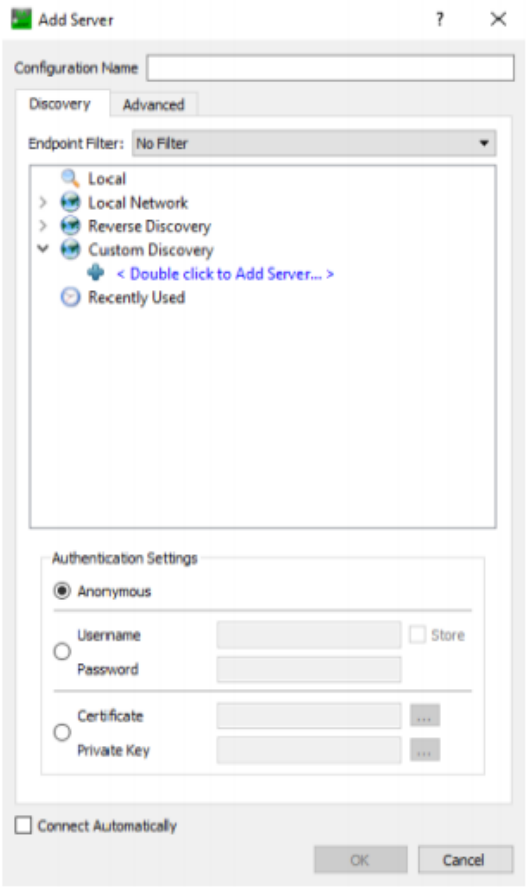

Figure 207 Adding a Server via UAExpert

2. In the popup window, enter the address opc.tcp://localhost:43344 (if ViewPLUS SCADA is running on the same computer).

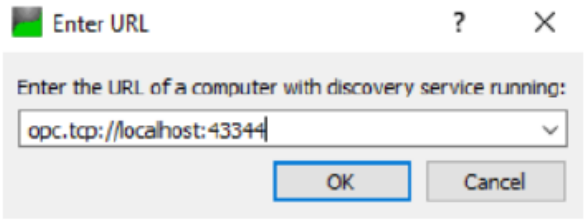

Figure 208 UAExpert Server Settings

16.2.3 Establishing a Connection

After the server appears in the list, select an appropriate endpoint to connect. It is recommended to choose the highest security level.

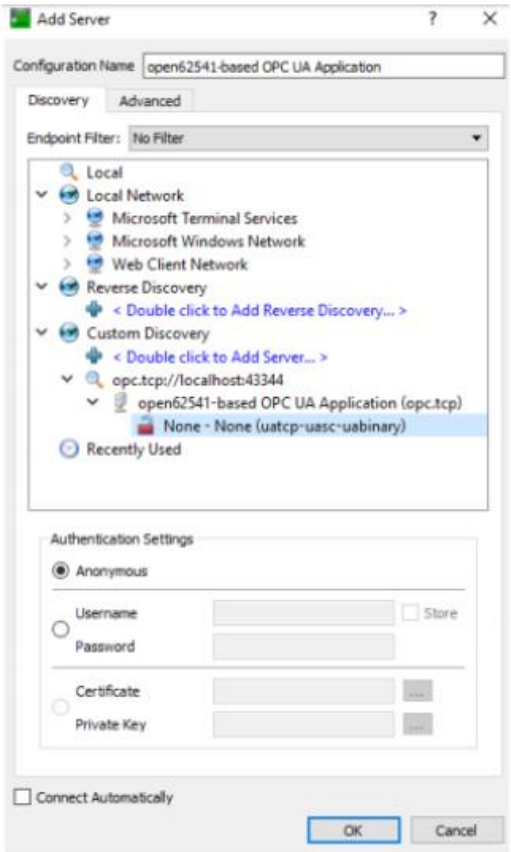

Figure 209 Endpoint List

16.2.4 User Login

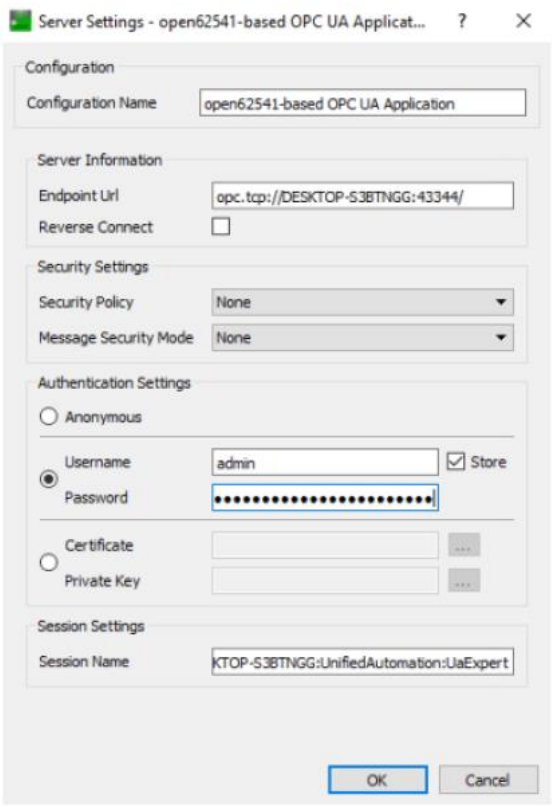

Figure 210 Username and Password Entry for Server Connection

Log in using the username and the hashed password stored in the ViewPLUS SCADA project database.

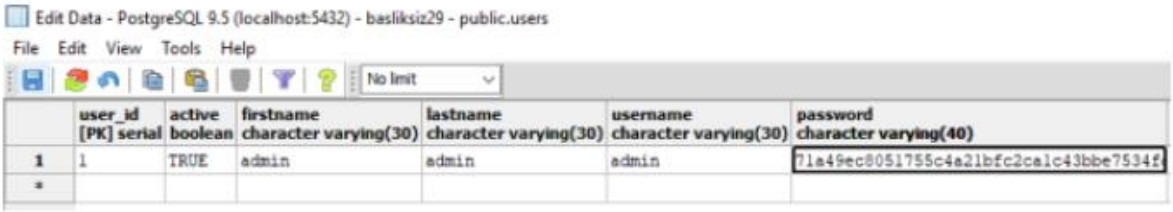

Figure 211 ViewPLUS SCADA Database User Password

16.3 Connecting via KEPServerEX

16.3.1 Installing KEPServerEX

Download and install KEPServerEX from:

http://www.opcturkey.com/indir

16.3.2 Defining Channel and Device

1. Creating a new channel and select OPC UA Client as the type.

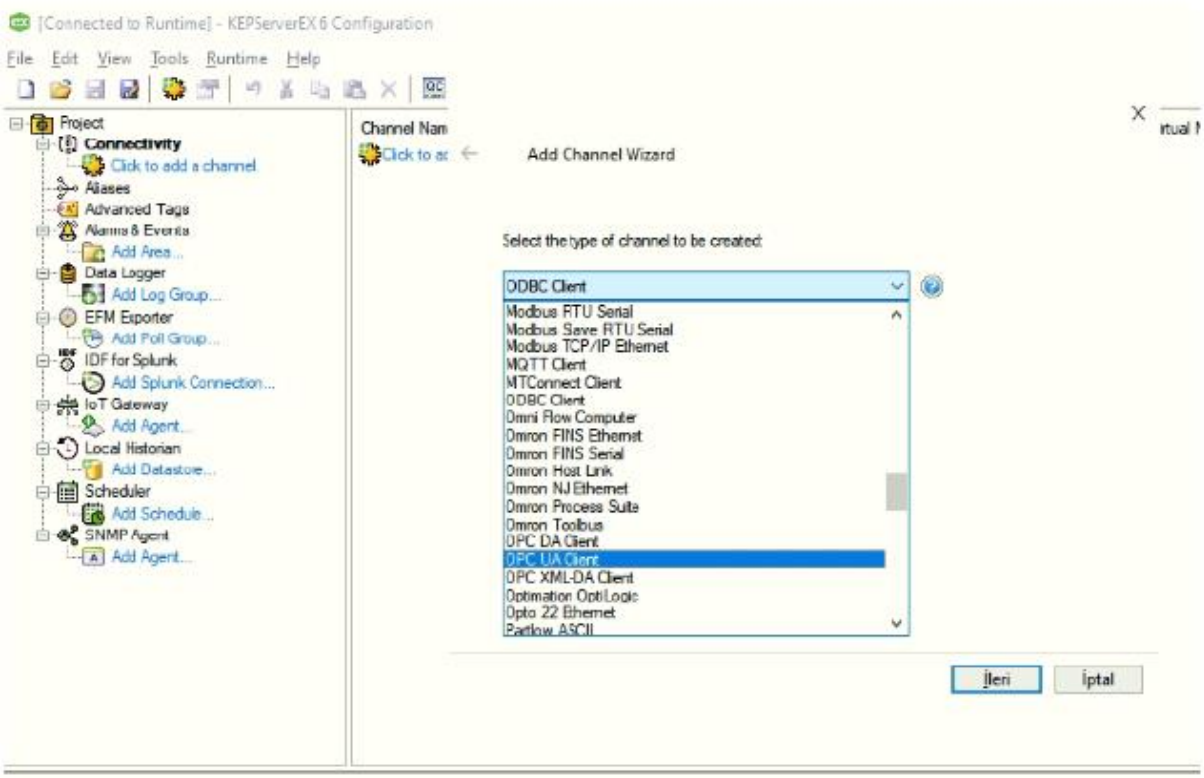

Figure 212 Adding OPC UA Client Channel in KEPServer

2. Set the Endpoint URL to opc.tcp://localhost:43344. The security policy can be set to None if needed.

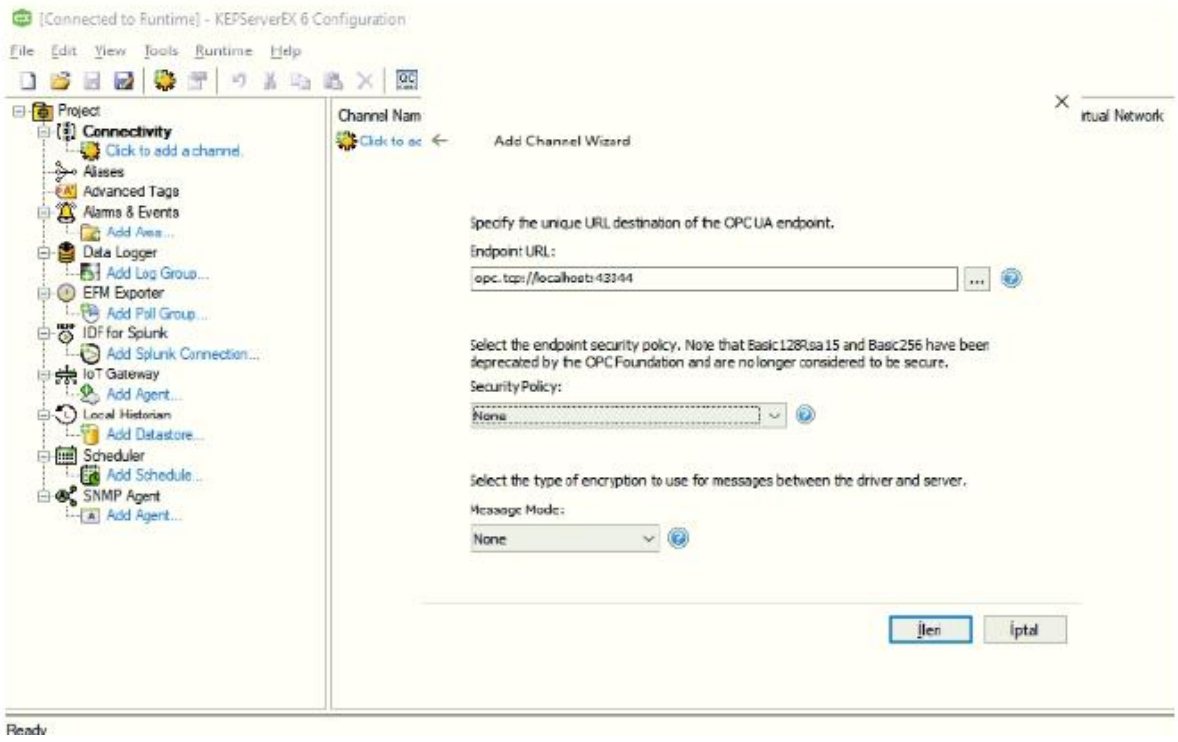

Figure 213 KEPServer Connection Settings

3. Enter the username and hashed password from the ViewPLUS SCADA project.

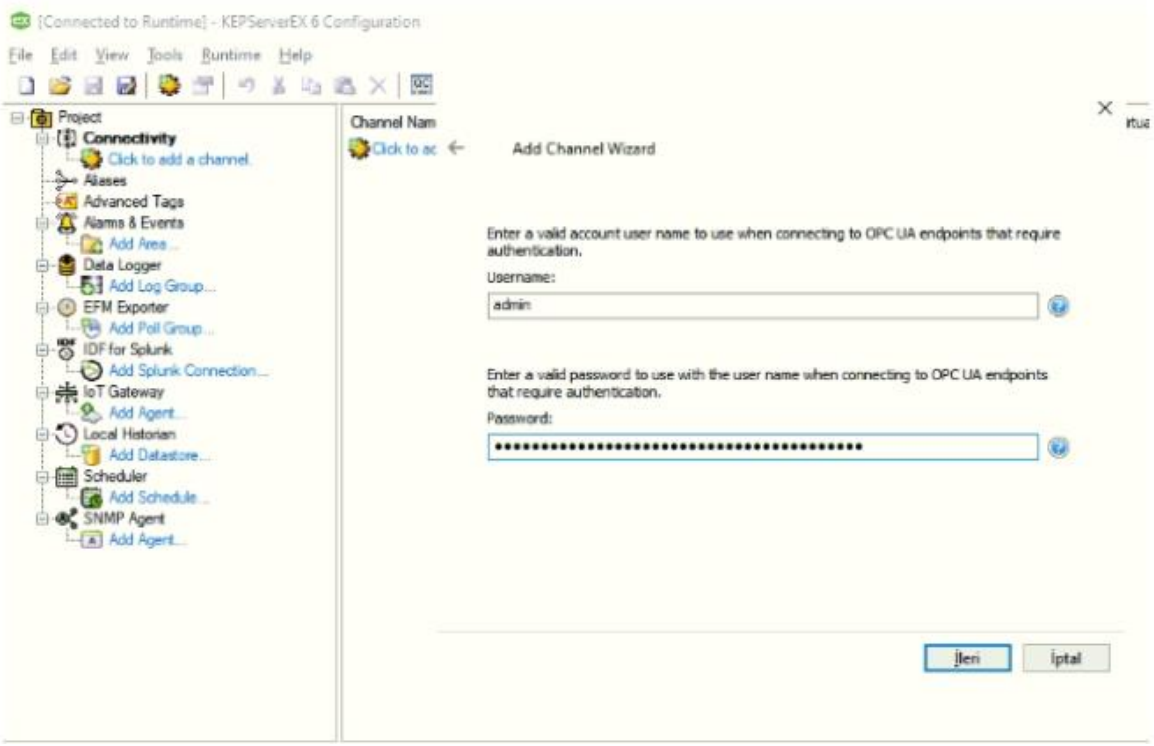

Figure 214 Defining Username and Password in KEPServer

4. After creating the channel, define the device.

16.3.3 Viewing Tags

Channels and tags defined in ViewPLUS SCADA can be viewed through the OPC server and mapped in the KEPServerEX interface.

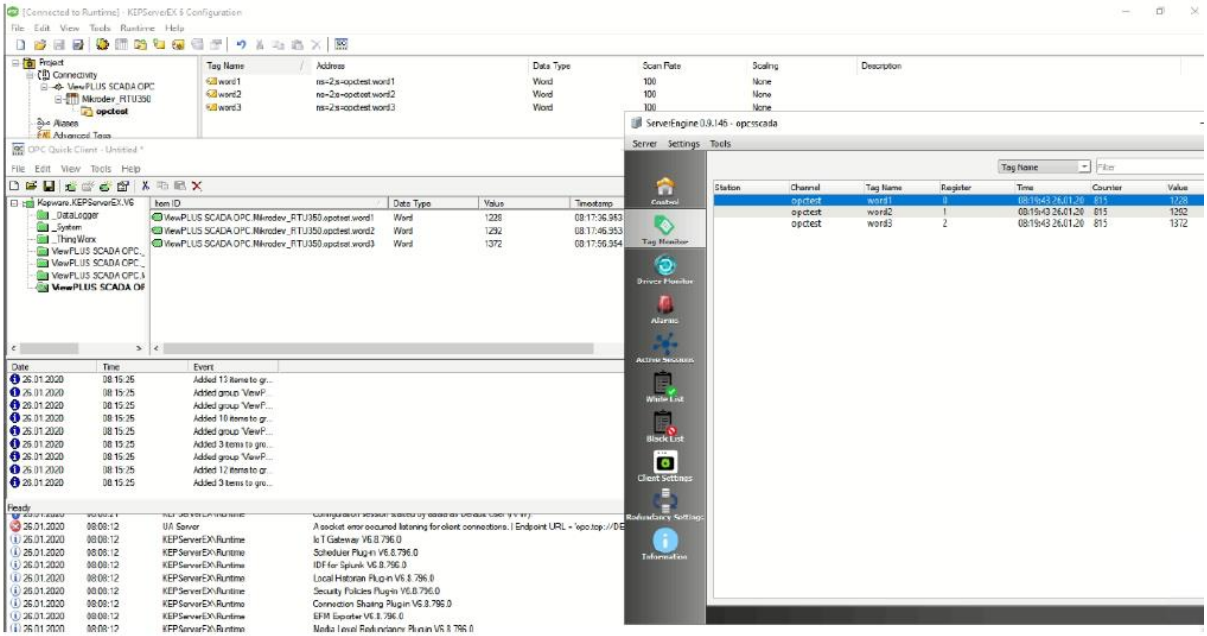

Figure 215 Viewing SCADA Tags in KEPServer

16.4 Creating OPC Certificates (for Linux)

Use the createcert.sh script file to generate certificates. This process requires a Linux system with OpenSSL installed.

Below is a secure and advanced OPC UA certification procedure. This script generates private keys and certificates for both the Certificate Authority (CA) and the server, and also supports CRL (Certificate Revocation List).

16.4.1 Certificate Authority

CA Creation

mkdir ca

openssl genpkey -algorithm RSA -pkeyopt rsa_keygen_bits:2048 -out ca/ca.key

openssl req -new -x509 -days 3600 -key ca/ca.key -subj "/O=MyServer/CN=localhost" -out ca/ca.crt

openssl x509 -in ca/ca.crt -inform pem -out ca/ca.crt.der -outform der

CRL Creation:

mkdir demoCA

touch ./demoCA/index.txt

echo "1000">./demoCA/crlnumber

openssl ca -crldays 3600 -keyfile ca/ca.key -cert ca/ca.crt -gencrl -out ca/ca.crl

openssl crl -in ca/ca.crl -inform pem -out ca/ca.der.crl -outform der

16.4.2 Creating Server Certificate

mkdir server

Define certificate extensions:

cat < server/exts.txt

[v3_ca]

subjectAltName=DNS:localhost,DNS:<PC-Adı>,IP:127.0.0.1,IP:<ScadaServerPCIP>,URI:urn:unconfigurated:application

basicConstraints=CA:TRUE

subjectKeyIdentifier=hash

authorityKeyIdentifier=keyid,issuer

keyUsage=digitalSignature,keyEncipherment

extendedKeyUsage=serverAuth,clientAuth,codeSigning

EOF

Generate server private key:

openssl genpkey -algorithm RSA -pkeyopt rsa_keygen_bits:2048 -out server/server.key

openssl rsa -in server/server.key -inform pem -out server/server.key.der -outform der

Create CSR

openssl req -new -sha256 -key server/server.key -subj "/O=MyServer/CN=localhost" -out server/server.csr

16.4.3 Signing the Server Certificate

openssl x509 -days 3600 -req -in server/server.csr -extensions v3_ca -extfile server/exts.txt -CAcreateserial -CA ca/ca.crt -CAkey ca/ca.key -out server/server.crt

openssl x509 -in server/server.crt -inform pem -out server/server.crt.der -outform der

16.4.4 Copying Certificates

For UA Expert

cp ca/ca.crt.der ~/.config/unifiedautomation/uaexpert/PKI/trusted/certs/

cp ca/ca.der.crl ~/.config/unifiedautomation/uaexpert/PKI/trusted/crl/

For ViewPLUS SCADA Project:

mkdir -p $PROJECT_PATH/certs

cp server/server.crt.der $PROJECT_PATH/certs/

cp server/server.key.der $PROJECT_PATH/certs/

As a result of these steps, the OPC UA server is securely certified and can establish trusted communication with clients such as UAExpert or KEPServerEX.