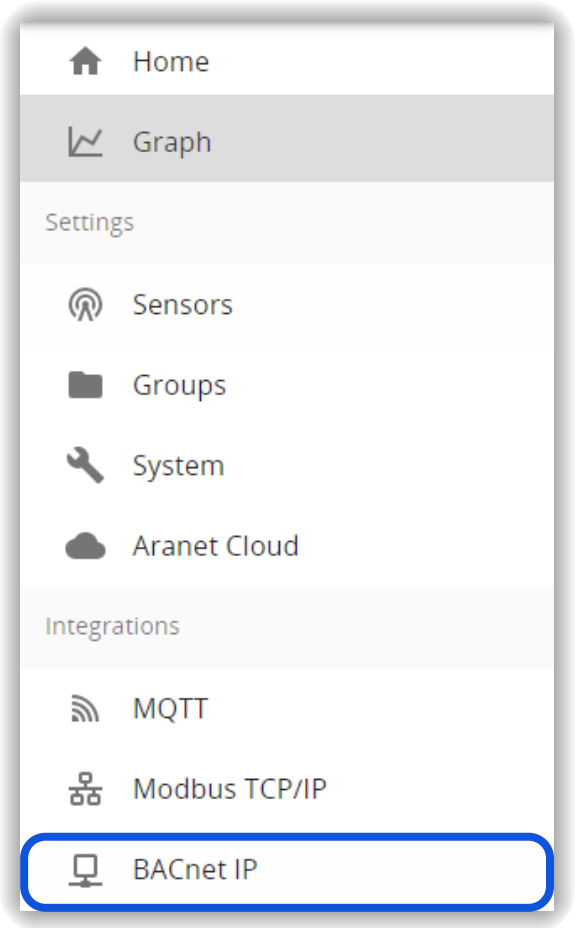
Aranet PRO BACnet/IP server

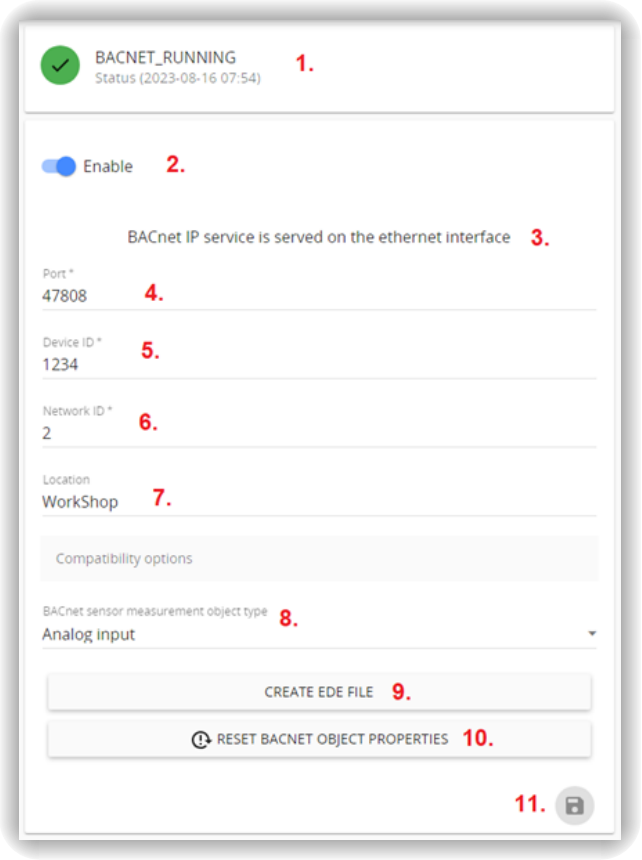
BACnet/IP server functionality of Aranet base stations is a powerful solution for seamless integration of Aranet PRO sensor readings into BACnet-enabled building automation and control networks. The Aranet BACnet IP compatibility streamlines installation and integration processes, providing a cost-effective and scalable solution for modern building management systems.

Enabling BACnet integration



- In the web GUI main page BACnet/IP server configuration is available under “BACnet/IP”.
- The BACnet/IP service can only be accessed through an Ethernet (wired) network interface.
- The full functionality of the BACnet/IP service requires an additional license to be uploaded.

Configuration options



1. Status of the BACnet/IP server process;
2. Enable/Disable the server process;
3. Information about the BACnet service availabilty on Ethernet interface (wired) only;
4. Specify the UDP port number (default: 47808);
5. The BACnet device *object instance ID* (default: 1234);
6. The Network number (default: 1);
7. Set the device *object property location*;
8. Select the BACnet *object type* to be used for sensor *measurement objects*;
9. Create/download the *engineering data exchange (EDE) file*;
10. Reset the values of the writeable BACnet object (Device, AI, AV) properties to defaults;
11. Save the BACnet/IP settings.

Available object types and properties

Object types used in Aranet PRO BACnet server implementation are (1) *Device object type*, (2) *Analog Input object type* or (3) *Analog Value object type* (depends on the *object type* selection in “Compatibility options”).

Device object properties

Property	Access rights
Active Change-of-Value (CoV) subscriptions	Read-only
APDU timeout	Read-only
Application software version	Read-only
Daylight saving status[1]	Read-Write
Database revision	Read-only
Description	Read-only
Device address binding	Read-only
Firmware revision	Read-only
Local date	Read-only
Local time	Read-only
Location	Read-only
Maximum accepted APDU length	Read-only

(Table continued in next page)

(Table continued from previous page)

Property	Access rights
Model name	Read-only
Number of APDU retries	Read-only
Object identifier	Read-only
Object list	Read-only
Object name	Read-Write
Object type	Read-only
Protocol object types supported	Read-only
Protocol revision	Read-only
Protocol services supported	Read-only
Protocol version	Read-only
Segmentation supported	Read-only
System status	Read-only
UTC offset[2]	Read-Write
Vendor identifier	Read-only
Vendor name	Read-only

[1] The BACnet client system needs manual updates for daylight saving status. It doesn't automatically update and can't be adjusted through the Aranet PRO base web GUI, which operates on UTC time for its backend processes.

[2] UTC offset must be updated from the BACnet client system as it is not being auto-updated and cannot be set using Aranet PRO base web GUI. BACnet *UTC offset* is inverse of common practice. If your *UTC offset* is -5 h of GMT, then BACnet *UTC offset* is +5 h. BACnet *UTC offset* is expressed in minutes, therefore convert the common *UTC offset* from hours to minutes.

Analog input / Analog value (measurement) object properties

Property	Access rights
Property	Access rights
CoV Increment	Read-Write
Description[3]	Read-only
Event state	Read-only
Object identifier	Read-only
Object name	Read-Write
Object type	Read-only
Out of service	Read-Write
Present value[4]	Read-only
Reliability	Read-only
Status flags	Read-only
Units	Read-only

(Table continued in next page)

(Table continued from previous page)

Property	Access rights
9997 (proprietary) — Time of last present value update	Read-only
9998 (proprietary) — Date of last present value update	Read-only

[3] Description is a combination of sensor hexadecimal UID, group name, sensor name, measurement type and unit, e.g., `[500ACE] roomEnv Contact pulse meter pulses (pulses)` where:

- `[500ACE]` — hexadecimal UID
- `roomEnv` — group name
- `Contact pulse meter` — sensor name
- `pulses` — measurement type
- `(pulses)` — unit

[4] Present value can be changed in case if *Out of service* value is set to "True". It applies for both supported *object types* (*AI* and *AV*).

Object name default assignment

By default, the *Device object name* has the same value as the *System name*. It is not updated on a *System name* change. The *Device object name* can only be changed from the BACnet client side.

The default name of a *Measurement object* is formed by combining the abbreviation for the object type and its instance ID, e.g., "AV–2", where *AV* stands for *Analog value*. The object name can be changed from the BACnet client side.

If a user adjusts the setting labeled *BACnet sensor measurement object type* in the GUI from *Analog Value* to *Analog Input*, the object names will be updated accordingly for those objects that still have their default names. For instance, "AV–2" would be updated to "AI-2" to reflect the change in object type from *Analog Value* to *Analog Input*.

Indications of alarm presence for measurement objects

Depending on the *alarm type* the property *Reliability* is updated for the *measurement object* to which it refers, or for all *measurement objects* of the corresponding sensor.

Alarm types	Alarm applies to	Reliability	Integer value
Threshold alarm	Measurement object	`UNRELIABLE_OTHER`	7
Battery alarm	Battery object	`UNRELIABLE_OTHER`	7
RSSI alarm	RSSI object	`COMMUNICATION_FAILURE`	12
Wrong channel alarm	All sensor objects	`CONFIGURATION_ERROR`	10
Error flag alarm	All sensor objects	`UNRELIABLE_OTHER`	7

Reliability property value is set to `NO_FAULT_DETECTED` (integer value: 0) if there is no alarm detected.

Units for Measurment object present value

The BACnet standard defines a set of units for representing the present value of a *Measurement object*. These units are used to specify the type and scale of the measured quantity.

Units property conversion table. All units which are not in this table do not have an appropriate BACnet unit integer. Units properties will be set to 95 (No units).

Bacnet-stack variable name	Unit	BACnet enumeration integer value
DEGREES_CELSIUS	C°	62
DEGREES_FAHRENHEIT	F°	64
DEGREES_KELVIN	K	63
PERCENT_RELATIVE_HUMIDITY	%	29
PARTS_PER_MILLION	ppm	96
AMPERES	A	3
VOLTS	V	5
SECONDS	s	73
OHMS	Ω	4
LUXES	lx	37
KILOGRAMS_PER_CUBIC_METER	kg/m^3	186
DECIBELS_MILLIVOLT	dBm	200
HECTOPASCALS	hPa	133
MILLIMETERS_OF_MERCURY	mmHg	59
BARS	bar	55
INCHES_OF_MERCURY	inHg	61
PASCALS	Pa	53
MILLIMETERS_OF_WATER	mmH_2O	206
MILLIBARS	mbar	134
METERS	m	31
CENTIMETERS	cm	118
MILLIMETERS	mm	30
MICROMETERS	µm	194
FEET	ft	33
INCHES	in	32
KILOGRAMS	kg	39
POUNDS_MASS	lb	40
NEWTON	N	153
METERS_PER_SECOND	m/s	74
KILOMETERS_PER_HOUR	km/h	75
MILES_PER_HOUR	mi/h	78
FEET_PER_SECOND	ft/s	76
PERCENT	%	98

(Table continued in next page)

(Table continued from previous page)

Bacnet-stack variable name	Unit	BACnet enumeration integer value
PARTS_PER_MILLION	ppm	96
NEWTON_METERS	Nm	160
METERS_PER_SECOND_PER_SECOND	m/s^2	166
JOULES	J	16
KILOJOULES	kJ	17
MEGAJOULES	MJ	126
BTUS	BTU	20
KILOWATT_HOURS	kWh	19
SIEMENS_PER_METER	S/m	174
WATTS	W	47
KILOWATTS	kW	48
MEGAWATTS	MW	49
HORSEPOWER	hp	51
CUBIC_METERS_PER_HOUR	m^3/h	135
LITERS_PER_HOUR	l/h	136
CUBIC_METERS_PER_MINUTE	m^3/min	165
LITERS_PER_MINUTE	l/min	88
CUBIC_METERS_PER_SECOND	m^3/s	85
LITERS_PER_SECOND	l/s	87
CUBIC_FEET_PER_SECOND	ft^3/s	142
CUBIC_METERS	m^3	80
LITERS	l	82
CUBIC_FEET	ft^3	79
US_GALLONS	gal	83
BECQUERELS	Bq	222
Custom Unit definition specific to Aranet PRO		
BECQUERELS_PER_CUBIC_METER	Bq/m^3	50001
PULSES	pulses	50002
FRACTION_OF_ONE	/	50003
MICROMOLE_PER_SECOND_PER_METER_SQUARED	$\mu mol/m^2s$	50004

For those measurements which have a string representation of its unit is included in the *Description* property.

Aranet PRO supported BACnet services

The following BACnet services are supported by the Aranet PRO base station integration:

- *Subscribe CoV* (execute)
- *Read Property* (execute)

- *Write Property* (execute)
- *Who-has* (execute)
- *Who-is* (execute)

When a device needs to locate another device offering a specific service or object, it broadcasts a *Who-Has* message on the network. This message includes details such as the *object type*, *instance number*, and *property* being sought. All devices on the network receive this broadcast. Devices capable of providing the requested service or object respond with an *I-Have* message, containing the network address of the device offering the desired service or object.

Similarly, a *Who-Is* broadcast is used to obtain network addresses of devices on the network. This is crucial for communication between devices without broadcasting to the entire system. A device seeking another device's address sends a message specifying a *Device Object Instance Number* or a range of *Instance Numbers*, like "Who-Is device 3001" or "Who-Is device 3000 to 3099".

Devices responding to the *Who-Is* message send an *I-Am* message either locally, remotely, or across the entire network. This allows devices needing information about the responders to acquire address details without generating additional network traffic. Additionally, it helps the responding device determine its route within the network for efficient communication.

Engineering Data Exchange file structure

The *EDE file* serves to integrate the Aranet PRO BACnet service into third-party systems through offline import. This file defines the structure of BACnet project data for a device. When changes are made to the *object list* in the Aranet PRO base, the outdated *EDE file*, which contains information about available objects, is removed. Users can then request the base to generate a new *EDE file*, which is in CSV format. The first six rows of the file provide descriptions about the BACnet project that generated it.

The *EDE file* structural components is in the table below.

Column name	Requirement	Description
Keyname	Mandatory	BACnet specific object type combined with instance identifier
Device object instance	Mandatory	Device object instance ID which owns this object
Object name	Mandatory	Name of the object
Object type	Mandatory	BACnet specific identifier used to identify object type
Object instance	Mandatory	Object instance identifier
Description	Optional	Object description auto-generated by BACnet upon object creation
State text reference	Optional	Not used
Present value default	Optional	Not used
Supports CoV	Optional	Does object supports Cange-of-Value
Commandable	Optional	Is commandable (using priority array)
Min. present value	Optional	Not used
Max. present value	Optional	Not used
High limit	Optional	Not used
Low limit	Optional	Not used

Unit-code	Optional	BACnet specific identifier used to identify the measurement unit
Vendor specific address	Optional	Not used
Element	Optional	Additional field which contains object description