

BACnet® Server for AcquiSuite®

by

Mark N. Shepard



BACnet is a registered trademark of ASHRAE. ASHRAE does not endorse, approve or test products for compliance with ASHRAE standards. Compliance of listed products to the requirements of ASHRAE Standard 135 is the responsibility of BACnet International (BI). BTL is a registered trademark of BI.

Applies to: **Obvius_BACnet**, v02.13.0312_01 or later

Document revised: **5/10/2013**

Copyright Information

Copyright © 2009, 2010, 2011,2012,2013 by Obvius

Obvius and **AcquiSuite** are trademarks of Obvius Holdings LLC.

BACnet® is a registered trademark of American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE). For more information see the <http://www.bacnet.org/Notice>.

Other brand and product names are trademarks or registered trademarks of their respective holders.

U.S. Government Restricted Rights: Use, duplication or disclosure by the Government is subject to restrictions set fourth in subparagraph (a) through (d) of the Commercial Computer Restricted Rights clause at FAR 52.227-19 when applicable, or subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013, and in similar clauses in the NASA FAR Supplement.

Limited Warranty

OBVIUS IS PROVIDING THIS WARRANTY IN LIEU OF ALL OTHER EXPRESS OR IMPLIED WARRANTIES, INCLUDING ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. THIS WARRANTY IS BUYER'S EXCLUSIVE REMEDY FOR ALL CLAIMS AGAINST OBVIUS. OBVIUS SHALL NOT BE LIABLE FOR ANY CONSEQUENTIAL OR INCIDENTAL DAMAGES. OBVIUS'S TOTAL LIABILITY FOR ALL CLAIMS SHALL BE LIMITED TO THE PRICE PAID FOR ITS PRODUCT.

Obvius promises buyer that any standard product manufactured by Obvius shall be free from all material defects in design, material, or manufacturing for a period of 2 years from the manufacture date; provided, however, that the warranty shall not extend to ordinary wear and tear or to normally replaceable components (e.g., batteries). During the warranty period, Obvius may repair or replace (in its sole discretion) any product suffering from a warranty defect and returned freight prepaid by buyer, with no charge to buyer for any warranty repair or replacement. The warranty shall remain in full force and effect for such 2 year period, provided that the product: (1) was installed, operated, and maintained properly; (2) has not been abused or misused; (3) has not been repaired, altered, or modified outside of Obvius's authorized facilities; (4) has not been sold subject to other warranty terms specified at the time of sale; and (5) is still owned by the original purchaser. This warranty provides specific legal rights that may be varied by state law. Obvius's products are not designed for life or safety applications.

Product Application Limitation

Obvius products are not intended for use in critical applications such as nuclear facilities, human implantable devices or life support.

Obvius is not liable, in whole or in part, for any claims or damages arising from such uses.

Obvius strongly believes in continuous improvement, therefore we must reserve the right to change specifications and product offerings without notice. Where possible, we will substitute products with equivalent functionality when necessary.

Obvius
3300 NW 211th Terrace
Hillsboro, OR 97124
ph: 503-601-2099

Table of Contents

1	OVERVIEW	7
1.1	FEATURES.....	7
1.2	PICS - PROTOCOL IMPLEMENTATION CONFORMANCE STATEMENT.....	8
2	PLANNING YOUR DEPLOYMENT	11
2.1	SOFTWARE INSTALLATION REQUIREMENTS	11
2.2	NUMBER OF ACQUISUITES REQUIRED.....	11
2.2.1	Case Study: Meters Connected via RS485 at 9600 Baud	13
2.2.2	Case Study: Meters Connected via Modhopper, 1-Hop Maximum.....	14
2.2.3	Case Study: Meters Connected via RS485 at 38k4 and 115k2 Baud	15
2.2.4	Case Study: "Consumption Only" Option at 9600 Baud	17
2.3	MAXIMUM BACNET CLIENTS	17
2.4	MINIMUM APDU_TIMEOUT.....	18
2.5	SUPPORTED RELAY TYPES	18
2.6	FAIL SAFE LOGIC DESIGN	18
3	INSTALLATION	19
3.1	OVER-THE-WEB INSTALLATION	19
3.2	"HAVE DISK" INSTALLATION.....	19
3.3	UN-INSTALLATION	20
4	SETUP	21
4.1	AUTOMATIC QUICK START.....	21
4.2	GUIDED SETUP.....	21
4.3	BACNET SETUP OPTIONS.....	23
4.4	HANDLING UNSUPPORTED METERS WITH THE MODBUS FRAMEWORK.....	25
5	BACNET DEVICE LIST	26
5.1	TYPICAL USES OF THE DEVICE LIST PAGE	26
6	BACNET DISCOVER	27

6.1	TYPICAL USES OF THE DISCOVER PAGE	28
6.2	BACNET DISCOVER → SHOW: [NETWORK], [FIRMWARE], [STATS]	28
6.2.1	show: [basic].....	28
6.2.2	show: [network].....	28
6.2.3	show: [firmware]	29
6.2.4	show: [stats].....	29
6.2.5	show: [all].....	30
6.3	BACNET DISCOVER → OBJECTS.....	30
6.3.1	Typical Uses of <i>Discover</i> → <i>Objects</i> link	31
6.4	BACNET DISCOVER → OBJECTS → EPICS.....	32
6.4.1	Typical Uses of the <i>EPICS</i>	32
7	HOW DO I.....	33
7.1	HOW DO I... CHOOSE DEVICE IDs AND NETWORK NUMBERS?	33
7.1.1	Background: Permanent Names	33
7.1.2	Background: Detecting Name and ID Conflicts.....	33
7.2	HOW DO I... CHOOSE TIMEOUTS?	33
7.3	HOW DO I... OPTIMIZE RESPONSE TIME?.....	34
7.4	HOW DO I... OPTIMIZE POLLING?.....	35
7.4.1	Background: Caching Behavior.....	35
7.5	HOW DO I... FIX UNIT ISSUES?.....	35
7.6	HOW DO I... USE RELAY OUTPUTS?	36
7.7	HOW DO I... SPAN SUBNETS WITH A BBMD?	36
7.8	HOW DO I... SPAN SUBNETS WITH FOREIGN DEVICE REGISTRATION?	38
7.8.1	Typical Uses of Foreign Device Registration:	38
8	TROUBLESHOOTING	39
8.1	SYMPTOM: NOT BTL LISTED?	39
8.2	SYMPTOM: CAN'T "DISCOVER" THE ACQUISUITE?.....	39
8.3	SYMPTOM: "DISCOVERY IS BROKEN OR NOT WORKING" OR "MY CONTROLLER DOESN'T SUPPORT DISCOVERY"?.....	41

8.4	SYMPTOM: BACNET TIMEOUT OR COMMUNICATION ERRORS?	42
8.5	SYMPTOM: DATA VALUES ARE NOT UPDATING (NOT REFRESHING)?	42
8.6	SYMPTOM: CAN'T WRITE (TO RELAYS)?	43
8.7	SYMPTOM: RELAYS TURN ON WHEN THEY SHOULDN'T?	46
8.7.1	Setting a "Stale Timeout"	46
8.7.2	How the AcquiSuite Signals Communications Failures	47
9	BACNET BACKGROUND	48
9.1	BACKGROUND ON "DEVICE OBJECTS"	48
9.2	BACKGROUND ON "ANALOG INPUT OBJECTS"	49
9.3	BACKGROUND ON "BINARY VALUE OBJECTS"	49
9.4	TERMINOLOGY: PICS, EPICS AND BIBBs	49
10	COMMON WIRESHARK FILTERS	50

1 Overview

1.1 Features

- BACnet/IP server which automatically shares any meter supported by the AcquiSuite platform and connected via **Modbus/485**, **Modbus/TCP** or **Modhopper®** wireless mesh network.
- Supports devices connected via the appropriate AcquiSuite “Modbus-to-Native” Gateway Modules, such as **Fronius**, **SMA** and **Refusol** PV-inverters.
- Conformance tested and accepted by [BACnet Testing Laboratories](#) as of March 2013.
- Adds BACnet/IP to the A7810 AcquiLite and A8814 AcquiSuite+.
- Integrates with [AcquiSuite Modbus Framework](#), allowing user-definition of new or unusual meters.
- [Automatic quick start](#): Simply connect a supported Modbus meter and it will be automatically detected and shared via BACnet.
- [Fast, guided setup](#): Server identifies problems, guides you to solutions.
- Engineering units: Modbus registers are automatically translated to engineering quantities with proper units.
- Caching to manage limited Modbus bandwidth. Cache times are automatically tuned based on the units of the data point and history of queries.
- Relay outputs supported, allowing complete control loops. See [supported relay types](#) for details.
- Includes BBMD server, allowing your BACnet network to span IP subnets without extra hardware.
- [BACnet discovery tools built-in](#), including detection of Device ID and Network Number conflicts.
- [Detailed statistics](#) to monitor performance.
- Allows incoming Foreign Device Registration (FDR), simplifying remote diagnostics.
- Compatible with AcquiSuite Remote Configuration.
- Supported on Obvius A8812 (AcquiSuite), A8810 (AcquiSuite EMB), A7810 (AcquiLite) and A8814 (AcquiSuite+).
- Supports BACnet protocol version 1, revision 12.
- 64 BBMD and 64 FD entries are allowed.

1.2 PICS - Protocol Implementation Conformance Statement

BACnet Vendor Name:	Obvius
BACnet Vendor ID:	556
Product Name:	AcquiSuite, AcquiLite
Product Model Number:	A8812, A8810, A7810, A8814
Product Description:	"BACnet Server for AcquiSuite" is an add-on software module for Obvius' "AcquiSuite" data acquisition server platform. The "AcquiSuite" is a Linux- based embedded controller with a web user-interface which automatically detects and recognizes a variety of power meters and sensors using a variety of protocols, chiefly Modbus/RTU and Modbus/TCP but also proprietary protocols. The "AcquiSuite" logs, stores and uploads data from these meters or sensors to a variety of databases, and may also act as a real-time protocol gateway.
	The "BACnet Server for AcquiSuite" allows the AcquiSuite to represent each Modbus/TCP or Modbus/RTU meter or sensor it recognizes as a BACnet device, with data points represented as the appropriate BACnet objects. BACnet queries are answered from cached data, and proprietary algorithms are used to decide when to refresh this cached data via the backend (e.g., Modbus) protocols.
	The "BACnet Server for AcquiSuite" also includes features in its web user- interface to scan for and display information about other BACnet devices on the network, intended for to help the user through initial configuration and deployment of the product and in diagnosing problems.
	Only BACnet/IP (Annex J) is supported at this time.

1.1.1. BACnet Standardized Device Profile (Annex L)

X	BACnet Smart Sensor (B-SS).
X	BACnet Smart Actuator (B-SA).

1.1.2. BACnet Interoperability Building Blocks (BIBBs) Supported (Annex K)

X	K.1.1 BIBB - Data Sharing - ReadProperty-A (DS-RP-A) [for network diagnostics]
X	K.1.2 BIBB - Data Sharing - ReadProperty-B (DS-RP-B)
X	K.1.3 BIBB - Data Sharing - ReadPropertyMultiple-A (DS-RPM-A) [for network diagnostics]
X	K.1.4 BIBB - Data Sharing - ReadPropertyMultiple-B (DS-RPM-B)
X	K.1.8 BIBB - Data Sharing - WriteProperty-B (DS-WP-B)
X	K.5.1 BIBB - Device Management - Dynamic Device Binding-A (DM-DDB-A)
X	K.5.2 BIBB - Device Management - Dynamic Device Binding-B (DM-DDB-B)
X	K.5.4 BIBB - Device Management - Dynamic Object Binding-B (DM-DOB-B)

1.1.3. Segmentation Capability:

None.

1.1.4. Standard Object Types Supported:

X	Device.
X	Analog Input.
X	Binary Value.

1.1.5. Unsupported Properties and Restrictions:

None.

1.1.6. Data Link Layer Options:

X	BACnet IP, (Annex J).
---	-----------------------

1.1.7. Device Address Binding:

Not supported.

1.1.8. Networking Options:

X	BACnet/IP Broadcast Management Device (BBMD). Max BDT slots: 64.
X	Registrations by Foreign Devices. Max FDT slots: 64.

1.1.9. Character Sets Supported:

Where support for multiple character sets is indicated, this does not imply that they can all be supported simultaneously.

X	ANSI X3.4/UTF-8
---	-----------------

1.1.10. Unsupported Functions of the BACnet Driver:

None.	
-------	--

1.1.11. Non-BACnet Equipment/Networks Supported:

If this product is a communication gateway, describe the types of non-BACnet equipment/networks(s) that the gateway supports

Various backend protocols, such as Modbus/TCP, Modbus/485, USB, Fronius, SMA, etc.	
--	--

2 Planning Your Deployment

The guidelines in this section will help you plan your deployment, determine equipment required and estimate performance before you begin your installation.

2.1 Software Installation Requirements

- AcquiSuite firmware version v02.13.0211 or later.
- One of the following models:
 - A8812 (AcquiSuite),
 - A8810 (AcquiSuite EMB),
 - A7810 (AcquiLite), or
 - A8814 (AcquiSuite+).
- An Internet connection to www.buildingmanageronline.com if over-the-web install is to be used.
- 1MB of free flash memory on the AcquiSuite. The BACnet module is approximately 750kB.
- Other BACnet/IP devices on the LAN to which the AcquiSuite is connected.

2.2 Number of AcquiSuites Required

In general, Obvius recommends a maximum of 32 meters per AcquiSuite, but this number may need to be reduced depending on several factors.

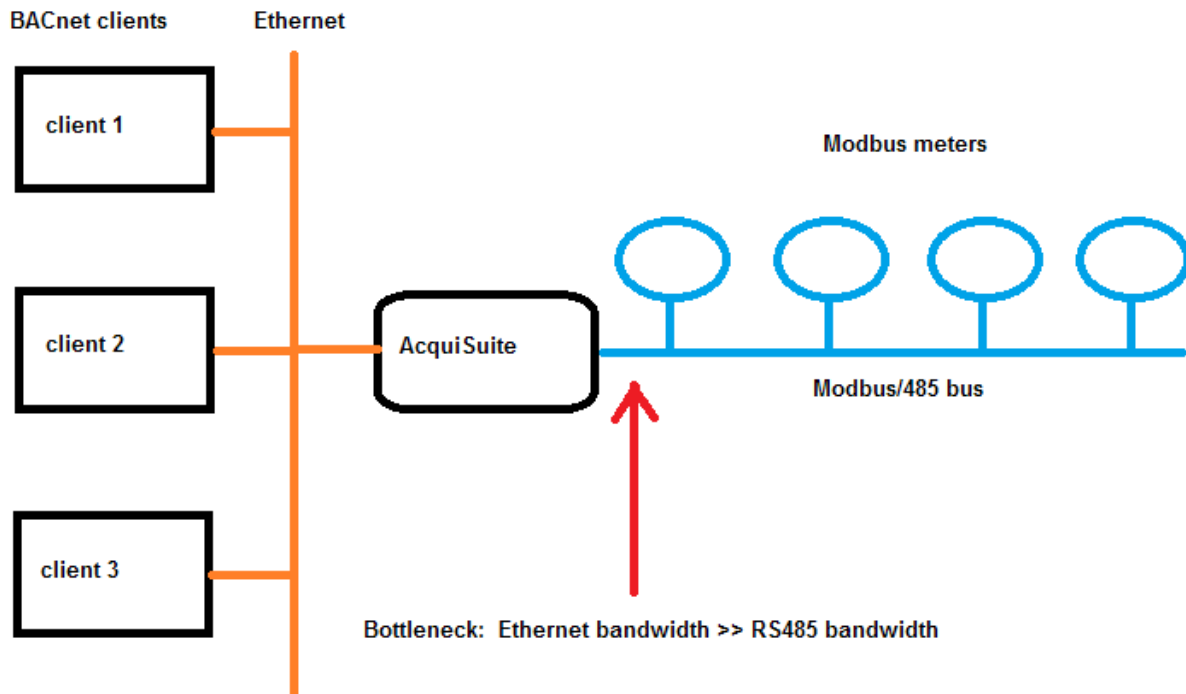
The number of AcquiSuites required to service a collection of meters depends on:

- the **desired data-refresh interval** or logging period,
- whether or not the AcquiSuites are also being asked to **control relays** (control-loops typically work better with lower latency, and some applications may not tolerate high latency),
- whether “**consumption only**” **logging is available and acceptable** (“consumption only” is an option available on some meters which reduces data logged by the AcquiSuite to just the “energy” or kWh points).

These factors in turn depend on:

- the meter’s maximum RS485 baud rate,
- the number of points per meter,
- the method of connection (RS485 or Modhopper; Modhoppers add latency),

Generally speaking, the performance bottleneck is the connection between the AcquiSuite and the Modbus meters – either Modbus/485 or Modhopper wireless mesh network.



This means the best **data-refresh interval** or logging period can be estimated from the RS485 baud rate, the meter’s response time, the amount of data being queried from the meter, and knowledge of how many “Modbus transactions” are needed to query all the meter’s points.

$$refresh_interval \approx num_meters * \left[\frac{(data_bytes_per_meter * 10 \text{ bits/Byte})}{RS485_baud_rate} + (meter_response_time * blocks_per_refresh) \right]$$

Note: if the meter is connected via Modhopper(s), “meter_response_time” should include 500 ms per “hop”.

The following case studies give the results of this calculation for several common situations.

Please see [Optimizing Polling](#) and [Optimizing Response Time](#) for a full discussion of factors affecting update rates.

The BACnet Server for AcquiSuite continually measures the average and worst-case response time of each backend meter; these statistics are displayed on the [BACnet → Device List → show: \[stats\] tab](#) and are useful for system performance tuning.

2.2.1 Case Study: Meters Connected via RS485 at 9600 Baud

The table below shows the best **data refresh interval** (in seconds) you can expect using Modbus/485 at 9600 baud for a variety of common meters.

The **green region** are the configurations recommended by Obvius.

The **gray (middle) region** is only relevant if you are attempting to control Modbus relays as well as reading meters. This region is where the Modbus/485 response time has exceeded the default BACnet APDU_Timeout of 6 seconds (minus 1). As a result, any attempt to write a relay will time-out. Increasing the APDU_Timeout will “solve” this problem, though at the expense of making any control-loops less responsive and more difficult to debug.

The **red region** is where the delay has become so great that the BACnet Server will always report the meter as being in a failure state.

Best Data Refresh Interval if using Modbus/485 at 9600 baud (Meter response time <200ms)			Modbus Baudrate = 9600 baud					
			Num. of Modhopper Hops = 0 500ms/hop					
			Log "consumption only"? 0 1=yes, 0=no					
			BACnet APDU Timeout = 6 sec					
Meter Type	Max Baud	Number of Points per Meter	Number of Meters per AcquiSuite					
			1	4	8	16	32	
PowerLogic Enercept Basic	9600		2	0.08	0.3	0.6	1.3	2.5
Veris H8035	9600		2	0.08	0.3	0.6	1.3	2.5
Veris H8036	9600		26	0.18	0.7	1.4	2.9	5.7
Fronius IG+ inverter	N/A		41	0.22	0.9	1.8	3.6	7.1
Obvius A8332 FlexIO	19200		42	0.25	1.0	2.0	4.1	8.1
Obvius A90DC SCM	9600		37	0.19	0.8	1.5	3.1	6.2
Veris H8238, 2 ch.	19200		56	0.59	2.3	4.7	9.4	18.7
Power Measurement ION 6200	19200		61	0.45	1.8	3.6	7.1	14.3
Veris H8238, 4 ch.	19200		112	1.16	4.7	9.3	18.6	37.2
Electro Industries Shark 100	57600		116	0.86	3.4	6.8	13.7	27.4
Eaton IQ 260	57600		119	0.87	3.5	6.9	13.9	27.8
Benjamin Electric PP+, 12	115200		218	1.14	4.5	9.1	18.2	36.4
Veris H8238, 8 ch.	19200		224	2.32	9.3	18.5	37.1	74.2
Veris E30A	38400	374	1.83	7.3	14.6	29.2	58.5	
Benjamin Electric PP+, 42	115200	668	3.36	13.5	26.9	53.8	107.6	
key =			ok, recommended					
			ok if reading / unworkable if writing to relays (increase APDU_Timeout as needed)					
			unworkable					

2.2.2 Case Study: Meters Connected via Modhopper, 1-Hop Maximum

The table below shows the effect of connecting meters via a Modhopper wireless mesh network. The increased latency due to packetizing data for RF transmission has slowed data refresh.

If more than two Modhoppers are being used, the AcquiSuite should be arranged in the center of the Modhopper network so that each meter can be reached by at most 1 "hop." Each "hop" adds 500 ms of latency. The total delay latency may be several times that for meters with many points (such as the Veris E30A or Benjamin Electric PP+) as these meters require several modbus round-trips for each update.

Best Data Refresh Interval if using Modhopper, 9600, 1-hop (Modbus timeout = 500ms)			Modbus Baudrate = 9600 baud				
			Num. of Modhopper Hops = 1 500ms/hop				
			Log "consumption only"? 0 1=yes, 0=no				
			BACnet APDU Timeout = 6 sec				
Meter Type	Max Baud	Number of Points per Meter	Number of Meters per AcquiSuite				
			1	4	8	16	32
PowerLogic Enercept Basic	9600	2	0.60	2.4	4.8	9.5	19.1
Veris H8035	9600	2	0.60	2.4	4.8	9.5	19.1
Veris H8036	9600	26	0.80	3.2	6.4	12.7	25.5
Fronius IG+ inverter	N/A	41	0.90	3.6	7.2	14.4	28.9
Obvius A8332 FlexIO	19200	42	0.86	3.5	6.9	13.8	27.7
Obvius A90DC SCM	9600	37	0.81	3.3	6.5	13.0	26.0
Veris H8238, 2 ch.	19200	56	1.33	5.3	10.6	21.2	42.5
Power Measurement ION 6200	19200	61	1.21	4.8	9.7	19.3	38.7
Veris H8238, 4 ch.	19200	112	2.14	8.6	17.1	34.2	68.4
Electro Industries Shark 100	57600	116	1.85	7.4	14.8	29.6	59.1
Eaton IQ 260	57600	119	1.87	7.5	15.0	30.0	59.9
Benjamin Electric PP+, 12	115200	218	2.55	10.2	20.4	40.9	81.7
Veris H8238, 8 ch.	19200	224	3.76	15.0	30.1	60.1	120.3
Veris E30A	38400	374	3.89	15.6	31.2	62.3	124.6
Benjamin Electric PP+, 42	115200	668	6.66	26.6	53.2	106.5	213.0
key =			ok, recommended				
			ok if reading / unworkable if writing to relays (increase APDU_Timeout as needed)				
			unworkable				

2.2.3 Case Study: Meters Connected via RS485 at 38k4 and 115k2 Baud

The following two tables show the effects of increasing the baud rate to 38,400 and 115,200 bits/sec. respectively.

Note that many meters cannot operate at these baud rates – these are **crossed-out in red**.

This configuration is mainly of with devices with hundreds of points, such as the Veris E30A BCM and larger Benjamin Electric panels.

RS485 at 38,400 baud – only 5 of the example meters support this baud rate:

Best Data Refresh Interval if using Modbus/485 at 38,400 baud (Meter response time <200ms)			Modbus Baudrate = 38400 baud					
			Num. of Modhopper Hops = 0 500ms/hop					
			Log "consumption only"? 0 1=yes, 0=no					
			BACnet APDU Timeout = 6 sec					
Meter Type	Max Baud	Number of Points per Meter	Number of Meters per AcquiSuite					
			1	4	8	16	32	
PowerLogic Enercept Basic	9600		2	0.06	0.2	0.5	0.9	1.9
Veris H8035	9600		2	0.06	0.2	0.5	0.9	1.9
Veris H8036	9600		26	0.08	0.3	0.7	1.3	2.7
Fronius IG+ inverter	N/A		41	0.07	0.3	0.6	1.2	2.3
Obvius A8332 FlexIO	49200		42	0.16	0.6	1.2	2.5	5.0
Obvius A90DC-SCM	9600		37	0.09	0.4	0.8	1.5	3.0
Veris H8238, 2 ch.	49200		56	0.39	1.6	3.1	6.2	12.4
Power Measurement ION-6200	49200		61	0.23	0.9	1.9	3.7	7.5
Veris H8238, 4 ch.	49200		112	0.77	3.1	6.2	12.4	24.8
Electro Industries Shark 100	57600		116	0.43	1.7	3.4	6.9	13.8
Eaton IQ 260	57600		119	0.43	1.7	3.5	6.9	13.9
Benjamin Electric PP+, 12	115200		218	0.41	1.6	3.3	6.5	13.1
Veris H8238, 8 ch.	49200		224	1.55	6.2	12.4	24.8	49.5
Veris E30A	38400	374	0.58	2.3	4.6	9.3	18.6	
Benjamin Electric PP+, 42	115200	668	1.17	4.7	9.3	18.7	37.3	
key =		ok, recommended						
		ok if reading / unworkable if writing to relays (increase APDU_Timeout as needed)						
		unworkable						

RS485 at 115,200 baud -- note that only 2 of the example meters support this baud rate:

Best Data Refresh Interval if using Modbus/485 at 115k2 baud (Meter response time <200ms)			Modbus Baudrate = 115200		baud			
			Num. of Modhopper Hops = 0		500ms/hop			
			Log "consumption only"? 0		1=yes, 0=no			
			BACnet APDU Timeout = 6		sec			
Meter Type	Max Baud	Number of Points per Meter	Number of Meters per AcquiSuite					
			1	4	8	16	32	
PowerLogic Enercept-Basic	9600		2	0.05	0.2	0.4	0.8	1.6
Veris H8035	9600		2	0.05	0.2	0.4	0.8	1.6
Veris H8036	9600		26	0.06	0.2	0.5	1.0	1.9
Fronius IG+ inverter	N/A		41	0.03	0.1	0.3	0.6	1.1
Obvius A8332-FlexIO	49200		42	0.13	0.5	1.0	2.1	4.1
Obvius A90DC-SCM	9600		37	0.07	0.3	0.6	1.1	2.2
Veris H8238, 2 ch.	49200		56	0.34	1.4	2.7	5.4	10.9
Power-Measurement ION-6200	49200		61	0.18	0.7	1.5	2.9	5.8
Veris H8238, 4 ch.	49200		112	0.68	2.7	5.4	10.9	21.7
Electro-Industries Shark-100	57600		116	0.32	1.3	2.6	5.1	10.3
Eaton IQ-260	57600		119	0.32	1.3	2.6	5.2	10.3
Benjamin Electric PP+, 12	115200		218	0.24	0.9	1.9	3.8	7.6
Veris H8238, 8 ch.	49200		224	1.36	5.4	10.9	21.7	43.5
Veris E30A	38400	374	0.28	1.1	2.3	4.5	9.1	
Benjamin Electric PP+, 42	115200	668	0.65	2.6	5.2	10.4	20.8	
		key = ok, recommended						
		ok if reading / unworkable if writing to relays (increase APDU_Timeout as needed)						
		unworkable						

2.2.4 Case Study: “Consumption Only” Option at 9600 Baud

If your application requires only “energy” (kWh) data, the AcquiSuite’s “consumption only” option is worth considering.

In this case study, the AcquiSuite’s “consumption only” option has been turned on for those meters which support the option (compare this table to [the above](#)).

This causes the AcquiSuite to only log the “energy” points (kWh) rather than all points, reducing the data to be logged and allowing a quicker data refresh interval.

Contact Obvius for details about whether a particular meter supports “consumption only” mode.

Best Data Refresh Interval if using "Consumption Only" @ 9600 (Meter response time < 200ms)			Modbus Baudrate =		9600	baud		
			Num. of Modhopper Hops =		0	500ms/hop		
			Log "consumption only"?		1	1=yes, 0=no		
			BACnet APDU Timeout =		6	sec		
Meter Type	Max Baud	Number of Points per Meter	Number of Meters per AcquiSuite					
			1	4	8	16	32	
PowerLogic Enercept Basic	9600		2	0.08	0.3	0.6	1.3	2.5
Veris H8035	9600		2	0.08	0.3	0.6	1.3	2.5
Veris H8036	9600		26	0.18	0.7	1.4	2.9	5.7
Fronius IG+ inverter	N/A		3	0.04	0.2	0.3	0.7	1.4
Obvius A8332 FlexIO	19200		42	0.25	1.0	2.0	4.1	8.1
Obvius A90DC SCM	9600		37	0.19	0.8	1.5	3.1	6.2
Veris H8238, 2 ch.	19200		56	0.59	2.3	4.7	9.4	18.7
Power Measurement ION 6200	19200		3	0.11	0.5	0.9	1.8	3.6
Veris H8238, 4 ch.	19200		112	1.16	4.7	9.3	18.6	37.2
Electro Industries Shark 100	57600		8	0.09	0.4	0.7	1.5	3.0
Eaton IQ 260	57600		8	0.09	0.4	0.7	1.5	3.0
Benjamin Electric PP+, 12	115200		218	1.14	4.5	9.1	18.2	36.4
Veris H8238, 8 ch.	19200		224	2.32	9.3	18.5	37.1	74.2
Veris E30A	38400	374	1.83	7.3	14.6	29.2	58.5	
Benjamin Electric PP+, 42	115200	668	3.36	13.5	26.9	53.8	107.6	
key =			ok, recommended					
			ok if reading / unworkable if writing to relays (increase APDU_Timeout as needed)					
			unworkable					

2.3 Maximum BACnet Clients

Obvius recommends a maximum of **128 simultaneous BACnet clients**. “Reads” are satisfied from cache. The achievable **data refresh interval** is typically limited by the AcquiSuite’s Modbus RS485 bandwidth, number and type of meters.

If writing (to relays) via BACnet, Obvius recommends a maximum of **8 simultaneous “writers.”** Writes are not cached; each write will result in a Modbus write to the relay device.

2.4 Minimum APDU_Timeout

Obvius recommends BACnet clients use a minimum APDU_Timeout of **6 seconds**.

If writing to relays, BACnet clients should use an APDU_Timeout which is greater than the actual **data refresh interval**, which depends on the system configuration. This may be estimated from the above tables or measured using [BACnet → Device List → show: \[stats\] tab](#).

2.5 Supported Relay Types

The BACnet Server for AcquiSuite supports the following Modbus relay devices.

- Obvius, A8812 Internal I/O,
- Obvius, A8332-8F2D Flex I/O
- Advantec, Adam 4068 and 4069
- Acromag, 902MB
- Coyote DataComm, Rio-9F
- Benjamin Electric, Power Panel Plus

2.6 Fail Safe Logic Design

If your application involves controlling relays, please consider **fail safe logic design**. Please see [Symptom: Relays Turn On When They Shouldn't](#) for a brief discussion of the issues.

3 Installation

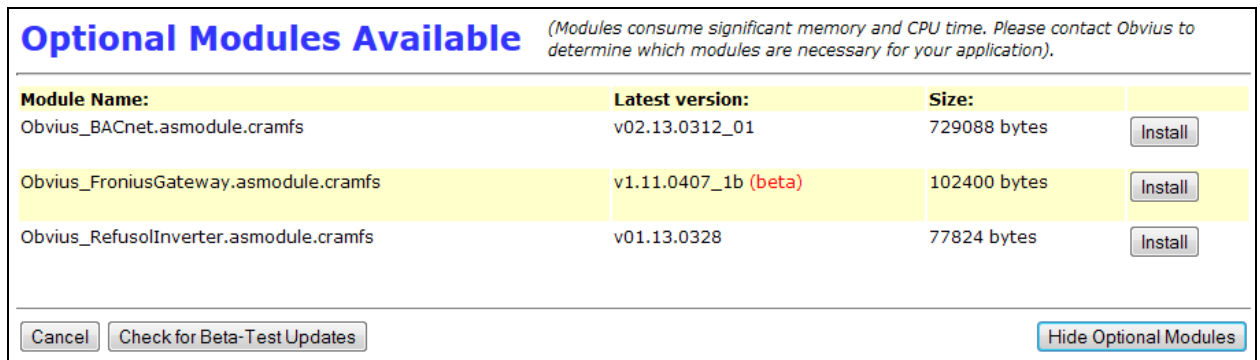
The “BACnet Server for AcquiSuite” module may be installed on any AcquiSuite or AcquiLite meeting the requirements below. There are no software licensing restrictions.

The module may be installed either over the web or uploading the module’s file to the AcquiSuite.

3.1 Over-the-web Installation

To install over the web,

1. Login to the AcquiSuite as “admin” and navigate to *System* → *Firmware Versions*.
2. Click “Check for Updates Now”.
3. Click “Show Optional Modules” (right-hand side; see screenshot):



Module Name:	Latest version:	Size:	
Obvius_BACnet.asmodule.cramfs	v02.13.0312_01	729088 bytes	<input type="button" value="Install"/>
Obvius_FroniusGateway.asmodule.cramfs	v1.11.0407_1b (beta)	102400 bytes	<input type="button" value="Install"/>
Obvius_RefusolInverter.asmodule.cramfs	v01.13.0328	77824 bytes	<input type="button" value="Install"/>

Buttons at the bottom:

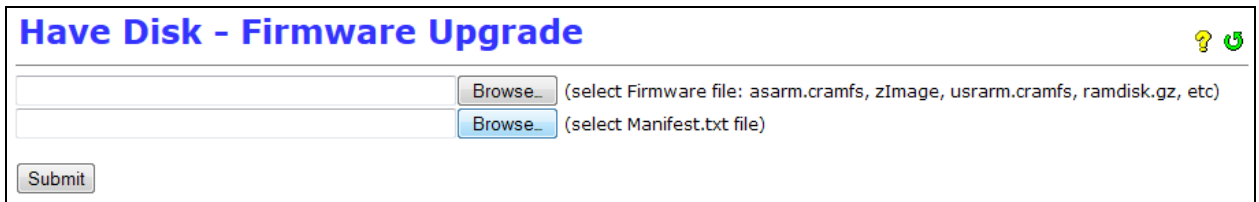
4. Find the “Obvius_BACnet.asmodule.cramfs” file and click “Install” next to it.
5. Once the file has downloaded successfully, click “Install and Reboot”.

3.2 “Have Disk” Installation

This installation method is used if your AcquiSuite does not have an working Internet connection to www.buildingmanageronline.com.

1. Contact Obvius Tech Support and request the .zip file for the “Obvius BACnet Server for AcquiSuite”. The file’s name should be “Obvius_BACnet_v02.13.0312_01.zip” or similar.
2. Unpack the .zip file. It should contain (among other files), “Obvius_BACnet.asmodule.cramfs” and “Manifest.txt”.
3. Login to the AcquiSuite as “admin”, and navigate to *System* → *Firmware Versions*.
4. Click “Have Disk”.
5. Click each Browse... button, selecting the “Obvius_BACnet.asmodule.cramfs” file and then the “Manifest.txt” file. Note that some browsers will allow you to drag-and-drop the files onto the input

fields.



Have Disk - Firmware Upgrade ? ↻

(select Firmware file: asarm.cramfs, zImage, usrarm.cramfs, ramdisk.gz, etc)

(select Manifest.txt file)

6. Click “Submit” to upload these two files.
7. Follow the instructions to reboot and complete the installation.

3.3 *Un-installation*

To remove the BACnet Server from an AcquiSuite,

1. Browse to System → Firmware Versions,
2. Find the “BACnet Server for AcquiSuite” module in the list of installed modules,
3. Set its **Action (at next boot)** to “Uninstall & Delete”.
4. Click Apply.
5. Reboot when prompted.

The BACnet Server will leave several configuration settings in the AcquiSuite’s loggerconfig.ini file. All BACnet related settings are prefixed with “BACNET...”. These should be harmless if left in loggerconfig.ini. They may be removed using a text editor such as “vi”.

4 Setup

The BACnet Server for AcquiSuite is intended to be extremely easy and quick to set up.

Note: All BACnet → Setup options take effect immediately when Apply is clicked, without requiring a reboot. Any changes to the AcquiSuite's meters, such as adding a new meter or changing the units of a point, are immediately reflected in the BACnet Server.

4.1 Automatic Quick Start

The fastest way to get started is to let the AcquiSuite handle everything automatically.

The AcquiSuite will automatically detect and identify meters attached via Modbus/485, will automatically configure BACnet options and will share all detected meters via BACnet.

To begin,

1. Set each Modbus/485 meter for 9600 baud, 8N1.
2. Connect each meter to the AcquiSuite's RS485 port.
3. If the BACnet Server is yet not installed, [install it](#).
4. Connect the AcquiSuite to your LAN, and set it for DHCP via its LCD menu.
5. Wait a few minutes for all Modbus/485 meters to be detected and to be shared via BACnet.

You can monitor the auto-detection by logging into the AcquiSuite's web UI as "admin", and navigating to *Modbus → Device List*.






You can see the BACnet objects the AcquiSuite is sharing by navigating to *BACnet → Device List*.

4.2 Guided Setup

Once you're up and running, you may use the *BACnet → Setup* page to fine-tune the server's configuration, for instance, to change the default BACnet device name or device ID.

The BACnet Server continually monitors the configuration of the AcquiSuite and will prompt you to fix any issues. These prompts are shown on the *BACnet → Setup* page, like so:

BACnet Server Status

Server status:	 OK
Device status:	 OK: 3 meters shared via BACnet
Network status:	 OK: Incoming FDR allowed
Config consistency:	 OK: Using recommended timeout [6 seconds]
Units & scaling status:	 OK: all meters' units converted to BACnet

The BACnet Server will detect and warn about the following common problems:

1. **Backend meter errors.** Any Modbus errors (or meters which simply aren't responding) will significantly slow BACnet data refresh.
2. **Incompatible options.** The AcquiSuite includes options to automatically search for Modbus devices, and to automatically scan multiple combinations of baud-rate, parity and stop-bits. These options are convenient but they can significantly slow BACnet data refresh.
3. **Inconsistent options.** By default the BACnet server will choose an APDU Timeout based on the AcquiSuite's backend Modbus/485 settings. Should you choose to manually override the APDU Timeout, or if the Modbus timeout is later changed, the BACnet Server will warn of any possible conflicts between these options, and recommend an APDU timeout.
4. **Unit-mapping issues.** The BACnet Server automatically maps the engineering-units of points from Modbus to BACnet, but in some cases, such as universal-input devices like the Obvius A8332-8F2D Flex I/O, where the user is allowed to enter the units of a point and has entered a non-standard unit abbreviation, the BACnet Server will warn that it couldn't map the units and show a list of all affected points so that you may make corrections.
5. **BACnet object names.** By default, the BACnet server selects a BACnet name for itself, derived from its IP-address. This name may change if the AcquiSuite is configured to use DHCP, so the AcquiSuite will prompt you to either assign it a static IP-address or choose a permanent name, so that your control-logic doesn't break should the name ever change.
6. **Device ID or Network ID conflicts.** BACnet requires that device names, device IDs and network IDs be unique across the network. The AcquiSuite's *BACnet → Discover* tool will automatically check for duplicate device IDs or network IDs between all devices it discovers (not just the AcquiSuite itself).

For example:

BACnet Server Status ? ↻

Server status: ✔ OK, but we recommend giving this server a [permanent name](#) or [static IP](#)

Device status: ⚠ WARNING: 4 meters shared, [1 meter error](#). Meter errors will slow BACnet responses

Network status: ✔ OK: Incoming FDR allowed

Config consistency: ✘ ERROR: Unworkable configuration! APDU Timeout would be too large! Please turn off [Search for Modbus Devices](#)

Units & scaling status: ⚠ WARNING: 1 meter has units which couldn't be converted to BACnet units, [View Log](#)

4.3 BACnet Setup Options

All BACnet Server options are set in one place, the *BACnet* → *Setup* page:

BACnet Server Status ? ↻

Server status: ✔ OK, but we recommend giving this server a [permanent name](#) or [static IP](#)


Device status: ✔ OK: 3 meters shared via BACnet

Network status: ✔ OK: Incoming FDR allowed

Config consistency: ✔ OK: Using recommended timeout [6 seconds]

Units & scaling status: ✔ OK: all meters' units converted to BACnet

Server uptime: 1 hrs, 31 min, 2 sec

Server version: v02.13.0312_01  Tested for conformance and accepted by BACnet® Testing Laboratories

BACnet protocol version: version 1, revision 12

BACnet/IP address: 10.55.70.3

Server's BACnet object name: Auto: AcquiSuite_70_3 Custom:

Server's BACnet base object ID: Auto: 3000 Custom: (0-4194302; use [Discover](#) to choose unused ID).

Server's virtual network number: Auto: 33399 Custom: (1-65534; use [Discover](#) to choose unused net number).

Server's BACnet/IP UDP port:

APDU timeout: 6 seconds or greater recommended.

APDU retries: We advertise these values; BACnet clients should use it when talking to us.

BBMD mode: The AcquiSuite can act as a BBMD if this subnet doesn't already have one. BACnet networks spanning multiple subnets require a BBMD on each subnet.

Distribute broadcasts to: List the IP:port of BBMD's on other subnets, separated by spaces. e.g.,
192.168.1.1 192.168.10.1:47809.

	Total	Per Minute	<input type="button" value="Clear Statistics"/>
BACnet packets received:	2	18.5	
BACnet Read-Property:	0 (0% cache hits)	0	
BACnet Write-Property:	0	0	
Backend Read Reqs / Errs:	0 / 0	0	
Backend Write Reqs / Errs:	0 / 0	0	
Internal errors:	0	--	

BACnet debug information:

Server's BACnet *object name*: Unique name of this server. Used as prefix for name of all data point objects. Must be unique on BACnet network. Automatically derived from server's current IP address if not specifically set.

Server's BACnet *base object ID (0-4194302)*: Unique object ID of this server and base of each meter's Object ID. Must be unique on the BACnet network. Automatically derived from server's current IP address if not specifically set. Use the [Discover](#) page to check for conflicts.

Server's virtual *network number (1-65534)*: The AcquiSuite assigns each meter it shares an address on a *virtual BACnet subnet*. This parameter sets the Network Number of that subnet. The value is arbitrary but must be unique on the BACnet network. Automatically derived from server's current IP address if not specifically set. Use the [Discover](#) page to check for conflicts.

Server's BACnet/IP *UDP port*: UDP port used for communication. The standard BACnet port is 47808 = 0xBAC0. May be changed to allow several independent BACnet networks to co-exist.

APDU timeout, APDU retries: Packet timeout and retry count *which BACnet peers should use when talking to this AcquiSuite*. We advertise these values. A warning is given if the timeout is less than the AcquiSuite's Modbus timeout. Use the [Discover](#) page to view the APDU timeout and APDU retries of other BACnet peers.

Note: [bacnet.org's Interpretation, 135-2004-20 - January 27, 2007](#) clarifies the meaning of APDU_Timeout and APDU_Retries.

BBMD mode: The AcquiSuite can act as a BBMD, a protocol feature allowing BACnet networks to [span IP subnets](#).

- **Allow incoming FDR (default):** By default, the AcquiSuite allows incoming Foreign Device Registration, which means BACnet peers on other subnets may use the AcquiSuite as a relay to join the local BACnet network. Up to 64 foreign devices may register simultaneously.
- **Full BBMD:** Select this option if you wish the AcquiSuite to distribute broadcasts to other IP subnets. Incoming FDR is also allowed.

BBMD mode:	<input type="radio"/> Full BBMD (distribute broadcast)	<i>The AcquiSuite can act as a BBMD if this subnet doesn't already have one. BACnet networks spanning multiple subnets require a BBMD on each subnet.</i>
Distribute broadcasts to:	<input type="text" value="192.168.10.47:47808 192.168.10.92:47809"/>	<i>List the IP:port of BBMD's on other subnets, separated by spaces. e.g., 192.168.1.1 192.168.10.1:47809.</i>

This option is only needed if:

- Your BACnet network spans multiple IP subnets, and
 - The local subnet doesn't already have a BBMD.
- **Disable all BBMD features:** Select this option to completely disable the BBMD service, including Foreign Device registration. This is not recommended but may be necessary to prevent *broadcast loops* or *broadcast storms* caused by other misbehaving or incorrectly configured BBMDs.

*Note: BBMDs may also be used to join together two or more **virtual BACnet networks** which use different UDP ports, e.g., 47808 and 47809.*

Distribute broadcasts to: If **BBMD mode = Full BBMD** is selected, enter the list of IP addresses (and optional UDP ports) of other BBMD services on your BACnet network.

This list should be identical among all BBMDs -- you may safely include the AcquiSuite's own IP address in its list. You should have at most one BBMD per IP subnet. See [How Do I... Span Subnets with a BBMD](#)

BACnet debug information: Level of detail displayed in the AcquiSuite's *Debugging Messages*.

Errors & Summary (default): Log only errors and summary info. Requires the AcquiSuite's *Debugging Messages* be enabled (via *System* → *System Log Files*).

- **Transitions:** Also log "transitions" such as changes to System_Status and Reliability Properties.
- **Full Debug:** Also log detailed debug info. This will slow the server significantly.
- **Full Debug (AREA):** Debug logging for AREA only.

*Note that **Full Debug** can significantly slow the BACnet server and should only be used until a problem is resolved.*

4.4 Handling Unsupported Meters with the Modbus Framework

If the AcquiSuite does not recognize a Modbus meter, use the AcquiSuite's *Modbus* → *Framework* feature to create a custom driver.

These so-called "Framework Drivers" will work with the BACnet Server.

See: "[AcquiSuite Modbus Framework](#)", available from [Obvius.com](#).

5 BACnet Device List

The **Device List** page shows the meters shared by the AcquiSuite as they appear in the BACnet world.

BACnet Device List - Local Devices ? ↻

Show: [\[basic\]](#) [\[network\]](#) [\[firmware\]](#) [\[stats\]](#) [\[all\]](#)

Device ID	Status	Device Name	BACnet Objects	Vendor	Model	Des
3000	alive, operational	AcquiSuite_70_3	objects	Obvius	A8812-1	TC50361 m
3001	alive, operational	AcquiSuite_70_3 Thermostat	objects	Obvius	Veris Industries,CWLPHTX,00101FC2	The
3249	alive, operational	AcquiSuite_70_3 Power Meter	objects	Obvius	Obvius Power Meter Simulator	Pow
3250	alive, operational	AcquiSuite_70_3 Internal I/O	objects	Obvius	Obvius, A8812, Internal I/O	Internal I/O, SN

Total devices: 4 Show: [\[basic\]](#) [\[network\]](#) [\[firmware\]](#) [\[stats\]](#) [\[all\]](#)

5.1 Typical Uses of the Device List page

- Dump Object List of each meter to see how exactly how the AcquiSuite will present it to the BACnet network.
- View BACnet/IP address-info for each shared meter using [\[network\]](#) tab.
- Check for duplicate BACnet Device ID and Network Numbers.
- View per-meter statistics to verify operation and performance using [\[stats\]](#) tab.

The **Device List** will also check for conflicting BACnet Device IDs and Network Numbers assigned to meters on this AcquiSuite and other BACnet devices.

6 BACnet Discover

The **Discover** page allows you to use the AcquiSuite to scan the BACnet network.

BACnet Discover

Scan from (Device ID): to (Device ID): (broadcasts Who-Is) Wait for response

- Press to scan the entire BACnet network.
- You may limit the scan to a range of BACnet Devices by entering *from* and *to* Device IDs (0 to 4194302).

Scan from (Device ID) _____, to (Device ID) _____: Enter a range of BACnet Device ID numbers or use the default. Narrowing the range is useful in large networks.

Note: The AcquiSuite always sends a global Who-Is broadcast to all networks.

Wait _____ for response: Increase the amount of time to wait for responses if the network is very slow.

: Click to begin scanning the network. The AcquiSuite will transmit a BACnet Who-Is broadcast message every 5 seconds and will collect responses.

BACnet Network

Scan from (Device ID): to (Device ID): (broadcasts Who-Is) Wait for response

🌀 Scanning BACnet Network... *Received I-Am Request from 2314945, MAC = 10.55.70.2.186.192* Show: [\[basic\]](#) [\[network\]](#) [\[firmware\]](#) [\[stats\]](#) [\[all\]](#)

🌀 Checking for duplicate Network & Device ID's...

For each BACnet device discovered, the AcquiSuite will:

- Ping the device's IP address (using an ICMP ECHO packet), printing either "**alive**" or "**no response to ICMP ping**".
- Query BACnet properties from the device.

The displayed information may be sorted by clicking on column headers.

BACnet Network

Scan from (Device ID): to (Device ID): (broadcasts Who-Is) Wait for response

Show: [\[basic\]](#) [\[network\]](#) [\[firmware\]](#) [\[stats\]](#) [\[all\]](#)

Device ID	Status	Device Name	BACnet Objects	Vendor	Model
3000	alive, operational	AcquiSuite_70_3	objects	Obvius	A8812-1
3001	alive, operational	AcquiSuite_70_3 Thermostat	objects	Obvius	Veris Industries,CWLP
3249	alive, operational	AcquiSuite_70_3 Power Meter	objects	Obvius	Obvius Power Mete
3250	alive, operational	AcquiSuite_70_3 Internal I/O	objects	Obvius	Obvius, A8812, In
211	alive, operational	ObviusJACE1_211	objects	Tridium	NiagaraAX St
2314945	alive, operational	Veris E50 Series Energy Meter - S/N: 4E341081	objects	Veris Industries, LLC	E50H5 Energy

Total devices: 6 Show: [\[basic\]](#) [\[network\]](#) [\[firmware\]](#) [\[stats\]](#) [\[all\]](#)

6.1 Typical Uses of the Discover page

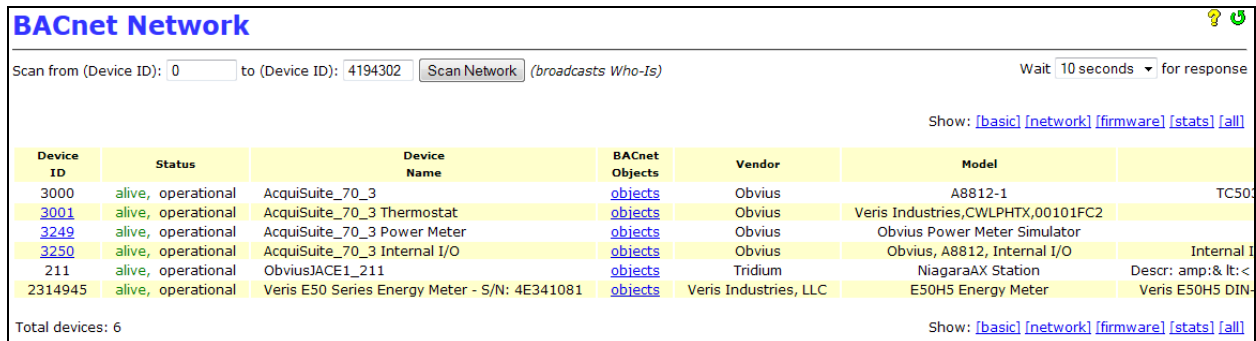
- See the Device IDs and Network Numbers already in use, to help choose these numbers for the AcquiSuite.
- See warnings if the AcquiSuite detects conflicts between *any* Device IDs or Network Numbers on your network.
- Verify BACnet communication between the AcquiSuite and other BACnet devices.
- Verify BACnet communication across subnets if using the BBMD or FDR features.
- Identify IP routing or firewall problems (the AcquiSuite automatically pings each BACnet device).
- Survey firmware versions of other BACnet devices on the network.

6.2 BACnet Discover → show: [network], [firmware], [stats]

The AcquiSuite organizes discovered information into several tabs.

6.2.1 show: [basic]

Shows "basic" properties of the device.



The screenshot shows the 'BACnet Network' interface. At the top, there are input fields for 'Scan from (Device ID): 0' and 'to (Device ID): 4194302', a 'Scan Network' button, and a 'Wait 10 seconds for response' dropdown. Below this is a 'Show:' menu with options: [basic] [network] [firmware] [stats] [all]. The main content is a table with the following data:

Device ID	Status	Device Name	BACnet Objects	Vendor	Model	
3000	alive, operational	AcquiSuite_70_3	objects	Obvius	A8812-1	TC50
3001	alive, operational	AcquiSuite_70_3 Thermostat	objects	Obvius	Veris Industries,CWLPHTX,00101FC2	
3249	alive, operational	AcquiSuite_70_3 Power Meter	objects	Obvius	Obvius Power Meter Simulator	
3250	alive, operational	AcquiSuite_70_3 Internal I/O	objects	Obvius	Obvius, A8812, Internal I/O	Internal I
211	alive, operational	ObviusJACE1_211	objects	Tridium	NiagaraAX Station	Descr: amp;& lt: <
2314945	alive, operational	Veris E50 Series Energy Meter - S/N: 4E341081	objects	Veris Industries, LLC	E50H5 Energy Meter	Veris E50H5 DIN-

Total devices: 6

Show: [basic] [network] [firmware] [stats] [all]

6.2.2 show: [network]

Shows network-level details (BACnet/IP address, network number, APDU timeouts and APDU retries).

Useful for debugging timeout errors.

BACnet Network

Scan from (Device ID): 0 to (Device ID): 4194302 [Scan Network](#) (broadcasts Who-Is) Wait 10 seconds for response

Show: [\[basic\]](#) [\[network\]](#) [\[firmware\]](#) [\[stats\]](#) [\[all\]](#)

Device ID	Status	Device Name	BACnet/IP MAC	NET	ADR	APDU Len	APDU Timeout (ms)	APDU Retries	Vendor
3000	alive, operational	AcquiSuite_70_3	10.55.70.3:47808	0	00	1476	6000	2	Obv
3001	alive, operational	AcquiSuite_70_3 Thermostat	10.55.70.3:47808	33399	03:46:36:01:BA:CO	1476	6000	2	Obv
3249	alive, operational	AcquiSuite_70_3 Power Meter	10.55.70.3:47808	33399	03:46:4A:F9:BA:CO	1476	6000	2	Obv
3250	alive, operational	AcquiSuite_70_3 Internal I/O	10.55.70.3:47808	33399	03:46:1B:FA:BA:CO	1476	6000	2	Obv
211	alive, operational	ObviusJACE1_211	10.55.70.2:47808	0	00	480	3000	3	Trid
2314945	alive, operational	Veris E50 Series Energy Meter - S/N: 4E341081	10.55.70.2:47808	1000	02	480	60000	0	Veris Indu

Total devices: 6 Show: [\[basic\]](#) [\[network\]](#) [\[firmware\]](#) [\[stats\]](#) [\[all\]](#)

6.2.3 show: [firmware]

Shows firmware versions. Useful for comparing versions to debugging compatibility problems.

Device ID: 4194302 [Scan Network](#) (broadcasts Who-Is) Wait 10 seconds for response

Show: [\[basic\]](#) [\[network\]](#) [\[firmware\]](#) [\[stats\]](#) [\[all\]](#)

Device Name	BACnet Objects	Vendor	Model	Firmware Rev	App S/W Vers
Thermostat	objects	Obvius	A8812-1	02.13.0312_01/0.8.0	02.13.0312_01/0.8.0
Power Meter	objects	Obvius	Veris Industries,CWLPHTX,00101FC2		GW=02.13.0312_01/0.8.0;
Power Meter Simulator	objects	Obvius	Obvius Power Meter Simulator		GW=02.13.0312_01/0.8.0;
Internal I/O	objects	Obvius	Obvius, A8812, Internal I/O	1.11	FW=1.11; GW=02.13.0312_01/0.8.0;
NiagaraAX Station	objects	Tridium	NiagaraAX Station	3.6.31.4	Tridium 3.6.31
Energy Meter - S/N: 4E341081	objects	Veris Industries, LLC	E50H5 Energy Meter	1018	RS=2004, OS=1041, BACnet Gateway=1018

Show: [\[basic\]](#) [\[network\]](#) [\[firmware\]](#) [\[stats\]](#) [\[all\]](#)

6.2.4 show: [stats]

BACnet Network

Scan from (Device ID): 0 to (Device ID): 4194302 [Scan Network](#) (broadcasts Who-Is) Wait 10 seconds for response

Show: [\[basic\]](#) [\[network\]](#) [\[firmware\]](#) [\[stats\]](#) [\[all\]](#)

Device ID	Status	Device Name	RP	RP Cache Hits (%)	WP	Dev Error (if any)	Data Age	Unit Issues	Read Reqs / Errs	Write Reqs / Errs	I/O Response Time, Avg (ms)	I/O Response Time, Worst (ms)	Location
3000	alive, operational	AcquiSuite_70_3	0	0	0	0	0	0	0 / 0	0 / 0	0	0.0	
3001	alive, operational	AcquiSuite_70_3 Thermostat	117	0	0	0	54s	0	0 / 0	0 / 0	0	0.0	mb485-001
3249	alive, operational	AcquiSuite_70_3 Power Meter	117	0	0	0	54s	0	0 / 0	0 / 0	0	0.0	simulated
3250	alive, operational	AcquiSuite_70_3 Internal I/O	117	0	0	0	1s*	0	0 / 0	0 / 0	0	0.0	mb485-250
211	alive, operational	ObviusJACE1_211	--	--	--	--	--	--	-- / --	-- / --	--	--	Loc: amp;< gt;> quot;' apos;' back;' dollar;\$
2314945	alive, operational	Veris E50 Series Energy Meter - S/N: 4E341081	--	--	--	--	--	--	-- / --	-- / --	--	--	Installed location not yet identified

Total devices: 6 Show: [\[basic\]](#) [\[network\]](#) [\[firmware\]](#) [\[stats\]](#) [\[all\]](#)

Shows performance statistics:

- **RP:** # of BACnet ReadProperty or ReadPropertyMultiple requests
- **WP:** # of BACnet WriteProperty requests
- **Dev Error:** Shows current error-code reported by the meter, or 0 if OK. Hover to see the error text.
- **Data Age:** Shows current age of cached data. An "*" (asterisk) indicates an update is pending. If data is not requested, it will not be updated in the cache and so may appear very old.
- **Unit Issues:** Shows the number of points for which units [could not be mapped to BACnet units.](#)

- **Read Reqs / Errs:** # of read errors vs. total reads. A “read error” is counted if, when answering a BACnet RP or RPM request, the meter’s data was not already cached and we failed to get a response from the meter in 65 seconds.
- **Write Reqs / Errs:** # of write errors vs. total writes. A “write error” is counted if we either timeout or get an error response when writing to a backend device.
- **I/O Response Time, Avg:** Moving average of response-time from this meter (reads or writes).
Note: This differs from a meter's RTT (displayed in the AcquiSuite's Modbus → Device List → [stats] tab) in that it includes the effect of the overall load on the AcquiSuite, the communications to all other meters, logging activity, etc.
- **I/O Response Time, Worst** = worst-case (peak) response-time from this meter, over the last 63 requests (reads or writes).

6.2.5 show: [all]

The *[all]* tab shows all fields in a single, very wide table. This is useful for cutting-and-pasting the table into other apps.

6.3 BACnet Discover → Objects

The Discover and Device List pages include an “objects” link next to each BACnet device. When clicked this queries and displays the **object list** of the local or remote BACnet device. For example:



Query (Device ID): **OBJECTS** (List objects and properties) Other commands: APDU Timeout: APDU Retries:

List of Objects in Test Device:

```
{
{
-- Warning: Property list is too large to query without segmentation, showing a subset of properties for this object-type
object-identifier: (Device, 211)
object-name "ObviousJACE1_211"
object-type: Device
system-status: operational
vendor-name: "Tridium"
vendor-identifier: 36
model-name: "NiagaraAX Station"
firmware-revision: "3.6.31.4"
application-software-version: "Tridium 3.6.31"
protocol-version: 1
protocol-revision: 4
protocol-services-supported: (
  T,T,T,T, -- AcknowledgeAlarm, COVNotification, EventNotification, GetAlarmSummary,
  T,T,T,T, -- GetEnrollmentSummary, SubscribeCOV, AtomicReadFile, AtomicWriteFile,
  T,T,F,F, -- AddListElement, RemoveListElement,,,
  T,F,T,T, -- ReadProperty,, ReadPropertyMultiple, WriteProperty,
  T,T,T,F, -- WritePropertyMultiple, DeviceCommunicationControl, PrivateTransfer,,
  T,F,F,F, -- ReinitializeDevice,,,
  F,F,T,T, -- ,, I-Am, I-Have,
  T,T,T,F, -- COVNotification, EventNotification, PrivateTransfer,,
  T,T,T,T, -- TimeSynchronization, Who-Has, Who-Is, ReadRange,
  T,F,T,T, -- UTCTimeSynchronization,, SubscribeCOVProperty, GetEventInformation,
)
protocol-object-types-supported: (
  T,T,T,T, -- Analog Input, Analog Output, Analog Value, Binary Input,
  T,T,T,F, -- Binary Output, Binary Value, Calendar,,
  T,F,T,F, -- Device,, File,,
  T,T,T,T, -- Loop, Multi-state Input, Multi-state Output, Notification Class,
  F,T,F,T, -- ,, Schedule,, Multi-state Value,
  T,F,F,F, -- TrendLog,,,
  F -- ,
)
object-list: {
  (Device, 211), (TrendLog, 0), (TrendLog, 1) }
max-apdu-length-accepted: 480
segmentation-supported: segmented-both
apdu-timeout: 3000
number-of-APDU-retries: 3
device-address-binding: { (Device, 556250),33399,034618FABAC0,(Device, 556001),33399,03463601BAC0,(Device, 556249),33399,03464
database-revision: 601
-- Found 3 Objects
},
{
{
object-identifier: (TrendLog, 0)
object-name "Pushbutton status Log"
object-type: TrendLog
enable: TRUE Writable
stop-when-full: FALSE
buffer-size: 500
log-buffer: ? -- BACnet Error: property: read-access-denied
```

Note:

- BACnet's "Objects" are equivalent to Modbus's "Register blocks".
- BACnet's "Properties" are equivalent to Modbus's individual "Registers".

See [Comparison of BACnet vs. Modbus.](#)

6.3.1 Typical Uses of Discover → Objects link

- See precisely how the AcquiSuite will present a modbus device to the BACnet world.
- List the BACnet Object Instance Numbers for particular objects (e.g., AI_5).

- Verify communication between the AcquiSuite and other devices on the BACnet network.
- General debugging and exploring.

6.4 BACnet Discover → Objects → EPICS

"EPICS" is a file-format describing a BACnet device's features. See [Terminology: PICS, EPICS and BIBBs](#).

After viewing a device's objects using the "objects" link discussed above, the EPICS button may be used to cause the AcquiSuite to generate an template EPICS file for the BACnet device (either local or remote).

Note that this EPICS file will need to be edited to fill in details which the AcquiSuite is not able to detect.

6.4.1 Typical Uses of the *EPICS*

- Provide meter info in standard format to other BACnet tools.
- Assist in generating EPICS files for AcquiSuite meters, for Q/A testing with tools such as VTS or [Visual Test Shell](#).

7 How Do I...

7.1 How Do I... Choose Device IDs and Network Numbers?

BACnet Device Names and BACnet Device IDs are used to look up BACnet addresses just as hostnames are used (with DNS) to look up IP addresses.

By default, AcquiSuite automatically generates these based on the current IP address.

7.1.1 Background: Permanent Names

We recommend giving the AcquiSuite a permanent (custom) name and permanent (custom) Device ID, or giving it a static IP address.

BACnet controllers will find the AcquiSuite by its Device ID number, which is automatically derived from the IP address. If you configure the AcquiSuite to use DHCP, the IP address could change causing your BACnet network to stop communicating.

7.1.2 Background: Detecting Name and ID Conflicts


Device names, Device IDs and Network Numbers **must be unique** across the BACnet network.

Use the [Discover](#) page to see what names and numbers are in use.

The Discover page will detect conflicts in Device ID and Network Number. For example:

BACnet Network

Scan from (Device ID): to (Device ID): (*broadcasts Who-Is*)

 Duplicate ID(s) detected: Device:556249@10.55.70.132, Device:556249@10.55.78.3

7.2 How Do I... Choose Timeouts?

The **APDU Timeout** setting tells the BACnet network how much time to allow the AcquiSuite to respond.

Note: The ASHRAE 135.1 BACnet Protocol specification is confusing on the meaning of the APDU Timeout, suggesting that all BACnet devices on a network need to have the same APDU Timeout value. [Bacnet.org's Interpretation, 135-2004-20 - January 27, 2007](#) clarifies the meaning of the APDU_Timeout and Number_of_APDU_Retries properties: They are values advertised by a BACnet device (server) to the any/all BACnet clients.

The AcquiSuite will recommend a timeout based on its current Modbus/485 configuration.


As a rule of thumb:

- The **APDU Timeout** value should be 2x the AcquiSuite's Modbus timeout (set in *Modbus* → *Setup*). If set to "Auto", the AcquiSuite will use its recommended value, automatically updating the APDU Timeout to the BACnet network whenever the AcquiSuite's Modbus settings are changed.
- The **APDU Retries** value defaults to 2 and should be increased for unreliable networks (such as wireless networks).

The **APDU Timeout** and **APDU Retries** may be increased for especially slow networks or networks with a high number of errors (e.g., wireless LANs).

7.3 How Do I... Optimize Response Time?

The AcquiSuite monitors its configuration and the status of meters, and warns of situations causing poor performance:

Device status:	 WARNING: 8 meters shared, 2 meter errors. Meter errors will slow BACnet responses
----------------	--

To obtain the best possible response time:

- 1 Fix any meter errors.
- 2 If meter errors cannot be fixed, remove the meter from the AcquiSuite's Modbus → Device List so that the AcquiSuite does not try to communicate with it (and incur a Modbus Timeout).
- 3 Adjust the AcquiSuite's Modbus Timeout (in *Modbus* → *Setup*) to be no more than necessary.
- 4 In *System* → *System Log Files*, turn off *Debugging Messages*.
- 5 In *BACnet* → *Setup*, ensure **BACnet debug information** is not set to Full Debug.
- 6 In *Modbus* → *Setup*, ensure **Modbus debug information** is set to None.
- 7 In *Modbus* → *Setup*, turn off **Search for Modbus Devices** option.
- 8 In *Modbus* → *Setup*, increase the **Data logging period** (to log more slowly). Data logging takes priority over BACnet requests.
- 9 Adjust your BACnet controller to poll at the minimum acceptable rate for your application.
- 10 Increase the AcquiSuite's RS485 baud rate, especially if using meters with hundreds of points.
- 11 Use [Device List](#) → *show: [stats]* to find response problems. In particular, the **Data Age** column shows the age of cached data and shows an * (asterisk) when an update is pending. This provides a real-time view of update behavior. Generally, Data Age should increase up to the polling interval chosen by the BACnet client(s). The **I/O Response Time, Worst** column shows the worst response time of the last 63 updates.

7.4 How Do I... Optimize Polling?

You may poll meters as fast as **once per second**.

We recommend choosing the **slowest acceptable polling rate** because the AcquiSuite's RS485 bandwidth is limited and much lower than Ethernet bandwidth. This leaves the greatest amount of bandwidth for other purposes.

Whatever rate you choose, the AcquiSuite is optimized for **polling at a constant rate**.

The *BACnet* → *Setup* page displays statistics on the total number and per-minute rate of BACnet queries. These may be used to verify the expected polling rate.

	Total	Per Minute
BACnet packets received:	638857	87.9
BACnet Read-Property:	5951627 (100% cache hits)	816
BACnet Write-Property:	423	0
Backend Read Reqs / Errs:	516872 / 0	63
Backend Write Reqs / Errs:	423 / 0	0

The *Device List* → *show: [stats]* tab will give you statistics on each device, allowing you to see which is the most active.

Click the column headers to sort by the number of requests.

Device ID	Status	Device Name	RP *	RP Cache Hits (%)	WP	Dev Error (if any)	Data Age	Unit Issues	Read Reqs / Errs	Write Reqs / Errs	I/O Response Time, Avg (ms)	I/O Response Time, Worst (ms)	Location
59000	alive, operational	AcquiSuite_10_59	0	0	0	0	0	0	0 / 0	0 / 0	0	0.0	
59001	alive, operational	AcquiSuite_10_59 Panel C New Capital Electric Added	1411	0	0	0	123s	0	2 / 0	0 / 0	0	0.0	mb485-001
59005	alive, operational	AcquiSuite_10_59 Panel B Old Existing Electrical Panel	1411	0	0	0	122s	0	2 / 0	0 / 0	0	0.0	mb485-005
59250	alive, operational	AcquiSuite_10_59 Internal I/O	263848	100	423	0	4s	0	97261 / 0	423 / 0	784ms	13.5s	mb485-250
59003	alive, operational	AcquiSuite_10_59 Panel A Old Existing Electrical Panel	5687135	100	0	0	1s	0	419782 / 0	0 / 0	58ms	1.5s	mb485-003

Total devices: 5

Show: [basic] [network] [firmware] [stats] [all]

7.4.1 Background: Caching Behavior

The AcquiSuite caches data it reads from backend meters.

Reading any point of a meter generally causes a read of all points, all of which are then cached.


The AcquiSuite's data logging cycle updates the cache as well.

Writes (e.g., to relays) are never cached.

7.5 How Do I... Fix Unit Issues?

AcquiSuite units (Volts, Amps, kWh, etc.) are automatically translated to the corresponding BACnet encoding.

If non-standard unit abbreviations have been entered, the AcquiSuite will warn of **Unit Issues**.

Units & scaling status:  WARNING: 1 meter has units which couldn't be converted to BACnet units, View Log

Click "View Log" to see each unit issue; correct the units of the meter and point in question.

*Note: In all cases, **the value of the data point will be correct** even if its units cannot be translated. Untranslatable units are encoded as dimensionless values (i.e., as a pure number). This allows new unit-translations to be added in the future without disrupting existing applications.*

7.6 How Do I... Use Relay Outputs?

The following relays are supported:

- Obvius, A8812 Internal I/O
- Obvius, A8332-8F2D Flex I/O
- Advantec, Adam 4068 and 4069
- Acromag, 902MB
- Coyote DataComm, Rio-9F
- Benjamin Electric, Power Panel Plus

Relays are modeled as BACnet **Binary Value** objects.

They are not "commandable" (i.e., they have no Priority_Array property).

In other words, the last write to the relay "wins".

Reading the **Present_Value** property of a Binary Value object returns the current state of the relay, which may be different than the last written value if the relay was changed by some means outside of BACnet (e.g., manual switch, Modbus/TCP write, AcquiSuite's web UI, power cycling, etc.)

*Note 1: Some controllers such as the Tridium JACE assume **Binary Value** objects are read-only. After discovering the relay's **Binary Value** object, you must change the JACE's point to be of type "BinaryWritable" so the JACE will write to it.*

Note 2: Some controllers allow setting limits on how frequently a relay is changed. This feature of the controller can give the impression that the relay is not functioning, when in fact, writes are being throttled.

7.7 How Do I... Span Subnets with a BBMD?

The **BBMD feature** lets you place the AcquiSuite on an IP subnet different from the one containing your BACnet controller.

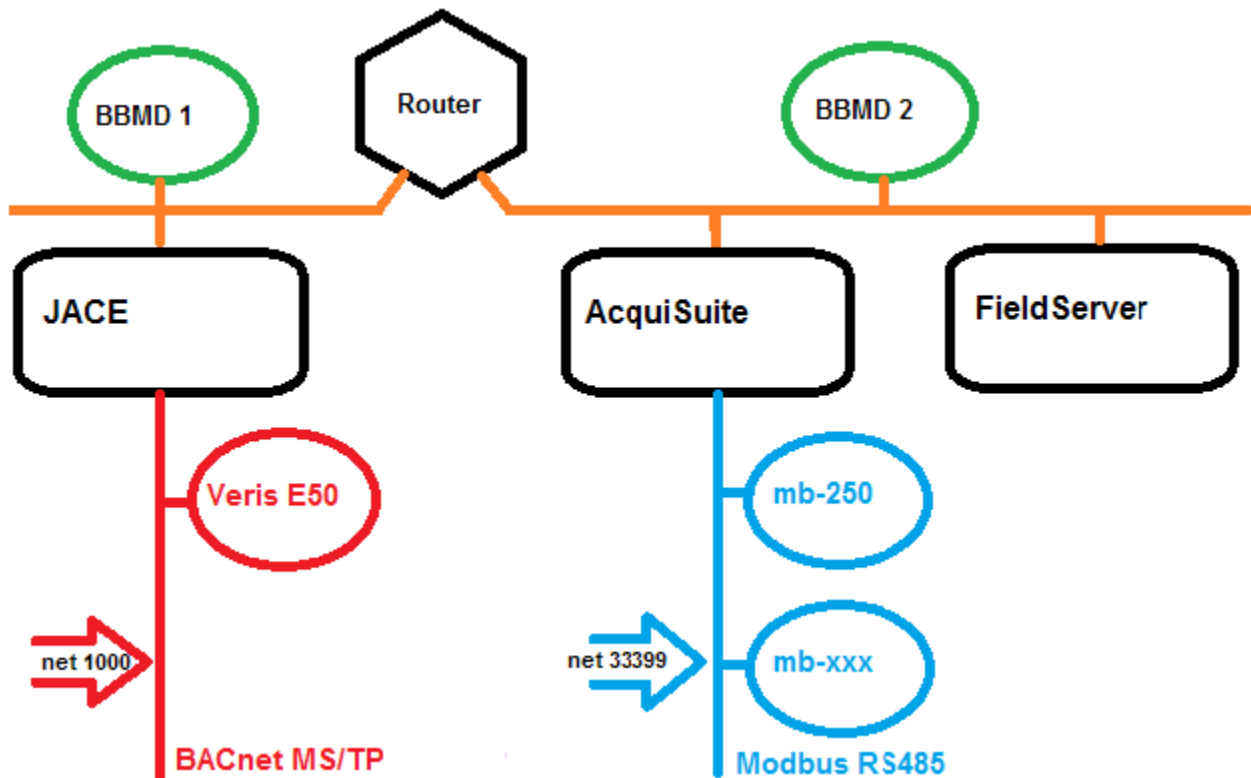
“BBMD” stands for **BACnet Broadcast Management Device**, a service which forwards BACnet broadcast messages from the local IP subnet to one or more remote IP subnets.

The BACnet Server for AcquiSuite is able to act as a full BBMD, to provide flexibility in placing the AcquiSuite on remote IP subnets.

BACnet/IP relies on broadcast messages (Who-Is, I-am) to discover devices. These broadcasts are usually blocked by IP routers.

When a BACnet network spans multiple IP subnets, a BBMD should be configured on each subnet with the IP addresses of all other BBMDs.

For example, in the following diagram, the BACnet network consists of two IP subnets (in orange), separated by an IP router.



The two BBMD’s forward BACnet/IP broadcasts between the two IP-subnets, allowing devices on each side of the router to discover devices on the other.

Note 1: BBMD’s only forward broadcasts. BACnet services such as ReadProperty or WriteProperty, which are never broadcast, pass through the IP-routers without any help.

*Note 2: The BBMD feature **is not needed** if your BACnet network is contained in a single IP subnet, or if the subnet already has a BBMD.*

7.8 How Do I... Span Subnets with Foreign Device Registration?

“FDR” stands for **Foreign Device registration**, a service which allows a BACnet device not on the local IP subnet to relay its communications through the AcquiSuite.

No configuration is needed to use FDR. It is a service the AcquiSuite provides, which any BACnet device may use.

Typically, BACnet devices or tools supporting FDR will provide an input-field where you can enter the AcquiSuite's IP address and UDP port.

7.8.1 Typical Uses of Foreign Device Registration:

- **Remote diagnostics.** BACnet/IP relies on broadcast messages which typically are not forwarded by IP routers. BACnet scanning or management tools which support FDR may relay through the AcquiSuite to access all BACnet devices on the AcquiSuite's subnet.
- **Stand-alone BACnet meters which don't support full BBMD mode,** may nevertheless support FDR. These meters can be configured to relay through an AcquiSuite to connect to a BACnet network spanning IP subnets.

Note: In the above cases, the AcquiSuite should be configured as a full BBMD if other BBMDs are present in the network. Otherwise, the "foreign device" relaying through the AcquiSuite will only be able to "see" the local subnet, not the entire BACnet network.

The FDR and BBMD features are compatible and may be used together.

64 FDR entries are allowed.

8 Troubleshooting

This section lists many common problems and possible solutions in order of likelihood.

8.1 Symptom: Not BTL Listed?

“BTL”, short for “BACnet Testing Laboratories”, verifies proper implementation of the BACnet protocol.

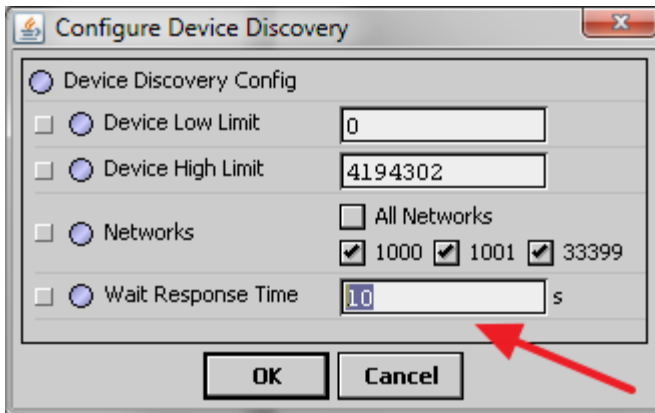
The “BACnet Server for AcquiSuite” was accepted by BTL as of version **v02.13.0312_01 or later**. This is available as a free upgrade. Earlier, “beta” versions should not be used.

8.2 Symptom: Can't “Discover” the AcquiSuite?

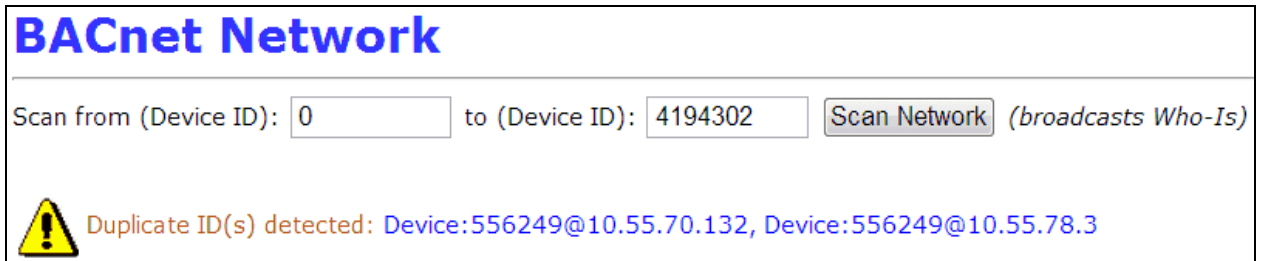
If your BACnet Controller fails to “discover” the AcquiSuite, check the following:

1. **Try “discovering” in the reverse direction:** Can the AcquiSuite see your Controller or other BACnet devices? Login to the AcquiSuite and use *BACnet → Discover*.
2. **Can you “discover” some objects on the AcquiSuite or none at all?** Even if no meters are connected to the AcquiSuite, it should always share at least one BACnet Device Object (representing the AcquiSuite itself).
3. **Basics: Check Ethernet cables, link-status LEDs, trying “pinging”...** Note that the AcquiSuite's “discovery” function will also ping each BACnet/IP device.
4. **Same UDP port?** Check that all BACnet/IP devices are using the same UDP port, by default 47808. The *BACnet → Setup* page will show the UDP port in use. Alternate UDP ports allow multiple distinct BACnet/IP networks to co-exist on a single LAN. *Note: that the [BBMD function](#) may be used to bridge between BACnet/IP networks on distinct ports.*
5. **Same LAN subnet?** Check that the AcquiSuite and your BACnet Controller are on the same IP subnet by comparing the IP addresses and subnet masks of each device. *Note: just because you can “ping” the opposite device does not mean they are on the same IP subnet – they may be separated by an IP router, which will block BACnet broadcasts.*
6. **If not on same subnet, is there a properly configured BBMD on each LAN?** Each BBMD must be configured with the IP-addresses (and UDP ports, if not the default) of all other BBMDs.
7. **Separated by NAT-router?** Are the AcquiSuite and the BACnet Controller separated by a NAT-router? “NAT” stands for “Network Address Translation” and allows several devices to share a single IP address. Any NAT-router must specifically support the BACnet protocol.
8. **Firewalls blocking UDP broadcasts?** Check that there are no firewalls between the AcquiSuite and your BACnet Controller, or if there are, that they are not performing a NAT function and that they are configured to forward broadcasts. Alternately, configure the AcquiSuite as a [BBMD](#).

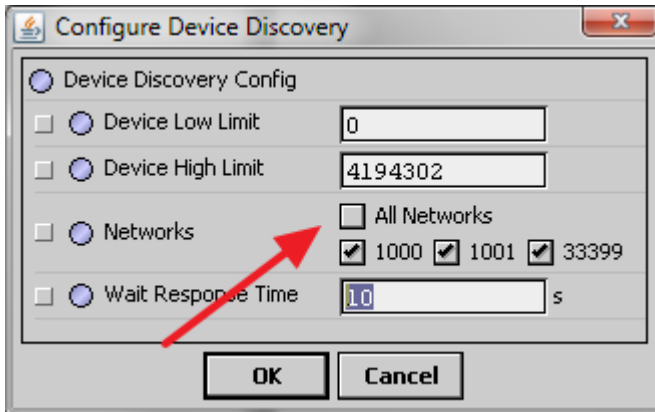
9. **Can your BACnet Controller wait longer for a “discovery” response?** The AcquiSuite should respond within a few seconds to any discovery (Who-Is) request, but in large networks with many responding devices, the BACnet Controller may be overwhelmed with responses. Try increasing the amount of time your Controller waits to collect responses. Here’s an example for the Tridium JACE:



10. **Are there any Conflicts or Duplicate Device IDs or Network Numbers in your BACnet Network?** BACnet requires that Device Names, Device IDs and Network Numbers should all be unique across the entire network. Your BACnet Controller may detect duplicates. If not, you can use the AcquiSuite’s [BACnet → Discover](#) page. This will check for conflicts of Device IDs and Network Numbers across your entire network.



11. **Device Name Conflicts?** Duplicate Device Names are less likely to cause problems, but should still be corrected. To detect duplicate names with the AcquiSuite, use the [BACnet → Discover](#) page and click the “Device Name” column to sort by name, then look for any duplicates.
12. Your BACnet Controller can discover the AcquiSuite itself, **but not meters attached to the AcquiSuite**. The AcquiSuite acts as a BACnet “virtual router”... this means that each meter shared by the AcquiSuite will appear to be on a separate BACnet subnet. Some Controllers may require you to select “All Networks” or “Global Broadcast” when doing a discovery:



Your Controller may also organize the BACnet devices **by network number**.

8.3 Symptom: “Discovery is Broken or Not Working” or “My Controller Doesn’t Support Discovery”?

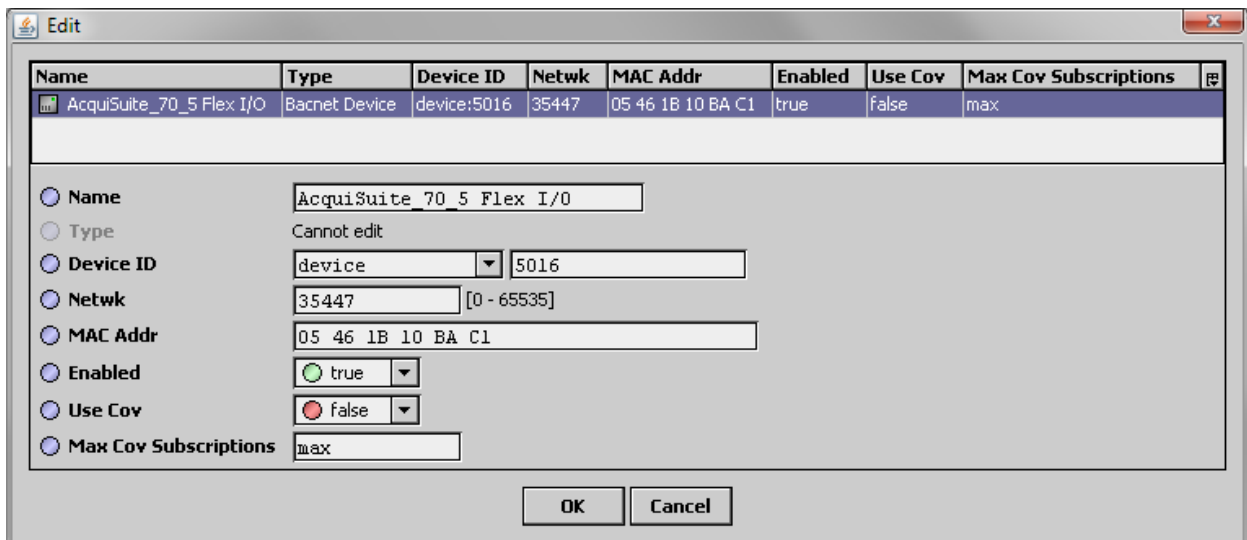
It is possible to connect to BACnet devices by manually entering their BACnet address information into your Controller. This is useful if:

- Discovery fails (and cannot be fixed by any of the above means),
- If the Controller doesn’t support discovery, or
- You need to re-number your network.

The AcquiSuite displays everything you need in the [show: \[network\]](#) tab of the [BACnet → Discovery](#) page:

- The NET and ADR columns give the network number and address, and
- The B/IP MAC column gives the BACnet/IP MAC address.
- The [“objects”](#) link will display a device’s Object Instance Numbers.

Here is an example of entering or changing address information on the Tridium JACE.



8.4 Symptom: BACnet Timeout or Communication Errors?

If your BACnet Controller fails to communicate after “discovery” is complete, or if it reports an unusually high number of communications or timeout errors, consider these possible causes:

1. **Recommended APDU Timeout in use?** The BACnet Server will recommend an APDU Timeout based on the AcquiSuite’s current Modbus configuration. This value is advertised to BACnet peers, but not all BACnet peers will use it. Verify the timeout on the BACnet Controller side.
2. **Review the AcquiSuite’s Modbus Timeout** (in *Modbus* → *Setup*). If the Modbus Timeout is incorrect for the meters being used, the recommended BACnet APDU Timeout will also be incorrect.
3. **Segmentation not supported.** Verify the BACnet Controller isn’t trying to use “Segmentation” when communicating with the AcquiSuite. Segmentation is not supported.
4. **Can ping? Both ways?** Verify you can ping the Controller from the AcquiSuite and vice-versa.

*Note: If you are spanning subnets with BBMDs, **you might be able to discover devices that you cannot directly communicate with!** This can happen if one BACnet device (on the remote network) has an incorrect default route.*

To guard against this unusual situation, the AcquiSuite will automatically ping each BACnet/IP device it discovers.

5. **Can ping with large packets?** Some types of network errors affect large packets more than small ones. The default size of packet used by “ping” is 64 bytes. Both Windows and Unix/Linux allow you to increase this default size when pinging to test the network’s handling of large packets. For example, on Windows, the command is: `ping -l SIZE HOSTNAME`

```
C:\Users\mns>ping -l 1500 10.55.70.1

Pinging 10.55.70.1 with 1500 bytes of data:
Reply from 10.55.70.1: bytes=1500 time=36ms TTL=64
Reply from 10.55.70.1: bytes=1500 time=26ms TTL=64
Reply from 10.55.70.1: bytes=1500 time=3ms TTL=64
Reply from 10.55.70.1: bytes=1500 time=2ms TTL=64

Ping statistics for 10.55.70.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 2ms, Maximum = 36ms, Average = 16ms
```

6. **Other devices are BTL Listed?** Verify the other devices are BTL Listed. If not, check for firmware updates. The AcquiSuite’s [Discover](#) → [show: \[firmware\]](#) tab will display the firmware versions of all devices.

8.5 Symptom: Data Values are not Updating (not Refreshing)?

This section deals with the symptom of data values which are not updating (not refreshing) or which the Controller reports as “stale” or “down”.

Database								6 objects	
Name	Exts	Device ID	Status	Netwk	MAC Addr	Vendor	Model	Firmware Rev	
Obvius_TC50361 Internal I/O		device:55625	{down,alarm,unackedAlarm}	33399	03 46 1B FA BA C0	Obvius	Obvius, # 1.11		FW
Obvius_TC50361 Thermostat		device:55600	{down}	33399	03 46 36 01 BA C0	Obvius	Veris Indt		GW
Obvius_TC50361 PowerMeter OK		device:55624	{down}	33399	03 46 4A F9 BA C0	Obvius	Obvius Pt		GW

1. **From the BACnet Controller’s point of view, are all devices “stale/down”, or just some?** All devices suddenly being “stale/down” suggests a networking problem between the Controller and the AcquiSuite. See the section above, [Symptom: BACnet Timeout or Communication Errors](#).
2. **Is meter data being updated (refreshed) on the AcquiSuite side?** The **Data Age** column of the [BACnet → Device List → show: \[stats\] tab](#) will show how old each meter’s data is, and the **RP** column (ReadProperty) will increment for each request. The **Data Age** column will also show an “*” (asterisk) when an update is pending but not yet completed. If the AcquiSuite is successfully completing requests, **Data Age** should periodically reset to 0. The length of time the “*” appears will let you visually gauge how fast the AcquiSuite is refreshing data.
3. If the problem appears to be on the AcquiSuite-side, see [How Do I Optimize Response Time](#) for solutions.
4. If the problem appears to be on the BACnet Controller side,
 - a. Check for [Device Name, Device ID or Network Number conflicts](#).
 - b. Check if the **Device ID of the AcquiSuite has simply changed**. The most likely reason this could occur is because the AcquiSuite hasn’t been assigned a [permanent BACnet name and ID](#), and it is using DHCP, and its DHCP-assigned address has changed.
 - c. Finally, try re-discovering the AcquiSuite from the Controller.

8.6 Symptom: Can’t Write (to Relays)?

If your BACnet Controller isn’t able to write to relays shared by the AcquiSuite, consider these possible causes:

1. **Supported relay type?** Make sure the [relay type is supported](#) by the BACnet Server. If not, please contact Obvius to request support for the new relay.
2. **Attempting to write to Binary Value object?** The AcquiSuite represents each relay as a pair of objects – an Analog Input object and a Binary Value object. Only the Binary Value object may be written. The Binary Value object’s name will be suffixed with “ Relay”. For example, here is how two relay points would appear to a Controller:

Name	Out	Object ID	Property ID	Index	Read	Write
Fan	0.00 {ok}	analogInput:41	Present Value	-1	Polled	readonly
Fan Relay	false {ok} @ 9	binaryValue:41	Present Value	-1	Polled	OK
Output 02	0.00 {ok}	analogInput:42	Present Value	-1	Polled	readonly
Output 02 Relay	true {ok} @ 9	binaryValue:42	Present Value	-1	Polled	OK

- Meter communication errors (at Modbus level)?** To verify that the AcquiSuite itself can successfully set or clear the relay, try changing the relay's state from the AcquiSuite's web UI. Login as "admin", navigate to *Modbus* → *Device List*, and click on the relay device. You should see a list of points. Locate the relay, and you should see buttons to turn it on or off, like so:

Fan	0	Turn On
Blinker	1	Turn Off

- APDU Timeout too low?** It possible, though unlikely, that the APDU Timeout is high enough to allow reads to succeed, but not writes. BACnet reads are satisfied very quickly from cache, while writes to relays always result in a Modbus write to the relay. This write must finish before the BACnet Server will reply with success. Assuming you've already verified the Modbus timeout is sufficient to write to the relay (above), try doubling the APDU Timeout.
- AcquiSuite's RS485 bus is overloaded?** Each BACnet write to a relay's **Present_Value** will result in a Modbus/485 request on the AcquiSuite's RS485 port. If the RS485 port is simply overloaded (all bandwidth in use), requests will be delayed, and if this delay is great enough, it will exceed the BACnet Server's APDU Timeout. To diagnose this, navigate to [BACnet](#) → [Device List](#) → [show: \[stats\] tab](#), and look at the "**I/O Response Time, Worst-case**" column (indicated by arrow).

BACnet Device List - Local Devices

Show: [basic] [network] [firmware] [stats] [all]

Device ID	Status	Device Name	RP	RP Cache Hits (%)	WP	Dev Error (if any)	Data Age	Unit Issues	Read Reqs / Errs	Write Reqs / Errs	I/O Response Time, Avg (ms)	I/O Response Time, Worst (ms)	Location
556000	alive, operational	Obvius_TC50361	0	0	0	0	0	0	0 / 0	0 / 0	0	0.0	
556001	alive, operational	Obvius_TC50361 Thermostat	40	0	0	0	53s	0	1 / 0	0 / 0	0	0.0	mb485-001
556249	alive, operational	Obvius_TC50361 Power Meter	40	0	0	0	53s	0	1 / 0	0 / 0	0	0.0	simulated
556250	alive, operational	Obvius_TC50361 Internal I/O	4601	100	2414	0	0*	0	4625 / 0	2414 / 0	215ms	1.5s	mb485-250

Total devices: 4

Show: [basic] [network] [firmware] [stats] [all]

This value should always be less than the APDU Timeout. If this is the problem, several solutions are possible:

- Increase the APDU Timeout,
 - [Optimize response time](#),
 - Reduce the load on the RS485 bus by [reducing polling rates](#) or reducing the AcquiSuite's **Data logging period**,
 - Increase the RS485 bus bandwidth by reconfiguring your meters to operate at a higher baud rate.
- Your Controller may require writing be enabled in software.** For instance, the Tridium JACE Controller, when adding a Binary Value point, assumes it is read-only. You must specifically tell

the JACE that the point is of type **“Boolean Writable”** before the JACE will let you write to it.

Name	Type	Enabled	Object ID	Property ID	Index	Tuning Policy Name
Output 01 Relay	Boolean Writable	true	binaryValue:41	Present Value	-1	defaultPolicy

Name:
 Type:
 Enabled: true
 Object ID:
 Property ID:
 Index:
 Tuning Policy Name:
 Data Type:
 Read:
 Write:
 Device Facets: priPV=false >>
 Facets: >>
 Conversion:

7. **Does your Controller limit frequency of writes to protect equipment?** As an example, the Tridium JACE’s **“Default Tuning Policy”** for BACnet limits writes to once per hour. This feature is intended to protect equipment from too-frequent switching, so you should adjust it with care.

Default Policy (Bacnet Tuning Policy)

Min Write Time: [0ms - +inf]
 Max Write Time: [0ms - +inf]
 Write On Start: true
 Write On Up: true
 Write On Enabled: true
 Stale Time: [0ms - +inf]
 Poll Frequency:
 Use Cov: false
 Use Confirmed Cov: true
 Cov Subscription Lifetime: min

8.7 Symptom: Relays Turn On When They Shouldn't?

This section deals with the symptom of relays turning on when they shouldn't, or turning on seemingly at random, or appearing to have "glitches".

All of these symptoms result from lack of **fail safe logic design**.

"Fail safe logic design" means designing your control logic so that if communication with a sensor fails for any reason, the logic outputs (to relays) go to the inactive (off) state, or at least remain unchanged.

The easiest way to ensure your logic design is "fail safe" is to test it by physically disconnecting sensors and observing the result.

Even in a properly working system, momentary communications failures are inevitable for these reasons:

1. The AcquiSuite gives priority to logging of data over responding to BACnet requests. At the start of each data logging cycle, the AcquiSuite will pause in servicing Modbus read-requests for approximately 10 seconds.
2. The [response time](#) of the AcquiSuite varies with the "load" on the RS485 bus, and this load may vary depending on the number of requests received over the network (from BACnet, from XML, from Modbus/TCP, etc.)
3. Electrical noise (such as lightning strikes) which disrupt Ethernet routers or switches or RS485 buses.
4. By default, the AcquiSuite checks for firmware updates. Installing an update means the AcquiSuite will reboot, causing a delay of approximately 1-2 minutes.

You can deal with these issues by,

- Properly setting your Controller's "Stale Timeout" (if it supports such a feature),
- Having your Controller monitor the **Status_Flags** and **Reliability** properties on objects shared by the AcquiSuite.
- Incorporating both of the above into your control logic.

8.7.1 Setting a "Stale Timeout"

How you set a "stale timeout" is Controller-specific. For instance, the Tridium JACE has a **"Stale Time"** parameter in its BACnet Tuning Policy, which sets the maximum response delay before a data point is considered "stale" (see below). This should be set as follows:

Stale Time >= (APDU_Timeout * Number_of_APDU_Retries)

Default Policy (Bacnet Tuning Policy)

<input type="checkbox"/> <input type="radio"/> Min Write Time	00001h 00m 00s	[0ms - +inf]
<input type="checkbox"/> <input type="radio"/> Max Write Time	00000h 00m 00s	[0ms - +inf]
<input type="checkbox"/> <input type="radio"/> Write On Start	<input checked="" type="radio"/> true	
<input type="checkbox"/> <input type="radio"/> Write On Up	<input checked="" type="radio"/> true	
<input type="checkbox"/> <input type="radio"/> Write On Enabled	<input checked="" type="radio"/> true	
<input type="checkbox"/> <input type="radio"/> Stale Time	00000h 00m 18s	[0ms - +inf]
<input type="checkbox"/> <input type="radio"/> Poll Frequency	Fast	
<input type="checkbox"/> <input type="radio"/> Use Cov	<input type="radio"/> false	
<input type="checkbox"/> <input type="radio"/> Use Confirmed Cov	<input checked="" type="radio"/> true	
<input type="checkbox"/> <input type="radio"/> Cov Subscription Lifetime	15	min

8.7.2 How the AcquiSuite Signals Communications Failures

The BACnet Server for AcquiSuite will signal communications failures by updating the **Status_Flags** and **Reliability** properties of each data point's Analog Input or Binary Value object, and by setting the **System_Status** property of each Device Object.

These properties may be monitored by the Controller. For instance, on the Tridium JACE, these properties may be "added" in the same way you would discover and "add" the **Present_Value** of an input (see below). These properties can then be used to "gate" your control logic.

Bacnet Discover Points

Discovered					
Object Name	Object ID	Property ID	Index	Value	Description
AcquiSuite_70_5 Flex I/O	device:5016	systemStatus		Operational	Flex I/O, SN=[001EC6040208]
Fan Temp	analogInput:1	presentValue		71.18	Fan Temp
Fan Temp	analogInput:1	objectIdentifier		analogInput:1	
Fan Temp	analogInput:1	objectName		Fan Temp	
Fan Temp	analogInput:1	objectType		Analog Input	
Fan Temp	analogInput:1	presentValue		71.24	
Fan Temp	analogInput:1	statusFlags		none	
Fan Temp	analogInput:1	eventState		Normal	
Fan Temp	analogInput:1	outOfService		false	
Fan Temp	analogInput:1	units		Degrees Fahrenheit	
Fan Temp	analogInput:1	description		Fan Temp	
Fan Temp	analogInput:1	reliability		No Fault Detected	

9 BACnet Background

Here is a comparison between BACnet and Modbus concepts and terms:

BACnet concept	Modbus concept
"Object"	Register block (group of related registers)
"Property"	Individual register
"Device Object" (DO)	Register block containing Meter name, firmware version, overall status flags, etc.
"Analog Input Object" (AI)	Register for individual data point
"Binary Value Object" (BV)	Register for individual relay output
"ReadProperty" (RP), "ReadPropertyMultiple" (RPM)	ReadCoilStatus, ReadHoldingRegister, ReportSlaveID, etc.
"WriteProperty" (WP)	PresetSingleRegister
MS/TP (BACnet over RS485)	Modbus/RTU (Modbus over RS485)
BACnet/IP (BACnet over UDP/IP, default port 47808)	Modbus/TCP (Modbus over TCP/IP, default port 502)

9.1 Background on "Device Objects"

A BACnet **Device Object** collects together information such as Description, Location or Firmware Version related to a single device, such as the AcquiSuite itself, or each meter shared by the AcquiSuite.

Each Device Object is assigned a Device ID number, from 0 to 4194302 (a 22-bit number, $2^{22}-2$), which must be unique across the BACnet network.

Each Device Object is also assigned a name.

These two properties, Device ID and Device Name, are used to look up a device's BACnet address, much like a hostname is used to look up an IP address.

9.2 Background on "Analog Input Objects"

A BACnet **Analog Input Object** represents a single data point on a meter.

These are numbered starting from 1: AI1, AI2, AI3, ... where AI1 = Point 1, AI2 = Point 2, etc.

Unlike Modbus, where registers may have several formats (integer, unsigned, float, little-endian, big-endian, etc.), Analog Input Objects always provide data in a single IEEE float format.

9.3 Background on "Binary Value Objects"

A BACnet **Binary Value Object** represents a digital flag with a value of either 0 or 1.

If the meter has relay outputs, for each relay, the AcquiSuite creates one **Binary Value Object**. The name of the object is suffixed with " Relay".

9.4 Terminology: PICS, EPICS and BIBBs

PICS = BACnet "Protocol Implementation Conformance Statement". This is a "BACnet spec sheet" in a standard format.

EPICS = Electronic PICS. This is a more precise, machine-readable version of the PICS.

BIBB = BACnet Interoperability Building Block. This is a fancy way to describe a collection of related protocol features. Each class of BACnet device is expected to implement certain BIBBs in order to be certified.

10 Common Wireshark Filters

Wireshark is a freely-available Ethernet packet capture and network monitoring tool. It is frequently used for debugging BACnet/IP networks. It is available from <https://www.wireshark.org>.

The following table lists several Wireshark filter expressions which are useful for BACnet debugging.

Wireshark Filter	What It Matches
<code>udp.port == 47808</code>	All BACnet packets on standard port
<code>udp.port == 47808 or udp.port == 47809</code>	BACnet packets on port 47808 or 47809
<code>udp.port == 47808 and not ip.addr == 1.2.3.4</code>	Exclude one "noisy" network node at 1.2.3.4
<code>udp.port == 47808 and not bacapp.confirmed_service</code>	Exclude Confirmed Services (such as ReadProperty and WriteProperty). Useful for debugging device and object discovery.
<code>udp.port == 47808 and not (bacapp.confirmed_service == 12 or bacapp.confirmed_service == 14)</code>	Exclude ReadProperty and ReadPropertyMultiple. Useful for debugging WriteProperty.
<code>bacapp.confirmed_service == 15 and ip.addr == 5.6.7.8</code>	BACnet WriteProperty packets to/from IP address 5.6.7.8
<code>bacapp.type > 4</code>	BACnet Errors (type=5), Rejects (type=6) or Aborts (type=7)
<code>bacapp.unconfirmed_service</code>	BACnet Unconfirmed Service requests (Who-Is, I-Am, Who-Has, I-Have, etc.)
<code>bacapp.unconfirmed_service == 8 or bacapp.unconfirmed_service == 0</code>	BACnet Who-Is or I-Am packets (used for device discovery)
<code>bacapp.unconfirmed_service == 7 or</code>	BACnet Who-Has or I-Have packets

<code>bacapp.unconfirmed_service == 1</code>	(used for object discovery)
<code>udp.port == 47808 and (bacnet.sadr_eth == 01:02:03:04:05:06 or bacnet.dadr_eth == 01:02:03:04:05:06)</code>	BACnet requests to/from a particular meter (the address is the 6 byte "ADR" field shown in BACnet → Device List → show: [network])

This page intentionally left blank.