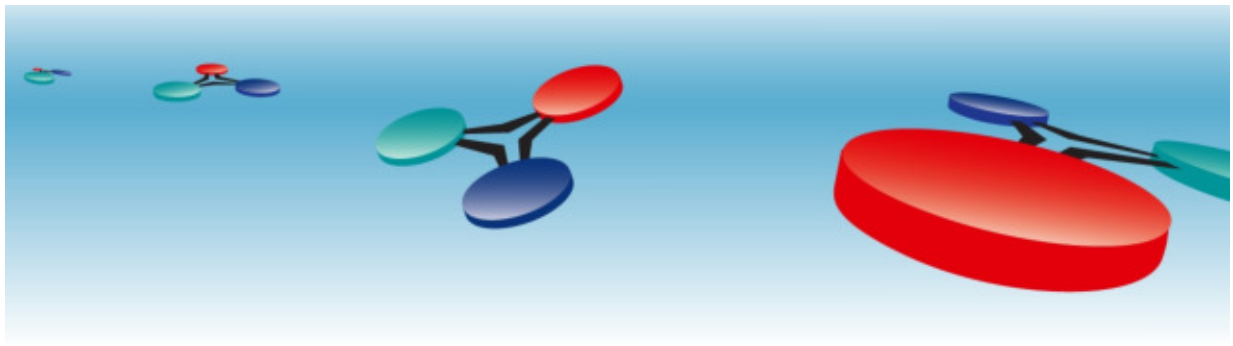

BACstac/Win v7.10

Installation Guide

Document Version: 1.0

Cimetrics, Inc.

BOSTON, MASSACHUSETTS
USA
TELEPHONE: +1 (617) 350-7550
FAX: +1 (617) 350-7552
E-MAIL: products@cimetrics.com (sales), support@cimetrics.com (tech. support)
WEB: http://www.cimetrics.com

cimetrics™

BACstac/Win v7.10Installation Guide

by Cimetrics, Inc.

Document Version: 1.0 Edition

Table of Contents

List of Tables

Chapter 1. Introduction

Thank you for your interest in Cimetrics' BACstac software. The BACstac Standard Edition provides you with the ability to create client and server application programs which use ASHRAE's BACnet protocol for communication. The BACstac Routing Edition also allows you to route between BACnet networks or make a proprietary network of devices appear as a virtual BACnet network containing multiple devices.

The BACstac Standard Edition and Routing Edition are available for Windows 7, Windows 8, Windows 8.1, Windows 10, Windows Server 2008, Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, and 32-bit embedded platforms.

This document describes how to install the BACstac software, as well as some features specific to the Windows version.

For a description of how to construct applications using the BACstac software, please refer to the BACstac Users' Guide. For a description of the data types and API routines available in Cimetrics' BACstac software, see the BACstac Programmers' Reference.

For a complete description of the BACnet protocol, please refer to American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc. ANSI/ASHRAE Standard 135-2020 or later, "BACnet™ - A Data Communication Protocol for Building Automation and Control Networks".

In order to use this software, you will need the following:

- Windows 7 or later
- Ethernet network interface card and appropriate cable
- The Npcap driver to use BACnet over Ethernet
- Microsoft Visual Studio 2022 or higher

Technical Support

Cimetrics offers technical support to assist you in developing applications using the BACstac software. The terms of the available technical support are found in your license agreement. You may contact us by any of the following means (e-mail is preferred):

telephone: +1 617-350-7550 (10 a.m. to 5 p.m. EDT)

fax: +1 617-350-7552

e-mail: support@cimetrics.com (mailto:support@cimetrics.com)

Chapter 2. Getting Started

The BACstac distribution consists of two parts: redistributable and developer parts. The redistributable part consists of binary files that you should install on user computers along with your BACstac application. The redistributable part for Windows includes the BACstac Service (BACSTAC.EXE), which provides the core of the BACnet protocol stack implementation. The developer part includes examples of BACstac applications, BACstac static libraries, and header files needed for developing BACstac applications.

> **Note:** To run BACstac applications, you must set up a correct BACstac configuration as described in the Configuration chapter.

Data Link Layers

The BACstac protocol stack supports several standard BACnet data link layers. Some of these data link layers require lower level operating system components to be installed as described below.

The BACnet/Ethernet data link layer uses the Npcap interface. The Npcap driver can be downloaded here (https://npcap.com/).

The BACnet/IP data link layer uses the TCP/IP protocol stack through the Windows Sockets API.

You can have any number of BACnet/Ethernet, BACnet/IP, BACnet/IPv6, BACnet/SC, and Virtual ports. A valid BACstac configuration should have at least one physical port.

Protocol Stack Installation

To install the BACstac stack on your computer, please run bacstac.msi located in the setup folder of the BACstac distributive. During the process of installation, the installer may ask you to reboot the computer. This happens when some installed components are used by the system and stopping the BACstac service is not sufficient to release them. To integrate BACstac stack installation into the installer of your BACstac based program, please use bacstac.msm located in the same directory. If you plan to use BACnet over Ethernet, you should also install Npcap driver.

> **Note:** During the installation, the BACstac installer asks for the correct BACstac configuration. The default configuration uses BACnet over IP with the default hostname as IP address. This configuration works for most installations. However, if the configuration happen to be invalid for any reason, the installation will fail complaining about inability to start the BACstac service. If you use BACnet/IP (the default configuration), please, make sure that you have at least one network adapter with TCP/IP enabled when you install bacstac.msi on the computer.

Uninstalling the Protocol Stack

To uninstall the BACstac protocol stack, click on the Start button, choose Settings and then click on the Control Panel in them. When the Control Panel dialog appears, double click on the Add or Remove Programs icon. In the Add or Remove Programs dialog, choose BACstac and click on the Remove button.

Protocol Stack Upgrade

During installation of BACstac 6.0 on a PC with BACstac 5.0 or an older version already installed, the installation program will ask you whether you want to remove the old version. Removing the old version is necessary for the successful installation of BACstac v6.0. If the BACstac was installed manually on that computer then you should agree, however if the BACstac was installed by a third-party program it may be safer to let that program to remove the previous version of BACstac.

Starting with BACstac 6.0, it is possible to have different versions of BACstac being installed on the same computer. If more than one BACstac is installed then the latest version is usually chosen to be active. The active BACstac service should be able to serve all BACstac applications. It is possible to install BACstac application based on different BACstac versions on the same computer. If the currently active BACstac service is uninstalled (as part of some BACstac based program) then another installed BACstac service is activated automatically.

> **Note:** The BACstac v6.0 Service is fully backward compatible with all BACstac 4.x and 5.0 versions, i.e. all BACstac 5.0 applications and BACstac 4.x applications using DLLs from previous BACstac versions can work correctly with the new service. However, new BACstac applications cannot work with old versions of the BACstac service.

Securing BACstac Installation

BACstac libraries use the BACstac named pipe to communicate with the BACstac service. Therefore, all BACstac applications need GENERIC_READ and FILE_WRITE_DATA access to the BACstac named pipe to work. The BACstac service grants those permissions to 'Everyone' (which is a special group that includes all authenticated users, the built-in Guest account, and several other built-in security accounts, such as SERVICE, LOCAL_SERVICE, NETWORK_SERVICE, and others).

Old BACstac libraries, which were released before March 2020, requested GENERIC_WRITE instead of FILE_-WRITE_DATA, and thus they requested more permissions that they actually need to work. It used to work, because the BACstac service granted GENERIC_WRITE to 'Everyone' anyway.

Starting with BACstac 7.4, we have tightened security on the BACstac named pipe. Now by default the BACstac service grants GENERIC_WRITE only if an old BACstac service is installed on the system, because it assumes that it could be an old BACstac application, which may require GENERIC_WRITE to work. This behavior is controlled the 'BACstacNamedPipeMode' parameter in the Cimetrics configuration in the Windows registry, which is located in 'HKLM\SOFTWARE\Cimetrics' on 32-bit systems and 'HKLM\SOFTWARE\Wow6432Node\Cimetrics' on 64-bit systems.

Two lowest bits of BACstacNamedPipeMode can have one of three values:

- 0 - If there is an old BACstac service installed on the system, then the new BACstac uses the backward compatibility mode, otherwise it uses the new mode with reduced permissions. This is the default value.
- 1 - The backward compatibility mode, i.e. it grants GENERIC_WRITE to 'Everyone' as BACstac did before.
- 2 - The new mode in which BACstac grants FILE_WRITE_DATA instead of GENERIC_WRITE to 'Everyone'.

If the BACnet/SC data link is used then read the "BACnet/SC Ports" section how to secure the private key, public certificates, and trusted CA certificates.

Development Environment Installation

To install the BACstac development environment, please unzip the BACstac archive to a directory which you have chosen to be the BACstac development directory on your computer.

The BACstac directory tree contains *bin*, *examples*, *include*, *lib*, and *setup* subdirectories. The README.TXT file at the top of distribution provides more details about each of them.

If you licensed the source code, you will find it in the *src* directory. Read the SOURCE.TXT file in it for information about creating a release or debug build for each included component: BACstac Service, libraries, etc. You might want to use debug build while developing your application because the debug version includes extra debug code (such as ASSERT() macros) as well as source-level debug information.

The BACstac distribution includes only the release build of all programs and libraries, therefore they contain no extra debug code or debug information.

Example Applications

The BACstac development distribution includes more than 20 examples of different types of BACstac applications (client, server, and gateway[*]) designed to illustrate various BACstac features. Please consult the README.TXT file in *examples* directory for instructions on how to build them.

Each BACstac server or gateway application represents a BACnet device, and therefore it must abide by the BACnet standard which, in particular, asserts that each device on the BACnet network shall have a unique Device ID and a unique Device Object Name. Each server or gateway application has some built-in values for them, which you might want to change before running it. You can also change a Device ID for a particular instance of one application by using the '*-i*' command line option.

Each gateway application requires an unused virtual network to attach to. The selection of the network is done by specifying its Port ID number in the current BACstac configuration (see chapter 4 for details). The default value of Port ID is 2, which corresponds the Port ID of the virtual network in default BACstac configuration. If you have changed the BACstac configuration, you must set an appropriate value for Port ID either by changing the built-in parameter, or by using the '-p' command line option.

Chapter 3. Windows-Specific Features

Multiple Processes

On Microsoft Windows, multiple processes can share access to the BACstac protocol stack simultaneously. Other protocols (such as UDP and TCP) include the concept of ports to allow multiple processes to share a common network interface. The BACnet protocol does not include a concept of an application or transport layer port, but there are several multiplexing mechanisms that fit the BACnet specification nicely:

- Network Layer (NL) multiplexing
- Application Layer (AL) or Service-based multiplexing
- Object-based multiplexing

With NL multiplexing, processes can be made to appear to be virtual devices on virtual networks, or a single process could be made to appear to be all of the devices on a virtual network. The BACstac Routing Edition allows this type of multiplexing. To accomplish this the NL has to act like a BACnet router. A Gateway application must also be capable of handling multiple "local" Devices.

With AL multiplexing, it is possible to concurrently run a single Server process and an unlimited number of Client processes. The BACstac Standard Edition allows this type of multiple process access (with a fixed total for the processes amount). This becomes possible because Confirmed Service requests are identified by an Invoke ID, and Unconfirmed Service requests are typically broadcasts or contain a Process ID field. Incoming replies and notifications can be matched up with the correct process, incoming broadcasts can be forwarded to all processes, and incoming requests go to the Server process. Local clients can talk to local servers transparently. Local messages do not show up on the network.

The BACstac Service implements AL multiplexing using a custom IPC mechanism. The IPC mechanism was chosen to allow portability to embedded systems, and it uses a fixed number of *slots*. The slots are recycled when an application stops running.

With Object-based multiplexing, Service requests that identify a particular Object (e.g. Read-Property) can be forwarded to a dedicated process. The BACstac Standard Edition does not support this type of multiplexing directly. A multiplexing server using an appropriate IPC mechanism (e.g. CORBA, COM, or UDP) can be built on top of the BACstac API.

Multiple Threads

Windows provides a multithreading environment. The BACstac software uses multiple threads to implement the protocol stack and it allows multiple application threads to use its API.

- Object Database locking is required for atomic service request handling in Servers
- API calls are multithread safe
- Hook processing is done in a separate thread
- Asynchronous hook completion is allowed outside of the hook-processing thread
- Asynchronous request completion notification is allowed in a dedicated thread

Chapter 4. Configuration

Configuring the Network Layer Ports

The Network Layer of the BACstac Standard Edition protocol stack dispatches messages from/to local and remote networks, through a single network port. The Network Layer of the BACstac Routing Edition protocol stack is a BACnet router between multiple physical and virtual network ports.

The Network Layer is a part of the BACstac service and maintains a routing table that can be changed by BACnet messages. When the BACstac service starts, it reads the initial values for the routing table from the Windows registry. When the information is changed by BACnet messages, it is written back to the registry.

The initial values for the routing table can be modified using the BACstac configurator, which can be invoked by clicking on `baccfg.bat`. If there is more than one version of BACstac is installed on the computer, the configurator of the currently active BACstac service is invoked regardless what `baccfg.bat` was clicked. Changed values of the Port table take effect only when the BACstac Service is restarted by rebooting or by using the Services Control Panel applet. You should not restart the BACstac Service while BACstac applications are running.

The routing table consists of a list of globally unique ports. Each port has a locally unique port ID, port type and BACnet network number. Ports can be physical or virtual. BACstac Routing Edition supports routing between multiple physical and virtual ports. At least one physical port (BACnet/Ethernet or BACnet/IP) must be included. Ports can be "unattached", so they are inaccessible until BACnet Network Layer messages assign network numbers to them.

For most installations, the only field in the default routing table that needs to be modified are the network numbers for attached each port. This is necessary, because each network must have a single unique network number. Each of the routers directly attached to a network must have this network number in their routing tables.

To add a new port to the BACstac configuration, invoke the BACstac configurator by clicking on `baccfg.bat`, then click on the "Add…" button and choose the desired port type. The Port Properties dialog box will be displayed. Fill in the fields and click OK.

Configuration of a BACnet/Ethernet port requires installation of the Npcap driver, which can be download from https://npcap.com/.

Port settings can be modified by clicking the Edit button in the Protocol Properties dialog box, which will call up a dialog box specific to the type of port. A port can be removed from the routing table with the Remove button in the Protocol Properties dialog box.

> **Important:** If you use the BACstac Routing Edition's BACnet routing capability, all network segments in a BACnet internetwork must be given unique network numbers. The BACnet Standard allows only one path between any two network segments. If this restriction is violated, a loop is created, and global broadcasts could be repeated around the loop up to 255 times (and a Who-Is/I-Am transaction could generate 64k broadcasts). The lifetime of a message is limited by the Hop Count field in the message. The BACstac Network Layer decrements this field value by the NL\Hop_Decrement parameter value specified in the Windows registry; the default value for this parameter is 1. The BACstac Network Layer attempts to detect loops and will discard suspicious broadcasts.

BACnet/Ethernet Ports

The BACnet/Ethernet Port Properties dialog box has fields for the port ID, network number, network interface card (NIC) and a check box to specify whether the port is attached.

An Npcap driver must be installed on the system. You can download it here (https://npcap.com/).

BACnet/IP Ports

The BACnet/IP Port Properties dialog box has three tabs, but you usually need to use the first tab only. The TCP/IP protocol (and any relevant drivers for the Ethernet card or Remote Access Services and modems) must be installed before using the BACnet/IP port.

The first tab of the BACnet/IP Port Properties dialog box contains fields for the port ID, network number, Adapter combobox, IP address, and UDP port number for this BACnet/IP port. It also contains a checkbox to specify whether this port is attached. You can choose a BACnet/IP binding either by choosing the desired adapter in the combobox, or by specifying the correct IP address. The first method is used when an IP address is received by DHCP or it may be changed often. The second method is useful when you don't want to bind to a specific adapter but you want to use a specific IP address for your BACnet device. If you have only one local network connection (one Ethernet card) or you are not interested in which of a few local connections will be used, then it is prefererable to leave the default values for both these fields -- Adapter: "Any", IP address: "Default". The UDP port number should probably be left as the default value of 47808 (in hexadecimal this is 0xBAC0).

The second tab is used to configure this BACnet/IP port to be a BACnet Broadcast Management Device (BBMD). BBMDs are used to propagate broadcasts. This is only necessary if your BACnet/IP network is going to span IP subnets or if you need to support connections from BACnet/IP Foreign Devices (FDs). Use this tab only if this device is to be a BBMD on the current subnet.

Starting with BACstac 6.1, it is possible to have more than one BBMD on the same IP subnet. It works only if other BBMDs on the same IP subnet also support Addendum 135-2008o. If there are two or more BBMDs on a single IP subnet, their BDTs must not contain any common entries. Also, masks referring to the same subnet are required to be identical in all BBMDs. For more detailed information, please, refer to Addendum 135-2008o.

Table entries are added using the three-edit fields and the Add/Remove buttons. Don't add an entry corresponding to the local BBMD — it is maintained internally by the BACstac. The Two-Hop Forwarding option overrides the local BBMD's subnet mask with the value 255.255.255.255. In addition, you can enable/disable the handling of Foreign Devices and specify the maximum size of the Foreign Device Table (FDT). Note that once configured as a BBMD, this port is required to respond to BBMD messages that can change the BDT and FDT. Changes to the BDT by BACnet/IP messages are stored in Registry, and will persist if the BACstac protocol is restarted, while changes to the FDT are not persistent.

The third tab is used to configure this BACnet/IP port to be a Foreign Device (FD). When the BACstac starts up, it will request that the specified BBMD include this port in the broadcasts for the BACnet/IP network. A time-to-live parameter is included in this request.

A BACnet/IP port can be configured as one of the following:

- a regular BACnet/IP port
- a BACnet/IP port and a BBMD device
- an FD port and FD device.

BACnet/SC Ports

The BACnet/SC Port Properties dialog box has three tabs. The first tab of the dialog box contains fields for the port ID, network number, URI of Primary and Failover hubs, paths to the Private key and the Operation certificate, and the folder that contains the list of public certificates trusted by this device to sign device certificates. It also contains a checkbox to specify whether this port is attached.

The creation of the private key, the public (operational) certificate related to the private key, and the management of the certificate signing authorities are site-specific deployment options. To connect to other devices, the public certificate of this device has to be signed by a certificate signing authority that other devices trust. To accept connections from other devices, the public certificate of other devices has to be signed by any certificate signing authority that this device trusts. The 'CA Certificates' folder contains the list of certificate signing authorities trusted by this device.

The private key should not be accessible by any regular user, and it requires the read permissions only for the SYSTEM account, under which the BACstac service is running. The public certificate can be readable by anyone, but it should be writable or deletable by anyone except the system administrator. The folder that contains CA certificates should be writable only by the system administrator and to be read by the SYSTEM account. When BACnet/SC is not used anymore, the private key should be removed from the computer to avoid potential leak of it.

The second tab is used to configure initiating and accepting direct connections between BACnet/SC devices. The binding parameters specify the address at which this device listens to incoming connections.

The bind address can be specified either by choosing the adapter or by specifying IP address. If both the 'Adapter' field and the 'IP Address' fields contain 'Any' then the data link binds to INADDR_ANY. If the adapter is 'Any' but the IP address is specified then this is used for binding. If the adapter is not 'Any' but the IP address is 'Any' then the first IP address assigned to this adapter is used. If both the adapter and IP address are specified then this IP address is used for binding only if this address is assigned to the specified adapter.

The list of 'Direct Connect Accept URIs' contains the list of URIs that can be used by other device to establish connection to this device. Because the current device can be behind a NAT router, this URI or URIs can differ then the IP address to which this device bound.

The third tab defines timeouts for different operations. 'Minimum Reconnect Time' defines the minimum delay between attempts to re-establish WebSocket connection to the Hub, in seconds. The BACnet/SC data link uses this timeout before the first attempt to re-establish connection and then it doubles the current retry timeout before each next attempt until it reaches 'Maximum Reconnect Time'. All subsequent attempts are performed with 'Maximum Reconnect Time'.

'Connect Wait Timeout' defines timeout to wait for a response to the Connect-Request message, in seconds. 'Disconnect Wait Timeout' defines timeout to wait for a response to the Disconnect-Request message, in seconds.

'Address Resolution Timeout' defines timeout to wait for a response to an Address-Resolution message, in seconds. 'Address Resolution Delay' defines the delay between Address-Resolution attempts, in seconds.

'Heartbeat Timeout' defines the interval between Heartbeat-Request messages, which are sent periodically to keep the established connection alive. 'Heartbeat Wait Timeout' defines timeout to wait for a Heartbeat-ACK message in seconds. If this parameter is not zero and Heartbeat-ACK has not been received in response to the Heartbeat-Request request within this timeout then the connection is closed automatically. Zero means infinite timeout, i.e. absence of Heartbeat-ACK message does not result in connection termination.

The BACnet/SC data link relies on the Libwebsockets and OpenSSL DLLs, which are installed together with the BACstac service.

Virtual Ports

A Virtual port just has a port ID, network number and “Attached” checkbox as parameters. Virtual ports are used by Gateway applications. Gateway applications attach to virtual ports using the port ID parameter, so it is important that the values in the routing table match the values used by applications. Only one Gateway application can be attached to a single virtual port.

Registry Entries

A number of BACstac protocol stack run-time parameters are stored in the Windows Registry. These values are created and initialized when the protocol stack is installed.

The Registry entries are located in:

HKEY_LOCAL_MACHINE\SOFTWARE\Cimetrics\BACstac

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\BACstac

The BACstac Service parameters (in ..\Services\BACstac) include:

Table 4-1. BACstac Service configuration parameters

Key	Default	Comments
Dl\Pdu_Pool_Size	200	PDU pool size This number specifies the maximum number of PDUs that can be held in all internal queues simultaneously. Its number should be more than the size of largest segmented window, otherwise segmentation may not work or work very slowly because of “lost” messages inside of the stack.
Dl\Socket_Buffer_Size	32k	size, in bytes, of the socket input and output buffer, used by BACnet/IP
Dl\Device_UUID	00000000-0000-0000-0000-000000000000	A unique device identifier. This value has to be initialized to a unique value during installation. Currently, the Device UUID is used only by the BACnet/SC data link.
Dl\SC\Allow_TLSv12	0	Allow to use TLS v1.2
Dl\SC\Log_Debug	0	Enable logging debug messages
Dl\SC\Log_Websock_Level	7	Log_Websock_Level is a bitmask from that contains LL_XXX flags from libwebsockets/include/libwebsockets/lws-logs.h
Nl\Ptp_Term_Time_Value	30	in seconds, used when PTP connection is initiated locally
Nl\Find_Dnet_Timeout	30 000	time in milliseconds to wait for I-Am-Router-To-Network in response to Who-Is-Router-To-Network sent by the BACstac, when it tries to locate a router for an unknown remote network
Nl\Connect_Ack_Timeout	1 000	time in ms, to wait after PTP connection is established, before sending any packet to it.

Key	Default	Comments
Nl\Connect_Timeout	120 000	time in milliseconds to wait for a PTP connection to be established
Nl\Ptp_Term_Time_Value	30	inactivity timeout for establishing a new PTP connection, in seconds. The value may be from 0 to 255; 0 means infinity
Nl\Ptp_Term_Delay	60 000	connection termination delay by I-Am-Router-To-Network, in milliseconds.
Nl\Ptp_Connect_Delay	1 000	Delay after receiving the I-Could-Be-Router-To-Network message before establishing a PTP connection to otherwise unreachable network, in milliseconds.
Nl\Ptp_Safety_Timeout	300 000	PTP safety timeout, in milliseconds.
Nl\Router_Busy_Timeout	30 000	Router-Busy-To-Network timeout, in milliseconds See ANSI/ASHRAE Standard 135-2020 or later, section 6.6.3.6 for details
Nl\Delay_PDU_On_Busy	120 000	the maximum delay time for APDUs to a busy network, in milliseconds. If the network is in the temporarily unreachable state for less than the specified value than packets from BACstac applications will be accepted and STATUS_OK is returned to the application. If the network remains in this state longer than the Delay_PDU_-On_Busy time, no new APDU to this network is accepted and the error DNET_BUSY is returned to the application. Zero value of this timeout has a special meaning: defer all messages to the busy network until the destination network becomes available. The default value for this option is 2 minutes. The meaning of this option has been changed in BACstac 6.3. Now it does not affect routing. Starting with BACstac 6.3, upon receiving a message whose destination is one of the temporarily unreachable networks, BACstac sends a Reject-Message-To-Network message with a reject reason of 2 to the originating node.
Nl\Fd_Reg_Timeout	30 000	time in milliseconds, to wait for BVLC-Result Success during Foreign Device registration. This option is deprecated in BACstac 4.3. Instead of it, "Bip\Num_Fd_Reg_Retries" should be used in the corresponding port entry.
Nl\Port_Table\Num_Entries		current number of routing table ports
Nl\Port_Table\EntryX*		routing table entries

Key	Default	Comments
Nl\Max_Pending_Xmt	150	maximum number of pending transmit packets When the target network is in search, then the PDU is stored in the pending queue until either the network is found or the search timeout expires.
Nl\Max_Known_Dnets	100	maximum number of known remote networks The size of the pool used to story all known remote networks.
Nl\Max_Dnets_In_Search	100	maximum number of remote networks in search simultaneously When the BACstac does not know the router for a remote network, it tries to locate it using a Who-Is-Router-To-Network message, and then waits up to Find_Dnet_Timeout (milliseconds) for an I-Am-Router message. This number specifies the maximum number of networks in search.
Nl\Hop_Decrement	1	hop counter decrement value when a packet is routed to another network (1..255)
Tsml\Max_Processes	10	maximum number of BACstac applications working simultaneously This number is limited only by available memory.
Tsml\Apdu_Max_Segs_-Present	1	To set max-segments-accepted fields in an APDU header. BACnet 2001 or later allows for a BACnet device to advertise the maximal number of segments that it can accept in the response. This functionality is enabled by default starting with BACstac 5.0. For better compatibility with old devices, which may be confused by this new parameter, you may want to change this value to zero.
Tsml\Bind_On_Start	0	Specifies whether all ports are should be bind on start (1) or later when an application is connected (0).
Tsml\ConnIpc	\\.\pipe\bacstac	Named pipe used for IPC connection between a BACstac application and the BACstac service. This parameter should not normally be changed by the user.

The modified value of port entries listed above will take effect after the BACstac service restarts.

The routing table port entries can be modified using the Protocol Properties dialog box.

In addition the library parameters used by each application include (the keys are repeated for Cli\ and Gtw\):

Table 4-2. Routing entries parameters

Key	Default	Comments
Network_Type		BACnet Network Type
Port		Port ID
Label		Port name for log messages
Dnet		Network number
Connected	1	Enabled to receive messages
Nic		Network Interface Card name
Bip*		BACnet/IP specific parameters
Bip\Bbmd	0	Enable BBMD functionality
Bip\Bdt		A space-separated list of all BDT entries. Each entry has the followng format: <ip>[/<subnet>][:<port>].
Bip\Fdt_Present	0	Enable Foreign Device Table
Bip\Max_Fdt_Entries	16	Maximum number of FDT entries
Bip\Foreign_Device	0	Work as a foreign device
Bip\Fd_Bbmd_Ip		IP address of the BBMD device to connect
Bip\Fd_Bbmd_Udp	47808	UDP port of the BBMD device to connect
Bip\Fd_Time_To_Live	1800	Time-to-live for registration, in seconds
Bip\Nat_Supported	0	indicates whether or not BACnet/IP NAT functionality as described in Clause J.7.8 is supported by this port. If it is is 1 (TRUE), the Network Port object has the BACnet_IP_NAT_Traversal and BACnet_IP_Global_Address properties, which has values defined below. If it is 0 (FALSE) then all following NAT related parameters are ignored.
Bip\Nat_Enabled	0	indicates whether or not this port is configured to operate in a NAT environment. Its value can be changed by writing the BACnet_IP_NAT_Traversal property in the Network port object.
Bip\Nat_Ip		If the BBMD is located behind an IP router that performs network address translation (NAT) then this field should contain the global IP address or hostname of the IP router
Bip\Nat_Udp	47808	The external UDP port that is configured on the router with NAT to be used for receiving BACnet/IP packets directed to this BBMD
Bip\Resolv_Timeout	3600	Expiration timeout for resolver in seconds; zero means infinite. The maximum possible value is about 1 million seconds (12 days).
Bip\Fd_Reg_Timeout	30 000	Time to wait before retrying the registration request, in milliseconds
Bip\Num_Fd_Reg_Attempts	0	number of attempts to register the Foreign Device, zero means infinity.
SC\Address_Resolution_Delay	30	Delay between Address-Resolution attempts, in seconds.
SC\Address_Resolution_Timeout	5	Timeout to wait for a response to an Address-Resolution message, in seconds.
SC\CA_Certificates		The folder that contains one or more Certificate Authority certificates of those CAs that are accepted as signers of peer's certificates.

Key	Default	Comments
SC\Connect_Wait_Timeout	10	Timeout to wait for a response to the Connect-Request message, in seconds.
SC\Direct_Connect_Accept_-Enable	1	Whether the node accepts WebSocket connections as BACnet/SC direct connections.
SC\Direct_Connect_Accept_URIs		This parameter contains a list of WebSocket URIs that can be used by other devices to establish a direct connect to this device. The format of URIs is defined in RFC 3986, and they are separated by a single space (X'20'). This parameter is used only if accepting of direct connect is enabled. If accepting of direct connect is enabled, this parameter must contain at least one WebSocket URI.
SC\Direct_Connect_Binding		This parameter specifies the IP address and TCP port to bind to. This parameter is used only if accepting of direct connect is enabled. This parameter is a combination of the binding address and the binding port separated by column (':'). For example: "127.0.0.1:4450" If the IP address and the Nic parameter are empty, it means to listen on all IP addresses (i.e. bind to INADDR_ANY). If IP address is specified and the Nic is parameter is empty, the data link binds to the specified address. If the IP address is empty and Nic is specified, the data link binds to the first IP address assigned to that Nic. If both parameters are non-empty then the IP address is only used if it is assigned to the nic interface, otherwise STATUS_BAD_CONFIG is returned.
SC\Direct_Connect_Initiate_-Enable	1	Whether the node supports initiating WebSocket connections as BACnet/SC direct connections.
SC\Auto_Population_Enable	0	Whether the SC\Direct_Connect_Accept_URIs should be populated with an automatically generated URIs using the IP addresses specified for the network interface.
SC\Disconnect_Wait_Timeout	10	Timeout to wait for a response to the Disconnect-Request message, in seconds.
SC\Failover_Hub_URI		The WebSocket URI of the failover BACnet/SC hub. It has the same format as Primary_Hub_URI.
SC\Heartbeat_Wait_Timeout	0	Timeout to wait for a Heartbeat-ACK message in seconds. If this parameter is not zero and Heartbeat-ACK was not received in response to the Heartbeat-Request request within this timeout then the connection is closed automatically. Zero means infinite timeout, i.e. absence of Heartbeat-ACK message does not result in connection termination.
SC\Heartbeat_Timeout	300	To keep the established connection alive, the initiating node sends Heartbeat-Request messages periodically with this period.

Key	Default	Comments
SC\Maximum_Reconnect_Time	60	Maximum delay between attempts to re-establish WebSocket connection to the Hub, in seconds. Standard recommends to set it to no more than 600 seconds (AB.6.1).
SC\Minimum_Reconnect_Time	10	Minimum delay between attempts to re-establish WebSocket connection to the Hub, in seconds. The BACnet/SC data link uses this timeout before the first attempt to re-establish connection and then it doubles the current retry timeout before each next attempt until it reaches the Maximum_Reconnect_Time. All subsequent attempts are performed with Maximum_Reconnect_Time.
SC\Operational_Certificate_File		The file that contains the public certificate of the node.
SC\Issuer_Certificate_File_1		The file that contains the first signing CA issuer certificate file the port accepts when it is establishing a secure connection.
SC\Issuer_Certificate_File_2		The file that contains the second signing CA issuer certificate file the port accepts when it is establishing a secure connection.
SC\Certificate_Signing_Request_File		The file that contains the PKCS #10 certificate signing request.
SC\Primary_Hub_URI		The WebSocket URI of the primary BACnet/SC hub. The URI should follow the "wss" URI scheme defined in in RFC 6455, Section 3, and it should contain at least the host name and port. For example: wss://server.example.com:4443/
SC\Private_Key		The file that contains the private key of the node.
SC\VMAC		Initial Virtual MAC of the BACnet/SC node. The VMAC should have 6 octets, which is written in the hex format and can be separated by either ’-’ or ’:’. For example: "010203040506" "01-02-03-04-05-06" "01:02:03:04:05:06" Value "FFFFFFFFFFFF" and "000000000000" are reserved and should not be used as VMAC. In case of VMAC collision the actual VMAC used by the device may differ from this parameter.

Each key in the above table has "Nl\Port_Table\EntryX\" prefix, where X is the routing entry number.

Table 4-3. BACstac Application configuration parameters

Key	Default	Comments
Srv\Cli_Tsm_Pool_Size	150	the maximum number of client TSMs, which determines the upper limit for the number of simultaneously concurrent requests that can be sent by one application
Srv\Srv_Tsm_Pool_Size	150	the maximum number of server TSMs, which determines the upper limit for the number of incoming requests that be processed simultaneously
Srv\Apdu_Length*	0	Maximum APDU length; 0 means the maximum supported value. For client and server applications, it is determined by the underlying data link. For gateway applications, 0 means 1476 (same as for BACnet/IP and BACnet/Ethernet). If the value specified is too large, then the maximum allowed value is used.
Srv\Apdu_Timeout*	6 000	in millseconds, time to wait for the response PDU

Key	Default	Comments
Srv\Num_Retries*	3	number of retries if request fails
Srv\Segmentation*	0	0=both, 1=xmit, 2=rcv, 3=none
Srv\Max_Seg_Accepted*	32	maximum number of segments that can be received in a segmented message
Srv\Max_Seg_To_Send	0	Maximum number of segments that can be transmitted in a request 0 means the same value as for Max_Seg_Accepted
Srv\Max_Process_IDs	16	Maximum number of process identifiers that can be registered by one application
Srv\Seg_Window_Size*	16	Window size used for segmented requests and responses
Srv\Seg_Timeout*	5 000	APDU_Segmented_Timeout, in milliseconds
Srv\All_Process_IDs	0	Pass a confirmed request to the hook even if its ProcessID was not registered by the program: 0 — do not pass; 1 — pass all

Each set of keys stands for the corresponding type of an application. If changed, the new values take effect immediately after restarting the application for which the change was made.

Parameters marked with a star are default values for BACnet APDU Properties. These parameters:

- should be network-visible as properties in the Device object for a host
- can be queried and set by an application at runtime
- can be overridden on a per-request basis using the “raw” API

To send a larger segmented BACnet request (such as write file transfer request), increase Max_Seg_To_Send; to receive larger BACnet responses (such as read file transfer response), increase Max_Seg_Accepted. Increasing the Seg-Window-Size may speed up the transfer by reducing the number of segment acknowledgments required.

If incoming or outgoing packets are being dropped, increase DL\Socket-Buffer-Size and, if necessary, Tsml\PDU-Pool-Size (see below for the case of peak traffic handling). It may also be necessary to reduce the level of application processing done before resources are freed, such as in I-Am hooks.

If the application works through a slow data link, the default configuration may not work properly. You may need to reduce the segmented window size and/or increase the segmented and total TSM timeout.

If large packet traffic peaks can be expected on your network you can try to experiment with the parameters to determine the most suitable configuration for handling this problem. To begin with, you can configure all of the significant parameters (they are TSML/PDU_Pool_Size, */Cli_Tsm_Pool_Size, */Srv_Tsm_Pool_Size) to the number of packets expected in the traffic peak for a time unit (e.g. 250 packets for 0.5 second). If needed, some of the parameters can also be decreased. For example, if you prefer to save memory resources you can try to decrease the TSML/PDU_Pool_Size parameter, which consumes approximately 1.5kB for each PDU.

Appendix A. Object Database Default Sizes

The BACstac Object Database allocates fixed space for each property stored in the database. For some properties this size is defined by the maximum size allowed by the BACnet standard, but many BACnet data types (such as lists, arrays, character strings, etc) do not have any limit on how long they can be, so the BACstac has reasonable default sizes for those properties. Automatic properties (whose value is calculated based on other properties) are not stored in the object database, so the described limits are not applied to them.

The default size for non-array properties is determined by the property's data type. For array properties, space is allocated for each array item separately using the same rules as for scalar properties. Table A-1 describes the default size for scalar and list properties based on their data type.

> **Note:** A BACstac application can overwrite the default size during initialization, and specify how much space should be allocated for each property based on its property identifier and object type.

Table A-1. The default property sizes in the Object Database

Data Type	Scalar Size, in bytes	List Size, in bytes
AnyPrimitive	10	100
BACnetAccessAuthenticationFactorDisable	3	60
BACnetAccessCredentialDisable	3	60
BACnetAccessCredentialDisableReason	3	60
BACnetAccessEvent	3	60
BACnetAccessPassbackMode	2	40
BACnetAccessRule	39	780
BACnetAccessThreatLevel	2	40
BACnetAccessUserType	3	60
BACnetAccessZoneOccupancyState	3	60
BACnetAccumulatorRecord	25	500
BACnetAcknowledgeAlarmInfo	17	340
BACnetAction	2	40
BACnetActionCommand	250	2500
BACnetActionList	250	2500
BACnetAddress	12	240
BACnetAddressBinding	17	500
BACnetAssignedAccessRights	14	280
BACnetAssignedLandingCalls	82	328
BACnetAuditLevel	2	40
BACnetAuditLogRecord	50	0
BACnetAuditOperation	4	80
BACnetAuditOperationFlags	11	220
BACnetAuthenticationFactor	250	2500

Data Type	Scalar Size, in bytes	List Size, in bytes
BACnetAuthenticationFactorFormat	8	160
BACnetAuthenticationFactorType	2	40
BACnetAuthenticationPolicy	250	2500
BACnetAuthenticationStatus	2	40
BACnetAuthorizationExemption	2	40
BACnetAuthorizationMode	3	60
BACnetBackupState	2	40
BACnetBDTEntry	114	500
BACnetBinaryLightingPV	2	40
BACnetBinaryPV	2	40
BACnetCalendarEntry	12	500
BACnetChannelValue	104	500
BACnetClientCOV	5	100
BACnetColorOperation	2	40
BACnetColorOperationInProgress	2	40
BACnetColorTransition	2	40
BACnetCOVMultipleSubscription	50	1000
BACnetCOVSubscription	51	500
BACnetCredentialAuthenticationFactor	250	2500
BACnetDailySchedule	500	2500
BACnetDateRange	10	200
BACnetDateTime	10	200
BACnetDestination	37	500
BACnetDeviceObjectReference	10	500
BACnetDeviceStatus	3	60
BACnetDevObjPropReference	19	500
BACnetDevObjPropValue	48	480
BACnetDoorAlarmState	3	500
BACnetDoorSecuredStatus	2	40
BACnetDoorStatus	3	60
BACnetDoorValue	2	40
BACnetEngineeringUnits	3	60
BACnetEscalatorFault	3	60
BACnetEscalatorMode	3	60
BACnetEscalatorOperationDirection	3	60
BACnetEventLogRecord	50	0
BACnetEventNotificationSubscription	28	560
BACnetEventParameter	500	2500
BACnetEventState	3	60
BACnetEventTransitionBits	3	60

Data Type	Scalar Size, in bytes	List Size, in bytes
BACnetEventType	3	60
BACnetFaultParameter	500	2500
BACnetFaultType	2	40
BACnetFDTEntry	14	140
BACnetFileAccessMethod	2	40
BACnetHealth	129	600
BACnetHostAddress	104	500
BACnetHostNPort	109	500
BACnetIPMode	2	40
BACnetLandingCallStatus	108	500
BACnetLandingDoorStatus	82	328
BACnetLifeSafetyMode	3	500
BACnetLifeSafetyOperation	3	60
BACnetLifeSafetyState	3	500
BACnetLiftCarCallList	42	168
BACnetLifeSafetyOperationInfo	8	16
BACnetLiftCarDirection	3	60
BACnetLiftCarDoorCommand	2	40
BACnetLiftCarDriveStatus	3	60
BACnetLiftCarMode	3	60
BACnetLiftFault	3	60
BACnetLiftGroupMode	2	40
BACnetLightingCommand	25	250
BACnetLightingInProgress	2	40
BACnetLightingTransition	2	40
BACnetLimitEnable	3	60
BACnetLockStatus	2	40
BACnetLoggingType	2	40
BACnetLogMultipleRecord	50	0
BACnetLogRecord	50	0
BACnetMaintenance	3	60
BACnetNameValue	208	2080
BACnetNameValueCollection	500	2000
BACnetNetworkNumberQuality	2	40
BACnetNetworkPortCommand	2	40
BACnetNetworkSecurityPolicy	4	80
BACnetNetworkType	2	40
BACnetNodeType	2	40
BACnetNotifyType	2	40
BACnetObjectIdentifier	5	100

Data Type	Scalar Size, in bytes	List Size, in bytes
BACnetObjectPropertyReference	14	280
BACnetObjectType	3	60
BACnetObjectTypesSupported	19	380
BACnetObjectSelector	5	100
BACnetOptionalBinaryPV	2	40
BACnetOptionalCharacterString	104	500
BACnetOptionalPriorityFilter	4	80
BACnetOptionalReal	5	100
BACnetOptionalUnsigned	5	100
BACnetPolarity	2	40
BACnetPortPermission	4	80
BACnetPrescale	10	200
BACnetPriorityFilter	4	80
BACnetProcessIdSelection	5	100
BACnetProgramError	3	60
BACnetProgramRequest	2	40
BACnetProgramState	2	40
BACnetPropertyReference	9	90
BACnetPropertyAccessResult	48	480
BACnetPropertyStates	5	100
BACnetPropertyValue	250	1250
BACnetProtocolLevel	2	40
BACnetRecipient	14	500
BACnetRelationship	3	60
BACnetReliability	3	60
BACnetRestartReason	2	40
BACnetRouterEntry	16	500
BACnetScale	5	100
BACnetSCConnectionState	2	40
BACnetSCDirectConnection	377	1500
BACnetSCFailedConnectionRequest	259	1300
BACnetSCHubConnection	136	700
BACnetSCHubConnectorState	2	40
BACnetSCHubFunctionConnection	273	1400
BACnetSecurityKeySet	68	136
BACnetSecurityLevel	2	40
BACnetSecurityPolicy	2	40
BACnetSegmentation	2	40
BACnetServicesSupported	9	180
BACnetSessionKey	22	440

Data Type	Scalar Size, in bytes	List Size, in bytes
BACnetSetpointReference	16	320
BACnetShedLevel	5	100
BACnetShedState	2	40
BACnetSilencedState	3	60
BACnetSpecialEvent	250	1250
BACnetStageLimitValue	14	112
BACnetStatusFlags	3	60
BACnetSuccessFilter	2	40
BACnetTimerState	2	40
BACnetTimerStateChangeValue	104	728
BACnetTimerTransition	2	40
BACnetTimeStamp	12	240
BACnetTimeValue	15	150
BACnetVMACEntry	28	500
BACnetValueSource	14	500
BACnetVTClass	3	60
BACnetVTSession	16	320
BACnetWeekNDay	4	80
BACnetWriteStatus	2	40
Bit String	5	100
Boolean	1	20
Character String	104	500
Date	5	100
Double	10	200
Enumeration	5	100
InstanceNumber	4	80
Integer	5	100
Null	1	20
Octet String	100	500
ReadAccessResult	200	4000
ReadAccessSpecification	97	4000
Real	5	100
Time	5	100
Unsigned	5	100
Unsigned8	2	40
Unsigned16	3	60
Unsigned32	5	100

Note: The list size for BACnetEventLogRecord, BACnetLogMultipleRecord and BACnetLogRecord is 0, because the list of these types is used to define the Log_Buffer property, which usually is not stored in the BACstac database.

The space reserved for BACnetPriorityArray is equal to 16 elements of BACnetPriorityValue, where the size of the latter is determined by the data type of the present-value property.

Array properties can be two kinds: fixed-size and variable-size arrays. Fixed-size array properties cannot be resized and the number of elements in them is prescribed by the BACnet standard. Variable-size arrays can have any number of elements accordingly to the standard. The BACstac allocates a fixed size for each property. If the application has not provided its own property description with its own array sizes, the default values as defined in Table A-2 are used.

Table A-2. Default for the maximum number of elements in variable-size array properties.

Object Types	Properties	Number Of Elements
<ANY>	Tags	10
Access_Credential	Assigned_Access_Rights	16
Access_Credential	Authentication_Factors	16
Access_Door	Door_Members	16
Access_Point	Authentication_Policy_List, Authentication_Policy_Names	16
Access_Point	Access_Doors	16
Access_Rights	Negative_Access_Rules	16
Access_Rights	Positive_Access_Rules	16
BitString_Value	Alarm_Values	16
BitString_Value	Bit_Text	24
Channel	List_Of_Object_Property_References, Execution_Delay, Control_Groups	32
CharacterString_Value	Alarm_Values, Fault_Values	40
Command	Action, Action_Text	40
Credential Data Input	Supported_Formats, Supported_Format_Classes	16
Device	Structured_Object_List	16
Device	Configuration_Files	16
Device	Subordinate_Proxy_Enable	16
Device	Auto_Subordinate_Discovery	16
Elevator_Group	Group_Members	32
Global_Group	Group_Members, Group_Member_Names, Present_Value	32
Lift	Floor_Text	64
Lift	Car_Door_Text, Assigned_Landing_Calls, Making_Car_Call, Registered_Car_Call, Car_Door_Status, Car_Door_Command, Landing_Door_Status.	4
Load_Control	Shed_Levels, Shed_Level_Descriptions	40
Multistate_Input, Multistate_Output, Multistate_Value	State_Text	40
Network_Port	Link_Speeds	20
Network_Port	IP_DNS_Server, IPv6_DNS_Server	16

Object Types	Properties	Number Of Elements
Network_Port	SC_Hub_Function_Accept_URIs, SC_Direct_Connect_Accept_URIs	10
Network_Security	Network_Access_Security_Policies	20
Notification_Forwarder	Port_Filter	20
Schedule	Exception_Schedule	40
Structured_View	Subordinate_List, Subordinate_Annotations	16
Structured_View	Subordinate_Tags, Subordinate_Node_Types, Subordinate_Relationships	4
Trendlog_Multiple	Log_Device_Object_Property	16

Note: The Object_List property in the Device object is always automatic, therefore the number of elements is determined by the number of objects in the BACstac database.

Table A-3. Sample Data Type Instances

Data Type	Min..Max Size, bytes	Instance	Instance Size, bytes
BACnetActionCommand	12..unlimited	deviceIdentifier is not present, propertyArrayIndex is not present, propertyValue is Real data type, priority is not present, postDelay is not present	16
Unsigned	2..5	255 < value < 65536	3
BACnetCalendarEntry	4..12	BACnetDateRange choice	12
BACnetCalendarEntry	4..12	BACnetWeekNDay choice	4
BACnetAddressBinding	9..18	network-number > 256, 6-byte MAC address	18
BACnetEventParameter	6..unlimited	out-of-range choice	24
BACnetSpecialEvent	9..unlimited	period is BACnetDateRange data type listOfTimeValues contains 16 Real values	176
ReadAccessSpecification	9..unlimited	10-item specifications, propertyArrayIndex is not present	27
BACnetDeviceObjectProperty-Reference	7..19	propertyArrayIndex is not present, DeviceIdentifier is not present	7
BACnetDeviceObjectProperty-Reference	7..19	propertyArrayIndex is not present, DeviceIdentifier is present	12
BACnetLogRecord	15..unlimited	logDatum is Real data type, statusFlags are present	22
BACnetEventLogRecord	17..unlimited	logDatum is BACnetLogStatus data type	17
BACnetLogMultipleRecord	17..unlimited	logData is sequence of 10 Boolean values	26

Data Type	Min..Max Size, bytes	Instance	Instance Size, bytes
ReadAccessResult	5..unlimited	10-item results, propertyArrayIndex is not present, propertyValue is Real data type	97
BACnetDestination	23..37	255 < processIdentifier < 65536, recipient is BACnetObjectIdentifier choice	27
BACnetRecipient	2..13	BACnetObjectIdentifier choice	5
BACnetRecipient	2..13	BACnetAddress choice, network-number > 255, MAC address 6 bytes	13
BACnetTimeValue	8..unlimited	value is Real data type	12
BACnetCOVSubscription	21..45	Object ID choice, Process ID < 256, Monitored Property Reference index is not present, Time Remaining is not present, COV Increment is not present	21
BACnetCOVSubscription	21..45	Address choice (DNET ≥ 256, MAC is 6 bytes), Process ID < 65536, Monitored Property Reference index is not present, Time Remaining < 65536, COV Increment is not present	33
BACnetAuthenticationPolicy	6..unlimited	Two policy items (with DeviceIdentifier, and index < 256), timeout < 256	34
BACnetAuthenticationFactor	5..unlimited	wiegand26 with format-class < 256	8
BACnetAccessRule	6..39	timeRange is specified with DeviceIdentifier, propertyIdentifier < 256, and without propertyArrayIndex; location is specified with DeviceIdentifier	32
BACnetAssignedAccessRights	9..14	DeviceIdentifier is not present	9
BACnetAuthenticationFactor-Format	2..8	vendor-id < 256; vendor-format < 256	6
BACnetCredential-AuthenticationFactor	9..unlimited	disable < 256, format-type = wiegand26, format-class < 256	12
BACnetPropertyAccessResult	10..unlimited	propertyIdentifier = Present_Value, no arrayIndex, no deviceIdentifier, propertyValue is the Real data type	14