BACstac/Win v7.8

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

BACstac/Win v7.8Installation Guide
by Cimetrics, Inc.
Document Version: 1.0 Edition
Copyright © 1999-2022 Cimetrics, Inc.

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 135-2016 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 WinPCap driver to use BACnet over Ethernet
- Microsoft Visual Studio 2013 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 WinPCap interface. The WinPCap driver can be downloaded here (http://www.winpcap.org/install/).

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

The BACnet/PTP data link layer uses an unoccupied serial port. If your PTP DL is modem-configured, it will be capable of dialing and receiving calls.

You can have any number of BACnet/Ethernet, BACnet/IP, Virtual, and BACnet/PTP ports. A valid BACstac configuration should have at least one physical port (i.e. a configuration that consists only of Virtual and/or PTP ports is not allowed).

Protocol Stack Installation

If you have BACstac for Projects, first of all you need to install the Sentinel driver for your dongle key, which can be downloaded here: http://www.safenet-inc.com/support-downloads/sentinel-drivers. During installation of BACstac, do not forget to insert your dongle key into the computer.

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 WinPCap 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 WinPCap driver, which can be download from http://www.winpcap.org/install/.

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 WinPCap driver must be installed on the system. You can download it here (http://www.winpcap.org/install/).

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 preferable 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/PTP Ports

The BACnet/PTP Properties dialog box is unique and although it has a port ID field, it doesn't have a network number field. This is because PTP ports are not regular network ports, but rather they are connections between half-routers. When a connection is established the routers on each end of a PTP link exchange lists of the network numbers accessible to them. The PTP Data Link Layer and Network Layer are described in Clauses 6 and 10 of the BACnet Standard, ANSI/ASHRAE 135-2016 or later. The BACnet/PTP Properties dialog box has edit fields to specify the port ID and an RS-232 port. The Configure button will call up a dialog box to modify the port settings for the RS-232 port. These settings should match the settings of the peer PTP device.

The next field in the dialog box is used to add "could be" network numbers using the Add/Remove buttons. A PTP half-router is required to respond to some requests for specific network numbers by saying I-Could-Be-Router-To-Network if it knows it can become a connection to the requested network by making a PTP link. The performance index is a parameter included in the I-Could-Be-Router-To-Network reply, to help the client decide which half-router is the best one for the connection. A low value in this field indicates a higher performance index, or desirable connection. The client will then send that half-router an Establish-Connection-To-Network request. The BACstac protocol network layer will attempt to establish a connection automatically on behalf of its applications (for both local and remote PTP half-routers).

When peer PTP half-routers attempt to establish a connection, the calling half-router sends a password. You can specify the password sent by this PTP port, in addition to listing passwords that this PTP port will accept.

An important field in this dialog box is the Permanent check box. This should be checked for hard-wired connections to PTP devices, since it will cause the BACstac Service to attempt to establish the PTP link at start-up. If this box is not checked, the PTP port will wait for the other device to initiate the connection, or the PTP port will wait for a client application to request the connection.

Both cable connection and phone-line connection are types of physical connections. The physical one is required for the logical connection discussed in Clause 10 of the BACnet Standard. The logical connection can be established only if the physical one is established already. Whenever a logical connection establishment is initiated by this port, physical connection is established automatically. If the logical connection is terminated the physical one is terminated too and vice versa.

To use the dialing ability of a PTP device effectively, the modem parameters should be customized. For this purpose, check the Modem checkbox of the new PTP Port Properties Dialog box. If you leave this box unchecked, the PTP port assumes that you will use a direct serial connection without enabling any Dial properties. Then click the "Dial…" button to enter the "Dial and Modem Properties" dialog box. This dialog contains the following fields:

- Phone numbers: "number-list"

  List of space-separated phone numbers to dial. The modem will attempt to dial all numbers in sequence. Each phone number is formatted in a modem-ready way (may include ',', 'w', etc.: see your modem manual). Example:

  "5551234 t8w095p,777-8888"
- Retries: n

  Number of times to dial through the entire phone list before giving up. Zero means don't dial. The default value is 1. -1 means infinitely (in fact, until the overall timeout expires). If this value is not present, the number of retries is -1.
- Interval: n sec

  Interval, in seconds, between two consecutive dial attempts. By default, the zero value is assumed.
- Rings: n

  Number of rings to allow before answering an incoming call. Default value is 1.
- Initialization string: "how-to-init-modem"

Character sequence used once to initialize modem. Defaults to the following string:

" atx4q0v1s0=0&d2&b1&c1"

Modem options set in this example are crucial for the BACstac to work. Please make sure that the modem options are set properly — either in the initialization string parameter or by some other means (i.e. by storing in an appropriate internal modem configuration). Description of each option follows. Some of the options may be specific for 3COM (formerly US Robotics) modems but analogs for other modems should exist.

x4: set modem result (error) code display filter

q0: enable modem result (error) codes

v1: display modem result (error) codes in verbal form

s0=0: disable modem autoanswer

&d2: use default DTR handling

&b1: Fixed serial port rate

&c1: Normal CD operations
- Dial prefix: "dial-string"

Character sequence to prepend to a telephone number before sending it to modem.

The default is "ATD". Example:

"atdp" — force pulse dialing.

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 list of 'Direct Connect VMACs' contains the list of VMACs to which the current device should establish a direct connection will be established if needed.

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.
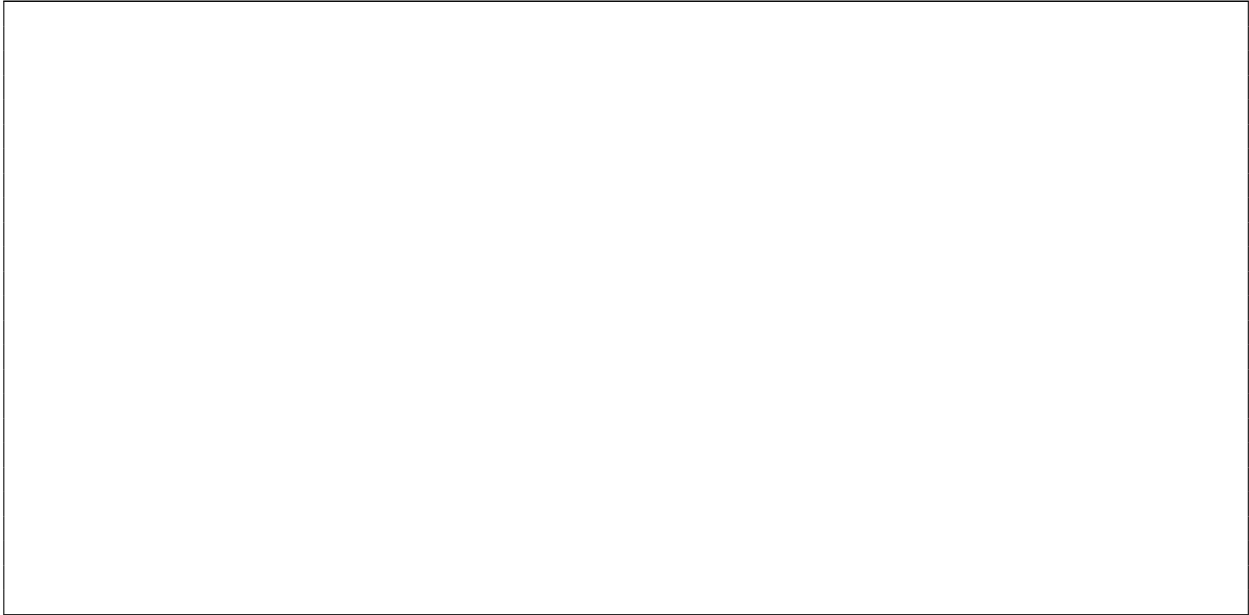
The Registry entries are located in:

HKEY_LOCAL_MACHINE\SOFTWARE\Cimetrics\BACstac

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\BACstac

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

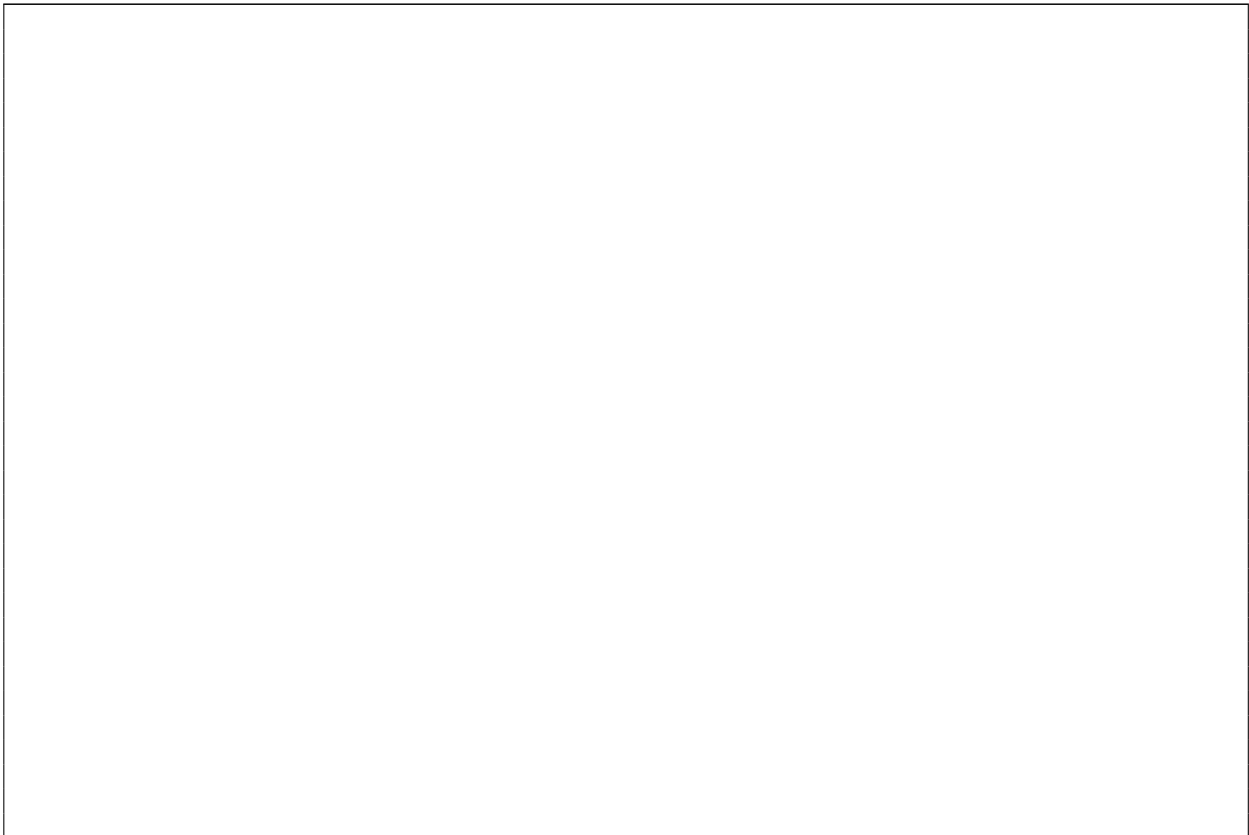**Table 4-1. BACstac Service configuration parameters**

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

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

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, such as PTP, 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. Also you may need reduce APDU length, because it is smaller for PTP than for Ethernet or BACnet/IP. However, reducing APDU length in the registry is not always the best solution, because it will decrease APDU length for all requests regardless of their destination and make it impossible for the program to receive an APDU larger than the specified value. Alternatively, APDU length may be decreased on a per-request basis using APDU properties in the "raw" API.

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.
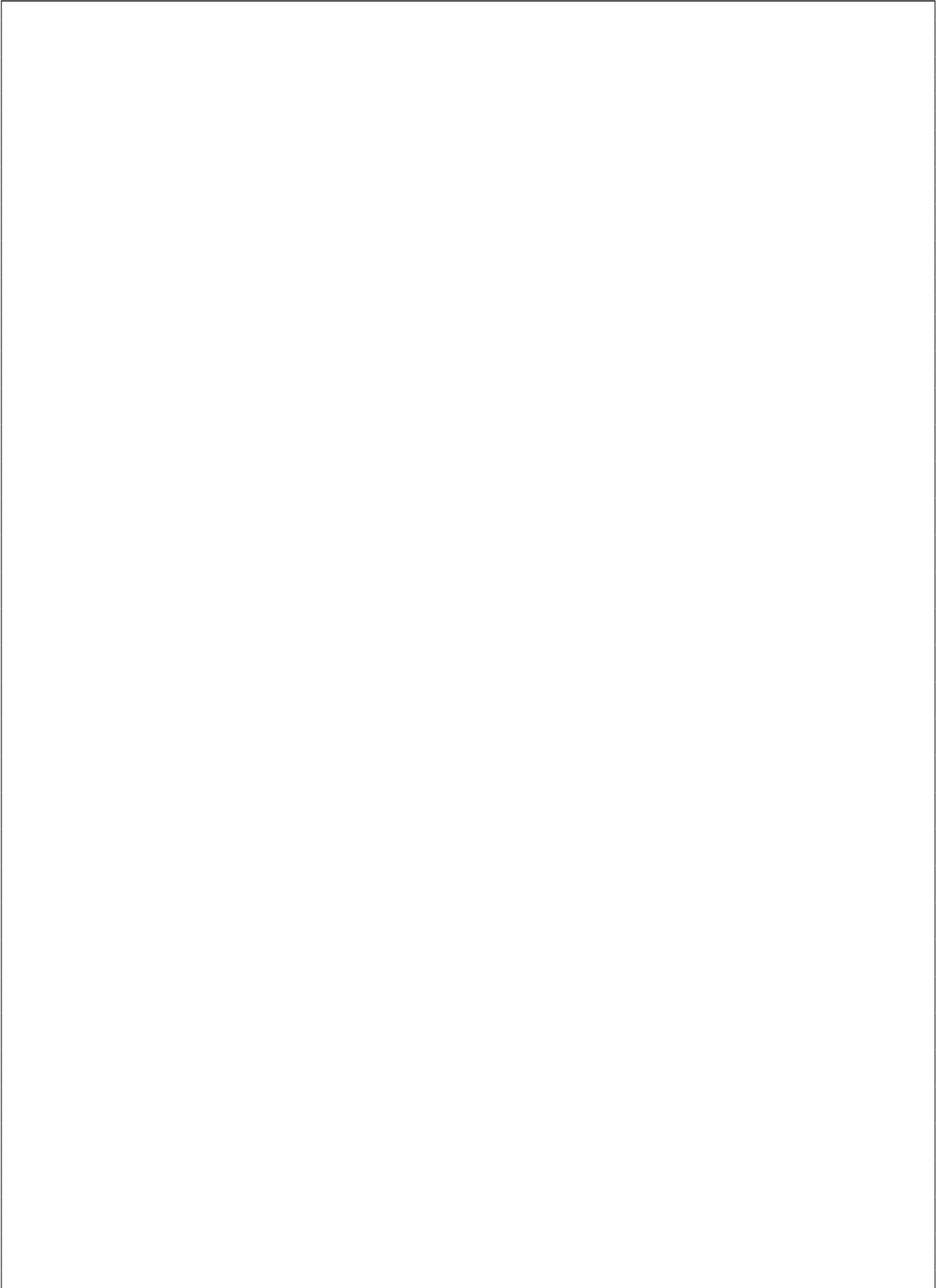
Appendix A. Object Database Default Sizes

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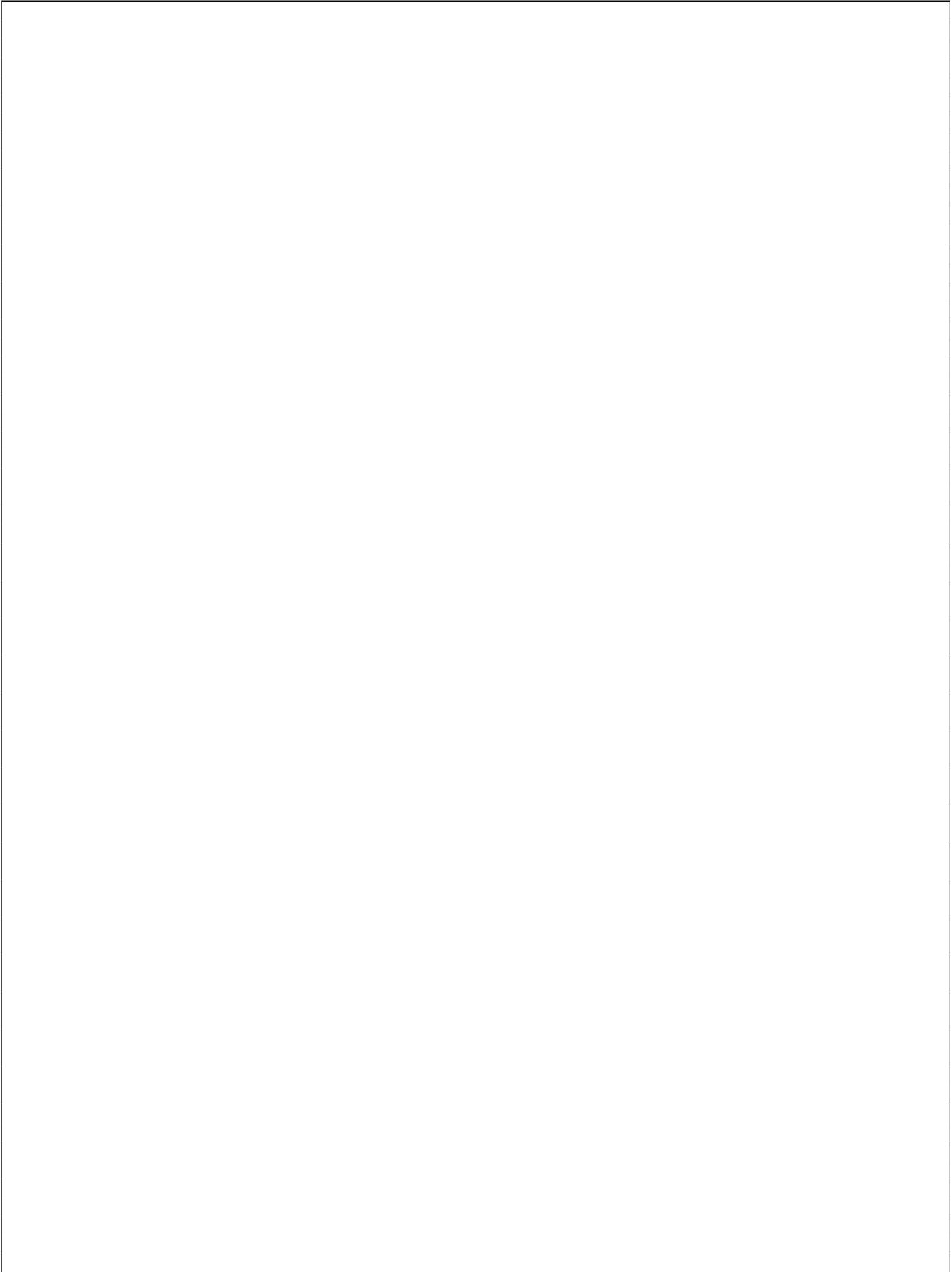
The default size for non-array properties is determined by the property's data type. For array properties, space is allocated for each array item separately using the same rules as for scalar properties. Table A-1 describes the default size for scalar and list properties based on their data type.

> **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

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.

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