BACset Users Guide

14-May-2007

Introduction

This document provides a Users Guide for the BACset Configuration Tool, which is used to configure and test Carel pCOWeb/pCOnet devices.

This document assumes you are familiar with BACnet® and BACnet terminology and the use of Microsoft® Windows®.

BACnet Requirements

BACnet MAC Layer Support

BACset supports all of the following BACnet Media Access Control (MAC) layers

- BACnet ISO 8802-3 "Ethernet"
- BACnet/IP
- BACnet Master Slave/Token Passing (MS/TP) over EIA-485

BACset as a BACnet Device

BACset, whose primary function is to act as a BACnet client, also acts responsibly in any BACnet network. As such it responds to readProperty requests on a single Object, that being its own Device Object (i.e. as it relates to other BACnet clients, BACset supports the DS-RP-B BIBB). By default, BACset is assigned the device instance 77077. This can be changed using the "BACdoor OEM Client" configuration dialog described in the **Configuring BACset** section of this document.

Windows and PC Requirements

Windows Requirements

This tool is compatible with Windows 2000, Windows 2003, Windows XP (both Home and Pro editions) and Windows Vista.

PC Requirements

This tool requires:

- an ISO 8802-3 "Ethernet" adapter be installed in your PC if you intend to use BACnet Ethernet or BACnet/IP
- the standard Windows TCP/IP protocol be enabled if you intend to use BACnet/IP
- an EIA-485 interface if you intend to use MS/TP. An EIA-485 interface can be achieved by one of following means:
    - by connecting an external EIA-232 to EIA-485 converter to an existing PC serial port
    - by inserting an internal EIA-485 PC adapter into your PC

Installing BACMAC2K

If you intend to use BACnet Ethernet, you must install the BACMAC2K BACnet MAC layer driver. The installation procedure of BACMAC2K is as follows:

- Using the Windows **Control Panel** click **Network Connections**
- Right click **Local Area Connections** for your Ethernet Adapter and then click **Properties**
- In the "Local Area Connection Properties" Window click **Install**
- In the "Select Network Component Type" Window select **Protocol** and then click **Add**
- In the "Select Network Protocol" Window click **Have Disk…**
- In the "Install from Disk" Window Use **Browse…** to navigate to the folder where you installed BACset (normally *Program Files\Carel\BACset*) and find the *drivers* folder. In the *drivers* folder select **bacmac2k.inf** and click **Open**
- In the "Install from Disk" Window click **OK**
- In the "Select Network Protocol" Window click **OK**
- **BACMAC2K BACnet MAC Layer Protocol** should appear in the list of **protocols** in the "Local Area Connection Properties" Window. *Make sure it's checked*.
- In the "Local Area Connection Properties" Window click **OK**

If you do not intend to use BACnet Ethernet, you do not have to install BACMAC2K. If you do not install BACMAC2K and you try to use BACnet Ethernet, you'll be warned with a popup message.

Using BACset

Introduction

BACset is an easy to use, Windows-based tool for configuring and testing the Carel pCOWeb and pCOnet devices. It has an intimate knowledge of the pCOWeb/pCOnet's features, which are detailed in the ***pCOWeb/pCOnet Features*** section of this document, i.e. how many and what types of objects the pCOWeb/pCOnet has, which properties are supported and which non-BACnet system functions are supported. This speeds up the configuration process, by eliminating the need for the auto-discovery of objects and properties.

Display Components

When BACset is started, a popup dialog is displayed for you to select the BACnet MAC layer type. After you have selected a BACnet MAC layer type, two or three dialogs are activated.

The main "BACset Configuration" window, which is visible on the display screen, contains the following tabs:

- BACnet Device
- BACnet Objects
- Notify Classes
- Test
- Database
- System
- Plugin

The “BACdoor OEM Client Status” Windows is minimized into the Windows Task Bar at the bottom of your display screen and is indicated by the  icon. Clicking on the icon makes the Status window visible.

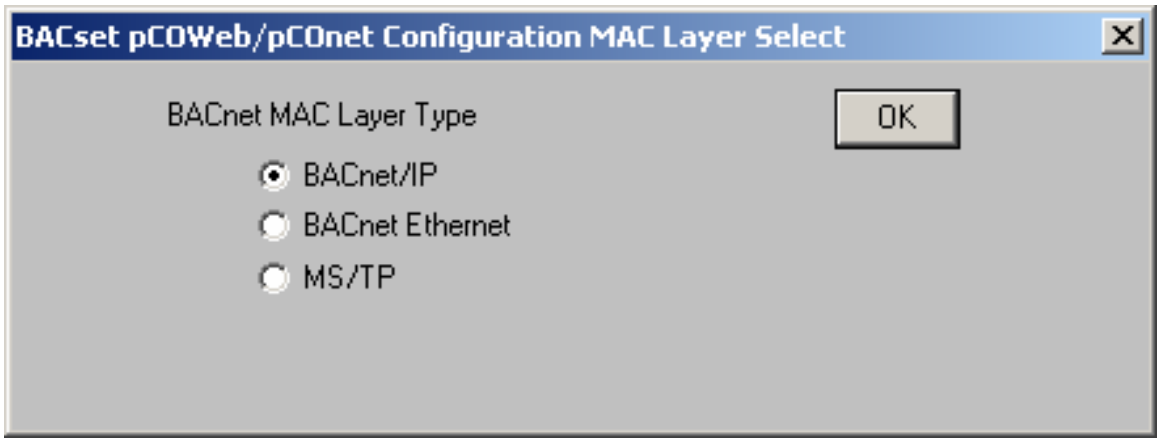
If you select the MS/TP MAC layer, the additional “MS/TP MAC Layer Status” Window is minimized into the Windows Tool Bar at the bottom of your display screen and is indicated by the  icon. Clicking on the icon makes the Status window visible.

The four windows are described in the following sections.

MAC Layer Selection Window

General

When you startup BACset, the “BACnet MAC Layer Select” Window is displayed from which you must select the type of BACnet MAC layer to use for the current session.



Using BACnet/IP

If you select BACnet/IP and your PC has been configured for more than one IP address, you may have to select which of the IP addresses to use. This is accomplished using the “BACdoor OEM Client” configuration dialog described in the **Configuring BACset** section of this document.

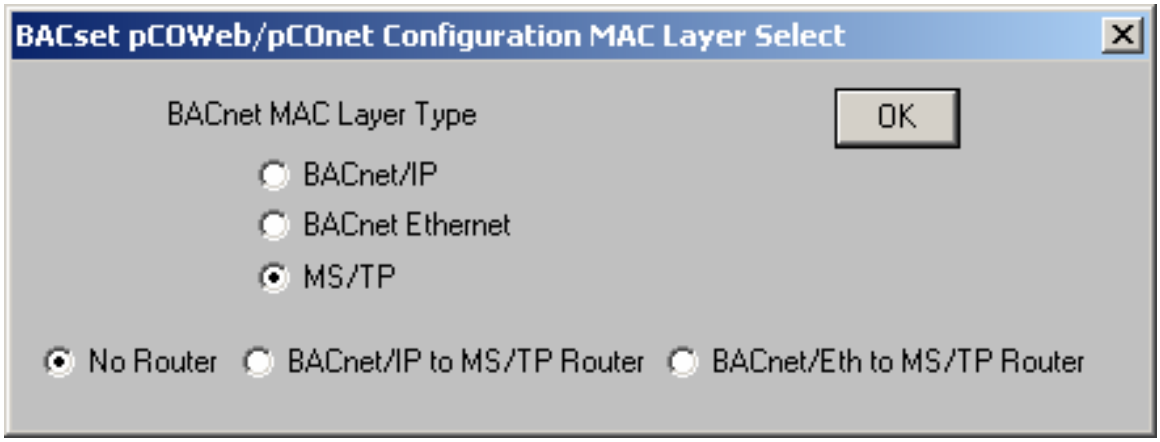
If you have only one IP address configured, which is the normal case, no selection is necessary, and BACset uses that one IP address.

Using BACnet Ethernet

If you have not installed BACMAC2K, you will get a popup Window stating “The BACMAC driver is not installed or it is not bound to a NIC”. To remedy this you must follow the installation procedure detailed previously in the **Installing BACMAC2K** section of this document.

Using BACnet MS/TP

When you select the MS/TP MAC layer, you are further prompted if the pCOnet you are configuring is connected to BACset through a BACnet/IP or BACnet Ethernet Router.



Normally BACset assumes that you are connected on the same type of MAC layer network as the type of PCOWeb/pCO net you are configuring. If that is the case you should select "No Router". If the pCOnet is however connected through a router, selecting that type of NMAC layer will result in erroneous communications so you must one of the other two choices in those cases.

If you select No Router, BACset is configured for MS/TP operation on COM1 at 38400 baud, with an MS/TP station address=1, MaxMaster=127 and MaxInfoFrames=10 by default. These parameters can be changed using the “BACdoor OEM Client” configuration dialog described in the **Configuring BACset** section of this document. *These default parameters are selected in order to operate with pCOnet devices that have Factory configurations (i.e station address=0 and MaxMaster=127) and still allow for changing the station address of the pCOnet to any value between 0 and 127. The MaxMaster value should be lowered for more efficient operation if it is known that the pCOnet devices will never exceed a station address of something less than 127.*

Automatic Selection of the MAC Layer

You can configure BACset to bypass the “MAC Layer Selection” Window and always start up with the same BACnet MAC layer type. The procedure to do so is:

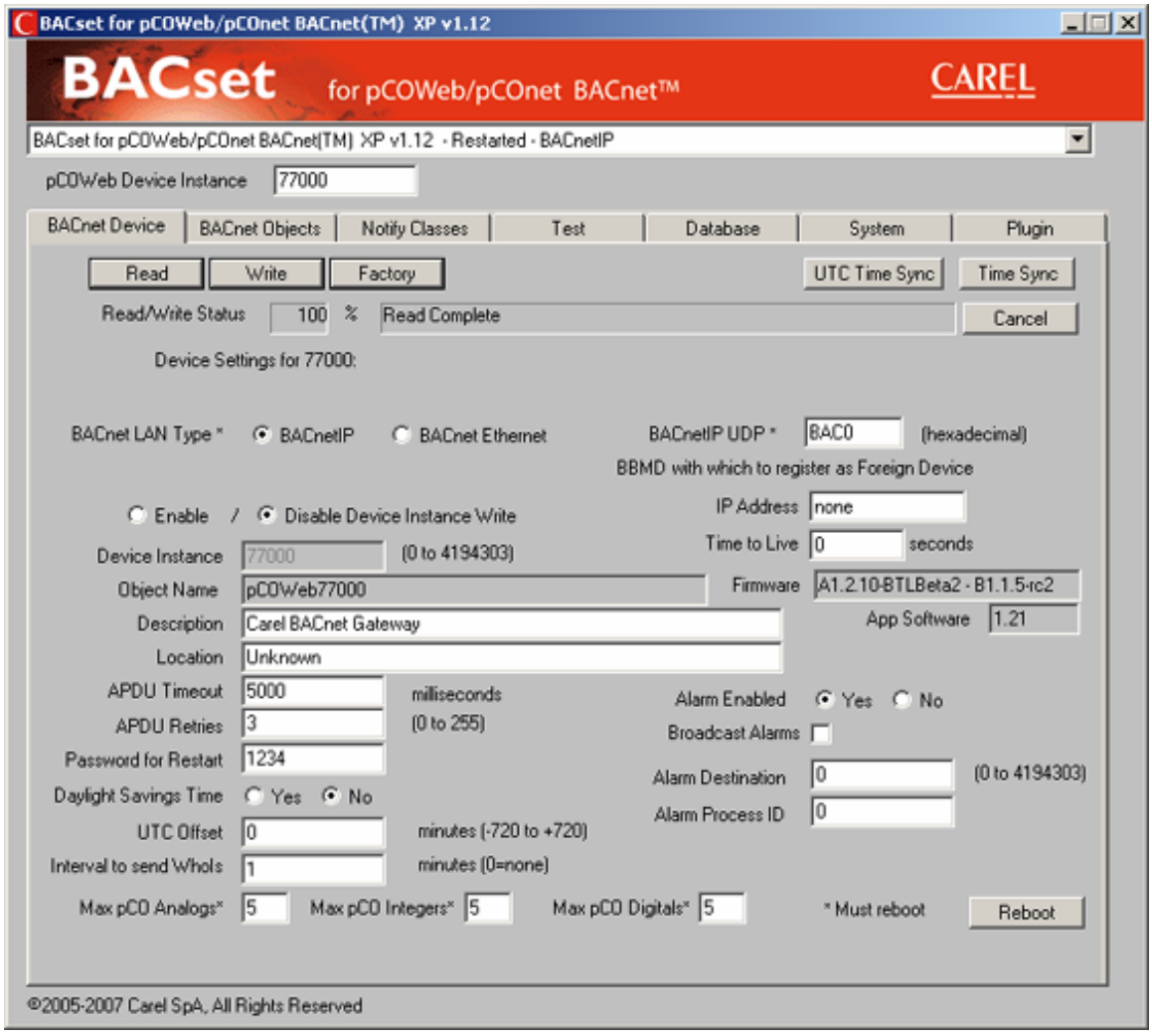
- Using a text editor such as Notepad, create a file named BACSET.INI in the Windows or WINNT folder. Note, if a file named BACSET.INI already exists, this step should be skipped
- Place the following line as the first line of the file
    - **[General]**
- Place a second line of any of the following
    - **MACtype=MSTP**
    - **MACtype=BIP**
    - **MACtype=Eth**
- If you want to connect to a pCOnet through a router and MAVtype=MSTP, then add another line of any of the following
    - **Router=BIP**
    - **Router=Eth**
- Save and close the file

BACset Configuration Window

General

After you have selected the BACnet MAC Layer, the main “BACset Configuration” window is displayed. Each of the five Tabs is described in the following sections.

BACnet Device Tab



This Tab is used to read and configure the BACnet properties of the Device Object for a pCOWeb/pCOnet device. The **pCOWeb Device Instance** is the Device Instance of a pCOWeb/pCOnet device.

Clicking **Read** reads all the properties for the pCOWeb/pCOnet Device Object and displays their values on the form.

Any of the values can be changed on the form and then committed to the pCOWeb/pCOnet by clicking **Write**. If you change the **Device Instance** property, you will also have to change the **pCOWeb Device Instance** to communicate further with that pCOWeb/pCOnet device, since changes to the **Device Instance** property take place immediately in the pCOWeb/pCOnet.

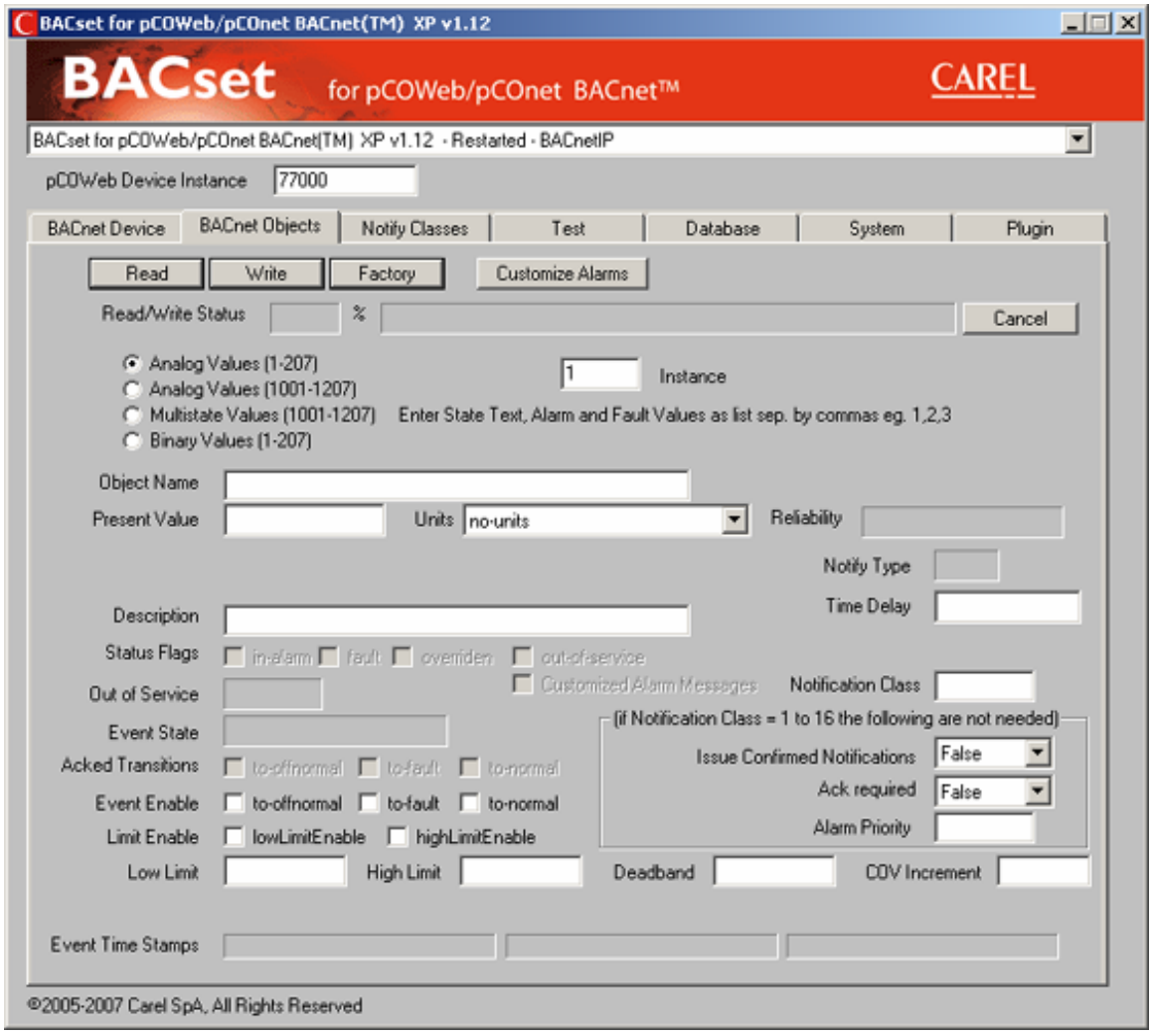
Clicking **Factory** resets all the values on the form to their factory or default values. The factory settings are not committed to the PCOWeb/pCOnet until **Write** is clicked.

Clicking **Time Sync** synchronizes the pCOWeb/pCOnet clock with the current time of the PC adjusted by the UTC offset.

Caution. BACset allows you to select 76800 baud for the pCOnet's baud rate. Since most Windows-based PCs do not support this baud rate without special drivers, future communication to the pCOnet using BACset may be impossible without booting the pCOnet in factory mode.

Clicking **Reboot** causes the pCOWeb/pCOnet to restart, if the **Password** matches the Password property of the PCOWeb/pCOnet's Device Object.

BACnet Objects Tab



This Tab is used to read and configure the BACnet properties of the Analog Value, Binary Value Objects and Multi-state Values for a pCOWeb/pCOnet device. The **pCOWeb Device Instance** is the Device Instance of a pCOWeb/pOCNet device.

Clicking **Read** reads all the properties for the selected pCOWeb/pCOnet Analog Value, Multi-state Value or Binary Value Object and displays their values on the form.

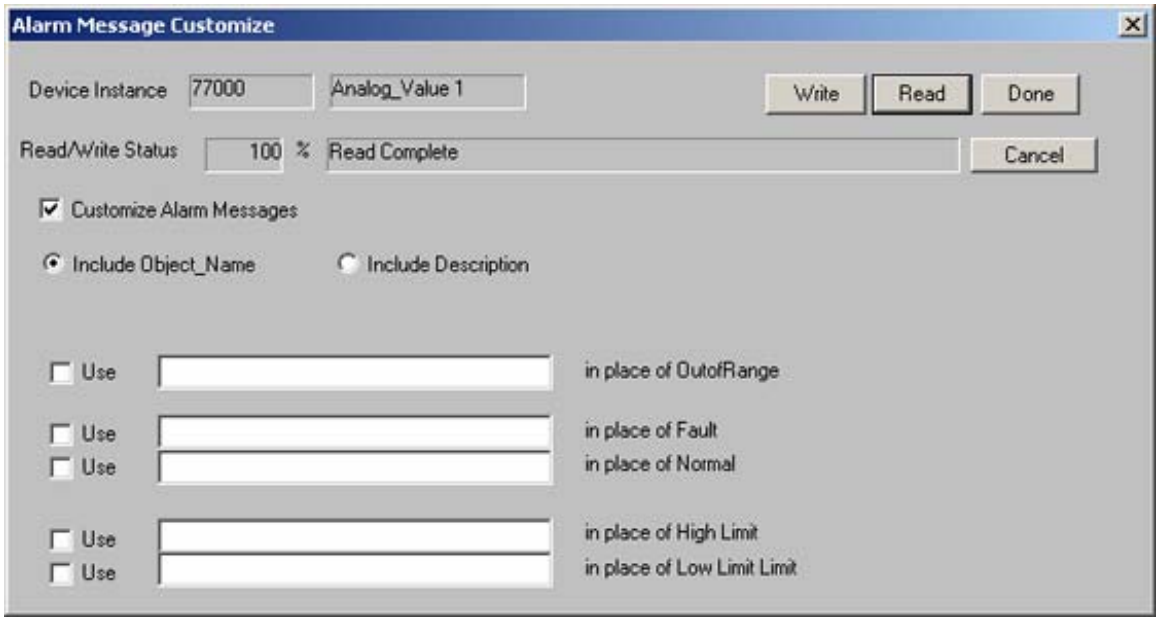
Any of the values for the writable properties can be changed on the form and then committed to the pCOWeb/pCOnet by clicking **Write**.

Clicking **Factory** resets all the values on the form to their factory or default values. The factory settings are not committed to the PCOWeb/pCOnet until **Write** is clicked.

Clicking **Customize Alarms** displays a separate form that allows you to customize the Alarm Message Text portion of BACnet alarms. Use of the form is described in the next section.

The Integer Type objects (instances 1001-1207) are defaulted to Analog Values. They may be reprogrammed as Multi-state Values on an individual basis by entering the Instance (1001-1207), clicking the Multistate Value radio button, entering all the new properties and clicking **Write**. Likewise, Multi-state Values may be reprogrammed as Analog Values on an individual basis by entering the Instance (1001-1207), clicking the Analog Value radio button, entering all the new properties and clicking **Write**. Please note that the programmability of Analog Values to Multi-state Values is supported in Firmware and Application Software versions 1.01 and later. The Firmware and Application Software versions are displayed for a Device on the Device tab.

Alarm Message Customization Form



This form is used to read and customize the parameters for the Alarm Message Text of BACnet alarms originating from the pCOnet/pCOWeb.

Clicking **Read** reads all the proprietary alarm customization properties for the selected pCOWeb/pCOnet Analog Value, Multi-state Value or Binary Value Object and displays their values on the form.

Any of the values for the writable properties can be changed on the form and then committed to the pCOWeb/pCOnet by clicking **Write**.

By default the Message Text that is included with ConfirmedEventNotification or UnconfirmedEventNotification looks similar to the following two examples:

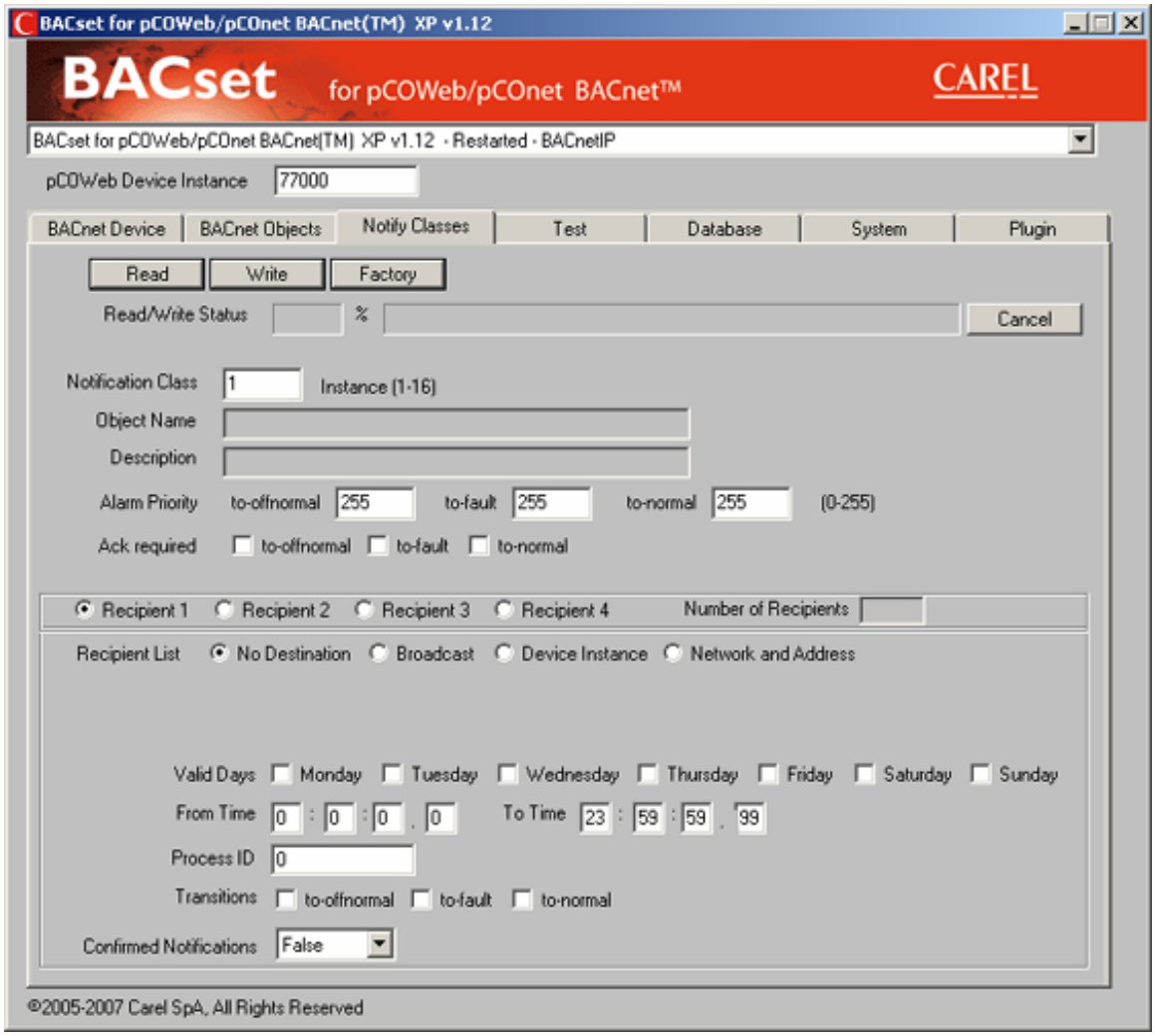
Object_Name (Analog Value 1) OutOfRange High Limit
Object_Name (Binary Value 1) ChangeofState Normal

The Alarm Message customization form allows you to replace the following default text with anything you wish:

- ChangeofState
- OutofRange
- Normal
- Offnormal
- Fault
- High Limit
- Low Limit

The custom text you choose can be up to 32 characters in length for each string. In addition you can replace the Object_Name (which is up to 32 characters long) with the Description property of the object (which is up to 64 characters long).

Notify Classes Tab



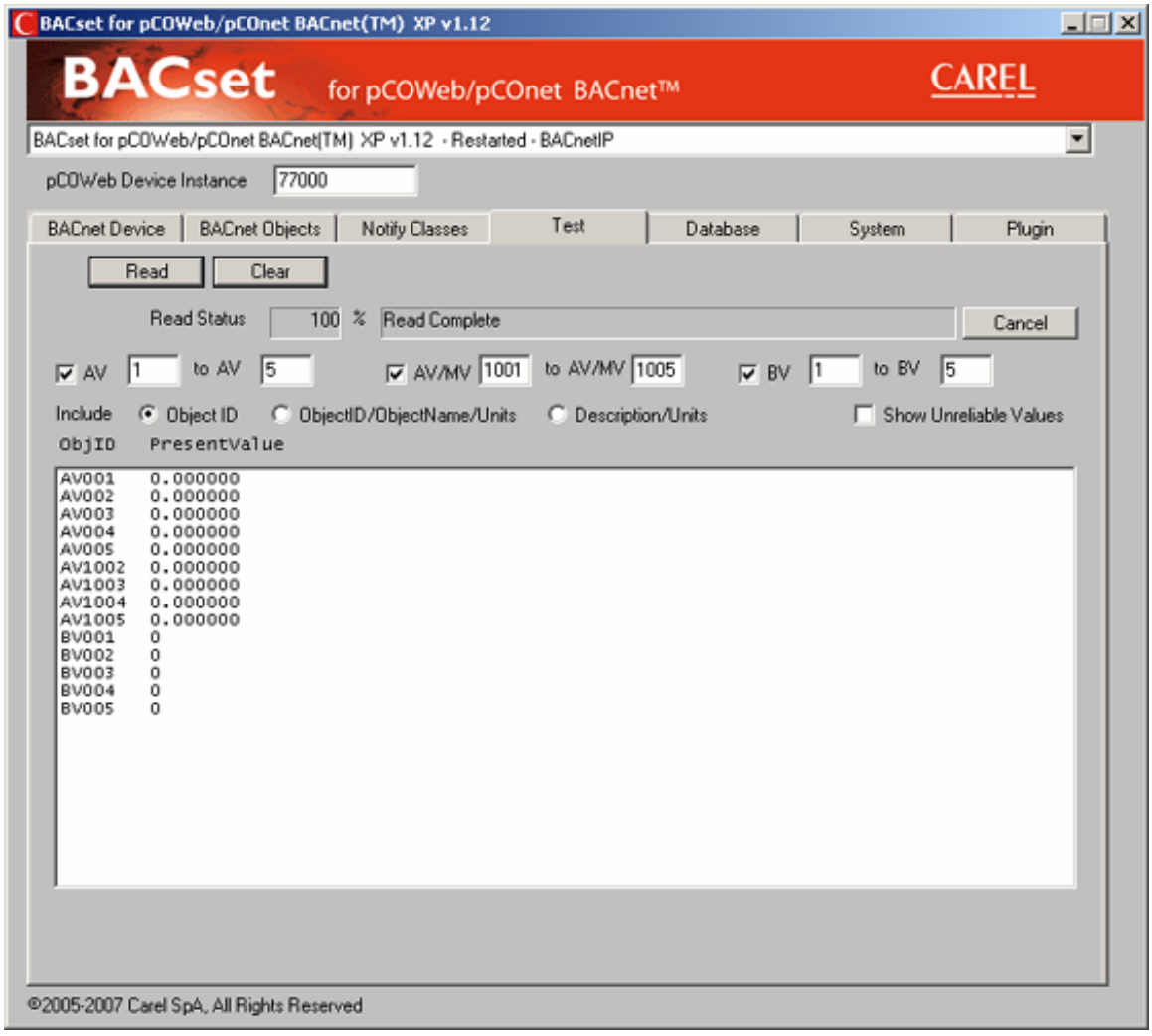
This Tab is used to read and configure the BACnet properties of the Notification Class Objects for a pCOWeb/pCOnet device. The **pCOWeb Device Instance** is the Device Instance of a pCOWeb/pOCNet device.

Clicking **Read** reads all the properties for the selected pCOWeb/pCOnet Notification Class Object and displays their values on the form.

Any of the values for the writable properties can be changed on the form and then committed to the pCOWeb/pCOnet by clicking **Write**.

Clicking **Factory** resets all the values on the form to their factory or default values. The factory settings are not committed to the PCOWeb/pCOnet until **Write** is clicked.

Test Tab



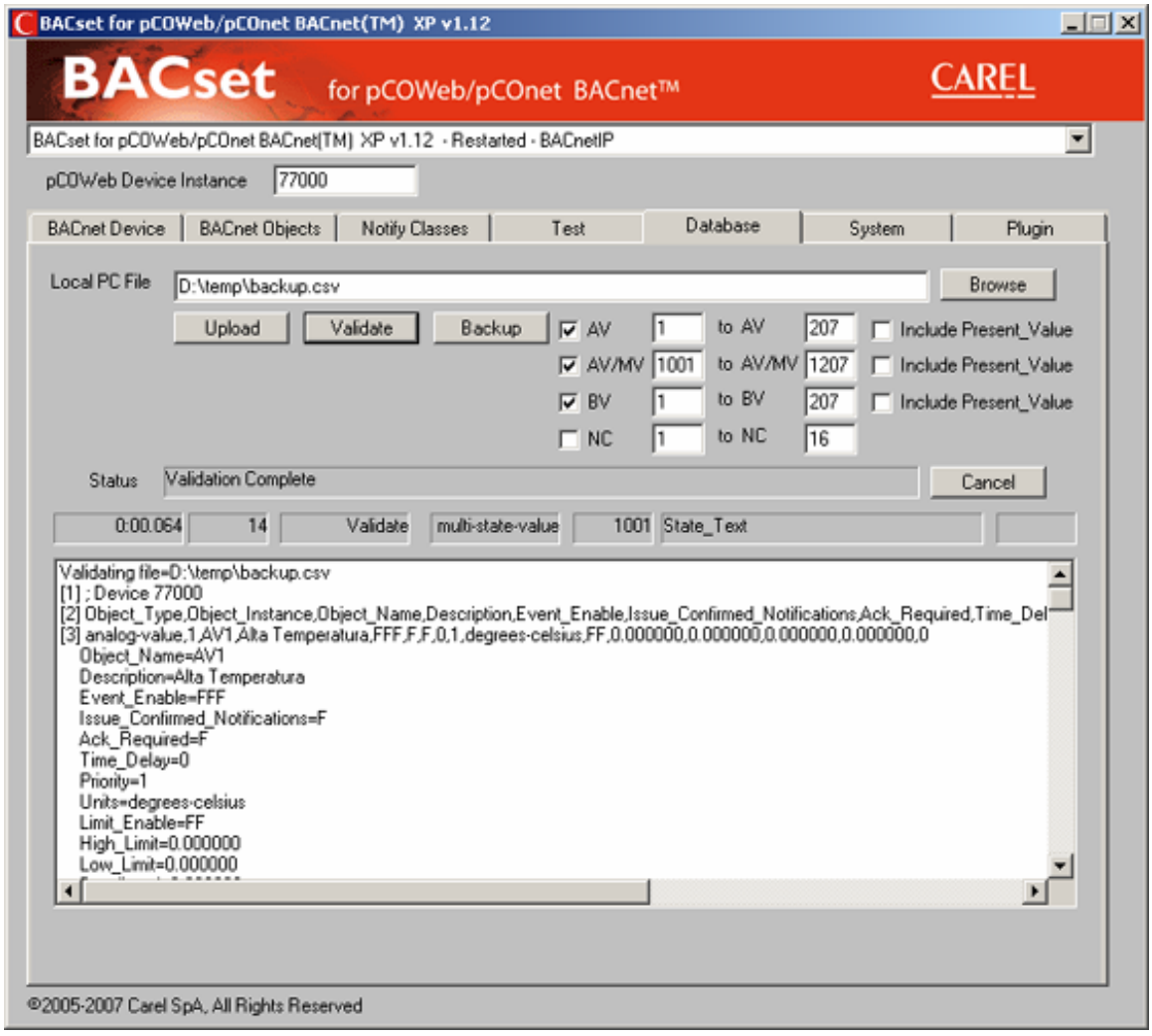
This Tab is used to test the reading of the BACnet properties of the Analog Value and Binary Value Objects for a pCOWeb/pCOnet device. The **pCOWeb Device Instance** is the Device Instance of a pCOWeb/pCOnet device.

The radio buttons are used to select the properties of the Objects to display.

- Selecting **Object ID** displays the Object_Identifier and Present_Value properties of every Analog Value and Binary Value in the pCOWeb/pCOnet.
- Selecting **ObjectID/ObjectName/Units** displays the Object_Identifier, Object_Name, Present_Value and Unit properties of every Analog Value and Binary Value in the pCOWeb/pCOnet.
- Selecting **Description/Units** displays the Description, Present_Value and Unit properties of every Analog Value and Binary Value in the pCOWeb/pCOnet.

If the **Show Unreliable Values** checkbox is checked, all Analog Value and Binary Values are displayed regardless of the value of their Reliability property. If the **Show Unreliable Values** checkbox is not checked, then only those Analog Value and Binary Values whose Reliability property is *no-fault-detected* will be displayed. Normally an Analog Value or Binary Value whose Reliability is set to *unreliable-other* indicates that the object is not valid for that particular pCOWeb/pCOnet configuration. See the ***pCOWeb/pCOnet Features*** section for details about the pCOWeb/pCOnet BACnet to pCO mapping and the use of the Reliability property.

Database Tab



This Tab is used to Upload and Backup part or all of the BACnet properties of the Analog, Multi-state, Binary and Notification Class Objects for a pCOWeb/pCOnet device using comma-separated (CSV) files. The **Device Instance** is the Device Instance of a pCOWeb/pCOnet device. CSV files can be created, modified, imported and exported by such general purpose tools as Microsoft Excel.

Browse is used to select a CSV file to Upload or Backup. Alternately the complete file name can be typed into the **Local PC File** edit box.

Clicking **Backup** causes all the writable properties for all the Analog Value, Multi-state and Binary Value Objects selected by the **AV1 to AV207**, **AV/MV1001 to AV/MV1207** and **BV1 to BV207** checkboxes to be read (using the BACnet readPropertyMultiple and readProperty services) from the pCOWeb/pCOnet device and saved in the specified CSV file. Notice that the ranges for the Objects are selectable, so partial Backups can be performed. This is particularly useful if you know that not all of the 207 Objects of a particular type are valid for a particular pCOWeb/pCOnet device. If the **Include Present_Value** checkbox is checked, the Present_Value property is included with the Backup, otherwise it is not. Progress is indicated in the list box and the line above the list box.

Clicking **Validate** reads the selected CSV file and checks its format. **Validate** does **not** check value ranges for validity, only that the property specifications are of the proper format. Progress is indicated in the list box and the line above the list box. Normally validation takes under a second even for a file with the full 621 entries, so it's always a good idea to validate a CSV file prior to uploading it.

Clicking **Upload** reads the selected CSV file and writes every property specified to every Object specified (using the BACnet writePropertyMultiple and writeProperty service). Progress is indicated in the list box and the line above the list box.

Database CSV File Format

A CSV file for use with BACset consists of the following three types of lines:

- Lines that begin with a semicolon (;) are considered *comments* and are ignored by BACset
- Lines that begin with the text *Object_Type,Object_Instance* are *template lines* and define a list of the standard BACnet properties for the *object lines* that follow them. Any or all of the standard properties can be included in the *template line* following the *Object_Type,Object_Instance.* The order of the remaining properties does not matter, however they must be exactly specified (e.g. Object_Name, Present_Value, etc.). Normally there would be two *template lines*, one for the Analog Values and one for the Binary Values, since they have several properties that differ. But any number of template lines can be included to customize which properties and Objects get Uploaded or Backed up. A *template line* is in effect until another *template line* is encountered or until the end of the file is reached
- All other lines are considered *object lines* and contain the values to be written for each of the properties of an object as specified in a *template line.* Each *object line* represents one Object in a pCOWeb/pCOnet. The values specified in the object lines must be in the same order as the properties listed in the *template line* directly above the *object line*. Because there are only two types of BACnet objects (other than the Device Object which cannot be included in the CSV file) in a pCOWeb/pCOnet, each *object line* must begin with one of the following strings (where ### represents the Instance of a BACnet Object):

    - *analog-value,###*
    - *binary-value,###*
    - *multi-state-value,###*
    - *notification-class,###*
    - *Analog_Value,###*
    - *Binary_Value,###*
    - *Multi_State_Value,###*
    - *Notification_Class,###*

All lines must be terminated by a new line (line feed) character.

The following is a simple example of a simple CSV file to configure a total of 6 Objects, 2 Analog Values, 2 Binary Values, 2 Multi-state Values and 2 Notification Class objects:

```
Object_Type,Object_Instance,Object_Name,Present_Value,Units
Analog-value,1,HUMIDITYSETPOINT,50.0,percent
Analog-value,2,TEMPSP,75.0,degrees-Fahrenheit
Object_Type,Object_Instance,Object_Name,Present_Value,Inactive_Text,Active_Text
binary-value,1,LIGHT,0,Off,On
binary-value,2,VALVE,0,Open,Closed
Object_Type,Object_Instance,Object_Name,Present_Value,Number_Of_States,State_Text
multi-state-value,1001,MS1001,1,4,”State 1,State 2,State 3,State 4”
multi-state-value,1002,MS1002,1,6,”STATE 1,STATE 2,STATE 3,STATE 4,STATE 5, STATE 6”
Object_Type,Object_Instance,Object_Name,Description,Ack_Required,Priority,Recipient_List
notification-class,1,NC1,Notification Class 1, T, "255,255,255",
                "validDays=MTWTFSS,fromTime=00:00:00.00,
                toTime=23:59:59.99,processIdentifier=100,IssueConfirmedNotifications=F,
                transitions=TFT,recipient=DE1000"
notification-class,2,NC2,Notification Class 2,T,"255,255,255",""
```

In this sample lines 1, 4, 7 and 10 are *template lines* and lines 2, 3, 5, 6, 8, 9, 11 and 12 are *object lines*. Note also that the State_Text, Priority and Recipient_List properties inside the object lines are themselves comma-separated and must be completely enclosed in double-quotes.

The following table summarizes the rules and format for the CSV files.

Template Line String (i.e. Property Name)	*Corresponding Object Line String*	*Valid for*
Object_Type	analog-value or Analog_Value	AV
Object_Type	binary-value or Binary_Value	BV
Object_Type	multi-state-value or Multi_State_Value	MV
Object_Type	notification-class or Notification_Class	NC
Object_Instance	1-207	AV
Object_Instance	1001 to 1207	AV, MV
Object_Instance	1-207	BV
Object_Instance	1-16	NC
Object_Name*	1 to 32 characters (should not contain spaces)	AV, BV, MV, NC
Description*	0 to 64 characters (spaces are ok)	AV, BV, MV, NC
Present_Value	x.xxxxx (floating point value)	AV
Present_Value	0 or 1	BV
Present_Value	1 to 32	MV
Units	Textual engineering units as described in Clause 21 (e.g. no-units, degrees-Fahrenheit, etc.) or corresponding numerical value (e.g. 95, 64, etc.)	AV
Event_Enable	FFF or 0x00 for none TFF or 0x80 for to-offnormal FFT or 0x20 for to-normal TFT 0r 0xA0 for to-offnormal or to-normal	AV, BV, MV
Notification_Class	0-16	AV, BV, MV
Issue_Confirmed_Notifications	T or F or 0 or 1	AV, BV, MV
Ack_Required	T or F or 0 or 1	AV, BV, MV
Time_Delay	xxxxx (integer value)	AV, BV, MV
Limit_Enable	FF or 0x00 for none TF or 0x80 for lowLimitEnable only FT or 0x40 for highLimitEnable only TT or 0xC0 for lowLimitEnable and highLimitEnable	AV
x.xxxxx (floating point)	x.xxxxx (floating point value)	AV
Low_Limit	x.xxxxx (floating point value)	AV
Deadband	x.xxxxx (floating point value)	AV
Inactive_Text	0 to 32 characters	BV
Active_Text	0 to 32 characters	BV
Alarm_Value	0 or 1	BV
Number_of_States	1 to 32	MV
State_Text	Up to 32 comma-separated strings of the form *statenumber=StateText.* The entire string must be enclosed in double quotes (e.g. “1=State 1,2=State 2,3=State3”) **	MV
Alarm_Values	Comma-separated list of states that are considered alarm states The entire string must be enclosed in double quotes (e.g. “4,5,6”) ****	MV
Fault_Values	Comma-separated list of states that are considered fault states The entire string must be enclosed in double quotes (e.g. “4,5,6”) ****	MV

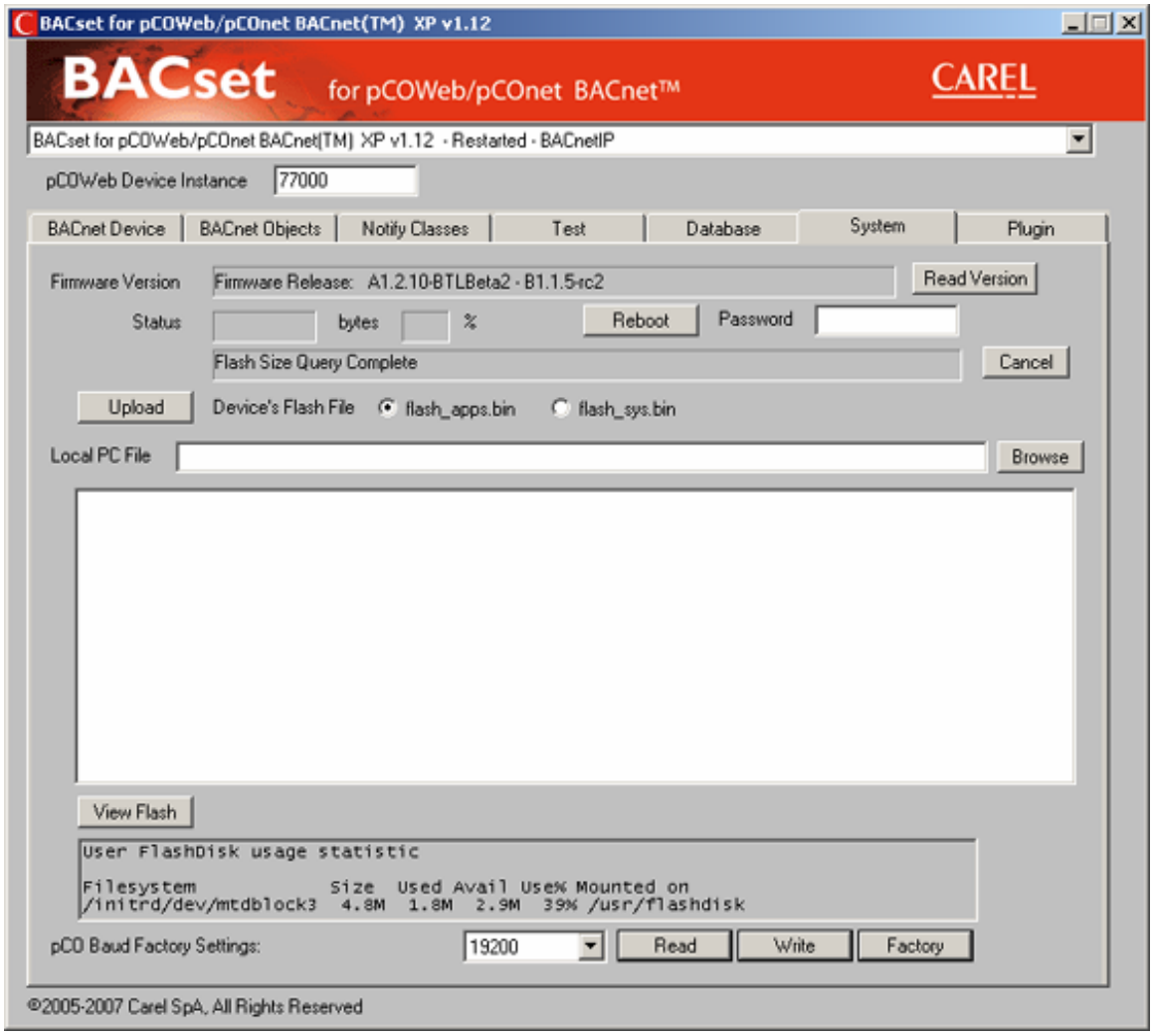
<table>
<tr><td>#xxxxx (where xxxxx = the proprietary property number, e.g. #3000)</td><td>Depends on the property</td><td>AV, BV, MV</td></tr>
<tr><td>Ack_Required</td><td>T or F or 0 or 1</td><td>NC</td></tr>
<tr><td>Priority</td><td>A comma-separated string consisting of exactly three numerical values from 0 to 255. The entire string must be enclosed in double quotes (e.g. “255,255,255”).</td><td>NC</td></tr>
<tr><td>Recipient_List</td><td>A comma-separated string consisting of components of the Recipient_List property. The components are identified by parameter name=value. The entire string must be enclosed in double quotes.

Parameter name — values
validDays — MTWTFSS (replace day of week with a dash (-) to disable that day (e.g. M-W-F-S)
fromTime — 00:00:00.00 to 23:59:59.99
toTime — 00:00:00.00 to 23:59:59.99
processIdentifier — 0 to 4294967295
IssueConfirmed Notifications — T or F or 0 or 1
transitions — xxx where x=T or F
recipient — DExxxx for devices or 0xHHHHHHHH for addresses

Example: (“validDays=MTWTFSS, fromTime=00:00:00.00, toTime=23:59:59.99,processIdentifier=100, IssueConfirmedNotifications=F, transitions=TFT,recipient=DE1000”). Note “” indicates no recipient, i.e. an empty list)</td><td>NC</td></tr>
</table>

***** Note: These properties although included for Notification Class objects are not written when uploading the csv file, since they are not writable properties for Notification Class objects.
****** Note: When these properties are entered using the Object tab, the double quotes must be omitted.

System Tab



This Tab is used to provide System configuration normally provided by the admin.html form available for pCOWeb devices. The **Device Instance** is the Device Instance of a pCOWeb/pCOnet device.

Clicking **Read Version** reads the Firmware/Software version of the pCOWeb/pCOnet.

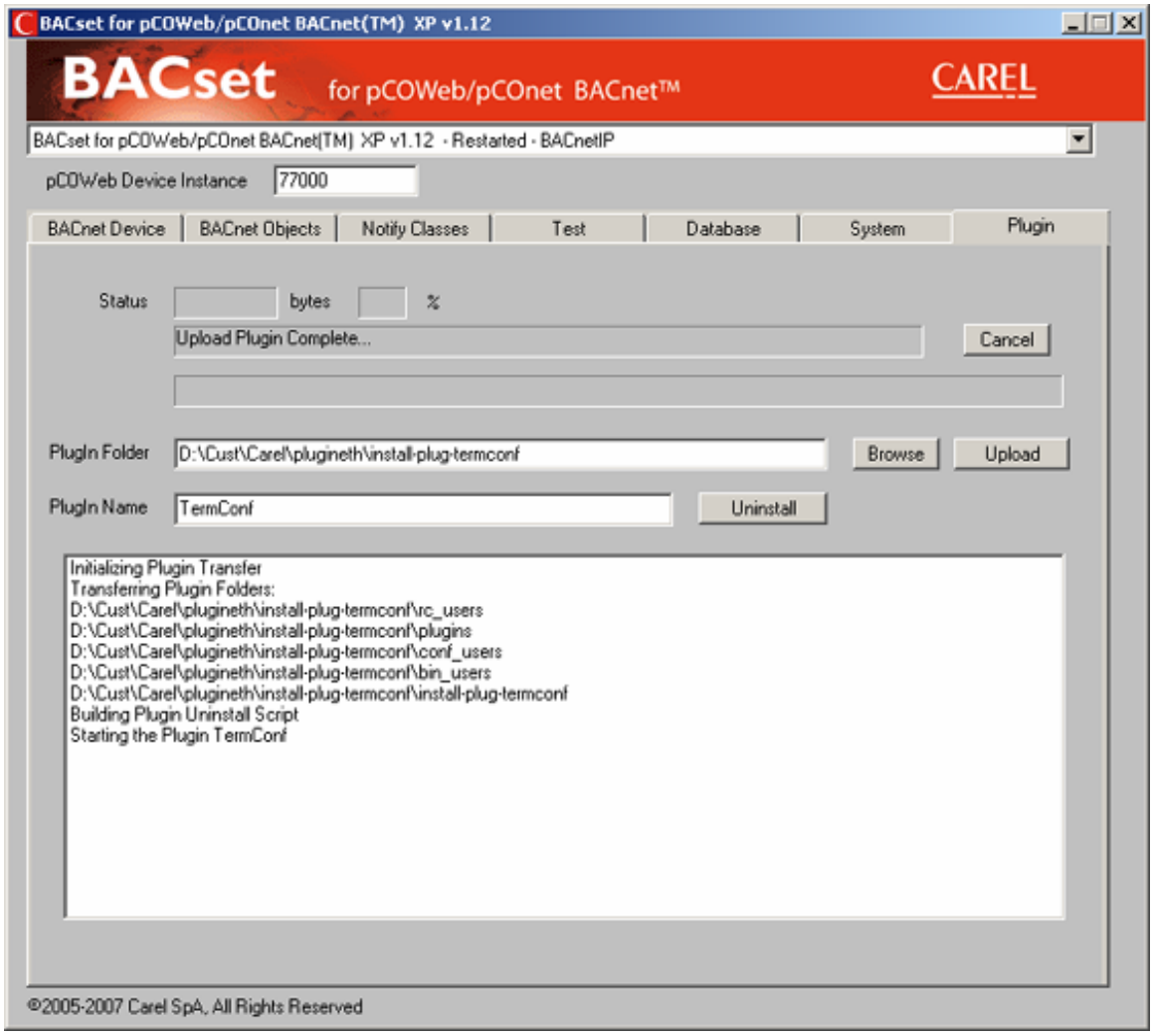
Clicking **Reboot** causes the pCOWeb/pCOnet to restart, if the **Password** matches the Password property of the PCOWeb/pCOnet's Device Object.

Clicking **View Flash** displays the Flash Memory statistics for the pCOWeb/pCOnet.

Browse is used to select a file to Upload. Alternately the complete file name can be typed into the **Local PC File** edit box.

Clicking **Upload** writes the PC file that is selected in the **Local PC File** edit box to the pCOWeb/pCOnet **flash_apps.bin** or **flash_sys.bin** file on the pCOWeb/pCOnet.

Plugin Tab



This Tab is used to install and uninstall plugin applications for the pCOWeb/pCOnet. The **Device Instance** is the Device Instance of a pCOWeb/pCOnet device.

Browse is used to select a plugin folder to Upload. Alternately the complete pathname of the folder can be typed into the **Plugin Folder** edit box.

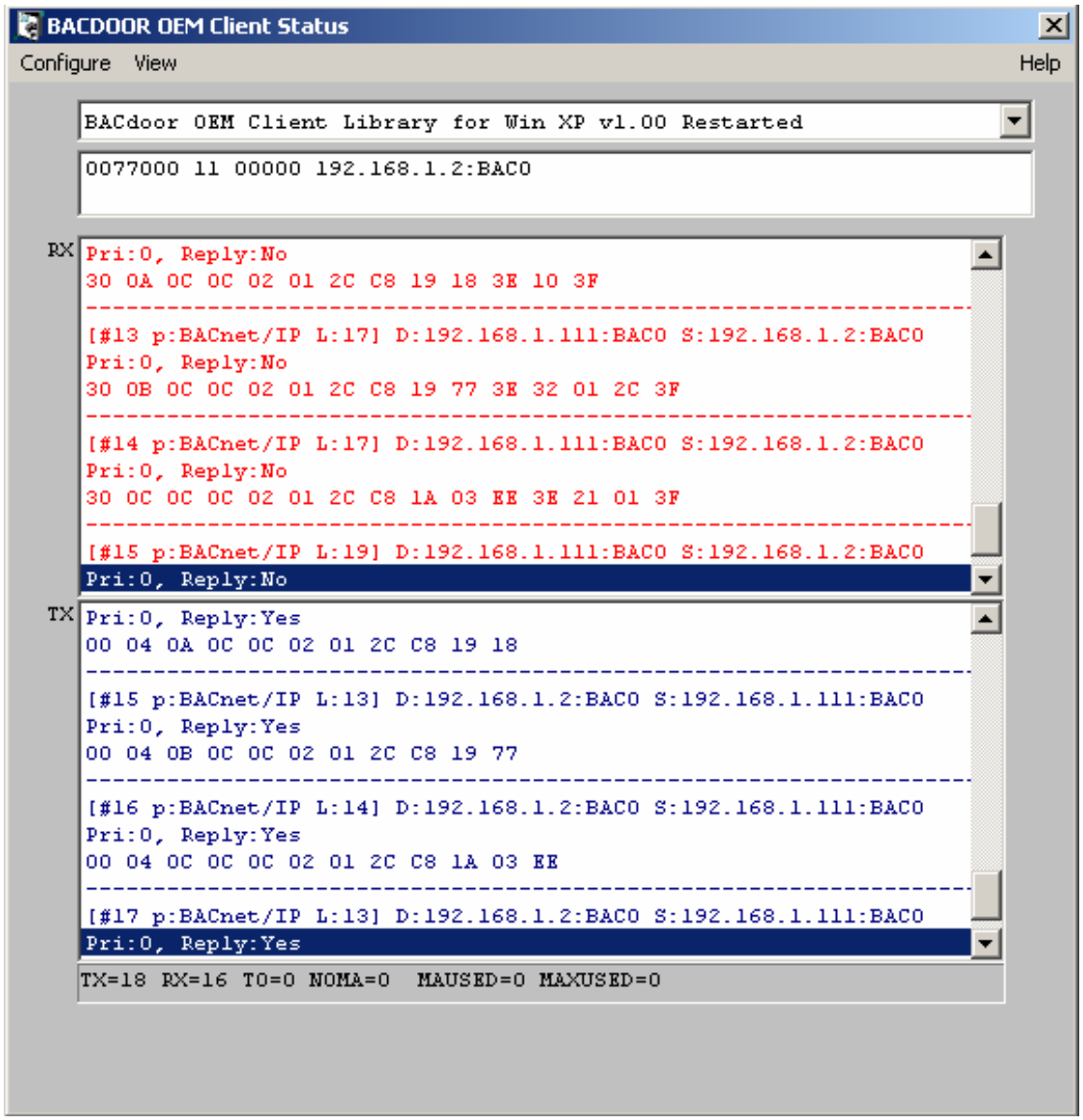
Clicking **Upload** searches the entire folder and all its subfolders that is selected in the **Plugin Folder** edit box and uploads all the files contained in them to the pCOWeb/pCOnet. Following the upload, a request is made to the pCOWeb/pCOnet to start the plugin application.

Clicking **Uninstall** makes a request to the pCOWeb/pCOnet to uninstall the plugin application.

BACdoor OEM Client Status Window

General

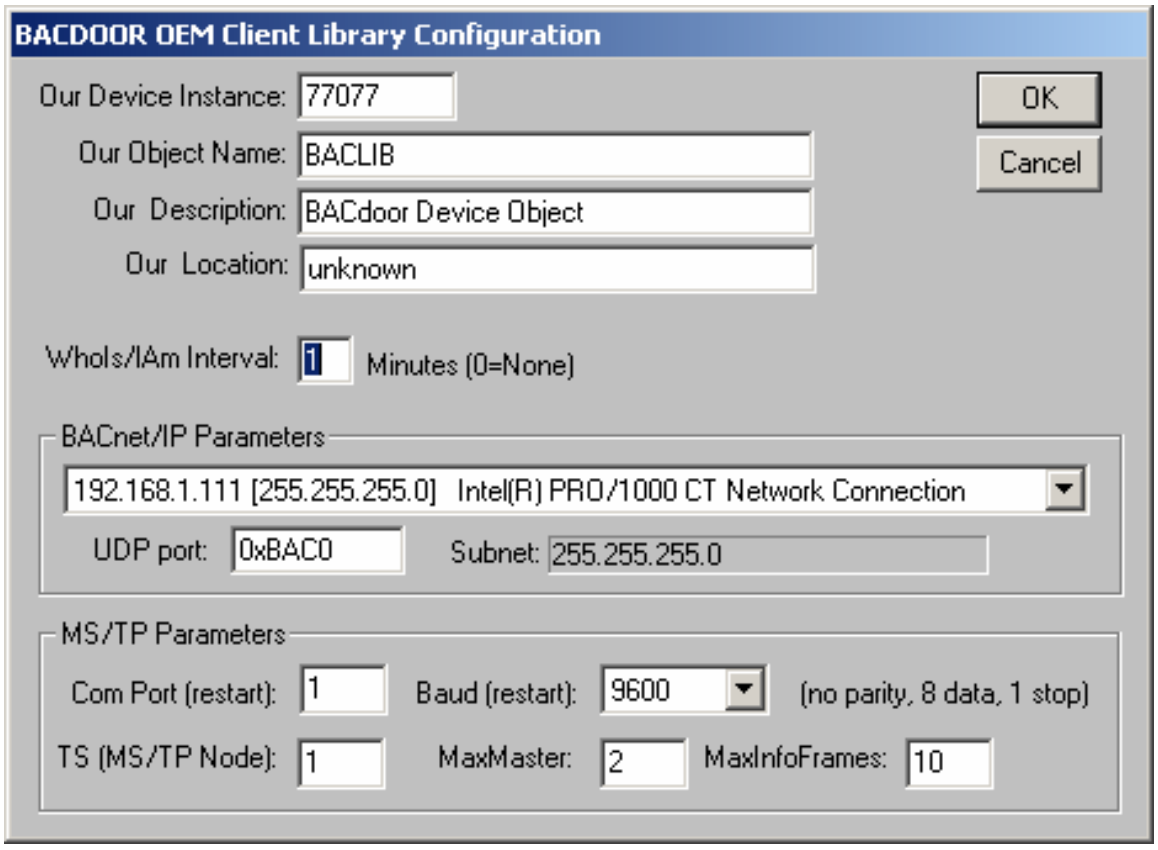
When you Click the  icon in the Windows Tool Bar at the bottom of your display screen, the "BACdoor OEM Client Status" Window is made visible.



This Window displays the status of BACnet packets as they are being exchanged between BACset, any pCOWeb/pCOnet device(s) and any other BACnet Devices on the network. It also displays a list of all BACnet Devices that initiate BACnet I_Am service packets.

Configuration of BACset

When you Click **Configure** in the "BACdoor OEM Client Status" Window's menu, a popup dialog appears that allows you to set operating parameters for the BACset Device itself.



Placement of the BACset configuration in this seemingly out of the way location is done on purpose because:

- configuring BACset is done very infrequently and
- so that the parameters are not confused with the pCOWeb/pCOnet Device properties

Most of the configuration parameters are self-explaining. The dropdown list in BACnet/IP Parameters is used only if your PC has been configured for more than one IP address.

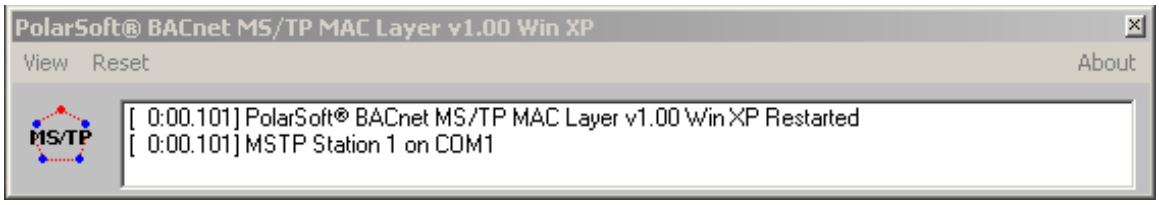
If you change the MS/TP baud rate, you must restart BACset for it to take effect.

Normally, this Window can remain minimized in the Task Bar. It is available for diagnostic purposes and of course to configure the BACset tool itself.

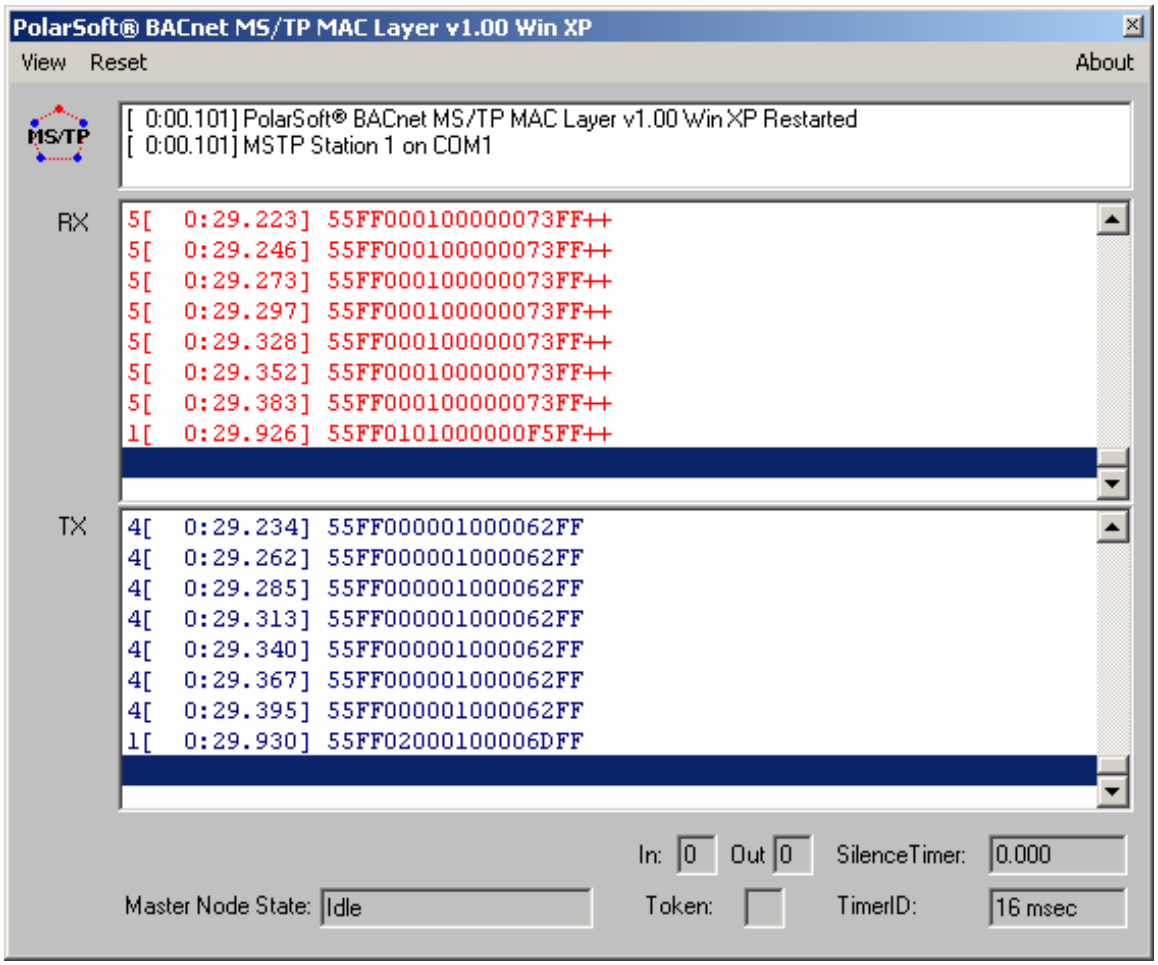
MS/TP MAC Layer Status Window

General

When you click the  icon in the Windows Tool Bar at the bottom of your display screen, the "MS/TP MAC Layer Status" Windows is made visible.



This Window can be expanded by clicking the **View-Trace All Messages** menu selection.



In its expanded form, this Window displays the status of MS/TP packets as they are being exchanged between BACset, any pCOnet device(s) and any other BACnet MS/TP Devices on the network. You should use the expanded window sparingly and for diagnostic purposes only, since it impacts the operation of the BACset MS/TP especially at higher baud rates. It is important to make sure this is turned off during an image Upload.

PCOWeb/pCOnet Features

General

BACset has an intimate knowledge of the configuration and features of the pCOWeb/pCOnet.

The Carel BACnet Gateway (**CBG**) component of the pCOWeb/pCOnet provides a gateway between the BACnet protocol and the Carel pCO protocol. **CBG** maps a maximum 207 pCO Analogs, a maximum 207 pCO Integers and a maximum 207 pCO Digitals in the following manner:

- pCO Analogs 1 to 207 *are mapped to* BACnet Analog Values 1 to 207
- pCO Integers 1 to 207 *are mapped to* BACnet Analog Values 1001 to 1207 but may be reprogrammed as Multi-state Values on an individual basis
- pCO Digitals 1 to 207 *are mapped to* BACnet Binary Values 1 to 207

From the factory, there are a total of 621 Analog Values, Multi-state Values and Binary Values in every pCOWeb/pCOnet. If the actual number of pCO points in-service is less than 621, the actual number of each may type can be configured from the BACnet Device tab. For any pCO point that is not in-service but is selected as mapped, **CBG** sets the BACnet properties in the corresponding BACnet object:

- Reliability = *unreliable-other*
- Status_Flag =*fault*

pCOWeb/pCOnet BACnet Device Object

CBG maintains a database in the pCOWeb/pCOnet for the all the *required* properties of the BACnet Device Object as well as the following *optional* properties:

- Daylight_Savings_Time *
- Description *
- Location *
- Local_Time
- Local_Date
- UTC_Offset *

In addition, the following standard properties are included in the Device Object in a non-standard way for the purpose of alarming:

- Recipient *
- Process_Identifier *

* *Indicates properties that are writable using BACnet writeProperty and writePropertyMultiple services through a BACnet client.*

The following standard properties are writable using BACnet writeProperty and writePropertyMultiple services using any BACnet client:

- APDU_Timeout
- Number_of_APDU_Retries
- Object_Identifier

pCOWeb/pCOnet BACnet Object Properties for Analog Values

CBG maintains a database in the pCOWeb/pCOnet for the all the *required* properties of the BACnet Analog Value Objects as well as the following *optional* properties:

- Acked_Transitions
- Deadband *
- Description *
- Event_Enable *
- Event_Time_Stamps
- High_Limit *
- Limit_Enable *
- Low_Limit *
- Notification_Class *
- Notify_Type
- Reliability
- Time_Delay

In addition, if the Notification_Class is set to 0, the following standard properties are included in the Analog Value Objects in a non-standard way for the purpose of alarming:

- Ack_Required *
- Issue_Confirmed_Notifications *
- Priority *

** Indicates properties that are writable using BACnet writeProperty and writePropertyMultiple services through a BACnet client.*

The following standard properties are writable using BACnet writeProperty and writePropertyMultiple services through any BACnet client:

- Object_Identifier **
- Object_Name
- Present_Value
- Units

*** This property is writable for instances 1001-1207 to reprogram Analog Value object types to Multi-state Value object types individually using BACnet writeProperty and writePropertyMultiple services through a BACnet client.*

pCOWeb/pCOnet BACnet Object Properties for Binary Values

CBG maintains a database for the all the *required* properties of the BACnet Binary Value Objects as well as the following *optional* properties:

- Acked_Transitions
- Active_Text *
- Alarm_Value *
- Description *
- Event_Enable *
- Event_Time_Stamps
- Inactive_Text *
- Notification_Class *
- Notify_Type
- Reliability
- Time_Delay

In addition, if the Notification_Class is set to 0, the following standard properties are included in the Binary Value Objects in a non-standard way for the purpose of alarming:

- Ack_Required *
- Issue_Confirmed_Notifications *
- Priority *

** Indicates properties that are writable using BACnet writeProperty and writePropertyMultiple services through a BACnet client.*

The following standard properties are writable using BACnet writeProperty and writePropertyMultiple services through any BACnet client:

- Object_Name
- Present_Value

pCOWeb/pCOnet BACnet Object Properties for Multi-state Values

CBG maintains a database in the pCOWeb/pCOnet for the all the *required* properties of the BACnet Multi-state Value Objects as well as the following *optional* properties:

- Acked_Transitions
- Alarm_Values*
- Deadband *
- Description *
- Event_Enable *
- Event_Time_Stamps
- Fault_Values*
- Notification_Class *
- Notify_Type
- Reliability
- State_Text *
- Time_Delay

In addition, if the Notification_Class is set to 0, the following standard properties are included in the Multi-state Value Objects in a non-standard way for the purpose of alarming:

- Ack_Required *
- Issue_Confirmed_Notifications *
- Priority *

** Indicates properties that are writable using BACnet writeProperty and writePropertyMultiple services through a BACnet client.*

The following standard properties are writable using BACnet writeProperty and writePropertyMultiple services through any BACnet client:

- Object_Identifier **
- Object_Name
- Present_Value
- Units

*** This property is writable for instances 1001-1207 to reprogram Multi-state Value object types to Analog Value object types individually using BACnet writeProperty and writePropertyMultiple services through a BACnet client.*

pCOWeb/pCOnet BACnet Object Properties for Notification Class Values

CBG maintains a database in the pCOWeb/pCOnet for the all the *required* properties of the BACnet Notification Class Objects as well as the following *optional* properties:

- Description