

BACset Users Guide

2-Dec-2011

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, Windows Vista (32-Bit) and Windows 7 (32-Bit).

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:
 - o by connecting an external EIA-232 to EIA-485 converter to an existing PC serial port
 - o 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.

Please note that BACMAC2K does not work with Vista or Windows 7 64-bit versions. A separate 64-bit driver is currently under development.

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:

- Device
- Objects
- Notify Classes
- Schedules
- Calendars
- 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.

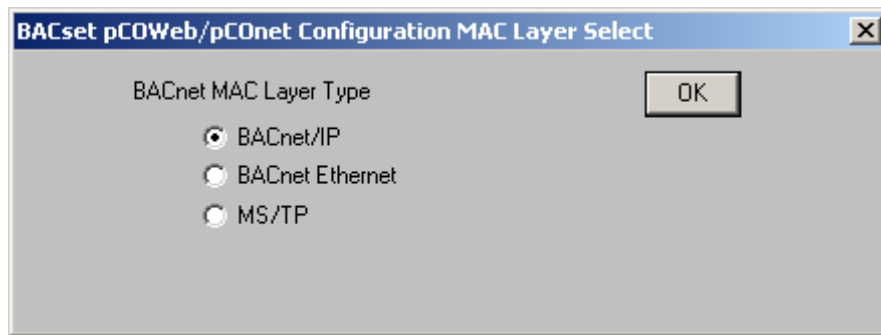
If you select the MS/TP MAC layer, the additional “MS/TP MAC Layer Status” Window is minimized into the Windows Tool Bar at the bottom of your display screen and is indicated by the  icon. Clicking on the icon makes the Status window visible.

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.

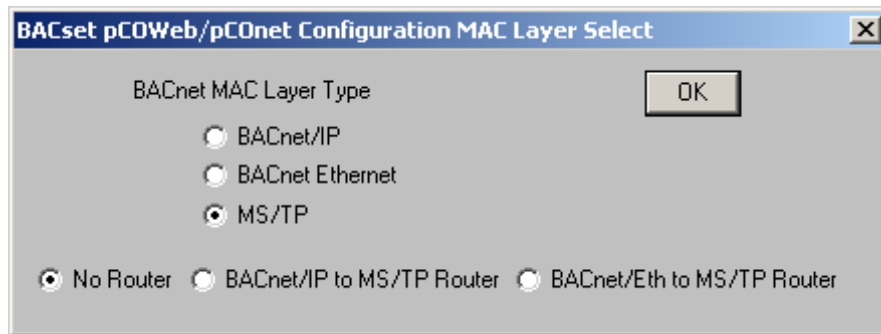
If you have only one IP address configured, which is the normal case, no selection is necessary, and BACset uses that one IP address.

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/pCOnet you are configuring. If that is the case you should select "No Router". If the pCOnet is however connected through a router you must chose one of the two Router selections.

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
 - o **[General]**
- Place a second line of any of the following
 - o **MACtype=MSTP**
 - o **MACtype=BIP**
 - o **MACtype=Eth**
- If you want to connect to a pConet through a router and MACtype=MSTP, then add another line of any of the following
 - o **Router=BIP**
 - o **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 nine Tabs is described in the following sections.

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.

Please note that you must use the System Tab to change the Interface Protocol between the PCOWeb/pConet and the Carel controller it's plugged into from Carel to MODBUS Extended and MODBUS Extended to Carel. Selecting MODBUS Extended allows you to extend the total number of Analog Value, Multistate Value and Binary Value objects from 621 maximum to 6144 maximum. When the Interface Protocol is changed, the change does not take effect until the next reboot of the pCOWeb/pConet. The default/factory Interface Protocol is Carel. In order to use the extended MODBUS Extended protocol, the Carel controller must also support the MODBUS Extended protocol.

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

BACset for pCOWeb/pCOnet BACnet(R) Windows XP v2.15

BACset for pCOWeb/pCOnet BACnet(R) Windows XP v2.15 - Restarted - BACnetIP

pCOWeb Device Instance: 77000

Device Objects Notify Classes Schedules Calendars Test Database System Plugin

Read Write Factory Customize Alarms

Read/Write Status: % Cancel

☒ Analog Values (1-207) ☐ Analog Values (1001-1207) ☐ Multistate Values (1001-1207)* ☐ Binary Values (1-207)

* For Multistate Values enter State Text, Alarm and Fault Values as list sep. by commas

Object Name: Present Value: Units: no-units Reliability: Notify Type: Time Delay:

Description: Status Flags: ☐ in-alarm ☐ fault ☐ overridden ☐ out-of-service ☒ Customized Alarm Messages Notification Class: 0=none

Out of Service: Event State: Acked Transitions: ☐ to-offnormal ☐ to-fault ☐ to-normal

Event Enable: ☐ to-offnormal ☐ to-fault ☐ to-normal

Limit Enable: ☐ lowLimitEnable ☐ highLimitEnable

Low Limit: High Limit: Deadband: COV Increment:

Event Time Stamps:

(For alarms without Notification Classes)

Issue Confirmed Notifications: False

Ack required: False

Alarm Priority:

Only for pCOWeb/pCOnet version earlier than 2.13

©2005-2011 Carel SpA, All Rights Reserved

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/pCOnet 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**. Please note that all properties for an Object are committed to the pCOWeb/pCOnet when **Write** is clicked.

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.

If the Interface Protocol selected is Carel, then there a maximum 207 Analog variables mapped to Analog Value objects (instances 1 to 207), 207 Carel Integer variables mapped to Analog or Multistate Values (instances 1001-1207) and 207 Carel Digital variables mapped to Binary Value objects (instances 1 to 207). If the Interface Protocol selected is MODBUS Extended, then there a maximum 2048 MODBUS Extended Analog variables mapped to Analog Value objects (instances 100001 to 102048), 2048 MODBUS Extended Integer variables mapped to Analog or Multistate Values (instances 200001-202048) and 2048 Carel Digital variables mapped to Binary Value objects (instances 100001 to 102048). Throughout this document, references to these objects assume the Interface Protocol is Carel.

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 for Analog Value Objects

Alarm Message Customize

Device Instance: 77000 Analog_Value 1 [Write] [Read] [Done]

Read/Write Status: 100 % Read Complete [Cancel]

☒ Customize Alarm Messages

☐ Include Object_Name ☒ Include Description

☒ Use: OUTOFRANGE in place of OutofRange

☒ Use: FAULT in place of Fault

☒ Use: NORMAL in place of Normal

☒ Use: HIGH LIMIT in place of High Limit

☒ Use: LOW LIMIT in place of Low Limit

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

Clicking **Read** reads all the proprietary alarm customization properties for the selected pCOWeb/pCOWeb Analog 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/pCOWeb by clicking **Write**.

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

A001 (Analog Value 1) OutOfRange High Limit
A001 (Analog Value 1) OutOfRange Low Limit
A001 (Analog Value 1) OutOfRange Normal Limit

where A001 is the default Object_Name for the Analog Value 1 object. If you have changed the Object_Name for the Analog Value 1 object then it will appear in place of A001.

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

- o ChangeofState
- o Normal
- o Offnormal
- o Fault

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).

So using the values in the example form above the Message Text that is included with ConfirmedEventNotification or UnconfirmedEventNotification looks similar to the following:

Analog Variable 1 (Analog Value 1) OUTOFRANGE HIGH LIMIT
Analog Variable 1 (Analog Value 1) OUTOFRANGE LOW LIMIT
Analog Variable 1 (Analog Value 1) OUTOFRANGE NORMAL LIMIT

where Analog Variable 1 is the default Description for the Analog Value 1 object. If you have changed the Description for the Analog Value 1 object then it will appear in place of Analog Variable 1.

Alarm Message Customization Form for Binary Value Objects

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

Clicking **Read** reads all the proprietary alarm customization properties for the selected pCOWeb/pCOnet 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:

D001 (Binary Value 1) ChangeofState Offnormal
D001 (Binary Value 1) ChangeofState Normal

where D001 is the default Object_Name for the Binary Value 1 object. If you have changed the Object_Name for the Analog Value 1 object then it will appear in place of D001.

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

- o ChangeofState
- o Normal
- o Offnormal
- o Fault

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).

So using the values in the example form above the Message Text that is included with ConfirmedEventNotification or UnconfirmedEventNotification looks similar to the following:

Digital Variable 1 (Binary Value 1) CHANGEOFSTATE OFFNORMAL
Digital Variable 1 (Binary Value 1) CHANGEOFSTATE NORMAL

where Digital Variable 1 is the default Description for the Binary Value 1 object. If you have changed the Description for the Binary Value 1 object then it will appear in place of Digital Variable 1.

Alarm Message Customization Form for Multistate Value Objects

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

Clicking **Read** reads all the proprietary alarm customization properties for the selected pCOWeb/pCOWeb Multi-state 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/pCOWeb by clicking **Write**.

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

M1001 (Multistate Value 1) ChangeofState Offnormal
M1001 (Multistate Value 1) ChangeofState Fault
M1001 (Multistate Value 1) ChangeofState Normal

where M10001 is the default Object_Name for the Multistate Value 1 object. If you have changed the Object_Name for the Analog Value 1 object then it will appear in place of M001.

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

- o ChangeofState
- o Normal
- o Offnormal
- o Fault

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).

So using the values in the example form above the Message Text that is included with ConfirmedEventNotification or UnconfirmedEventNotification looks similar to the following:

Integer Variable 1 (Multistate Value 1) CHANGEOFSTATE OFFNORMAL
Integer Variable 1 (Multistate Value 1) CHANGEOFSTATE FAULT
Integer Variable 1 (Multistate Value 1) CHANGEOFSTATE NORMAL

where Integer Variable 1 is the default Description for the Multistate Value 1 object. If you have changed the Description for the Multistate Value 1 object then it will appear in place of Integer Variable 1.

Notify Classes Tab

The screenshot shows the 'BACset for pCOWeb/pCOnet BACnet(R) Windows XP v2.15' application window. The title bar includes the text 'BACset for pCOWeb/pCOnet BACnet(R) Windows XP v2.15' and the CAREL logo. The main window has a red header with 'BACset for pCOWeb/pCOnet BACnet®' and the CAREL logo. Below the header is a dropdown menu showing 'BACset for pCOWeb/pCOnet BACnet(R) Windows XP v2.15 - Restarted - BACnetIP'. The 'pCOWeb Device Instance' is set to '77000'. The 'Notify Classes' tab is selected, showing a 'Read', 'Write', and 'Factory' button set. Below these are 'Read/Write Status' and 'Cancel' buttons. The 'Notification Class' is set to '1' with an 'Instance (1-16)' field. The 'Object Name' and 'Description' fields are empty. The 'Alarm Priority' section has three input fields: 'to-offnormal' (255), 'to-fault' (255), and 'to-normal' (255), with a range '(0-255)' indicated. The 'Ack required' section has three checkboxes: 'to-offnormal', 'to-fault', and 'to-normal'. The 'Recipient List' section has four radio buttons: 'Recipient 1', 'Recipient 2', 'Recipient 3', and 'Recipient 4'. Below these are radio buttons for 'No Destination', 'Device Instance', 'Local Address', 'Remote Address', 'Local Broadcast', and 'Remote or Global Broadcast'. The 'Valid Days' section has checkboxes for 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', and 'Sunday'. The 'From Time' is '0 : 0 : 0' and the 'To Time' is '23 : 59 : 59'. The 'Process ID' is '0'. The 'Transitions' section has three checkboxes: 'to-offnormal', 'to-fault', and 'to-normal'. The 'Confirmed Notifications' dropdown is set to 'False'. The footer text is '©2005-2011 Carel SpA, All Rights Reserved'.

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/pCOnet 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**. Please note that all properties for a Notification Class are committed to the pCOWeb/pCOnet when **Write** is clicked.

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.

The Recipient List is a BACnet list of recipient destinations to which BACnet alarms and events are reported. In the PCOWeb/pCOnet there can be a maximum of 4 recipients per Notification Class Object. To setup one or more Recipients:

1. Select a desired Recipient, e.g. Recipient 1
2. Change the parameters for that recipient
3. Select the next recipient and select the parameters for that Recipient
4. Repeat this procedure for all Recipients, selecting No Destination for the unused Recipients.

Once all recipients have been changed, click **Write** to commit the changes to the pCOWeb/pCOnet.

Schedules Tab

BACset for pCOWeb/pCOnet BACnet(R) Windows XP v2.15

BACset for pCOWeb/pCOnet BACnet® **CAREL**

BACset for pCOWeb/pCOnet BACnet(R) Windows XP v2.15 - Restarted - BACnetIP

pCOWeb Device Instance 77000

Device Objects Notify Classes Schedules Calendars Test Database System Plugin

Read Write Factory

Read/Write Status % Cancel

Schedule 1 Instance (1-16)

Object Name

Description

Present Value Reliability

Schedule Default Data Type Null Value Null Priority for Writing 16

Status Flags invalid fault overridden out-of-service Out of Service

Obj Property Ref List Ref 1 none Instance Number of Refs 0 (64 max) Empty

Effective Period Start Date any any any End Date any any any

Schedule Mon Number of Schedules 7 Time Vals 0 (6 max) Empty (Time Vals this sched)

Data Type None Time Value 1 Time 0 : 00 Value

Schedule

☒ Weekly

☐ Exception

©2005-2011 Carel SpA, All Rights Reserved

This Tab is used to read and configure the BACnet properties of the Schedule Objects 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 selected pCOWeb/pCOnet Schedule 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**. Please note that all properties for a Schedule are committed to the pCOWeb/pCOnet when **Write** is clicked.

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.

The Object Property Reference is a BACnet list of object references in the pCOWeb/pCOnet whose Present Value properties are written with specific values at specific times on specific days. In the PCOWeb/pCOnet there can be a maximum of 64 references per Schedule Object. To setup one or more reference:

1. Select a desired reference number from the dropdown list
2. Select the Object Type and enter the Object Instance
3. Select the next reference and select the parameters for that reference
4. Repeat this procedure for all references, selecting none for the unused references

The actual schedules of a Schedule objects can be either Weekly or Exception types. Any Schedule Object can contain either or both types of schedules.

Weekly Schedules are events that happen on a daily basis. For each Schedule Object, the Weekly Schedule is a BACnet array of exactly 7 elements, one for each day of the week. Each entry in a Weekly Schedule is a BACnet list of Time and Value pairs. The Time is the time-of-day for which the corresponding Value is written to the entire Object Property Reference list. In the pCOWeb/pCOnet there can be a maximum of 6 Time and Value pairs for each Weekly Schedule array element. To setup one or more weekly schedule:

1. Select Weekly Schedule using the radio button
2. Select the desired Week Day from the dropdown list, e.g. Monday
3. For the Week Day define each Time and Value pair (1 to 6) using the dropdown list appropriate for the Week Day. Then select the corresponding time in hours and minutes, and the BACnet Datatype and Value to write to the Object Property Reference list. None should be selected in the Data Type dropdown list for the unused Time and Value pairs for the Week Day
4. Select the next Week Day and define the Time and Value pairs for that entry
5. Repeat this procedure for all Week Days, clicking Empty for days that have no Weekly Schedule

The Exception Schedule is a BACnet array that contains a set Special Events that may happen in addition to or in place of the Weekly Schedule for a Schedule Object. For each Schedule Object in the pCOWeb/pCOnet, the Exception Schedule is a BACnet array of 0 to 4 elements. Each entry in an Exception Schedule is a BACnet list of Time and Value pairs. The Time is the time-of-day for which the corresponding Value is written to the entire Object Property Reference list. In the pCOWeb/pCOnet there can be a maximum of 6 Time and Value pairs for each Exception Schedule array element. To setup one or more Exception Schedule properties:

1. Select Exception Schedule using the radio button
2. Select the desired Exception Schedule (1 to 4) from the dropdown list
3. For the Exception Schedule define each Time and Value pair appropriate for the Exception. Define the Time and Value pair (1 to 6) using the dropdown list and then select the corresponding time in hours and minutes, and the BACnet Datatype and Value to write to the Object Property Reference list. None should be selected in the Data Type dropdown list for the unused Time and Value pairs for the Exception Schedule
4. Select the Date Range type using the None, Calendar Ref or Calendar Entry for the Exception Schedule. None disables an exception, Calendar Reference references a Calendar Object that identifies a date, date range or month/week-of-month/day-of-week for the Exception Schedule and Calendar Entry defines a date, date range or month/week-of-month/day-of-week directly in the Exception Schedule
5. Select the next Exception schedule and define the Time and Value pairs for that entry
6. Repeat this procedure for all Exception schedules, clicking None for Exceptions that are not used

Once all Object Property References, Weekly and Exception Schedules and all their Time and Value pairs have been changed, click **Write** to commit the changes to the pCOWeb/pCOnet.

Calendars Tab

The screenshot shows the BACset software interface for Windows XP v2.15. The title bar reads "BACset for pCOWeb/pCOnet BACnet(R) Windows XP v2.15". The main window has a red header with "BACset" and "for pCOWeb/pCOnet BACnet®" and the CAREL logo. Below the header, a dropdown menu shows "BACset for pCOWeb/pCOnet BACnet(R) Windows XP v2.15 - Restarted - BACnetIP". A text field for "pCOWeb Device Instance" contains the value "77000".

The interface features a tabbed menu with the following tabs: Device, Objects, Notify Classes, Schedules, **Calendars**, Test, Database, System, and Plugin. The "Calendars" tab is active, displaying three buttons: "Read", "Write", and "Factory". Below these buttons is a "Read/Write Status" section with a percentage input field and a "Cancel" button.

The main configuration area includes the following fields:

- Calendar:** A dropdown menu showing "1" and the text "Instance (1-16)".
- Object Name:** A text input field.
- Description:** A text input field.
- Present Value:** A text input field.

A "Date List" section contains a "Calendar Entry" dropdown (set to "1") and a "Number of Calendar Entries" input field (set to "0" with a "(16 max)" label). Below this are four radio button options: "None (Empty)" (selected), "Date", "Date Range", and "Week and Day".

At the bottom left, the copyright notice "©2005-2011 Carel SpA, All Rights Reserved" is displayed.

This Tab is used to read and configure the BACnet properties of the Calendar Objects 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 selected pCOWeb/pCOnet Calendar 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**. Please note that all properties for a Calendar are committed to the pCOWeb/pCOnet when **Write** is clicked.

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.

The Date List is a BACnet list of calendar entries that identify specific dates, date ranges and month/week-of-month/day-of-week during which the Calendar Object is in effect. Calendar Objects can be used with Schedule Objects to extend the period in which Exception Schedules are in Effect. In the PCOWeb/pCOnet there can be a maximum of 16 calendar entries per Calendar Object. To setup one or more Date List properties:

1. Select a desired Calendar Entry from the dropdown list, e.g. Calendar Entry 1
2. Change the specific dates, date ranges and month/week-of-month/day-of-week for that calendar entry
3. Select the next calendar entry and select the parameters for that Calendar Entry
4. Repeat this procedure for all Calendar Entries, selecting None (Empty) for the unused calendar entries.

Once all calendar entries have been changed, click **Write** to commit the changes to the pCOWeb/pCOnet.

Test Tab

The screenshot shows the 'Test' tab of the BACset software. The window title is 'BACset for pCOWeb/pCOnet BACnet(R) Windows XP v2.15'. The interface includes a header with the 'BACset' logo and 'CAREL' branding. Below the header, there's a dropdown menu showing 'BACset for pCOWeb/pCOnet BACnet(R) Windows XP v2.15 - Restarted - BACnetIP'. A text field for 'pCOWeb Device Instance' contains the value '77000'. The 'Test' tab is selected among other tabs: Device, Objects, Notify Classes, Schedules, Calendars, Test, Database, System, and Plugin. The 'Test' tab contains a 'Read' button, a 'Clear' button, and a 'Read Status' field set to '100 %'. A 'Read Complete' button is also present. Below these, there are three checked checkboxes for reading ranges: 'AV 1 to AV 5', 'AV/MV 1001 to AV/MV 1005', and 'BV 1 to BV 5'. There are also radio buttons for 'Include' with options: 'Object ID' (selected), 'ObjectID/ObjectName/Units', and 'Description/Units'. A 'Show Unreliable Values' checkbox is also present. The main area displays a table of object data:

ObjID	Presentvalue
AV001	0.000000
AV002	0.000000
AV003	0.000000
AV004	0.000000
AV005	0.000000
MV1001	1
AV1002	0.000000
AV1003	0.000000
AV1004	0.000000
AV1005	0.000000
BV001	0
BV002	0
BV003	0
BV004	0
BV005	0

At the bottom, there is a copyright notice: '©2005-2011 Carel SpA, All Rights Reserved'.

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

BACset for pCOWeb/pCOnet BACnet(R) Windows XP v2.15

BACset for pCOWeb/pCOnet BACnet® **CAREL**

BACset for pCOWeb/pCOnet BACnet(R) Windows XP v2.15 - Restarted - BACnetIP

pCOWeb Device Instance: 77000

Device | Objects | Notify Classes | Schedules | Calendars | Test | Database | System | Plugin

Local PC File: D:\Cust\Carel\XML and CSV files\carel.csv [Browse]

[Upload] [Validate] [Backup] ☒ AV 1 to AV 207 ☐ Include Present_Value

for interface protocol: Carel ☒ AV/MV 1001 to AV/MV 1207 ☐ Include Present_Value

actual interface protocol: Carel ☒ BV 1 to BV 207 ☐ Include Present_Value

☐ NC 1 to NC 16

Status: Validation Complete [Cancel]

0:01.039 627 [Validate]

Validating file=D:\Cust\Carel\XML and CSV files\carel.csv

[1] : Device 77000

[2] Object_Type, Object_Instance, Object_Name, Description, Event_Enable, Issue_Confirmed_Notifications, Ack_Required, Time_Del

[3] analog-value, 1, A001, Analog Variable 1, FFF.F.F, 0, 0, 1, no-units, FF, 0.000000, 0.000000, 0.000000, W, 0.000000, F, ...

Object_Name=A001

Description=Analog Variable 1

Event_Enable=FFF

Issue_Confirmed_Notifications (not used)=F

Ack_Required (not used)=F

Time_Delay=0

Notification_Class=0

Priority (not used)=1

Units=no-units

Limit_Enable=FF

High_Limit=0.000000

Low_Limit=0.000000

Deadband=0.000000

COV_Increment=0.000000

ReadWriteCommandable=W

©2005-2011 Carel SpA, All Rights Reserved

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, Binary Value and Notification Class Objects selected by the **AV1 to AV207**, **AV/MV1001 to AV/MV1207**, **BV1 to BV207** and **NC1 to NC16** 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. *Please note that because of their complex data structures, Schedule and Calendar objects cannot be Uploaded or Backed up.*

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):
 - o *analog-value,####*
 - o *binary-value,####*
 - o *multi-state-value,####*
 - o *notification-class,####*
 - o *Analog_Value,####*
 - o *Binary_Value,####*
 - o *Multi_State_Value,####*
 - o *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,Description,Event_Enable,
Issue_Confirmed_Notifications,Ack_Required,Time_Delay,Notification_Class,Priority,
Units,Limit_Enable,High_Limit,Low_Limit,Deadband,COV_Increment,
ReadWriteCommandable,Relinquish_Default,Alarm_Custom,
Alarm_Property,Alarm_OutOfRange,Alarm_Normal,Alarm_Fault,
Alarm_HighLimit,Alarm_LowLimit
analog-value,1,A001,Analog Variable 1,
FFF,F,F,0,0,1,
no-units,FF,0.000000,0.000000,0.000000, 0.000000,
W,0.000000,F
"""
analog-value,2,A002,Analog Variable 2,
FFF,F,F,0,0,1,
no-units,FF,0.000000,0.000000,0.000000, 0.000000,
W,0.000000
Object_Type,Object_Instance,Object_Name,Description,Event_Enable,
Issue_Confirmed_Notifications,Ack_Required,Time_Delay,Notification_Class,Priority,
Number_Of_States,Alarm_Values,Fault_Values,ReadWriteCommandable
Relinquish_Default,Alarm_Custom,Alarm_Property,Alarm_ChangeofState,
Alarm_Normal,Alarm_Fault,Alarm_OffNormal,
State_Text
multi-state-value,1001,I001,Integer Variable 1,FFF,
F,F,0,0,1,
4,"3","4",W,
0,F,
"""
"1=State1,2=State2,3=State2,4=State4"
multi-state-value,1002,I002,Integer Variable 2,FFF,
F,F,0,0,1,
6,"3,4","5,6",,W,
0,F,
"""
"1=STATE1,2=STATE2,3=STATE3,4=STATE4,5=STATE5,6=STATE6"
Object_Type,Object_Instance,Object_Name,Description,Event_Enable,
Issue_Confirmed_Notifications,Ack_Required,Time_Delay,Notification_Class,Priority,
Inactive_Text,Active_Text,Alarm_Value,ReadWriteCommandable,Relinquish_Default
Alarm_Custom,
Alarm_Property,Alarm_ChangeofState,Alarm_Normal,Alarm_Fault,Alarm_OffNormal
binary-value,1,D001,Digital Variable 1,FFF,F,F,0,0,1,
Off,On,0,W,0,F,
"""
binary-value,2,D002,Digital Variable 2,FFF,F,F,0,0,1,
Off,On,0,W,0,F,
"""
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 the [blue lines](#) are *template lines* and the remaining lines 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. Please note that line breaks have been added to the above lines for readability. Actual CSV files should contain only one line break.

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

<i>Template Line String (i.e. Property Name)</i>	<i>Corresponding Object Line String</i>	<i>Valid for</i>
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 or 100001-102046 if MODBUS Extended	AV
Object_Instance	1001 to 1207 or 200001-202046 if MODBUS Extended	AV, MV
Object_Instance	1-207 or 100001-102046 if MODBUS Extended	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 Or 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
High_Limit, Low_Limit, Deadband	x.xxxxx (floating point value)	AV
Relinquish_Default	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
Relinquish_Default	0 or 1	BV
Number_of_States	1 to 32	MV
State_Text	Up to 32 comma-separated strings of the form	MV

	<i>statenumber=StateText</i> . The entire string must be enclosed in double quotes (e.g. "1=State 1,2=State 2,3=State3") **																	
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																
Relinquish_Default	1 to 32	MV																
ReadWriteCommandable	R, W or C	AV, BV, MV																
#xxxxx (where xxxxx = the proprietary property number, e.g. #3000)	Depends on the property	AV, BV, MV																
Alarm_Custom	T or F or 0 or 1 indicating that there are custom Alarm strings for the object	AV, BV, MV																
Alarm_Property	Object_Name or Description indicating the property to be used in the Alarm Message for the object.	AV, BV, MV																
Alarm_ChangeofState	Substitute text for "ChangeOfState" in Alarm messages	BV, MV																
Alarm_OutOfRange	Substitute text for "OutOfRange" in Alarm messages	AV																
Alarm_Normal	Substitute text for "Normal" in Alarm messages	AV, BV, MV																
Alarm_Fault	Substitute text for "Fault" in Alarm messages	AV, BV, MV																
Alarm_OffNormal	Substitute text for "Offnormal" in Alarm messages	BV, MV																
Alarm_HighLimit	Substitute text for "High Limit" in Alarm messages	AV																
Alarm_LowLimit	Substitute text for "Low Limit" in Alarm messages	AV																
Ack_Required	T or F or 0 or 1	NC																
Priority	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").	NC																
Recipient_List	A comma-separated string consisting of components of the Recipient_List property. The components are identified by <i>parameter name=value</i>. The entire string must be enclosed in double quotes. <table><tr><td><i>Parameter name</i></td><td><i>values</i></td></tr><tr><td>validDays</td><td>MTWTFSS (replace day of week with a dash (-) to disable that day (e.g. M-W-F-S)</td></tr><tr><td>fromTime</td><td>00:00:00.00 to 23:59:59.99</td></tr><tr><td>toTime</td><td>00:00:00.00 to 23:59:59.99</td></tr><tr><td>processIdentifier</td><td>0 to 4294967295</td></tr><tr><td>IssueConfirmed</td><td>T or F or 0 or 1</td></tr><tr><td>Notifications</td><td></td></tr><tr><td>transitions</td><td>xxx where x=T or F</td></tr></table>	<i>Parameter name</i>	<i>values</i>	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	T or F or 0 or 1	Notifications		transitions	xxx where x=T or F	NC
<i>Parameter name</i>	<i>values</i>																	
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	T or F or 0 or 1																	
Notifications																		
transitions	xxx where x=T or F																	

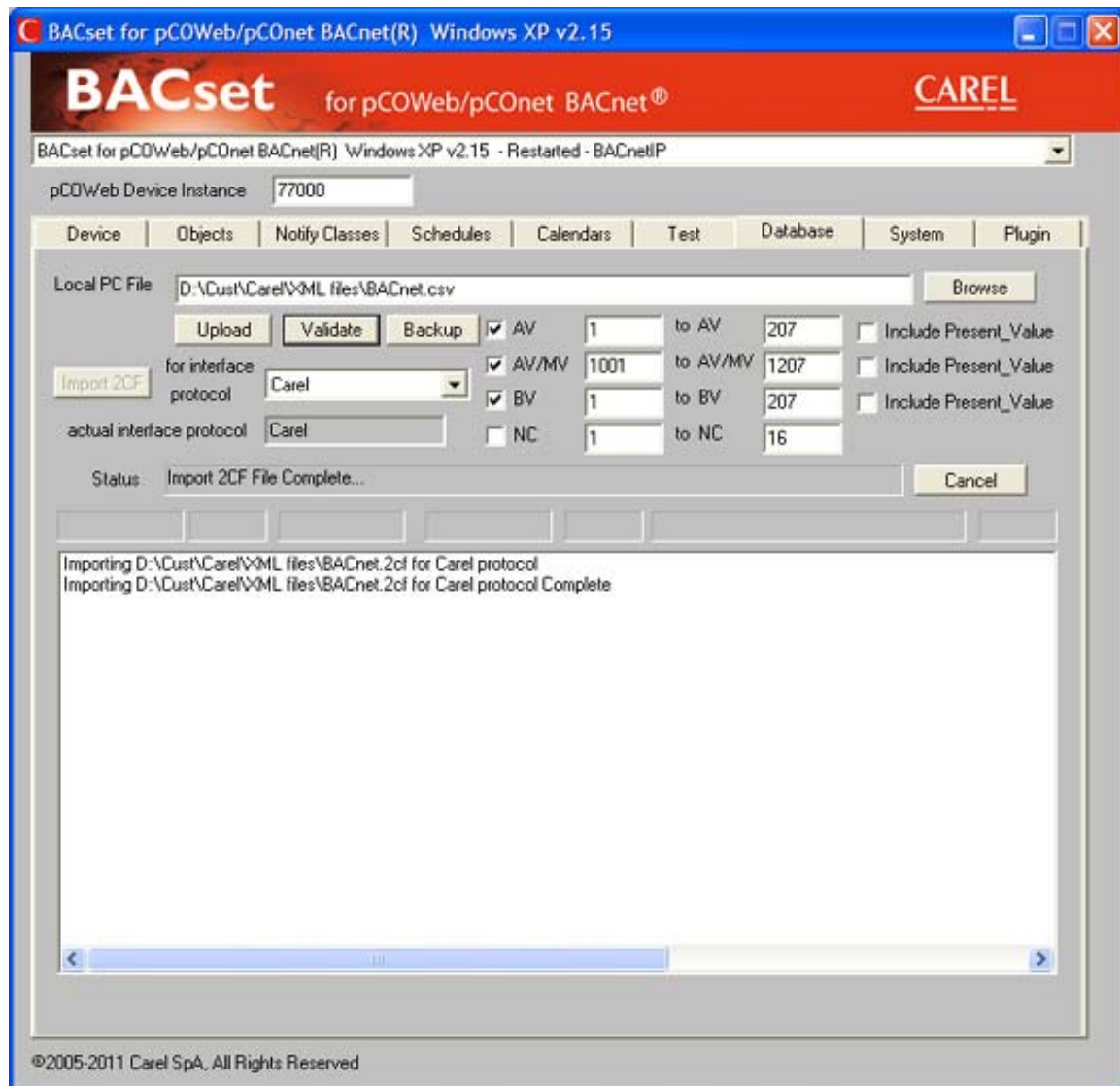
	recipient DExxxx for devices or 0xHHHHHHHH for addresses Example: ("validDays=MTWTFSS, fromTime=00:00:00.00, toTime=23:59:59.99,processIdentifier=100, IssueConfirmedNotifications=F, transitions=TFT,recipient=DE1000"). Note "" indicates no recipient, i.e. an empty list)	
--	--	--

* 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.**

*** Note: These properties are no valid for pCOWeb/pCONet Application_Software_Version 2.13 and later.

Importing 2CF Database Files



BACset allows the importation of XML files in the Carel 2CF format. The result of the Import is a BACset CSV file, which can be managed in the same manner as any other CSV file. The Carel 2CF format is not described in this document.

System Tab

The screenshot shows the 'System' tab of the BACset software. The title bar reads 'BACset for pCOWeb/pCOnet BACnet(R) Windows XP v2.15'. The main header has 'BACset' in large red letters and 'for pCOWeb/pCOnet BACnet®' in smaller text, with the 'CAREL' logo on the right. Below the header is a dropdown menu showing 'BACset for pCOWeb/pCOnet BACnet(R) Windows XP v2.15 - Restarted - BACnetIP'. A text field for 'pCOWeb Device Instance' contains the value '77000'. A series of tabs at the top include 'Device', 'Objects', 'Notify Classes', 'Schedules', 'Calendars', 'Test', 'Database', 'System' (which is active), and 'Plugin'. The 'System' tab content includes: 'Firmware Version' and 'Firmware Release: A1.2.10.215.2 - B1.2.1' with a 'Read Version' button; a 'Status' section with 'bytes' and '%' fields, a 'Reboot' button, and a 'Password' field; a 'Flash Size Query Complete' message with a 'Cancel' button; an 'Upload' button and a 'Device's Flash File' section with radio buttons for 'flash_apps.bin' (selected) and 'flash_sys.bin'; a 'Local PC File' text field with a 'Browse' button; a large empty rectangular area; 'View Flash' and 'View Memory' buttons; a 'User FlashDisk usage statistic' table; and 'Interface Protocol Type Factory Settings' and 'Carel/MODBUS Baud Factory Settings' sections with dropdown menus and buttons.

Filesystem	Size	Used	Avail	Use%	Mounted on
/initrd/dev/mtdblock3	4.8M	1.9M	2.9M	40%	/usr/flashdisk

Interface Protocol Type Factory Settings: Carel [Read] [Write] [Factory]
Carel/MODBUS Baud Factory Settings: 19200 [Must Reboot]

©2005-2011 Carel SpA. All Rights Reserved

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.

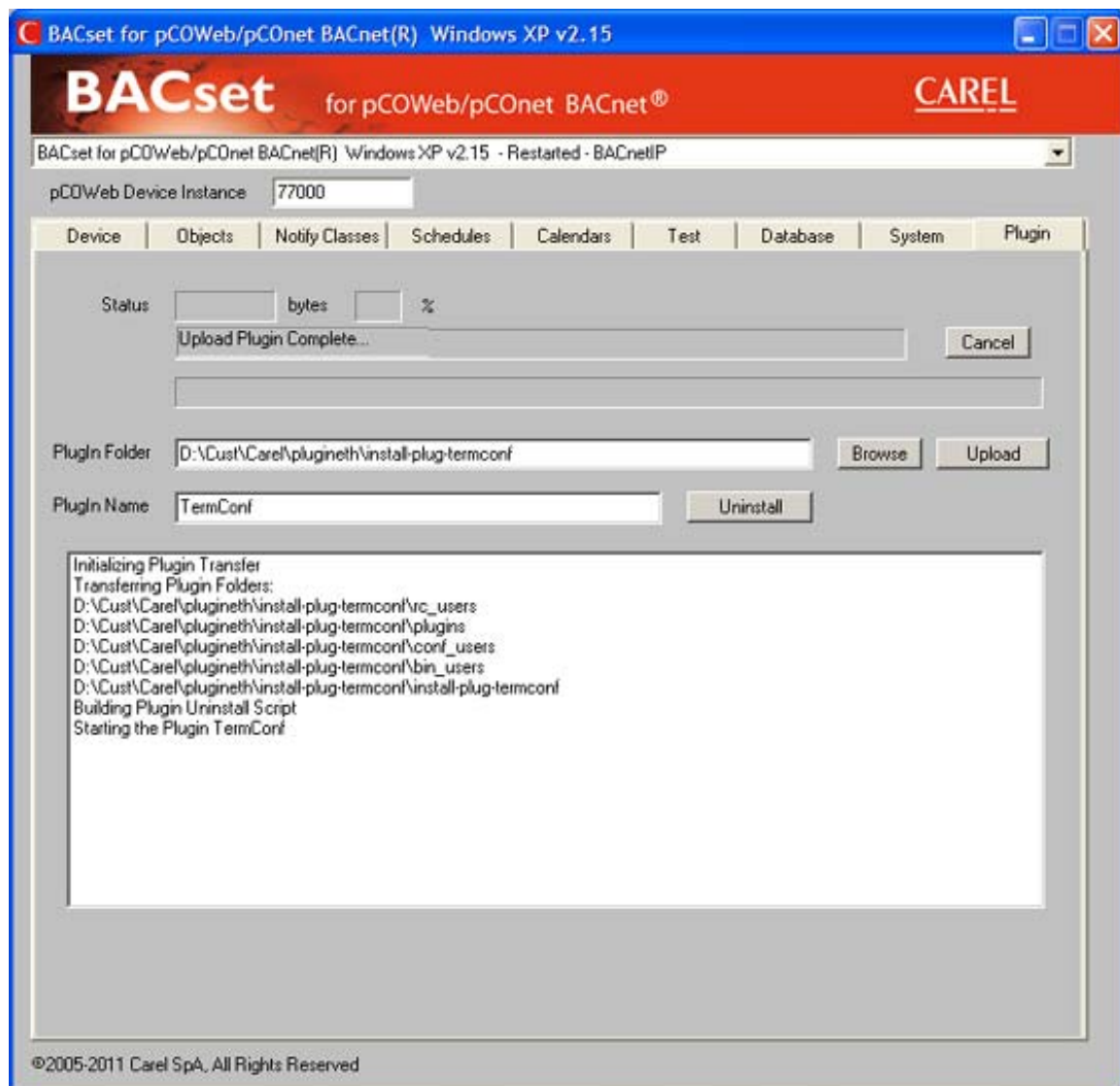
Clicking **View Flash** displays the Flash Memory statistics for the pCOWeb/pCOnet. Clicking **View Memory** displays the RAM memory statistics for the pCOWeb/pCOnet.

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.

Clicking **Read** at the bottom of the form reads the current Interface Protocol and its baud rate between the pCOWeb/pCOnet and the Carel controller that it's plugged into. Clicking **Write** allows you to change the Interface Protocol and/or the baud rate. By default this Interface is pCO and the baud rate is 19200. Please note, this is **not** the BACnet MS/TP baud rate.

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.

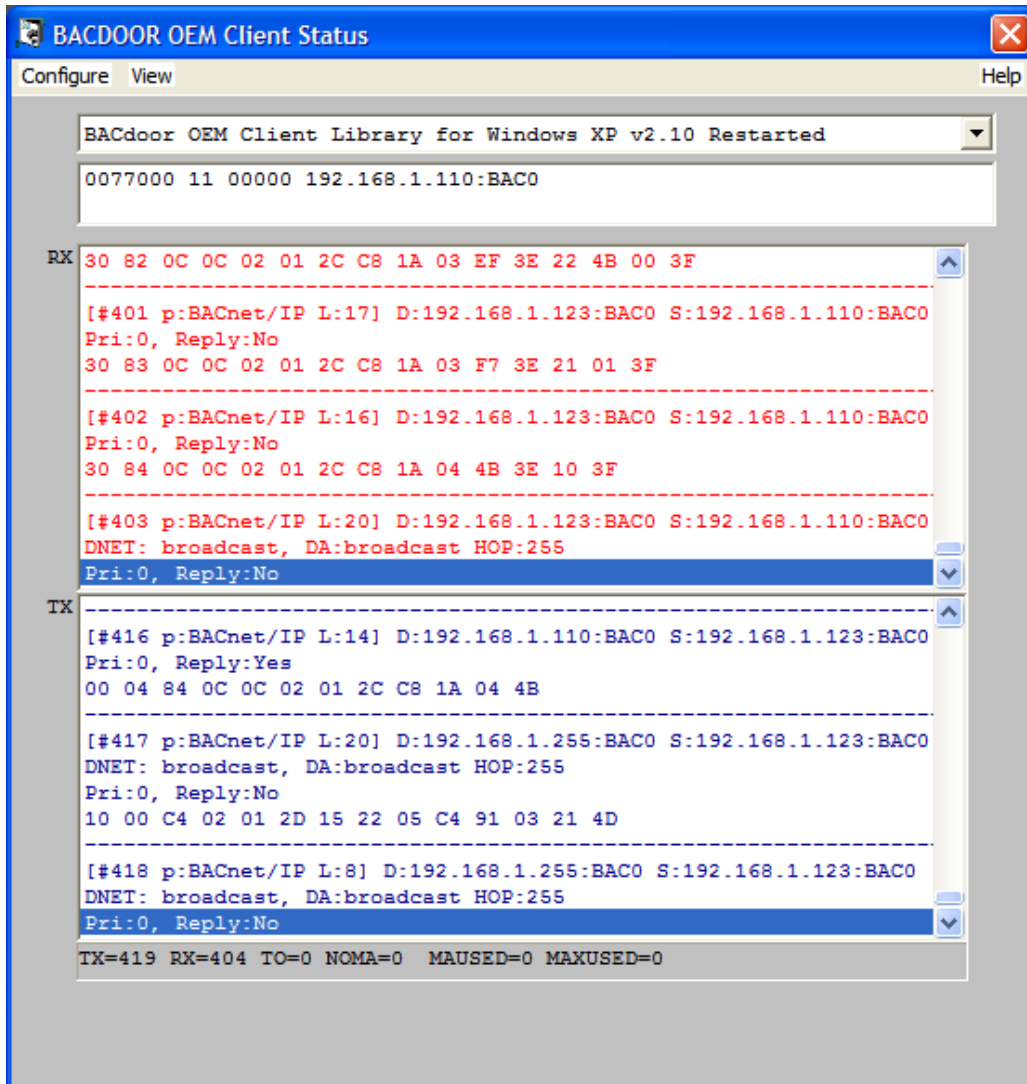
Clicking **Upload** searches the entire folder and all its subfolders that is selected in the **Plugin Folder** edit box and uploads all the files contained in them to the pCOWeb/pCOnet. Following the upload, a request is made to the pCOWeb/pCOnet to start the plugin application.

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.

The screenshot shows a Windows-style dialog box titled "BACDOOR OEM Client Library Configuration". It contains several input fields and sections:

- Our Device Instance:** 77077
- Our Object Name:** BACLIB
- Our Description:** BACdoor Device Object
- Our Location:** unknown
- Whols/IAm Interval:** 1 Minutes (0=None)
- BACnet/IP Parameters:**
 - IP Address: 192.168.1.111 [255.255.255.0] Intel(R) PRO/1000 CT Network Connection (dropdown)
 - UDP port: 0xBAC0
 - Subnet: 255.255.255.0
- MS/TP Parameters:**
 - Com Port (restart): 1
 - Baud (restart): 9600 (dropdown) (no parity, 8 data, 1 stop)
 - TS (MS/TP Node): 1
 - MaxMaster: 2
 - MaxInfoFrames: 10

Buttons for "OK" and "Cancel" are located on the right side of the dialog.

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.

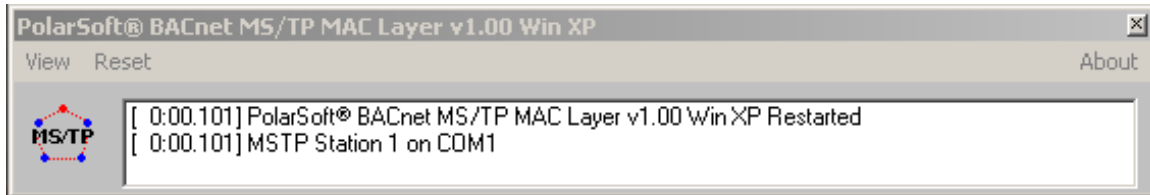
If you change the MS/TP baud rate, you must restart BACset for it to take effect.

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.

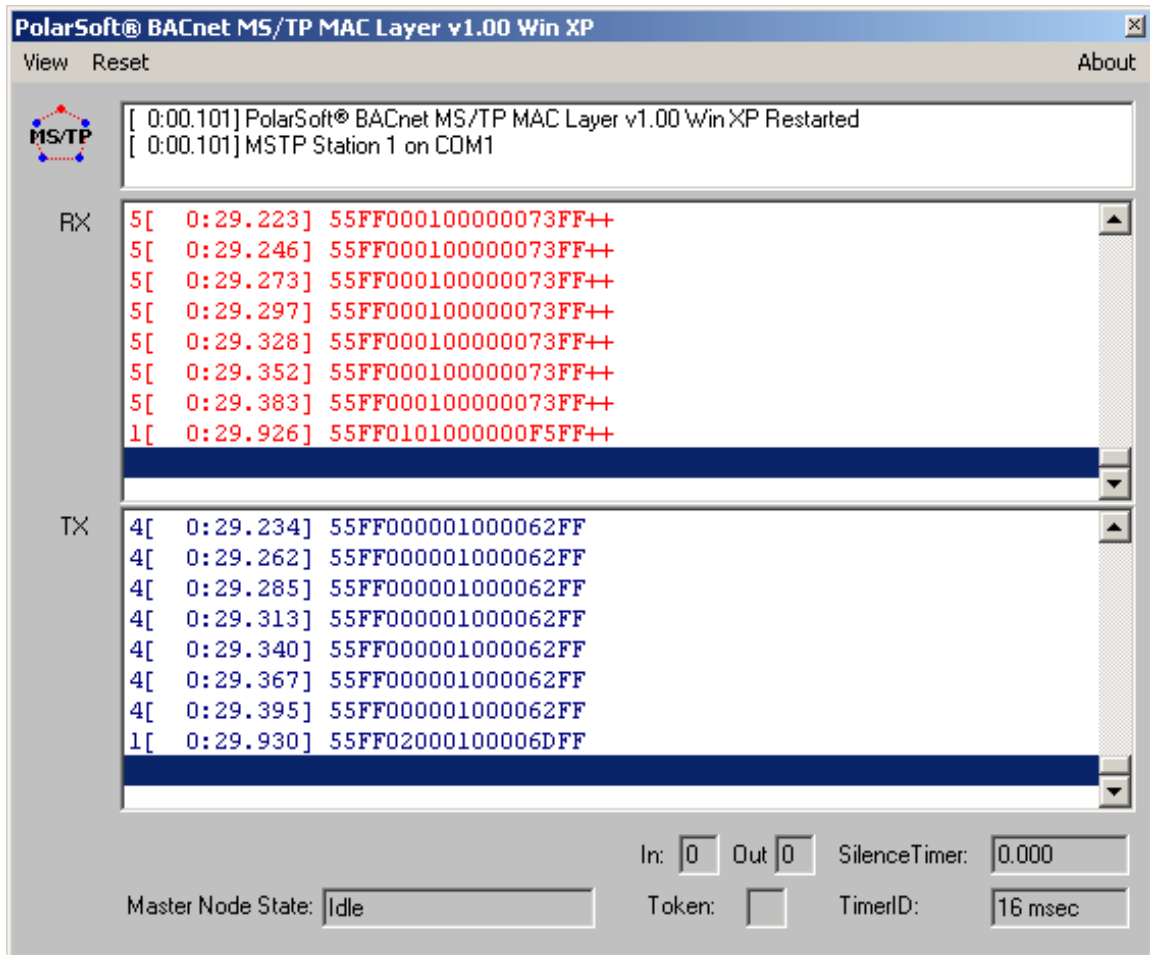
MS/TP MAC Layer Status Window

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 interface protocol or the MODBUS Extended interface protocol in the Carel controller. If the Carel interface protocol is used, **CBG** maps a maximum 207 pCO Analog points, a maximum 207 pCO Integer points and a maximum 207 pCO Digital points from the Carel controller in the following manner:

- Analog points 1 to 207 *are mapped to* BACnet Analog Values 1 to 207
- Integer points 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 Digital points 1 to 207 *are mapped to* BACnet Binary Values 1 to 207

If the MODBUS Extended interface protocol is used, **CBG** maps a maximum 2048 Analog points, a maximum 2048 Integer points and a maximum 2048 Digital points from the Carel controller in the following manner:

- Analog points 1 to 2048 *are mapped to* BACnet Analog Values 100001 to 102048
- Integer points 1 to 207 *are mapped to* BACnet Analog Values 200001 to 202048 but may be reprogrammed as Multi-state Values on an individual basis
- Digital points 1 to 2048 *are mapped to* BACnet Binary Values 100001 to 102048

From the factory, there are a total of 621 Analog Values, Multi-state Values and Binary Values in every pCOWeb/pConet and the interface protocol is pCO. If the actual number of pCO points in-service is less than 621, the actual number of each 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 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^{1,2}
- Process_Identifier^{1,2}

Notes:

1. Indicates properties that are writable using BACnet writeProperty and writePropertyMultiple services through a BACnet client.
2. These properties are not supported in pCOWeb/pCOnet in Application_Software_Version 2.13 and later

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
- Priority_Array (if commandable)
- Relinquish_Default * (if commandable)
- Proprietary Property 1014 * (0=read-only, 1=writable but not commandable. 2=commandable)

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 ^{1,2}
- Issue_Confirmed_Notifications ^{1,2}
- Priority ^{1,2}

Notes:

1. Indicates properties that are writable using BACnet writeProperty and writePropertyMultiple services through a BACnet client.
2. These properties are not supported in pCOWeb/pCOnet in Application_Software_Version 2.13 and later

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
- Priority_Array (if commandable)
- Relinquish_Default * (if commandable)
- Proprietary Property 1014 * (0=read-only, 1=writable but not commandable.
2=commandable)

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 ^{1,2}
- Issue_Confirmed_Notifications ^{1,2}
- Priority ^{1,2}

Notes:

1. Indicates properties that are writable using BACnet writeProperty and writePropertyMultiple services through a BACnet client.
2. These properties are not supported in pCOWeb/pCOnet in Application_Software_Version 2.13 and later

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 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
- Priority_Array (if commandable)
- Relinquish_Default * (if commandable)
- Proprietary Property 1014 * (0=read-only, 1=writable but not commandable. 2=commandable)

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 ^{1,2}
- Issue_Confirmed_Notifications ^{1,2}
- Priority ^{1,2}

Notes:

1. Indicates properties that are writable using BACnet writeProperty and writePropertyMultiple services through a BACnet client.
2. These properties are not supported in pCOWeb/pCOnet in Application_Software_Version 2.13 and later

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 Objects

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

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

- Alarm_Priority
- Ack_Required
- Recipient_List

pCOWeb/pCOnet BACnet Object Properties for Schedule Objects

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

- Description *
- Both Weekly_Schedule * and Exception_Schedule * properties are supported

pCOWeb/pCOnet BACnet Object Properties for Calendar Objects

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

- Description *