pCOWeb for the pCO controllers

PRELIMINARY

Manuale d'uso

User manual

CAREL

Technology & Evolution

We wish to save you time and money!
We can assure you that the thorough reading of this manual will guarantee correct installation and safe use of the product described

OVERVIEW ..... - 5 -
HARDWARE INSTALLATION ..... - 5 -
PCOWEB ADMINISTRATION ..... - 5 -
DEFAULT WEB PAGE ..... - 5 -
WEB ADMINISTRATOR PAGE ..... - 6 -
View used/free flashdisk space: ..... - 7 -
Set directory password: ..... - 7 -
Adjust files and directories attributes access: ..... - 7 -
Flush disk cache: ..... - 7 -
Backup / Upload flash images: ..... - 8 -
Format flashdisk: ..... - 8 -
Get/Set var: ..... - 8 -
HTTP PROTOCOL ..... - 8 -
INTRODUCTION ..... - 8 -
USER HTML PAGES ..... - 8 -
Note – Changing Passwords: ..... - 9 -
SPECIAL TAGS ..... - 10 -
Getting Data ..... - 10 -
Setting Data ..... - 11 -
FTP PROTOCOL (OPTIONAL) ..... - 12 -
SNMP PROTOCOL ..... - 14 -
GENERAL OID FORMAT: ..... - 14 -
Note – SNMP R/W: ..... - 14 -
TRAPS ..... - 14 -
Introduction ..... - 14 -
SYSTEM TRAPS ..... - 15 -
Default traps ..... - 15 -
Default trap community ..... - 15 -
Send authfail trap ..... - 15 -
System traps hosts table ..... - 15 -
System trap destination host #N ..... - 15 -
Note – Trap Host IP: ..... - 15 -
CONTROLLER TRAPS ..... - 15 -
Top Half of Trap setup Page ..... - 16 -
Note – Trap Host IP: ..... - 17 -
System traps ..... - 17 -
Enabled ..... - 17 -
Trap OID ..... - 17 -
Acknowledge and Ack interval ..... - 17 -
Bottom Half of Trap Setup Page ..... - 18 -
Variable: 1 ↓SELECT ..... - 18 -
Enabled and Destination Columns ..... - 18 -
Trigger ..... - 18 -
Trap OID ..... - 19 -
Acknowledge and Ack interval ..... - 19 -
Vars OID (optional) ..... - 19 -
BACNET PROTOCOL ..... - 20 -
BACNET/IP AND BACNET ETHERNET CONFIGURATION ..... - 20 -
Device Properties: ..... - 21 -
Alarm Parameters: ..... - 22 -
Clock Parameters: ..... - 22 -

Submit Button: ..... - 22 -
PCO* TO BACNET DATABASE ..... - 22 -
SUPPORTED BACNET OBJECTS ..... - 23 -
BIBB SUPPORT ..... - 25 -
OBJECT SUPPORT OF INTRINSIC ALARMING ..... - 25 -

Overview

The pCOWEB interface card was designed to provide the pco* controller family connectivity to four increasingly used protocols: HTTP SERVER, FTP, SNMP, and BACNET. The card provides a "gateway" connection between the communicated variables from the controller allowing all 199 digital, 127 integer and 127 analogue variables to be visible to the end user.

Hardware Installation

The pCOWEB interface card connects to the pco* by the supervisory port. The card can be inserted directly into the controller.

Figure 1 – pCOWeb Ethernet Card

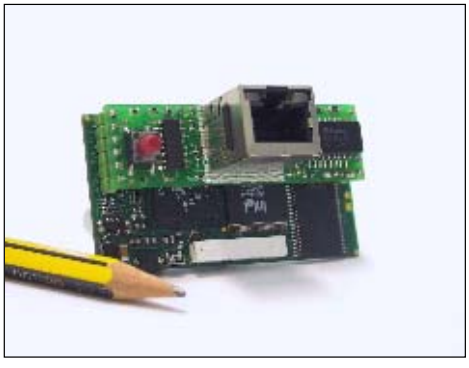


Figure 2 – Remove Cover

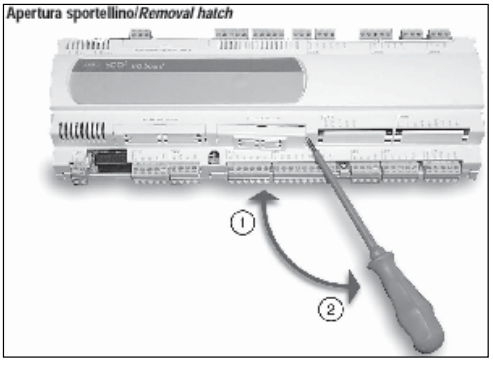


Figure 3 – Insert the pCOWeb Card

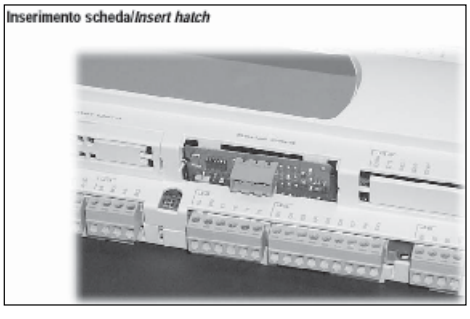
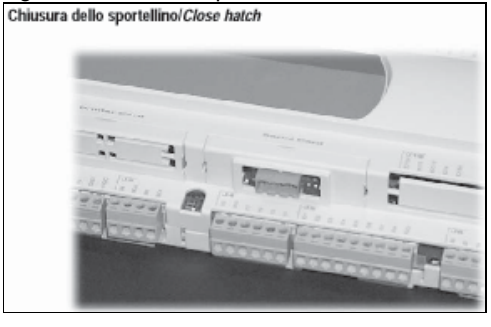


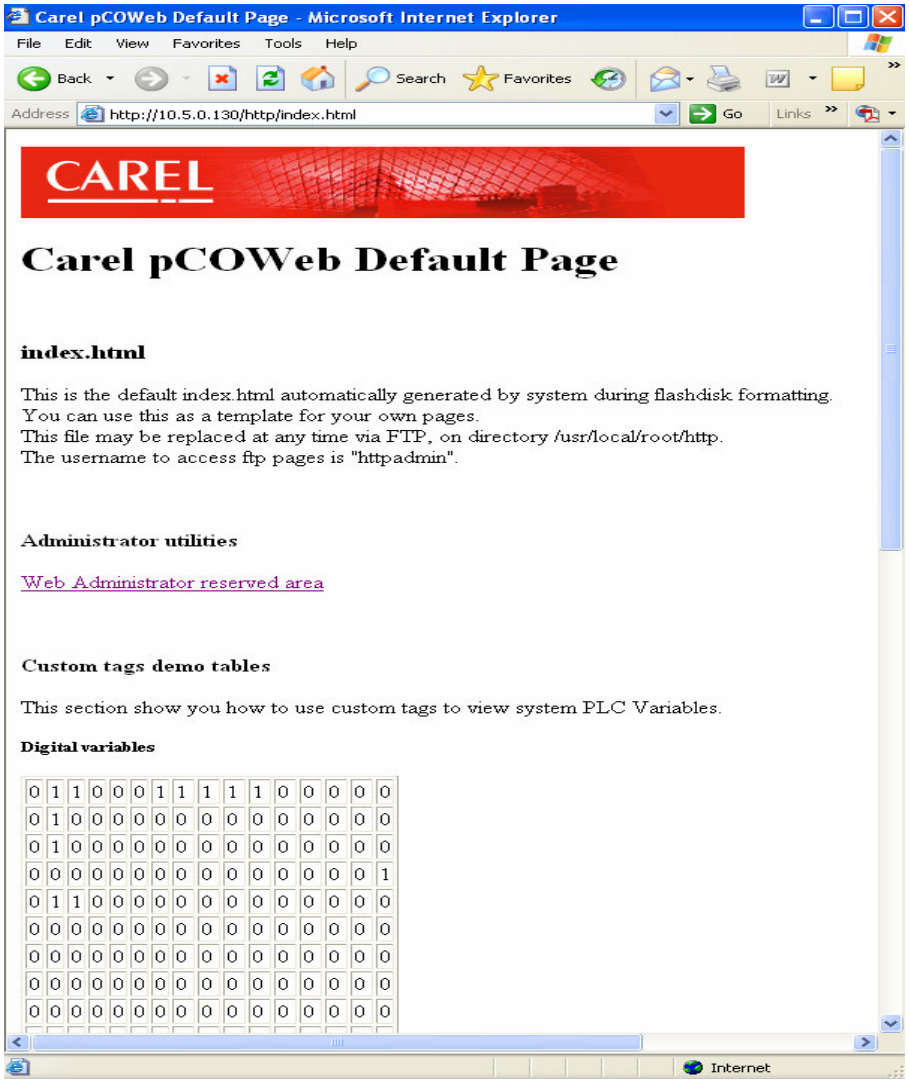
Figure 4 – Inset the pCOWeb Cover



pCOWeb Administration

Default Web Page

The pCOWEB defaults to DHCP for its IP address. If this is not desired, you can force the address 172.16.0.1 by holding down the reset button during power-up until the LED blinks red. Connect to the pCOWEB with a browser and you will be automatically redirected to the standard default page shown below.



This is the default index.html page. It provides you with a link to the administrator reserved area and displays the current values of all variables that are being read from the controller. This file is stored in **`usr/local/root/flash/http/`** and is also the folder where your custom html files will be stored.

Web Administrator Page

The adminpage.html gives some useful configuration features.

View used/free flashdisk space:
Check for the amount of free writable space inside the FLASH memory.

Set directory password*:*
Useful to set/change the logins to access the directory in the writable area of the FLASH.

Adjust files and directories attributes access:
Needed to set the properties of the user html and cgi files downloaded into pCOWeb; you must launch this item to make the http server able to open those user files. Every time you create a new page or inadventertly change the properties of some user files, it is advisable to launch this item.

Flush disk cache:
Every change in the user files or parameters is temporarily written in RAM and later saved in FLASH. This is made to save time for the writing cycles in FLASH. This can lead to data

loss or corruption if pCOWeb is powered down before the saving in FLASH. This button lets you to force the save.

Backup / Upload flash images:
These two items lets you to download an exact copy of the flashdisk to your PC and later to upload again into a pCOWeb. The previous contents of the target pCOWeb flaskdisk will be lost.
Note that the flashdisk contains also all the user specified configuration parameters (IP address, for instance).
The Upload item lets you also to upgrade the firmware into pCOWeb by picking-up the proper files from new releases from Carel. The Upload task will automatically understand what type the block is and will write it into the right place.

Format flashdisk:
This item will erase ALL the writable FLASH space and will recreate the factory situation inside it. It is intended to recover from corruption of flash file system or to bring the flash disk to the factory situation, but you can also choose to keep the user configuration parameters (not the html pages): in this case the parameters will be first saved in a temporary RAM zone and further moved to the FLASH after reformatting.

Get/Set var:
It is a demo page that lets you to set some digital / analogue / integer variables.

HTTP PROTOCOL

Introduction

Creating custom html pages for your pCOWEB requires a two step process, downloading the new pages using FTP and setting the proper access properties. Custom pages are stored in the writable space of the FLASH memory. The amount flash available will very between versions, but will be around 2.5mb to 4mb. For the purpose of this manual, we will use the SmartFTP program (http://www.smartftp.com) to easily download the pages to the pCOWEB via FTP protocol.

The http server of pCOWeb needs to have the html pages with some specific owner/group/access property settings. After the pages are downloaded, a special task has to be launched to set properties. If these properties are not set, it will not be possible to retrieve the pages.

User html pages

The process for updating pages in the pcoweb is:

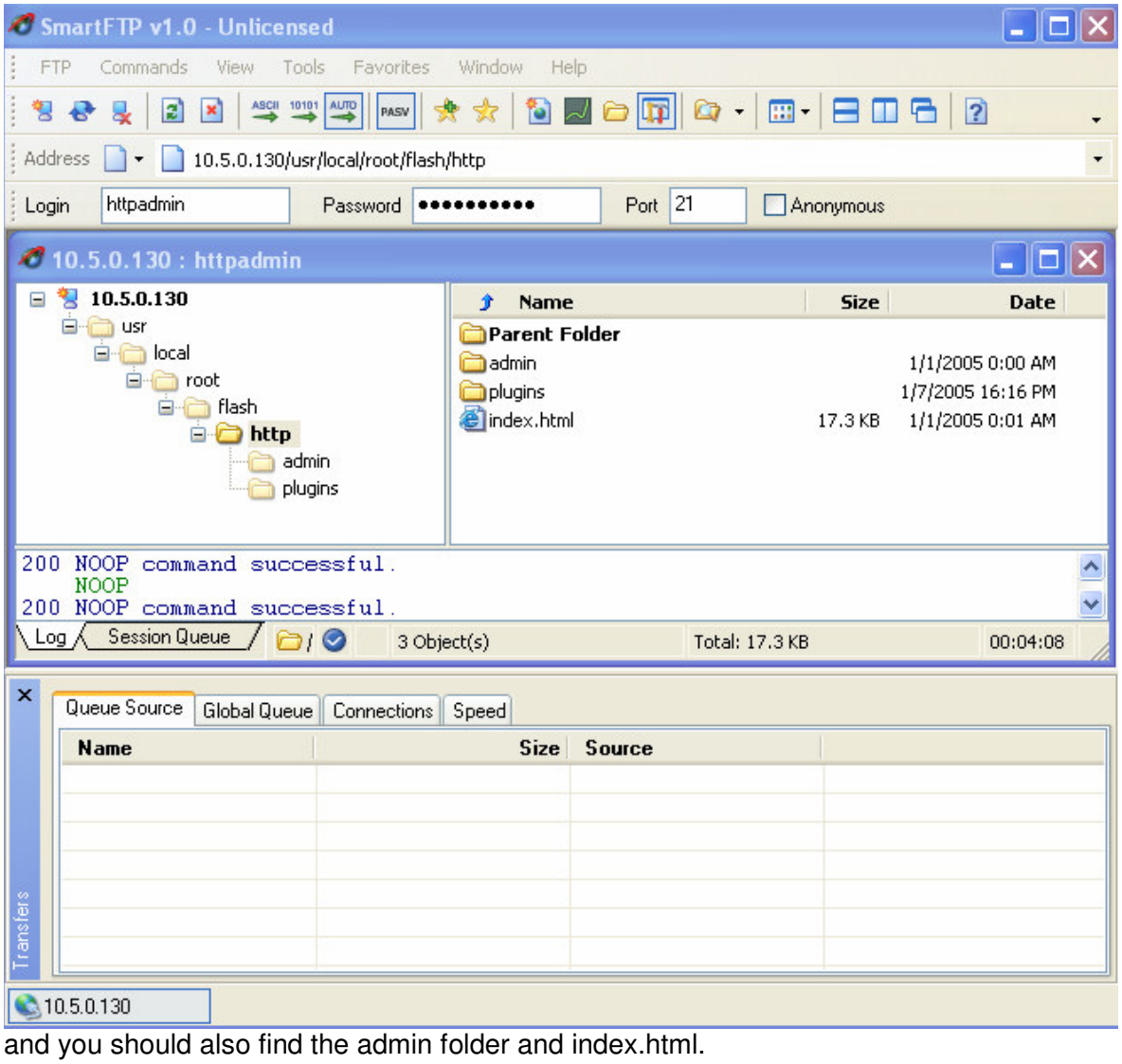
1. Rename the default index.html page with another name, for example admin.html.
2. Download a custom made index.html page + any other page linked to it. These pages are not visible at this point, the proper flags need to be set.
3. To set the flags, link to the administrator page, previously renamed, by typing: http://your_pcoweb_ip/http/admin.html.
4. Enter the Web Administrator Area with user name: “admin” and password: “fadmin”. Launch the “Adjust files and directories attributes access”.
5. Repeat steps 2-4 every time a page is added or updated.

To administer the web pages you can use any FTP client, for the following examples we will use SmartFTP (free) to explore the directory tree of pCOWeb. The user name and password for web administration is:

username: ***httpadmin***
password: ***fhttpadmin***

Note – Changing Passwords:
The password can be changed by accessing the pCOWeb html configuration pages. The factory password is simply "f" followed by the username. The username cannot be changed.

After the login you will see the contents of the folder:
`/usr/local/root/flash/http`



and you should also find the admin folder and index.html.

The /flash/http directory is the right place to store your html pages and/or directories. The /flash/http/index.html is the default page that will be served out when you type http://<pCOWeb IP address or name to be resolved by a DNS>[Enter] into a browser.

If you need to build some CGI scripts, the right place for those files is: `/flash/usr-cgi/`. If you put them in a place other than that, they will not be executed.

The directory `/usr/local/root/` is the root directory for http server. The http server will automatically add the path `/usr/local/root/` to every page name requested by a client. For example, look at "index.html":

When you write: http://<pCOWeb IP address or name to be resolved by a DNS>[Enter], the http server will automatically serve the following page: `(/usr/local/root/)/http/index.html` The parenthesis contain the root path for the http server; on future examples this will be omitted. This page is stored inside a read-only area of the FLASH memory. If you open this page with a text editor, you'll find inside a redirection to another page: `/http/index.html`.
The http server will then search this new page inside the directory: /http. Actually this directory is a link to the directory flash/http: the http server will then look into: /flash/http. The /flash area is a writable space inside the FLASH memory. When you first receive pCOWeb, or after a "Format flashdisk" action, this directory will contain a factory index.html, but you can change it as you want.

Special Tags

Look into `/flash/http/index.html` you'll find some interesting rows:

index.html

```
1  <html>
2  <!-- The tag filter musb be specified in the first 10 lines,
3       after the '<html>' statment, with the sintax: -->
4  <!--tagparser="/pcotagfilt"-->
5  <title>Carel pCOWeb Default Page</title>
```

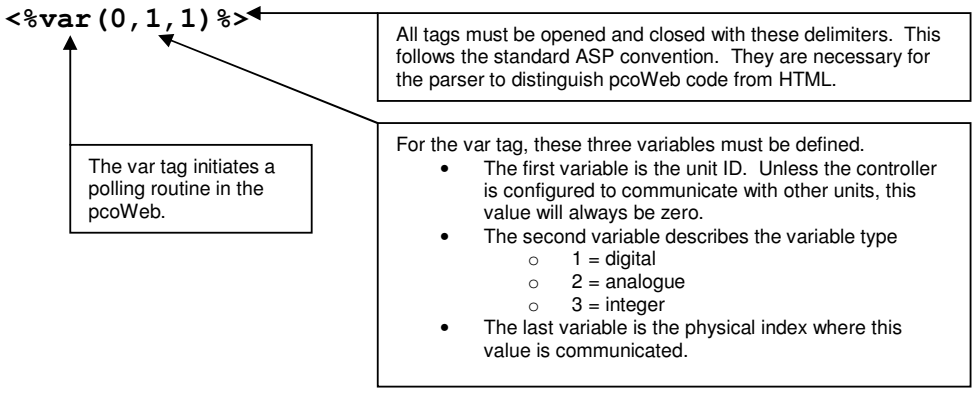
<!--tagparser="/pcotagfilt"-->

This row must be inserted within the first 10 lines: it tells to http server to call the /pcotagfilt pre-process application (we call it "filter") that will recognize all the Carel-defined tags and will perform the corresponding actions. If this row is not present the Carel-defined tags will be ignored and no values of the pCO controller will be visible.

```
40 <H4>Custom tags demo tables</H4>
41 This section show you how to use custom tags to view system PLC Variables
42 <h5>Digital variables</h5>
43 <table border=1>
44 <tr><td><%var(0,1,1)%></td><td><%var(0,1,2)%></td><td><%var(0,1,3)%></td>
45 <tr><td><%var(0,1,17)%></td><td><%var(0,1,18)%></td><td><%var(0,1,19)%></
```

Line 44 and 45 above show several Carel-defined tags: they are useful to retrieve the actual value of the pCO variables. For the syntax and details, see the list of the tags below.

Getting Data



The other data retrieval method is the **`setres`** tag.

This tag acts as a set-var verification system. When you change a system variable through the pcoWeb, a value can be communicated back to the browser. This value indicates whether the previous operation was successful.
In the following example, the **`setres`** operation will display an image based on the result:

```
<img src="img/<%setres('undefined','green','red')%>.gif"…
```

In this example, a different image is displayed based on the value of **`setres`**. This function is only called after the pcoWeb initiates a SET command. The parameters of the tag are booleans, and are mutually exclusive. The first parameter returning true means that the SET function returned 0, or a null result. This could be caused by communication errors with the pcoWeb or incorrect variable settings. The second parameter returns true when the variable has been set. The third parameter returns true when the transaction times out. The variable may have in fact been changed, but the transaction exceeded the timeout setting. This may be encountered if the pcoWeb resides on a network instead of a direct connection.

Setting Data

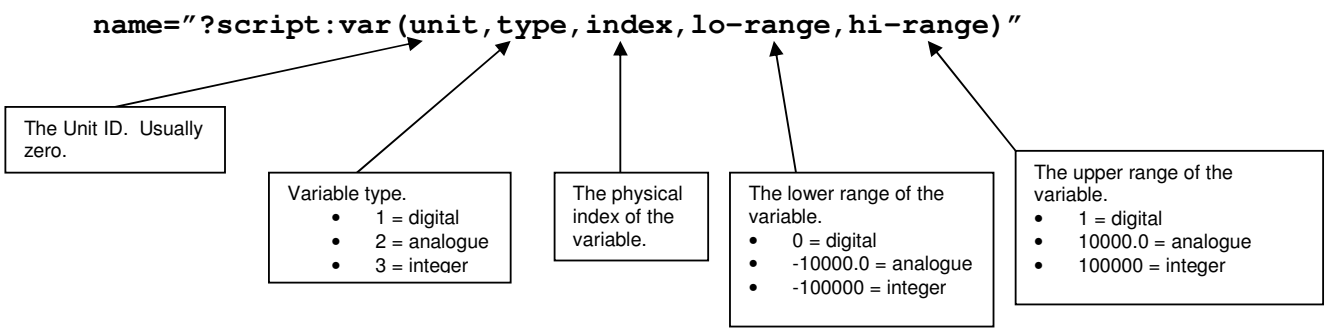
Setting data to the pcoWeb uses the standard HTML form submission methods. The standard form tag for pcoWeb must only contain the '`method`' property.

Example:
`<form method="get">`

The '`action`' tag should be omitted in the pcoWeb pages.

```
36 <form method="GET">
37 Var 1: <input type="text" name="?script:var(0,1,1,0,1)" value="<%var(0,1,1)%>"><br>
38 Var 2: <input type="text" name="?script:var(0,1,2,0,1)" value="<%var(0,1,2)%>"><br>
39 Var 3: <input type="text" name="?script:var(0,1,3,0,1)" value="<%var(0,1,3)%>"><br>
40 Var 4: <input type="text" name="?script:var(0,1,4,0,1)" value="<%var(0,1,4)%>"><br>
41 Var 5: <input type="text" name="?script:var(0,1,5,0,1)" value="<%var(0,1,5)%>"><br>
42 Var 6: <input type="text" name="?script:var(0,1,6,0,1)" value="<%var(0,1,6)%>"><br>
```

Setting a variable must follow this format:

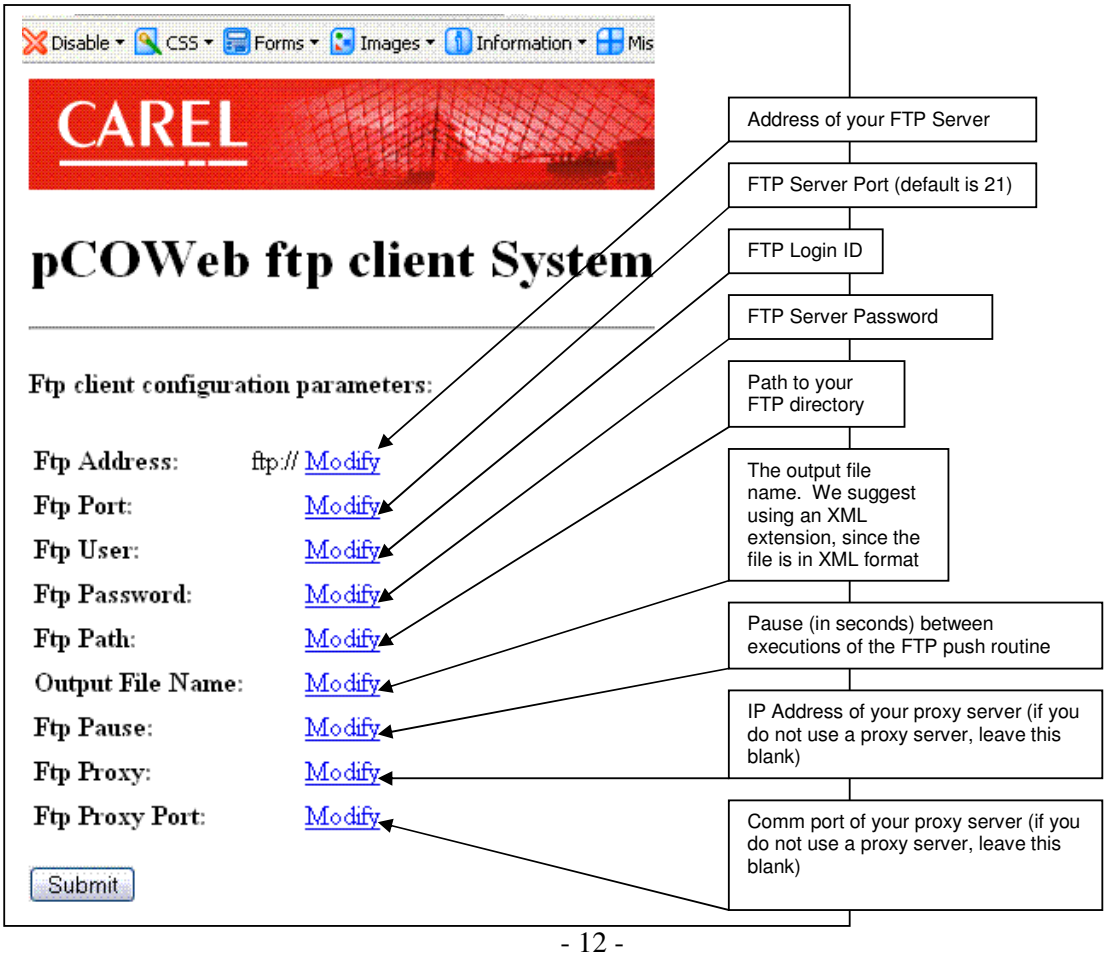


FTP PROTOCOL (optional)

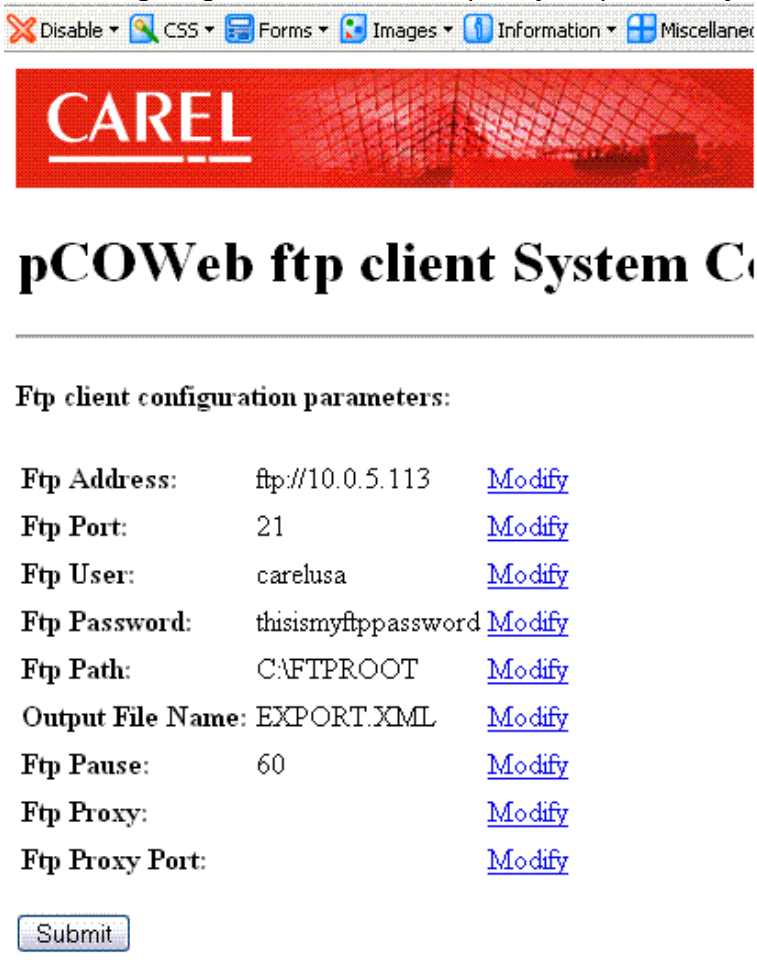
Note: pcoWeb does not ship with FTP Push functionality. To use FTP Push, your pcoWeb must be configured by CAREL, or you will receive instructions to configure it yourself. This is not a standard application on the pcoWeb due to memory limitations on the card.

Direct your browser to the following page:
http://path_to_your_pcoWeb/http/plugins/ftp_client.html

There you will see the following:



After configuring the FTP Push setup for your pcoWeb, your screen will look similar to this:



You must then power down the unit for the FTP routine to work. It will begin automatically when the unit powers up.

SNMP PROTOCOL

SNMP TREE for pCOWeb (from the release A1.0.1 – B1.0.0)

General OID format:

1.3.6.1.4.1.9839(Carel).A.B[.C.D.[E]]

A.B.C.D.E explanation

A=1: pCOWeb properties

1.1= AgentRelease (ReadOnly) = A1.0.1 – B1.0.0 pCOWeb sw release

1.2= AgentCode (ReadOnly) = “pCOWeb”

A=2: pCO properties

2.0.10 = pCOStatus (ReadOnly) = (0=pCO offline, 1=pCO init, 2=pCO online)
2.0.11 = pCOerrors (ReadOnly) = (# of communication retries; 0 when ok or when it reaches 5 retries (->pCO offline))

2.1.C.D = pCO variables (ReadWrite)
C = variable type (1:digital, 2:analog, 3:integer)
D =variable index (1, ...,207)

E = always 0.

Note – SNMP R/W:
The actual ReadWrite property depends only from the particular application software downloaded into the pCO controller; the ReadWrite attached as remark near the previous “pCO variables” row means only that a new value received from the network and intended to replace a old value of a particular pCO variable will be sent to pCO, and a variable reading performed by a remote host via the network will get the value that the pCO passes to pCOWeb, no matter of the actual meaning of this value: it could be a variable used or not by the particular pCO application.

Example:
1.3.6.1.4.1.9839.2.1.2.45.0:
Analog variable with index number 45 (the ending 0 is always needed as every object is treated as a scalar value)

TRAPS

Introduction
The TRAPs represent a way by which pCOWeb can send a warning – a SNMP coded message – to some receivers when a particular status comes active.
The main reason for this mechanism is useful is that it is an “asyncronous”, i.e. a remote host is able to get a status warning from pCOWeb without any active remote polling activity. The only need at the supervisor side is the presence of a running task able to detect the messages that could come from pCOWeb.

pCOWeb is able to send two types of trap: system traps and controller traps.

SYSTEM TRAPS

To set these traps you have to click onto the link “SNMP System defaults configurations” of the Web Administrator page.
Note: all the settings in this page will be active from the next reboot of pCOWeb.

Default traps
Default trap community
The “Community” in SNMP standard is a string useful to filter the access from a host toward pCOWeb: if the community string of the host is not the same of that of pCOWeb, the access will be denied (it is then somewhat similar to a password).
For the TRAPS, the meaning of the “Community” string is similar, except that the roles pCOWeb-host results inverted: when a pCOWeb generates a TRAP, it puts the “Default trap community” string into the TRAP message. This gives the host a chance to filter some unwanted TRAPS: the host that receives the TRAPS can then decide which TRAPS to show and which ones to ignore.

Send authfail trap
This flag lets the user to choose if the pCOWeb have to generate a particular TRAP (authfail trap) when an attempt has been detected to access pCOWeb with a wrong community by a host. The trap will be sent to every specified host (see below).

System traps hosts table
System trap destination host #N
Each row of that section lets you to to set the IP address of a host. It must have the usual format: A.B.C.D (every number in decimal format). Example: 192.168.10.2.
When a host is set, pCOWeb will send a specific trap EVERY START and EVERY STOP of the SNMP application (usually at boot and at reboot). So, when you power-up pCOWeb or when you reboot it, a *coldstart trap* will be sent to every IP address of the hosts. Similarly, when a reboot is requested, a *[enterprises].8072.4.0.2* trap type will be sent. The only way to disable these traps to a specified host is to clear the corresponding IP address.
Note – Trap Host IP:
The four IP addresses for the hosts shouldn’t be confused with those available in the html page that you can see when you click on the link “SNMP pCO variables traps configuration” of the “WEB Administrator page”: in this “SNMP System defaults configurations” page the four IP addresses will be used ONLY in relation to the settings available in this html page.

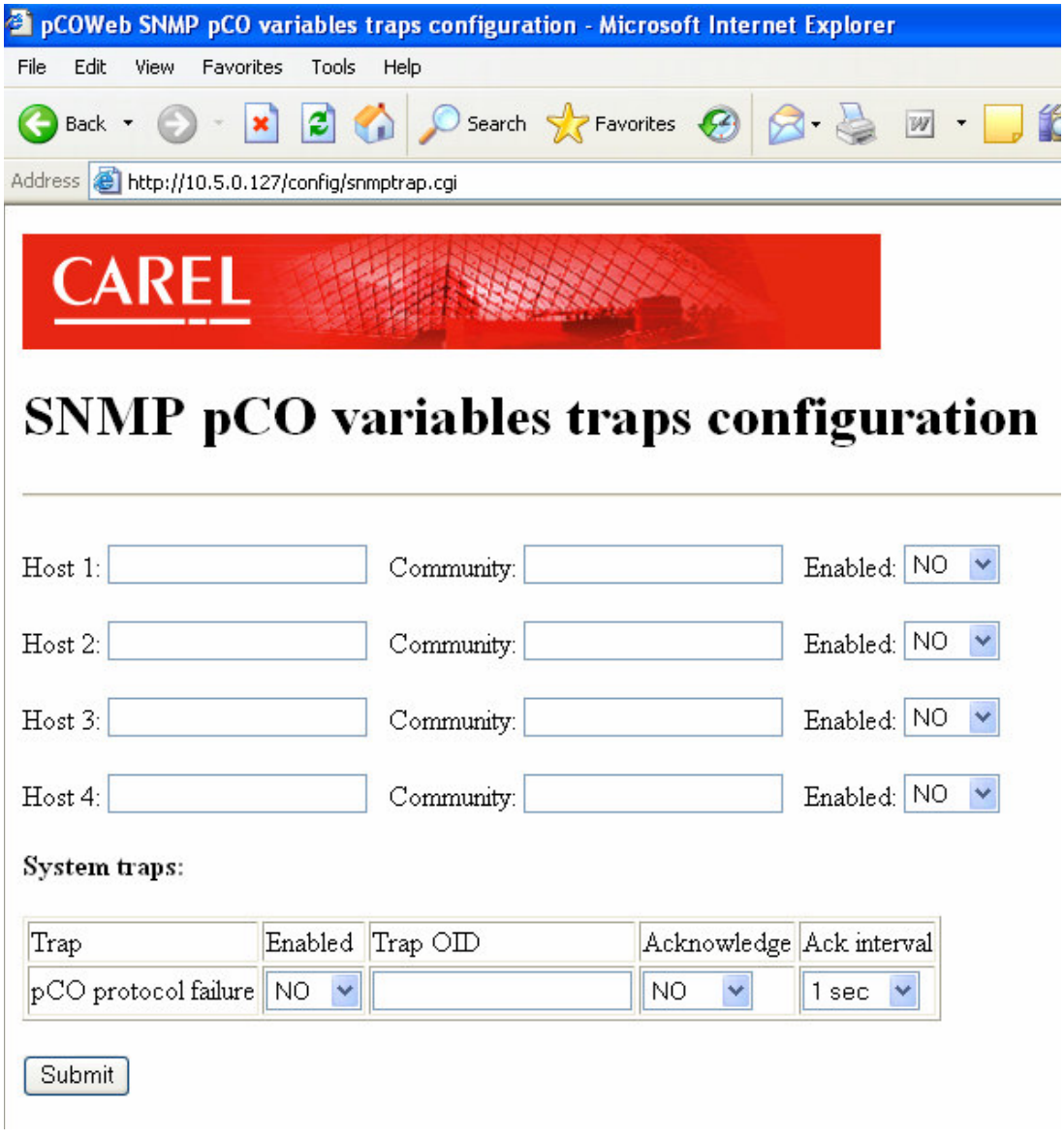
CONTROLLER TRAPS

This type of TRAPS is sent when some selectable status comes active in the pCO controller into which the pCOWeb was plugged in.
To set these traps you have to use the link “SNMP pCO variables traps configuration” of the Web Administrator page.
Note: all the settings in this page will be active once you have got the refresh of the html page after a “Submit” button pressed (it is not necessary to reboot pCOWeb).

Note – Two Submit Buttons: the page is divided into two half pages (top / down). Each of these has its own "Submit" button; please note that a "Submit" button will confirm ONLY the parameters/selections of the related half page, with no effect to those of the other half page.

Let's examine the page.

Top Half of Trap setup Page



The top zone of the top half page is devoted to the settings for up to four destination hosts: IP address, Community, enabling flag.

IP address: must have the usual format: Example: 192.168.10.2.

Community: the “Community” in SNMP standard is a string useful to filter the access from a host toward pCOWeb: if the community string of the host is not the same of that of pCOWeb, the access will be denied (it is then somewhat similar to a password).
Enabling flag: when this flag is set to “No”, it overrides the settings of the rest of the html page: no controller TRAPS will be sent to that host; when this flag is set to “Yes”, the TRAPS sent to that host will depend on the selections of the rest of the html page.

Note – Trap Host IP:
The four IP addresses for the hosts shouldn’t be confused with those available in the html page that you can see when you click on the link “SNMP System defaults configurations” of the Web Administrator page: in this “SNMP pCO* variables traps configuration” page the four IP addresses will be used ONLY in relation to the settings available in this html page.

System traps

This table contains only one system trap: its name is “pCO protocol failure”. When enabled, the TRAP will be sent to all the hosts set in the above section of the html page in occurrence of the continuous communication between pCOWeb and pCO* becomes impossible. Normally this could never happen, but it does, it is possible that there is some hardware or pCO application software problems. You can check this situation in the left LED (“status”) of pCOWeb: it will appear red with dark flashing every time a request generated from pCOWeb is not followed by the right answer from pCO*.

Enabled
Enabling flag (only for “pCO protocol failure” trap).

Trap OID
Trap Object Identifier (a sort of name of the trap). This information will be embedded inside the message when the TRAP SNMP message is formatted before being sent to the hosts.
This information is requested by the syntax of all the TRAP messages, except for the system TRAPS, for which the OID is predefined. If you didn’t specify anything inside this field, the TRAP will not be generated.

The form of the Trap OID is, accordingly to the rules of SNMP, a sequence of numbers separated by periods. This syntax is the basic form by which SNMP describes every object. An example for this field is:

1.3.6.1.4.1.9839.3.1

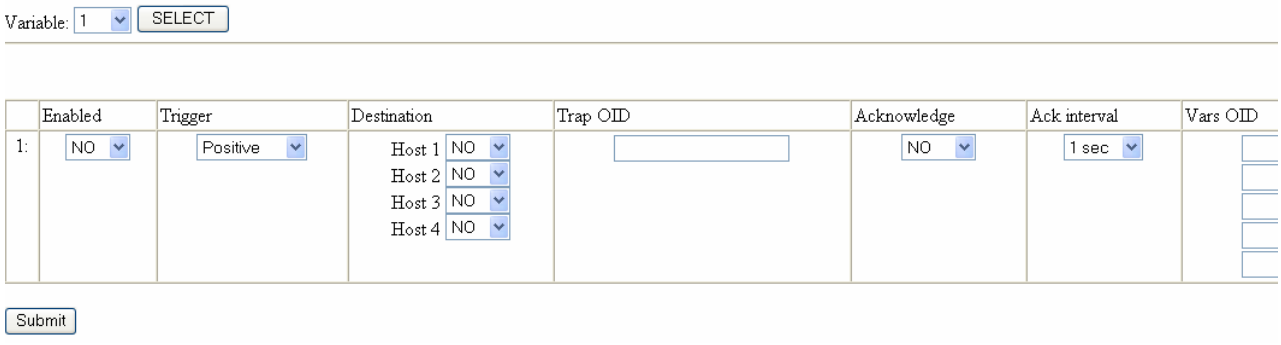
By this information the receiver will be able to distinguish each TRAP from each other. So the Trap OID represents the type of the TRAP. See also the previous section describing the SNMP tree for pCOWeb

Acknowledge and Ack interval
By these selection tools you can decide to activate (or not) the Acknowledge feature for the pCO protocol failure trap. For example:

Suppose that you have enabled this feature to *Acnowledge*=5 (times) with a *Ack interval*=10 seconds. When pCOWeb sends the trap to Host1, it will begin to wait for a special reply from the corresponding Host1 saying “ok, I received this trap”. If pCOWeb is not able to get this Acknowledge within 10 seconds from the first sending, it will send the

same trap one more time, and then it will start waiting for the Acknowledge. This mechanism will stop as soon as pCOWeb will get the Acknowledge, or after a maximum of 5 further traps after the first sending.

Bottom Half of Trap Setup Page



This half page lets you to configure the traps related to state changes of each digital variable of the pCO* controller.

Variable: 1 ↓SELECT
When you enter the page, you will be able to see the settings of the Digital Variable #1, as reported by the selector "Variable:" which is set to 1 by default. The configuration of the traps for this Digital Variable is contained in the table below the Variable selector. By this table you can change the configuration of the traps for the Digital Variable #1. If you want to check/change the Configuration of another digital variable, you first have to select another Variable number, then you have to click on the "SELECT" button and wait for the half page refresh. The configuration table shows in the most left column the number of the Digital variable selected.

Enabled and Destination Columns
If Enabled "NO" is selected, the state changes of the currently selected Digital Variable will never generate traps; if Enabled "YES" is selected, the traps generation and sending will depend on the "Destination" enabling flags. For instance, to let a trap to be sent to the Host1, all three Enabling flags ("Enabled" for the Host1 in the top half page, "Enabled" in the bottom half page, Destination - Host1 in the bottom half page) have to be set to "YES".

Trigger
Let you to select which state transition will generate the trap: Positive: from 0 to 1 ransition Negative: from 1 to 0 transition or Pos & Neg: every transition.

Note – Trap Trigger: at pCOWeb power-up and after every revivel of a previously failed communication state with pCO*, pCOWeb will send a trap for each digital enabled variable accordingly to the Trigger selection and the following rules:

Trigger selection= Positive: a trap will be sent if the value of the variable is found to be 1
Trigger selection= Negative: a trap will be sent if the value of the variable is found to be 0
Trigger selection= Pos & Neg: a trap will be sent regardless the value of the variable.

Note that this selection together with the enabling of several traps & hosts could result in a delay from the power-up to the complete acquisition of all the values of the pCO controller

variables. Until the acquisition you will read “U” (undefined) as the result of a read of a not acquired variable. You can realize a slow refresh by checking the “status” led in pCOWeb: slow or no blinking means slow or no refresh.

Trap OID
Trap Object Identifier (a sort of name of the trap). This information will be embedded inside the message when the TRAP SNMP message is formatted before being sent to the hosts. This information is requested by the syntax of all the TRAP messages, except for the system TRAPS, for which the OID is predefined. If you didn’t specify anything inside this field, the TRAP will not be generated.

The form of the Trap OID is, accordingly to the rules of SNMP, a sequence of numbers separated by periods. This syntax is the basic form by which SNMP describes every object. An example for this field is:

1.3.6.1.4.1.9839.3.1

By this information the receiver will be able to distinguish each TRAP from each other. So the Trap OID represents the type of the TRAP. See also the previous section describing the SNMP tree for pCOWeb

Acknowledge and Ack interval
By these selection tools you can decide to activate (or not) the Acknowledge feature for the pCO protocol failure trap. For example:

Suppose that you have enabled this feature to *Acnowledge*=5 (times) with a *Ack interval*=10 seconds. When pCOWeb sends the trap to Host1, it will begin to wait for a special reply from the corresponding Host1 saying “ok, I received this trap”. If pCOWeb is not able to get this Acknowledge within 10 seconds from the first sending, it will send the same trap one more time, and then it will start waiting for the Acknowledge. This mechanism will stop as soon as pCOWeb will get the Acknowledge, or after a maximum of 5 further traps after the first sending.

Vars OID (optional)
A trap message can carry some additional informations: pCOWeb can fill the trap message with the values of up to five variables of the pCO controller read at the time the trap was generated. By the fields *Vars OID* you can specify the OIDs for these variables.

At the moment the trap will be generated, pCOWeb will get the values of the variables whose (valid) OIDs are specified within the fields and will fill the trap message with them. By this, the host that receives the trap can also report these informations, normally useful to better understand the reasons the trap was generated by the application in pCO*.

BACNET PROTOCOL

The BACnet protocol is the ASHRAE building automation and control networking protocol that provides a mechanism by which computerized equipment can exchange information regardless of the particular building function they perform. The pcoweb provides Carel pCO* controllers with a gateway to the BACnet protocol. The pCOWeb appears to be *native BACnet device* as it reflects the status of various real-time measurements and commanded setpoints in a single Carel pCO* controller. The protocols supported are BACnet IP, BACnet Ethernet and BACnet MSTP.

Due to the small form factor, there are two different models of the pcoweb. The first (PCO1000WB0) has a connection for an Ethernet networks and has Ethernet 8802-3 and BACnet/IP capabilities. The second (??????????) has a connection for an EIA-485 network and provides the BACnet MS/TP protocol.

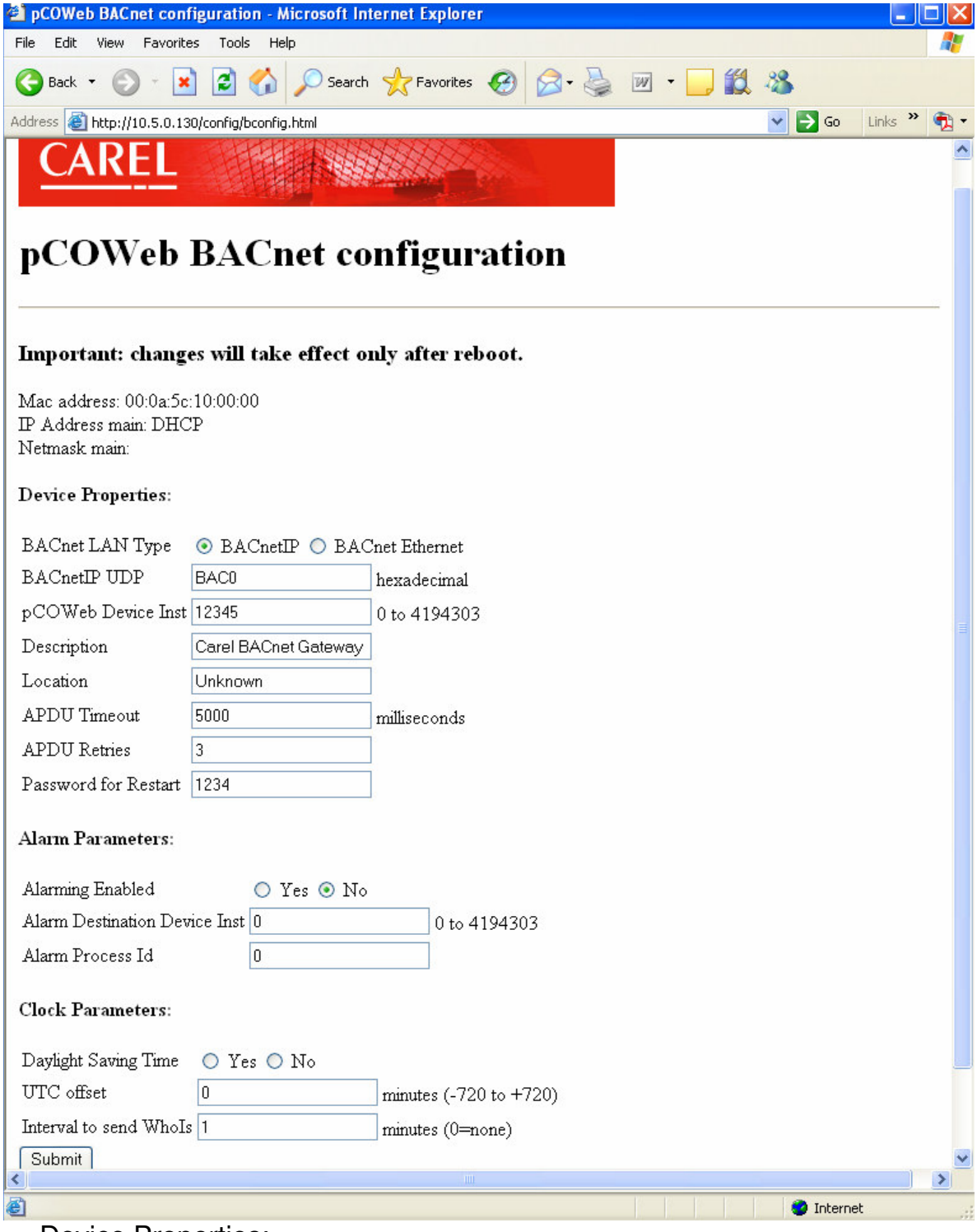
Part Number	BACnet Protocol
PCO1000WB0	**BACnet/IP and BACnet Ethernet**
PCO1485WB0	**BACnet MSTP**

BACnet/IP and BACnet Ethernet configuration

To configure the pCOWeb BACnet you must connect the pCOWeb to your Ethernet and use a browser to goto http://your-pcoweb-ip. Select *web administrator reserved area*. The password for this area defaults to:

username: admin
password: fadmin

Select *BACnet configuration.* This will display the following page:



Device Properties:

1. BACnet LAN Type: select the protocol to use BACnet/IP or BACnet Ethernet.
2. BACnetIP UDP: if BACnetIP is used then a UDP port must be assigned, the default port for BACnet is “BAC0” hexadecimal.
3. pCOWeb Device Instance: This is the device instance of the pCOWeb on the network. It defaults to 77000 but can be set from 0 to 4194303. Every device on the network must have a unique instance number.

4. Description: A 64 character field that can be used to describe the device.
5. Location: A 64 character field that can be used to describe the physical location.
6. APDU Timeout: Indicates the amount of time in milliseconds between retransmission of an APDU requiring acknowledgement for which no acknowledgement has been received. The default setting of this property is 5000milliseconds. This value must be non-zero if the APDU Retries is non-zero. In order to achieve reliable communication, it is recommended that the value of the APDU Timeout be the same in all intercommunicating devices.
7. APDU Retries: Indicates the maximum number of times that an APDU shall be retransmitted. The default value is 3. If the value of this property is non-zero, then the APDU Timeout field must be non-zero.
8. Password for Restart: This is the reinitialize password. The default is 1234. The value can up to 9 characters long. It is required from the responding BACnet-user prior to executing the ReinitializeDevice service.

Alarm Parameters:

1. Alarming Enabled: Set to yes to enable the intrinsic alarming features of AV and BV objects.
2. Alarm Destination Device Inst: Insert the device instance of the destination for the alarms. Allowable range is 0-4194303.
3. Alarm Process Id: Insert the process Id or handle which is meaningfull to the alarm destination device. This handle will be used to identify the process within the destination device that should receive the alarm notification.
4. Interval to send WhoIs: The pCOWeb will send a WhoIs command to the alarm destination device. WhoIs will continuely be sent at the interval provided until a valid response is received. Setting this value to 0 will disable this command.

Clock Parameters:

1. Daylight Saving Time: Check yes if it is required to automatically track daylight savings time.
2. UTC offset: Universal Time Coordinated offset is used to enter an offset in minute resolution. The UTC time zone is GMT (or Zulu time in military terms). For example you would enter -300 to get Eastern Standard Time (EST).

Submit Button:

Pressing the “submit” button will save your changes. Then you must reboot the pCOWeb by pressing the reset button on the card or by resetting power to the pCO* controller.

pCO* to BACnet Database

The pCO* communicated variables are maped to analog_value and binary_value BACnet objects. A table-based list of BACnet-visible values is maintained by the pCOWeb. Each interface table value is accessible by BACnet clients as a Present_Value property of a standard object. The Carel variable for each interface table slot defines the object type implicitly.

Analog variables are mapped to BACnet Analog Value object type with an instance number equal to their variable number, e.g. analog variable 17 would appear to be BACnet Analog Value 17. Because analog variables are implicitly scaled "times 10" their conversion to REAL in BACnet is scaled accordingly. So the value "123" which is taken to mean "12.3" is converted to a floating point "12.3" as the value for the AV object Present_Value property.

Integer variables are mapped to BACnet Analog Value object type with an instance number equal to their variable number plus 1000, e.g. integer variable 17 would appear to be BACnet Analog Value 1017. The integer value shall be converted to REAL in BACnet. So the value "123" should be converted to a floating point "123.0" as the value for the AV object Present_Value property.

Digital variables are mapped to BACnet Binary Value object type with an instance number equal to their variable number, e.g. digital variable 17 would appear to be BACnet Binary Value 17.

The following table summarizes the relationship.

Carel variables	BACnet variables
Digital variable 1	Binary value instance 1
Digital variable 199	Binary value instance 199
Analog variable 1	Analog value instance 1
Analog variable 127	Analog value instance 127
Integer variable 1	Analog value instance 1001
Integer variable 127	Analog value instance 1127

Supported BACnet Objects

The pCOWeb supports analog_value, binary_value and device object types.

The following properties of those objects are supported in addition to the writable Description property. In general standard object property values other than Present_Value are derived as follows:

- Object_Identifier generated automatically from request
- Object_Name generated by taking a single letter A, I or D concatenated with the variable number as 3 decimal digits, e.g. "A025"
- Object_Type generated automatically from request
- Polarity always NORMAL
- Description writable
- Status_Flags always {FALSE,FALSE,FALSE,FALSE}
- Event_State always NORMAL
- Reliability generated automaticaly once pCOWeb has received a valid reading from pCO* controller.
- Out_Of_Service always FALSE
- Units writable
- Inactive_Text writable, default to "0"

- Active_Text writable, default to "1"
- Time_Delay writable
- Alarm_Value writable (BV only)
- High_Limit writable (AV only)
- Low_Limit writable (AV only)
- Deadband writable (AV only)
- Limit_Enable writable (AV only)
- Event_Enable writable
- Acked_Transitions generated by alarm logic
- Notify_Type always ALARM
- Event_Time_Stamps generated by alarm logic

From a BACnet perspective, the pCOWeb appears to be a single BACnet device representing the pCO* controller. Consequently it provides a Device object. The Device object is homed on the BACnet LAN. The Device instance and properties are programmed through either the WEB interface or by the Carel BACnet Gateway Configuration software.

The Device Object supports the following properties:

- Object_Identifier
- Object_Name (configurable)
- Object_Type
- System_Status
- Vendor_Name
- Vendor_Identifier
- Model_Name
- Firmware_Revision
- Application_Software_Version
- Location (writable)
- Description (writable)
- Protocol_Version
- Protocol_Revision
- Protocol_Services_Supported
- Protocol_Object_Types_Supported
- Object_List
- Max_APDU_Length_Supported
- Segmentation_Supported (not supported)
- APDU_Timeout
- Number_Of_APDU_Retries
- Max_Master (only present on MS/TP version, writable)
- Max_Info_Frames (only present on MS/TP version, writable)
- Device_Address_Binding
- Database_Revision

BIBB Support

The pCOWeb supports the following specific BIBBs per their relevant definitions in Addenda D/Annex K to BACnet:

DS-RP-B, DS-RPM-B, DS-WP-B, DS-WPM-B
i.e. act as a server that can execute Read/Write single or multiple properties

DM-DDB-B, DM-DOB-B
i.e. dynamic device and object binding. Support initiating I-Am in response to Who-Is, and at configurable interval (Who-Is only, Who-Is plus startup, Who-Is plus every N minutes). This option is configurable through software.

DM-DCC-B, DM-RD-B
i.e. DeviceCommunicationControl and ReinitializeDevice. These features shall support a Reinit Password configurable through software.

AE-N-I-B, AE-ACK-B, AE-INFO-B
i.e. generate EventNotifications when programmed alarm conditions occur; accept acknowledgements for those notifications marked as requiring acknowledgement, and support GetEventInformation requests.

Object Support of Intrinsic Alarming

In lieu of full support for Notification Class objects, the intrinsic alarming features of AV and BV objects determine the destination for event notifications using added properties in the Device object, and individual AV and BV objects:

The Device object includes these standard properties used in a non-standard manner

- Recipient — writable, representing the device instance to send event notifications to.
- Process_Identifier — writable, representing the process identifier of the recipient process in the device to which notifications are sent.

The AV and BV objects include these standard properties used in a non-standard manner

- Issue_Confirmed_Notifications — writable, whether alarms from this object should be sent as confirmed or unconfirmed.
- Ack_Required — writable, whether alarms from this object require individual human acknowledgement.
- Priority — writable, what the alarm priority for this object is in event notifications (note that all transitions use the same priority)

The AV and BV objects include alarm detection and reporting logic to implement high/low limit alarming for AV and Change-Of-State alarming for BV. The detection algorithms behave as described in 12.3.16, 12.3.17 and 12.7.23 of 135-2001 for standard intrinsic alarming. The resulting event notifications are reported to a single destination specified device configuration. The Description property is used as the alarm message description in event notifications.