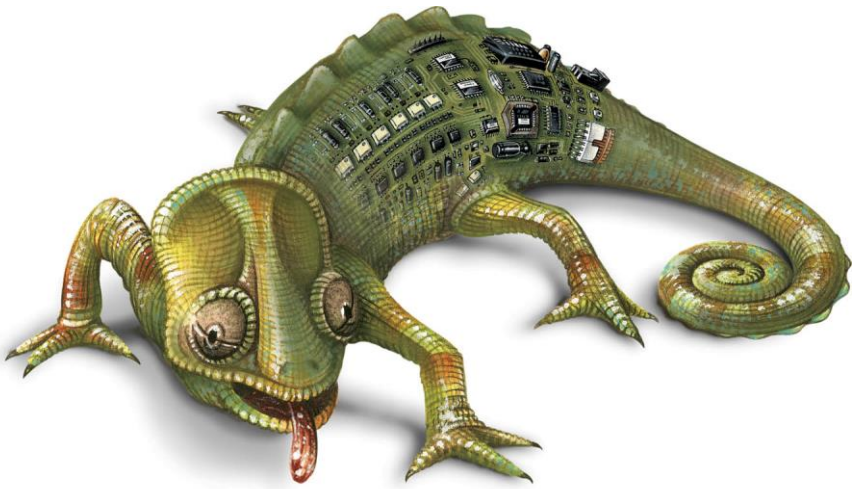


K6 SERIES

kameleon™

ADAPTABILITY IN LIGHTING CONTROL

BACnet™

BACNET KC600

LIGHTING CONTROLLER

OPERATING MANUAL

121-39633C

gentec ENERGY MANAGEMENT

2625, Dalton Avenue, Quebec, (Quebec), G1P 3S9, Tel.: (418) 651-8000

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

Used as a stand-alone unit, the *BACnet* lighting control board *"KC600"* allows for:

- reading, switching ON or OFF 16 individual relays to manage lights;
- reading the status of 2 wall switches.

Used along with *KC621* (16 inputs) and *KC631* cards (16 outputs), the *BACnet* lighting control board *"KC600"* allows for:

- reading, switching ON or OFF up to 64 individual relays to manage lights;
- reading the status of up to 66 wall switches.

Used alone, with or without *KC6XX* complementary cards or integrated with other *KC600* cards in a *BACnet* network, the *KC600* allows the creation of:

- **groups** composed of individual outputs located on one or many *KC600*;
- internal **links** between an input and one or many groups on the same *KC600;*
- elaborated **schedules** for automated control, applied to one or many groups defined on the same *KC600*.

Though a network composed only with *KC600* cards can manage by itself the lighting of one or many buildings, this control may be shared, even be mastered by a *BACnet* monitoring/control console, the *KC600* being easily integrated with other devices communicating through *BACnet* protocol, no matter the manufacturer.

1.1 Acronyms and Abbreviations

APDU — ***A****pplication layer* ***P****rotocol* ***D****ata* ***U****nit*. The *APDU* of *BACnet* is a heading (fixed format) showing the service request nature, the quantity of bytes identifying the application layer operation and information about the message possible segmentation. The *APDU* of *BACnet* also comprises a variable section, depending on the data accompanying the service request.

ASHRAE — ***A****merican* ***S****ociety of* ***H****eating,* ***R****efrigerating and* ***A****ir-Conditioning* ***E****ngineers*: A committee of this society, along with representatives from all sectors of the industry, is responsible of the updating, support and maintenance of the *BACnet* standard.

BACnet — ***B****uilding* ***A****utomation &* ***C****ontrol* ***Net****works*: Communication protocol developed as a standard by *ASHRAE* for the complete automation and control of buildings.

BBMD — ***BACnet****/IP* ***B****roadcast* ***M****anagement* ***D****evice:* Broadcast management is accomplished by defining the capabilities of a new device called a *BBMD*. It receives ***"broadcast"*** messages on one subnet and forwards them to another subnet or individually to the *"Foreign Devices"* (*FD* / *BACnet* devices on another subnet), which addresses are registered in the *"Foreign Devices Table" (FDT)* of the *BBMD*.

DST — ***D****aylight* ***S****aving* ***T****ime:* practice of advancing clocks during the lighter months so that evenings have more daylight and mornings have less. Typically clocks are adjusted forward one hour near the start of spring and are adjusted backward in the autumn. The exact moment, when to apply or not this time shift, is defines with the time zone currently set in the *KC600*.

FD — ***F**oreign **D**evice: BACnet* device that has an IP subnet address different from those comprising the *BACnet/IP* network which the device seeks to join. The foreign device may be a full-time node on the foreign subnet or may be a part-time participant. If you want a *BACnet* device on a subnet to receive *BACnet* ***"broadcast"*** messages from another subnet, you may configure your device so it will send requests, at fixed interval, to a *BBMD* for registration in the *"Foreign Devices Table" (FDT)* of the *BBMD*.

GMT — ***G**reenwich **M**ean **T**ime* (also known as *"Western European Time"* and *"Zulu time"*): originally referred to the mean solar time at the Royal Observatory in Greenwich, London, it was adopted as a global time standard for most part of the XXth century. It was replaced in 1972 by the *"Coordinated Universal Time" (UTC)*, similar to the GMT reference.

Kameleon — Lighting control system designed by *Gentec Inc.*

KC600 — Lighting control board, controllable via *BACnet IP* service requests, allowing for connections of 2 inputs and 16 outputs, when used as a stand-alone unit. Via a *Modbus* port on the board, the *KC600* can communicate with *KC621* and *KC631* boards to increase up to 66 the amount of available inputs, and up to 64 the amount of outputs.

KC621 — "*Slave*" lighting control board, accessible via *Modbus* communication, it allows for controlling 16 inputs that can be connected to:

- *Douglas WR-8501* type switches or equivalent,
- dry contacts (switches, movement or light detectors, etc.)

KC631 — *"Slave"* lighting control board, accessible via *Modbus* communication, it allows the control of 16 outputs to connect 2-wire relays, *"locking circuit"* type *WR6161* (polarized 24 volts).

KCF600 — Configuration software used to set all parameters for a *KC600* and to collect its complete configuration (in *XML* format). It also permits the transmission of a new software version for managing the board operation.

LED — ***L**ight-**E**mitting **D**iode:* An electronic semiconductor device that emits light when an electric current passes through it. LEDs are used as indicator lamps in many devices and are increasingly used for other lighting.

Local time — It is the time indicated by your watch, by the clocks in your area. You get the *"local time"* from a *"UTC time"* by (1) subtracting the *"UTC offset"* associated with the local time zone and (2) you add (usually) one hour if the time zone definition indicates that *"Daylight Saving Time" (DST)* must be applied at the current date.

Modbus — Communication protocol used for automatons networks, for programmable logic controllers, under level 7 (application) of the OSI Model. It operates in master/slave mode: only a master (*KC600* in this case) can initiate a service request, the slaves (*KC621, KC631*) are limited to responding to service requests from the master.

PC — **P**ersonal **C**omputer.

OFF Only — *Option* attached to the groups; refer to the *"List of the 4 available "options"*, in section *2.5.2.5 KC600 group "options.*

ON Only — *Option* attached to the groups; refer to the *"List of the 4 available "options",* in section *2.5.2.5 KC600 group "options.*

Output — Corresponds to a relay accessible via *BACnet* service requests towards *KC600*. The relay may be directly connected to one of the 16 outputs on the *KC600* or to a *Modbus* board (*KC631*) controlled by the *KC600*.

Time zone A *"time zone"* is a region that has a uniform standard time for legal, commercial, and social purposes. In the definition of a "time zone", the main parameters are (1) the *"UTC offset"* for the time shift and (2) usually when to apply the *"Daylight Saving Time" (DST)*, though some "time zones" do not use any DST.

TOE *"Time On Extension": Option* attached to the groups; refer to the *"List of the 4 available "options",* in section *2.5.2.5 KC600 group "options.*

UTC *"Coordinated Universal Time"* is the primary time standard by which the world regulates clocks and time. It relates to the earth rotation, thus varying slowly. It is one of several closely related successors to Greenwich Mean Time (GMT). For most purposes, UTC is synonymous with GMT.

Warning *Option* attached to the groups; refer to the *"List of the 4 available "options",* in section *2.5.2.5 KC600 group "options*

1.2 References

***"Gentec"* Documents:**

Gentec #	**Document Description**
121-38950	Set of specifications for *BACnet KC600* lighting board.

Other Documents:

- **ANSI/ASHRAE Standard 135-2010** manual: *"BACnet, A Data Communication Protocol for Building Automation and Control Networks"*

2. Configuration software (*KCF600*)

2.1 Console start up

2.1.1 Login window

Before being able to visualise the *KC600* in your network, the *KCF600* console starts up by displaying this window:

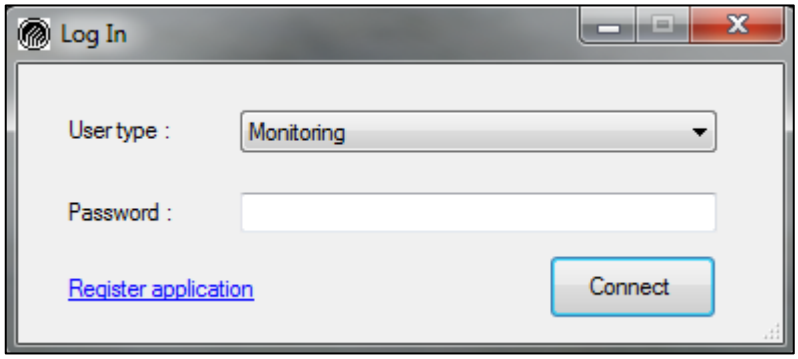


Figure 1: Login window of the KCF600 console.

- **User type:** you must first choose the *"User type"* for which you were assigned a password. There are 4 user types listed in order of permissiveness, based on their access rights to the different console functionalities:

    - **Monitoring:** You will be limited to visualise only the different configurations contained in the various "*installations*" (refer to section: *2.2 Choosing an "Installation" / "Installation" tab*), already saved on disk, without being able to:
        - edit any parameter of the configuration,
        - save the configuration of a *KC600* on disk,
        - communicate with the *KC600* on your network (neither for writing, nor for reading to get an update of the displayed configurations).

    - **Configuration:** this user type allows you to:
        - create, edit and rename *"installations"* (refer to section: *2.2 Choosing an "Installation" / "Installation" tab*),
        - add or delete *KC600* in the **current** *installation* (refer to section: *"2.3 Management of all KC600 part of an "Installation" / "KC600" tab"*),
        - edit or exchange *KC600* configuration files, between the *KCF600* console and a *KC600* controller or between *installations*,
        - manage groups composed of outputs on one or many *KC600*,
        - manage schedules applied to one or many groups.

    - **Installation:** this profile has the same rights as the "*Configuration*" profile, except it can also perform software updates on the *KC600*.

- **Administration:** this profile has the same rights as the *Installation* profile, plus it can:
  - modify the users passwords,
  - change the console display language.

- **Password:** this is where you type the password associated to the user type, selected in the login window, to be able to access the main configuration screen.

Register application

This *"button"* only appears if the console has not been registered with a valid license key allowing you to access all the functionalities of the application.

- Click on this link to open a window titled: "*Application registration*" where you will be able to enter the license key. You must contact *Gentec* to get this key. This will allow you to unlock every features of the *KCF600* software.
- Having the full version of the *KCF600* software provides you the possibility to view, create or configure groups of outputs for your *KC600* cards.
- After entering a valid key, the "*Register application*" button will disappear from your login screen.

2.1.2 Login error messages

- ***Connection error*:** if you entered the wrong password for the selected user type, a "*Connection error*" screen will be displayed. Note that if you enter a bad password 5 times in a row, the application will exit.
- ***File error*:** after validating your password, the configuration console reads all *XML* file associated with the current *"installation"* (defined in next chapter):
  - Each *XML* file defines the parameters of a *KC600* part of the *"installation"*.
  - If a flaw is detected for at least one of these files, a window will appear, listing all files with a problem.
  - In this error window, a button allows you to reject all defective files detected. If you keep them, they will be detected again as incompatible the next time someone will log in the *KCF600*.
- ***Installation folder not found*:** refer to *Figure 2: Message telling you the default installation folder does not* ***exist.***

2.2 Choosing an *"Installation" / "Installation" tab*

An "*Installation*" is a group of *KC600* controllers, in one or more buildings, that can be monitored/controlled by the same *BACnet* network management console.

- The first time the console is loaded, the application will look for the "*Default*" *installation* folder. If this folder does not exist, a message will be displayed and the folder will automatically be created:

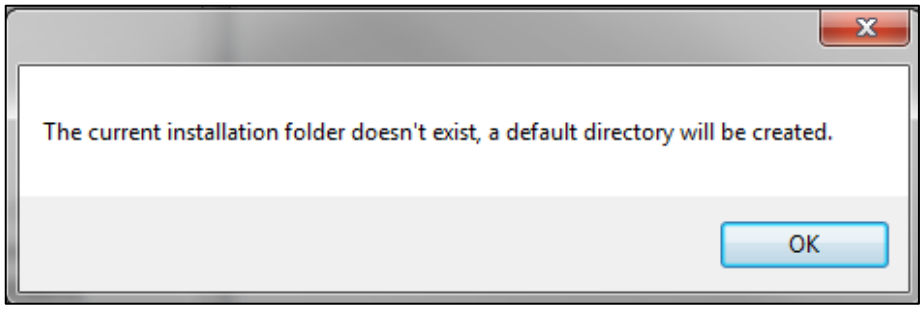


Figure 2: Message telling you the default installation folder does not exist.

- For the following console start-ups, the *KCF600* will just get the *installation* folder that was opened the last time the console exited.

For a new *installation*, an empty window (without any *KC600*) will be displayed:

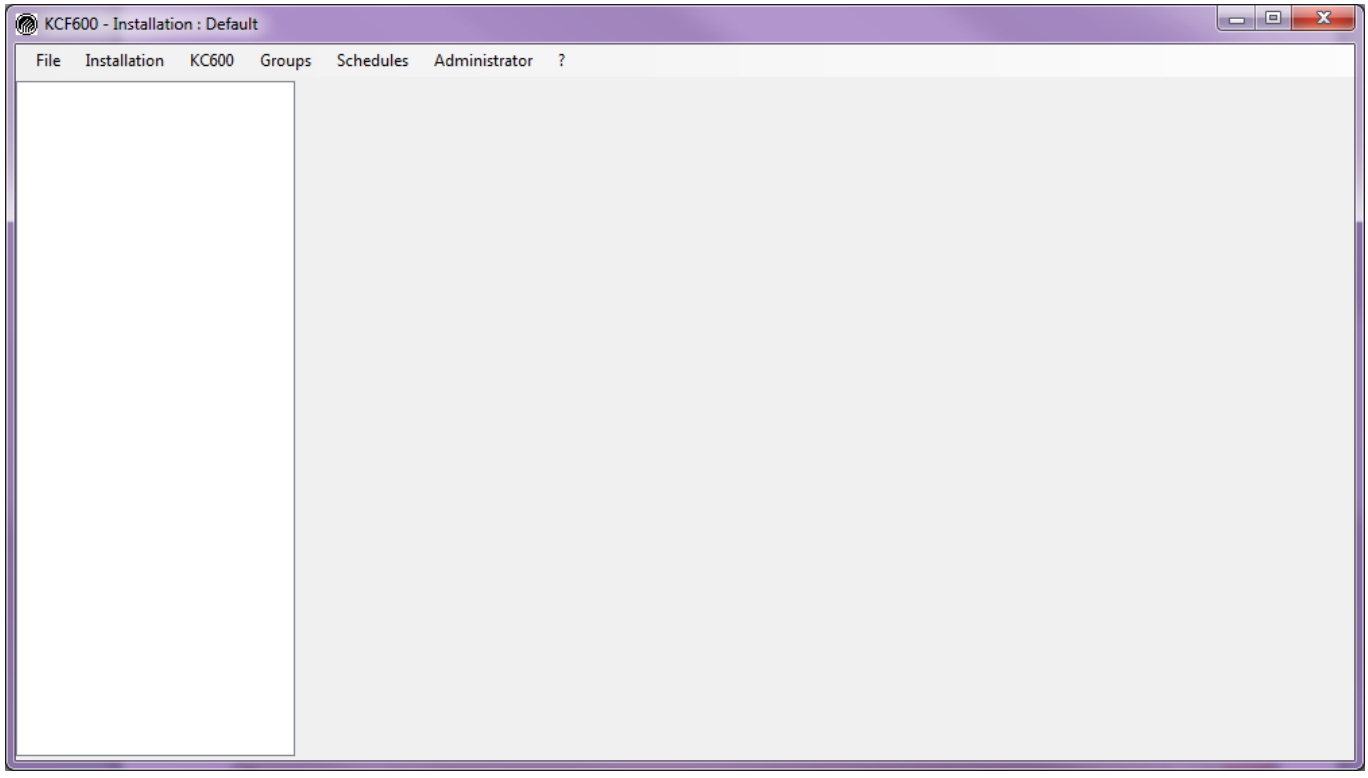


Figure 3: Software main window displaying an empty installation folder

- The name of the currently displayed *installation* can be found in the window title.
- The list (empty for a new *installation*) of all *KC600*, part of the current *installation*, is displayed in the white square on the left section of the console main window; refer to section *2.5.1 Left part of the main window: list of KC600 and their components.*

- The *"installations"* management is made from the "*Installation*" tab. The available menu choices are:

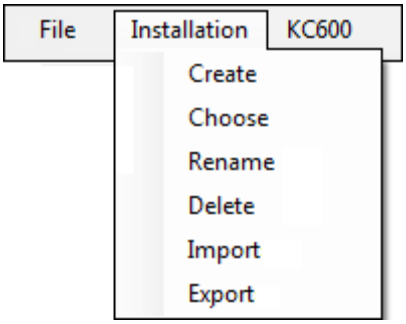


Figure 4: Installation menu items.

2.2.1 Create

This option is for creating a new *installation*. A window appears allowing you to choose the name you wish to give to the new *installation*:

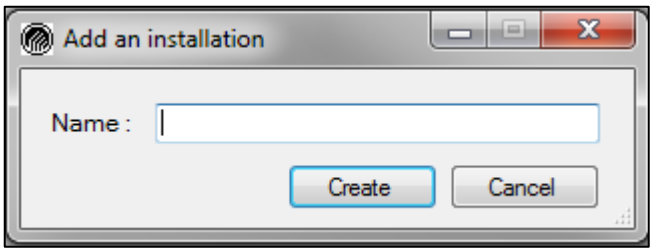


Figure 5: Installation creation window: entering the name of a new installation

- *"**Create**"* button: after entering a name (not already used by another *installation*), press this button to generate a new empty *installation* (refer to *Figure 3*), without any *KC600*.
- *"**Cancel**"* button: click on this option to ignore the name entered in the box (if there is one) and return to the *installation* that was already loaded on screen.

2.2.2 Choose

Select this option to switch between existing *installations*. A window will appear displaying the list of the *installations* already created:

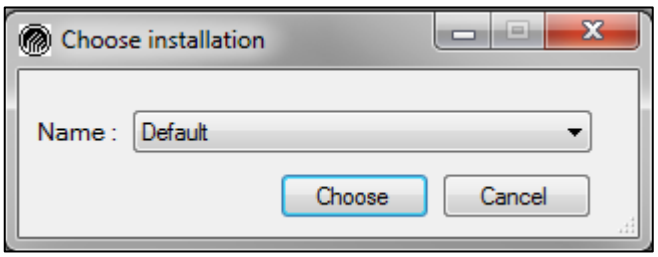


Figure 6: Window allowing the selection of an installation from the list of the created ones.

After choosing an *installation* from the list, you can select:

- ***"Choose"*** button: to have the selected *installation* displayed in the main window.
- ***"Cancel"*** button: to return to the *installation* that was already loaded on screen, without considering the *"Name"* that was just selected in the window on *Figure 6*.

2.2.3 Rename

Select this option to change the name of the *installation* currently displayed. A window will appear to allow you to enter a new name:

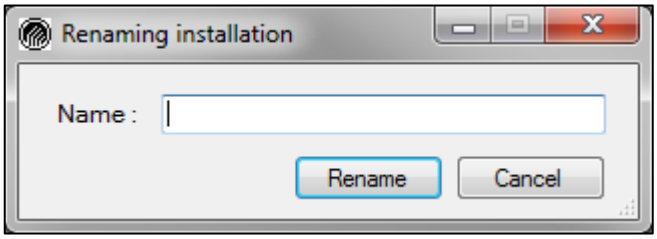


Figure 7: Window allowing you to change the name of the current installation.

- ***"Rename"*** button: to apply to the string entered in the textbox to the current *installation* displayed. If this name is already used by another *installation*, a warning message will notify you about the problem before rejecting it.
- ***"Cancel"*** button: to return to the main window without applying the name entered in the textbox.

2.2.4 Delete

Select this option to eliminate an *installation*, which includes the deletion of all *KC600* definitions added to this installation. After selecting the *"Delete"* option in the menu, you will get the following window to allow the selection of which installation to remove from all the ones already created:

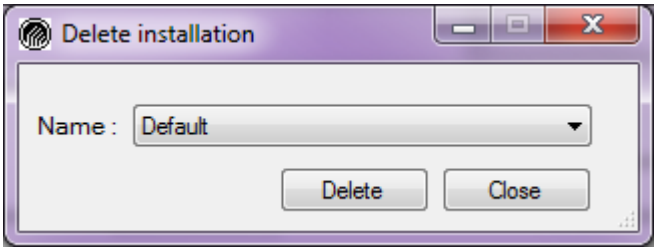


Figure 8: Window allowing you to change the name of the current installation.

- ***"Delete"*** button: to apply the deletion, though another window will open to ask you to confirm (or not) the elimination of selected installation.
- ***"Close"*** button: you may stay in this window to eliminate one or many installations. Once you have completed your task(s) (or if you changed your mind), you click the *"Close"* button to return to the main window.

2.2.5 Import/Export

These choices allow the transfer on an *"installation"* of *KC600s* toward another configuration console, on another computer, through a USB key, a file hosting service (like *"Dropbox"*), an email, etc.

- ***"Export":*** this option allows you to save all the files part of an *"installation"* in a single compressed file (with extension: *"*.gz"*). The selection of this option is followed by the opening of a navigation window to determine where to save the compressed file, even to give it another name than the one proposed.
- ***"Import":*** this option allows you to retrieve the compressed file of an *"installation"* (with extension: *"*.gz"*), file previously *"exported"* from another configuration console *KCF600*. The selection of this option is followed by the opening of a navigation window to choose which file to retrieve from which directory.

2.3 Management of all *KC600* part of an *"Installation" / "KC600" tab*

- The *"KC600"* tab does not appear in the menu if you logged yourself with the *"Monitoring"* profile; refer to section: *"2.1.1 Login window"*.
- The composition of an *"installation"* is realized by adding *KC600* one by one, from the *KCF600* console.
- There are 2 ways for adding *KC600* to an *installation*:
    - From the *"KC600"* menu, by selecting the item *"Add"*. This process requires that the configuration console can communicate with the *KC600* through an *Ethernet* connection.
    - From the *"File"* menu, by selecting the *"Import"* option to transfer a valid *XML* configuration file (for a *KC600*) from another *installation* directory; refer to section: *"2.4 Adding a KC600 configuration from another "Installation" / "File" tab"*.
- The *KC600* management (excluding the "*settings*") is done from the "*KC600*" menu tab. The available choices are:

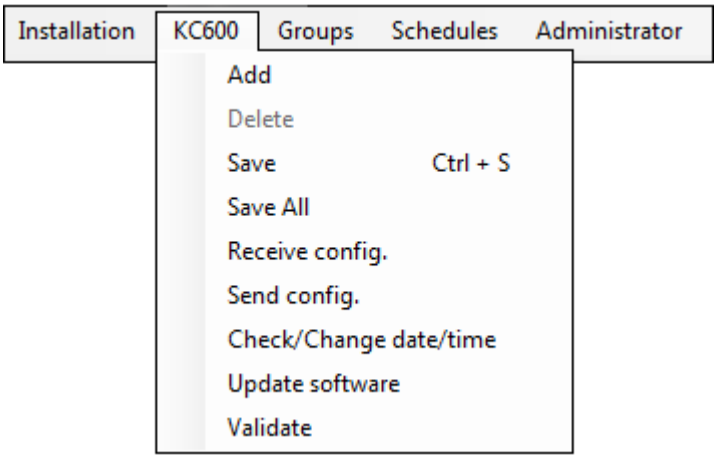


Figure 9: KC600 menu items.

2.3.1 Add

There are **prerequisites** to be able to add a *KC600* via the "*Add*" menu item:

- The *KC600* must be linked to the same *Ethernet* network as the console computer or be directly connected to this computer; refer to: *"Annex 1. First Time Attribution of Network Coordinates to a KC600 Card"*.
- The computer network card must be configured to share the same *Ethernet* network as the *KC600* controller (refer to the same *annex*) or a router must allow the communication with a *KC600* on another network than the one currently used by the computer.
- You must know the IP address of each *KC600* you wish to add to your *installation.*
    - **For an unconfigured *KC600*:** for a new *KC600* that was never integrated in an IP network, it is recommended to connect the card directly to the computer, except if it is the only unconfigured card in your network. Refer to: *"Annex 1. First Time Attribution of Network Coordinates to a KC600 Card"*.

To integrate a card in a *BACnet* network, you need to enter certain basic information concerning the *KC600*:

- an ***IP address***;
- a ***device instance number*** (*"BACnet Object Identifier"*) that must be unique on your *BACnet* network, to identify the *KC600* as a *BACnet* object (of type: "*device*").

- **For an already configured *KC600*:**
  - You need to know the IP address that was previously attributed to the card you want to integrate to your *installation*.

After selecting "*Add*" in the "*KC600*" menu, you will see this window being displayed:

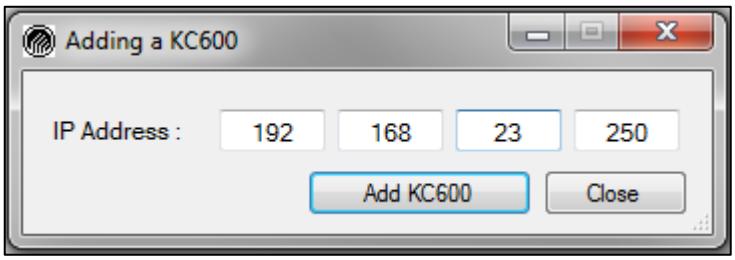


Figure 10: Window used to add a KC600 to the current installation.

- The IP address, initially displayed when the window appears, is the default one attributed to a new *KC600* card that was never configured (or integrated in an IP network):
  - **For an unconfigured *KC600*:** you must keep the default address in the window to communicate with the new *KC600* card.
  - **For an already configured *KC600*:** you must enter the IP address of the *KC600* you want to reach.

- Then click on the "*Add KC600*" button in this window; regardless of the response from the *KC600*, this window will remain displayed until you press the *"Close"* button or the button: (which allows you to add more than one *KC600* at a time, each one having its own IP address):
  - If the *KC600* responds, it then provides its configuration to the console.
    - If this *KC600* was not already part of the current *installation*, it will be added to the list displayed in the main window; see *Figure 17: Main window: list of KC600 included in the current installation.*
    - If the *KC600* responds but provides an *instance number* already used by a *KC600* in your current *installation* list on screen, a message will ask you if you want to replace the configuration stored (with your console) by the one just received.
    - The same will happen if you entered an IP address already used by one of the *KC600s* part of the current *"installation"*.
  - If the *KC600* does not answer, a message window will notify you that the "*TCP connection failed*". To determine why the connection failed:
    - Check if the IP address you entered, in the window of *Figure 10*, is the right one.
    - If there is no doubt that you entered the right IP address, check if you can reach that address by transmitting it a *"ping"* command, according to the *"procedure"* of *Annex 1. First Time Attribution of Network Coordinates to a KC600 Card*.

- If the card still does not answer, check that it is powered on and properly plugged into your *Ethernet* network. To validate that the card is functioning properly, refer to section *3.2 Light Emitting Diodes (LEDs) on KC600 Board*.
- If the previous step does not fix your problem, check the address that was attributed to the *KC600* card by following the directions from section *3.1.3 QNX Console Port (RS232)*.
- Final solution: replace the card by another *KC600* controller!

- After closing the *"Add KC600"* window, you may set or change the configuration of the *KC600* from the main window; refer to *2.5.2.1 KC600 card configuration parameters*.

2.3.2 Delete

To remove a *KC600:*

- You must select one from the list of *KC600* for the current *"installation"* on screen (refer to *Figure 17*).

  NOTE: You must have selected the name of a *KC600* from this list for the "*Delete*" option to be available in the "*KC600*" menu.
- After selecting the *"Delete"* option or by pressing the *"Delete"* key from your computer keyboard, a window will appear to ask you to confirm or abandon the deletion process.
- You can only delete one *KC600* at a time.

2.3.3 Save

By selecting the "*Save*" option in the "*KC600*" menu, you save every parameter of the currently selected *KC600* in a directory named after the *"installation"* on screen, in the *"WORK"* directory on the configuration console. You may achieve the same action through the shortcut key combination: "*Ctrl + S*".

2.3.4 Save All

By selecting the "*Save All*" option in the "*KC600*" menu, you will update the storage for the configuration of every *KC600* appearing in the current *"installation"* list.

2.3.5 Receive config.

By selecting the "*Receive*" option in the "*KC600*" menu, a window appears with the list of every *KC600* already integrated in the current *installation*:

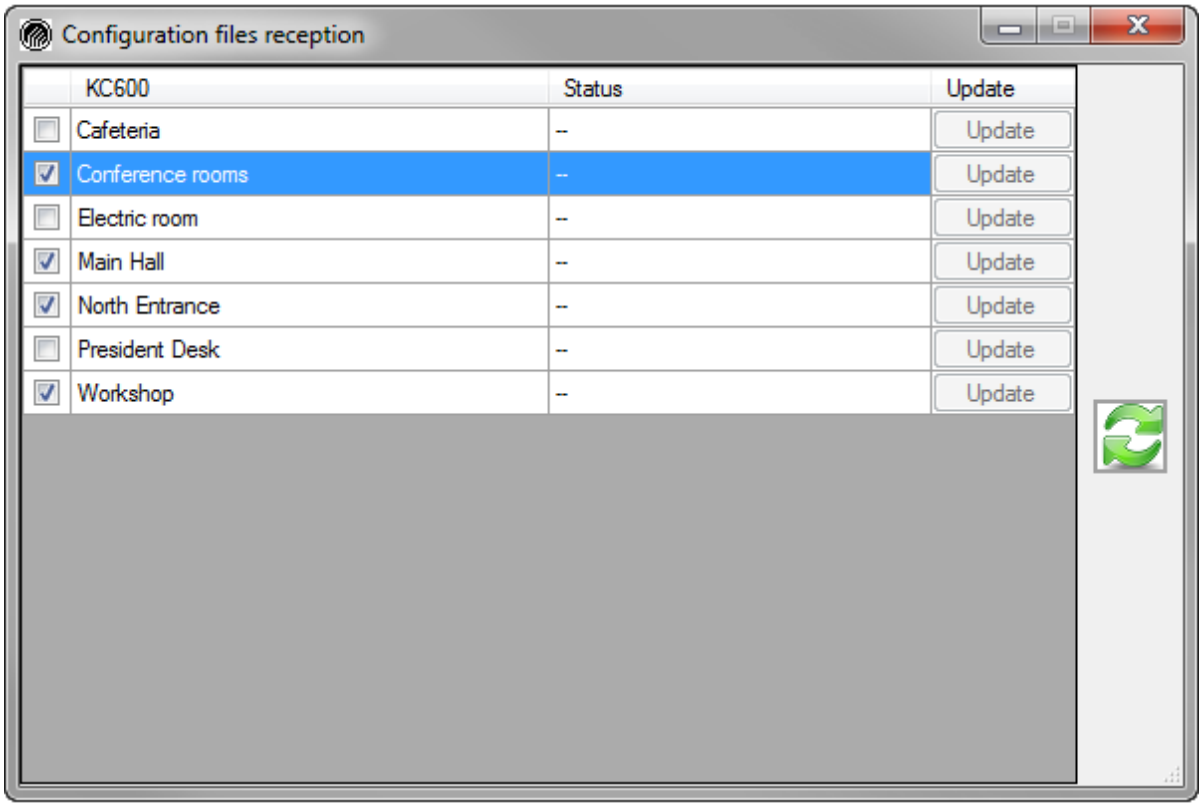


Figure 11: Window allowing you to receive the configuration of the KC600 part of your current installation.

- Check the box for every *KC600* that you want to update with the actual configuration contained in the corresponding *KC600* controller(s).
- When the mouse cursor is in the frame displaying the list of *KC600*, you may click the right button of your mouse to display a sub-menu allowing you to:
    - *"Select all"* (to check every *KC600* in the list),
    - *"Deselect all"* (to uncheck every *KC600* in the list).
- To start the reception of the configuration files from the *KC600*, press the button:  .

"*Status*" column messages

After starting the reception process:

- The *KC600* that were not selected remain with the status message with two dashes *("--").*
- When a *KC600* controller is currently being interrogated, the *"Status"* field displays: *"Processing…"*, followed soon by a message indicating how the exchanges went:
  - **If the connection succeeded**, the text displays:
    - *"No significant differences"*, if no important changes between the configuration of the *KC600* and the one in the console were found concerning:
      - the *KC600* main characteristics (mainly its name and *instance number*);
      - the number of *Modbus* cards controlled by the *KC600* (to add inputs/outputs) and the address of each *Modbus* card.

      NOTE: The files may be quite different, but not regarding the basic parameters that identify a *KC600*. Thus, it is better to *"update"* the *KC600* configuration on screen with the data received (see below).
    - *"Different files"*, if there are significant differences (for the characteristics just described) between the configuration in the *KC600* and the one in the console.
    - In both cases, an **"*Update*" button** becomes available to replace the current configuration (in the console) with the one retrieved from the *KC600* card(s).
  - **If the *KC600* did not answer at all**, the "*Status*" field indicates: *"Connection to the KC600 failed".* In that case, try a second time before checking if the *KC600* is plugged in, running well and properly connected to your network. Refer to *2.3.1 Add KC600* for the steps to follow when you cannot access a *KC600* controller from your console
  - **If the connection to the *KC600* has been established but there were problems during the data exchange,** you will get a message indicating the nature of the failure:
    - *"Communication error, try again"*: definitely the most common error message encountered, even though it happens rarely. In that particular case, try again once or twice before concluding that the *KC600* is malfunctioning…
    - *"FTP connection error":* an error occurred after a successful connection to the IP address.
    - *"Invalid file, transmission failed":* if an error was detected in the content of the configuration file received from the *KC600*.
    - *"An exception occurred"*: for any other exchange error.

- When the "*Reception*" requests are completed, you may close the window by clicking on button:  .

2.3.6 Send config.

By selecting the "*Transmit*" option in the "*KC600*" menu, a window appears with the list of every *KC600* integrated in the current *installation*:

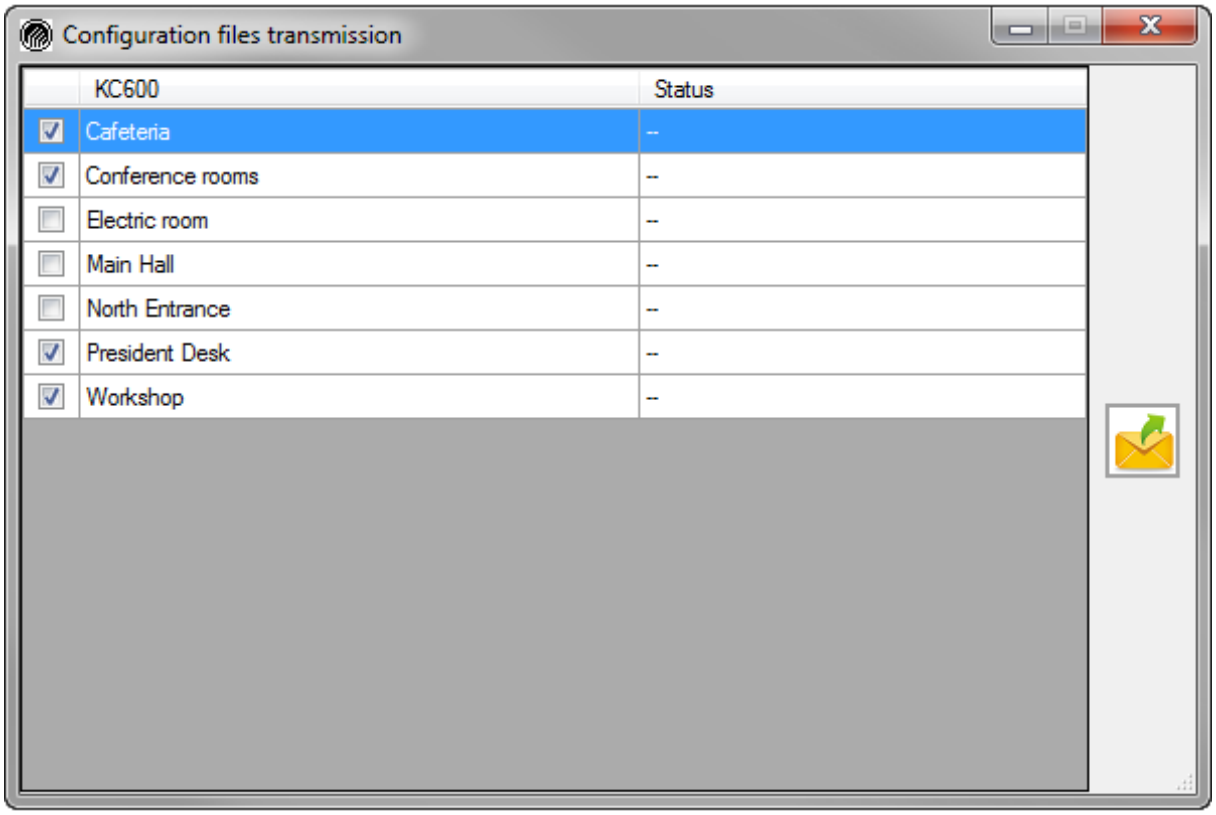


Figure 12: Window to transmit the current configuration to the selected KC600.

- Check every *KC600* for which you want to upload the configuration contained in the *KCF600* console.
- When the mouse cursor is in the frame displaying the list of *KC600*, you may click the right button of your mouse to display a sub-menu allowing you to:
  - *"Select all"* (to check every *KC600* in the list),
  - *"Deselect all"* (to uncheck every *KC600* in the list).
- To start the transmission, press on button:  .

"Status" column messages

- The *KC600* that were not selected remain with the status message with two dashes *("--")*.
- When a *KC600* controller is being interrogated, the "*Status*" field displays: *"Processing…"* followed by a message indicating how the exchanges went:
  - If the transfer succeeded, the message will indicate "*Transmission successful*".
  - If the *KC600* did not answer or answered with an error, refer to the error message listed in section "*Status*" column messages, chapter: *"2.3.5 Receive config."*
- When the "*Transmission*" requests are completed, you may close the window by clicking on button:  .

2.3.7 Check/Change date/time

By selecting the "*Check/Change date/time*" option in the "*KC600*" menu, a window appears with the list of every *KC600* already integrated in the current *installation*:

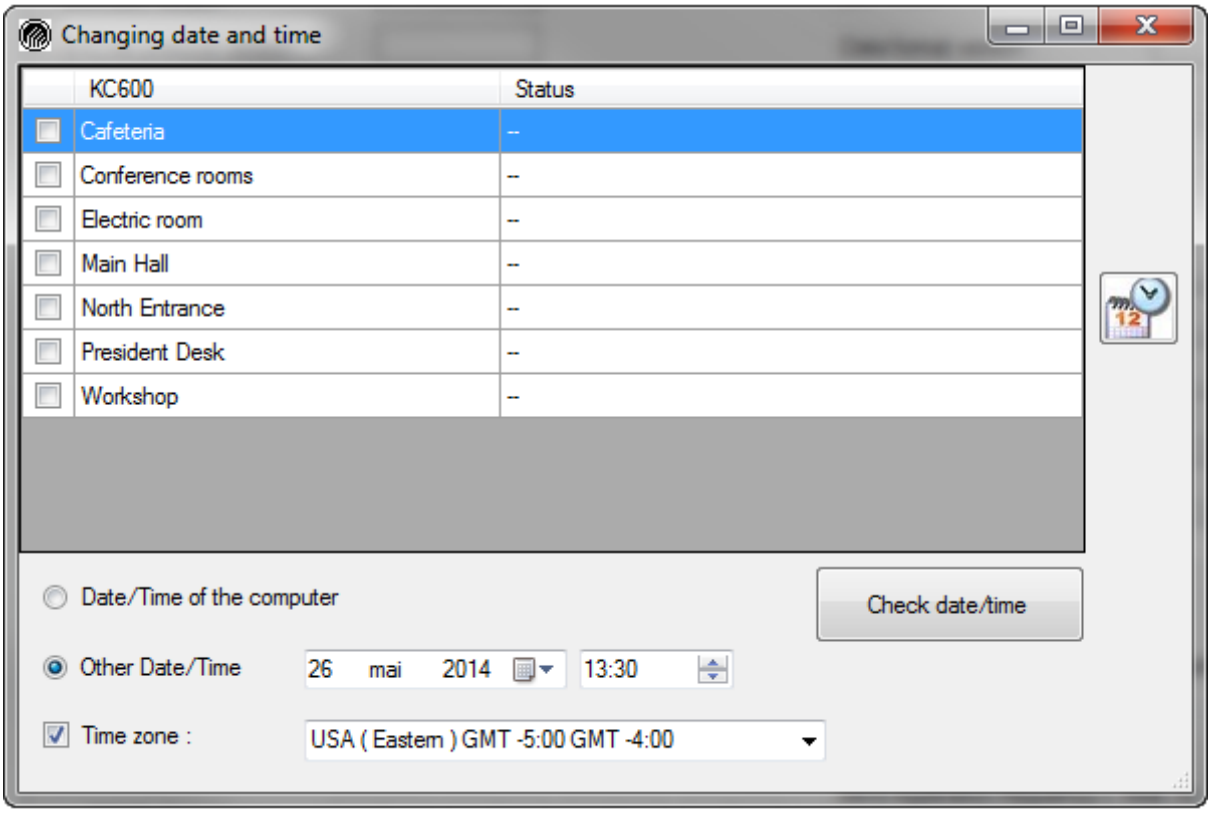


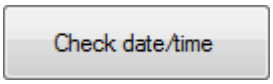
Figure 13: Window to set date and/or time for selected KC600 controllers.

Upper section: *"KC600"* column

This column lists the names of all *KC600* already integrated in the current *installation*:

- Check every *KC600* for which you want to transmit or receive the date/time coordinates.
- When the mouse cursor is in the frame displaying the list of *KC600*, you may click the right button of your mouse to display a sub-menu allowing you to:
    - *"Select all"* (to check every *KC600* in the list),
    - *"Deselect all"* (to uncheck every *KC600* in the list).

2 buttons: 2 possible actions

- **Get date and time:** 

  Press this button to verify what is the current date and time inside each one of the *KC600* selected in the upper section.

  BEWARE: The time zone is not returned by this command.

- **Send date and time:** 

  Press this button to transmit the date and time selected in the **window lower section**:

  - **Date/Time of the computer:**

    By checking this option, the selected *KC600* will be synchronized with your computer (date and time).

  - **Other Date/Time:**

    By checking this option, you may set the selected *KC600* to any date and time that you want.

    - **Date change:**
      - You can click on any of the 3 fields *(day / month / year)* to change its value from the keyboard, by entering the required text (or numbers) or by using the up and down arrow keys.
      - You can also click on the icon appearing right to the date, to display a *Windows* calendar to set the date.
    - **Time change:**

      You can click on each one of the 2 fields *(hours: minutes)* to change the corresponding value:
      - from the keyboard, by entering the required numbers or by using the up and down arrow keys;
      - by clicking on the arrows appearing to the right of the 2 time fields *(hours: minutes)*.

  - **Time zone:**

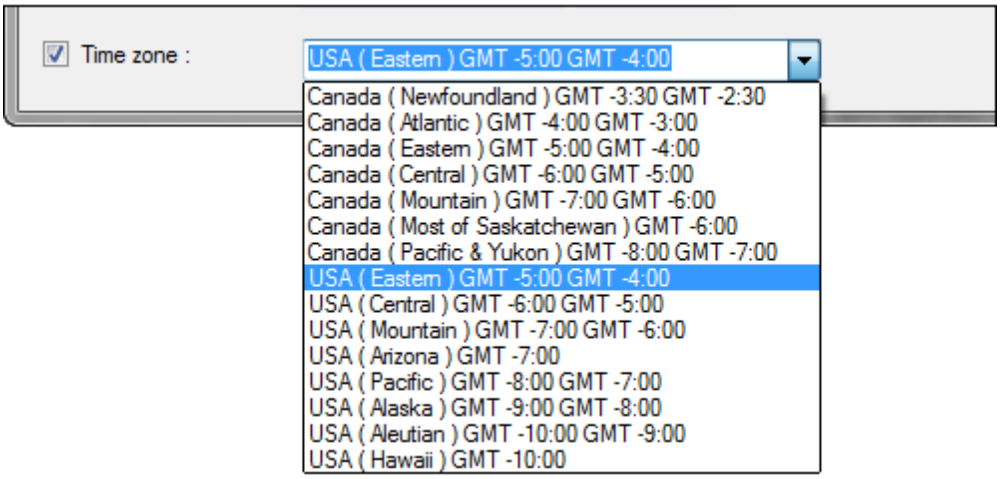


Figure 14: Section to set the time zone, from a list of the ones already defined for North America.

Checkbox left to the *"Time Zone"*:

- When checked, the transmission of date/time coordinates will automatically send the selected value for the time zone as well (see parameters: *"Longitude"* and *"Latitude"* in the text describing *Parameters used to configure the KC600 as a BACnet object of "Device" type*, in section *2.5.2.1 KC600 card configuration parameters*).

Choosing a *"Time Zone"*:

- The list of available time zones is limited to the ones in Canada and in the United States.
- It is very importance to establish the right time zone based on the installation site of the *KC600* network, since this information is used to calculate:
  - the sunrise and sunset time (used in the definition of some schedules);
  - the automated change of times occurring at certain dates of the year (for daylight saving times).
- This information is not saved in the *KC600* configuration file and will not be obtained by pressing the "*Check date/time"* button.
- The value displayed in the time zone box:
  - may not correspond to the value contained in the selected *KC600*;
  - consists of the last value selected by the user…

Upper section: *"Status"* column

This column displays messages informing you on how the exchanges are going between the *KC600* and the console, after starting one of the 2 possible actions (send/receive):

- *"--"*: indicates simply the absence of message, especially for the *KC600* that were not selected…
- *"Processing…"*: displayed when a *KC600* controller is being interrogated.
- *"Connection to the KC600 failed":* this message indicates that the console was not able to establish a good connection with the *KC600* controller.
- An exchange completed with success is signalled by one of those messages:
  - For the action: ***"Check date/time"***

    "KC600 : 20YY-MM-DD HH:MM / This PC : HH:MM"

    where:
    - *"20YY-MM-DD"* is the date,
    - "*HH:MM"* is the time.

    Note that the *"PC"* time, appearing to the right, is the computer time at the moment the answer was received from the *KC600,* which allows you to compare if the *KC600* is well synchronized with your computer.

  - For the action: ***"Send date/time"***

    "Transmission successful".

Synchronization methods through *BACnet* request:

BACnet write commands, to set the *KC600* date/time values:

- ***"Time Synchronization" command:***

  This *BACnet* request sets simultaneously **the date and the time** as they are entered with the command, i.e. regardless of the *"UTC"* reference:
  - without applying any offset related to the current time zone in the *KC600* controller,
  - without time adjustment according to the application of *"Daylight Saving Time"*.

- ***"UTC Time Synchronization"* command*:***

  The *"Coordinated Universal Time"* is a time scale adopted as an international time standard by most countries around the world.

  The difference between this *BACnet* command and the previous one: transmitted date and time refer here to *"time zone zero"*, also called: *"Western European Time"*. So the time parameter passed here is the *"UTC time"* (instead of the *"local time"*).
  - The *"local time"* in the *KC600* will need to be adjusted according to the new *"UTC time"* received and the effective *"time zone"* in the *KC600*. To determine the new *"local time"*:
    1. the *"UTC Offset"*, positive or negative, must be subtracted from *"UTC time"* received with the command;
    2. if *"Daylight Saving Time"* is effective for the current date in the *KC600*, for the time resulting from the subtraction, an hour must be added to the result of step 1.
  - By applying these offsets, you may get a time shift causing a date change, one day backward or forward.

- ***"utc-offset"*** property of the *"device"* object, representing the *KC600* controller:

  By sending a *BACnet* request: *"WriteProperty"* for this parameter, you may change the *"time zone"* inside the *KC600*, the signed value provided representing a time shift in minutes to apply to the current ***"UTC time"*** in the *KC600*.
  - After the reception of a new "utc-offset" parameter, the "local time" will be adjusted automatically.
  - *BACnet* standard: *"ANSI/ASHRAE Standard 135-2010"* allows *"utc-offset"* values between:

    -780 and +780 minutes.
  - You may enter offset values not corresponding to any real time zone, not used anywhere on the globe.
  - As this *"WriteProperty"* command does not provide information regarding when to apply *"Daylight Saving Time",* the new *"time zone"* will use the settings from the previous *"time zone"* (selected from the *KCF600* console).
  - If you connected your computer to the *KC600 "console port"* (refer to section: *"3.1.3 QNX Console Port (RS232)"*), after rebooting your *KC600* controller, you will see (through a serial communication utility software) the effective *"time zone"* as follows (for pre-defined *"time zone"*: *"Canada (Eastern) GMT -5 :00 GMT -4 :00"*):

    ```
    TIMEZONE: "EST05EDT04,M3.2.0/2,M11.1.0/2"
    ```

    Starting with this *"time zone"*, if you send a *BACnet* command to change the *"utc-offset"* property (let's say by setting an offset of 233 minutes ➔ 3:53 hours), you will see then the following string:

    ```
    TIMEZONE: "BACnet03:53DST_Shift_Hour02:53,M3.2.0/2,M11.1.0/2"
    ```

***BACnet* read commands, to get the current date/time values in the *KC600*:**

All related *BACnet* requests require a call to only one command: *"ReadProperty"*, targeting our *KC600* as a *"device"* object. The property to read must be one of the following:

- ***"utc-offset"***: to get the time shift (positive or negative value in minutes) set with the effective *"time zone"*, the value returned excluding the advanced hour applied for *"Daylight Saving Time"* period.
- ***"daylight-saving-status"***: to receive a Boolean value indicating if the *"Daylight Saving Time"* is currently effective (*"True"*) or not in the *KC600* controller.
- ***"local-date"***: to receive the current date, including the adjustments imposed by the effective *"time zone"* in the *KC600* controller.
- ***"local-time"***: to receive the "local time", including the adjustments imposed by the effective *"time zone"* in the *KC600* controller.

2.3.8 Update software

This option does not appear with the menu tab *"KC600"* if you logged yourself with the *"Monitoring"* or the *"Configuration"* profile; refer to section: *"2.1.1 Login window"*.

By selecting the "*Update software*" option in the "*KC600*" menu, a window appears with the list of every *KC600* integrated in the current *installation*:

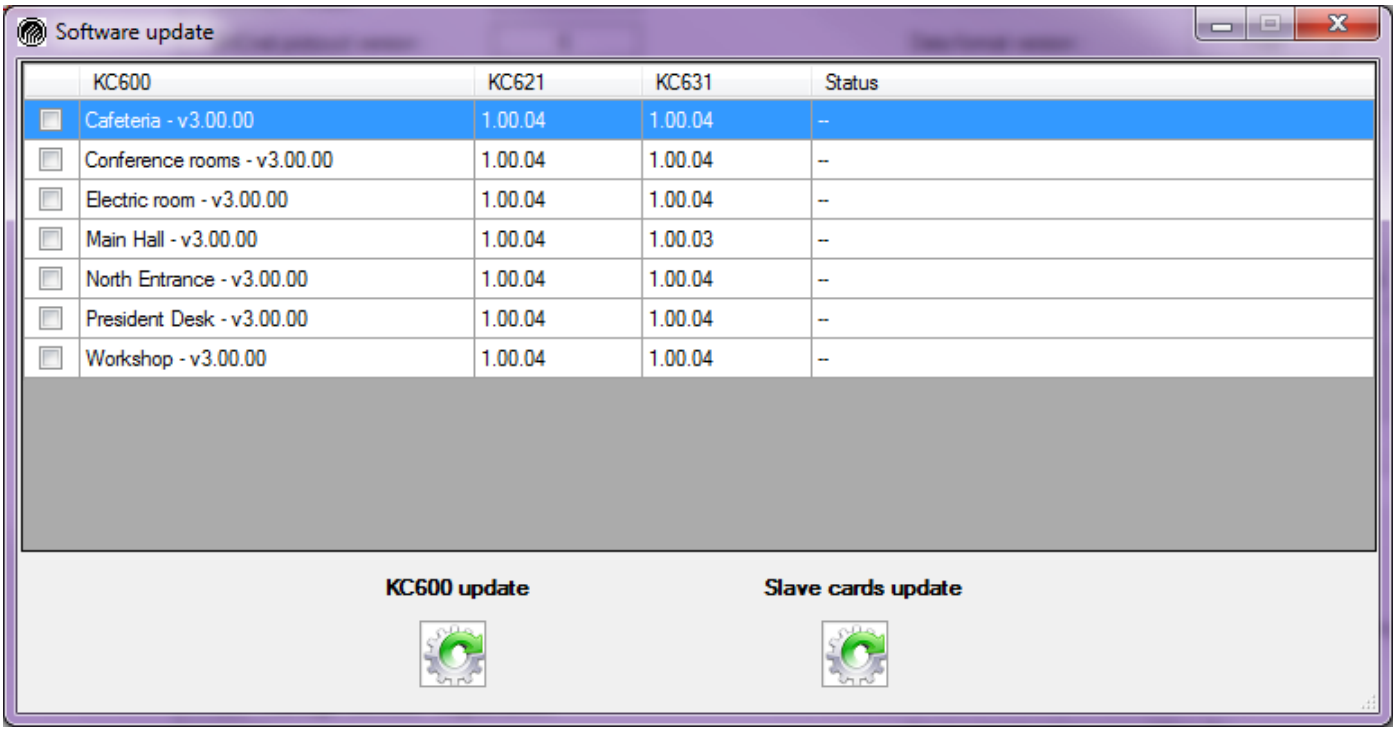


Figure 15: Software update window.

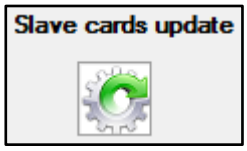
- Check every *KC600* for which you want a software update.

- When the mouse cursor is in the frame displaying the list of *KC600*, you may click the right button of your mouse to display a sub-menu allowing you to:
    - *"Select all"* (to check every *KC600* in the list),
    - *"Deselect all"* (to uncheck every *KC600* in the list).

- Button 

:

  If a new software version is available for the management of *KC600* control cards, it will be provided by your distributor as a compressed file (with the extension: *"*.zip"*).
    - If the name of this file is not composed with the term: *"Update"*, then you will lose the configuration in each one of the *KC600s* targeted by this update. So you will need to retransmit, to these *KC600s*, the configuration stored in your *KCF600* configuration console.
    - By clicking on this button, a navigation window will open, allowing you to select the file to download toward all *KC600s* checked in the upper section of the *"Software update"* window.

- Button 

:

  If a new software version is available for the management of any additional card to complete a *KC600*, such as the *KC621* (for 16 additional inputs) or the *KC631* (for 16 additional outputs), it will be provided by your distributor as a compressed file (with the extension: *"*.zip"*), where:
    - *"XXX"* is the version number,
    - *"X"* stays as a letter for a simultaneous update of the *KC621s* and the *KC631s*. Otherwise, "*X"* is replaced by a number to identify which type of extension card (*KC6**2**1* or *KC6**3**1*) will be updated.

  By clicking on this button, a navigation window will open, allowing you to select the file to download toward all *KC600s* checked in the upper section of the *"Software update"* window.

"*Status*" column messages

You must check in this column if the update process was successful for each *KC600* checked in the first column:

- The *KC600* that were not selected remain with the status message with two dashes *("--")*.
- When a *KC600* controller is being interrogated, the "*Status*" field displays: *"Processing…"* followed by a message indicating how the exchanges went:
    - If the transfer succeeded, the message will indicate "*Transmission successful*".
    - If the *KC600* did not answer or answered with an error, refer to the error message listed in section "*Status*" column messages, chapter: *"2.3.5 Receive config."*

- When the "*Update*" requests are completed, you may close the window by clicking on button:  .

2.3.9 Validate

By selecting the "*Validate*" option in the "*KC600*" menu, the right section of the main window will display the results for the links analysis between the different *KC600* configurations in the current *installation*:

- For each inspected *KC600*, a line will indicate:

  "*Validating configuration of KC600 #XXXXX*",

  where "*XXXXX*" is the *BACnet instance number* of the *KC600* being validated.

- If an error is detected in the configuration of this *KC600*, it will be detailed on the following line.

- The most frequent types of errors found are usually related to the associations established in the *groups'* definitions. They happen:

  1. … after a configuration update from the *KC600* to the console (refer to section: *"2.3.5 Receive config."*).
     - A group may then be composed of **outputs** that do not exist anymore because they were on a *KC631* card:
       - that stopped communicating with the *KC600* to which it was connected, because it has been unplugged or the cable linking the cards is defective;
       - that had its *Modbus* address changed, which means the identification number of every output on the card has also changed (refer to section: *"2.5.2.3 KC600 outputs configuration parameters"*).
     - An **input**, configured to control a group, does not exist anymore because it was on a *KC621* card that stopped responding (for similar reasons as for the previous case).
  2. … after deleting a *KC600* from the current *"installation"*, executed from the configuration console (refer to section *2.3.2 Delete*). If some outputs from this *KC600* had previously been included in a group, the validation process will signal the anomalies resulting from this deletion.

- The very last text line indicates the number of errors detected by the validation process.

2.4 Adding a *KC600* configuration from another *"Installation" / "File" tab*

The "*File*" tab allows you to perform 2 types of actions:

- the management of configuration files ("*XML*" format/extension), that define all parameters for a *KC600*;
- how to leave the application (partially or completely).

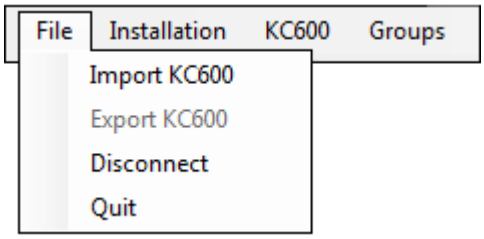


Figure 16: "File" menu items

2.4.1 Configuration files management: *"Import KC600"*

- Each *"installation"* corresponds to a directory (on disk), meaning a folder that has the name of the *installation*, defined in the *KCF600* workspace (= "*WORK*" folder).
- In every directory representing an *installation*, you can find the "*XML*" configuration files of each *KC600* added to the *installation*.
- By clicking on the "*Import KC600*" option, you will get a navigation window to add an "*XML*" configuration file to your current *installation*.
    - The selection of an invalid file will be rejected and an error message will be displayed.
    - If the imported file has an *instance number* already used by another *KC600* in your *installation*, a message will notify you about the problem, asking you if you want to replace the current configuration by the imported one.
    - You can use this option to build up a "*virtual*" *installation*, which could be useful if you want to prepare an *installation* before showing up to the deployment site.
        - You could import repeatedly a model file, but each time you retrieve the file, you must change 2 parameters: the *instance number* and the *name* of the *KC600*, both requiring to be unique in one *installation*; refer to section *2.5.2.1 KC600 card configuration parameters*.
        - When you change the *instance number* through the *KCF600*, the *XML* file is renamed automatically with the new number in the directory for the loaded *installation*.
        - Once on site, a technician must get the *IP communication coordinates* (from the person in charge of the network) and set these values in the *KCF600* for each controller card. Then he must click on the *"KC600"* tab (in the menu bar) to *"send the configurations"* established *"virtually"* for each controller.

2.4.2 Configuration files management: *"Export KC600"*

You may refer to the previous item.

- The "*Export KC600*" item is only available after selecting the **name of a *KC600***, in the list of your current *installation*, but not if you selected an element part of this *KC600*...
- By clicking on this menu item, a navigation window will be displayed allowing you to copy the "*XML*" configuration file wherever you choose.
    - You export a file usually to keep a *"backup"* copy of a *KC600* configuration file.
    - It is strongly advised not to change the name set to the *KC600* by the configuration software.
    - You cannot export a file in the current *installation* folder. If you want to do so, use instead the *"KC600"* tab (in the menu bar) to *"Save"* the configuration; refer to section *2.3.3 Save*.

2.4.3 Software "*partial*" exit: *"Disconnect"*

This option does not close completely the application. It disconnects the user and sends him back to the "*Login*" window (see section *2.1.1 Login window*).

2.4.4 Software "*complete*" exit: *"Quit"*

This option disconnects the user and completely closes the *KCF600* application.

2.5 *KC600* configuration / Main window composition

After completing the *"login"* process, the main window opens displaying the *"installation"* that was loaded the last time the application was used.

2.5.1 Left part of the main window: list of *KC600* and their components

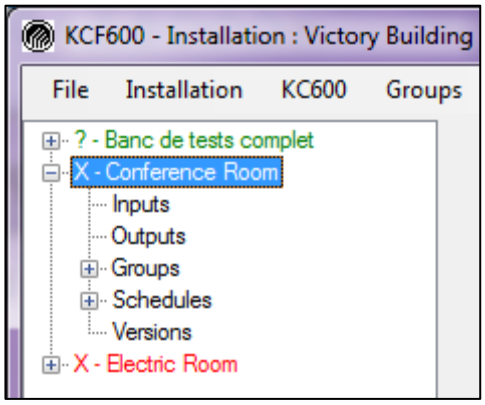


Figure 17: Main window: list of KC600 included in the current installation.

The left column lists alphabetically (except right after renaming a *KC600*) every *KC600* part of the current *"installation"*.

2.5.1.1 *KC600* identification

Each line identifying a *KC600* is composed of 3 sections (from left to right):

1. A **"+"** or **"-"** sign:
   - By clicking on the *"+"*, you can visualize every *BACnet* object types (except the *"Calendars"*) composing the *KC600* (refer to section *2.5.1.2 KC600 composition*).
   - By clicking on the *"-"*, you can hide the objects composition of the *KC600*, only the device name being shown.
2. A symbol indicating the **state of the *KC600* configuration** in the console, compared to the one in the corresponding *KC600* card:
   - **"?":** the interrogation mark means the application does not know if the last recorded configuration is the same as the one in the *KC600* card. It also indicates that no attempt was made to communicate with the *KC600*, so we do not know if it is still connected to the *Ethernet* network. This is why, when the *KCF600* loads an installation, all its *KC600* are marked with a quotation mark.
   - **"0":** this symbol confirms that the configuration on screen and the one in the controller are the same. So the *KCF600* established communication with the corresponding *KC600*:
     - by receiving and updating the configuration file in the controller,
     - by sending/transmitting the configuration on screen to the *KC600*.
   - **"X":** this symbol indicates that the configuration in the console was modified, therefore does not correspond to the one in the *KC600* controller (refer to section: *"2.3.6 Send config."*). After a successful transfer of the configuration to the controller, the symbol: *"X"* will be replaced by a: *"0"*.
3. The name attributed to the *KC600* controller.

The line color identifying a *KC600* indicates if the configuration in the *KCF600* is different compared to the one saved on disk, in the directory of the *"installation*.

- ***Red:*** it indicates that the configuration on screen was not saved on disk after the last modifications made by the user.
- ***Green:*** it indicates that the configuration file in the *installation* folder is up to date (was saved after the last modifications) with what we have on screen. The characters that identify a *KC600* will pass from ***red*** to ***green***:
  - after saving on disk the modified configuration. Remember that if these changes were not sent to the *KC600* controller, the ***"X"*** symbol will still appears to the left of the *KC600* name;
  - after transmitting with success a modified configuration to the *KC600* card, as the configuration is then automatically saved on disk.
- ***Black (highlighted text):*** regardless of the symbol appearing to the left, the name of the *KC600* in the list will be black (highlighted in gray) when it is the currently selected *KC600* in the list. To see the real *"state color"* for this *KC600*, you must select another *KC600* in the list or an object composing the *KC600*.

  NOTE: According to the parameters set for your display and the version of your *Windows* system, the text will remain highlighted, but not necessarily in black with a grey background…

2.5.1.2 *KC600* composition

If the line identifying a *KC600* in the list starts with the *"–"* symbol, you can see the 4 object types* composing the device (excluding the *"Calendars"* objects, representing the *special days* when there are schedules with *exception days* referring to such days).

1. **Inputs:** by clicking on that item, the right window displays the list of all inputs directly controlled by the *KC600*; refer to section *2.5.2.2 KC600 inputs configuration parameters*.
2. **Outputs:** by clicking on that item, the right window displays the list of all outputs directly controlled by the *KC600*; refer to section: *2.5.2.3 KC600 outputs configuration parameters*.
3. **Groups:** the display mode for groups differs from the previous types:
   - If you click on the *"Groups"* text, the right frame of the window will be empty.
   - If there is no *"+"* (or *"-"*) symbol appearing left to the *"Groups"* text, it means this *KC600* does not have any group.
   - You have to click on the *"+"* symbol to see the list of groups controlled by the *KC600*.
   - By clicking on any name in this list, the information of the corresponding group will be displayed in the right part of the main window; refer to section *2.5.2.4 KC600 group configuration parameters*.
4. **Schedules:** the display mode for schedules is the same as for the groups; refer to the specification for the *"Groups"* just above.
   - By clicking on the name of a schedule in this list, the right part of the main window will show the composition of this object, including the group(s) targeted, the list of events to turn them on or off, during a *normal week* and for *special days*; refer to section *2.5.2.6 KC600 schedules configuration parameters*.

5. **Versions:**

Card type	Address	Version number
KC600	0	3.00.00
KC600 (BSP)	0	3.0.0
KC631	1	1.00.04
KC621	2	1.00.04
KC621	8	1.00.04
KC631	9	1.00.04
KC621	13	1.00.04
KC631	15	1.00.04

Figure 18: Software versions list for each control card linked directly to a KC600.

- After updating the *KCF600* console with the configuration established in the selected *KC600* (refer to section: *2.3.5 Receive config.*), click on this option to get the software versions currently running in the *KC600* controller itself (the software managing the behaviour of the *KC600*, plus the *"BSP"* software loaded at start up) and the versions for each additional card (*KC621* et *KC631*) connected directly to this *KC600*.
- The (Modbus) *"Address"* column reflects the number established on the rotary switch (enabling values: between 1 and 15, "0" representing the *KC600* itself), found on each extension card; refer to the *"Address Rotary Switch"* sections, at the end of this document.

*** Notes:**

- If your application is not registered (refer to section *2.1.1 Login window*), only *"inputs"* and *"outputs"* will be displayed, not the *"Groups"* neither the *"Schedules"…*

  "Calendar" BACnet objects are not appearing in this window. They are defined in sub-menu: *"Special days"*, fount under the title menu: *"Schedules"*. From a *BACnet* network monitoring console, a *KC600* will only show as *"Calendar"* objects the *"special days"* pointed by any schedule(s) defined in this *KC600*, if there are reference(s) to calendar(s).

2.5.2 Right part of the main window: selected *KC600* parameters

What is shown in this section of the window depends of the object type selected in the *KC600* list to the left.

2.5.2.1 *KC600* card configuration parameters

By clicking on the name of a *KC600* in the left part of your main window, the following parameters should appear on the right side:

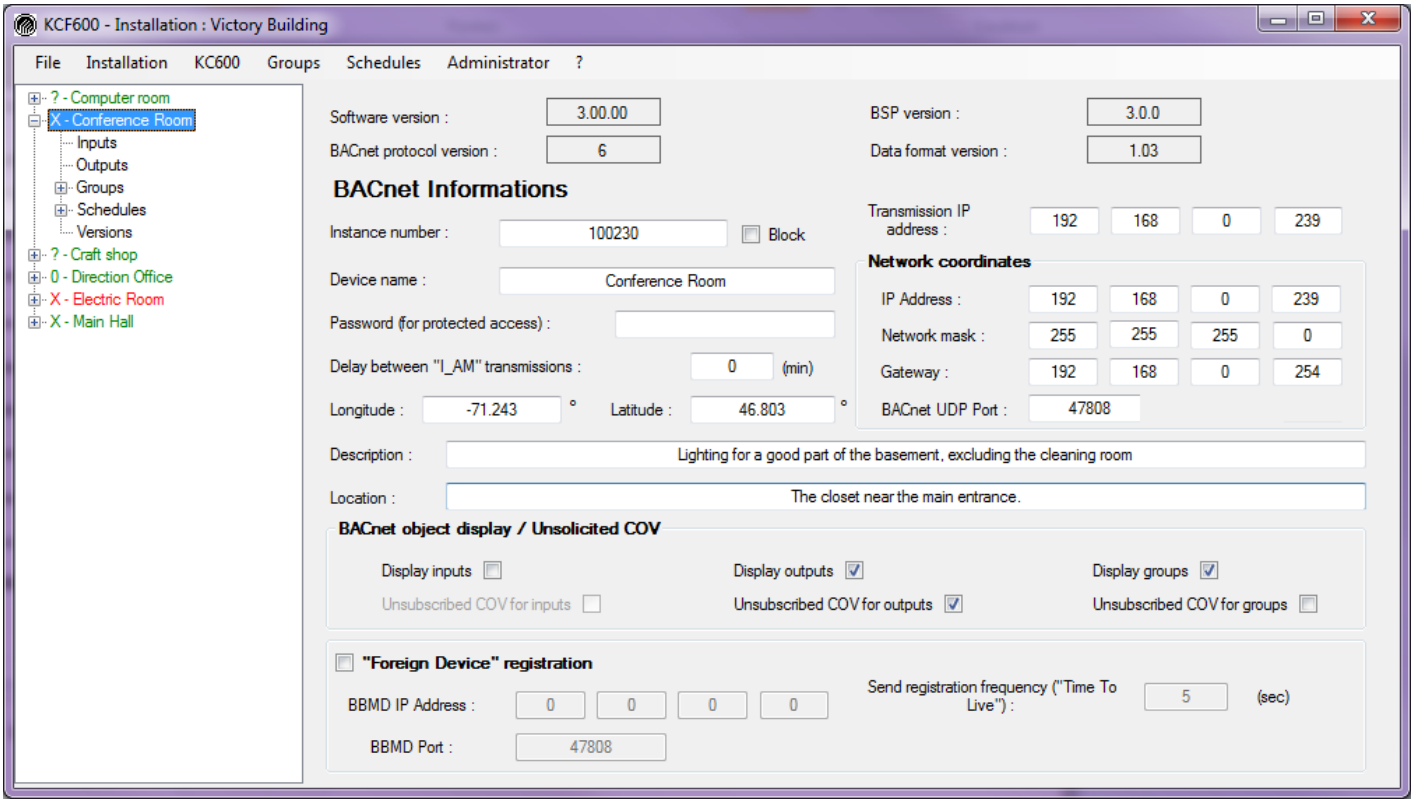


Figure 19: KC600 parameters configuration window.

Software versions

The parameters at the top of the window show the versions for the various software components in the *KC600*:

- the ***"Software version"*** of the main application managing the *KC600*;
- the ***"BACnet protocol version"*** supported by the *KC600 "BACnet stack",* which is the part of the application software that takes care of all *BACnet* communications. The protocol version is an indication of the *BACnet* commands and types of objects that are supported by your *KC600* controller. Also, the answer to some *BACnet* requests may vary depending on the protocol version supported;
- the ***"BSP version"*** (*"Board Support Package"*) reflects:
  - under which version of *QNX* system our main software is running to manage the *KC600*;
  - certain libraries updates used by QNX, including the ones particular to *KC600* management;
  - the loader version that rules the *KC600* card start-up;

- the *XML* ***"Data format version"*** is an indication on how the information is organized in the *XML* configuration file that defines all parameters for a *KC600* controller. This number increases with each new version, with the addition of new parameters and sometimes due to the reorganization of the structure.

BACnet Information

IP communication

The right side of the window allows you to establish the different parameters for IP communication with the *KC600* controller:

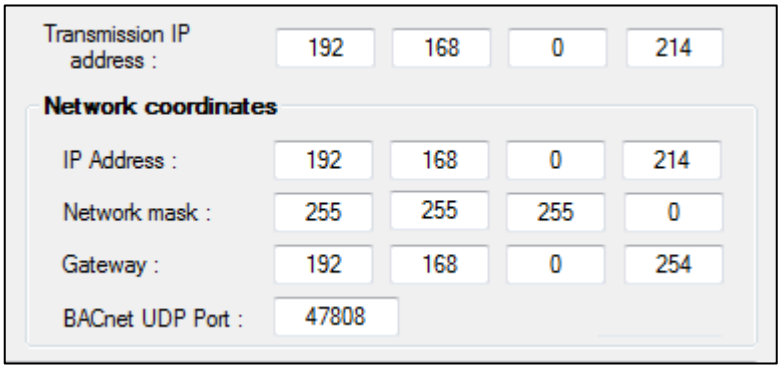


Figure 20: BACnet Information - Coordinates used to integrate the KC600 to an IP network.

- ***"Transmission IP address":*** this is the *"target address"* to reach a *KC600* through *TCP-IP* communication (not *UDP-IP*), to transmit/receive a configuration file to/from the *KC600*.

Network coordinates

At start, this bloc reflects the current IP address information of the *KC600* controller. However, if you change those parameters and then send the modified configuration to the *KC600,* the device will respond to those new IP coordinates. Before performing this task, make sure those IP coordinates were provided by a client's representative who has the competence and responsibility to manage and configure the network to which the *KC600* must be integrated.

- ***"IP address":*** not to be confused with the "*Transmission IP address*" parameter:
  - This field is used to set the new IP address to which the *KC600* will answer after receiving the modified configuration. In other words, this field is used to change the IP address of the *KC600*.

- ***"Network mask":***

  It is important to give the right ***"network mask"*** to the network coordinates for your *KC600* because failing to do so will result in:
  - the communication between the console and the controller working fine,
  - but the *KC600* will be unable to communicate with the other devices on your *BACnet IP* network.

- ***"Gateway":***
  - You must enter a value in this field even if there is no gateway.
  - Generally, this field contains the same values as the *"IP address"*, except for the numbers where the *"Network mask"* is *"0"*. You must fill those fields (in the *"Gateway"* address) with value: *"254"* (refer to *Figure 20*).

    Technical note: In fact, to complete the *"Gateway"* address generated from the *"IP address"*, the zero bits in the "*Network mask"* should be reversed for the *"Gateway"* address, except for the less significant bit that must be kept to zero for each group of 8 bits in this mask…

- ***"BACNET UDP Port":***
  - Normally established to "47808", ("**BAC**0" in hexadecimal representation for *"**BAC**net"*).
  - Except for rare cases, this parameter does not need to be modified.
  - Values higher than "65535 / 0xFFFF" are rejected. Make sure also not to use a port number that is already used or reserved for other services.

NOTE on bad values in block *"Network coordinates"*:

- If you enter incoherent values in the fields of section *"Network coordinates"* and send this configuration to the *KC600*, the device will then reboot with the new coordinates, but it will reject them (after trying to apply the new address) to take back automatically the previous valid coordinates.

  Examples of rejected coordinates:
  - if you enter an *"IP address"* that does not respect the format of the *"Network mask"* (the initial address byte(s) being not shared);
  - a *"Gateway address"* set to a value that is not compatible with the 2 other network coordinates…

Parameters used to configure the *KC600* as a *BACnet* object of *"Device"* type

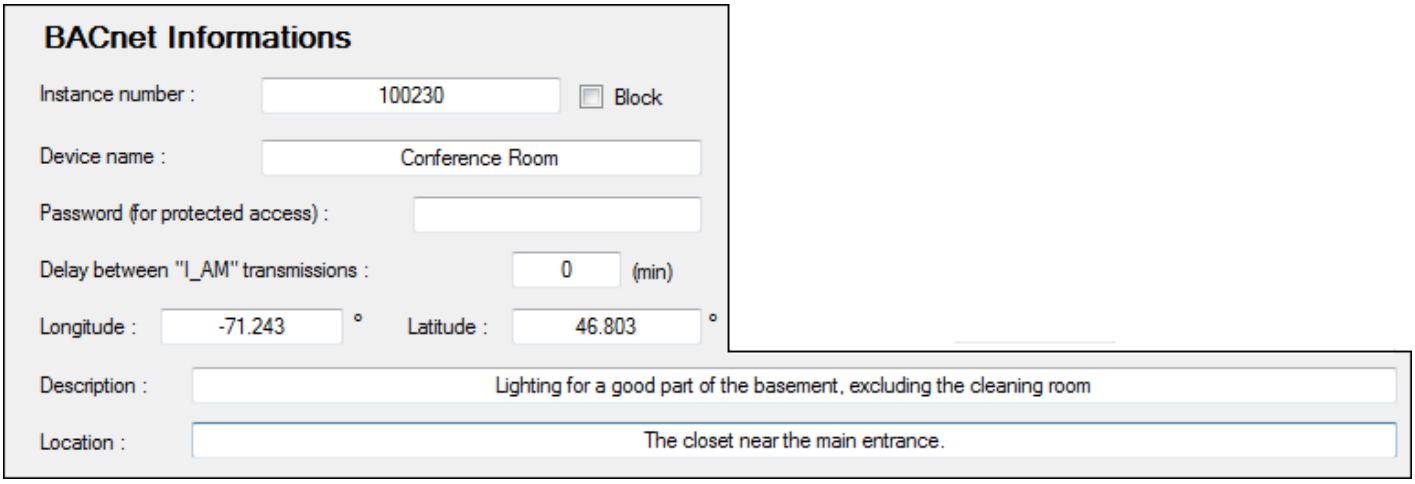


Figure 21: BACnet information: parameters used to configure the KC600 as a BACnet "device" object.

- ***"Instance number"* (or *"BACnet Object Instance Number"*):**

  It is the **identification number** which characterises the *KC600* as a unique *BACnet* object in a *BACnet* network. It acts as a serial number allowing the differentiation of each *BACnet* object. This number must not be shared by any other *BACnet* object installed on the same network. Note that this number is not a network address and it will be used by a *BACnet* console (or another *BACnet* controller) to discover the (IP) network address of a *KC600* controller.

  - You can enter values between 0 and 0x3FFFFF (= 4194303), this last value being used only to indicate that this parameter is *"not established*".
  - This information is not only essential to identify a *KC600* in an *"installation"*, but it is also used to reference this *KC600* outputs for groups on other *KC600*, part of the same *"installation"*. So if you change it, a message will notify you that this change will "*force the saving of every configuration in your installation*":
    - You may either click on "*Yes*" or "*No*" in this message box to confirm or reject the instance number change.
    - If you click on *"Yes"*, the console will update every link between this *KC600* and the others in the current *installation* (including all the descriptions based on a model, all the outputs part of group(s) and referring to this *KC600* …)
    - If you click on *"No"*, the previous instance number will be set back.

- ***"Block"* check box*:***

  Check this box if you want to prevent the possibility to change the *KC600 "device instance number"* from any console that manages a *BACnet* network.

- ***"Device name":***

  Just like the device *instance number*, the name given to the *KC600* must be unique between all device objects found in your *BACnet* network. It allows users to differentiate each controller without having to remember their instance numbers.

  - As long as a custom name is not set by a user, from the *KCF600* or from any *BACnet* console for monitoring/controlling a network, a default name is attributed automatically, composed of:

- the device *instance number*,
- the console display language,

the model used being:

- *"KC600 Lighting Ctrl - XXXXX"* in English,
- *"Ctrl d'Eclairage KC600 - XXXX"* in French,

where *"XXXXX"* is the device *instance number*.

- From the moment the device name is set to a custom string, the model is abandoned, a change of the instance number or the console display language having no more effect on the name.
- Use "*printable*" characters for the device name, which exclude graphic characters and control characters. The French characters can be used, but some *BACnet* consoles may not recognize them…

- ***"Password" (for protected access):***

  Some *BACnet* commands are password secured, like the reinitialize commands or the ones to deactivate a *BACnet* device for a certain period of time.

  - The *KCF600* console and the *KC600* controller allow a password with a maximum length of 20 alphanumerical characters (spaces starting or ending a password being removed).
  - **Warning:** the password is case sensitive…
  - You may leave this textbox empty (default value) if you do not want these requests to be password protected.

- ***"Delay between "I_AM" transmissions":***

  The *KC600* controller sends *"I_AM" BACnet* messages to identify itself on the network:

  1. when the controller starts up, after powering it up or after a "*Cold Start*" reboot (initiated by pushing the "*reset*" button or after receiving a *BACnet* "*Reinitialize - Cold Start*" request);
  2. after receiving a *"Who_Is"* command from a *BACnet* console or another device. The *KC600* only answers to *"Who_Is"* requests when they aim a range of instance numbers that includes its own one;
  3. automatically at the frequency entered in the textbox: *"Delay between "I_AM" transmissions"*:
     - A zero value, strongly recommended, avoids the periodic transmission of "*I_AM*" messages without preventing the *KC600* from answering in cases 1 and 2. A zero value optimizes *BACnet* communication.
     - For a non-zero value = *"X"*: the *KC600* will automatically transmit an "*I_AM*" request every "*X*" minutes. A non-zero value helps for a speedy detection of the *KC600*, without having to reboot the *KC600* or to start the detection process through a *BACnet* monitoring/control console.

- ***"Longitude" / "Latitude":***

  For schedules using the sunrise or sunset time, you must establish the geographical coordinates of the installation site (available on specialized internet sites like *"Google Maps Latitude, Longitude"*):

  - ***Longitude***: angle between +180° (to the right) and -180° (to the left), referencing the Greenwich meridian located in England.

- ***Latitude***: angle between +90° (above) and -90° (below), referencing the earth's equator, halfway between the 2 poles.

- ***"Description":***

  This textbox allows you to write a general "*description*" about what the *KC600* is used for, such as which section(s) of the building is lighted by the controller...

  - This field is not mandatory for the good functioning of the device; therefore you can leave the default value or replace it by an empty string.
  - A default *description* is given based on a model that changes according to the console display language. The default text specifies that the *KC600 "Controls up to 66 inputs/switches, 64 outputs/relays"*.
  - If you enter a custom *description* (from the *KCF600* or any *BACnet* monitoring/control console), the default model will then be abandoned.

- ***"Location"***

  You indicate in this field where is installed the *KC600* controller, strictly informative.

  - This field is not mandatory for the good functioning of the device; therefore it can be left empty.
  - A default *location* is given based on a model that changes according to the console display language. The default text only mentions to "*Indicate here where the KC600 is installed".*
  - If you enter a custom *location*, the default model will then be abandoned.

BACnet object display / Unsolicited COV

Figure 22: Choosing which objects are displayed in a BACnet console, with possibility to send their "Change Of Value".

This section establishes the *"object types"* that will be available when a *BACnet* monitoring/control console displays all *BACnet* objects found in our *KC600*.

- ***Display inputs:***

  - By checking this box, a *BACnet* console will detect every "*Binary Input*" objects controlled by a *KC600* controller, which means both inputs on the *KC600* and all 16 inputs on each *KC621* card connected to the *KC600*.
  - Making this type of object available to a *BACnet* monitoring/control console provides few pertinent data; it may be required to:
    - use the state change of an input to trigger another *BACnet* object;
    - view the state of each input in a diagram representing the floors of the implementation building.
  - Keeping the *"Binary Inputs"* invisible to *BACnet* will not prevent the good functioning of the links between inputs and groups (refer to section *2.5.2.4 KC600 group configuration parameters*), because these links apply inside the same *KC600*, shared by both types of objects.

- ***Display (individual) outputs:***
    - By checking this box, a *BACnet* console will detect every "*Binary Output*" type objects representing an individual output, directly controlled by the *KC600* controller, which means all 16 outputs connected on the *KC600* card and all 16 additional ones on each *KC631* card linked to the *KC600*.
    - It is **strongly recommended to always check this box**, individual outputs being regularly required to:
        - view or change the current state of each relay in a diagram;
        - see every priority currently used for each output;
        - if you created a group in *KC600* #1 including relays on *KC600* #2, then if the group is set to *"ON"* or to *"OFF"*, the remote relays on *KC600* #2 will be set or reset accordingly, but only if the individual outputs on this *KC600* are visible/displayed…

- ***Display groups (of outputs):***
    - The "*groups of outputs*" are "*proprietary*" *Gentec* objects; they are not related to the *BACnet* "*Group*" object type. In the *BACnet* environment, they are represented by the same type of object as for the individual outputs: "*Binary Output*".
    - By checking this box, a *BACnet* console will detect every *"Binary Output*" representing a group of relays in the *KC600* (which excludes the individual outputs).
    - This box is usually checked for the same reason as for the individual outputs, even though it is not as important because:
        - usually, groups are not represented in a building diagram;
        - the state of a group does not necessarily represents the state of the relays composing it;
        - if you linked an input to a group, not displaying the groups will not affect this link.

COV ("*Change Of Value*") services:

- These services prevent the use of continuous *"polling"* to detect possible changes with the state or status for any object controlled by a device (in our *KC600*, this detection is limited to the *"Binary Input"* and *"Binary Output"* objects). Thanks to the COV services, when such a change happens, a message is transmitted automatically by the *"device"* controlling the modified object.
- The *"notification"* message sent by the subscribed object provides 2 parameters (for our objects in the *KC600*):
  - *"Present Value"* ➔ the new *"ON"* or *"OFF"* state;
  - "*Status Flag"* ➔ 4 flags (in one byte):
    - in *"FAULT"*, to signal a loss of communication or in the case of an individual output, if the relay is disconnected;
    - *"IN_ALARM"*, this flag being raised automatically when the previous flag is set;
    - *"OVERRIDDEN"*, not used in the *KC600*;
    - *"OUT_OF_SERVICE"*, which is raised when the corresponding simulation mode is activated from a *BACnet* console. When you send a change of state to an object in this mode, the new state is not applied to the hardware associated with this object, the new value being managed *"virtually"*.
- The *KC600* supports 2 types of *"COV"* services:
  - ***"Subscribed COV"* (with or without confirmation):** this type of service requires a subscription issued from a *BACnet* console or a *BACnet* object to the *KC600*:
    - one subscription is required for one object at a time:
      - each object may supports many subscriptions, from different sources;
      - there must be one subscription for each object that needs to be monitored;
    - each subscription of this type must be renewed by the subscriber after a certain period of time (in minutes);
    - when the subscribed object changes, it will only contact the subscriber(s), through *"unicast"* message(s).
  - ***"Unsolicited COV"*:** this type of service can be activated by checking a box in the *KCF600* configuration console. In such a way that:
    - all *KC600*'s objects, that correspond to the type associated with the checked box, are then subscribed to transmit a *"COV notification"* message with each occurrence of a change in their state or status;
    - this type of subscription doesn't need to be renewed;
    - all devices, sharing the same *BACnet* network as the *KC600*, will then be notified by a single *"broadcast"* message with each detection of a new state or status.

- ***Unsubscribed/Unsolicited COV for inputs*:**
  - This check box is available only if you previously checked the box: *"Display inputs"*.
  - A "*broadcast"* message will be transmitted every time a change of *"On/Off"* state or status (refer to the theory in the upper frame) happens for any input, represented as a *"binary input"* object in the *KC600*.

- ***Unsubscribed/Unsolicited COV for (individual) outputs*:**
  - This check box is available only if you previously checked the box: *"Display outputs"*.
  - A "*broadcast"* message will be transmitted every time a change of *"On/Off"* state or status (refer to the theory in the upper frame) happens for any individual relay, represented as a *"binary output"* object in the *KC600*, which excludes the *"binary output"* objects associated with the groups of outputs.

- ***Unsubscribed/Unsolicited COV for groups (of outputs)*:**
  - This check box is available only if you previously checked the box: *"Display groups"*.
  - A "*broadcast"* message will be transmitted for each *"On/Off"* state change or status change (refer to the theory in the upper frame) that happens for any group of outputs, represented as a *"binary output"* object in the *KC600*, which excludes the *"binary output"* objects associated with the individual relays.

"Foreign Device" registration

Figure 23: BACnet Information: "Foreign Device" registration.

BBMD: some *BACnet* devices can manage a *"BACnet Broadcast Management Device"*.

- In general, routers do not allow the propagation of *"broadcasts"* between subnets. In *BACnet*, "*Who_Is*" requests are regularly used, in *"broadcast"* transmissions, not only to detect *BACnet* devices present on the network, but also to discover their IP address from their instance number (mechanism used by the *KC600* for groups composed of outputs on remote *KC600*).
- *BBMD* can relay a *"broadcast" BACnet* request to other remote subnets also containing *BBMD* (one by subnet).
- *BBMD* are composed of 2 tables with network coordinates to contact when a *"broadcast"* request is received on the local network:
  - a coordinates table (IP address, network mask and UDP port number) of each *BBMD* to contact on other subnets;
  - a table of IP addresses for each "*Foreign Devices*" installed on subnets remote to the one of the *BBMD*.

Foreign Device: when a *BACnet* device is installed on a subnet where there is no *BBMD* (usually because the device is alone on this subnet, often on a temporary base), then it has to register as a "*Foreign Device*" in a *BBMD* installed on a remote network. By doing so, the device will be contacted when a *"broadcast"* command is sent on this remote network.

- ***"Foreign Device" registration* checkbox*:***

  This box is checked to activate the cyclic transmissions of *"FD"* registration requests to a *BBMD*.

- ***BBMD IP Address*:**

  It is the IP address of the device where is located the *BBMD*, on another subnet (separated from the *KC600* network by a router or installed in this router), to which the *"FD"* registration request will be sent.

- ***BBMD Port*:**

  - Except in very rare cases, this parameter does not need to be modified.
  - It is usually set to the same port as the one used by the *KC600* for *BACnet* communication; refer to parameter *"BACnet UDP port"* in the *"network coordinates"* block in this chapter.
  - Rejected values: beyond 0xFFFF / 65535. Also avoid using a port number reserved for other purposes…

- ***Send Registration Frequency ("Time To Live")*:**

  - The "*FD*" registration request must be sent regularly for the *BBMD* to keep the address of the *KC600* in its "*Foreign Device*" table.
  - A *BBMD* receiving this request saves the incoming address for a "*Time To Live*" period passed in parameter, plus a supplemental 30 seconds delay. After this period, the "*Foreign Device*" address is deleted from the corresponding *BBMD* table.
  - The lowest accepted value is 5 seconds, but this minimum value is much too fast when you consider the 30 seconds delay of grace applied automatically by the *BBMD*…

2.5.2.2 *KC600* inputs configuration parameters

On the left part of the main window, if you click on a line called "*Inputs*", the right part of the window will display the list of all inputs controlled by the selected *KC600*:

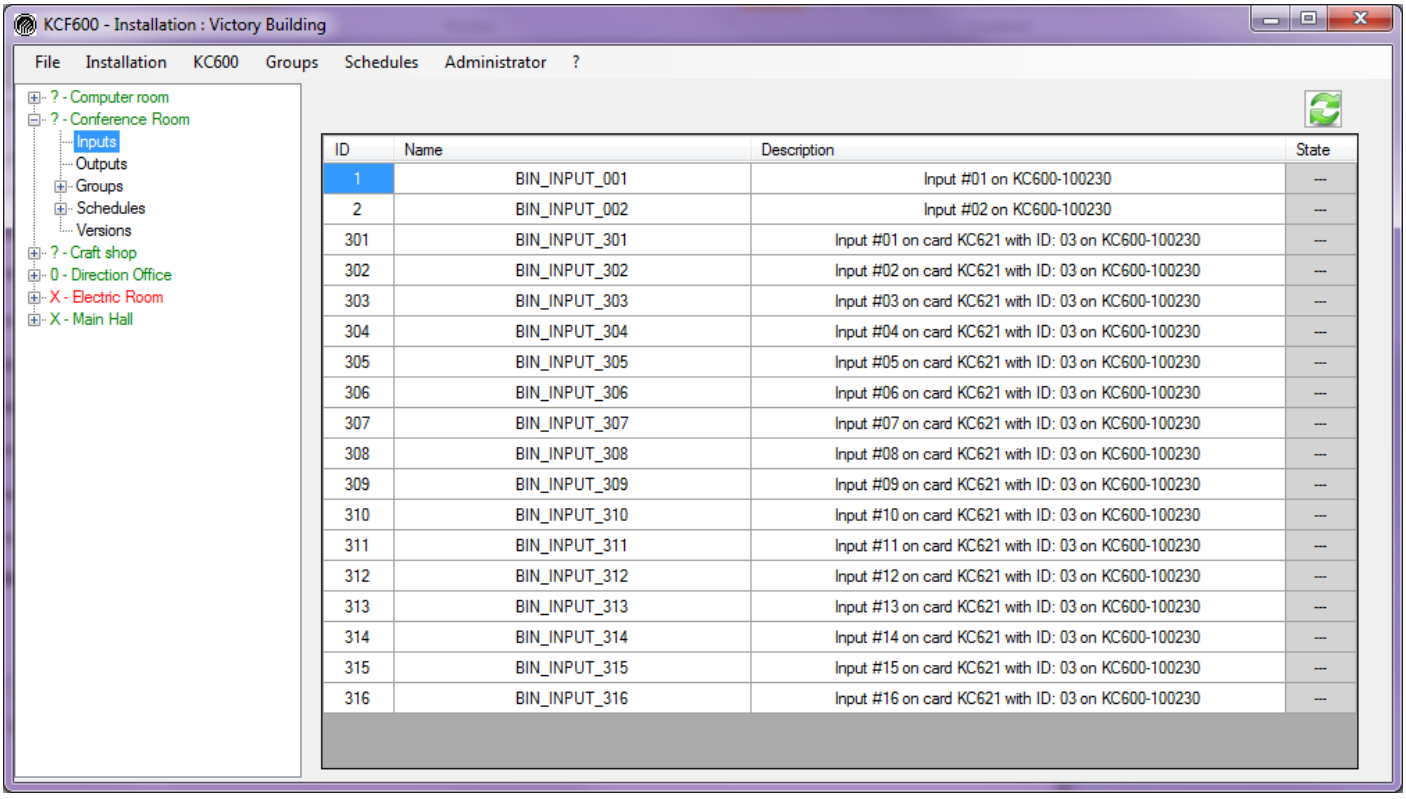


Figure 24: Window showing the main parameters for the inputs in a KC600.

The window is subdivided in 3 columns of information, to identify each input, and 1 column for the input state:

- **Column *"ID"*:**

  It is the *object identifier* of the input as a *BACnet* object of type: "*Binary Input*". This number is fixed by the *KC600* controller and cannot be edited. This number is established according to the following model:

  - The 2 least significant decimal numbers (units and ten values) indicate the position of the input on the card (*KC600* or *KC621*) where they are attached. On a *KC621* card, this number (between 1 and 16) is clearly indicated near the connectors to plug the inputs.
  - The most significant numbers (hundreds and thousands) are obtained by multiplying the ***"card number"*** by 100, this number corresponding to the *"Modbus"* address set on the card where the input is found:
    - The *KC600* card number is zero, so every input on this card has either the ID *"001"* or *"002"* (as there are only 2 inputs on a *KC600*).
    - A *KC621* card is controlled by a *KC600* card via its "*Modbus*" port. The *Modbus* address of this type of card, value between 1 and 15 (= 0xF), is set by a rotary switch (refer to section *4.2 Address Rotary Switch* for a *KC621* card). Note that if you set this switch to zero, the *Modbus* communication is disabled…

      Example: On *Figure 24*, we see binary inputs with numbers between **2**01 and **2**16, which means these inputs are on a *KC621* card with a *Modbus* address set to: "**2**".

- **Column *"Name"*:**

  It is the name that identifies the input in the *KC600*.

  - It can be edited by clicking on the name in the list or from a *BACnet* monitoring/control console.
  - This name must be unique between every *BACnet* object that composes the *KC600*.
  - The *KC600* limits the number of characters to 64 (excluding the *"End Of String"* character) for this property.
  - The default name given to each input is based on the following model: *"BIN_INPUT_XXX"* where:
    - *"BIN_INPUT_"* indicates the *BACnet* object type.
    - *"XXX"* is the input identification number.
  - As soon as a user changes the name of the input, the default model is abandoned.
  - Use "printable" characters for the input name, which exclude graphical and control characters. The French characters can be used although some *BACnet* consoles may not recognize them…

- **Column *"Description"*:**

  This parameter is optional and may be left empty. It is used to provide more information on the nature or the location of the input.

  - It can be edited by clicking on the name in the list or by using a *BACnet* network monitoring/control console.
  - The number of characters is limited to 255 for this property (excluding the *"End Of String"* character).
  - The default description given to each input is based on the following model:
    - for a *KC600*: *"Input #XX on KC600-ZZZZZZ";*
    - for a *KC621*:

      "Input #XX on card KC621 with ID: YY on KC600-ZZZZZZ",

    where: *XX* = number of the input on the corresponding card.

    YY = "*Modbus*" address established on the *KC621* card.

    ZZZZZZ = instance number of the *KC600* controlling the *KC621* card.
  - If you choose French as the display language of the console, each description will be automatically translated. For this change of language to be reflected in the *KC600*, you need to transmit the configuration on screen to the *KC600*.
  - After entering a custom description, the model is abandoned, a change of the console display language or the *KC600* instance number having no more effect on the description.

- **Column *"State"*:**

  This column allows you to check the current state of each input controlled by the *KC600* card.

  - To start the reading process, you must click on button:  just above the *"State"* column.
  - Each data box indicates the state: 

or 

for the corresponding input.

- If the *KC600* card does not answer, you will receive a message to signal the problem. Then, the data in the boxes in the column *"State"* will remain the same, unchanged.
- If a *KC621* card is disconnected, the corresponding box will indicate: ERR for each input connected to this extension card.
- You cannot change the state of an input from the *KCF600* console.

2.5.2.3 *KC600* outputs configuration parameters

On the left part of the main window, if you click on a line with text: "*Outputs*", the right part of the window will display the list of outputs contained in the selected *KC600*:

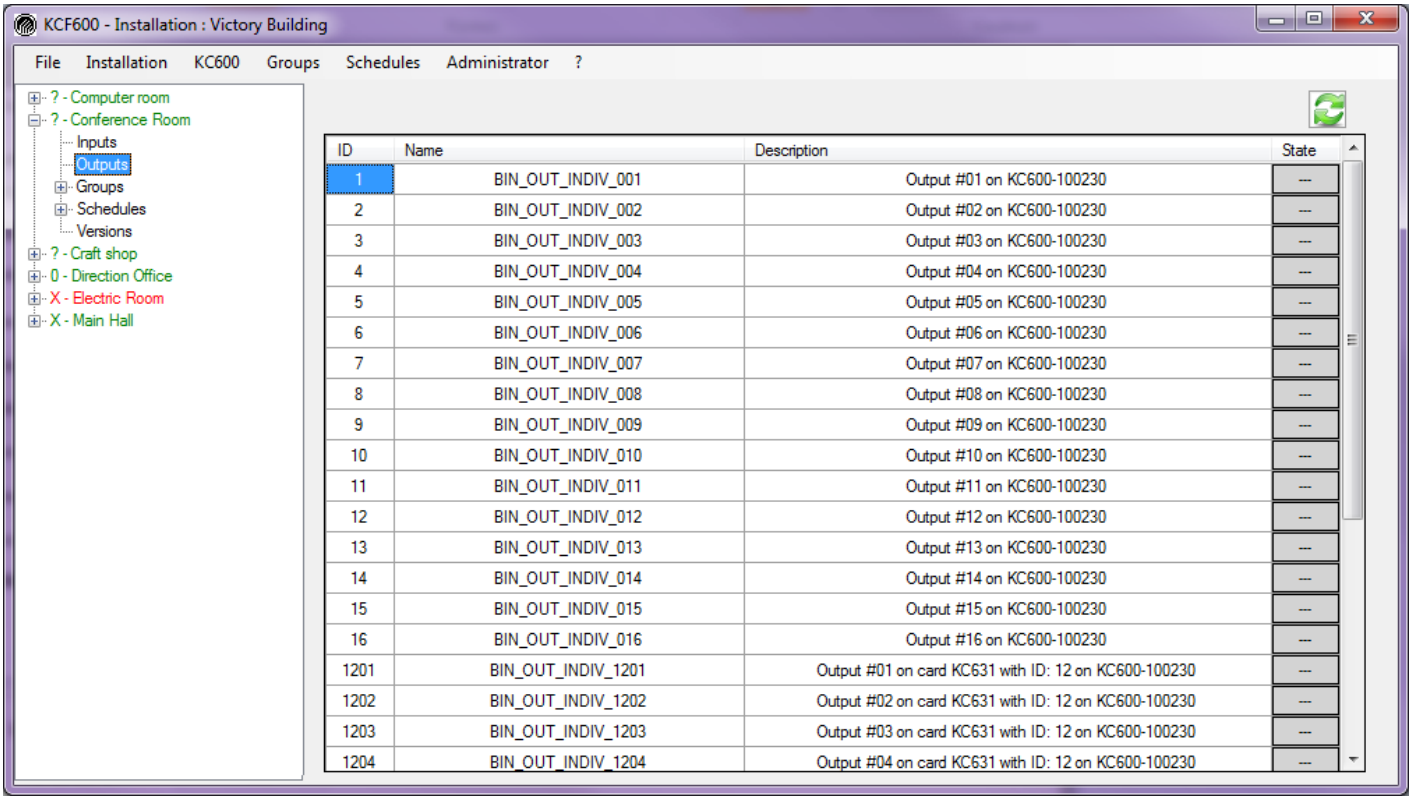


Figure 25: Window showing the list of outputs in a KC600 and their main parameters.

The window is subdivided in 3 columns of information, to identify each individual output, and 1 column for the output state:

- **Column *"ID"*:**

  It is the *object identifier* of the output as a *BACnet* "*Binary Output*" object type. This number is fixed by the *KC600* controller and cannot be edited. This number is established according to the following model:

  - The 2 least significant decimal numbers (units and ten values) indicate the position of the output on the card (*KC600* or *KC631*) where they are attached. On a *KC600* and a *KC631*, this number (between 1 and 16) is clearly indicated near the connectors to plug the outputs.
  - The most significant numbers (hundreds and thousands) are obtained by multiplying the ***"card number"*** by 100, this number corresponding to the *"Modbus"* address set on the card where the output is found:
    - The *KC600* card number is zero, so each of the 16 outputs on that card has an ID between "001" and "016".
    - A *KC631* card being controlled by a *KC600* card via its "*Modbus*" port, its card number corresponds to its communication address, value between 1 and 15 (= 0xF) set by a rotary switch (refer to section *4.2 Address Rotary Switch* for a *KC631*).

      Example: On *Figure 25*, we can see binary outputs with numbers between **1**01 and **1**05, which refers to a *KC631* card with a *Modbus* address set to: "**1**".

- **Column *"Name"*:**

  It is the name that identifies the output in the *KC600*.

  - It can be edited by clicking on the name in the list or by using a *BACnet* monitoring/control console.
  - This name must be unique between every *BACnet* object that composes the *KC600*.
  - The *KC600* limits the number of characters to 64 (excluding the *"End Of String"* character) for this property.
  - The default name given to each individual output is based on the following model: *"BIN_OUT_INDIV_XXX"* where:
    - *"BIN_OUT_INDIV_"* indicates the *BACnet* object type associated with an individual output.
    - *"XXX"* is the output identification number.
  - As soon as a user changes the name of the output, the default model is abandoned.
  - Use "*printable*" characters for the output name, which exclude graphical and control characters. The French characters can be used although some *BACnet* consoles may not recognize them…

- **Column *"Description"*:**

  This parameter is optional and may be left empty. It is used to provide more information on the nature or the area lighted by the output.

  - It can be edited by clicking on the name in the list or by using a *BACnet* monitoring/control console.
  - The number of characters is limited to 255 for this property (excluding the *"End Of String"* character).
  - The default description given to each individual output is based on the following model:
    - for a *KC600*: *"Output #XX on KC600-ZZZZZZ";*
    - for a *KC631*:

      "Output #XX on card KC631 with ID: YY on KC600-ZZZZZZ",

    where: *XX* = number of the output on the corresponding card.

    YY = "*Modbus*" address established on the *KC631* card.

    ZZZZZZ = instance number of the *KC600* controlling the *KC631* card.
  - If you choose French as the display language of the console, each description will be automatically translated. For this change of language to be reflected in the *KC600*, you need to transmit the configuration on screen to the *KC600*.
  - After entering a custom description, the model is abandoned, a change of the console display language having no more effect on the description.

- **Column *"State"*:**

  This column allows you to check the current state of each individual output/relay controlled by the *KC600* card.

  - To start the reading process, you must click on button:  just above the *"State"* column.
  - Each data box indicates the state:  or  for the corresponding output.
  - If the *KC600* card does not answer, you will receive a message to signal the problem. Then, the data in the boxes in the column *"State"* will remain the same, unchanged.
  - If a relay is defective or removed from the *KC600* card or from a *KC631* extension card, the corresponding box will indicate:  . The same will happen for all outputs attached to a *KC631* card, if this extension card is disconnected from the *KC600*.
  - Once you have read the state of all individual outputs, you may click on any *"ON/OFF"* box to set the corresponding relay to its reverse state, as long as no error was signaled for this output.

    Note that the transmitted command:
    - will be applied to the highest non-null priority (i.e. closest to priority "1") of the *"binary output"* object representing the target relay;
    - will deactivate all running options for this output (refer to section: *2.5.2.5 KC600 group "options"*)

2.5.2.4 *KC600* group configuration parameters

On the left part of the main window, if you click on a line with text: "*Groups*", the right part of the window will be empty. To visualize the groups in a *KC600*, you must:

- If at least one group was created, you have a **"+"** or **"-"** sign on the left of the word "*Groups*" in the list.
- When the sign is: *"-"*, you see a list composed of the name(s) of each group created for the current *KC600*. You must click on the name of the group you want to edit or visualize to have its parameters shown in the right part of the window:

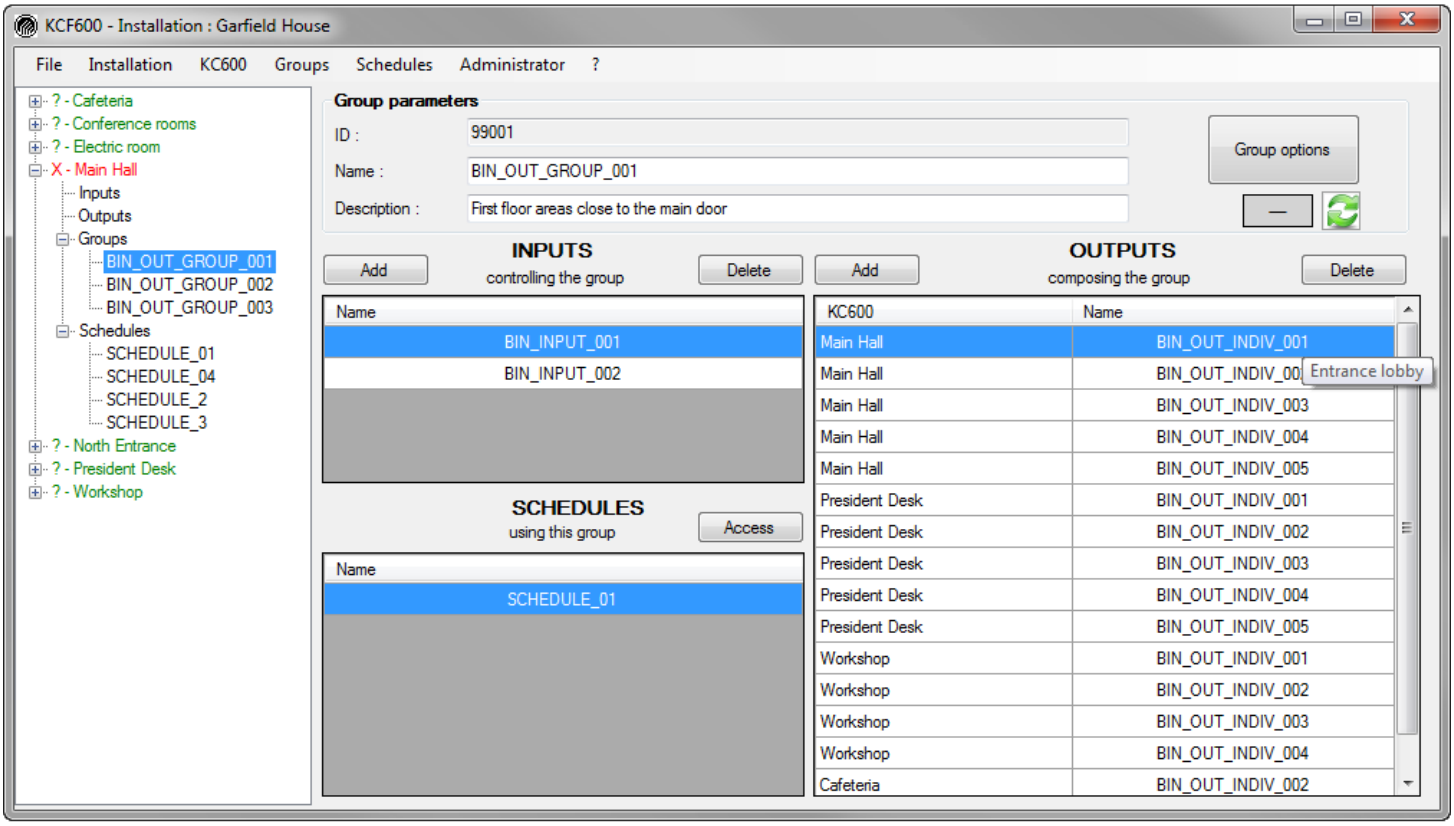


Figure 26: Window allowing the visualization of a group in a KC600 and its main parameters.

This window allows "**edition**" of all parameters defining a group. To "**add**", "**delete**" or "**copy**" of a group, refer to section *2.6 Group management in a KC600 /* "*Groups"* tab.

Notes on managing groups from any *BACnet* monitoring/control console:

- Groups are represented by *BACnet* objects of type: *"Binary Output"*, just like individual outputs.
- Their *"ON" / "OFF"* state can be changed through *BACnet Write* requests for the parameter: *"Present Value"*, with a priority between 1 and 16.
- Basic properties (*name, polarity, relinquish default + description*) for this *"Binary Output"* object can be modified from any *BACnet* console. But the identification/*instance* number is fixed and the group composition, the link between an input and a group, even the schedules that may apply to a group cannot be modified, except from the *KCF600* configuration console.

Top section: *"Group parameters"*

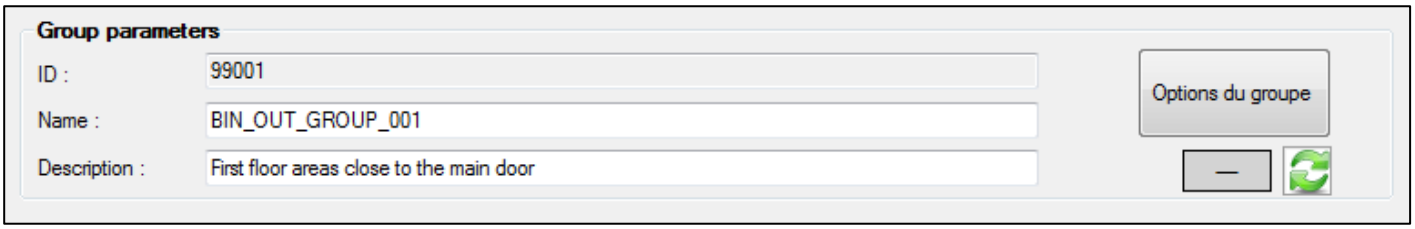


Figure 27: Window top section with the editable strings to identify the group.

- ***"ID"*:**

  This is the identification number ("*Object Identifier"*) of the group as a *BACnet* object of type: "*Binary Output*". This number is fixed by the console and cannot be changed. It is set according to the following model:

  - The 2 least significant numbers (units and ten values) indicate the number of the group, based on the smallest available number, between 1 and 80, when the group is created.
  - The most significant numbers are fixed to *"99000"*, value added to the least significant numbers to provide the resulting identification number for the group. This fixed number was chosen to distinctly differentiate *instance numbers* for *"Binary Outputs"* representing groups from the ones used for individual outputs.

- ***"Name"*:**

  It is the name that identifies the group in the *KC600*:

  - It can be edited by clicking in the corresponding text box or by using a *BACnet* monitoring/control console.
  - This name must be unique between every *BACnet* object that composes the *KC600*.
  - The *KC600* limits the number of characters to 64 (excluding the *"End Of String"* character) for this property.
  - The default name is based on the model: *"BIN_OUT_GROUP_XX"* where *"XX"* corresponds to the least significant numbers of the group identification number (value between 01 and 80).
  - Use "printable" characters for the group name, which exclude graphical and control characters. The French characters can be used although some *BACnet* consoles may not recognize them…

- ***"Description"*:**

  This parameter is optional and may be left empty. It is used to provide more information on the nature of the group or the area lighted by the outputs composing the group.

  - It can be edited by clicking on the name in the list or by using a *BACnet* monitoring/control console.
  - The number of characters is limited to 255 for this property (excluding the *"End Of String"* character).
  - The default description given to each group of outputs is based on the following model:

    "Group Nb. XX of KC600 #YYYYY",

    where: *XX* = least significant digits of the group ID (value between 01 and 80).

    YYYYY = instance number of the *KC600* where the group is defined.

- If you choose French as the display language of the console, each description will be automatically translated. For this change of language to be reflected in the *KC600*, you need to transmit the configuration on screen to the *KC600*.
- After entering a custom description, the model is abandoned, a change of the console display language or the instance number of the *KC600* having no more effect on the description.

- ***"Group options"* button:**

  This button gives access to the *"options"* window, subject of chapter: *2.5.2.5 KC600 group "options*.

- **Group *"State"* section:**

  This section appears below the *"Group options"* button:

  - You start the current group reading process by clicking on button:  .
  - The *"state"* returned is shown in the box left to the button just pressed:  or  .
  - If you don't receive an answer from the corresponding *KC600*, you will get a message to inform you that the communication could not be established with the target card. The date in the "state" box will then remain unchanged.

    Note: because all groups are defined in the *KC600*'s memory, this is the only error message that you can get after starting the reading process.
  - After reading the current state, you may click on the *"ON/OFF"* box to flip the group's state to its reverse value.

    Note that the transmitted command:
    - will be applied to the highest non-null priority (i.e. closest to priority "1") of the *"binary output"* object representing the target group;
    - will deactivate all running options for this group (refer to section: *2.5.2.5 KC600 group "options"*)

Lower middle section: *"Group composition"*

- ***"INPUTS controlling the group"*:**

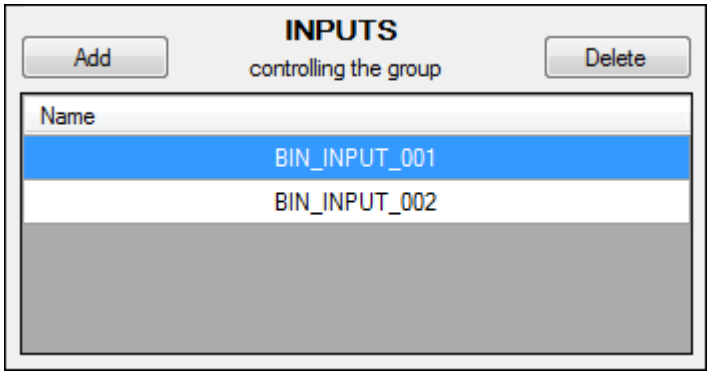


Figure 28: Window middle section: selection of inputs to control the group.

The left side list of the group parameters displays the inputs (connected to a switch, a movement detector, a light detector, etc.) that control the group. Through this link, a change of state (*"ON"* or *"OFF"*) of one of these inputs will be reflected in the priority table of the connected group.

- The list is empty if no input has been attached to the currently selected group.
- When you leave your mouse cursor over an input name in the list, the *description* of this input will appear in a *"tooltip"* (after a short delay).
- The names are listed here without any reference to the *KC600* where they are found because only local inputs of a *KC600* may control a local group on this same *KC600*.
- ***"Delete"* button**:

  To remove an input from the list, you must select it by clicking on its name, then press the button *"Delete"* at the top, right to the title for the window shown on *Figure 28*.
  - Only one input at a time can be removed from this list.
  - A message box will ask you to confirm the deletion.
- ***"Add"* button**:

  Clicking on the "*Add*" button, on top left of the input list, opens up a window allowing you to add one or more inputs to control the group:

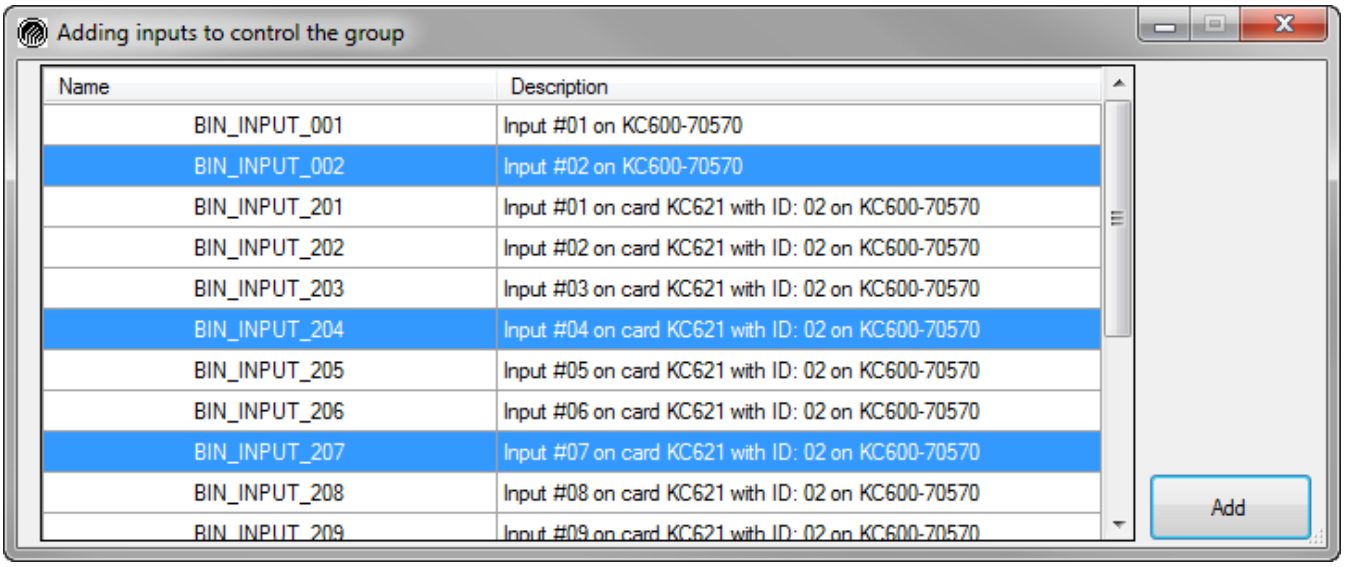


Figure 29: Window used to add one or more input to control a selected group.

- The only inputs available in the list are the ones contained in the same *KC600* as the selected group.
- Inputs that were already selected to control the current group are not displayed in the list.
- You can select one or more input at a time (refer to *Figure 29*) by holding "*Ctrl*" or "*Shift*" key (for consecutive selections) while clicking on the input names. Then press the "*Add*" button to include the selected inputs in the list of *Figure 28*.
- Once you have added all the required inputs, close the window by clicking on the button.

- ***"OUTPUTS composing the group"*:**

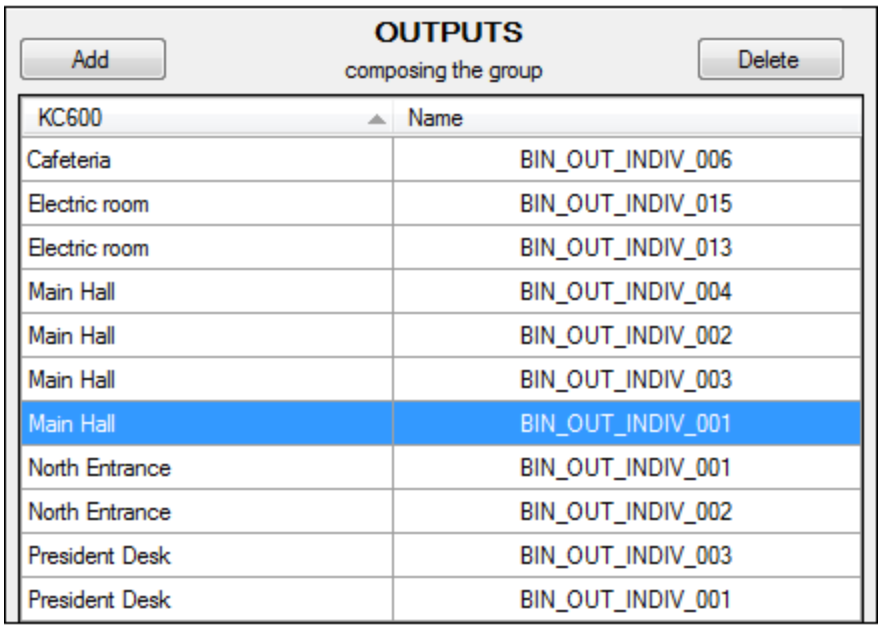


Figure 30: Window showing the outputs controlled by the group.

The right side list displays every individual outputs that are part of the selected group.

- The list is empty if no individual output was associated yet to the group.
- Unlike the inputs, you can select any individual outputs from different *KC600*; these outputs do not need to be local to the group to which they are attached. This is why there are 2 fields displayed in the *"Outputs"* list:
  - ***"KC600":*** this column indicates the name of the *KC600* to which the output is attached.
  - ***"Name":*** this column shows the name of the individual outputs selected to compose the group.
  - If you leave your mouse on top of the name of an output or a *KC600*, their description will be displayed in a *"tooltip"*.
  - If there are outputs names showed with a (yellow) colored background, there are 2 possible causes:
    - the *KC600*, where the corresponding outputs are found, was eliminated from the current *"installation"* list;
    - the *KC631*, on which were connected the unavailable outputs, is not communicating anymore with the *KC600* card or the *KC631 Modbus* address was changed (and each output instance number is based on their *KC631 Modbus* address, so they are not recognized anymore; refer to section *2.5.2.3 KC600 outputs configuration parameters*).

  Example: after deleting the *KC600* named *"Main Hall"* from the current *"installation"*, the group outputs shown on *Figure 30* would look instead like the representation on *Figure 31*:

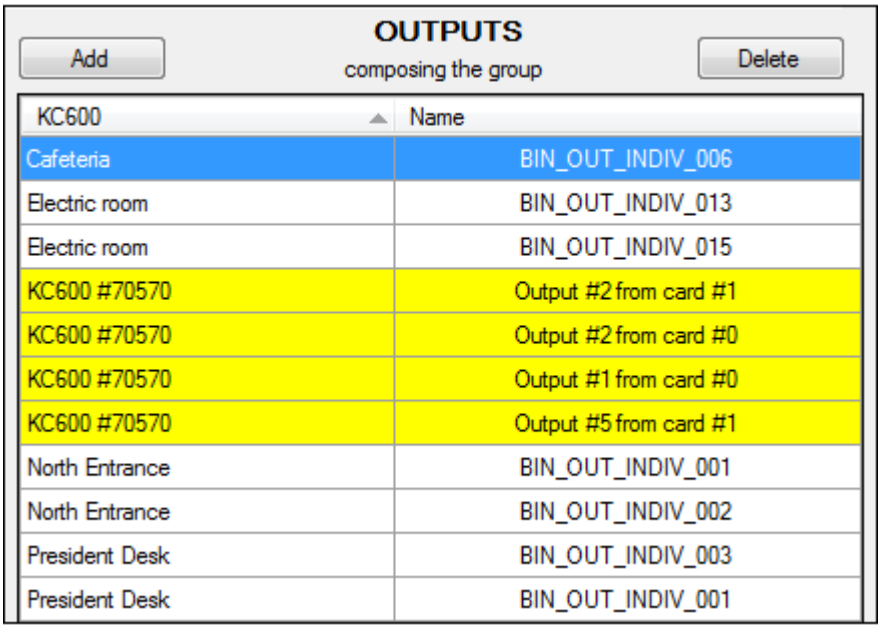


Figure 31: Detection of invalid outputs in the group composition.

- The name that was identifying the eliminated *KC600* was replaced by the missing controller type (*KC600*) and what was its *instance number* (70570).
- The name that was identifying the absent output, on the eliminated *KC600*, was replaced by:
  - the output **connection number** on the controller card (*KC600* or *KC631*), corresponding to the least significant digits of the output *instance number*, which is a value between 1 and 16,
  - followed by the *"Modbus"* address of this controller card (*"0"* for a *KC600*, a value between *"1"* and "*15"* for a *KC631*); refer to the *ID* definition in section *2.5.2.3 KC600 outputs configuration parameters*.
- On the example of *Figure 31*, we can see that all unavailable outputs come from the same *KC600 #70570*:
  - we have outputs 1 and 2 that were connected directly on the *KC600* card, which tells us that this controller is not anymore part of the current *"installation"*;
  - we have outputs 5 and 11 that were connected to a *KC631* card at *Modbus* address *"1"* to communicate with the same *KC600*.

  If only the outputs on the *KC631* at *Modbus* address *"1"* had been colored with a yellow background, then it would indicate that last time the *KCF600* console requested to *"Receive a configuration"* from this *KC600*, the outputs associated to the *KC631* at *Modbus* address *#1* were absent. But as the *KC600* answered successfully to the request, it means that it is still part of the *installation*. 2 possible causes for the elimination of these outputs:
  - the *KC631* card was unplugged (disconnected or powered off), so it stopped communicating with the related *KC600*;
  - the *Modbus* address was changed on the *KC631* card. As the outputs are identified by an *instance number* based on this address, they are not recognized anymore by their former reference number; refer to section *2.5.2.3 KC600 outputs configuration parameters*.

- ***"Delete"* button**:

  To remove an output from the list, you must select it by clicking on its name, then press the button *"Delete"* at the top, right to the title for the window shown on *Figure 30*.

  - Only one output at a time can be removed from this list.
  - A message box will ask you to confirm the deletion.

- ***"Add"* button**:

  Clicking on the "*Add*" button, on top left of the output list, opens up a window allowing you to add one or more outputs to be part of the group:

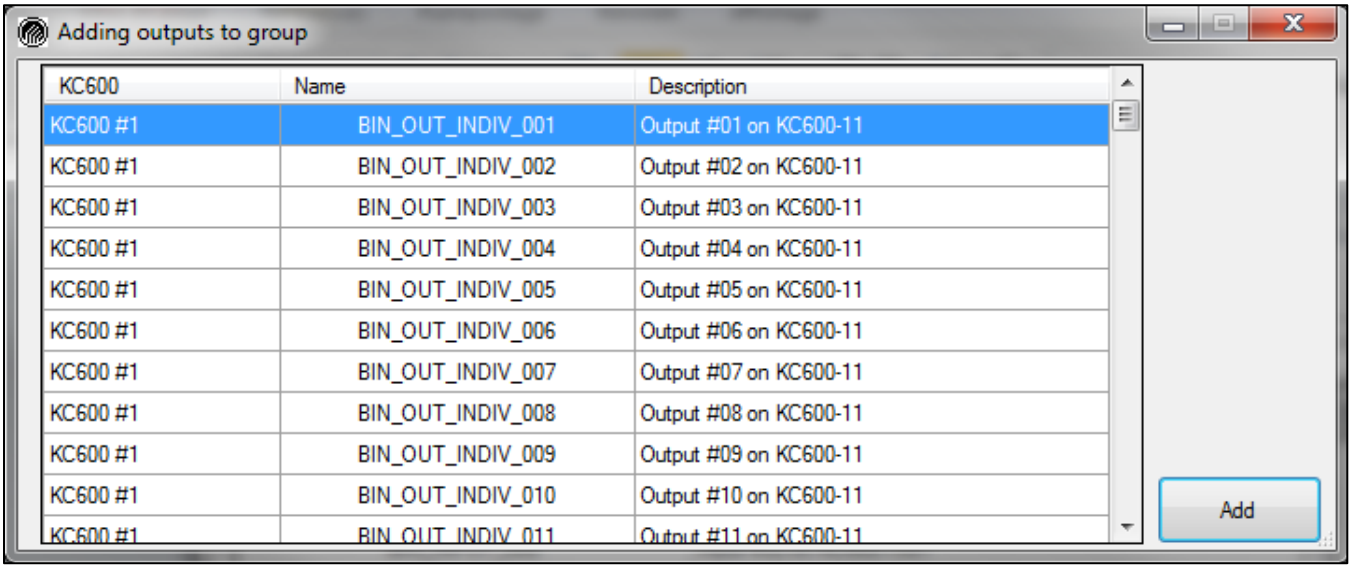


Figure 32: Window used to add one or more output controlled by the selected group.

  - Every output of every *KC600* defined in the current *"installation"* is displayed in that list, as long as they are not already selected in the composition of the group.
  - When this window is opened, the outputs are sorted according to the alphabetical order of the names in the *KC600* column, order that can be changed by clicking on a column title.
  - You can select one or more output at a time by holding "*Ctrl*" or "*Shift*" key (for consecutive selections) while clicking on the output names. Then press the "*Add*" button to include the selected outputs in the list of *Figure 30*.
  - Once you have added all the required outputs, close the window by clicking on the button.

- ***"SCHEDULES applied to this group"*:**

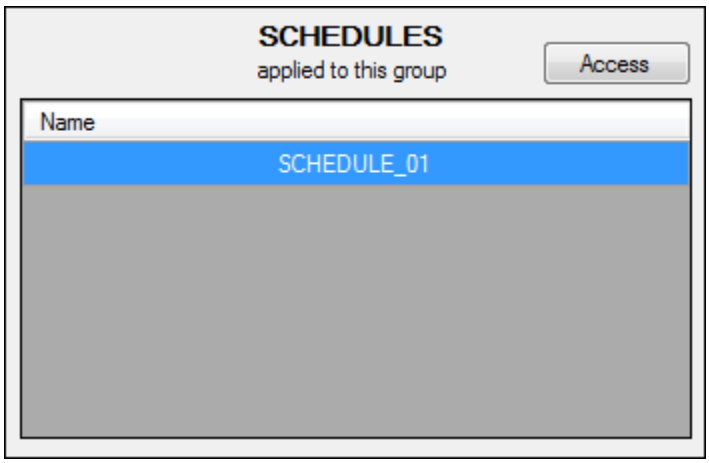


Figure 33: Left side section displaying the schedules applied to the selected group.

The bottom left side list of the *"group composition"*, in the main window (see *Figure 26*), presents the names of the schedules already created, generating events that apply to the current group.

- The list is empty if no schedule was created for the current group.
- When you leave your mouse cursor over a schedule in the list, the description of this schedule will appear in a *"tooltip"* (after a short delay).
- **"*Access*" button:**

  To see the parameters of a schedule in the list of *Figure 33*, you must select one by clicking on its name, then press on the "*Access*" button:

  - this action will switch the main display from the *"Group"* view to the *"Schedule"* view (as shown on *Figure 38*);
  - the current selection in the left part of the window will be the *"schedule name"* selected, under the label *"Schedule"* instead of label *"Group"* (as on *Figure 26*).

2.5.2.5 *KC600* group *"options"*

The addition of *"options"* was required to fulfill the needs of our clients concerning lighting management, some of these options being regularly mentioned in submissions.

- These *"options"* are associated with groups' definitions.
- They apply to the individual outputs composing the group, when the group receives a request to set its state to *"ON", "OFF"* or *"Relinquish"*, no matter the priority passed with the command. The origin of the request is important to determine how the options will be applied (refer to the next page on the *Options configuration window*).

List of the 4 available *"options"*

- **OFF Only:** When this *option* is activated, the requests to turn *"ON"* the individual outputs (composing the group) are ignored. When the *"OFF Only" option* is activated, the *"ON Only" option* is not available.

- **ON Only:** When this *option* is activated, the requests to turn *"OFF"* the individual outputs (composing the group) are ignored. When the *"ON Only" option* is activated, the *"OFF Only" option* is not available.

- **TOE:** *"**T**ime **O**n **E**xtension":* when this *option* is applied to the individual outputs part of the group, each time one of those outputs is turned on, a timer is started to automatically turn off the output after a certain period of time (configurable in the group *options* window of *Figure 34*, the default value being 120 minutes).

- **Warning:** *Warning* period to signal the imminent closing of the lights: with this *option*, when an individual output that is turned *"ON"* receives an *"OFF"* request, except when it comes from an input (that is usually a wall switch), the output will go *"OFF"* briefly before coming back *"ON"* for a short period of time (configurable in the group *options* window of *Figure 34*). After this short *"courtesy"* delay, the individual output will be turned *"OFF"* for good.

  This "*Warning"* period is often associated to the "*TOE"* period to signal that the lights are about to be turned off, which gives time for someone in the room to reach the wall switch for turning it *"ON"* to restart the *"TOE"* cycle.

Options configuration window

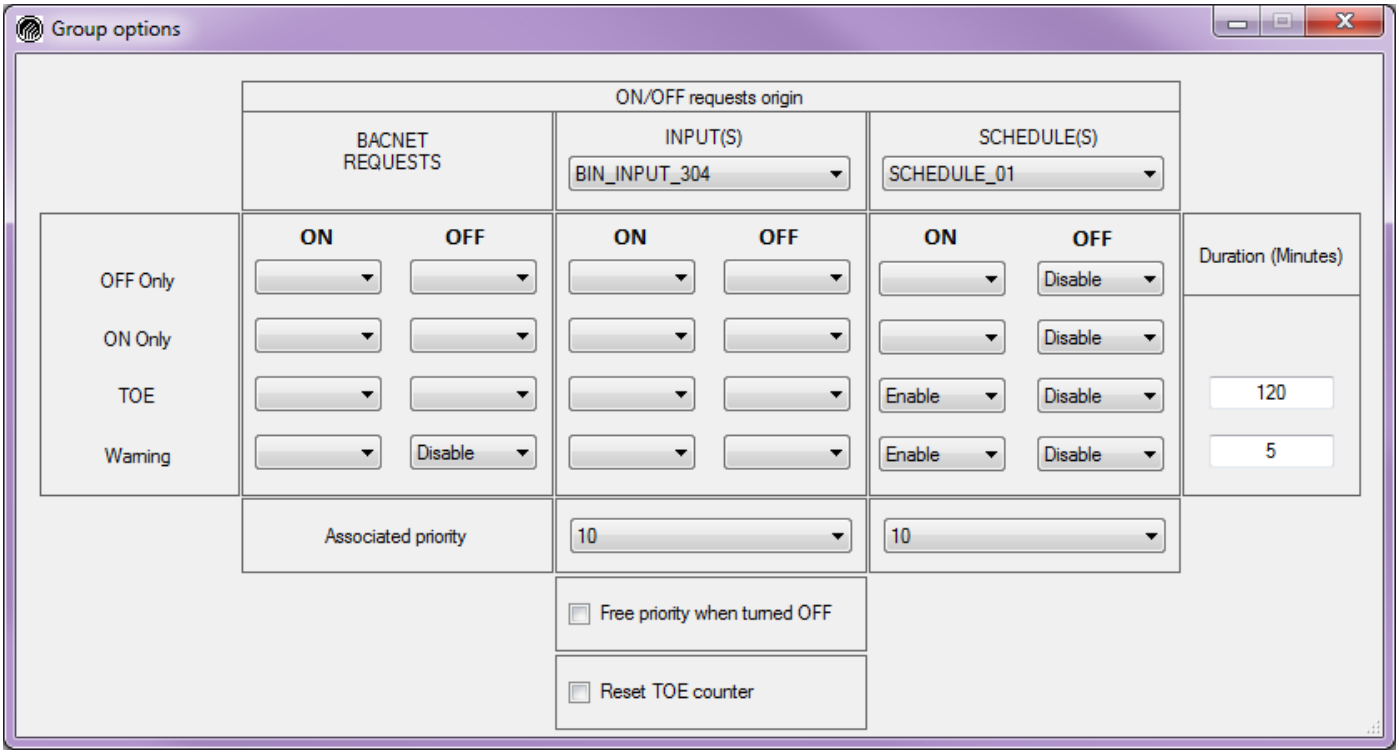


Figure 34: Window to establish the options of a group.

1. ***Options* window subdivisions:**

Prerequisite notions: group <u>priority</u> table

- Each group is represented in the *BACnet* environment by an object of type *"Binary Output"*. This object type implicates a 16 priorities array (refer to *Annex 2. Priority tables for binary outputs*), each priority being initialised to *"NULL"*.

- Every *BACnet* request, transmitted to change the *"ON/OFF"* state of a group, comes with a priority (value between 1 and 16 inclusively) and one out of 3 possible state commands:
  - set passed priority to *"ON";*
  - set passed priority to *"OFF";*
  - free (*"Relinquish"*, set to *"NULL")* this priority.

- Following such commands, the highest (closest to *"1"*) non null priority (set to *"ON"* or *"OFF"*), in the object priority table, becomes the ***"effective priority"***, its current state being applied to the group.

Application of the group *"ON / OFF"* state to the individual outputs:

- The result of an *"ON / OFF* or *Relinquish"* request, addressed to a group, is relayed to the individual outputs (part of this group) as an *"ON"* or *"OFF"* command only if the original request affects the group ***"effective priority"***, which happens when the request priority:
  - has the same priority as the current *"effective priority"* for the group,
  - or if the priority received provokes a new *"effective priority"* for the group.
- Such a command, retransmitted to the individual outputs, does not really have a priority; it is applied by changing the *"effective priority"* of each output, no matter which priority is effective.
  - If an individual output was not having any *"effective priority"*, the *"ON / OFF"* state transferred will be applied to priority 16.
  - The request may be rejected at the individual output level, according to the current *options* enabled with the output. For **example**, if an output has the *option "ON Only"* enabled, a request to set *it "OFF"* will be rejected (if the output *"effective priority"* is lower than 6; see further text).

Application of the *options* to the group and its individual outputs:

- A set of *"options"* is relayed to the individual outputs (part of this group) only if the group receives an *"ON / OFF* or *Relinquish"* request that affects its *"effective priority"*, as described above. In other words, an individual output will receive a set of *"options"* only when the group needs also to transmit an *"ON"* or *"OFF"* request to them.
- The pair: *"state"* (*"ON"* or *"OFF"*) and *"options"* is transmitted according to the following sequence:
  - first, the group will send the *options* applicable to the outputs,
  - then it will transmit the *"ON / OFF"* request to these outputs, which will apply or not the passed state, according to the *options* just received (if the *"OFF Only"* or the *"ON Only"* is enabled) and the *"effective priority"* (according to certain rules described further).

***"ON / OFF"* pairs of 4 *options* = *"Options application mode"*:**

- According to the origin of the request, if the group *"effective priority"* is touched by this request, the resulting *"ON"* or *"OFF"* state (for the group) will determine which of the 2 *"pools of options"* is sent to the individual outputs.
- Each *"pool of options"* consists in a pair of 4 *options*:
  - the 1st block of 4 *options* indicates how to apply these *options* to the individual outputs when they receive an *"ON"* command from the group;
  - the 2nd block of 4 *options* indicates how to apply these *options* when that same request is for setting them *"OFF"*.
- Only the block of 4 *options* that corresponds to the state retransmitted to the individual outputs will be attached to the command issued by the group.

Origin of the request to change the group state:

- 3 possible sources:
  - a ***BACnet* request**, transmitted by a console to manage a *BACnet* network or by another *BACnet* controller;
  - an **input** change of state (usually from a wall switch) connected to the group through the *KCF600* configuration console;
  - a **schedule** that requires the application of an *"event"*, programmed in the *KCF600* configuration console to change the group state at a certain time (and date).

Options window subdivisions:

Columns:

- **3 main columns** according to the origin of the request to change the group state:
  - BACNET REQUESTS.
  - INPUT(S):
    - A combo box allows you to select for which input, that controls the current group, we want to view or edit the *"options application mode"*.
    - This list is empty if no input was connected to the current group.
  - SCHEDULE(S): refer to the INPUTS.

- Under each main column, **2 sub-columns** according to the group final state (*"ON"* or *"OFF"*) after receiving a request *("ON" / "OFF"* or *"Relinquish"*) affecting the group *"effective priority"*:
  - **ON:** in the corresponding box, we set how the *option* will be applied (3 possible states explained further) to the group's individual outputs, when the group **resulting state** is *"ON"* , no matter if it results in a new state or not for the group.
  - **OFF:** same principle when the group **resulting state** is *"OFF"* following a request.
- **Column to the extreme right: *Duration*** (in *Minutes*) for the 2 *options* working with a countdown: *"Time On Extension"* and *"Warning"*.

Lines:

- **4 main lines** to define how to apply each one of the 4 *options* transferable to each individual output part of the group:
  - *"OFF Only"*
  - *"ON Only"*
  - *"Time ON Extension"*
  - *"Warning"*

- **Bottom line - *Associated priority*:**
  - For ***BACnet* requests:**
    - No priority needs to be configured in this column because each *BACnet* request, to change the state of a *"Binary Output"* object, comes with a priority between 1 and 16.
  - For the ***Input(s)*:**

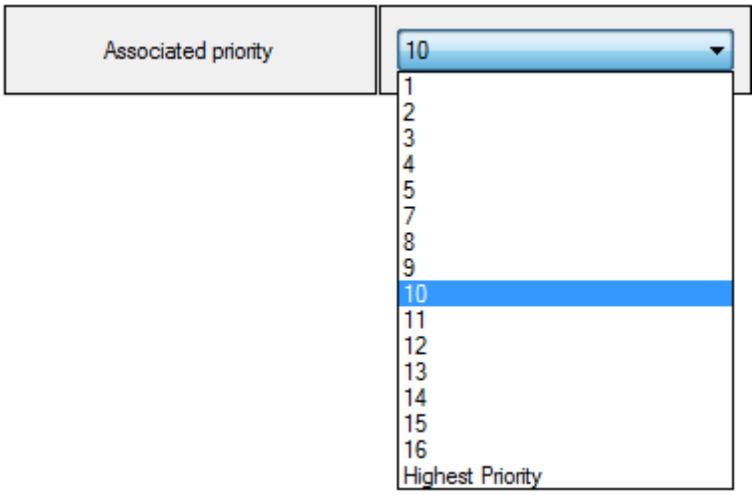


Figure 35: List of available priorities to transmit to the group after an input change of state.

- This priority, set from the list of *Figure 35*, is only associated with the input shown in the selection box and not to all inputs (if there are many) linked to control the group:

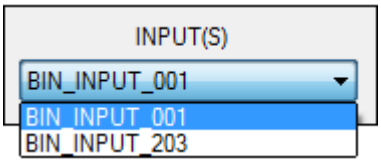
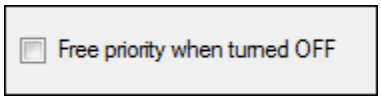


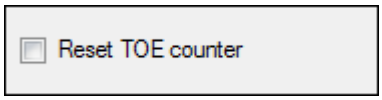
Figure 36: Selection of a single input already linked to control the group.

- **Highest Priority:** in addition to the usual 16 priority levels, a 17th priority is available in the list shown on *Figure 35*. It forces the group state to follow the one set at the input, no matter what is the current *effective priority* for the group.
  - The input state will then be automatically transferred to the *effective value* in the group table of priorities. If there were no *"effective priority"*, priority 16 will be used.
  - Though this mechanism is not compliant with the *BACnet* standard, this behaviour is expected by a client who has absolutely no knowledge about how *BACnet* works, especially regarding the priority tables…

- 

  If the priority selected is between 1 and 16, then we can check this supplemental option. For example:

  - If you set the priority associated with the input to 10, when the input is set to *"OFF"*, a command will be sent to the group to *"Relinquish"* priority 10 in the group priority table.
  - If this release of priority 10 causes a new *effective priority* for the group, the state *"ON"* or *"OFF"* for this new priority will be transmitted to each individual outputs part of this group.

- 

  Though this feature is not associated as such to the selected priority, it is mainly used for an input used for a movement detector. This feature will be effective only if the *"Time On Extension"* option is enabled:

  - **If the box is checked:** the *"TOE"* countdown will start when the input signal, which was *"ON"*, is switched to *"OFF"*. If the countdown was running, each *"ON"* request will restart the counter, which then keeps its initial value as long as the input signal is *"ON"*.
  - **If the box is unchecked:** the *"TOE"* countdown will start when the input signal is turned *"ON"* while there was no countdown running. All subsequent *"ON"* requests will be ignored as long as the countdown is not completed. This means that once you have turned on the lighting, it will turn off automatically after a fixed period of time, which is the *"TOE"* period set in the *KCF600* console.

- For the **Schedule(s):**
  - The priority, set from the list of *Figure 35*, is only associated with the schedule shown in the selection box and not to all schedules (if there are many) linked to control the group:
  - However, do not forget that each schedule is usually composed of many events, each event:
    - happening at a certain date or during a series of dates (or a series of week days), once or repeatedly (in a cyclic manner);
    - to turn *"ON"* or *"OFF"* a group of outputs, at a precise time or at different moments of the day.
  - The priority set here will apply to this schedule for every occurrence of an event defined with this schedule.
  - **Highest Priority:** refer to the corresponding priority for the *"Input(s)"* above.

2. Possible states to apply the *Options*:

3 possible *"configuration"* states to apply the *options*:

For each one of the 4 *options* associated with the state: *"ON"* or *"OFF"*, we may set, in the *options* window (refer to *Figure 34*), one of the 3 possible *"screen"* states for an *option*:

- Disable : to deactivate the *option,* to make sure it will not be applied by the individual outputs part of the group.
- Enable : to activate the *option,* so it will be applied by the individual outputs part of the group.
- / *"Keep"*: this state indicates to the individual outputs to *" Keep"* the way the option was applied (*"Enable"* or *"Disable"*) prior to this call.
    - The *"option"* state: *"Keep"* is only stored at the group level.
    - When the *options* are transferred from the group to each individual output, *"Keep"* indicates to an output to preserve the currently applied *"Enable"* or *"Disable"* state for the option (the initial state being: *"Disable"*).

***BACnet* properties to represent the 4 *options*:**

- 4 "*proprietary*" properties were added to the *BACnet* object: "*Binary Output*", object used to represent all groups and all individual outputs in the *KC600,* to reflect the current state of each one of these 4 *options*:
    - Property #900 = *"OFF_Only" option.*
    - Property #901 = *"ON_Only" option.*
    - Property #902 = *"TOE" option.*
    - Property #903 = *"Warning" option.*
- However, very few *BACnet* consoles allow you to visualize such "*proprietary*" properties.

4 possible states for the properties that represent the *options*:

- **Property values for group *options*:**
  - According to the source of the last request received, a pair of 4 *options* is attached to the group.
  - Depending on the *"ON"* or *"OFF"* group state resulting of the last request received (to *"ON"*, *"OFF"* or *"NULL"* / *"Relinquish"*) that affected the *"effective priority"*:
    - the values for the *options* properties, shown by the group, will be the ones of the 4 *options* (from the pair) associated with the group actual *"ON"* or the *"OFF"* state;
    - these *"4 effective options"* for the group will be then transferred to each output (composing the group) to be applied at once.
  - Possible values for each *option* property for a group:
    - **0 :** the *option* has to be *"Disabled"*.
    - **1 :** the *option* has to be *"Enabled"*.
    - **2 :** the individual outputs must *"Keep"* the *option* state currently applied (*"Enabled"* or *"Disabled"*).
    - **3 :** *impossible value for a group option property…*
- **Property values for individual output *options*:**
  - According to the last *option* transferred to the output by the group, the *options* for an output will be either *"Enabled"* or *"Disabled"*, so there are only ***2 configuration states*** possible for these *options*.
  - However, a third specific state is available here, that takes into account if the ***option* is applied** or not, due to an *"effective priority"* (in the output table) with a value between 1 and 6 (included). In that case, all *"effective options"* will be bypassed, suspended. As soon as a command is received (by the output) with a priority between 7 and 16, the current *options* will be applied again.
  - Possible values for each *option* property for an individual output:
    - **0 :** the *option* is *"Disabled"*.
    - **1 :** the *option* is *"Enabled"*.
    - **2 :** *impossible value for an output option property…*
    - **3 :** the *option* is ignored due to a high level (emergency) priority (between 1 and 6).

3. Practical example on how the *options* are applied:

From the *"options"* window shown on *Figure 34*, we will only use 2 options, with the settings as shown below:

	BACNET REQUESTS		INPUT(S)		SCHEDULE(S)		
	ON	**OFF**	**ON**	**OFF**	**ON**	**OFF**	**Duration (Minutes)**
TOE					Enable	Disable	120
Warning		Disable			Enable	Disable	5
	Associated priority		10		10		
			☐ Free priority when…				

Figure 37: Table summarizing the main options used for our example.

Consider that the implied group has an empty priority table (all 16 priorities set to *"NULL"*).

- **Schedule *"ON"*:**
  - If we only have one schedule: "*SCHEDULE_01"* and it sends a request to set the group *"ON"* (at the end of the day), then the *"ON"* state will be associated with the group priority *"10"*.
  - As this priority is the *"effective"* one, both options for the *"TOE"* and the *"Warning"* will be *"Enabled"* for each individual output part of the group.
  - The request state will then be retransmitted to the individual outputs to turn them *"ON"* (no matter their current *"effective priority"*). These outputs will remain turned *"ON"* for 120 minutes, which is the duration set for the *"TOE option"* in the rightmost column in the table of *Figure 37*.
  - At the end of this period, all the group outputs will be turned off automatically, for about 1 second, then they will be back *"ON"* by themselves for the duration set for the *"Warning option"*, which is 5 minutes in the table of *Figure 37*.
  - After this delay of 5 minutes, the outputs will turn themselves *"OFF"* and they will stay *"OFF"* until they receive a new command…
- **Switch turned *"ON"*:**
  - If we turn *"ON"* a wall switch, linked to the only input that controls our group, a request will be sent to the group to set it *"ON"* (with priority *"10"*, which is still the group *"effective priority"*).
  - This request targeting the *"effective priority"*, it will be retransmitted to the group outputs, indicating them to *"Keep"* the current way to apply the options (*TOE* and *Warning*).
  - If neither the group itself, nor the group outputs receive a request to change their *"ON"* or *"OFF"* state, as both options *"TOE"* and *"Warning"* are still enabled, we will find again the same behaviour as for the previous scenario, for the *"Schedule ON"*, with periods of 120 and 5 minutes.

- ***"ON / OFF" BACnet* request to an INDIVIDUAL OUTPUT**:

  NOTE: Direct requests to the outputs do not have any effect on the *"options"* configuration (a higher priority request bypassing the *"options"* without changing them). Only the group requests may alter the options applied to the outputs.

  - If during the countdown period of 120 minutes for the *"TOE" option*, a *BACnet* request is transmitted directly to an individual output (part of the group) to:
    - … turn it *"OFF"*: the output will turn itself *"ON"* again for a 5 minutes period to *"warn"* the people in the corresponding room that the lights are about to be turned *"OFF"*, which gives them the time to reach the switch to turn *"ON"* the lights for another *"extension"* period of 120 minutes;
    - … turn it *"ON"*: the output will stay *"ON"* but the *"TOE"* countdown (for this output) will be reset to 120 minutes before the lights go *"OFF"* briefly, to initiate the short *"Warning"* period. This means that this output will go *"OFF"* long after the other ones part of the same group.

- **Switch turned *"OFF"***:
  - If we turn *"OFF"* the same wall switch, while the 120 minutes *"TOE"* countdown is running, a request (of priority *"10"*) will be sent to the group, commanding the *options* to keep the way they are currently applied.
  - Because the *"OFF"* request comes from an *"input"*, the *"Warning"* period is then always bypassed (to prevent the absurdity of seeing the lights turned *"ON"* for a 5 minutes period as we just turned *"OFF"* the switch…)

- ***"ON / OFF" BACnet* request to the GROUP**:
  - If we send a *BACnet* request to set the group *"OFF"* (with priority *"10"*) during the 120 minutes *"TOE"* countdown, the lights will be turned *"OFF"* without any *"Warning"* because we set in the configuration table of *Figure 37* that *"Warning option"* must be disabled for such as request.

- **Switch turned *"ON"***:
  - If we turn *"ON"* the same wall switch, every group output will be turned *"ON"* for the 120 minutes *"TOE"* period. At the end of this countdown, the lights will go *"OFF"* without a *"Warning"* period (because this option was *"disabled"* by the previous *BACnet "OFF"* command and for an input, the *"options"* table was set to *"keep"* the current application for the *"Warning"*).

- **Schedule *"OFF"*:**
  - Any event of "*SCHEDULE_01"* will disable all options (shown in the table) for the individual outputs part of the target group.

  NOTE: In everyday practice, for rooms without windows, it is better to create another schedule to control the same group, in such a way that:

  - The first schedule turns *"ON"* (or keeps *"ON"*) the group outputs, at the beginning of the evening, by enabling the options so the lights will go *"OFF"*, for the rest of the night, at the end of the *"TOE"* countdown.

- The second schedule turns *"ON"* the same outputs in the morning, disabling all options, to keep the lights *"ON"* all day long (except when they are turned *"OFF"* from the wall switch or from a *BACnet* command).

Warning regarding the compliance of *options* in the *BACnet* environment

- The notion of *"Options"* was added by *Gentec* to answer to the needs of their customers in lighting management. This *"proprietary"* notion is therefore not described, neither considered, in *ASHREA* standard that defines how *BACnet* devices must communicate with each other in a network.
- As the **activated *options*** must interact with *BACnet* requests to change the state of *"Binary Output"* objects, their application is often conflicting with the rules established by *ASHREA* standard for this type of object…
- Some examples of non-conformity:
  - If the "*ON_Only*" *option* is activated for an output with an *"effective priority"* of 10, it will not answer to an "*OFF*" request of priority 8. The same goes for an "*ON*" request when the output "*OFF_Only*" *option* is set.
  - With the "*TOE*" *option* enabled, an output turned "*ON*" will automatically be shut off at the end of the *"TOE"* duration, all by itself, not in response to any *BACnet* request to turn it *"OFF"*.
  - Take an individual output turned "*ON*", with an *"effective priority"* of 10 and the "*Warning*" *option* activated. Send to this output a request to turn it "*OFF*", with a priority level 8. Though the output should be viewed as *"OFF"*, due to the activation of the *"Warning"* period, the output will still appear as being *"ON"* in a *BACnet* monitoring/control console, for the duration of the "*Warning*" countdown…

2.5.2.6 *KC600* schedules configuration parameters

On the left part of the main window, if you click on a line with text: "*Schedules*", the right part of the window will be empty. To visualize the schedules in a *KC600*, you must:

- If at least one schedule was created, you have a **"+"** or **"-"** sign on the left of the word "*Schedules*" in the list.
- When the sign is: *"-"*, you see a list composed of the name(s) of each schedule created for the current *KC600*. You must click on the name of the schedule you want to edit or visualize to have its parameters shown in the right part of the window:

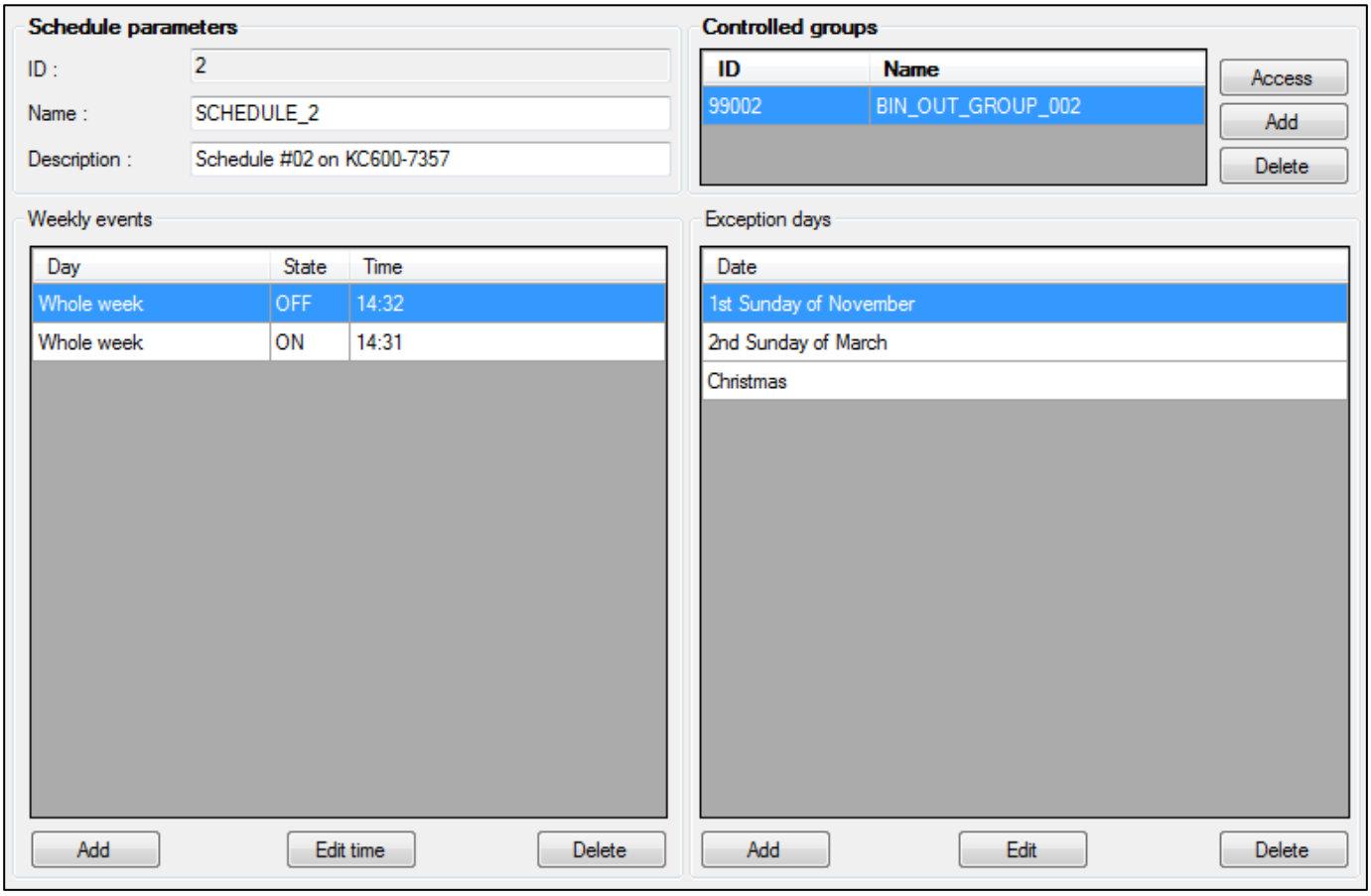


Figure 38: Right frame of main window, showing the parameters of a schedule in a KC600.

This window allows "**edition**" of all parameters defining a schedule, including the addition/edition of events to change the state of one or more group(s) for some date(s), at certain time(s). To "**add**" or "**delete**" a schedule, or even or manage the list of "*special days*", refer to section *2.7 Schedule management in a KC600 / "Schedules*.

Note on the way to manage schedules with a console to control a *BACnet* network:

- Schedules are represented by a *BACnet* object of type*: "Schedule"*, for which you cannot change the "*ON*" / "*OFF*" state with a *BACnet* write request (for the "*Present value*" parameter), except in "*Out Of Service*" mode.
- The name and description of a "*Schedule*" object can be edited, but not its identification number. Also, even though they are available, the main parameters defining the schedule events can only be modified using the *KCF600* configuration console.

Upper left section: "Schedule parameters"

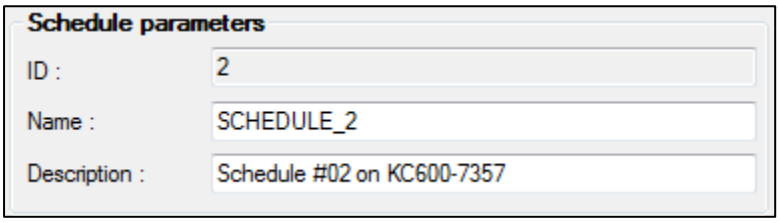


Figure 39: Parameters identifying a schedule in a BACnet network.

- ***"ID"*:**

  It is the identification number *("Object Identifier")* of our schedule, in its representation as a *BACnet* object of type: "*Schedule*". This number is set by the *KCF600* configuration console, according to the lowest available number (not already used by another schedule in the same *KC600*) and it cannot be modified.

- ***"Name"*:**

  It is the name identifying the schedule in the *KC600*.

  - You can edit it from the window by clicking on the corresponding field. You can also change the name of the corresponding *"Schedule"* object from a *BACnet* monitoring/control console.
  - This name must be unique between every *BACnet* object that composes the *KC600*.
  - The *KC600* limits the number of character to 64 for this property.
  - The default model for a schedule name is: *"SCHEDULE_XX"*, where *"XX"* is a value between 01 and 24.
  - Use "printable" characters for the schedule name, which exclude graphical and control characters. The French characters can be used although some *BACnet* consoles may not recognize them…

- ***"Description"*:**

  This optional string may be left empty. It is used to provide more information on the nature of the schedule, to describe what it is used for and for which particular dates it was created.

  - It can be edited by clicking on the description field in the *KCF600* configuration console or from a *BACnet* control console.
  - The number of characters is limited to 255 for this property (excluding the *"End Of String"* character).
  - The default, a default string is set according to the language selected for the console display, following the model:

    "Schedule #XX on KC600 #YYYYY",

    where: *XX* = schedule identification number (value between 01 and 24).
    YYYYY = instance number of the *KC600* where the schedule is found.
  - If you choose French as the display language of the console, each description following the model will be automatically translated. For this change of language to be reflected in the *KC600*, you need to transmit the configuration on screen to the *KC600.*
  - After entering a custom description, the model is abandoned, a change of the console display language or the instance number of the *KC600* having no more effect on the description.

Upper right section: *"Controlled groups"*

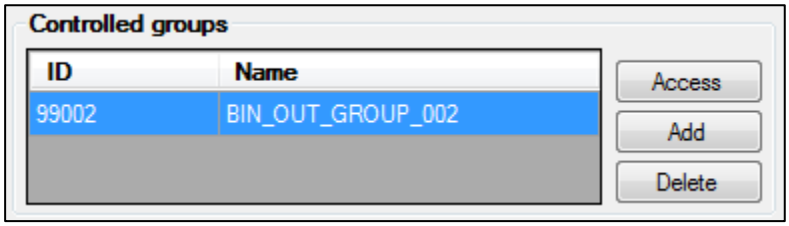


Figure 40: List of groups controlled by the selected schedule.

Initially empty, this list indicates the group(s) that will be affected by every event set for the selected schedule. Those events are defined in the two sections in the lower part of the *KCF600* main window.

- ***"Group identification"* columns:**
  - ***"ID":*** is the identification number ("*Object Identifier"*) of the group controlled by the schedule as a *BACnet* object of type: "*Binary Output*".
  - ***"Name":*** is the name identifying the group, unique between all *BACnet* objects found on the same *KC600* where the selected schedule resides.
  - When you leave your mouse cursor over a group in the list, the *description* of this group will appear in a *"tooltip"* (after a short delay).

- ***Group "Access" button*:**

  By clicking on this button, the main window will display the selected group parameters instead of the ones for a schedule. The left frame of the main window will also reflect this new selection, in the list of the objects for the same *KC600* (refer to *Figure 26*).

- ***Group "Add" button*:**

  Clicking on that button gives you access to a new window, which lists all groups:
  - that are defined in the *KC600* where is found the currently selected schedule;
  - that are not selected yet to be controlled by that schedule.

- ***Group "Delete" button*:**

  Click on this button to remove a group selected from the list shown on *Figure 40*. A message will ask you to confirm or cancel the deletion request.

Lower section: *Events composing a schedule*

A schedule is composed of at least one event, each event consisting in:

- an action to turn on/off one or many groups of outputs (selected during the previous step),
- at a precise moment in a year or for many dates,
- at a precise time or referring to the sunset or sunrise time,
- on a cyclic occurrence or only once.

The lower part of the window is subdivided according to 2 types of events:

- the ones happening regularly, on a *weekly* basis, during a *normal week*, consisting of ***"Weekly events"***;
- the ones happening on particular dates, on an *annual* basis or *only one time*, consisting of *"**Exception days"*** that override the regular cycle of the "*Normal week"* events. These *exception days* are usually holidays, celebration days, some being specific to certain countries, regions, enterprises... When a date coincides with an "*Exception day*" (programmed in a schedule), this special event will be applied instead of any *"Weekly event"* programmed, in the same schedule, to happen the same day.

Bottom left section: *"Weekly events"*

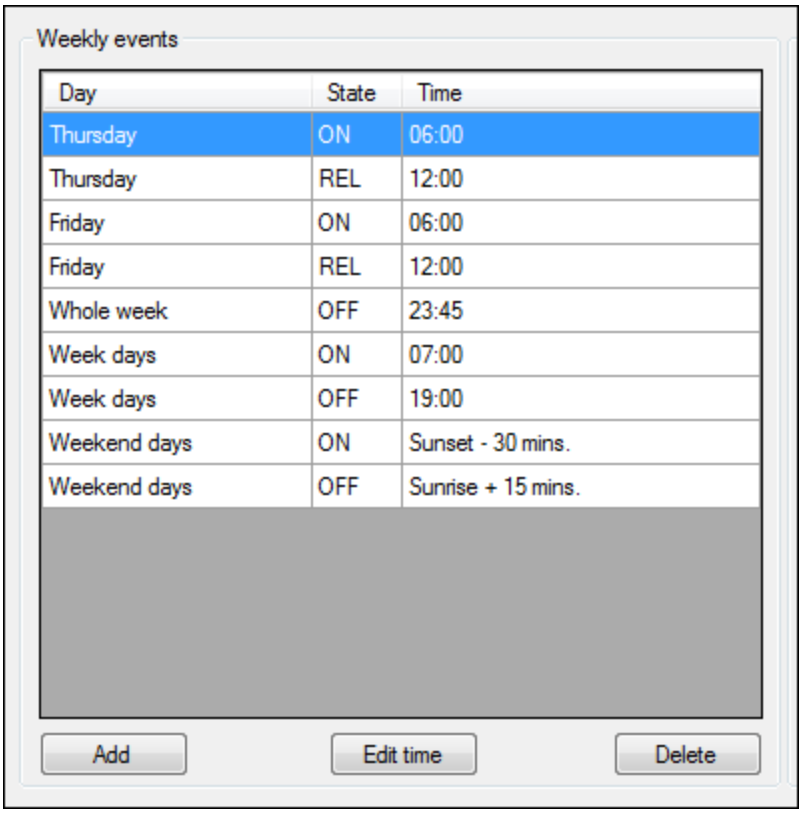


Figure 41: List of events programmed for a "Normal week"

This frame lists the events already created to happen on a weekly base, for the usual lighting management, indicating which days of the week and at what time the associated groups will be turned on or off (without consideration for the *exception days*).

- ***Columns composing the list of "Weekly events":***
  - ***"Day":*** this field indicates for which day(s) of the week the event will be applied. 10 possible values:
    - A single day, between Sunday and Saturday;
    - *"Week days"* for events happening every days between Monday and Friday (both included);
    - *"Weekend days"* for events only happening on Saturday and Sunday;
    - *"Whole week"* for events happening for each of the 7 days of the week.
  - ***"State":*** 3 possible values to send to the group(s):
    - *"ON"*, to set the priority applied by the schedule;
    - *"OFF"*, to reset the priority applied by the schedule;
    - *"REL"*, to *"Relinquish" (free)* the priority applied by the schedule.

      NOTES concerning the priority applied with the *"state"*:
      - The group options window (refer to section *2.5.2.5 KC600 group "options"*) allows the user to set for which priority the schedule *"states"* will be applied for this group.
      - Thus, a single schedule, associated with many groups, may be applied with a priority specific/different for each group controlled by the schedule.
      - If the *"state"* applied by the schedule is to "relinquish" the highest non-null priority, the new *("ON/OFF")* group *state* will determine which options, associated with the schedule, must be applied to the group.
  - ***"Time":*** time *(hours: minutes)* at which the *"State"* will be applied to the list of groups controlled by the schedule (refer to *Figure 40*) for the day(s) indicated in the "*Day"* column.

- ***Button to "Delete" a "Weekly event":***
  - You can delete a *"normal week"* event by selecting it in the list and then, by clicking on the *"Delete"* button below the frame. A message window will appear, asking you to confirm or cancel the deletion.
  - Only one event at a time can be deleted.

- ***Button to "Edit the time" of a "Weekly event":***
  - You cannot edit an event listed in the frame for *"Weekly events"*, except to change at what time the event happens.
  - You can *"Edit time"* for a single event at a time.
  - If you wish to change other parameters of an event, you will have to *"Delete"* it and create (*"Add"*) a new one with the appropriate parameters.
  - By clicking on button: *"Edit time"* under the frame for *"Weekly events"*, a window for *"Editing a weekly event"* will open, limiting you to change the happening time for the *"ON" / "OFF"* ***"State(s)"*** already established for the currently selected event. This window is identical to the one on *Figure 42*, all fields other than the ones for *setting time* being disabled in our case here.

- ***Button to "Add" a "Weekly event"*:**

  Click on the "*Add*" button, under the frame for *"Weekly events"*, to open a window for *"Adding a weekly event"* to the currently selected schedule:

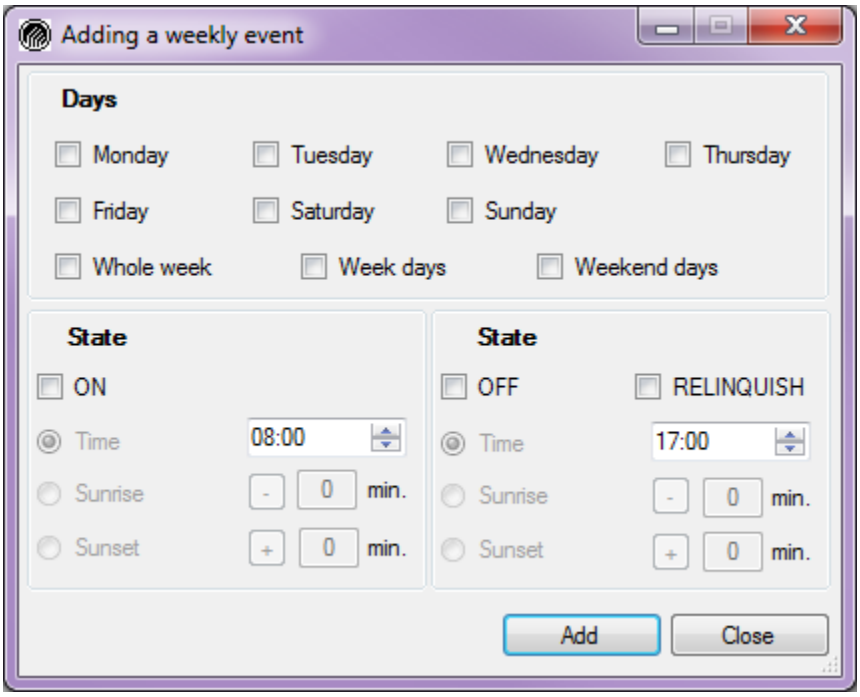
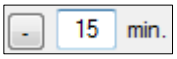


Figure 42: Window allowing you to program weekly events.

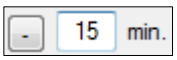
- ***"Days"* section:**
  - This block allows you to select for which day(s) you want to apply the event(s) defined in the lower sections.
  - When a box is checked for a sequence of days, on the last line of this block, the other possibilities are disabled
  - For each selected checkbox, one (or two if you checked both *"ON"* and *"OFF"*) event will be created, in the list of *Figure 41*, each time the *"Add"* button will be pressed.
  - .
- ***"State: ON"* section:**
  - To create an event to turn on the priority of each group of outputs associated to the schedule, you must check the ***"ON"*** box.
    - It is not necessary to create an *"ON"* event.
    - You can create both an *"ON"* and an *"OFF"* event at the same time.
  - Then 3 radio buttons are enabled to allow you to set the time at which the lights will be turned on.
    - ***"Time":*** for a fixed time of the day (hours: minutes), on a scale of 24 hours.
    - ***"Sunrise":*** the corresponding time changes every day, based on the date. It is calculated using the geographical coordinates (*"Longitude" / "Latitude"*) established in the *KC600* parameters; refer to item: *Parameters used to configure the KC600 as a BACnet object of "Device" type*, in section *2.5.2.1 KC600 card configuration parameters.*



: this block allows you to set an offset to apply the event not exactly at the sunrise time. Click on the left side button to display:

- : to delay the event a certain number of minutes BEFORE sunrise;
- : to delay the event a certain number of minutes AFTER sunrise.

- ***"Sunset":*** this time changes every day; refer to "*Sunrise"*.



: just like for the sunrise time, you can set here the offset for the application of an event around sunset time:

- : to delay the event a certain number of minutes BEFORE sunset;
- : to delay the event a certain number of minutes AFTER sunset.

- **"*State: OFF or RELINQUISH"* section:**
  - To create an event to turn off or to relinquish/free the priority defined for the link between the schedule and each one of the groups it controls. To activate this process, you must check the ***"OFF"*** or ***"RELINQUISH"*** box; refer to the previous section about *"State: ON"*.

    NOTE: you usually *"relinquish"* this way a priority to get all the groups priorities set to null, which forces the application of the state defined for parameter: *"Default Relinquish"*, mostly set to *"OFF"*.

- **"Add" button in the "Adding a weekly event" window:**
  - Once you defined your events (*"ON"* and/or *"OFF"*) in the current window, you can click the "*Add*" button at the bottom right of the window.
  - Each time you will press this *"Add"* button, the selected event(s) will be added to the "*Weekly events"* list of the schedule main window (refer to *Figure 41*):
    - This feature allows you to add many weekly events without having to close and reopen the window each time.
    - However, if you don't make any changes and click repeatedly on the "*Add*" button, you will add the same events repeatedly in the list shown on the left side of the main window …
    - If you click the *"Add"* button without checking the *"ON"* or the *"OFF"* box, nothing will be inserted in the weekly events list.
    - If both *"ON"* and *"OFF"* boxes are checked, then 2 events at a time will be added to the list each time you will click on the "*Add"* button, assuming that at least one day is selected in the upper section.
    - If you individually checked each day of the week, then clicking on the *"Add"* button will cause:
      - 7 additional events in the weekly events list (if you only checked one of the *"ON"* / *"OFF"* boxes);
      - 14 additional events if you checked both boxes.
    - For each day (from Sunday to Saturday), you are limited to 10 events.

Example: if you already programmed 5 events for the "*Week days"*, 2 events for the "*Weekend days"* and none for the "*Whole week"*, you will still be able to create:

- 5 Mondays, 5 Tuesdays, 5 Wednesdays, 5 Thursdays and 5 Fridays;
- 8 Saturdays and 8 Sundays;
- if no individual day has been chosen: 5 "*Whole week*" events.

- ***"Close"* button in the *"Adding a weekly event"* window:**
  - You can click on this button or the  button to close the window. After doing so, you will then return to the schedule main window.
  - If you did not click on the *"Add"* button in the *"Adding a weekly event"* window before closing it, you will lose the event(s) set, without any warning message.

Bottom right section: Events programmed for the *"Exception days"*

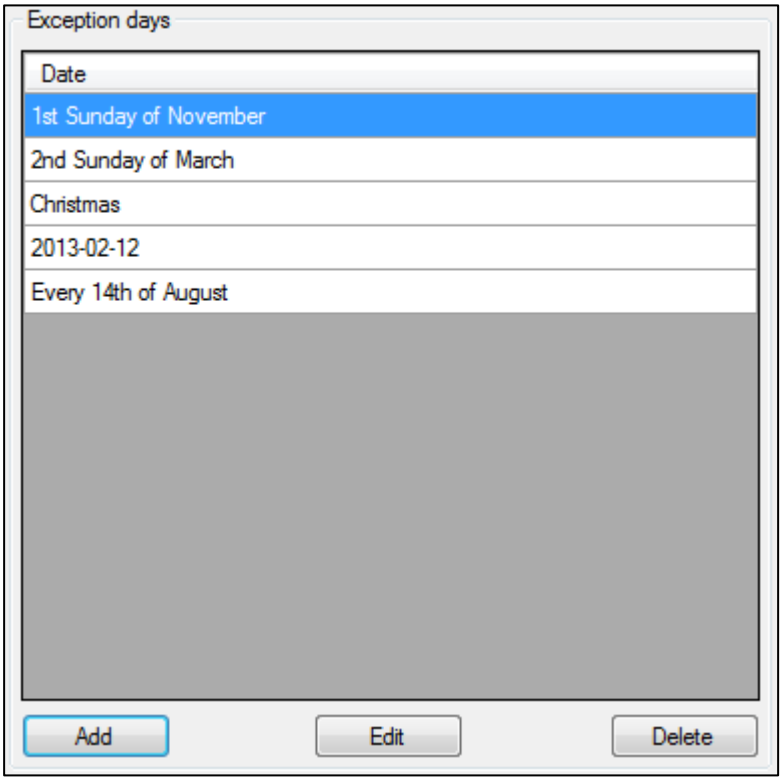


Figure 43: List of the events programmed for "Exception days".

This list displays the events created for particular dates throughout the year.

- In the *"Weekly events"* list (from *Figure 41*), each line indicates when an event will happen and what state will be applied to the group(s) controlled by the schedule.

  In the "*Exception days*" list (from *Figure* ***43***), only the date to which the event applies is indicated in order to simplify the display; no time, no state to apply since each *"Date"* appearing here may be composed of one or many events, that can occur at different times with different states.

- Each event defined in the "*Exception days*" list takes precedence over the events established in the *"Weekly events"* list (for a *normal week*).

- ***Button to "Edit" the events of an "Exception day":***

  Press this button to ***"visualize"*** or ***"edit"*** the composition of a *"Date"* selected in the "*Exception days*" list. The window that opens then is shown on *Figure 44*.

  - Opposite to the *"Weekly events"* for a *normal week*, you can edit here the whole configuration of an *exception date* (without being limited to the application time).
  - Only one *"Date"* at a time can be edited.

- ***Button to "Delete" an "Exception day":***

  - You can eliminate an exception *"Date"* (and all its associated events) by selecting one from the list (of *Figure 43*), then by clicking on the "*Delete"* button, under the frame. A message will ask you to confirm or cancel the deletion.
  - Only one *"Date"* can be deleted (and selected) at a time.

- ***Button to "Add" an "Exception day":***

  Click this button to access the window allowing you to define events happening at particular dates of the year:

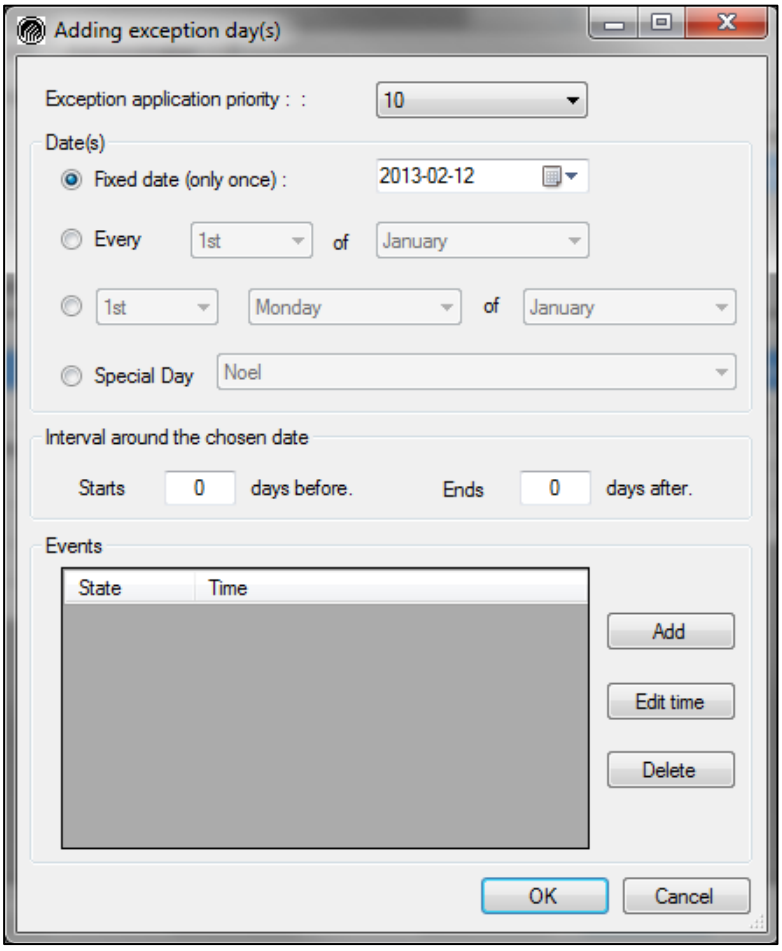


Figure 44: Window used to define the events for an exception "Date".

- ***"Exception application priority"***:
    - This priority (a value between 1 and 16) does not apply to the group(s) associated with the schedule.
    - It is the priority set for one *"Exception day"* to compare it to the priorities of other *"Exception days"* defined for the same schedule. That way, if more than one of the exception *"Dates"* share a common *"Date"*:
        - The *"Exception day"* with the highest priority (*"1"* being the highest) will be the one applied for the current schedule.
        - If there is more than one *"Exception day"* set to the same highest priority (for a common *"Date"*), the one that will be used is the oldest one created, not corresponding the order when you sort them by *"Date"*. When you first select a schedule from the objects of a *KC600*, you get the *"creation sort order"* in the list of *"Exception days"* for a schedule.
    - Remember that when we reach the date of an "*Exception day*", any event defined with the lowest priority will override all events defined in the *"Weekly events"* frame. The *"normal days"* events are therefore ignored when a *"special day"* occurs.

4 types of e*xception "Date(s)"*: (refer to *Figure 44*)

1. ***"Fixed date (only once)"*:**

   Select this type if you want a date that will happen only once, at a precise date in one single year, and never again afterward. You may set this date:

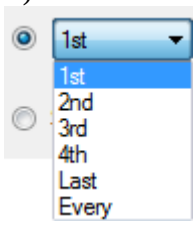
   - by typing numbers directly in the year, month and date fields;
   - by clicking on one of the 3 fields, then by using the up (⇧) or down (⇩) arrow keys, from your keyboard, to change the numerical value;
   - by clicking on the icon right to the numbers, to show the date manager in the form of a calendar.

2. ***"Every (Day) of (Month)"*:**

   To set a precise date that comes back every year.

3. ***"(Xth) (Day of the week) of (Month)"*:**

   This option is very versatile:

   - For a cyclic execution that repeats based on:
     - a particular day from the 7 days of the week,
     - a precise day that happens in a particular month,
     - or that must be repeated every month…
   - The first field (*"Xth"*) determines the occurrence of the *"day of the week"* during the selected *"Month"*: 

     , offering even the possibility to apply the "*Exception day*" every time the *"day of the week"* happens during the "*Month"*.
   - The second field ("*Day of the week"*) determines for which of the 7 days of the week (from Sunday to Saturday) the event(s) must be applied.
   - The third field ("*Month*"), allows you to choose for which of the 12 months the change of state must happen. This field also has a 13th choice, allowing you to apply the event(s) "*Every month*".

4. ***"Special Day"*:**

   This option allows you to choose one of the pre-determined "*Special days*":

   - saved with the *KCF600* console, thus available for each *"installation"*,
   - created from the corresponding option in the "*Schedule*" tab, in the window menu.

- For more details, refer to section: *2.7.3 Special days (for schedules): "Calendars"*.

- ***"Interval around the chosen date"***:
  - The parameters in this block are optional and can be left to zero.
  - They are mainly used when a fixed date was selected during the previous step.
  - It allows you to extend the selected date by making it:
    - *"Starts"* a certain number of days BEFORE the chosen *"Date"*,
    - to *"End"* a certain number of days AFTER the chosen *"Date"*.
  - For example: you chose the date type "*Special day*" and selected "*Christmas*" (which also could have been done by selecting the second type of date: "*Every 25th of December*"…). If your enterprise is closed between the 24th of December and the 2nd of January (included) and you wish to configure a particular lighting schedule for this period, you have to set the following parameters:

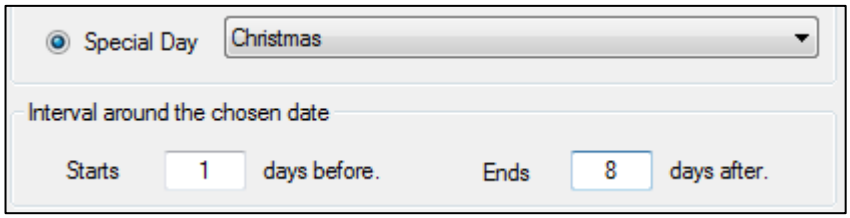


Figure 45: Parameters used to set an interval around the reference date.

- **List of programmed "Events":** (refer to *Figure 44*)

  This list displays the events ("*Time*" of the event and "*State*" to apply to the associated group(s)) that will happen for each day corresponding to the "*Date"* set above for the "*Exception day*", provided that the priority for this *"Date"* is the highest one at that moment.
  - You may leave the list empty if you do not want any state change to the group(s), associated with the current schedule, for the chosen period.
  - You can define up to 10 events in that list.
  - When you open the window, the events are listed in their order of creation. This order can be modified by clicking on the column headers (*"State"* and *"Time"*).

- ***"Add" / "Edit time" / "Delete"* buttons:**

  Refer to a previous section, in this chapter, titled: *Bottom left section: "Weekly events"* (starting with *Figure 41*), as this section works just the same.

2.6 Group management in a *KC600* / *"Groups"* tab

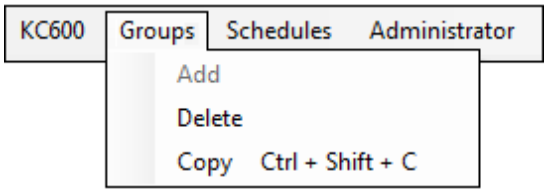


Figure 46: Items available with the "Groups" menu tab.

- The *"Groups"* tab does not appear in the menu:
  - if you logged yourself with the *"Monitoring"* profile (refer to section: *"2.1.1 Login window"*);
  - if you did not *register the application* yet…
- This menu only offers to ***create***, ***copy*** or ***delete*** groups. If you want to edit group parameters, refer to section *2.5.2.4 KC600 group configuration parameters*.

2.6.1 Add (a group)

- For this option to be available, you must first click in the left side list of the main window (refer to section *2.5.1 Left part of the main window: list of KC600 and their components*):
  - … on the name of a *KC600*;
  - … on the label *"Groups"* for the *KC600* where you want to create a new group.
- You can create up to 80 groups for each *KC600*.
- The created group is initially *"empty"*, without any output to compose the group or any input to control it. The default name follows the model: *"BIN_OUT_GROUP_XX"* where "*XX*" is the lowest available ID between 1 and 80.

2.6.2 Delete (a group)

- For this option to be available, you must first select a group name in the left side list of the main window.
- A confirmation window will appear asking you to confirm or cancel the deletion process.
- **NOTE:** If the group was controlled by one or many schedules, its deletion will provoke its elimination from all these schedules.

2.6.3 Copy (a group)

- For this option to be available, you must first select a group name in the left side list of the main window.
- A window will appear with the list of all *KC600* in the current *installation.* You must select to which *KC600* the selected group will be copied and then click on button:
  - "*Choose*" to start the transfer,
  - "*Cancel*" to abort the process.

2.7 Schedule management in a *KC600* / *"Schedules"* tab

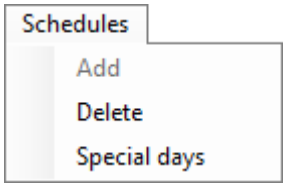


Figure 47: Items available under the "Schedule" menu tab

- The *"Schedule"* tab does not appear in the menu:
  - if you logged yourself with the *"Monitoring"* profile (refer to section: *"2.1.1 Login window"*);
  - if you did not *register the application* yet…
- This menu only offers to ***"add"*** or ***"delete"*** schedules. If you want to ***"edit"*** schedule parameters, refer to section *2.5.2.6 KC600 schedules configuration parameters*.
- This menu also allows you to manage the list of **"*Special days*"**, available for "*Exception days*" in each schedule for any *KC600* in every *"installation"* of your *KCF600* configuration console.

2.7.1 Add (a schedule)

- For this option to be available, you must first click in the left side list of the main window (refer to section *2.5.1 Left part of the main window: list of KC600 and their components*):
  - … on the name of a *KC600*;
  - … on the label *"Schedules"* for the *KC600* where you want to create a new group.
- You can create a maximum of 24 schedules in each *KC600*.
- An empty schedule is then created with the default name *"SCHEDULE_XX"* where "*XX*" is the lowest available ID (between 01 and 24) that is not already used to identify a schedule in a *KC600*

2.7.2 Delete (a schedule)

- For this option to be available, you must first select a *"schedule name"* in the left side list of the main window, appearing under the label: *"Schedules"*.
- A confirmation window will appear, asking you to confirm or cancel the deletion process.

2.7.3 Special days (for schedules): *"Calendars"*

- Here you can create **cyclic dates** like:
  - holidays and celebration days (*"Labour Day", "Thanksgiving", "Christmas", "New Year Day", "Independence Day",* etc.)
  - special vacation days,
  - special days for your enterprise, etc.

  which happen at a precise moment every year or each month.

- Particularities for this option:
  - *"Special days"* are not specific to only one "*installation*" or to a single *KC600*. What you create here is saved with the configuration console, so these *"special days"* will be available for every *"installation"* retrieved through the *KCF600*.
  - When a *"special day"* is used inside a schedule, this *"date"* is then represented by a *BACnet* object of type: "*Calendar*" in the *KC600* where is defined the schedule. Only the *"special day"* inserted is represented that way, not all "s*pecial days*" available for the *KC600* (but not used by its schedules).
- The "*Special days*" option is always available in the menu, regardless of the selection in the left side list.
- When you select this option, the following window is displayed:

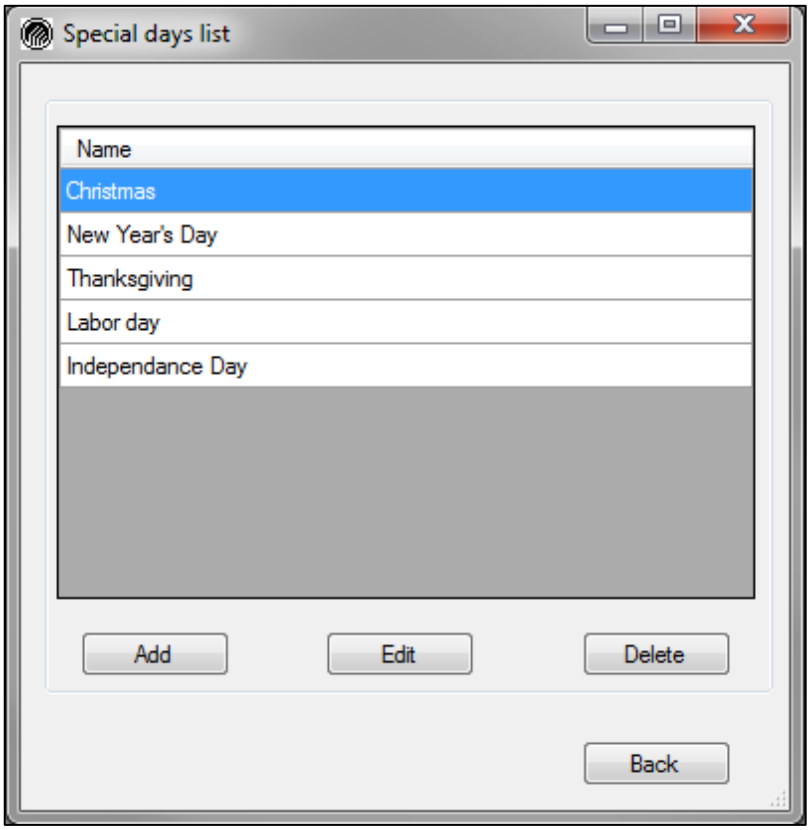


Figure 48: Window showing you the list of "Special days" already defined.

- ***"Special days list"*:**
  - All special days created for the current *KCF600* console are listed in that window.
  - If you want the names to appear in alphabetical order, press the "*Name*" column header.
- **"*Delete*" button:**

  Click that button to delete the currently selected "*Special day*".

  Warning: a warning message will ask you to confirm the elimination of the selected *"special day"* by notifying you of the consequences of this deletion: the removal of all events programmed to happen on this *"special day"*, used as an *"exception day"* in one or many schedules.

- ***"Edit"* button:**

  Press this button to open a window allowing you to edit the selected "*Special day*". Refer to the "*Add button*" section for more information on this window, only the title changing to indicate the *"edition"* instead of the *"addition"*.

- ***"Add"* button:**

  Press this button to open a window allowing you to create a new "*special day*":

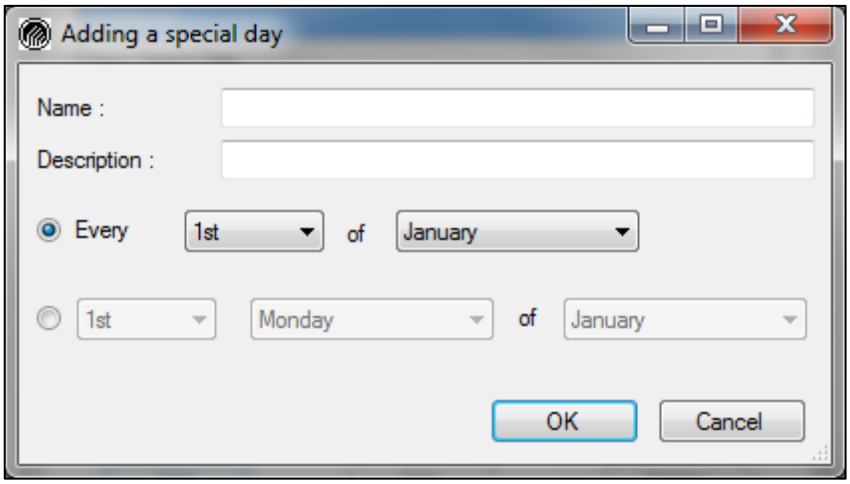


Figure 49: Window used to create/edit (based on the title) a "special day".

***"Adding a special day"* (or *"Editing a special day"*) window**

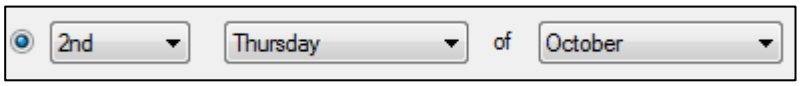
- ***"Name":***
  - It is the name that will appear in the list of *"special days"*, which can be used in a schedule to define an "*exception day*".
  - If this *"special day"* is used in a schedule, it will be represented by a *BACnet* object of type: *"Calendar"*. This *"name"* will then identify the *"Calendar"* object when the *KC600* will be asked to provide the list of its components/objects by a *BACnet* network management console.
  - Only limitations for the name:
    - It must be unique amongst all names used to identify other *"special days"* in the list.
    - It must be composed of at least 1 character.

- ***"Description":***
  - This information can be left empty if no additional information needs to be provided.
  - Again here, this information will be available in the *"Description"* parameter of the *"Calendar"* object, if the *"special day"* is used in a *KC600* schedule.

- ***Choice of date formats:***

  2 possibilities:

  - To establish a fixed date (day and month) that comes back every year (just like *"Christmas"* is always on December 25th).
  - If you enter an impossible date (for example: April the 31st) or a date that doesn't come back every year (for example: February the 29th), a message will inform you that your choice has been rejected.



  - For a cyclic execution that repeats itself based on:
    - a particular day among the 7 days of the week,
    - day that must happen during a precise month,
    - otherwise that must be repeated every month…
  - The first field determines the occurrence of the *"day of the week"* during the selected *"Month"*: *1st*, *2nd*, *3rd*, *4th* or *Last.* A 6th option allows this occurrence to happen every week of the selected *"Month"*.
  - The second field ("*Day of the week"*) determines for which of the 7 days of the week the date comes up.
  - The third field ("*Month*"), allows you to choose for which of the 12 months the date happens. This field also offers a 13th choice, allowing you to select "*Every month*".

Note:

For a holiday that lasts more than one day, you only have to establish here the principal date of the event. When creating an *"Exception day"* in a schedule, you will be able to select this *"special day"*, then set:

- a *"start date"* by adding an offset for the numbers of days BEFORE this reference date,
- an *"end date"* by adding an offset for the numbers of days AFTER this reference date.

Refer to the paragraph titled: *Bottom right section: Events programmed for the "Exception days"*, from chapter *2.5.2.6 KC600 schedules configuration parameters*.

2.7.4 *BACnet* representation for *KC600*'s *"calendars"* and *"schedules"*

The interface for our *KCF600* configuration console was first designed according to the needs of our clients, second to answer *BACnet* requirements. Thus the *KC600* must apply certain "conversions" to represent in *BACnet* some functionalities specific to our system.

2.7.4.1 Representation for *"calendars"*

The *"calendars"* are created as *"Special days"* from the option: *"Schedules"* appearing in the top menu, in the header of *KCF600* console.

- These *"special days"* are not created specifically for certain *KC600* controllers, but for the *KCF600* console. Thus, these *"calendars"* object are available not only for any *KC600* controller defined in the loaded installation, but also for any installation retrieved in the *KCF600* console.
- The *BACnet* types of *"calendars"* that we may create this way will target:
    - a single day of a year,
    - a particular day during a month,
    - one of the 7 days of the week for a target month or for every month.

  But unlike *BACnet* model, the *"special days" / "calendars"*, available in *KCF600* console, never target a period of consecutive days.
- Schedules may have *"Exception days"* (parameter: *"Exception schedules"* in *BACnet*) pointing to a *"special day"* (*BACnet* object: *"Calendar"*) from *KCF600* console.
    - No matter the number of *"special days" / "calendars"* created in *KCF600* console, only the ones referred by some schedules in a *KC600* controller will appear as *"Calendars"* in the list of *BACnet* objects defined in this *KC600*.
    - In *KCF600* console, a *"day/schedule of exception"* may point to a *"calendar"* with the addition of an *"offset"* in day(s) before and/or after the date referred by the calendar.
        - If all the schedules' references to the same calendar use the same *"offset"* values (null or other) around this target date, then the *BACnet* object: *"Calendar"*, in the *KC600*, will be represented with the calendar type: *"Date Range"* to emphasize the period over a few days.
        - If all the schedules' references to the same calendar, in a single *KC600*, use different *"offsets"*, then:
            - the corresponding *BACnet* object: *"Calendar"* will be represented, in the *KC600*, as defined in the *KCF600* console,
            - but the references to this date will be replaced in the schedules by local *"date ranges"*, not implying any link to a *"Calendar"* object.

2.7.4.2 *BACnet* representation for *"schedules"*

- All schedules created (from the *KCF600* console) in a *KC600* controller are represented as *BACnet* objects of type: *"Schedule"*.
- For any programmable event in a schedule, the *KCF600* console offers the possibility to apply *"sunset"* and *"sunrise"* times, with or without *"offsets"* in minutes, to trigger an event. When you send a *"Read Property"* command to get the application time of an event that refers to such a "variable" reference (changing daily), the answer provided by the *KC600* will indicate the exact time, including the *"offset"* time shift, corresponding to the **date** or the period associated with the event .

  Examples:

  - For events defined in the *"weekly schedule"*, the time returned by the *KC600* will corresponds to a fixed time at which the sun rises (or goes down) for the next day of the week (Sunday to Saturday) closer to the current date.
  - For *"exception days"* (*"exception schedule"* in *BACnet*), the corresponding time will be computed according to the closest target date or period in the future.
- At the end of section: *2.5.2.5 KC600 group "options"*, there is a *"Warning regarding the compliance of options in the BACnet environment"*, the application of these particular options being reflected in proprietary properties 900 to 903, for each *BACnet* object of type *"Binary Output"*. These proprietary parameters may be activated or not by schedules, this feature being configured from the *"Group options' window"*, but this association is not visible in *BACnet* objects of type: *"Schedule"*.
- Certain schedules may not be represented in *BACnet* exactly as they are defined in the *KCF600* configuration console. These schedules imply *"exception days"* including date *"offset(s)"* around a reference date that is not a fixed one during the year.
  - **For examples:**
    - for every Mondays of each month, with the addition of a one day *"offset"* after the reference (to include also every Tuesdays);
    - the last Wednesday in February, with *"offsets"* of minus one day and plus 2 days…
  - This feature to add date *"offsets"* offers more possibilities for the client, to determine when to apply events, at the expense of representing in *BACnet* all possible combinations now available.
  - For these special cases, the *"offsets"* values will be omitted in the *BACnet* representation of the *"Schedule"* objects, implying such a conflict.
- Most parameters, attached to *"Schedule"* objects, **cannot be changed through a *"Write Property"*** *BACnet* command, to make sure not to lose any special parameter, in our schedule configuration, which are "invisible" to *BACnet* or that can't be represented in *BACnet*, as mentioned in this section.

2.8 "Administrator" menu items / "Administrator" tab

If you are logged in the *KCF600* console as an *"Administrator"*, there will be an additional tab in the top menu:

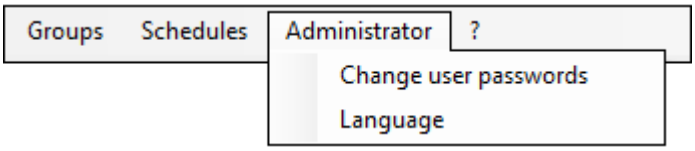


Figure 50: "Administrator" menu items.

2.8.1 Change user passwords

After selecting this option, a window will be displayed allowing you to change the login password for each one of the 4 user types supported by the *KCF600* console (refer to section *2.1.1 Login window*):

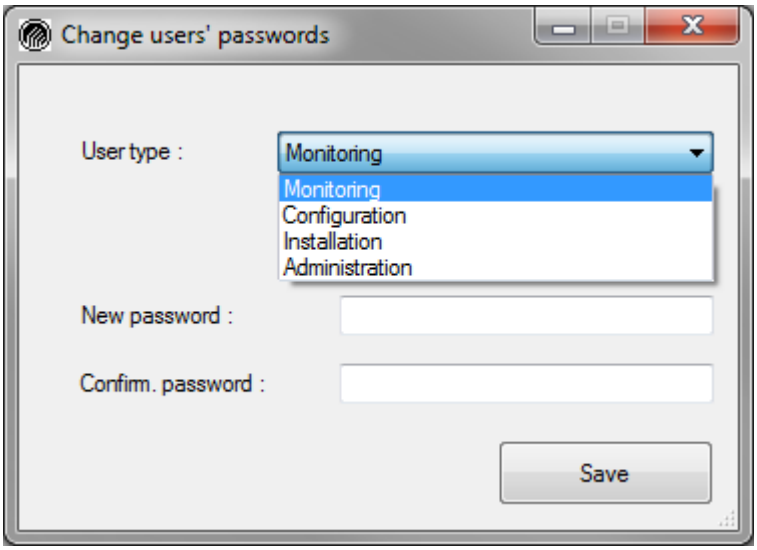


Figure 51: Change users' passwords window.

- Only for the *"Monitoring"* profile: you may leave both password boxes empty if you do not want any password for this profile.
- For the other 3 types of profiles, a password is required.
- It must be composed of at least one character, maximum 20. The password is case sensitive.
- You must enter the same password in both password boxes, in the lower part of the window, to make sure that no mistyping happened while entering the password.
- **Special case:** for the "*Administrator*" profile, a change of password requires that we fill a 3rd field: the *"Old password"* must be entered as a supplemental safety measure to make sure that the change is really made by the *"Administrator"* himself.
- You must press the "*Save*" button to save all your password changes.
- If you enter a new password for a profile 1, but you change the user type for a profile 2 before pressing the *"Save"* button, then the password will not be saved with profile 1; it will be saved instead with the last *user type* selected, which is profile 2.

2.8.2 Language

A language change has impacts on 2 levels:

1. the console display is translated;
2. for every *BACnet* object (*KC600* as a *Device*, *Binary Inputs*, *Binary Outputs*), their description (and the device location) will be translated if it is still based on the default model.

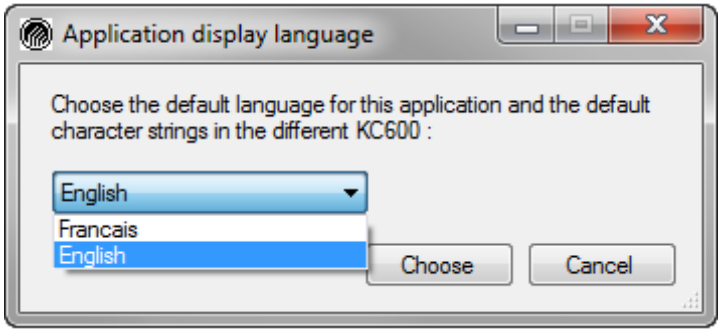


Figure 52: Application display language window.

- To confirm the language change, press the "*Choose*" button.
- Right after this *"first confirmation"*, another window will ask you for a *"second confirmation"* while notifying you of all the consequences following a change of language:
  - every configuration in the current *installation* will be saved,
  - the console will restart,
  - all character strings based on the default models, for the *BACnet* objects of each *KC600* configuration, will be translated. But you will need to send back the configurations (menu *"KC600 / Send config."*) to all *KC600* for the changes to take place.

3.*KC600* Control Board

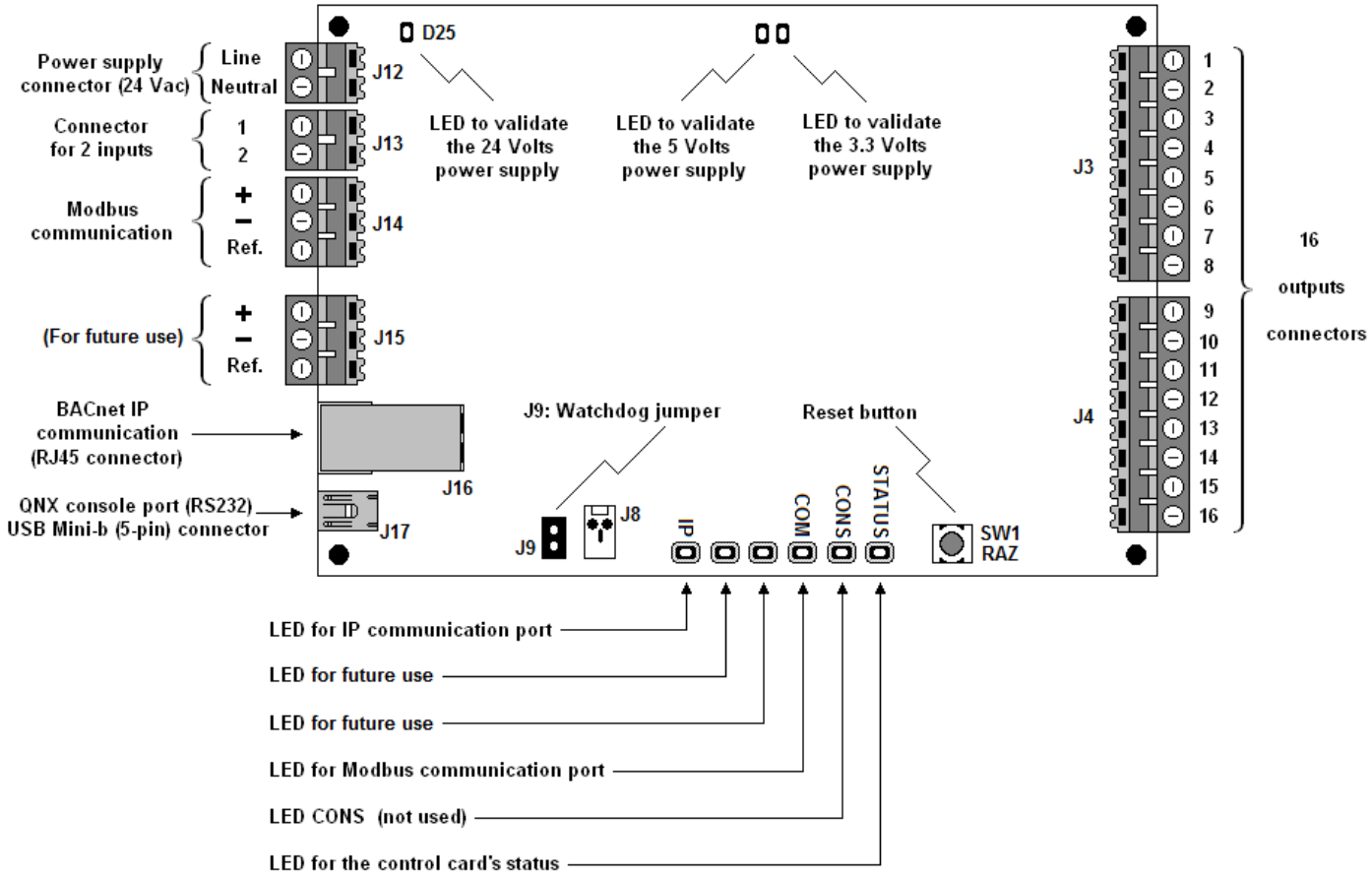


Figure 53: Main components identified on the KC600 board.

3.1 *KC600* Board Connection Points

The main connection points are shown at left and right on *Figure 53*. This chapter gives details only about those requiring more information.

3.1.1 *Modbus* Communication Port

- This port provides connection for one or more *KC621* and/or *KC631* and/or *KC632* boards to add supplemental inputs and/or outputs to the *KC600*.
- If the *KC600* does not have to control more than 2 inputs nor more than 16 outputs, there is no need to connect anything to this port.
- The *Modbus* port uses the following communication parameters:
  - 8 data bits
  - 1 stop bit
  - Parity = None
  - Data Type = Binary
  - Baud Rate = 19200.

3.1.2 *BACnet IP* Communication Port

- When a cable is plugged in the RJ45 socket, if the board is detecting connection to the IP network, the small two rectangular LEDs are lit in the socket upper section, above the attached cable:
  - upper left, the green LED shall be continuously lit to confirm the IP network detection, the proper connection to the network;
  - upper right, the ocher/orange LED flashes indicating IP messages exchanges.

3.1.3 QNX Console Port *(RS232)*

- To validate proper starting up of QNX in the *KC600* and to allow for exchanges with this operating system, you must connect the *KC600* board to a computer USB port via a **"Mini USB" cable** (Type A to Mini-B 5-pin, male to male).
- Under *MS Windows* systems, this port is shown as an RS232 port (*COMX* port where *X* is the number of this serial port). To determine the port number, you need to refer to the list provided by opening the *"Control Panel" – "System" – "Hardware" – "Device Manager" – "Ports (COM and LPT)"*.
- With a communication utility software like *"HyperTerminal"* or *"Tera Term"*, select the corresponding *"COMX"* port and configure it as follows:
  - Baud Rate = 115200
  - Data bits = 8
  - Parity = None
  - Stop bit = 1
  - Flow Control = None.
- Once these required connections and communication utility configuration are completed, restart the related *KC600* board.
- After rebooting the *KC600* board, among all data returned via this serial port, you need to check the **board network coordinates** (IP address, network mask and gateway address), shown when QNX has almost completed starting up in the *KC600* board.

3.2 Light Emitting Diodes (LEDs) on *KC600* Board

3.2.1 Three LEDs about Electric Power on the Board

- These three LEDs are shown in the upper part of . They show that the board is in operation and they validate that the 3 voltage power levels (24 volts, 5 volts and 3.3 volts) are within required nominal voltages.

3.2.2 LED for the Control Card's Status

- After initialization sequence completion, this LED (lower part on *Figure 9*, the rightmost LED of this group) shall **flash green** to confirm proper operation of the software managing the *KC600*, at one-second interval.
- This LED goes **ocher/orange** during the start-up of the application managing the *KC600*.
- If this LED remains **unlit** while the board is powered (see item 3.2.1) and if the initialization sequence (lasting many seconds) is completed, this means that the application did not start, being missing or corrupted.

3.2.3 Five LEDs for Communication Status with Various Ports

All these LEDs are shown in the lower part of *Figure 9*. The important ones to watch are:

- **Communication LED via IP port:**
  - This LED goes **ocher/orange** for the whole start-up period of the application managing the *KC600* if all goes well during IP port initialization.
  - It turns to **red** if an error is detected during the configuration or after the reception of an erroneous message at this port. It remains lit red momentarily during a temporary error and permanently if the detected error prevents proper IP communication operation.
  - This LED goes momentarily **green**:
    - upon receiving an IP message (*TCP* or *BACnet IP*) addressed to the *KC600* card and if no error is detected during the exchange;
    - during an IP message transmission (*TCP* or *BACnet IP*) by the *KC600* card.
  - Monitor this LED (making sure first that service requests are continuously transmitted by a *BACnet* control console) in order to confirm proper reception of ***BACnet IP*** messages.
  - **NOTE**: Do not mistake this LED with the others on the *BACnet IP* communication port connector (see corresponding item); here the LED only turns green to signal the detection of valid *BACnet IP* messages, addressed to the *KC600* card, which excludes any *TCP-IP* message like those transmitted by the *KCF600* configuration console.

- **Communication LED via *Modbus* port:**
  - This LED goes **ochre/orange** during **initial** start-up period of the application managing the *KC600*. During the second start-up phase, while all other LEDs in this block remain ochre/orange:
    - it will flash **green** every time a *Modbus* address (between 1 and 15) will be queried to discover if a card is connected to this address. In such a case, the flashing will be brief before going to the next request. Otherwise, almost one second will elapse before next address query;
    - if the software managing the *Modbus* service requests does not start (because it is missing or corrupted), the LED will flash continuously in **red**, even during steady state (see below).
  - Once the initializations are completed, the card will run in steady mode:
    - The LED will light up briefly in **green** for each successful read (or write) request to a *Modbus* card.
    - The LED may light up briefly in **red** (half a second ON) for every service request without answer from a *Modbus* card. The initialization phase being completed, if you have at least one *Modbus* card connected to the *KC600*, you may verify that the LED flashes in red by unplugging the *Modbus* port.
    - The LED will flash continuously in **red** if the software, managing *Modbus* service requests in the *KC600*, is missing or corrupted. To distinguish this case from the previous one, unplug the *Modbus* port on the *KC600* card and restart it. If the LED still flashes in red, then the problem is related to the *Modbus* manager that does not start.

4. *KC621* Card to Add 16 Inputs to the *KC600*

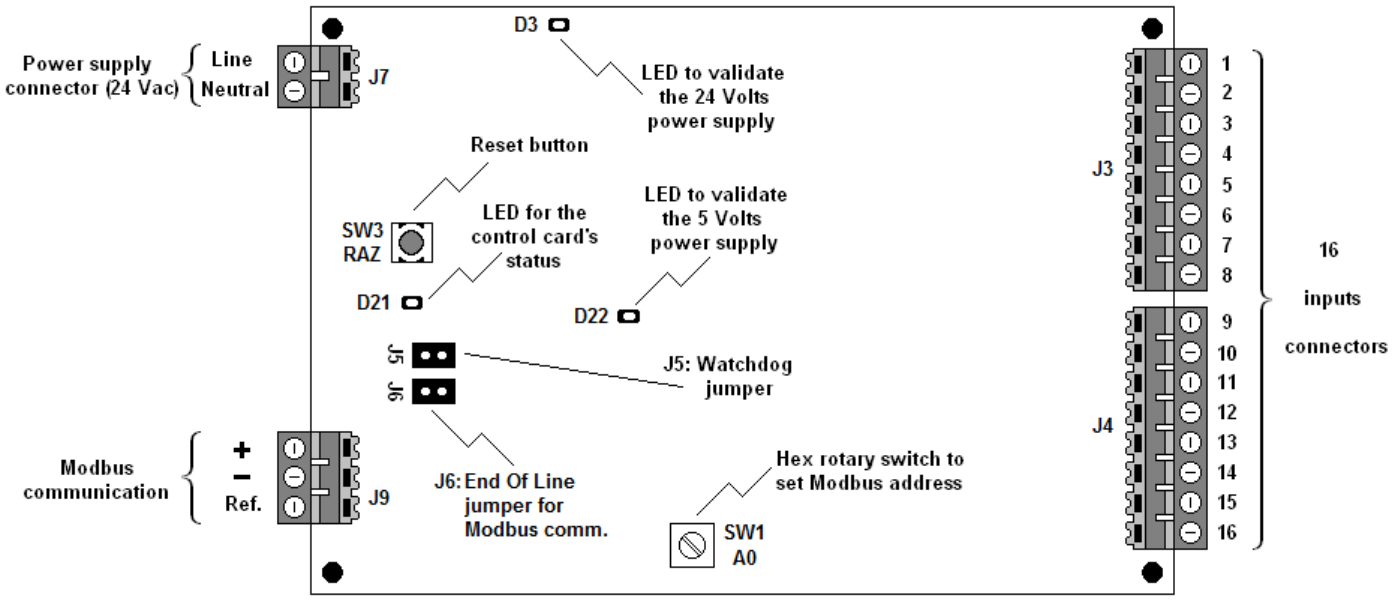


Figure 54: Main components identified on the KC621 board.

4.1 Light Emitting Diodes (LEDs) on *KC621* Card

- **2 power LEDs:**
  - Check these LEDs to confirm that the card is properly powered ON.
- **Status LED:**
  - This LED flashes continuously when the software managing the *KC621* card is running properly.

4.2 Address Rotary Switch

- **16 possible positions indicated with hexadecimal numbers:**
  - At position "0", the card will not communicate.
  - Any of the other 15 positions (between "1" and "F") will unlock *Modbus* communication.
  - **WARNING:**
    - Each *Modbus* card, either with inputs or outputs, must have a unique address, different from the other cards connected to the same *KC600*. Thus, we cannot have 2 *Modbus* cards sharing the same address on a common *Modbus* communication line.
    - Any time a *Modbus* card address is changed, the card must be restarted in order to activate this new value (you may simply press the *"Reset"* button for 2 seconds).
    - For the *KC600* card to detect this new address, it must also be restarted. Do so once all addresses changes are completed, with prior restart for each *Modbus* card with a new address.

5. *KC631* Card to Add 16 Outputs to the *KC600*

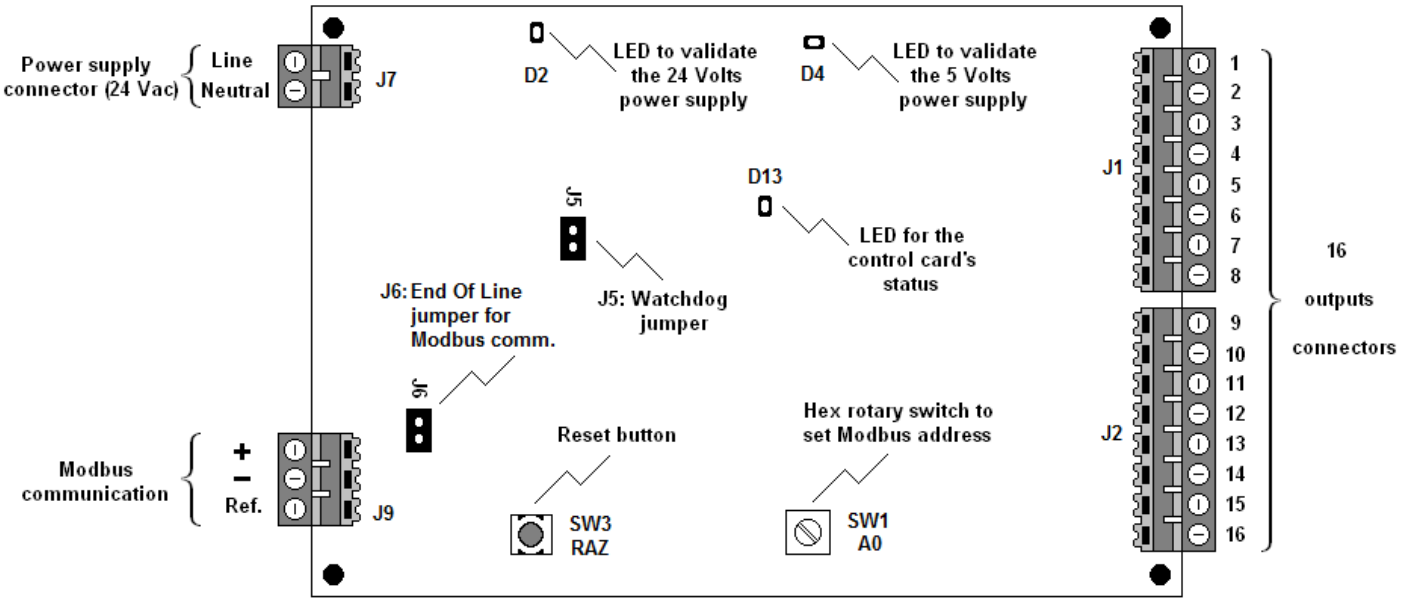


Figure 55: Main components identified on the KC631 board.

5.1 Light Emitting Diodes (LEDs) on *KC631* Card

- **2 power LEDs:**
  - Look for these LEDs to confirm that the card is properly powered ON.
- **Status LED:**
  - This LED flashes continuously when the software managing the *KC621* card is running properly.

5.2 Address Rotary Switch

- **16 possible positions indicated with hexadecimal numbers:**
  - At position "0", the card will not communicate.
  - Any of the other 15 positions (between "1" and "F") will unlock *Modbus* communication.
  - **WARNING:**
    - Each *Modbus* card, either with inputs or outputs, must have a unique address, different from the other cards connected to the same *KC600*. Thus, we cannot have 2 *Modbus* cards sharing the same address on a common *Modbus* communication line.
    - Any time a *Modbus* card address is changed, the card must be restarted in order to activate this new value (you may simply press the *"Reset"* button for 2 seconds).
    - For the *KC600* card to detect this new address, it must also be restarted. Do so once all addresses changes are completed, with prior restart for each *Modbus* card with a new address.

ANNEXES

Annex 1. First Time Attribution of Network Coordinates to a *KC600* Card

By default, any *KC600* is set with the following network coordinates:

IP Address:	`192.168.23.250`
Network Mask:	`255.255.00.00`
Gateway:	`192.168.254.254`

These values have been selected to minimize conflict possibilities (with other network components), of addresses sharing, when inserting an unconfigured *KC600* in a *BACnet* network. When the *KC600* uses these basic coordinates, it is recommended to connect the computer with the *KCF600* configuration software directly to the *Ethernet* port of a *KC600* card.

WARNING: To connect a computer directly to the *Ethernet* port of a *KC600* card:

- do not use a *"straight through cable"* (also called *"patch cable"*). This type of cable is used to connect an interface card (from a computer in our case) to a *network switch*, a *hub*, or a *router*;
- you must use:
  - an ***"Ethernet crossover cable"*** between the PC and the *KC600* card
  - or a *"network switch"* as a go-between the PC and the *KC600* card (with *straight through cables*), the switch requiring to be isolated from the rest of the *network* for direct configuration of a single *KC600*.

PROCEDURE:

- First connect the *KC600* card USB port (RS232) to the computer (refer to section *3.1.3 QNX Console Port (RS232)*) to get the current card network coordinates.
- Then, connect the computer to the *KC600* via its *Ethernet* port (if not already done). To check if it is possible to gain access to the card address from the computer, open a *"Command prompt"* window (see *Figure 56*), normally available in the sub-menu *"Accessories"* from *MS Windows* program list. Then, write a *"ping"* command as follows (with the default IP address of a *KC600* card as an example):

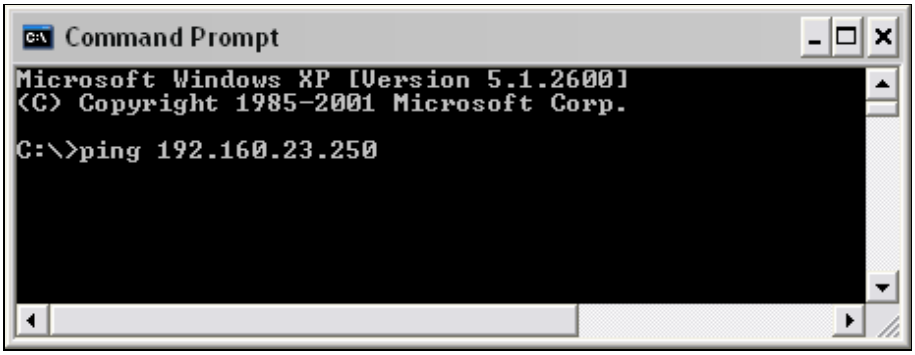


Figure 56: Command prompt window (under MS Windows) with a "ping" command.

- If the *"ping"* command returns an answer, you may start the *KCF600* configuration console to assign the required coordinates to the *KC600* for its integration in the *BACnet IP* network.

- If the *"ping"* command returns no answer, the computer network coordinates will have to be changed (currently attributed automatically by the DHCP service):
  - **Under *MS Windows 7***: *Control Panel* → *Network and Sharing Center* → *Change Adapter Settings* → *Properties* on the network card → *Internet Protocol Version 4* → *Properties*
  - **Under *MS Windows XP*:** *Control Panel* → *Network Connections* → ***Local*** *Area Connection* → *Properties* of the network card → *Internet Protocol (TCP/IP)* → *Properties*

Under *MS Windows XP*, the target screen is as follows:

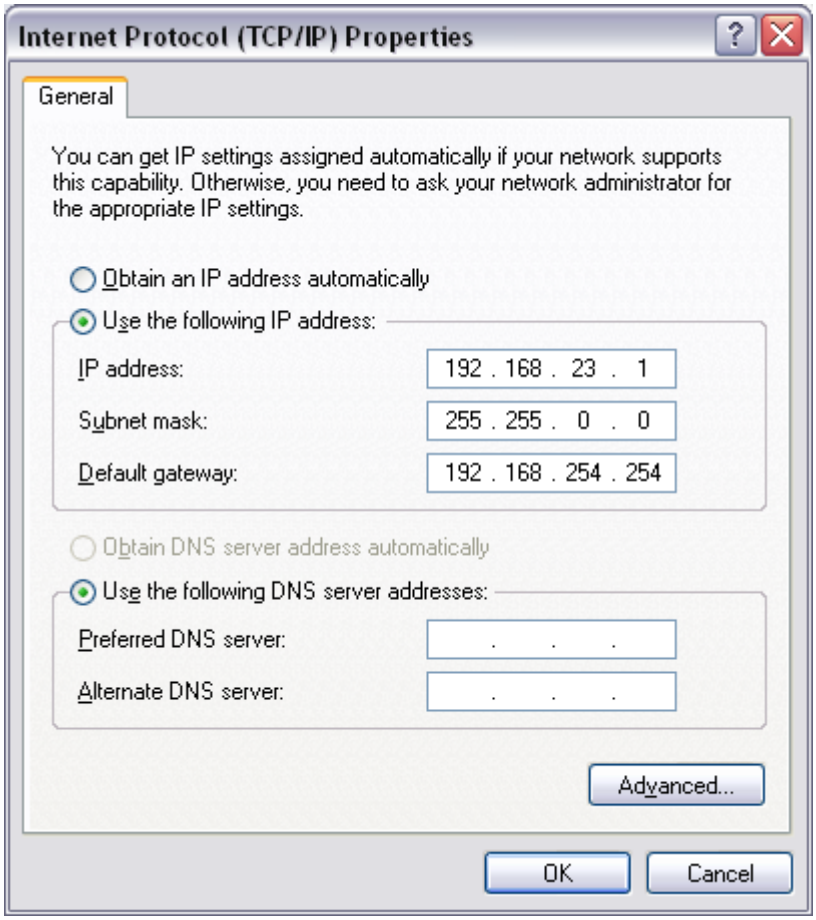


Figure 57: MS Windows XP screen to set the computer network address.

Select *"Use the following IP address"* and set your computer to the network coordinates shown on *Figure 57*. It is only when the *"OK"* button is pressed on this screen (and the following window is closed, if required, without clicking on *"Cancel"*), that the new network coordinates are applied to the computer.

- From the ***"Command Prompt"*** window (see *Figure 56*), send another *"ping"* command to the *KC600* card IP address to confirm that the communication is properly established between the devices. Otherwise, check the following:
  - the proper *Ethernet* cable is used,
  - it is correctly connected at both ends,
  - the IP exchanges management is working properly in the *KC600* card; check the followings:
    - if the software controlling the *KC600* is well started, the status LED should flash in green (refer to section *3.2.2 LED for the Control Card's Status*);
    - if both LEDs integrated to the IP port connector, on the *KC600* card, signal the detection of IP communication (refer to section *3.1.2 BACnet IP Communication Port*).
- Once the IP communication is confirmed between both devices, you can use the *KCF600* configuration console to establish the network coordinates for the *KC600* card; refer to section *2.3.1 Configuration Exchanges with KC600:*
  - first read the current configuration in the *KC600,*
  - set the new network coordinates for the *KC600* card with the *KCF600* console,
  - then transmit the new configuration.
- **WARNING:** Once the configuration of all *KC600* is completed, do not forget to reset the computer network address so that it is attributed automatically in a computers network, by checking the box *"Obtain an IP Address Automatically"* (refer to *Figure 57*) on the *MS Windows* screen to set the *"Internet Protocol (TCP/IP) Properties"*.

Annex 2. Priority tables for binary outputs

(Refer to section 19.2 of the *BACnet* standard: *Command Prioritization)*

Priority tables according to *ASHREA* standard

If a *"BACnet Device"* is composed of "*Binary Outputs"*, we must be able to send it write commands to change its state through the property: "*Present value"*. This property must be "*commandable*", so a request to change the state must be composed of 2 parameters: the desired state and a **priority**. This system was established in *ASHREA* standard to consider that many "*commanding entities*" (a *BACnet* console/application, an operator, etc.) can send requests to the same output in order to change its current state ("*Present value*"). To avoid conflicts between these requests, a priority is attributed to each of the *BACnet* "*commanding entities*" that may send such requests. However, a *BACnet* console (for example) is not limited to one pre-established priority level and can transmit requests with different priorities (not recommended by the *ASHREA* standard, according to *section 19.2.1.3*).

- The transmitted state to a binary output is a value of type: *"BACNET_BINARY_PV"*. The 3 states allowed for this type are:
  - BINARY_INACTIVE = 0 ➔ OFF
  - BINARY_ACTIVE = 1 ➔ ON
  - BINARY_NULL = 2 ➔ NULL
- The **priority** is a value between 1 and 16, *"1"* being the highest priority (applied before all others) and *"16"* being the lowest one.
- Some priorities are already predefined:
  - **Priority 1:** *Manual-Life Safety*;
  - **Priority 2:** *Automatic-Life Safety*;
  - **Priority 5:** *Critical Equipment Control* ;
  - **Priority 6:** *Minimum On/Off;* this priority can only be used if an algorithm is implemented, regarding both optional properties: *"Minimum_On_Time"* and *"Minimum_Off_Time"* (refer to *section 19.2.3* of the *ASHRAE* standard).
  - **Priority 8:** *Manual Operator*.
- In every "*BACnet Device*" composed of "*Binary Outputs*", there is a ***"priority table"*** (*"Priority_Array"*) for each of the binary output. Each table then has 2 dimensions (*Priority/State*) to associate a state for each of one of the 16 priorities:

Priorités (#)	État (*On/Off/Null*)	Etat équivalent *BACnet*
Priorité 1	NULL	BINARY_NULL
Priorité 2	NULL	BINARY_NULL
Priorité 3	NULL	BINARY_NULL
Priorité 4	NULL	BINARY_NULL
Priorité 5	**OFF**	**BINARY_INACTIVE**
Priorité 6	NULL	BINARY_NULL
Priorité 7	NULL	BINARY_NULL
Priorité 8	**ON**	**BINARY_ACTIVE**
Priorité 9	NULL	BINARY_NULL
Priorité 10	NULL	BINARY_NULL
Priorité 11	NULL	BINARY_NULL
Priorité 12	NULL	BINARY_NULL
Priorité 13	NULL	BINARY_NULL
Priorité 14	NULL	BINARY_NULL
Priorité 15	NULL	BINARY_NULL
Priorité 16	NULL	BINARY_NULL

Figure 58: Example of a priority table with 2 effective priorities (5 and 8).

- The state applied to the physical component associated to this binary output is the state of the highest non-null priority (called ***"effective priority"***) appearing in this table.
- This "*currently applied*" state of the binary output is saved in the "*Present_Value*" property.
- If all 16 priorities are set to *"NULL"*, then the "*Present_Value*" is set to the value defined for the mandatory and configurable parameter: *"Relinquish_Default"*, which is *"ON"* or *"OFF"*.
- At start up, when a device boots with objects of type "*Binary Output*":
  - the 16 priorities of this table are set to: *"BINARY_NULL";*
  - the *"Present_Value"* property corresponds to the state (*"ON"* or *"OFF"*) that was read, during the initialisation phase, for the physical component connected to the "*Binary Output*".
- If a binary output receives a *BACnet* request of priority 8 (for example) to turn *"ON"* the output, the value *"BINARY_ACTIVE"* will be assigned to priority 8 in its priority table. As only priority 8 is not set to the value *"BINARY_NULL"*, the state (*"ON"*) of priority 8 will be applied to the output and will be saved in the "*Present_Value"* parameter.
- Next, if the same output receives a priority 5 request to turn it *"OFF"*, there will be 2 non-null states in the priority table; refer to *Figure 58*. The non-null state of the highest priority (5) being *"BINARY_INACTIVE"*, the corresponding output will be set to "*OFF*" (and saved in the "*Present_Value"* parameter).
- If the output receives a priority 7 request to turn it "*ON*", as this priority is lower than the (highest) *"effective priority"* (5), it will be saved in the priority table but not applied to the current state of the output, without any effect on the "*Present_Value"* parameter:

Priorités (#)	État (*On/Off/Null*)	Etat équivalent *BACnet*
Priorité 1	NULL	BINARY_NULL
Priorité 2	NULL	BINARY_NULL
Priorité 3	NULL	BINARY_NULL
Priorité 4	NULL	BINARY_NULL
Priorité 5	**OFF**	**BINARY_INACTIVE**
Priorité 6	NULL	BINARY_NULL
Priorité 7	**ON**	**BINARY_ACTIVE**
Priorité 8	**ON**	**BINARY_ACTIVE**
Priorité 9	NULL	BINARY_NULL
Priorité 10	NULL	BINARY_NULL
Priorité 11	NULL	BINARY_NULL
Priorité 12	NULL	BINARY_NULL
Priorité 13	NULL	BINARY_NULL
Priorité 14	NULL	BINARY_NULL
Priorité 15	NULL	BINARY_NULL
Priorité 16	NULL	BINARY_NULL

Figure 59: Example of a priority table with 3 non-null priorities (5, 7 and 8).

- If the state of the command is *"BINARY_NULL"*, then it means a request to *"relinquish"* the passed priority.
- Back to our example, if the output receives a "*relinquish*" command of priority 5, this priority is then set back to "*BINARY_NULL"* in the table. We have then 2 priorities (7 and 8) in the priority table with non-null states. Since priority 7 is the (highest) *effective priority*, its associated state is applied to the output (and to the "*Present_Value"* parameter) which goes from *"OFF"* to *"ON"*; see *Figure 60*:

Priorités (#)	État (*On/Off/Null*)	Etat équivalent *BACnet*
Priorité 1	NULL	BINARY_NULL
Priorité 2	NULL	BINARY_NULL
Priorité 3	NULL	BINARY_NULL
Priorité 4	NULL	BINARY_NULL
Priorité 5	NULL	BINARY_NULL
Priorité 6	NULL	BINARY_NULL
Priorité 7	**ON**	**BINARY_ACTIVE**
Priorité 8	**ON**	**BINARY_ACTIVE**
Priorité 9	NULL	BINARY_NULL
Priorité 10	NULL	BINARY_NULL
Priorité 11	NULL	BINARY_NULL
Priorité 12	NULL	BINARY_NULL
Priorité 13	NULL	BINARY_NULL
Priorité 14	NULL	BINARY_NULL
Priorité 15	NULL	BINARY_NULL
Priorité 16	NULL	BINARY_NULL

Figure 60: State of the priority table after relinquishing priority 5.

- If the output receives a request to relinquish priority 7, only priority 8 will be remaining with a non-null value and its state (*"ON"*) will be applied to the output.
- Finally, by relinquishing the last effective priority (8), every priority in the table will be set to *"BINARY_NULL"*. The state of the "*Relinquish_Default*" value, associated with the binary output object, will be applied and its value saved in the "*Present Value*" parameter.

END PAGE