

# Generic Commands

This page is under construction

When an object is added to the System, and if this object allows it, it is possible to access its many commands and use them on buttons, inside the logic module, universal gateway, etc.

The commands are grouped into 5 categories:

- **System**
- **Reports and Gateways**
- **Internal Services**
- **Multimedia**
- **Security**

The category will only be visible if the related objects have been added inside the System tab.

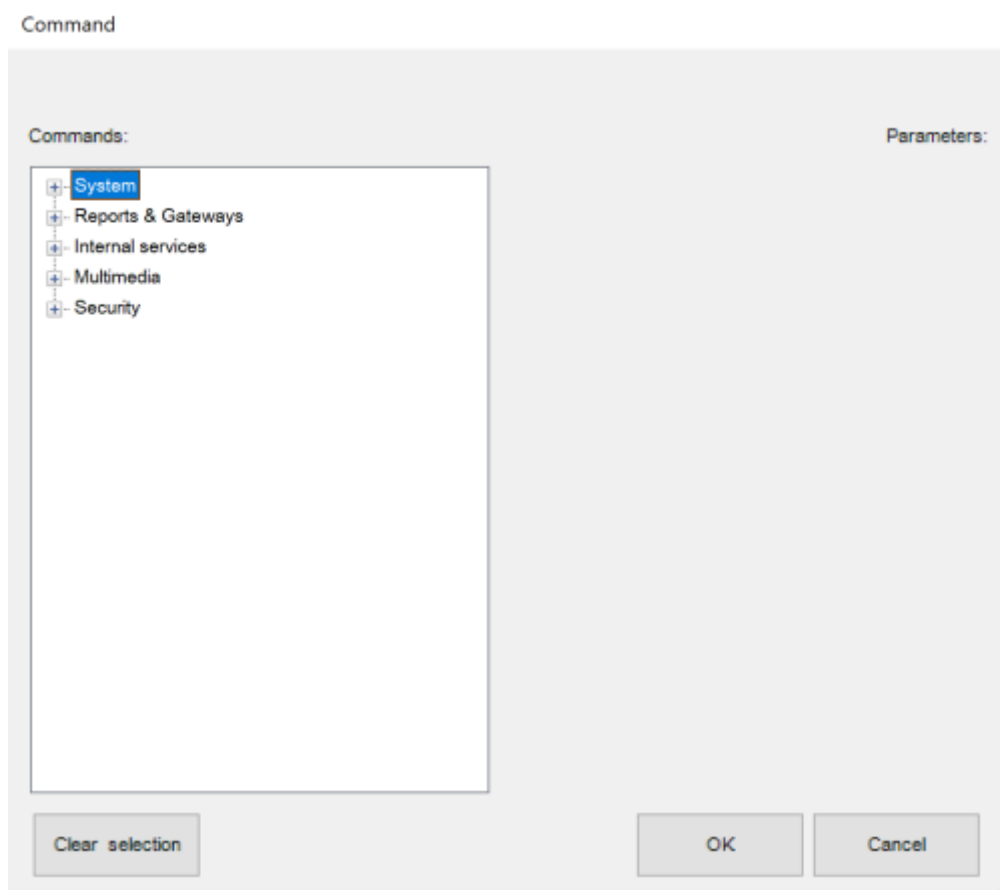


Figure 1: Thinknx Objects Commands

## System

The available commands when selecting the **System** category are the following:

### Send KNX Bit telegram

This command is used to send a 1 bit value to the KNX bus.

- **KNX Group** Group address to send the value.
- **KNX value** Can be either 1 or 0.

A generic button can be configured to send a 1 bit KNX telegram to open a door.

### Send KNX Byte Telegram

This command is used to send a 1 byte value to the KNX bus.

- **KNX Group** Group address to send the value.
- **KNX value** Can be a value between 0-255.

A generic button can be configured to send a fixed dimming value to a dimming circuit, for example sending the value 128 to make the light go to 50%.

### Send KNX Float 2 Byte Telegram

This command is used to send a 2 byte float value to the KNX bus.

- **KNX Group** Group address to send the value.
- **KNX value** Can be a value between -671088,64 and 670760,96.

A generic button can be configured to send a fixed temperature value to a thermostat, for example sending the value 21.5 as a setpoint.

### Send KNX Float 4 Byte Telegram

This command is used to send a 4 byte float value to the KNX bus.

- **KNX Group** Group address to send the value.
- **KNX value** 4-Octet Float Value IEEE 754

### Send KNX String Telegram

This command is used to send a string of 14 bytes to the KNX bus.

- **KNX Group** Group address to send the value.
- **KNX value** string text to be entered.

### Send KNX 4 Bit Telegram

This command is used to send a 4 bit value to the KNX bus.

- **KNX Group** Group address to send the value.
- **KNX value** Can be a value between 0 and 15.

A generic button can be configured to decrease (value 0-7) or increase ( value 8-15) a light dimming value. It can also be used to change a shutter position (use values 0-7 for UP, and 8-15 for DOWN). For example, sending the value 5 to a dimmer will decrease the brightness by 6%, while sending the value 10 to a shutter will lower its position by 50%.

### [Read KNX group address](#)

This command is used to read a value from the KNX bus.

- **KNX Group** Group address to read the value from.

If a device installed on the KNX bus does not update its status automatically, this command can be used to force the reading on the group address. It can be used on a generic button to read the values when clicking the button, or inside a logic module whenever the device is turned ON.

### [Make a pause for a fixed interval](#)

This command pauses the system for a specified time interval configurable in milliseconds.

- **Interval** Time interval in milliseconds.

When saving a scene, sometimes a pause between two commands is necessary, for example, when saving the TV channel number. Another application is creating a pause before turning off the final light in a Goodbye scene, making sure that the client has left the house.

### [Change UI Function and Page: GENERAL](#)

This command allows to redirect **all clients** to a specific function and page.

- **Function** insert the Index number of the destination function.
- **Page** insert the Index number of the destination page.

Using the Universal Gateway, a scenario can be configured to redirect the users to the Main Entrance camera page whenever the doorbell rings.

### [Change UI Function and Page: SPECIFIC CLIENT](#)

This command allows the **specific client** who has sent it to access a certain function and page.

- **Function** insert the Index number of the destination function.
- **Page** insert the Index number of the destination page.

Invisible generic buttons can be placed on the house plan for each room, allowing every single client to navigate through the rooms by clicking on each area.

## Send Push Notification

This command allows to send push notifications to the clients. By accessing the web page of the server and clicking on Server -> Licenses and Codes, it is possible to enable/disable the receipt of push notifications for each client.

- **Message** insert the message to be displayed on the clients.
- **Maximum number of push notifications with the same identifier allowed in 10 min**  
This parameter is not mandatory. However, if a value has been entered, the system will make sure to limit the number of push notifications sent to this number in 10 min. It is quite useful when the trigger for sending the push notifications is being sent frequently on the bus.
- **Push Notification Identifier** A small descriptive text to allow the system to differentiate a push notification from another one when calculating the rate limit. Mandatory only if the previous parameter has been used.

Using the Universal Gateway, a push notification can be sent to the client when a 1-bit value is received from the bus to indicate that the Water Tank Level is low. The message in that case can be "Low Water Level!". If the 1-bit is being sent periodically to the bus with a small interval, then a maximum number of push notifications can be inserted in the related parameter.

## Execute MS Windows Command: GENERAL

This command allows to launch an .exe file directly from the ThinKnx application on all Players for Windows.

- **Command**
- **Parameter**

## Execute MS Windows Command: SPECIFIC CLIENT

This command allows to launch an .exe file directly from the ThinKnx application on all Players for Windows.

- **Command**
- **Parameter**

## Recall iOS app with url

This property allows to launch a specific application on iOS devices directly from the ThinKnx app by typing the related URL.

- **URL used to recall iOS app** Type in the correct URL to open the desired installed application, for example <http://www.google.com> will automatically browse the google safari page. Another example is typing sonos:// to open the SONOS app.

A generic button can be configured for the client to open another application on the iOS device.

### Send Email to recipients

This command allows to send an email message using a default ThinKnx account.

- **Email Subject** Enter the email title here.
- **Email Recipients** Enter destination email accounts.
- **Email Content** Enter the email content here.

If the client's house is empty and a presence has been detected inside (1-bit KNX telegram), a scenario can be programmed on the Universal Gateway to send an email to the owner of the house.

### Send DTMF tone during intercom call

This command allows to send a DTMF tone used in telephony while an intercom call is running on the client.

- **DTMF tones sequence** enter the sequence of numbers to send during the intercom call. Sometimes it is necessary to end the sequence with the symbol "#", depending on the application.

A generic button can be configured in the Intercom page to send a DTMF tone to open the main gate whenever a call is established.

### Make SIP call

When using the Thinknx server as PBX, it is possible to make internal calls between all the devices.

- **Extension to call** Enter the extension of the destination device.

A house owner can have a generic button on his application to call the extension of the touch screen installed in Kitchen to communication with the help.

### Start audio notification sound in all the clients

This command allows to play an audio sound in all the clients.

- **Sound to play** Choose between different beeps to play: 0=Beep\_1, 1=Beep\_2, 2=Alarm\_1, 3=Alarm\_2, 4=Siren\_1, 5=Siren\_2.
- **Duration of the sound to play** Specify the duration of the sound to play in seconds. If 0 is entered, the sound will play endlessly until a Stop command has been sent.

An emergency push button can be installed in bathrooms to send a KNX 1-bit value. Using the Universal Gateway, this value received can launch the command to start audio notification on all clients.

## Stop audio notification sound in all clients

This command allows to stop the audio sound previously launched in all clients.  
See above command for example.

## Reports and Gateways

The below list contains all the possible objects that could be added to the Reports and Gateways category:

- [Gateway Modbus](#)
- [Lutron Project](#)
- [Serial Gateway](#)
- [Ethernet Gateway](#)
- Gateway MyHome
- Voice Control Gateway
- IFTTT Account
- SMS Gateway

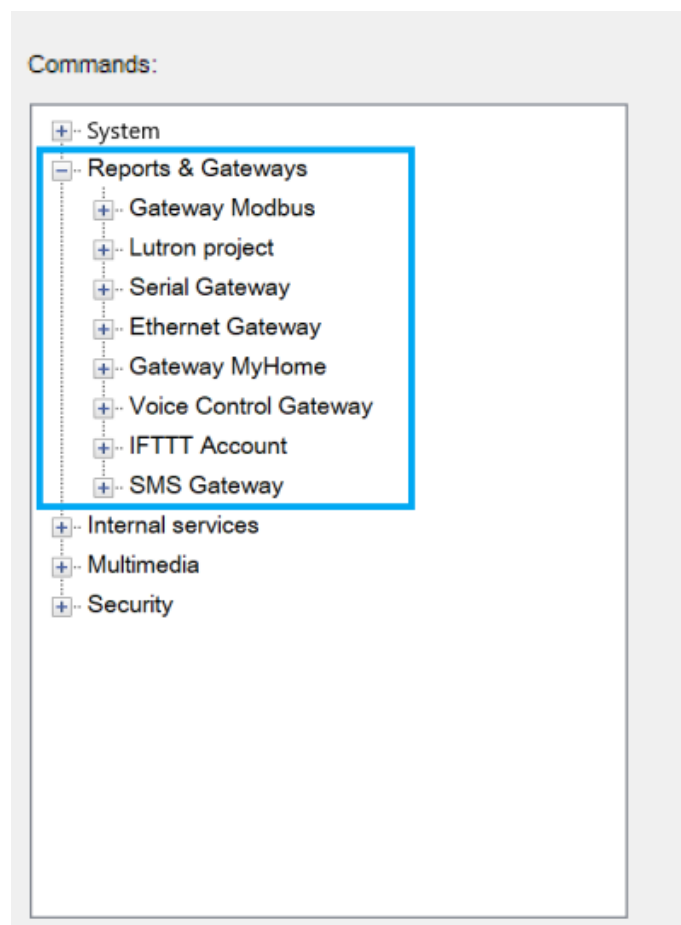


Figure 2: Reports and Gateways

### Gateway Modbus

The available commands when selecting the **Gateway Modbus** category under **Reports and Gateways** are the following:

#### [Send Modbus value with function 5](#)

This function is used to send a value to a Modbus device supporting function 5, known as “write single coil”.

- **Modbus address** Modbus address.
- **Coil Address** Discrete output address (coils are 1-bit registers).
- **Value** could be 1 or 0.

### [Send Modbus value with function 6](#)

This function is used to send a value to a Modbus device supporting function 6, known as “Write Single Holding Register”.

- **Modbus Address** Modbus address.
- **Register Address** Slave register.
- **Value** value to write.
- **N.Byte** Number of bytes.

### [Send Modbus value with function 16](#)

This function is used to send a value to a Modbus device supporting function 16, known as “Write Multiple Holding Registers”.

- **Modbus Address** Modbus address.
- **First Register Address** Slave register.
- **Value** value to write.
- **N.Byte** Number of bytes.

### [Send Modbus value with function 22](#)

This function is used to send a value to a Modbus device supporting function 22. The function can be used to set or clear individual bits in the register.

- **Modbus Address** Modbus address.
- **Register Address** Slave register.
- **Value** value to write.
- **Selected bits mask** mask value of the selected bits to be changed.

## **Lutron Project**

The available commands when selecting the **Lutron Project** category under **Reports and Gateways** are the following:

### [Simulate keypad button press](#)

This command is used to simulate a keypad button press on one of the Lutron devices installed.

- **System address of the keypad** Address of the keypad with the desired button to simulate.

Communication is done by connecting to the Lutron processor. Address shall be written using the following template "processor:link:address".

- **Button number to operate** Number of the button to press on the Lutron keypad.

### Simulate keypad button release

This command is used to simulate a keypad button release on one of the Lutron devices installed.

- **System address of the keypad** Address of the keypad with the desired button to simulate. Communication is done by connecting to the Lutron processor. Address shall be written using the following template "processor:link:address".
- **Button number to operate** Number of the button to release on the Lutron keypad.

### Simulate keypad button hold

This command is used to simulate a keypad button hold on one of the Lutron devices installed.

- **System address of the keypad** Address of the keypad with the desired button to simulate. Communication is done by connecting to the Lutron processor. Address shall be written using the following template "processor:link:address".
- **Button number to operate** Number of the button to hold on the Lutron keypad.

### Simulate keypad button double tap

This command is used to simulate a keypad button double tap on one of the Lutron devices installed.

- **System address of the keypad** Address of the keypad with the desired button to simulate. Communication is done by connecting to the Lutron processor. Address shall be written using the following template "processor:link:address".
- **Button number to operate** Number of the button to double tap on the Lutron keypad.

### Select Sivoia state

This command is used to control Lutron shutters/blinds through a Sivoia control device.

- **System address of Sivoia Control** Address of the Sivoia Control device. Communication is done by connecting to the Lutron processor. Address shall be written using the following template "processor:link:address".
- **Sivoia state to select** O=open, C=close, S=stop.

### Select Grafik Eye scene

This command is used to play a scene configured with Lutron Grafik Eye system.

- **System address of Grafik Eye to control** Address of the Grafik Eye device. Communication is done by connecting to the Lutron processor. Address shall be written using the following



template "processor:link:address".

- **Scene number to select** Input the scene number (0=OFF).

### Select Level Lights

This command is used to send the lights to a certain level.

- **Output integration ID** ID of the output to control.
- **Level** Input a number between 0 and 255 to send the lighting level.

### Select Raising Lights

This command is used to start raising the lighting level of the desired output.

- **Output integration ID** ID of the output to control.

### Select Lowering Lights

This command is used to start lowering the lighting level of the desired output.

- **Output integration ID** ID of the output to control.

### Stop Raising or Lowering Lights

This command is used to stop raising/lowering the lighting level of the desired output.

- **Output integration ID** ID of the output to control.

## Serial Gateway

The available commands when selecting the **Serial Gateway** category under **Reports and Gateways** are the following:

### Send string with no termination

This command is used to send a text without termination.

- **String to send** input the string that will be sent to the serial gateway.

### Send string with CR termination

This command is used when sending a text with CR termination, marking the end of the line.

- **String to send** input the string that will be sent to the serial gateway.

## Send hexadecimal string with no termination

This command is used to send an hexadecimal string without termination.

- **String to send** input the string that will be sent to the serial gateway.

## Send hexadecimal string with CR termination

This command is used when sending an hexadecimal string with CR termination, marking the end of the line.

- **String to send** input the string that will be sent to the serial gateway.

## Ethernet Gateway

The available commands when selecting the **Ethernet Gateway** category under **Reports and Gateways** are the following:

### Make an HTTP GET call to a specified url

From:  
<http://www.thinknx.com/wiki/> - Learning Thinknx

Permanent link:  
[http://www.thinknx.com/wiki/doku.php?id=generic\\_commands&rev=1554196639](http://www.thinknx.com/wiki/doku.php?id=generic_commands&rev=1554196639)

Last update: **2019/04/02 11:17**

