

## “WILDCARD” DRIVERS FOR CRESTRON HOME

Revision: 1.02

Date: 20 September 2022

### LICENCING

Please visit <https://shop.ultamation.com/> to obtain a licence key for your Crestron processor. Each module TYPE requires a specific licence, but each TYPE can be used multiple times with the same licence of the same Crestron Home processor.

### CRESTRON HOME ONLY

This document describes the configuration of Ultamation's range of "Wildcard" custom device drivers for Crestron Home only. Wildcard drivers are the Swiss Army Knife of drivers to integrate equipment that is otherwise difficult to include into a Crestron Home system.

There are two "classes" of wildcard driver. Control drivers are used to send commands to devices. The devices in question must be VERY simple to control and there is no feedback.

Listener drivers are used to receive packets to trigger events within Crestron Home. The packets must be of a specific format (described later) and can trigger up to 20 different events. Multiple trigger tiles can be used in a system but they will each need their own port.

### ADDITIONAL READING

Please refer to the Using 3<sup>rd</sup> Party driver document for instructions on installation and configuration of the driver with Crestron Home.

### SUMMARY - WHAT DO THESE MODULES DO?

There are times when you don't need a fancy "do everything" module for a complex piece of equipment, and all you really want is something to fire a simple command at a device so that it can perform some other function. This might be a TV lift, projection screen masking, a door opener, or even turning an amp on or off.

For those situations, you may have had to introduce a second processor simply to send string of characters to a serial port, and there will still be times when you need richer integration with another processor (using [SIMPL Home](#) for example), but for the quick and easy occasions, Ultamation's "Wildcard" drivers are here to keep it simple (KISS).

We have a suite of drivers that provide a one-way, fire-and-forget solution that allows an integrator to send literal strings via serial, TCP, UDP and even (very simple) HTTP/S GET and HTTP/S POST.

We have now added the mirror to these drivers by providing some simple input triggers that allow external systems to trigger events in Crestron Home. Those events are generic Crestron Home sequences – so can do anything from turn on some lights, to closing a relay to setting up your house for a party. The challenge is now how creative you can be with Crestron Home.

We'll now describe the general module structure, followed by the specifics for each variation.

## COMMAND MODULE DESCRIPTION

The extension module provides basic transmission of data via various transports from the standard Crestron Home user interface via the powerful Extensions Device framework.

Each device to be controlled will require a separate definition using the driver, but the same licence key can be used for multiple devices on a single processor. In the case of serial devices, only one driver instances can connect to a physical serial port.

After importing the driver (see the general "How to" document), the Wildcard tile(s) will be found under:

Devices > Custom Device > Ultamation

### Adding the Tile

Ensure a room is selected, and then click on the '+' against the appropriate Wildcard driver item.

You will be asked to provide a descriptive name for the tile – this can be anything you wish.

---

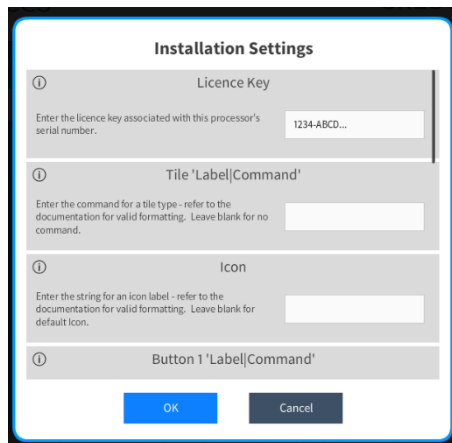
For the serial driver, you will also need to specify the serial port which the device to be controlled is connected to.

For the TCP and UDP drivers, you will need to specify the IP Address and Port for the host device you wish to deliver strings to.

For the HTTP/S GET and POST driver, you will need to specify the IP Address and Port of the HTTP/S host, choose whether you require HTTP or HTTPS and you can OPTIONALLY also specify a username and password. If these are filled in, then the HTTP/S client will use these credentials in each request using BASIC authentication. This is the ONLY authentication method supported at present.

---

The device will be added to your system and you will be asked to enter some more settings. These settings provide the licence key to ensure that the integration will function after the initial one hour trial period.



The image shows a screenshot of the 'Installation Settings' dialog box. It has four sections, each with a help icon (i) and a text input field. The first section is 'Licence Key' with the instruction 'Enter the licence key associated with this processor's serial number.' and a placeholder '1234-ABCD...'. The second section is 'Tile 'Label|Command'' with the instruction 'Enter the command for a tile type - refer to the documentation for valid formatting. Leave blank for no command.' The third section is 'Icon' with the instruction 'Enter the string for an icon label - refer to the documentation for valid formatting. Leave blank for default icon.' The fourth section is 'Button 1 'Label|Command'' with the same instruction as the tile section. At the bottom are 'OK' and 'Cancel' buttons.

The first parameter will be the licence key that was emailed to you with your order acknowledgement. The same key should be used in each tile OF THE SAME TYPE in the system (though different Crestron Home systems will require different licence keys). If you do not enter a valid licence key here, the system will function for 1 hour and then control will be suspended. The licence key can be entered at any time after initial

setup without having to reconfigure each tile.

The second parameter is used to provide a title and "command" to execute when the tile is tapped in the Crestron Home UI.

The format of this parameter must take the following format:

<Display Name>|<Data to Send>

The display name will be shown on the control (tile or button). A pipe '|' separator must follow the display name.

After the pipe, you specify the data to be sent when the action is triggered. This can take the form of simple ASCII text, and we also support the standard Crestron form of escaping hexadecimal characters – i.e. \x0D = a byte value of 13, or \x22 the quote (") character. If you wish to include leading or trailing spaces in the data, you can enclose the entire string in double quotes.

If the parameter is left blank, the Tile will not have an action associated with it.

Some examples are:

**Press Me|This is the data to send\x0D\x0A**

**Send Binary|\x01\x02\x03\x04**

**Open Gate|api/v1/door/5s**

The format for HTTP/S GET is the same as for Serial, TCP and UDP, except that the "data" here is the URL endpoint to be requested on the HTTP/S host. i.e. if you wanted to make a request to [http://www.host.com/open\\_the\\_gate.cgi](http://www.host.com/open_the_gate.cgi), the "data" would be the section following the host – i.e. "open\_the\_gate.cgi"

The format for HTTP/S POST is as for HTTP/S GET with an extra pipe, and data field which defines the POST data. The format here is:


<Display Name>|<URL endpoint>|<POST data>

A POST request usually sends a block of data within the body of the request, rather than encoded within the URL endpoint.

An example of an HTTP POST is the Nexus 21 TV lift:

```
Home|api/command|{"COMMAND":"HOME"}
```

```
Preset 1|api/command|{"COMMAND":"MEMORY POSITION 1"}
```



### Icon

Enter the string for an icon label - refer to the documentation for valid formatting. Leave blank for default Icon.

Next input is the icon label. Here you can select an icon for the tile from the Crestron Device Icon datasheet. If the chosen icon isn't valid, it will default to the Crestron default. If it's left blank, it will default to a standard icon. Sometimes an CH app restart will be needed immediately after changing the icon, this is a bug within CH and is only needed after updating the label. The link for this is below:

[https://sdkcon78221.crestron.com/sdk/Crestron\\_Certified\\_Drivers\\_SDK/Content/Extension-Device-Icons.pdf](https://sdkcon78221.crestron.com/sdk/Crestron_Certified_Drivers_SDK/Content/Extension-Device-Icons.pdf)



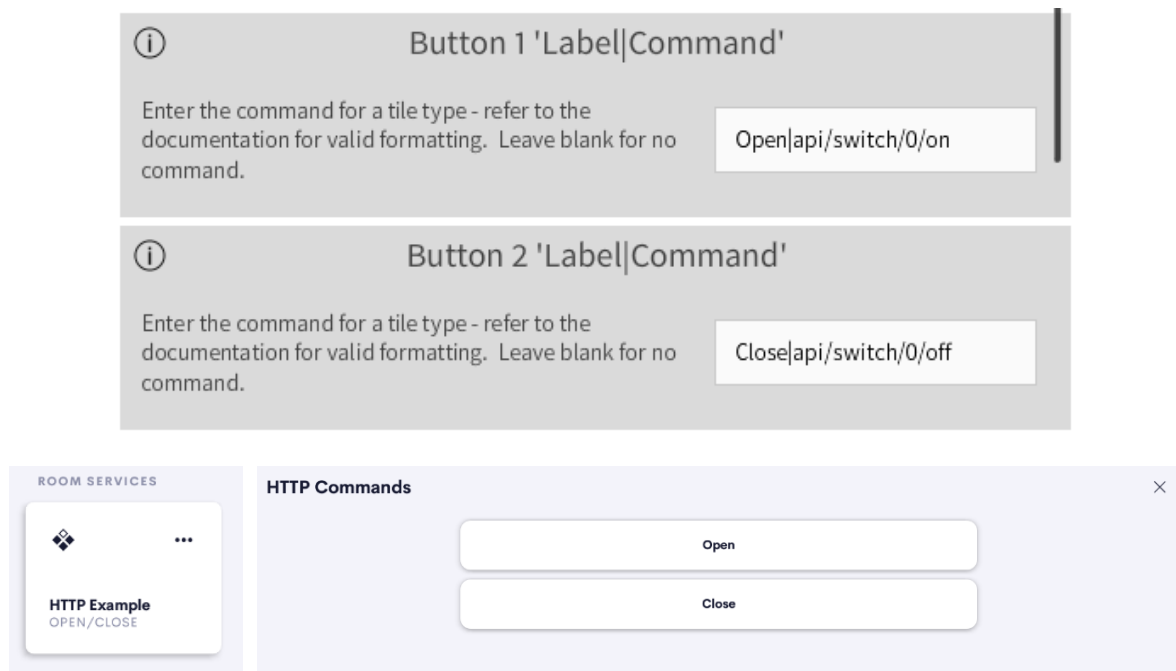
### Layout Title

Enter a header text for the page which appears after clicking the three dots on the tile.

The next input field is the layout title text. This allows you to set a custom title text for the page which appears after you click the three dots on the tile. If this is left blank the title will be set to a default value

Following the Layout Title field there are 8 button labels & commands. These take exactly the same format as the tile parameter, but for each defined parameter, a button will be presented when the tile is opened. This allows the user interface to offer several fixed commands against buttons.

For example:



## SERIAL, TCP, UDP

These drivers could not be simpler. You specify a string against each action, and the action will transmit that string to the device when it is triggered.

There is no feedback or error checking.

## HTTP/S GET

This is much the same as for the other transports, except that we also support BASIC authentication for HTTP or HTTPS requests. Note: this is:

- ☼ HTTP or HTTPS (assumes valid certificates)
- ☼ "No Auth" or "Basic Auth" (i.e. not Digest, etc.)
- ☼ GET only, not PUT, POST etc.

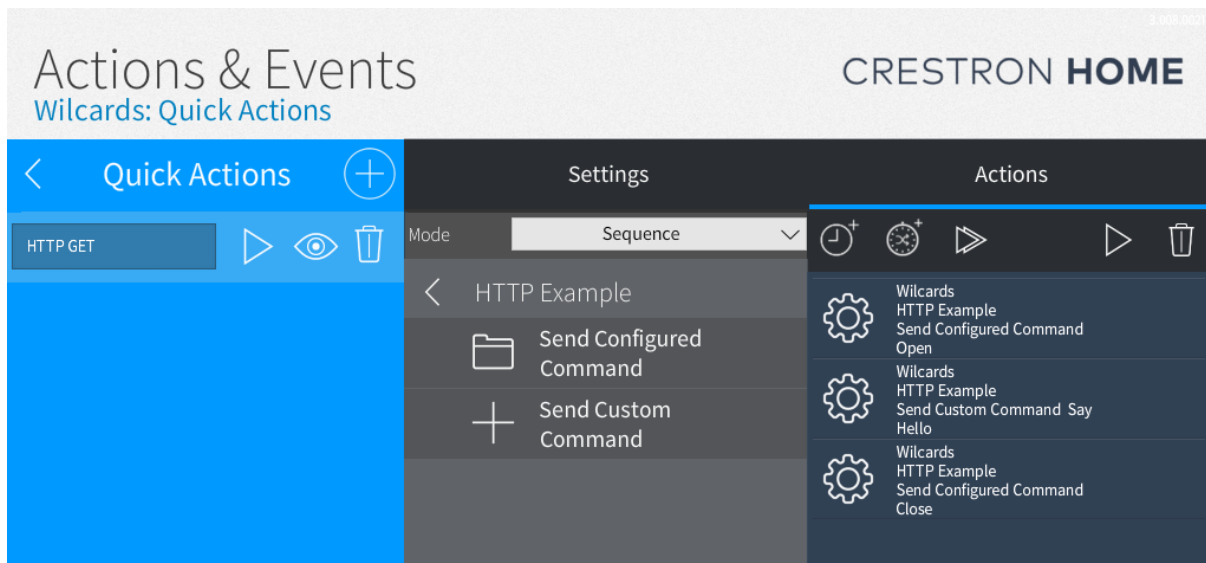
If you wish to use HTTPS for the GET request, select the "Use HTTPS" attribute and you will likely need to override the port to your HTTPS port (generally 443.)

## HTTPS/S POST

As for HTTP/S GET but with an extended syntax to allow definition of POST body data.

## QUICK ACTIONS & SEQUENCES

Each tile also exposes two programmable functions. "Send Configured Command" allows you to recall the pre-programmed strings from the tile or button.



"Send Custom Command" allows you to send any string to the device's transport.

For HTTP/S POST, the Send Custom Command also provides a field for the POST data.

## LISTENER MODULE DESCRIPTION

The listener extension modules do not have any user controllable features.

Add a listener in the normal way to a room.

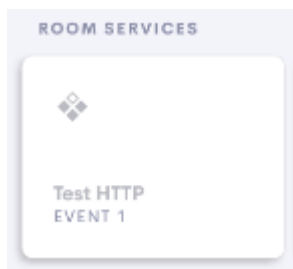
The IP address is not used in this case, so you can enter 0.0.0.0 here.

The port is crucially important, and you should select a port that is not used by any other service on the Crestron Processor. The default ports we have provided should be ok in most circumstances. If you add multiple instances of any of the listener drivers to a Crestron Home system you must use a unique port for each one. This port should be noted down as this is part of the address you will need for whatever service is sending a request.

Enter the licence key as normal, and you should be ready to hook up events.

A tile will be created in the room (so you can put it in a hidden room if you don't want the client to see it) but it will be greyed out as there is no user action.

The status text below the name is useful to see what the LAST event received was. This will show the last processed event ID even if it doesn't have any attached actions or is invalid.



Each listener will provide up to 20 events that can be used to trigger Quick Actions or Sequences within Crestron Home – the events are numbered 1-20 and correspond exactly to the value sent to the listener by the remote service.

Set up events in the normal way within Crestron Home.

## TRIGGERING EVENTS

### Http Listener

The HTTP Listener is triggered by sending a basic HTTP GET request to the address of the processor, with the selected port number, followed by the event number.

e.g. To trigger Event 1:

*http://<proc\_ip>:13257/1*

Any web browser can be used to test this.

If you wish to combine events, you can trigger multiple events using the '\$' character as a separator.

e.g. To trigger Events 2, 5 and 8:

*http://<proc\_ip>:13257/2\$5\$8*

### TCP Listener

The TCP Listener simply receives data on the selected port. We use the '\$' character to terminate a request, so this MUST be used at the end. Carriage returns are ignored.

Simply transmit the data to the listening port over TCP. You can disconnect after each request or leave the connection open. Up to 5 remote endpoints can connect simultaneously to the listener.

e.g. To trigger Event 3, send:

*3\$*

Or to sent multiple events 4, 5 and 6, send:

*4\$5\$6\$*

Event IDs that fall outside of 1-20 will be ignored. Leading zeros should NOT be used.



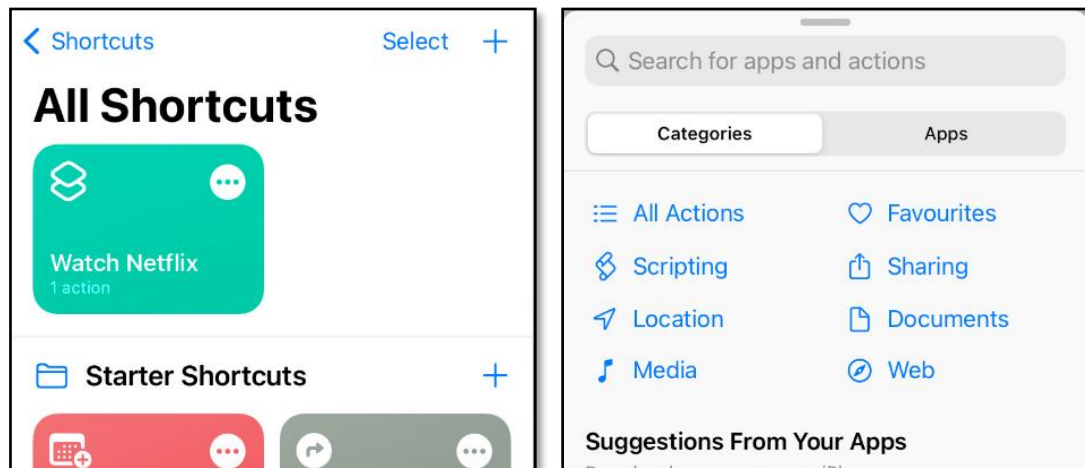
## HTTP Event Listener Example with Siri & Shortcuts

As an example of what can be achieved very simply, we will trigger Crestron Home using Siri.

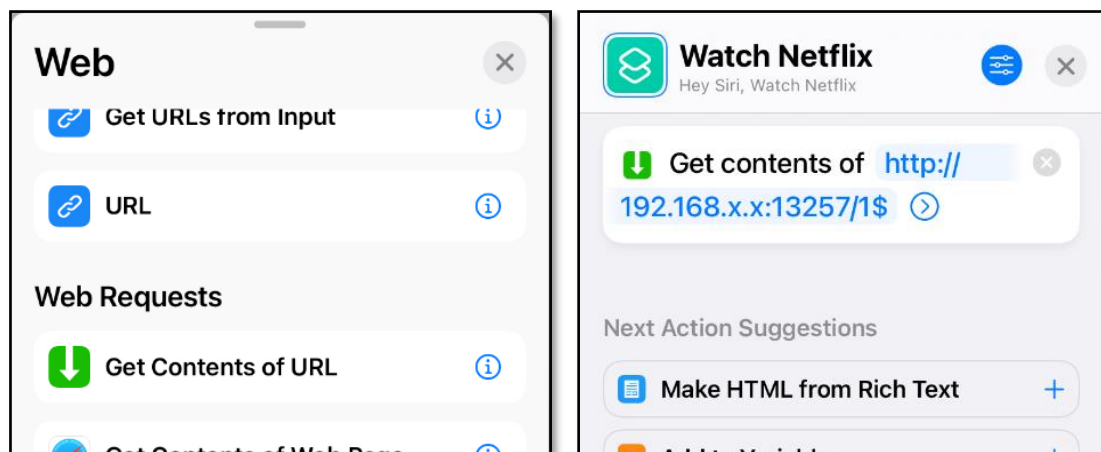
First, set up Crestron Home with the HTTP Listener with an Event to turn on the room and select Netflix (for example). This will be on "Event 1".

Next, using iOS (or MacOS) add a shortcut – which we'll call "Watch Netflix".

The shortcut should then use the "Web" action, under which you will find "Get Contents of URL" (as shown below)



Finally, enter the web request to trigger the Crestron Home event. An example is shown below, but you should use your own processor's IP address and the port used to configure the Listener. The "1\$" denotes "Event 1".



Then simply say "Hey Siri, Watch Netflix" and your phone will send Event 1 to your Crestron Processor.

Simple, but very effective!

## SUPPORT

If you have any issues with a driver or installation please let us know by contacting Ultamation support on [support@ultamation.com](mailto:support@ultamation.com) and please include as much detail about your issue as possible, such a recent processor error log.

Licence verification messages are posted to the error log, so please ensure you have checked this.

## LICENCE

This module (including software, images and any and all other associated assets distributed as part of the purchased download package) is licenced on a PER PROCESSOR basis.

A licence key is generated at the point of purchase and is linked at that time to specific information that MUST be provided at the time of purchase. A purchase should not be completed without correct information as refunds cannot be issued for errors or changes made to details following purchase.

The licence key for each device will be delivered via email along with links to download the module. There is no physical delivery.

The module is provided without any warranty with respect to the reliability of the controlled device or changes to device protocol. We will endeavour, through best efforts, to maintain the module's functionality and any bug fixes will be provided free-of-charge. Additional functionality may be released as a variation of this module and this will be a separate, purchasable, product.