

CCI Crestron Home Bridge Suite Overview

Control Concepts, Inc.

Date: 6/17/2022

Revision: 0

General Overview

The CCI Crestron Home Bridge Suite is made up of multiple SIMPL Windows modules and a Crestron Home driver that when used together will allow you to integrate a SIMPL Windows program with a Crestron Home system.

The suite creates a bridge between the processor running the SIMPL Windows program and the processor running Crestron Home. The SIMPL Windows program will send Crestron Home information about what components have been included in the SIMPL Windows program. At this point, Crestron Home will automatically add these new components to its UI. From that point on, any changes on the Crestron Home side will be sent to SIMPL Windows for handling, and vice versa.

Below is a more detailed description of the various parts of the suite as well as a brief walkthrough on their usage and functionality.

Walkthrough and Usage

SIMPL Windows

The SIMPL Windows modules make up the “server side” of the solution. It is responsible for defining which UI components will be made available in Crestron Home, sending updated control values for these components to Crestron Home and receiving feedback updates when they are updated from the Crestron Home interface.

The module suite is broken into two main parts: command processor and components.

The command processor is responsible for communicating with a single instance of the corresponding Crestron Home driver on a specific IP port.

The individual component modules correspond to specific UI elements available in Crestron Home, such as buttons, toggles, checkboxes, etc. The component modules register themselves to a specific command processor module.

Each command processor will start up a secure TCP server instance and begin listening for a connection from Crestron Home on a specific IP port. Once the corresponding Crestron Home driver has been loaded and configured for this IP port (described later in this document), the command processor module will send the Crestron Home driver all the information it needs to create a corresponding “Tile” and main “Layout” and add all the UI elements that have been registered for the command processor. Once this process has completed, you will have full interoperability between the Crestron Home components and the SIMPL Windows components. For example, pressing a button in Crestron Home will send the button press event

to the corresponding SIMPL Windows component or sending an analog value to a gauge component will send that value to the corresponding gauge in Crestron Home and the UI will update.

A demo program has been included in the module suite package and serves to provide a reference on how this suite might be used in a real life program.

Four folders have been included in the Logic section of the program. Each of these folders has an “Emulated Device” and “Crestron Home” subfolder.



The “Emulated Device” folder consists of some logic to emulate the behavior of a real example device. In a real world program, this might be replaced by a device module controlling a particular device.

The “Crestron Home” folder consists of the corresponding modules from this suite which will be responsible for linking with Crestron Home. Each folder has a single instance of the command processor module and some number of component modules so each one of these folders will create a “Tile”, main “Layout” and add the components onto that “Layout” in the order in which they are defined.

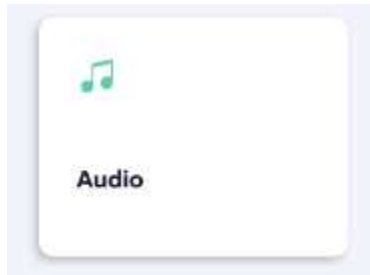
Let’s take, for example, the first subfolder: “Audio”. It is made up of the following components (in addition to the command processor):

- A subheader for the Preset section
- A button group which itself contains two buttons for recalling presets
- A subheader for the Volume section
- A toggle slider which provides a slider and a toggle for adjusting volume
- A text entry component to allow for changing the name of the volume control

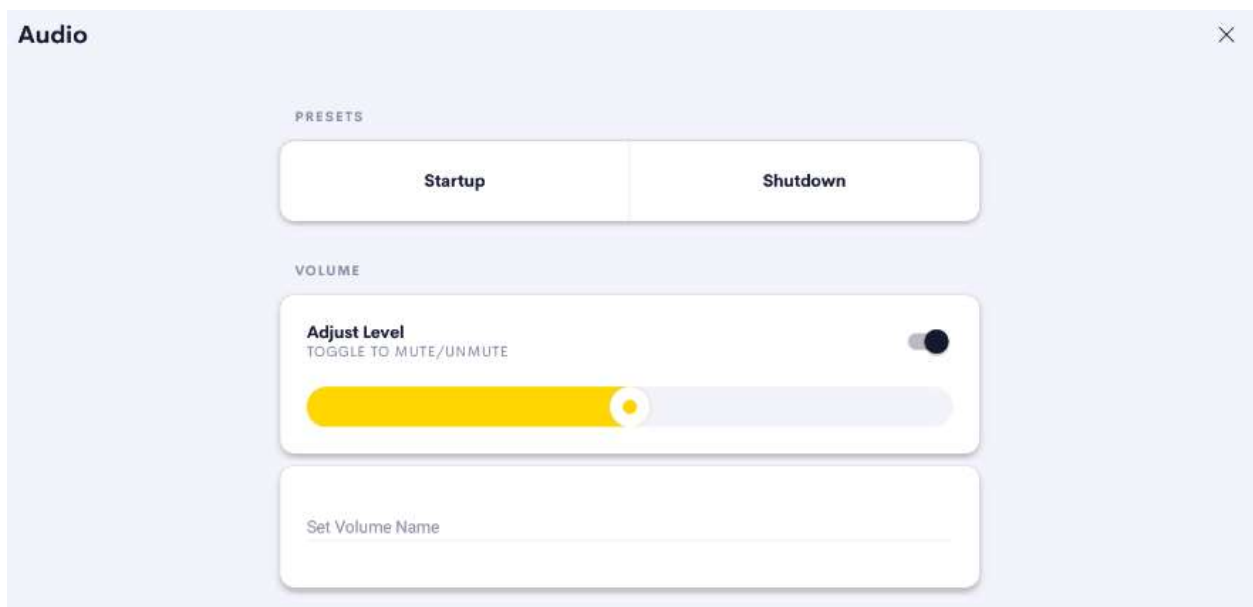


Once the SIMPL Windows program is loaded along with the corresponding Crestron Home driver, the Crestron Home UI will automatically add the following items:

A tile that corresponds to the command processor and includes any information from the command processor module that we've programmed (such as label, icon, status text, etc):



A main layout (that we navigate to by pressing the tile) which contains all the components we've added in our Audio section (as noted above):



Now that the components have been added, interfacing with the Crestron Home UI will send the updated values to SIMPL Windows (such as pressing one of the preset buttons or adjusting the slider). Conversely, setting these values from SIMPL Windows will update the Crestron Home UI. The components are now linked and Crestron Home is effectively acting like another touchpanel in your SIMPL Windows program.

Crestron Home

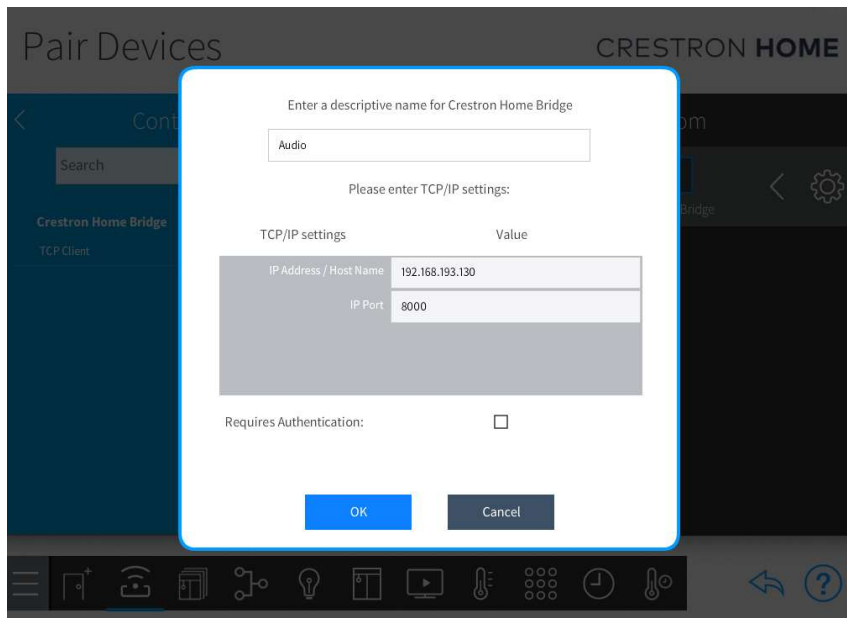
The Crestron Home driver makes up the “client side” of the solution. Instances of the driver are added to rooms in Crestron Home as needed. Each instance of the driver corresponds to a single instance of a command processor module, and therefore, each instance of the driver will create a Tile, Layout and all the components bound to a particular command processor.

In the demo program, as noted above, there are 4 folders, each of which contains a command processor and some number of components bound to each one. As such, if we wanted to match all the functionality defined in the demo program, we would need to add 4 instances of the driver to Crestron Home, each of which would communicate with a different command processor.

The command processor module has a parameter for IP_Port. This is the key differentiator and each instance of the command processor and the Crestron Home driver **MUST** reference a different port. In addition, if you are using multiple programs or multiple processors all communicating with the same Crestron Home processor, you cannot duplicate IP_Ports across **ANY** of these programs. This value must be unique for every single driver instance you are including in Crestron Home.

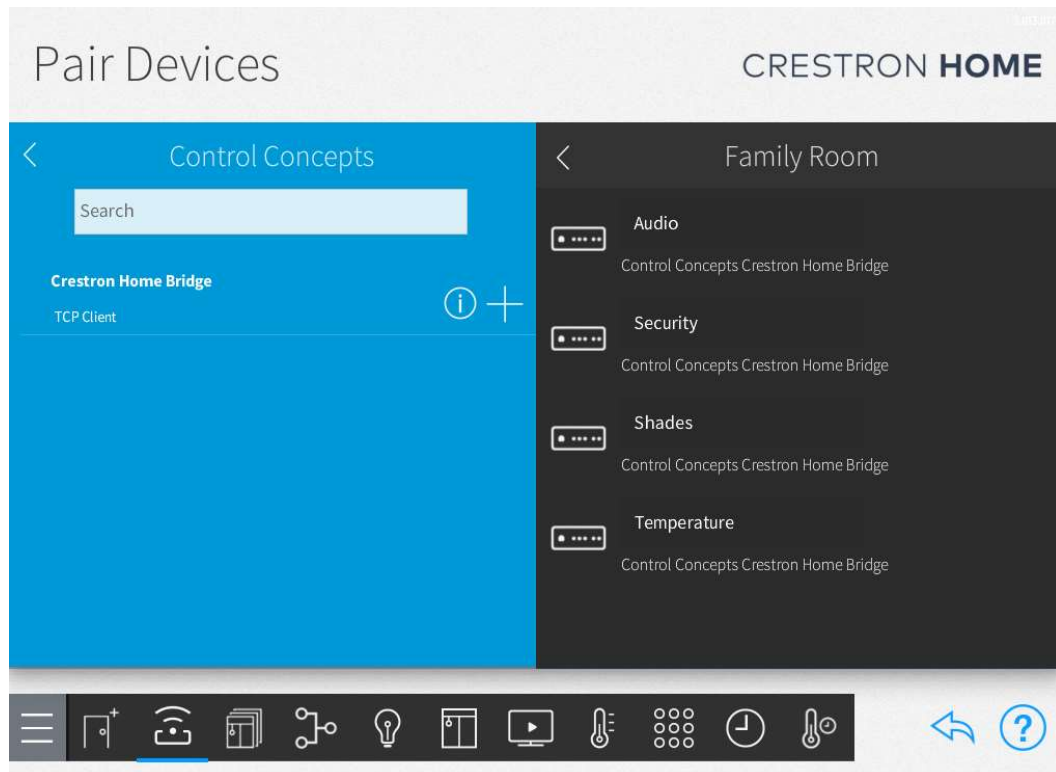
CCI Crestron Home Bridge - Command Processor			
+	Comm	Comm	+
+	Tile	Tile	+
+	Layout	Layout	+
Command_Processor_ID	1d		
IP_Port	8000d		
Tile_Icon	IcAudioOn		
Tile_Secondary_Icon	""		
Tile_Alternate_Icon_Text	""		
Tile_Display_On_Page	Room		
Layout_ID	main		
Layout_Title	Audio		
Layout_Subtitle	""		

As we can see above, the Audio section uses port 8000 (the default starting port). As such, when adding the corresponding driver to Crestron Home, we will give it a name (in this case we'll call it the same thing as the tile: "Audio"), enter the IP address of the processor running the SIMPL Windows program and enter the port to use that corresponds to the IP_Port field on the command processor we want to link to.

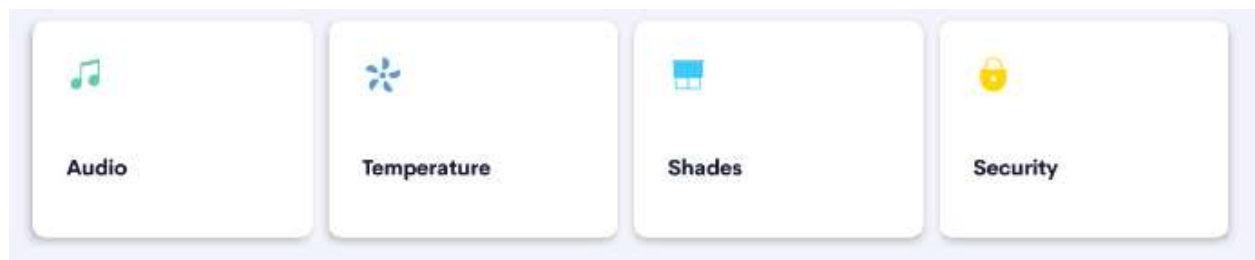


Once this information is entered and we press OK, the driver will attempt a secure TCP connection to the IP address and port defined. If we have already loaded our SIMPL Windows program (which should be done first), the SIMPL Windows command processor will, upon a successful connection from the driver, send the driver all the information it needs to build its UI and add the Tile, Layout and Controls necessary.

After adding in the driver for the Audio section of our demo program, we can then add in the drivers for the Temperature, Shades and Security section also found in the demo. Once we've done that, we will end up with these drivers in Crestron Home:



And these corresponding tiles, each of which will, when pressed, bring up its own layout with its own controls:



It's up to you as the programmer to determine how best to organize your code and create the command processors and components needed to generate the appropriate Tile, Layouts and Controls for your deployment.

Included Components

The module suite comes in two possible flavors: a free/Lite version and a paid/Premium version. Both versions use the same Crestron Home driver and SIMPL Windows command processor. The only difference is that the paid/Premium version includes more components than the free/Lite version, allowing for more customization and specialization of the Crestron Home interface.

Free / Lite Version

- Control Group
- Button Group
- Button
- Text Display
- Toggle Slider
- Toggle Slider Signed
- Subheader
- Checkbox
- Toggle
- Text Entry
- Directional Keypad
- Keypad
- Radial Gauge
- Radial Gauge Signed
- Raise/Lower with Text
- Raise/Lower with Text Signed

Paid/Premium Version

All the components listed above plus:

- Selector Button
- Segmented Slider
- Status and Button
- Status and Navigation
- List Button
- Thermostat
- Alert
- Layouts (Multiple)

Please note that as of this writing the Premium version is still under development and has not been released yet. We expect to complete the development of the Premium components shortly.