SIMPL Home Platform For Crestron Home

Revision: 4.07
Date: 22 October 2026

Supported Installation

This is a Crestron Home driver for Crestron Home processors using a SIMPL module on Crestron Custom processors in addition.

Description

The SIMPL Home suite of drivers provides a unique way to combine the elegant simplicity of the Crestron Home user experience with the power and flexibility of SIMPL Windows and custom Crestron programming.

What started with a simple button panel emulation to bring 3rd party lighting solutions into Crestron Home has grown into a sophisticated collection of devices covering thermostats, keypads, security systems and AV switchers with more device type support being added all the time.

SIMPL Home acts as a proxy for a standard Crestron Home driver, passing the commands and feedback through to a connection point in SIMPL Windows that can then exploit the power and broad product support established across 25 years of custom Crestron development.

COPYRIGHT © 2026 ULTAMATION LIMITED

OS4 onwards – Which platform driver to use

As of revision 3.00, SIMPL Home supports Crestron Home OS4. This involved creating a new platform driver called "SIMPL Home Entities" which supports the new features brought by OS4. Please refer to the following table to know which platform driver to use for which SIMPL endpoint. It may be the case that you need an instance of "SIMPL Home" and instance of "SIMPL Home Entities" to access all the endpoints in one SIMPL program. Multiple platform drivers can connect to one SIMPL program.

SIMPL Endpoints	Compatible Crestron Home Platform Driver
Basic Tile	SIMPL Home
Advanced Tile	SIMPL Home
Keypad	SIMPL Home
Thermostat	SIMPL Home
Scripted	SIMPL Home
Modular	SIMPL Home
Integrated Security	SIMPL Home
Integrated Display	SIMPL Home
Source	SIMPL Home
AV Distribution	SIMPL Home
AV Receiver	SIMPL Home
Security	SIMPL Home
Light	SIMPL Home Entities
Scene & Scene Manager	SIMPL Home Entities
Entry (Garage Door/Gate)	SIMPL Home Entities
Lock	SIMPL Home Entities
NVX MultiView Switcher	SIMPL Home
NVX MultiView Layout	SIMPL Home

COPYRIGHT © 2026 ULTAMATION LIMITED

Some things SIMPL Home CANNOT do…

It's worth pointing out that SIMPL Home can only bridge elements of Crestron Home than provide hooks we can take hold of. As Crestron expose features, we will always look to exploit them via SIMPL Home, but there are several things that simply cannot be done. We'll list a few of the more common use cases below:

- Can SIMPL Home make thermostats appear in Crestron Home the same way as native HVAC?
  No – Crestron have yet to provide an HVAC driver type.
- Can SIMPL Home make shades appear in Crestron Home the same way as native shading?
  No – Crestron have yet to provide a shading driver type.
- Can SIMPL Home be used to integrate 3rd party UIs or controllers or provide Crestron Home control from a traditional custom Crestron system?
  No – Drivers, such as the endpoints exposed through SIMPL Home sit behind the UI of Crestron Home. With the limited exception of Extension devices, we have no control over the UI of Crestron Home.
- Can SIMPL Home drive the Crestron Home UI (e.g. switch to the cameras page on a security event)?
  No – 3rd party drivers have no control over the Crestron Home user experience.

COPYRIGHT © 2026 ULTAMATION LIMITED

CONTENTS

Supported Installation 1
Description 1
OS4 onwards – Which platform driver to use 2
Some things SIMPL Home CANNOT do 3
Contents 4
Programming & Configuration Notes 8
Setup the SIMPL Module – Host 8
Note on Versioning 8
Basic Tile Example 10
User Interface Object Signals 11
Slider Toggles 12
Action Buttons 12
Value Sliders 13
Advanced Tile Example 14
Keypad Tile Example 16
Thermostat Tile Example 17
Usage Notes 19
Scripted Tile Example 20
Modular Tile Example 23
UI Components 25
Integrated Security Example 30
Security Area 30
Keypad Emulation 32
Security Zone 34
Integrated Display Example 35
Signals: 35
Source Example 37
Parameters 38

COPYRIGHT © 2026 ULTAMATION LIMITED

Outputs 44
Inputs 48
AV Distribution Example 50
Parameters 50
Inputs 50
Outputs 50
AV Receiver Example 51
Parameters 52
Outputs 58
Inputs 62
Lighting Example 64
Parameters 64
Inputs 65
Outputs 65
Scene Manager Example 67
Parameters 67
Inputs 67
Output 67
Entry Example 68
Parameters 68
Inputs 68
Output 68
Lock Example 69
Parameters 69
Inputs 69
Outputs 71
Settings 71
NVX MultiView Example 73
Recommended Prerequisites 74

COPYRIGHT © 2026 ULTAMATION LIMITED

Setup in SIMPL 75
Setup in Crestron Home 79
Crestron Console Diagnostic Commands 82
Version Checking 83
Installation 84
Adding the Driver 84
Adding Devices 86
Creating Translation Files 87
Quick Actions & Sequences 88
Trigger Action 88
Set Value 88
Slider Toggle 88
Toggle Button 88
Trigger Tile 88
Events 89
Scripted Tile Programming Guide 90
SIMPL Home Scripted 93
SIMPL Home Scripted Digital Property 93
SIMPL Home Scripted Analog Property 93
SIMPL Home Scripted String Property 94
SIMPL Home Scripted List Property 94
SIMPL Home Command 95
SIMPL Home Scripted Action 95
SIMPL Home Scripted Event 96
SIMPL Home Scripted Object List Property 96
Object List Properties 97
Statically Declared List Items 97
Support 99
Licensing 99

COPYRIGHT © 2026 ULTAMATION LIMITED

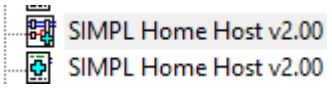
Cloud Licence.....100

COPYRIGHT © 2026 ULTAMATION LIMITED

Programming & Configuration Notes

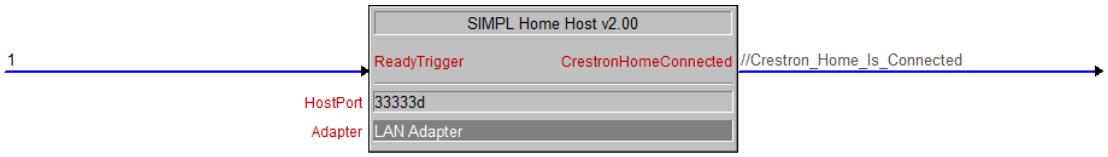
Setup the SIMPL Module – Host

Please ensure that you add the SIMPL Windows SIMPL Home Host module (not the SIMPL+ module of the same name.)



✓ Correct
✗ Incorrect

This wrapper enables the use of certain console commands, detailed later in this document.



Configure the parameters of the SIMPL Host Module as follows:

ServerPort – Enter the server port, this must be an available port and match with the port entered on the Crestron Home side. The default of "33333" will normally be acceptable.

Adapter – Leave the adapter as default unless something specific is being utilised.

The signals are used as follows:

ReadyTrigger – This signal MUST by pulsed high when the control system is ready to accept connections from the Crestron Home system. A '1' on this signal will is acceptable, but you can defer initialisation in order to manage load in heavily loaded systems.

CrestronHomeConnected – This signal indicates that Crestron Home client is currently connected.

Only one instance of the host module should be used in a program.

Note on Versioning

Following the release of version 2.00, the module is not backwards compatible with differing major releases. Both the SIMPL Home Host and Crestron Home ends will

COPYRIGHT © 2026 ULTAMATION LIMITED

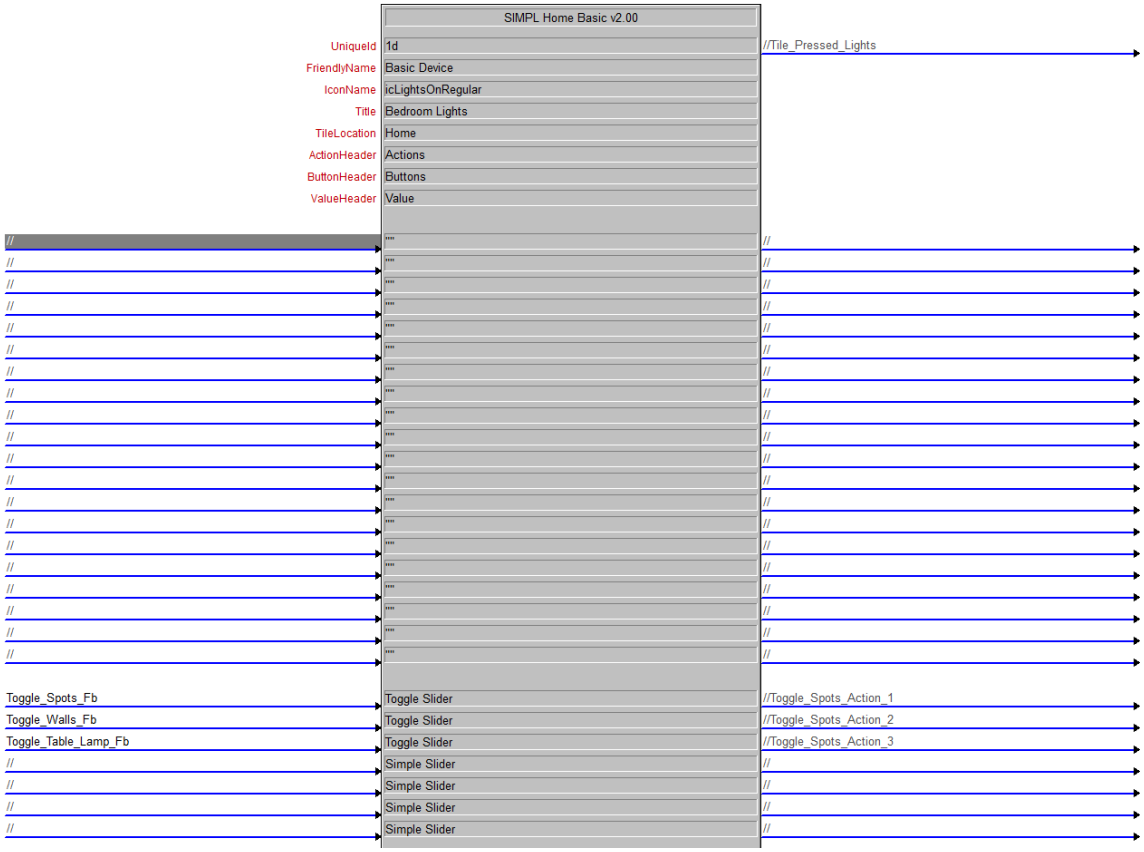
reject communication with any major version that does not match their own. The console will throw warnings on both sides for differing minor versions, but will function as normal.

The following sections describe the individual endpoints and their parameters/signals. Some parameters (marked with ǂ) are common to all endpoints, so please refer to the Basic Tile description for these generic parameters – they will not be repeated on subsequent endpoint descriptions.

COPYRIGHT © 2026 ULTAMATION LIMITED

Basic Tile Example

The "Basic" endpoint creates a configurable tile in the Crestron Home client. This can be used for various purposes but is often a good candidate for 3rd party lighting solutions where buttons can be used to recall preset scenes or buttons and sliders are used to switch or dim loads.



For a basic device, the parameters to setup are as follows:

UniqueId ǂ – This is an integer which uniquely identifies the keypad within the solution.

FriendlyName ǂ – This is the name which will appear in Crestron Home platform folder when the platform driver connects to the SIMPL Windows instance.

IconName – This allows the programmer to specify an icon for the tile within Crestron Home.

Only Crestron Supported Icons are available, a list of available icons can be found at:
https://sdkcon78221.crestron.com/sdk/Crestron_Certified_Drivers_SDK/Content/Extension-Device-Icons.pdf

COPYRIGHT © 2026 ULTAMATION LIMITED

Title – This title will appear as the header for the Tile when it is expanded.

TileLocation ǂ – This will set the location of the tile. You can choose to display the tile in the home page, room page or both. **Important note**: If you set the tile to Home, it must include features in navigation. Home tiles, cannot just be button presses.

***Header** – This will appear as the header for the section in question and is displayed in the tile. These headers are present for Buttons, Actions and Values. If you leave a header blank (i.e ""), any items present in the field subgroup will be grouped more closely with the previous field.

For instructions on creating translation files for this tile, see Creating Translation Files below.

User Interface Object Signals

Toggle Buttons

Toggle buttons are displayed with a label and a toggle state in the Crestron Home UI and can be used when a digital signal control is required – such as "On/Off" for a lighting load, or "Open/Close" for a gate. Up to 20 buttons can be defined.

ButtonLabel[-] Parameter – This is the label for the button. If left blank (""), the button will not be created.

ButtonFb[-] Digital Input - This is used to set the current state for the button – edge triggered, high, or low.

ButtonPress[-] Digital Output – This is the current output state of the button – latched high or low. NOTE: The name of this signal is misleading – it is LATCHED, not PULSED.

COPYRIGHT © 2026 ULTAMATION LIMITED

Slider Toggles

The next set of signals/parameters relate to the analogue sliders. Each slider can be represented as a Simple Slider or as a Toggle Slider (a yellow slider with a toggle button, suitable for lighting load control). Up to 20 sliders can be defined.

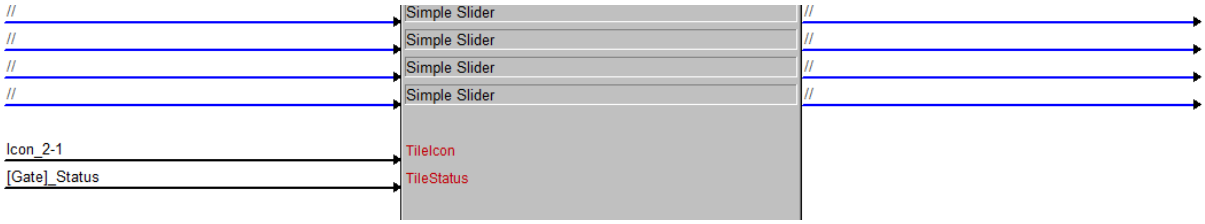
ToggleSelect[-] Parameter – This drop down allows you to specify the style for the slider. If Simple Slider is selected, the ToggleFb[-] and ToggleState[-] signals have no effect.

ToggleFb[-] Digital Input – As for Toggle Buttons ButtonFb[-].

ToggleState[-] Digital Output – As for Toggle Buttons ButtonPress[-].

TileIcon – Allows the user to input feedback for the tile icon, which can be used to update the icon when an action is performed, or a value changed. The serial value passed in must be a valid icon identifier (as given in the Crestron Icon document linked above).

TileStatus – enables status feedback to be passed to the tile. In the example below, the mode that the gate is in is passed to the tile and displayed accordingly.



Action Buttons

Action buttons are single press actions. Crestron Home does not currently support Press and Hold models. Each button can have an icon, label and trigger an event in SIMPL Windows. Up to 20 action buttons can be defined.

ActionLabel[-] Parameter - This is the label for the button. If left blank (“”), the button will not be created.

ActionIcon[-] Serial Input – Passing in a correctly formatted icon identifier will display the icon on the button. For example “icShadesOpen”.

ActionPress[-] Digital Output – A pulsed signal indicating the button has been pressed. NOTE: Press and Hold is NOT POSSIBLE in Crestron Home.

COPYRIGHT © 2026 ULTAMATION LIMITED

Value Sliders

Sliders of two styles can be defined. The style is controlled by the ToggleSelect parameter mentioned above. Up to 20 sliders can be defined.

ValueLabel[-] Parameter - This is the label for the slider. If left blank (""), the slider will not be created.

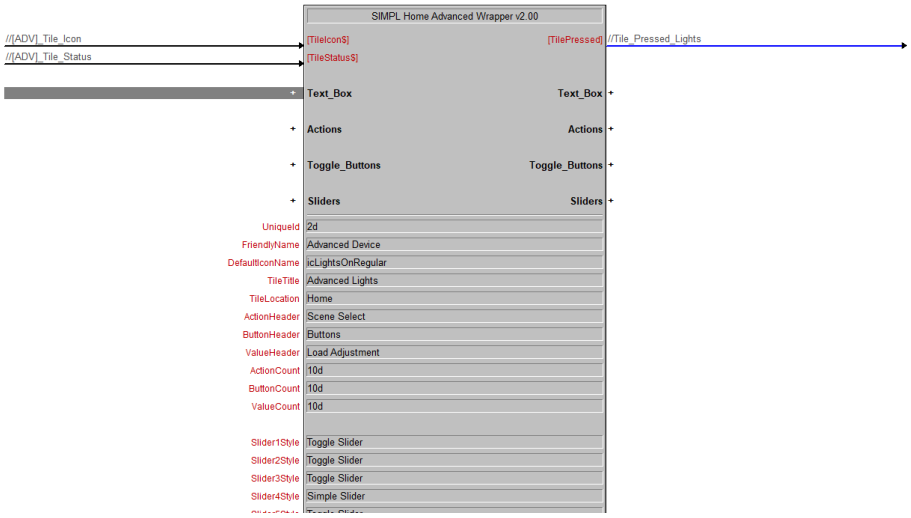
ValueFb[-] Analogue Input - This is used to set the current value for the slider (full scale 0-65535).

ValueSet[-] Analogue Output – This is the current value of the slider (full scale 0-65535).

COPYRIGHT © 2026 ULTAMATION LIMITED

Advanced Tile Example

The advanced keypad (UMC Wrapper) follows the same outline as the basic keypad above, with a few additions. For example, the addition of integer inputs for the number of actions, buttons and values needed. Shown below:

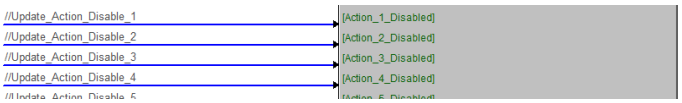


Unlike the Basic tile, these values are used to setup the number of UI elements. Meaning you aren't required to fill in the labels for each individually initially. Rather, these labels can be fed in via serial inputs, making the label visibility dynamic. If you pass in an empty string to the label, the button or slider will disappear. Pass in a non-empty value and the button/slider will reappear. The image below shows where to pass in this label for the "Action button" elements.



If you set all of these values to 0, this will disable the UI within the tile, creating a tile action with no internal detail.

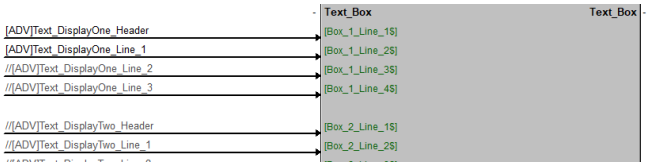
In addition, the Advanced tile provides the facility to disable and reenabled buttons and sliders using digital inputs.



Another addition to the Advanced tile is the ability to send text up to three text boxes displayed within the tile. The text box visibility is set by text within the display lines. Any text will cause the visibility to be enabled and sending a blank string will disable it. The headers do not affect the visibility. **Important note**: Navigation will

COPYRIGHT © 2026 ULTAMATION LIMITED

not be displayed and therefore the text boxes unless there are controls within the tile. If you want just a blank text box, use modular.

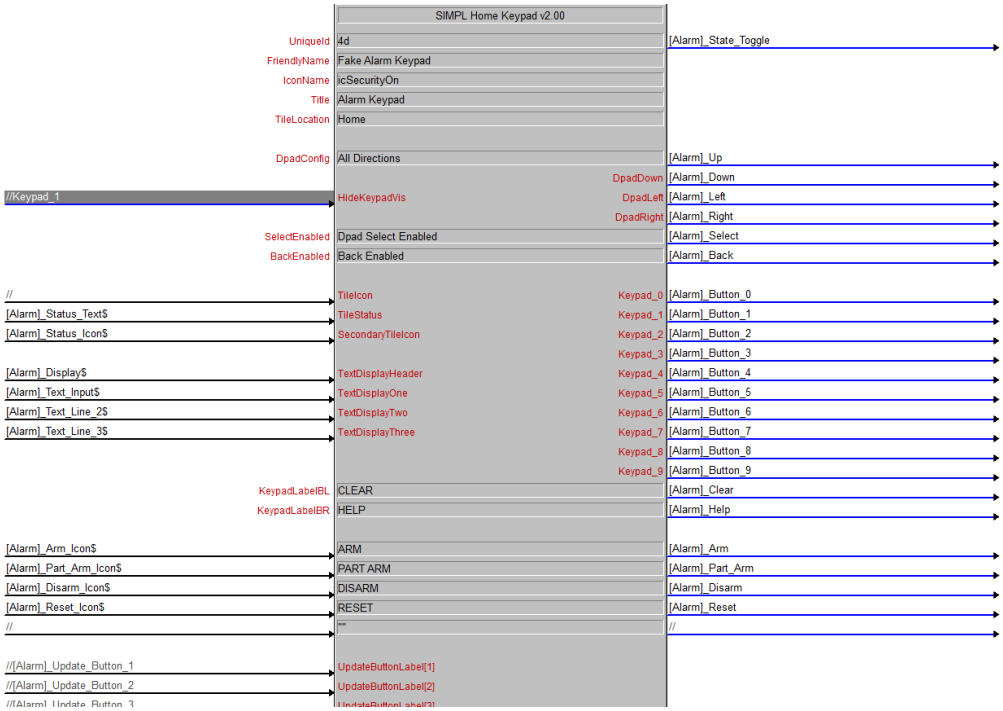


COPYRIGHT © 2026 ULTAMATION LIMITED

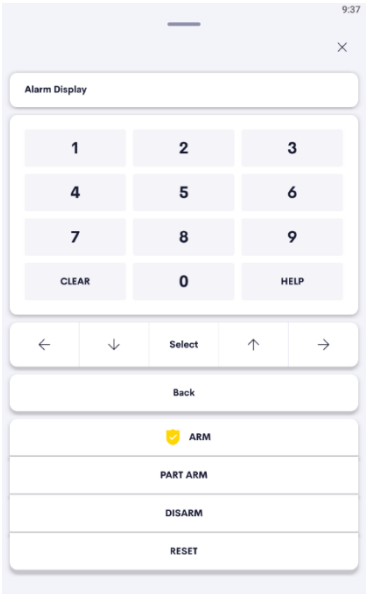
Keypad Tile Example

With the keypad device you can setup a standard keypad plus 2 optional actions, navigation buttons, an updateable text area and up to 5 function buttons.

The tile itself can also act as a trigger, just like the Basic/Advanced buttons.



The results can present a simple security alarm style keypad such as shown in the image on the right.



This example also shows the directional control, A simple text display area and 4 function buttons with dynamic icons.

DpadConfig Parameter – This dropdown allows you to select No direction pad, All directions, or Up and Down only.

SelectEnabled Parameter – Enables the optional "SELECT" button in the centre of the navigation control.

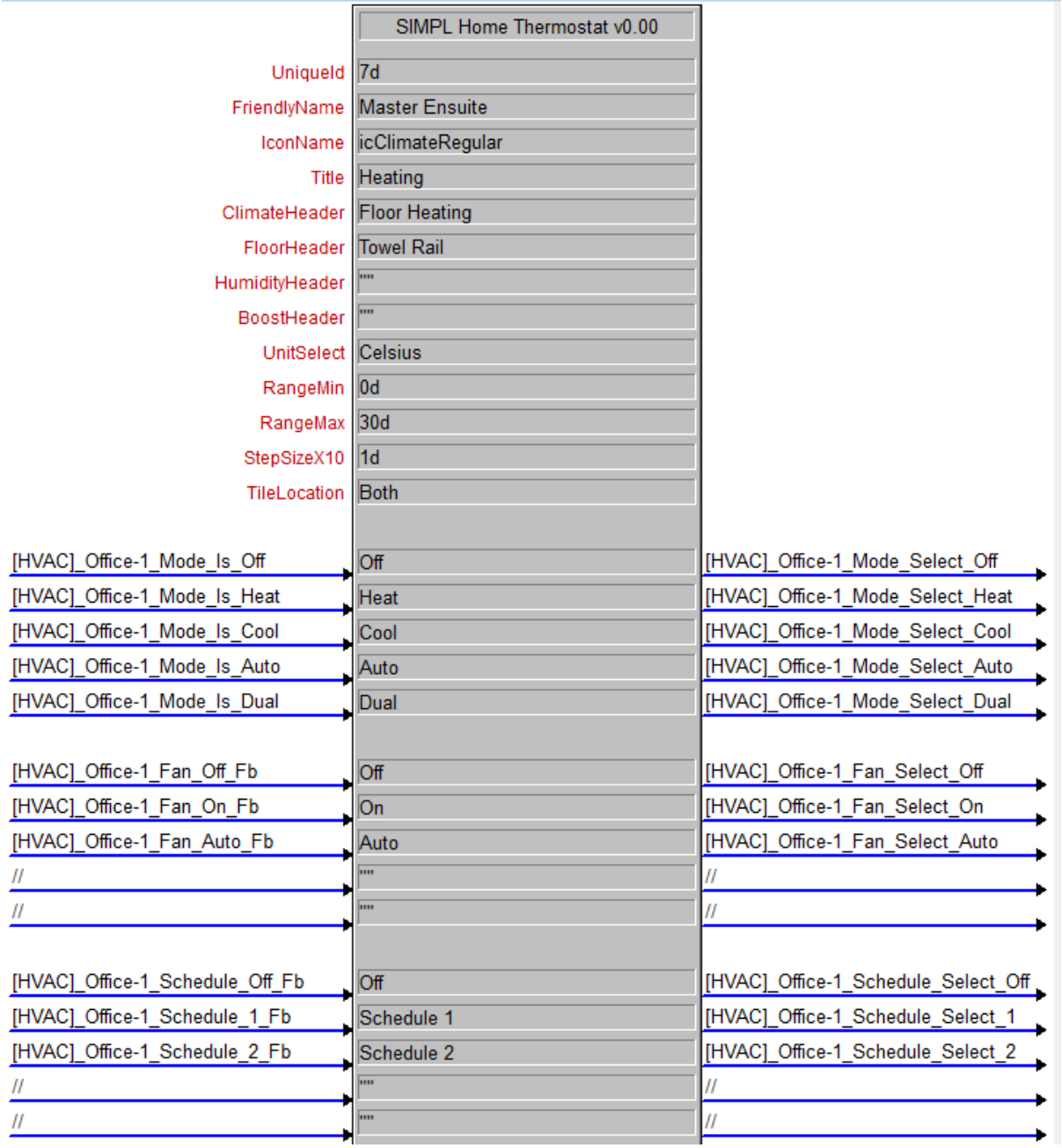
BackEnabled Parameter – Enables the optional "Back" button below the navigation control.

For instructions on creating translation files for this tile, see Creating Translation Files below.

COPYRIGHT © 2026 ULTAMATION LIMITED

Thermostat Tile Example

With the Thermostat endpoint, you can integrate any 3rd party thermostat with Crestron Home, through SIMPL Windows.

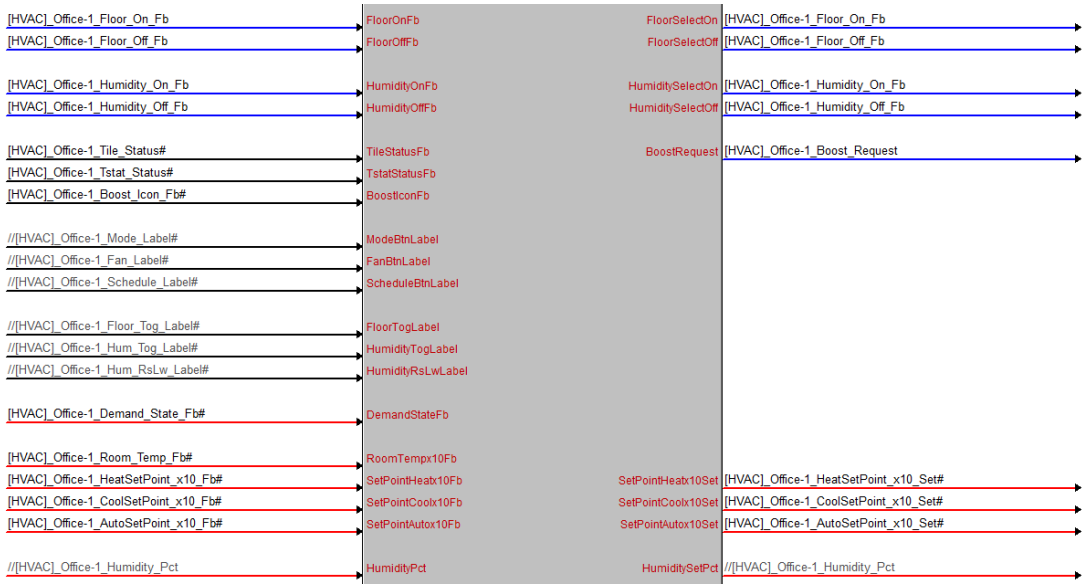


The thermostat module allows for 5 operating modes, 5 fan and 5 schedule settings to be selected. The configuration is determined at SIMPL Windows design time by defining labels for valid options and blank strings any option that should not be presented. The example program shown above has 5 modes, 3 fan and 3 schedule options. It is important to note that these are order specific to an extent. For example, the “off” state for all of them is the first in the list. For modes, heat,

COPYRIGHT © 2026 ULTAMATION LIMITED

cool, auto and dual are all in a specific order. So, care must be taken when setting up these.

Additionally, there are string inputs for dynamic status feedback on both the top level tile and the status text within the thermostat. You can also change the icon of the Boost function to indicate the current boost state.



For Demand_State_Fb you can input a value of 0, 1 or 2 to display a (1) demand heating icon or a (2) demand cooling icon respectively. Passing 0 or any other value sent through the analogue input will disable the icon.

Further, there are signals for supplementary controls such as floor heating and humidity. These can be disabled by leaving their respective header properties blank (“”).

All analogue temperature values (other than the temperature range limit parameters) are expressed as the decimal value x10. i.e. a temperature of 22.2 degrees should be expressed as “222”. This allows SIMPL Windows to work with decimal values. This has particular importance given the notes below. The StepSize parameter allows you to decide how much the controls in Crestron Home increment/decrement the values on one press. This value should be expressed as the decimal value x10.

For convenience, you can now rename the Labels for the Mode, Fan and Schedule buttons as well as the Floor Heating and Humidity controls. This can be done via the corresponding serial input.

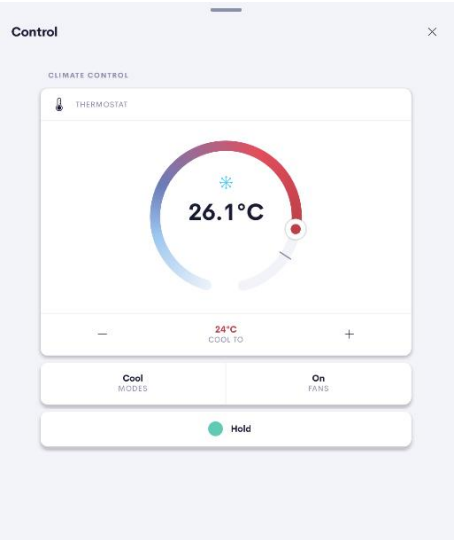
COPYRIGHT © 2026 ULTAMATION LIMITED

For instructions on creating translation files for this tile, see Creating Translation Files below.

Usage Notes

To avoid issues with controls being hidden in Crestron Home you may see slight changes to Setpoints to avoid them going out of range. The cool setpoint needs to be higher than the heat setpoint for “Dual” mode so this will now enforce and update to keep that true.

An example is shown here with the heat modes and fans on or off. As well as an action for holding the thermostat temperature.



All temperature values sent to to Crestron Home must be scaled up by ten. This is to preserve the decimal place when the value is converted to and from analogue.

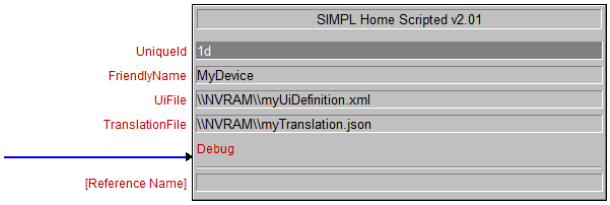
COPYRIGHT © 2026 ULTAMATION LIMITED

Scripted Tile Example

The full power of Crestron Home extension tiles has been released with the "Scripted Tile" object. This is a complicated subject and as such, the feature is fully documented in its own section of this document.

In summary though, Scripted Tiles allow the SIMPL programmer complete freedom to define a Crestron Extension device from the XML UI Definition and JSON translation files through to SIMPL objects representing each individual property so you can drive every aspect of the extension devices. We also support Crestron's rather opaque Object List model providing mechanisms for both static and dynamic lists, though the latter are generally only practical when used from SIMPLSharpPro.

An extension device is defined at its root by a SIMPL Home Scripted module.



The UniqueId is, as for all other endpoints, the primary identifier for the extension tile, and all child modules must refer back to this value.

UiFile – a file path to a user created UI XML definition. This file MUST adhere to Crestron's XML definition DTD. See the detailed documentation for more information.

TranslationFile – a file path to a user created JSON translation file. This file MUST adhere to Crestron's JSON structure.

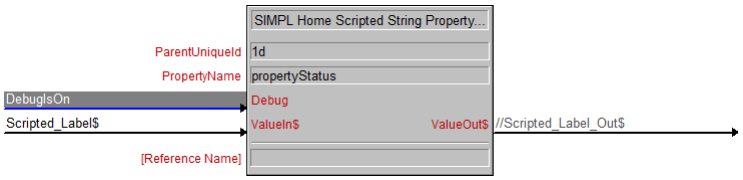
To be useful, the extension device will also require properties. There are various property types (Boolean, 16-bit Unsigned Integer and String) and these are used to drive the properties defined in your XML UI file. **The XML file and Translation file must be uploaded to the SIMPL processor.**

COPYRIGHT © 2026 ULTAMATION LIMITED

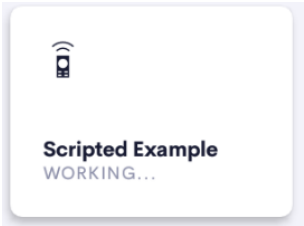
An illustration of the start of the XML file is shown below.

```
<?xml version="1.0" encoding="UTF-8" ?>
<uidefinition xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
              xsi:noNamespaceSchemaLocation="http://prd-use-rad-
assets.azurewebsites.net/ExtensionsSchemaDefinition.xsd">
  <version ver="2.0" />
  <tile icon="#icRemoteRegular" showinhomepage="#false" showinroompage="#true"
status="{propertyStatus}" navigation="show:sample" />
  <layouts>
    <layout id="sample" isdefaultlayout="#true" title="^labelHeader" >
... and so on ...
```

An example of a string property for the tile "Status" text is given below. Note the ParentUniqueId refers back to the main tile UniqueId parameter.



Changes made by SIMPL to the ValueIn$ input will be reflected in the tile, as in this example where "Scripted_Label$" has been set to "Working..."



Scripted Tiles support the following child modules:

- Scripted Digital Property – for Boolean (True/False) properties
- Scripted Analog Property (Unsigned, 0-65535) – for value properties
- Scripted String Property – for string properties
- Scripted List Property – for string lists (used with segmented sliders and selector buttons)
- Scripted Command – for button commands (with support for an optional argument)
- Scripted Action – for button actions (with support for an optional argument)
- Scripted Event – up to 10 events can be defined in Crestron Home
- Scripted Object List Property – used to define object lists for button lists
  Digital, Analog, String and List Properties can all be defined as children to the Object List Property. This defines the "class" of object, and new items can be added and removed dynamically from the list.
  This is rather awkward in SIMPL's static environment so we have also

COPYRIGHT © 2026 ULTAMATION LIMITED

provided a way to statically define list items at design time, along with up to 5 properties for each object instance.

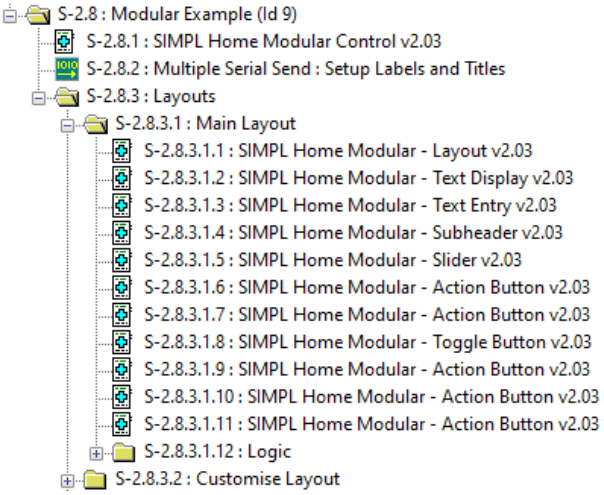
This covers nearly everything that is possible in Crestron Home extensions, without limitation or compromise.

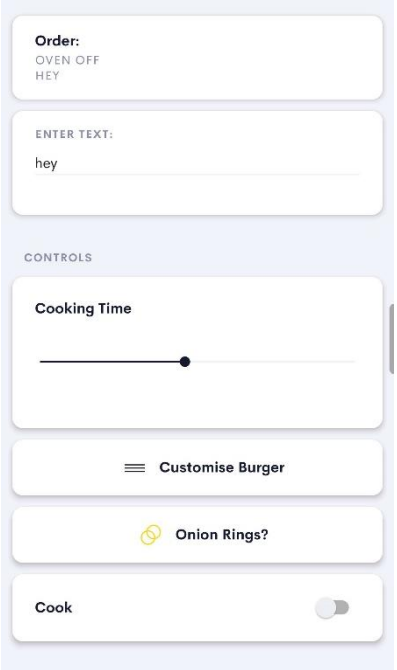
Please refer to the section “Scripted Tile Programming Guide” for detailed information.

COPYRIGHT © 2026 ULTAMATION LIMITED

Modular Tile Example

Modular tile gives you the creative blocks needed to bring your integration ideas to life. Like the Scripted Tile, the Modular tile allows you to create your own custom UI tile in Crestron Home. However, compared to the Scripted Tile, there's a lot less faff. Simply select a component that you would like and drag it into position in the Simpl Program.



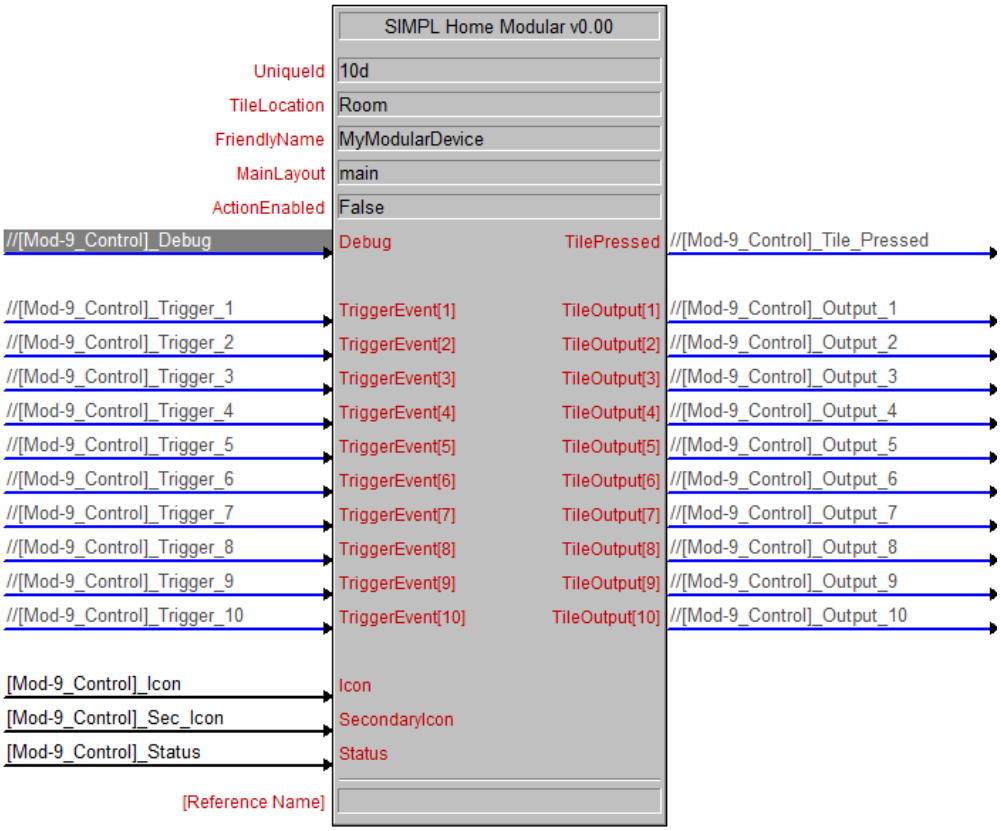


If you keep all the Ui Components in the same sub folder as a layout, they will be organised in the order that they are listed in Simpl Windows. As you can see in the image above. This gives you a large amount of flexibility in what components you'd like to use and how. It also makes it relatively straightforward to setup a component. Just drag and drop into position!

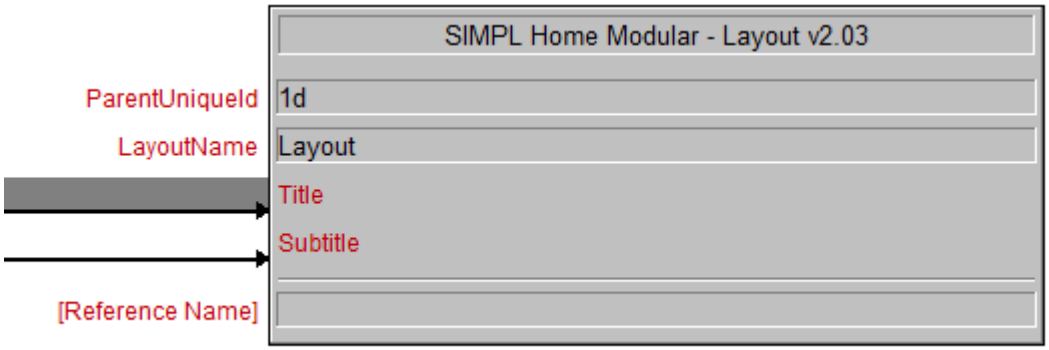
The tile itself is created from one matching Control Tile. The control tile, shown below, has the controls for the tile such as Status, Tile actions and Tile Icons as well as the Unique Id necessary to tie all sub Ui Components together. You can also set the Tile Location, however like advanced tile, you cannot have a blank tile in the Home page. So, keep that in mind when deciding on the location. The tile has a

COPYRIGHT © 2026 ULTAMATION LIMITED

parameter called ActionEnabled that when false enables the tile to go straight to the layout when pressed. When true the tile does the action when pressed.



Each Layout has a matching ParentUniqueId, this must correspond to a control tile Unique Id, or the layout and its subcomponents will not be utilised.



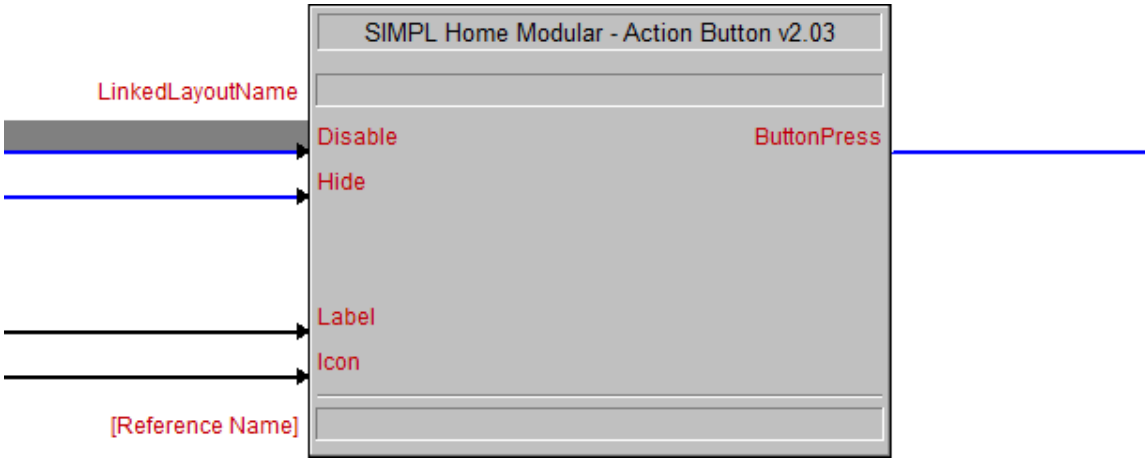
A layout is very simple to setup, you simply need to enter a name and provide a ParentUniqueId as described. The layout name will be used for navigation, so it won't be visible to the end user, but you will need it to create action buttons which affect navigation. There will be one layout which is shown by default on a tile press. This MainLayout ID must correspond to a layout, or the tile will not show anything.

COPYRIGHT © 2026 ULTAMATION LIMITED

For instructions on creating translation files for this tile, see Creating Translation Files below.

UI Components

The example shown below is for the action button. In this case, the action button component is automatically matched with the layout tile in the same Subfolder. This means you don't need to worry about inputting a consistent module comment, but you do need to consider the layout order in your Simpl Windows



program.

The action button parameter, LinkedLayoutName, allows you to specify navigation parameters for a button. This can be left blank. However, if you choose to enter a matching layout name, that layout will be displayed when the button is used in Crestron Home.

Most of the UI components have the following common signals:

Disable – When set high, it will disable the control feature. When sent low, it will enable the component control.

Hide – When set high, it will hide the UI component in question. When sent low, it will reveal the component.

Label – This string input can perform slightly different functions depending upon the component in question, however it usually gives the component a named label.

COPYRIGHT © 2026 ULTAMATION LIMITED

Icon - This string input allows the user to set a standard Crestron icon. Where the icon is displayed will differ depending on component. The supported icon keys can be found here: https://sdkcon78221.crestron.com/sdk/Crestron_Certified_Drivers_SDK/Content/Topics/Driver-SDK-V1/Create-a-Driver/Device-Types/Extensions/UI-Files.htm

Modular Tiles currently support the following UI components. Notes are also provided here on their unique inputs, outputs, and parameters.

- Action Button
  - ButtonPress – pulses when the button is pressed.
- Checkbox
  - ToggledFb – Setting this high will check the checkbox. Setting it low will uncheck the checkbox.
  - Toggled – This goes high when the checkbox is checked and low when the checkbox is unchecked.
- Status Button
  - ButtonPress – pulses when the button is pressed.
- Text Display
- Text Entry
- Toggle Button
  - ToggledFb – Setting this high will switch on the toggle. Setting it low will switch the toggle off.
  - Toggled – This goes high when the toggle is on and low when the toggle is off.
- Radial Gauge
  - ColourValueFb – Dark Grey = 1, Mid Grey = 2, Blue = 3, Light Blue = 4, Red = 5, Green = 6, Orange = 7, Purple = 8, Pink = 9, Yellow = 10
- Raise and Lower
  - Format – This signal should be a string containing the string "%s". This is how you control the display format of the value of the component, where %s is the value. For example, if you wanted a the value to show 50% when the value is 50, you would set this signal to "%s %". If you wanted it to display "50 Seconds" you would set this signal to "%s Seconds".
  - MinValue, MaxValue and StepSize define the boundaries and step size of the value of the component. For example, if you wanted a

COPYRIGHT © 2026 ULTAMATION LIMITED

percentage value where pressing the raise lower incremented/decremented the value by 1%, you would have a MinValue of 0d, a MaxValue of 100d, and a StepSize of 1d.

- Segmented Slider
    - SegSliderValueFb and SegSliderValueSet have ranges of 1 to 5.
- Slider
    - MinValue, MaxValue, and StepSize define the bounds and step size of the value of the component. They work similarly to the parameters of the same name in the Raise and Lower module, described above.
- Toggle Slider
    - ToggledFb – Setting this high will switch on the toggle. Setting it low will switch the toggle off.
    - Toggled – This goes high when the toggle is on and low when the toggle is off.
- Sub headers
- Thermostat
    - The MinValue, MaxValue, and StepValue parameters apply to the current temperature value, and the three set points. The heat point's max is set to MaxValue-1, and the cool set point's min is set to MinValue +1. This is necessary for the dual mode to work.
    - Ensure that the values of the current temperature and set points are within their respective ranges.
    - To use the dual mode, the heat set point must be strictly less than the cool set point.
    - All temperature values must be scaled up by 10. This allows for fractional values.
- Button Group
    - Button groups must be placed in a folder inside a layout folder. You can place up to 5 of either Action Buttons or Selector Buttons and they will appear as a singular button group in Crestron Home
- Control Group
    - Control Groups work similarly to the button groups above. They must be placed in a folder inside a layout folder, and you can place UI components in the folder to make them appear joined in Crestron Home.
- Selector Button

COPYRIGHT © 2026 ULTAMATION LIMITED

    - AvailableValue –There are 3 different versions of the Selector Button module. The difference between these is the number of AvailableValues you can have. For the purpose of reducing processor resource use, please use the version of the module which has the least amount of AvailableValues that you require. If you need more than 20, contact Ultamation and we can provide you with a modified module. Each string entered into an AvailableValue field will show in the list to select from in Crestron Home.
    - SelectorButtonValueSet – equates to the index of the item currently selected, where 0d is unselected and 1d is the first item.
- List Button
    - AvailableValue – Similar to Selector Button, there are 3 versions of the module. If you require more than 20 values in the list, please contact Ultamation.
    - ButtonPress – High when the corresponding list item is pressed.
- Keypad
    - FunctionOneLabel/FunctionTwoLabel – if input, the corresponding function button will show on the keypad with the input label. If this is left empty, then the function buttons will not appear, leaving only the number buttons.
- Directional Keypad
- Status and Navigation
    - Status – The secondary label, appearing under the main label

Information on components and what they will look like within Crestron Home can be found at:

https://sdkcon78221.crestron.com/sdk/Crestron_Certified_Drivers_SDK/Content/Topics/Driver-SDK-V1/Create-a-Driver/Device-Types/Extensions/UI-Files.htm

The endpoint examples shown so far have all been “Extension devices” within Crestron Home. These are user interface objects that can be used to reflect useful information and controls, but Crestron Home itself is not “aware” of what the device under control actually is.

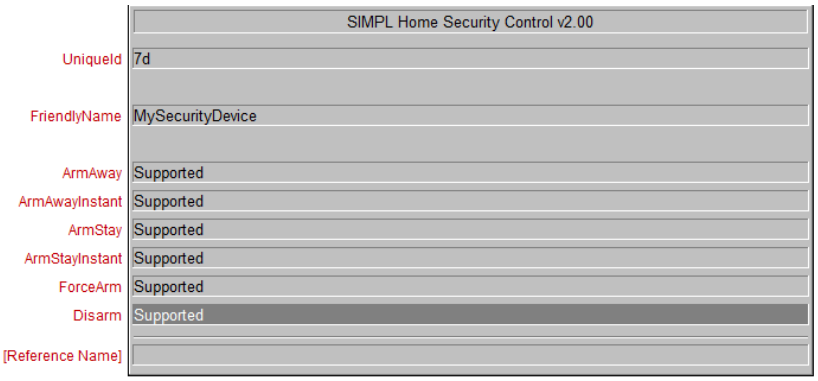
Starting with v2.00 of SIMPL Home we are now including support for selected Crestron Home device types to provide the deepest level of integration possible. The first such device type is the “Security System” and this provides the

COPYRIGHT © 2026 ULTAMATION LIMITED

mechanism to integrate any 3rd party security system within Crestron Home for which there is an existing Crestron SIMPL integration.

COPYRIGHT © 2026 ULTAMATION LIMITED

Integrated Security Example



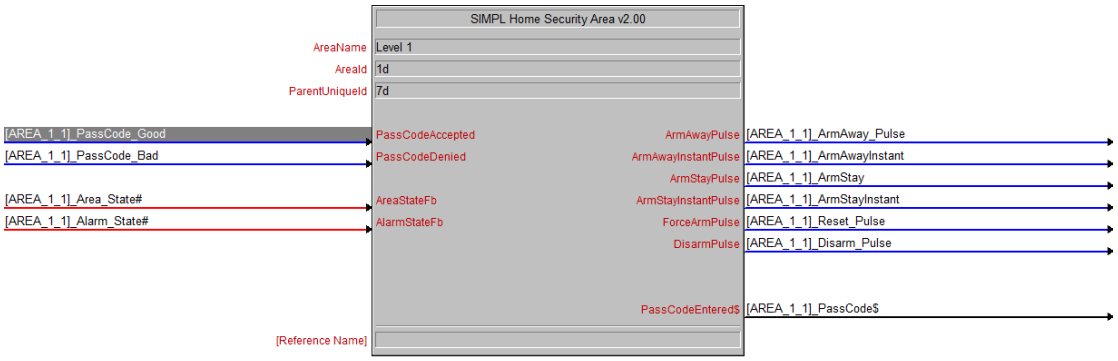
To setup a security system with using the integrated Crestron Security System driver model, you will need to add the SIMPL Home Security Control Module. This module acts as the focal point which all other security sub-type modules point towards. Therefore, if you wish to create areas and an emulated keypad for the security system, **the UniqueId must line up with the ParentUniqueId** of each sub module. The FriendlyName of the device will show up when you select the device from the SIMPL Home platform child folder in Crestron Home. Therefore, it should be unique to this instance.

There are 6 more fields, corresponding to supported Area Commands which can be used with your selected security system and are valid for use with Crestron Home. These are drop down fields which either enable or disable a specific command.

Note: Although the Crestron CCD SDK does support additional area commands, these do not appear to be supported by Crestron Home at this time.

In addition to the central control module, you will then add an optional keypad emulation module and one or more area modules.

Security Area



COPYRIGHT © 2026 ULTAMATION LIMITED

Each area module must have a unique AreaId and a ParentUniqueId corresponding to its central control module's UniqueId (7 in the example).

The Output signals on the right represent different state commands from Crestron Home. These represent the commands sent from the area and relate to the commands you have enabled in the Control tile. You cannot make specific commands available for different areas in Crestron Home at this time.

On the input side, the analog input signals can be used to send a specific area state or alarm state feedback to Crestron Home. We have created "helper" modules to send suitable values to these inputs (see the example SIMPL program). The values are as follows:

Area State Values

ArmedStay	1
ArmedStayInstant	2
ArmedAway	3
ArmedAwayInstant	4
ArmedBypass	5
ArmedBypassInstant	6
ExitDelayActive	7
EntryDelayActive	8
Disarmed	9
Ready	10
NotReady	11

Alarm State Values

Idle	0
Alarm	1
Fire	2
Burglary	3
Medical	4
Tamper	5

The Passcode signals can be used to verify if a passcode is valid for your physical security system or not.

PasscodeEntered$ will send the value entered within Crestron Home when prompted for the PIN.

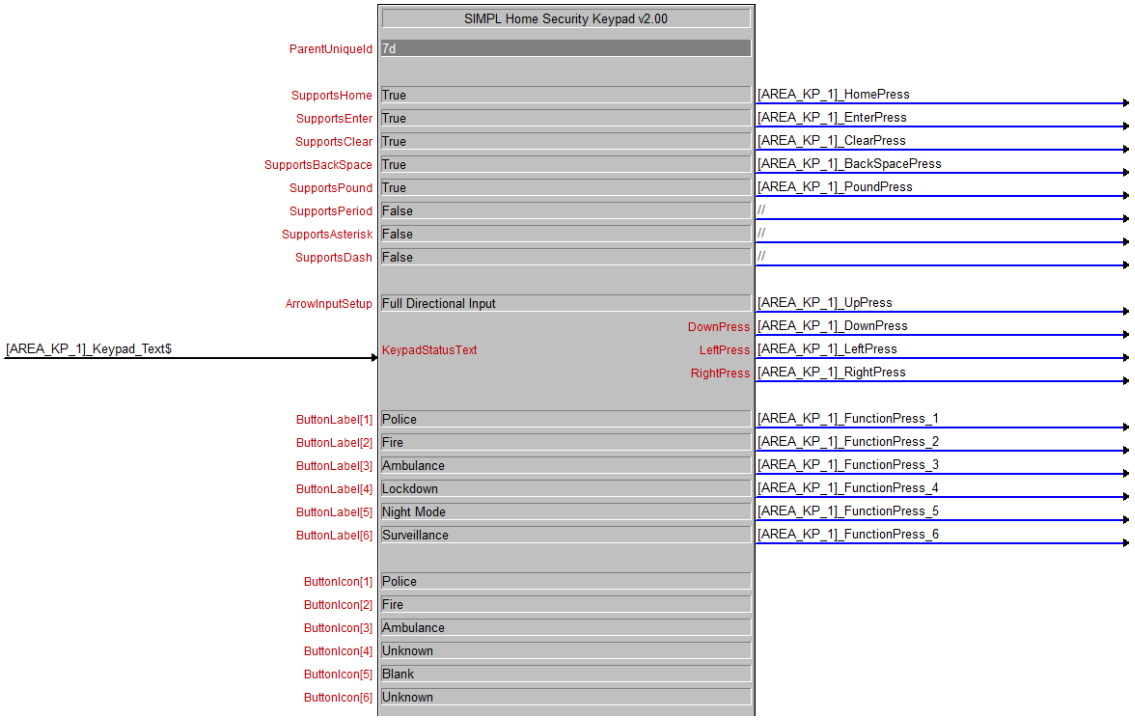
Once verified, you should then pulse either PassCodeAccepted or PassCodeDenied to inform Crestron Home that the code was accepted or not.

COPYRIGHT © 2026 ULTAMATION LIMITED

There appears to be a bug in Crestron Home that means an incorrect passcode message is not always displayed to the end user.

Keypad Emulation

A single keypad emulation module can be added to each security control module. This enables a virtual security keypad within Crestron Home.



As for “Areas”, the ParentUniqueId must match the UniqueId of the control module.

The SupportsParameters define which keypad keys should be displayed on the UI and a press in Crestron Home will pulse the corresponding digital outputs. Similarly, the navigation keypad can be enabled in various modes.

Passing a serial value to the KeypadStatusText input will display the message within Crestron Home.

A keypad can also present a number of function buttons. These buttons have fixed labels and icons (the icons, unlike most other icon labels, are limited to a predefined set – this is a constraint of the CCD SDK/Crestron Home).

The visibility and colours of “virtual LEDs” are both setup as parameters and cannot be altered dynamically. To enable an LED, simply give it a label, and – as for other endpoints - any blank labels (“”) will disable the corresponding LED. The analog

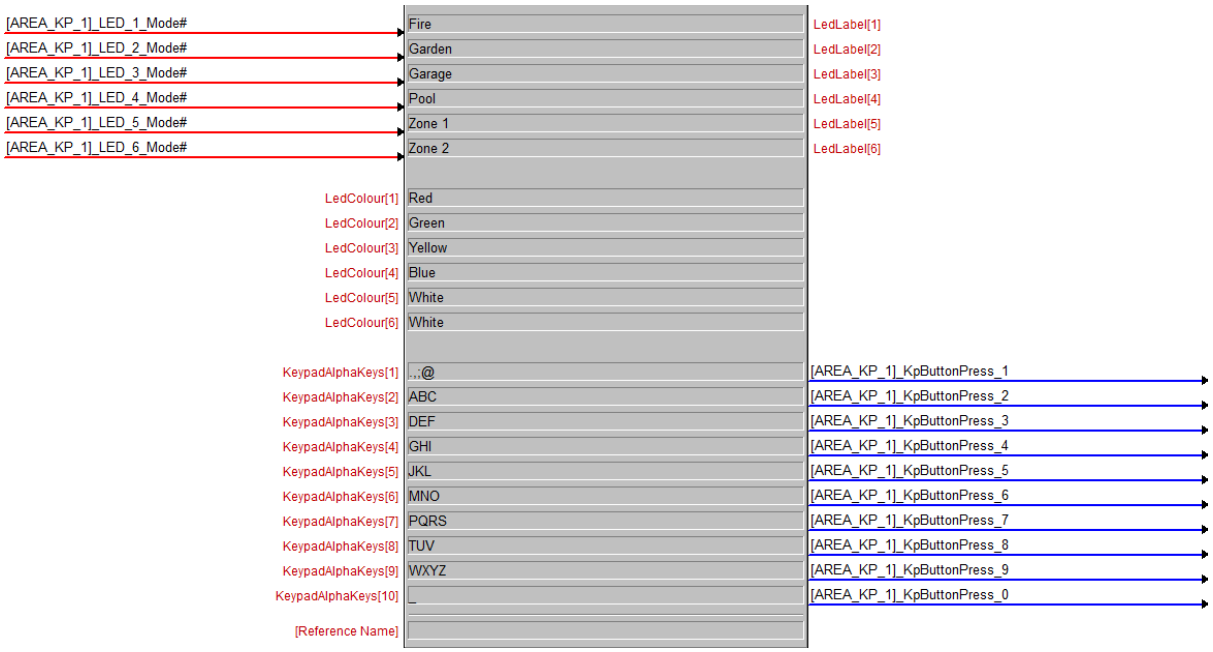
COPYRIGHT © 2026 ULTAMATION LIMITED

input signals LedStates[-] determine the state of each LED and there is, once again, a helper module with corresponding digital signals.

LED State Values

Off	1
On	2
Blinking	3

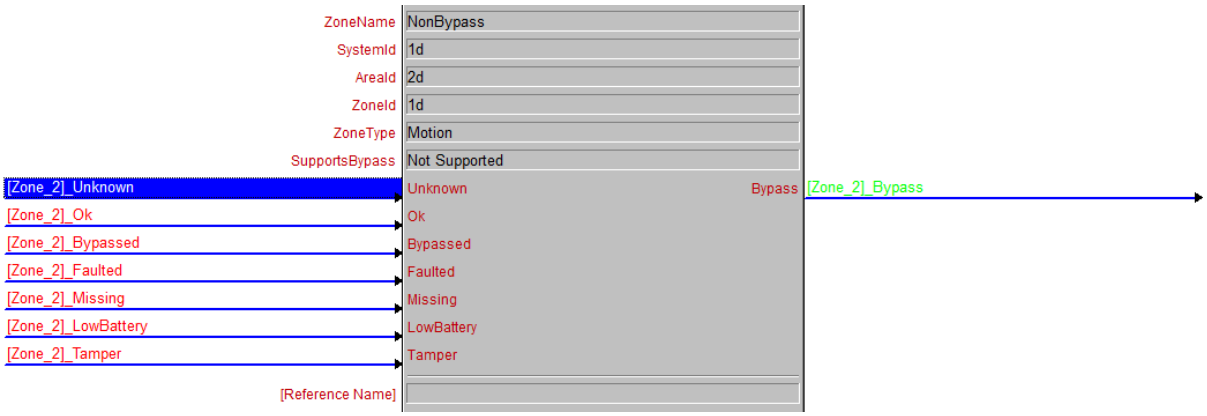
LED colours are set from a pre-defined selection supported by Crestron Home.



The last parameters are the KeypadAlphakeys[-], here you can input a string of up to 4 characters for each numerical input which will be displayed as alternate inputs on the numerical buttons. These indices range from 1 to 10, with 10 being assigned to the 0 key. There are then a set of digital output signals to receive the input from Crestron Home for each button.

COPYRIGHT © 2026 ULTAMATION LIMITED

Security Zone



For zones, There are multiple required parameters.

The SystemId must match the UniqueId of the security device module.

The AreaId parameter must match the AreaId of the area module you wish the zone to be assigned to. In Crestron Home, zones *must* be in an area, so this parameter is required.

The ZoneId is a unique identifier for that zone, and must be unique among all other zones assigned to that same area.

The ZoneType is a dropdown list selection of the possible zone types supported by Crestron Home.

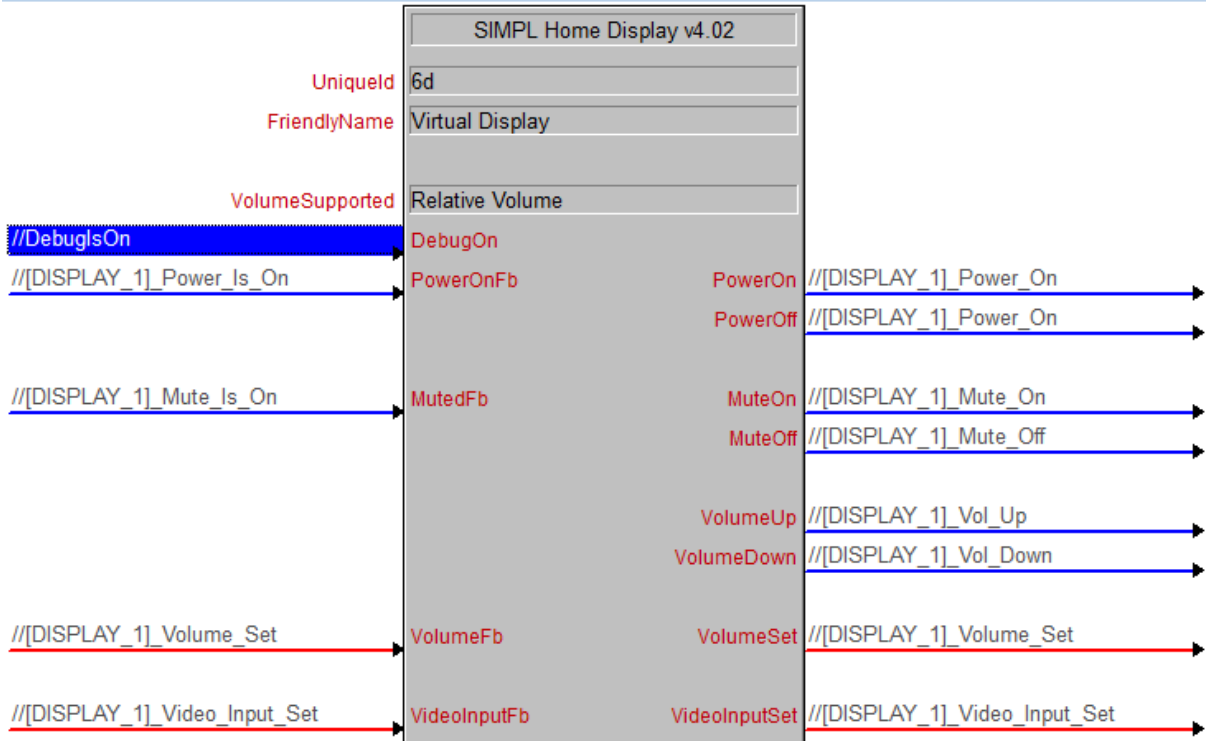
SupportsBypass will determine if the zone can be bypassed and unbypassed.

There are then 7 digital inputs that describe the state of the zone. Each one, when high, will send that state to the Crestron Home processor, and respective information will be shown to the user in the Crestron Home UI. There is a single digital output, which goes high when the zone has been bypassed and low when it is not.

COPYRIGHT © 2026 ULTAMATION LIMITED

Integrated Display Example

Setting up and integrating a display module within Simpl Home is relatively straightforward. All you need to add is the Simpl Home Display Module shown below and configure it as desired. You have the standard Unique Id and Friend Name.



Additionally, there is a **Volume_Supported** parameter. This allows the user to specify whether their device supports volume control from Crestron Home and if so in what format: Relative or Discrete.

Digital signals are always set on the Crestron Home side as Discrete High or Low values. However, the programmer can handle them as a toggle on the Simpl end. The feedback from the Host end will only have one Digital signal for power and one for mute indicating if either is on or off.

Signals:

Power_On_Fb - if high Crestron home will be up notified of the device powering on and if it is sent low it will be notified of the display powering down. Due to a limitation in Crestron Home, you cannot currently use this show that the device has powered down in Crestron Home. This is included for posterity in the event this is updated. CH will still show the device controls as if it is turned on.

COPYRIGHT © 2026 ULTAMATION LIMITED

Power_Is_On – allows for a power on command signal to be received from Crestron Home.

Power_Is_Off – allows for a power off command signal to be received from Crestron Home.

Mute_Is_On– allows for a mute on command signal to be received from Crestron Home.

Mute_Is_Off – allows for a mute off command signal to be received from Crestron Home.

Volume_Up – allows for a volume up command signal to be received from Crestron Home.

Volume_Down – allows for a volume down command signal to be received from Crestron Home.

Muted_Fb - to show mute on and off in Crestron Home. Works as expected.

Volume_Fb - sends an integer value update to Crestron Home, used to verify that a volume change has occurred and update CH UI feedback.

Volume_Set – allows for volume update command to be send from Crestron Home to the connected display device.

Video_Input_Fb - sends an integer value update to Crestron Home, used to select a video input. These go from 1 to 5, depending on how many inputs you are using.

Video_Input_Set – allows for Video Input command to be sent from Crestron Home updating the selected video inputs for a connected display device.

COPYRIGHT © 2026 ULTAMATION LIMITED

Source Example

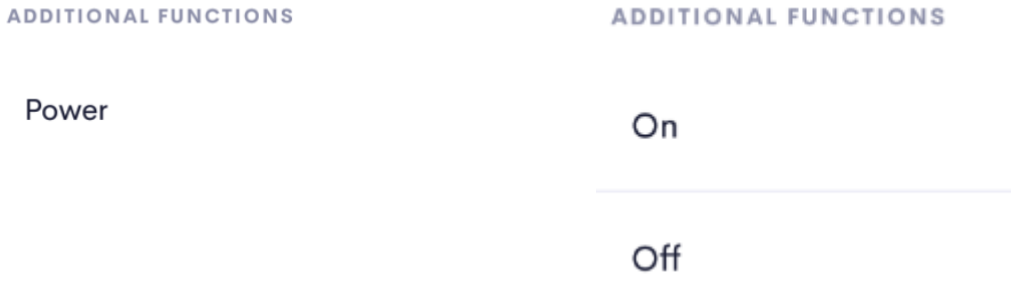
The Source module brings AV source devices to Simpl Home. This module is highly configurable, allowing you to specify what controls are supported by the source device you're integrating. The images below explain how the Simpl module's parameters, outputs, and inputs map to the UI of a source device in Crestron Home.

	SIMPL Home Source v3.00
UniqueId	12d
FriendlyName	SourceExample
PowerTypeSupported	Discrete power
MuteTypeSupported	Discrete mute
MuteFeedbackSupported	True
VolumeTypeSupported	Discrete Volume
VolumePercentFeedbackSupported	True
BackSupported	True
MenuSupported	True
ExitSupported	True
HomeSupported	True
PageControlsSupported	True
ArrowKeysSupported	True
ChannelControlsSupported	True
ScanReverseSupported	True
PlaySupported	True
ScanForwardSupported	True
SkipReverseSupported	True
PauseSupported	True
SkipForwardSupported	True
StopSupported	True
ReplaySupported	True
RecordSupported	True
LiveSupported	True
DVRSupported	True
GuideSupported	True
InfoSupported	True
LastSupported	True
ColourButtonsSupported	True
NumberKeyPadSupported	True
DashSupported	True
ClearSupported	True
EnterSupported	True

COPYRIGHT © 2026 ULTAMATION LIMITED

Parameters

1. PowerTypeSupported – The options for this are None, Toggle Power, and Discrete Power. Selecting None will result in no power controls under additional functions in the UI, and no power signal output from the Simpl module. Selecting Toggle Power results in the Power button showing under additional functions. In this case, the Power_Toggle output should be monitored for power commands. Selecting Discrete Power results in the On and Off buttons showing under additional functions. In this case, the Power_On and Power_Off outputs should be monitored for power commands.



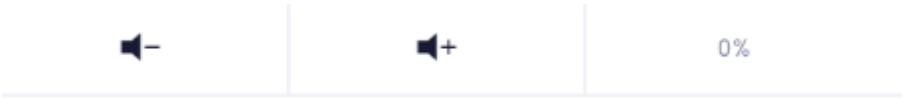
2. MuteTypeSupported – The options for this are None, Toggle Mute, and Discrete Mute. Selecting None will result in no mute control on the UI and no Mute outputs from the Simpl Module. Selecting Toggle Mute or Discrete Mute will result in the mute button showing on the UI. For Toggle Mute, the Toggle_Mute output should be monitored for mute commands. For Discrete_Mute, the Mute_On and Mute_Off outputs should be monitored.

3. MuteFeedbackSupported – Setting this to true will mean that Crestron Home is expecting feedback on the device's mute status through the Simpl module's Mute_On_Fb and Mute_Off_Fb inputs. If this is not enabled, then the mute button will be shown only in mute on mode.

4. VolumeTypeSupported – The options for this are None, Relative Volume, and Discrete Volume. Selecting None will result in no volume control on the UI and no volume outputs from the Simpl Module. Selecting Relative Volume will result in volume up and volume down controls been shown on the UI. In this case, the Volume_Up and Volume_Down outputs should be monitored

COPYRIGHT © 2026 ULTAMATION LIMITED

for volume commands. If Volume Percent Feedback is also supported, then a volume percentage will be shown next to the volume controls.



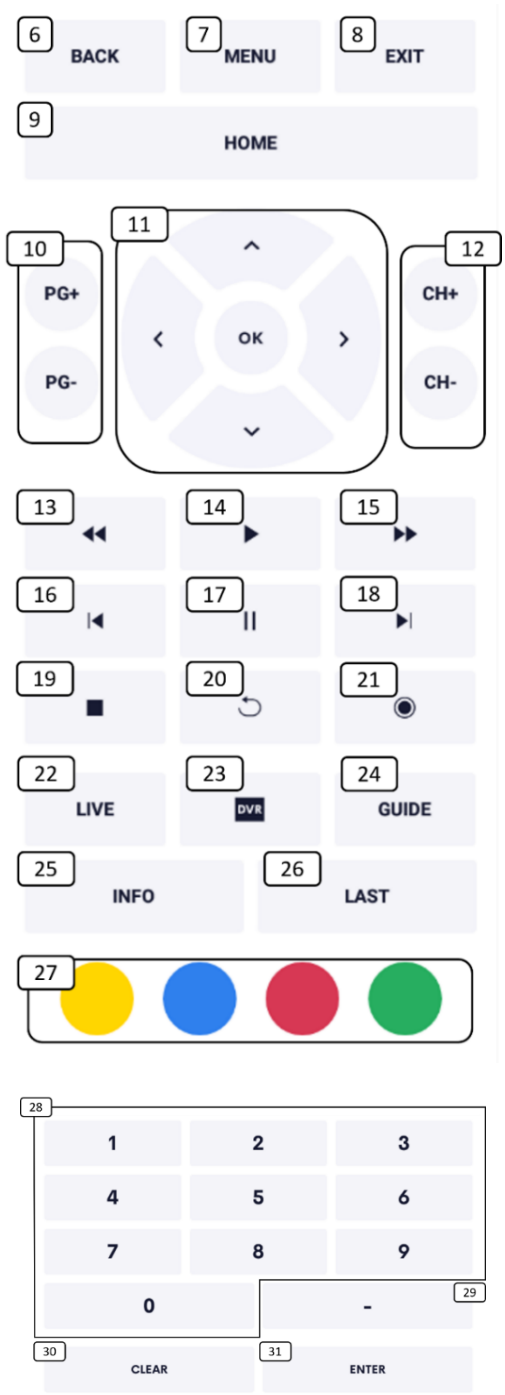
Selecting Discrete Volume will result in a volume slider been shown on the UI. In this case, the Volume_Set output should be monitored for volume commands. This volume type requires that Volume Percent Feedback is also supported.



5. VolumePercentFeedbackSupported - Setting this to true will mean that feedback on the device's volume level is expected through the Simpl module's Volume_FB input. The input is expected to be in the full scale 0-65535. E.g., 0%=0d, 50%=32768d, 100%=65535d.

COPYRIGHT © 2026 ULTAMATION LIMITED

The annotated images below show how the numbered parameters correspond to the UI elements.



6. **BackSupported** -Setting this to true will show the back button. Pressing the back button will pulse the Back output.

7. **MenuSupported** - Setting this to true will show the menu button. Pressing the menu button will pulse the Menu output.

COPYRIGHT © 2026 ULTAMATION LIMITED

8. ExitSupported - Setting this to true will show the exit button. Pressing the exit button will pulse the Exit output.

9. HomeSupported - Setting this to true will show the home button. Pressing the home button will pulse the Home output.

10. PageControlsSupported - Setting this to true will show the page up and page down buttons. Pressing the page up button will pulse the Page_Up output. Pressing the page down button will pulse the Page_Down output. Pressing and holding one of these buttons will cause its corresponding output to be held high whilst the button is held.

11. ArrowKeysSupported – Setting this to true will show the arrow keys and the select button, which is marked as OK on the UI. Pressing the select button will pulse the Select output. Pressing one of the arrow keys will pulse the corresponding Arrow_Key output. Pressing and holding one of the arrow buttons will cause its output to be held high whilst the button is held.

12. ChannelControlsSupported - Setting this to true will show the channel up and channel down buttons. Pressing the channel up button will pulse the Channel_Up output. Pressing the channel down button will pulse the Channel_Down output. Pressing and holding one of these buttons will cause its corresponding output to be held high whilst the button is held.

13. ScanReverseSupported - Setting this to true will show the reverse scan button. Pressing the reverse scan button will pulse the Reverse_Scan output.

14. PlaySupported - Setting this to true will show the play button. Pressing the play button will pulse the Play output.

15. ScanForwardSupported – Setting this to true will show the forward scan button. Pressing the forward scan button will pulse the Forward_Scan output.

16. SkipReverseSupported – Setting this to true will show the reverse skip button. Pressing the reverse skip button will pulse the Reverse_Skip output.

COPYRIGHT © 2026 ULTAMATION LIMITED

17. PauseSupported – Setting this to true will show the pause button. Pressing the pause button will pulse the Pause output.

18. SkipForwardSupported - Setting this to true will show the forward skip button. Pressing the forward skip button will pulse the Forward_Skip output.

19. StopSupported - Setting this to true will show the stop button. Pressing the stop button will pulse the Stop output.

20. ReplaySupported - Setting this to true will show the replay button. Pressing the replay button will pulse the Replay output.

21. RecordSupported - Setting this to true will show the record button. Pressing the record button will pulse the Record output.

22. LiveSupported - Setting this to true will show the live button. Pressing the live button will pulse the Live output.

23. DVRSupported - Setting this to true will show the DVR button. Pressing the DVR button will pulse the DVR output.

24. GuideSupported - Setting this to true will show the guide button. Pressing the guide button will pulse the Guide output.

25. InfoSupported - Setting this to true will show the Info button. Pressing the info button will pulse the Info output.

26. LastSupported - Setting this to true will show the last button. Pressing the last button will pulse the Last output.

27. ColourButtonsSupported - Setting this to true will show the colour buttons; red, green, yellow, blue. Pressing a colour button will pulse its corresponding Colour_Button output.

COPYRIGHT © 2026 ULTAMATION LIMITED

28. NumberKeyPadSupported - Setting this to true will show the number buttons, 0 to 9. Pressing a number button will change the value of the Keypad_Number output to that button's corresponding number. This parameter must be true for dash, clear, and enter to be supportable.
29. DashSupported – The NumberKeyPadSupported parameter must also be set to true for dash to be supported. Setting this to true will show the dash button. Pressing the dash button will pulse the Dash output.

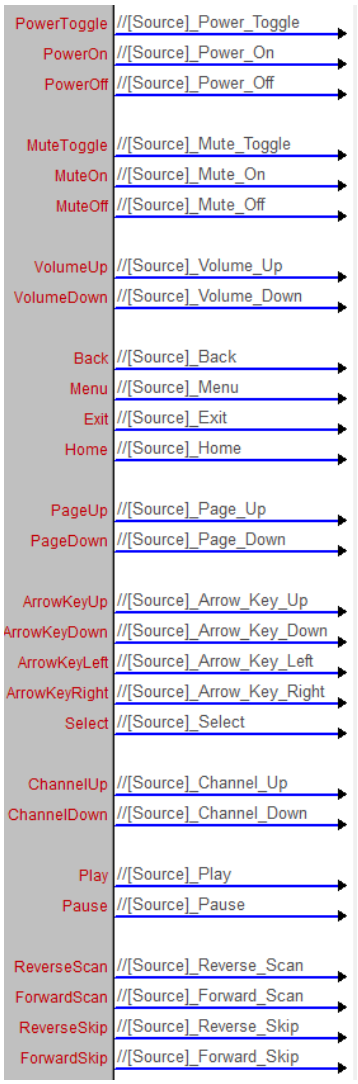
30. ClearSupported - The NumberKeyPadSupported parameter must also be set to true for clear to be supported. Setting this to true will show the clear button. Pressing the clear button will pulse the Clear output.

31. EnterSupported - The NumberKeyPadSupported parameter must also be set to true for enter to be supported. Setting this to true will show the enter button. Pressing the enter button will pulse the Enter output.

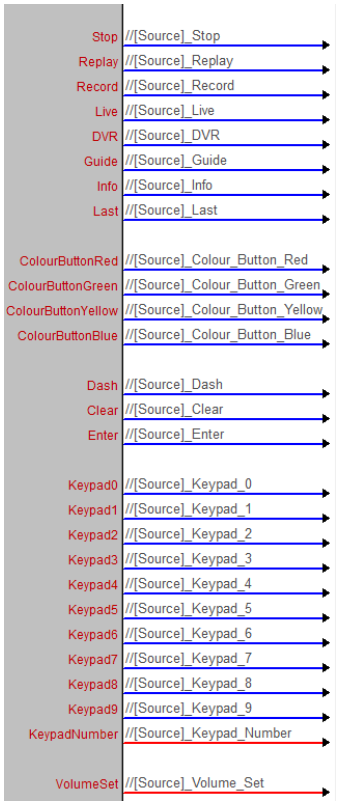
32. SetChannelSupported – If this parameter is set to true, when a favourite is pressed within the Crestron Home app, the keypad analog and digitals will output the corresponding Channel number one digit at a time.

COPYRIGHT © 2026 ULTAMATION LIMITED

Outputs



COPYRIGHT © 2026 ULTAMATION LIMITED



Power_Toggle – Used when PowerTypeSupported is set to Toggle Power. This output will pulse when the source is turned on/off from the UI.

Power_On – Used when PowerTypeSupported is set to Discrete Power. This output will go high when the source is turned on from the UI. It will go low when the source is turned off from the UI.

Power_Off – Used when PowerTypeSupported is set to Discrete Power. This output will go low when the source is turned on from the UI. It will go high when the source is turned off from the UI.

Mute_Toggle – Used when MuteTypeSupported is set to Toggle Mute. This output will pulse when the source is muted/unmuted from the UI.

Mute_On – Used when MuteTypeSupported is set to Discrete Mute. This output will go high when the source is muted from the UI. It will go low when the source is unmuted from the UI.

Mute_Off – Used when MuteTypeSupported is set to Discrete Mute. This output will go low when the source is muted from the UI. It will go high when the source is unmuted from the UI.

COPYRIGHT © 2026 ULTAMATION LIMITED

Volume_Up – Used when VolumeTypeSupported is set to Relative Volume. This output will pulse when the UI's volume up button is pressed. It will stay high whilst the UI's volume up button is pressed and held.

Volume_Down – Used when VolumeTypeSupported is set to Relative Volume. This output will pulse when the UI's volume down button is pressed. It will stay high whilst the UI's volume down button is pressed and held.

Back – This output will pulse when the UI's back button is pressed.

Menu – This output will pulse when the UI's menu button is pressed.

Exit – This output will pulse when the UI's exit button is pressed.

Home – This output will pulse when the UI's home button is pressed.

Page_Up – This output will pulse when the UI's page up button is pressed. It will stay high whilst the UI's page up button is pressed and held.

Page_Down – This output will pulse when the UI's page down button is pressed. It will stay high whilst the UI's page down button is pressed and held.

Arrow_Key_Up – This output will pulse when the UI's up arrow key button is pressed. It will stay high whilst the UI's up arrow key button is pressed and held.

Arrow_Key_Down – This output will pulse when the UI's down arrow key button is pressed. It will stay high whilst the UI's down arrow key button is pressed and held.

Arrow_Key_Left – This output will pulse when the UI's left arrow key button is pressed. It will stay high whilst the UI's left arrow key button is pressed and held.

Arrow_Key_Right – This output will pulse when the UI's right arrow key button is pressed. It will stay high whilst the UI's right arrow key button is pressed and held.

Select – This output will pulse when the UI's select button is pressed. This button is labelled as OK in the UI.

Channel_Up – This output will pulse when the UI's channel up button is pressed. It will stay high whilst the UI's channel up button is pressed and held.

Channel_Down – This output will pulse when the UI's channel down button is pressed. It will stay high whilst the UI's channel down button is pressed and held.

Play – This output will pulse when the UI's play button is pressed.

COPYRIGHT © 2026 ULTAMATION LIMITED

Pause – This output will pulse when the UI's pause button is pressed.

Reverse_Scan – This output will pulse when the UI's reverse scan button is pressed.

Forward_Scan – This output will pulse when the UI's forward scan button is pressed.

Reverse_Skip – This output will pulse when the UI's reverse skip button is pressed.

Forward_Skip – This output will pulse when the UI's forward skip button is pressed.

Stop – This output will pulse when the UI's stop button is pressed.

Replay – This output will pulse when the UI's replay button is pressed.

Record – This output will pulse when the UI's record button is pressed.

Live – This output will pulse when the UI's live button is pressed.

DVR – This output will pulse when the UI's DVR button is pressed.

Guide – This output will pulse when the UI's guide button is pressed.

Info – This output will pulse when the UI's info button is pressed.

Last – This output will pulse when the UI's last button is pressed.

Colour_Button_Red - This output will pulse when the UI's red colour button is pressed.

Colour_Button_Green - This output will pulse when the UI's green colour button is pressed.

Colour_Button_Yellow - This output will pulse when the UI's yellow colour button is pressed.

Colour_Button_Blue – This output will pulse when the UI's blue colour button is pressed.

Dash – This output will pulse when the UI's dash button is pressed.

Clear – This output will pulse when the UI's clear button is pressed.

Enter – This output will pulse when the UI's enter button is pressed.

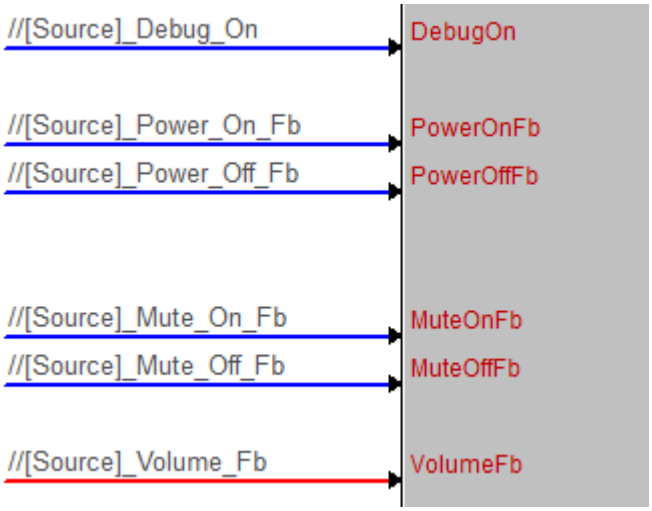
COPYRIGHT © 2026 ULTAMATION LIMITED

Keypad_0-9 – These outputs will pulse with the corresponding keypad pressed within Crestron Home.

Keypad_Number – This output's value will change when one of the UI's keypad number buttons is pressed. It will change to the number corresponding to the pressed button.

Volume_Set – This output's value will change when the volume is changed from UI. The value will be in the full scale 0-65535. E.g., 0d =0%, 32768d =50%, 65535d=100%.

Inputs



Debug_On – Setting this high will enable additional logging for debugging purposes.

Power_On_Fb – Power state feedback. Setting this high will instruct Crestron Home that the source device is powered on.

Power_Off_Fb - Power state feedback. Setting this high will instruct Crestron Home that the source device is powered off.

Mute_On_Fb - Mute state feedback. Setting this high will instruct Crestron Home that the source device is muted.

Mute_Off_Fb - Mute state feedback. Setting this high will instruct Crestron Home that the source device is unmuted.

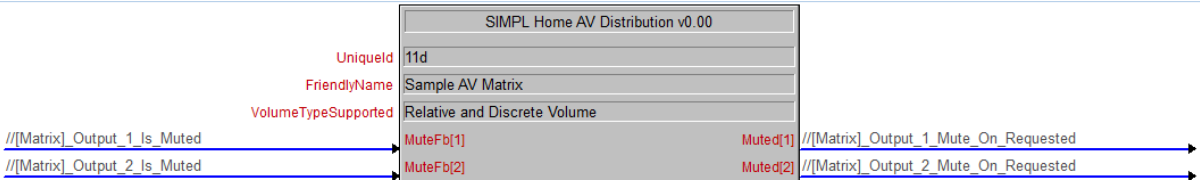
COPYRIGHT © 2026 ULTAMATION LIMITED

Volume_Fb – Volume level feedback. Must be given in full scale 0-65535. E.g., 0%=0d, 50%=32768d, 100%=65535d. This analogue input is for informing Crestron Home of the volume level on the source device. A change in this value will update the volume slider or volume percentage on the UI, depending on the parameter configuration. See the previous section for more details on this.

COPYRIGHT © 2026 ULTAMATION LIMITED

AV Distribution Example

The AV Distribution module brings AV Distribution devices (AV switchers, matrices etc.) to SIMPL Home. It Provides inputs and outputs for up to 32 AV outputs.



Parameters

VolumeTypeSupported – You can choose between **Relative**, **Discrete** and **Relative and Discrete**. **Relative** will use volume up and down commands and show up as two buttons in Crestron Home. **Discrete** is a slider that will just set the volume directly.

Inputs

MuteFb[1-32] – Setting these high will inform Crestron Home that the respective output is muted.

VolumeFb[1-32] – Setting these analog values will inform Crestron Home that the volume on the respective output has been changed to its new value.

Outputs

Muted[1-32] – These will go high when the user presses mute on one of the AV outputs in Crestron Home.

VolumeUp[1-32] – These will go high when the user presses the volume up button in **Relative** mode

VolumeDown[1-32] – These will go high when the user presses the volume down button in **Relative** mode.

Volume[1-32] – These will change when the user changes the volume of one of the AV outputs in Crestron Home.

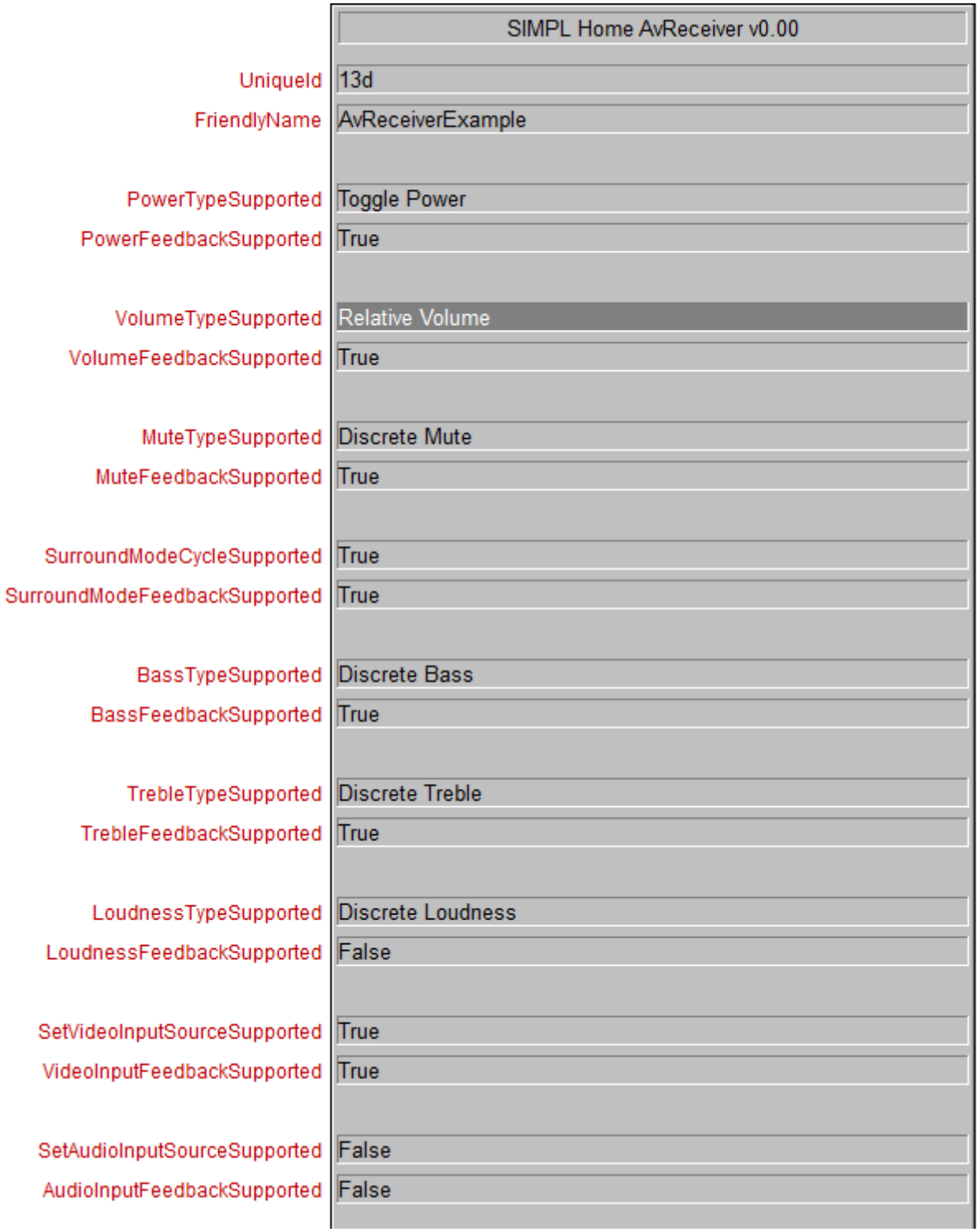
VideoOutput[1-32] – These will change when the user changes the video routing of an output in Crestron Home.

AudioOutput[1-32] – These will change when the user changes the audio routing of an output in Crestron Home.

COPYRIGHT © 2026 ULTAMATION LIMITED

AV Receiver Example

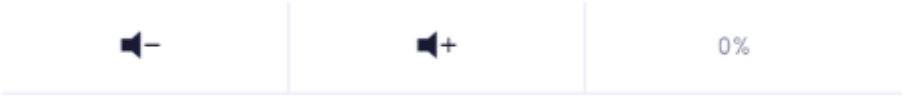
The AV Receiver module brings AV Receiver devices to SIMPL Home. This module is highly configurable, allowing you to specify what controls are supported by the AV Receiver device you're integrating. The images below explain how the SIMPL module's parameters, outputs, and inputs map to the AV Receiver device in Crestron Home.



COPYRIGHT © 2026 ULTAMATION LIMITED

Parameters

1. PowerTypeSupported – The options for this are None, Toggle Power, Discrete Power and Toggle and Discrete Power.
    a. None will result in no power controls,
    b. Toggle Power will send power using the Power_Toggle output by pulsing the output.
    c. Discrete will trigger either the Power_On or the Power_Off when triggered within Crestron Home.
2. PowerFeedbackSupported – Setting this to true will mean that Crestron Home is expecting feedback on the device's power status through the SIMPL module's Power_Fb input. This will be used to keep Crestron Home up to date with the current power state of the AVR.
3. VolumeTypeSupported – The options for this are None, Relative Volume, Discrete Volume, and Relative and Discrete Volume.
    a. None will result in no volume control on the UI and no volume outputs from the SIMPL Module.
    b. Relative Volume will result in volume up and volume down controls been shown on the UI. In this case, the Volume_Up and Volume_Down outputs should be monitored for volume commands. If Volume Percent Feedback is also supported, then a volume percentage will be shown next to the volume controls.



    c. Discrete Volume will result in a volume slider been shown on the UI. In this case, the Volume_Set output should be monitored for volume commands. This volume type requires that Volume Percent Feedback is also supported.

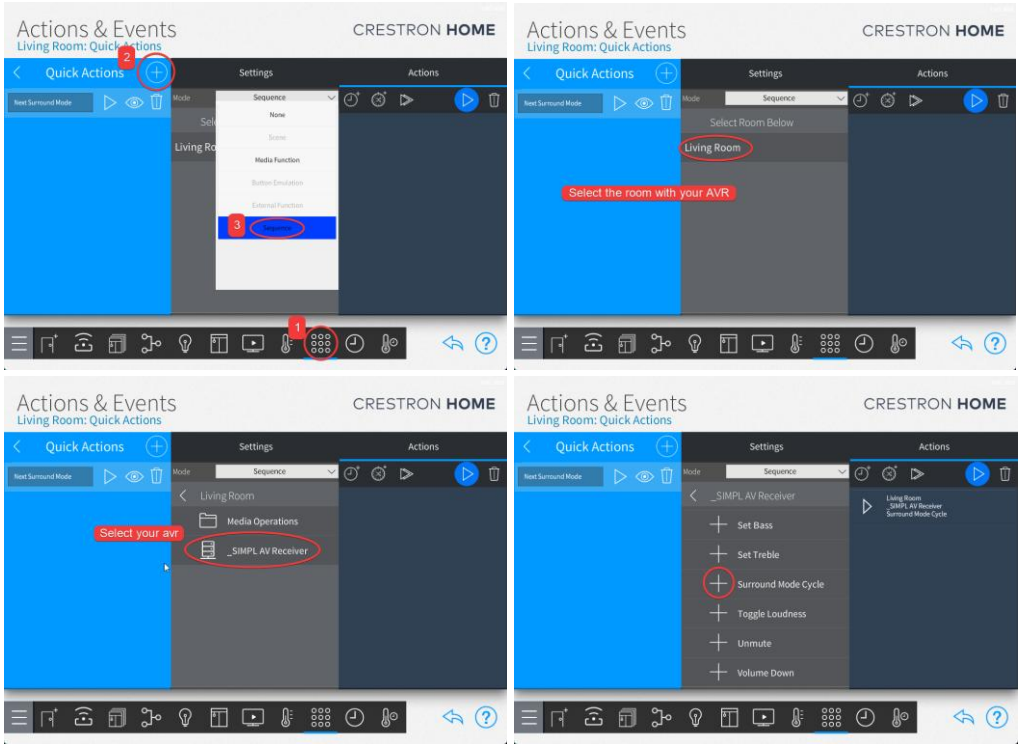


4. VolumeFeedbackSupported – Setting this to true will mean that Crestron Home is expecting feedback for this device's current volume level through the SIMPL module's Volume_Fb# input. This will be expected to be in the full scale 0-65535d, e.g., 0% - 0d, 50% - 32768d, 100% - 65535d.
5. MuteTypeSupported - The options for this are None, Toggle Mute, Discrete Mute, and Toggle and Discrete Mute.

COPYRIGHT © 2026 ULTAMATION LIMITED

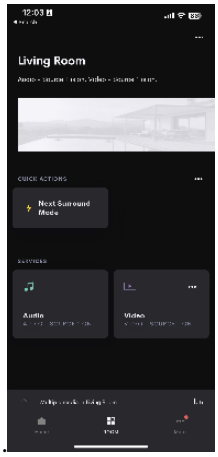
a. **None** will result in no mute control on the UI and no Mute outputs from the SIMPL Module. Selecting Toggle Mute or Discrete mute will result in the mute button showing on the UI.
b. **Toggle Mute**, the Toggle_Mute output should be monitored for mute commands, this will be a pulse.
c. **Discrete_Mute**, the Mute_On and Mute_Off outputs should be monitored for Pulses. They should alternate based on what Crestron home considers the mute state of the AVR.

6. **MuteFeedbackSupported** – Setting this to true will mean that Crestron Home is expecting feedback for this device's current mute state through the SIMPL module's Mute_Fb input. If this is not enabled, then the mute button will be shown only in mute on state.
7. **SurroundModeCycleSupported –** The options for this are True or False. Setting this to true will show the option to cycle through surround modes in the quick actions section, which you can assign to a shortcut as shown below.



This will create a button on the App UI for cycling surround modes, Called whatever you named the quick action. Selecting this button will pulse the Surround_Mode_Cycle output Signal.

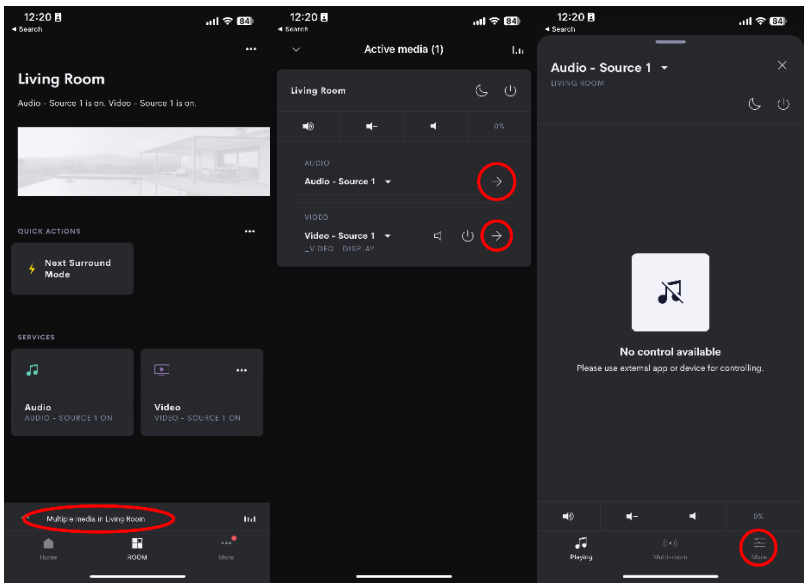
COPYRIGHT © 2026 ULTAMATION LIMITED



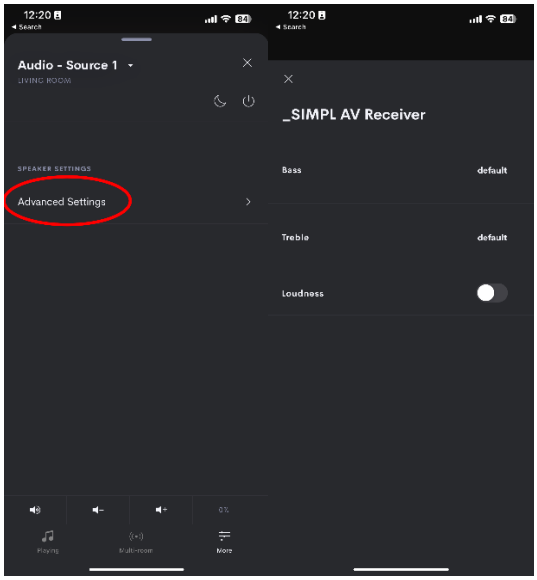
8. **SurroundModeFeedbackSupported** – Setting this to true will mean that Crestron Home is expecting feedback for this device's current Surround Mode state through the SIMPL module's Surround_Mode_Fb input. This will update the internal surround mode to a value from 0-10. The values correspond to surround modes supported by Crestron Home, they are mapped in the following way:
   0 - Unknown

1 – Movie	2 – Music	3 – Game	4 - Direct	5 – PureDirect
6 – Stereo	7 – Auto	8 – DolbyDigital	9 – DtsSurround	10 - Thx

The Following (Bass Treble Loudness) are shown in the advanced menu, here is a visual of how to access this menu on the app:



COPYRIGHT © 2026 ULTAMATION LIMITED



9. **BassTypeSupported** – The options for this are None, Relative Bass, Discrete Bass, and Relative and Discrete Bass. Selecting None will result in no bass controls being shown, however note that with None selected but feedback on you will still see the bass percentage. Selecting Relative Bass will result in plus/minus buttons showing, and again if feedback is supported you will see a bass percentage. Selecting Discrete Bass will result in a slider being shown, this slider will output to the Bass# Signal, and will output from -50d [65486d] (0% Crestron Home) to 50d (100% Crestron Home) however note that the slider will not be shown if feedback is not supported.
10. **BassFeedbackSupported** – Setting this to true will mean that Crestron Home is expecting feedback for this devices current Bass level through the SIMPL module's Bass_Fb# input. This will take in values from -50d [65486d] (0% Crestron Home) to 50d (100% Crestron Home) and will allow for a visual update of what the bass it set to on the AVR.
11. **TrebleTypeSupported** – The options for this are None, Relative Treble, Discrete Treble, and Relative and Discrete Treble.
    a. **None** will result in No Treble controls being shown, however note that with None selected but feedback on you will still see the Treble percentage.
    b. **Relative Treble** will result in plus/minus buttons showing, and again if feedback is supported you will see a bass percentage.
    c. **Discrete Treble** will result in a slider being shown, this slider will output to the Treble# Signal, and will output from -50d [65486d] (0%

COPYRIGHT © 2026 ULTAMATION LIMITED

Crestron Home) to 50d (100% Crestron Home) however note that the slider will not be shown if feedback is not supported.

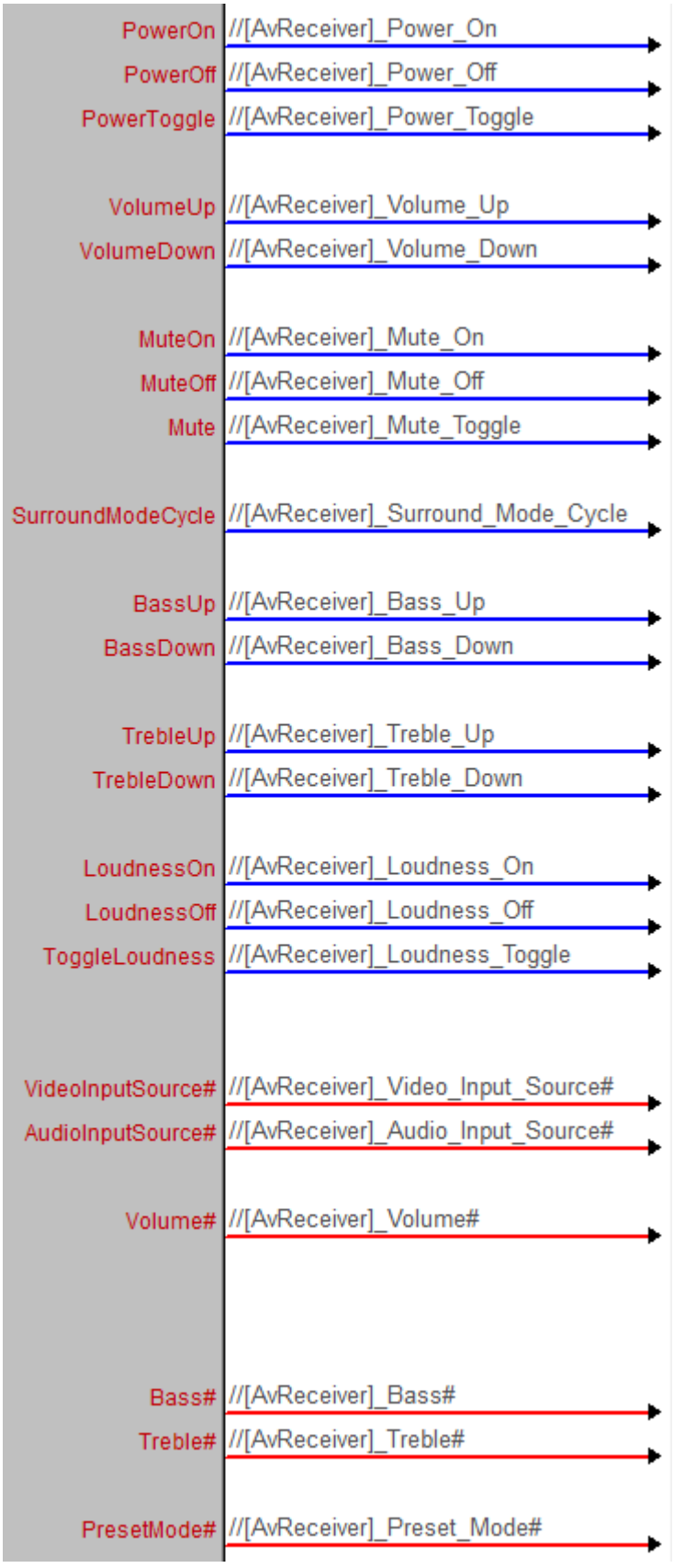
12. TrebleFeedbackSupported – Setting this to true will mean that Crestron Home is expecting feedback for this devices current Treble level through the SIMPL module's Treble_Fb# input. This will take in values from -50d [65486d] (0% Crestron Home) to 50d (100% Crestron Home) and will allow for a visual update of what the Treble it set to on the AVR.
13. LoudnessTypeSupported – The options for this are None, Relative Loudness, Discrete Loudness and Relative and Discrete Loudness. Currently we believe there is a Crestron home problem with Relative Loudness so currently (as of the 25th of august 2023) both None, and Relative Loudness aren't recommended by us, However Discrete Loudness is working and it outputs to the Loudness_On / Loudness_Off Signals. These will pulse according to what loudness state Crestron Home believes the AVR is in.
14. LoudnessFeedbackSupported – Setting this to true will mean that Crestron Home is expecting feedback for this devices current Loudness state through the SIMPL module's Loudness_Fb input. This will update the internal state of Loudness, however due to Crestron Home Feedback doesn't actually change the visual state of the Loudness switch.
15. SetVideoInputSourceSupported – The options for this are either True or False. Setting this to True will tell Crestron Home whether the AVR supports Video inputs.
16. VideoInputFeedbackSupported – Setting this to true will mean that Crestron Home is expecting feedback for this device's current Video Source Selected through the SIMPL module's Video_Input_Source_Fb# input. The feedback will change both the Audio and Video state of Crestron home since Crestron Home links Video and Audio. If you want to separate these two you will need to set the video to a particular source FIRST, then afterwards set the Audio input which will only set audio.
17. SetAudioInputSourceSupported – The options for this are either True or False. Setting this to True will tell Crestron Home whether the AVR supports Audio inputs.
18. AudioInputFeedbackSupported – Setting this to true will mean that Crestron Home is expecting feedback for this device's current Audio Source Selected through the SIMPL module's Audio_Input_Source_Fb# input. The feedback will change only the Audio state of Crestron home, so again if you want to

COPYRIGHT © 2026 ULTAMATION LIMITED

separate these audio and video inputs you will need to set the video to a particular source FIRST, then afterwards set the Audio input which will only set audio.

COPYRIGHT © 2026 ULTAMATION LIMITED

Outputs



COPYRIGHT © 2026 ULTAMATION LIMITED

1. Power_On – Used when PowerTypeSupported is set to Discrete Power. This will output a pulse to turn the AV Receiver on when Crestron Home thinks the AV Receiver is off and needs to be turned on.

2. Power_Off - Used when PowerTypeSupported is set to Discrete Power. This will output a pulse to turn the AV Receiver off when Crestron Home thinks the AV Receiver is on and needs to be turned off, or when the user clicks the power button in the UI.

3. Power_Toggle – Used when PowerTypeSupported is set to Toggle Power. This will output a pulse to toggle the power state of the AV Receiver.

4. Volume_Up – Used when VolumeTypeSupported is set to Relative Volume. This output will pulse when the UI's volume up button is pressed. When the UI's volume up button is pressed and held it will either send lots of pulses to ramp up the volume or will stay high the whole-time volume up is held.

5. Volume_Down - Used when VolumeTypeSupported is set to Relative Volume. This output will pulse when the UI's volume down button is pressed. When the UI's volume down button is pressed and held it will either send lots of pulses to ramp down the volume or will stay high the whole-time volume down is held.

6. Mute_On – Used when MuteTypeSupported is set to Discrete Mute. This output will pulse when the AV Receiver is muted from the UI.

7. Mute_Off - Used when MuteTypeSupported is set to Discrete Mute. This output will pulse when the AV Receiver is unmuted from the UI.

8. Mute - Used when MuteTypeSupported is set to Toggle Mute. This output will pulse when the AV Receiver is muted/unmuted from the UI.

9. Surround_Mode_Cycle – Used when SurroundModeCycleSupported is set to true. This output will pulse when the AV Receiver's Surround Mode is changed from the UI.

COPYRIGHT © 2026 ULTAMATION LIMITED

10. Bass_Up – Used when BassTypeSupported is set to Relative. This output will pulse when the UI's bass up button is pressed. When the UI's bass up button is pressed and held it will stay high the whole-time bass up is held.

11. Bass_Down – Used when BassTypeSupported is set to Relative. This output will pulse when the UI's bass down button is pressed. When the UI's bass down button is pressed and held it will stay high the whole-time bass down is held.

12. Treble_Up – Used when TrebleTypeSupported is set to Relative. This output will pulse when the UI's treble up button is pressed. When the UI's treble up button is pressed and held it will stay high the whole-time treble up is held.

13. Treble_Down – Used when TrebleTypeSupported is set to Relative. This output will pulse when the UI's treble down button is pressed. When the UI's treble down button is pressed and held it will stay high the whole-time treble down is held.

14. Loudness_On - Used when LoudnessTypeSupported is set to Discrete Loudness. This output will pulse when the AV Receiver's Loudness it turned on from the UI.

15. Loudness_Off - Used when LoudnessTypeSupported is set to Discrete Loudness. This output will pulse when the AV Receiver's Loudness it turned off from the UI.

16. Toggle_Loudness - Used when LoudnessTypeSupported is set to Toggle Loudness. This output will pulse when the AV Receiver's Loudness it turned on/off from the UI.

17. Video_Input_Source# - Used when SetVideoInputSourceSupported is set to True. This output will be the number of the Video Source the user has selected. Input 1 will go to 1d, Input 2 will go to 2d and so on. This will only change when the AV Receivers input and output supports video.

COPYRIGHT © 2026 ULTAMATION LIMITED

18. Audio_Input_Source# - Used when SetAudioInputSourceSupported is set to True. This output will be the number of the Audio Source the user has selected. Input 1 will go to 1d, Input 2 will go to 2d and so on. This will change whenever the AV Receiver's input and output supports audio but will also change when the AV Receivers output supports audio and video.

19. Volume# - Used when VolumeTypeSupported is set to Discrete Volume. This output will be from 0d-65535d (Full scale) with 0d being 0% volume and 65535d being 100%, this will be changed when the user changes the volume via the volume slider and can be changed to a specific value using the quick actions (Just remember in quick actions this will be 0-100% but will end up being a full-scale analogue signal).

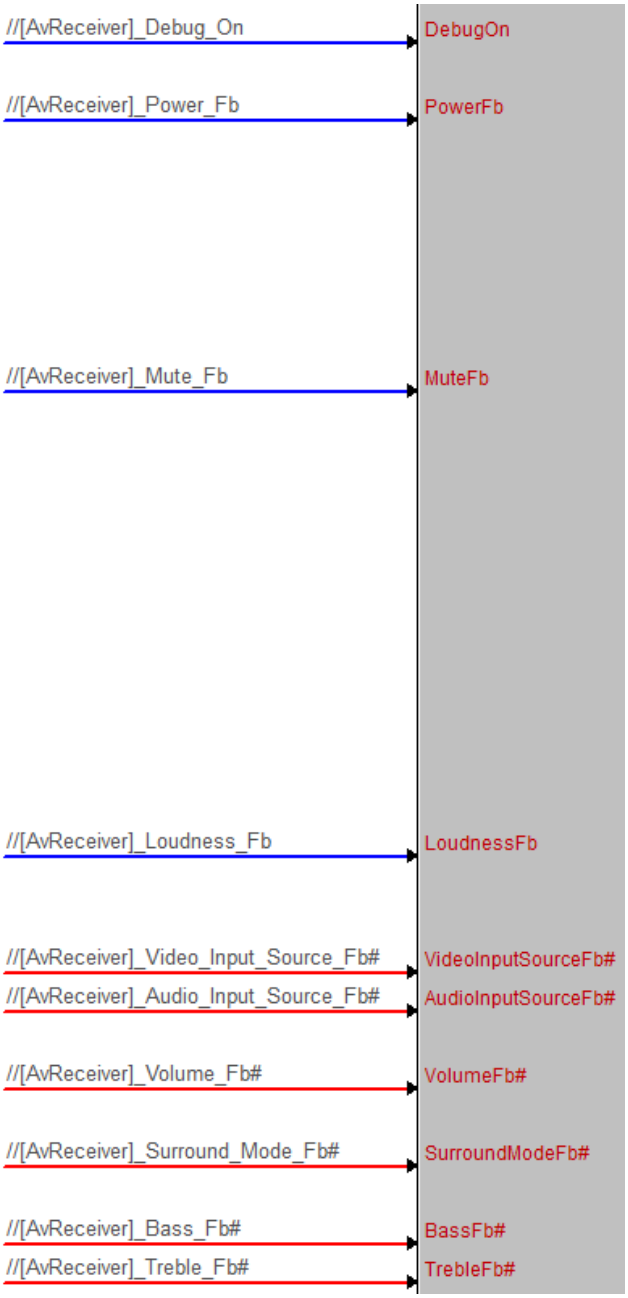
20. Bass# - Used when BassTypeSupported is set to Discrete Bass. This output will be from -50 [65486d] (0% on Crestron Home) to 50d (100% on Crestron Home) this output will be changed when the user changes the bass level via the slider in the advanced menu, or it can be changed to a specific value using the quick actions (Just remember in quick actions this will be 0-100% but will end up being shifted to -50 to 50 on the signal).

21. Treble# - Used when TrebleTypeSupported is set to Discrete Treble. This output will be from -50 [65486d] (0% on Crestron Home) to 50d (100% on Crestron Home) this output will be changed when the user changes the Treble level via the slider in the advanced menu, or it can be changed to a specific value using the quick actions (Just remember in quick actions this will be 0-100% but will end up being shifted to -50 to 50 on the signal).

22. Preset_Mode# - This output will be from 1-16 and it can be used for anything you wish. The way you would change it is you can access these presets in the quick actions menu. For example, you can find Preset 1 in quick actions and when triggered it will set this signal to 1, and Preset 2 will set this analogue to 2 and so on. This output can be used for anything needed or can be ignored.

COPYRIGHT © 2026 ULTAMATION LIMITED

Inputs



1. Debug_On – Setting this high will enable additional logging for debug purposes.
2. Power_Fb – Power state feedback. Setting this high will instruct Crestron Home that the AV Receiver is powered on.
3. Mute_Fb – Mute State feedback. Setting this high will instruct Crestron Home that the AV Receiver device is muted.
4. Loudness_Fb – Loudness state feedback. Setting this high will instruct Crestron Home that the AV Receiver device has got loudness on.

COPYRIGHT © 2026 ULTAMATION LIMITED

5. Video_Input_Source_Fb# - Video Input Source Feedback. Setting this to 1d-16d instruct Crestron Home that the current Input Source is Input 1 – Input 16. This will change both the current Video and Audio source on Crestron Home.
6. Audio_Input_Source_Fb# - Audio Input Source Feedback. Setting this to 1d-16d instruct Crestron Home that the current Audio Input Source is Input 1 – Input 16.
7. Volume_Fb# - Volume Feedback. Setting this to 0d-65535d will instruct Crestron Home that the current volume level is 0%-100%, e.g., setting this to 32768d will be 50% volume.
8. Surround_Mode_Fb# - Surround Mode Feedback. Crestron Home is expecting this to be a value 0d-10d, this will instruct Crestron Home what the current Surround mode is. 0d-10d map to the following Surround Modes:
   0- Unknown
   1- Movie
   2- Music
   3- Game
   4- Direct
   5- PureDirect
   6- Stereo
   7- Auto
   8- DoblyDigital
   9- DtsSurround
   10- Thx
9. Bass_Fb# - Bass Feedback. Crestron Home is expecting this to be from -50 [65486d] to 50 which it will see as 0% to 100%. This will instruct Crestron Home of what the current Bass value is set to.
10. Treble_Fb# - Treble Feedback. Crestron Home is expecting this to be from -50 [65486d] to 50 which it will see as 0% to 100%. This will instruct Crestron Home of what the current Treble value is set to.

COPYRIGHT © 2026 ULTAMATION LIMITED

Lighting Example

Light endpoints can only be accessed via the SIMPL Home Entities platform driver.



Parameters

LightType – Should match the type of light being integrated. Determines the capabilities of the light in Crestron Home and therefore which outputs and inputs can be used on the SIMPL module. The available types are:

- Intensity Only (brightness)
- Intensity and Tuneable White (brightness and CCT)
- Intensity and Tuneable Colour (brightness, hue, and saturation)
- Intensity. Tuneable White, and Tuneable Colour (brightness, CCT, hue, and saturation)
- Intensity. Tuneable Emulated White, and Tuneable Colour (brightness, emulated CCT, hue, and saturation)
- Switched (on/off)

CctMin – Correlated colour temperature minimum. Sets the minimum value for the light's tuneable white range, measured in Kelvin. Must be within the range 1000-20,000.

COPYRIGHT © 2026 ULTAMATION LIMITED

CctMax - Correlated colour temperature maximum. Sets the maximum value for the light's tuneable white range, measured in Kelvin. Must be within the range 1000-20,000.

RampEnabled – When enabled Crestron will send the ramping values, when disabled only the final values will be sent from Crestron.

Inputs

IsOnFb – Switched Light only. Set high for light on. Set low for light off.

IntensityFb – Intensity feedback. Must be given in full scale 0-65535 representing 0-100%.

CctFb – Correlated colour temperature feedback. Must be within the range defined by CctMin and CctMax.

HueFb – Hue feedback. Must be given as the value in degrees multiplied by 10. E.g., 120.5° should be entered as 1205. Range is 0-3600.

SaturationFb – Saturation feedback. Must be given in full scale 0-65535 representing 0-100%.

Outputs

TurnOn – Switched Light only. Pulses when the light is turned on in Crestron Home.

TurnOff – Switched Light only. Pulses when the light is turned off in Crestron Home.

IntensityRaise – Goes high when the Brighter button is held in the Crestron Home room. Goes low when the Brighter button is released.

IntensityLower – Goes high when the Dimmer button is held in the Crestron Home room. Goes low when the Dimmer button is released.

CctRaise – Goes high when the Cooler button is held in the Crestron Home room. Goes low when the Cooler button is released.

CctLower – Goes high when the Warmer button is held in the Crestron Home room. Goes low when the Warmer button is released.

Intensity – Reflects the value of the intensity slider in the UI. given in full scale 0-65535 representing 0-100%.

COPYRIGHT © 2026 ULTAMATION LIMITED

Cct – Correlated colour temperature. Reflects the value of colour temperature slider in the UI. This will be within the range defined by CctMin and CctMax.

CctMinFb – This gets set to the value of the CctMin parameter at startup.

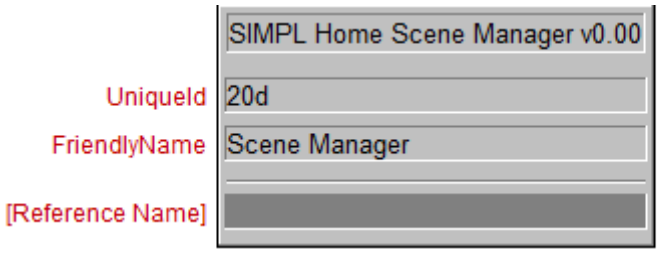
CctMaxFb – This gets set to the value of the CctMax parameter at startup.

Hue – Reflects the value of the hue slider in the UI. given as the value in degrees multiplied by 10. E.g., 120.5° will be output as 1205. Range is 0-3600.

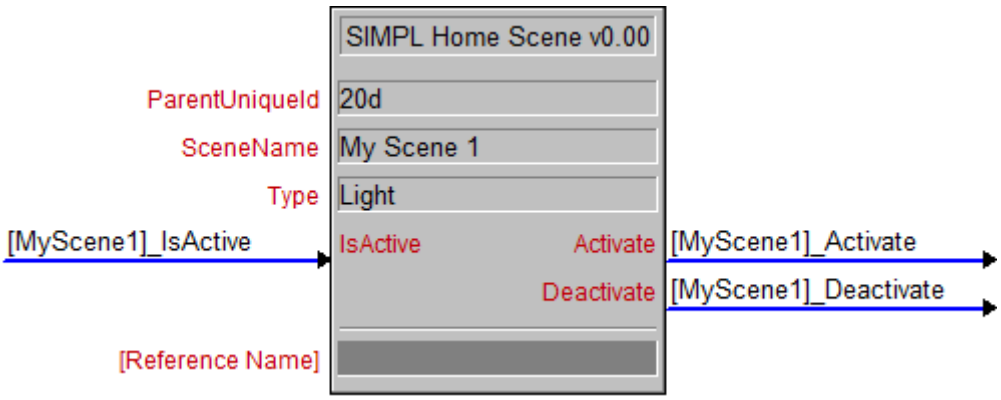
Saturation – Reflects the value of the saturation slider in the UI. given in full scale 0-65535 representing 0-100%.

COPYRIGHT © 2026 ULTAMATION LIMITED

Scene Manager Example



The Scene Manager is used to create and control scenes. A Crestron Home scene can be created by linking a Scene module to a Scene Manager via the Manager's unique Id. A scene will appear in the Crestron Home room that its Scene Manager is added to. The Scene Manager is not visible to the end-user.



Parameters

ParentUniqueId – The Unique Id of the Scene Manager this scene will be linked to.

SceneName – The name the scene will be given in Crestron Home.

Type – The type of scene. This effects how Crestron Home displays it.

Inputs

IsActive – Set high to display the scene as active in Crestron Home. Set low to display the scene as inactive in Crestron Home.

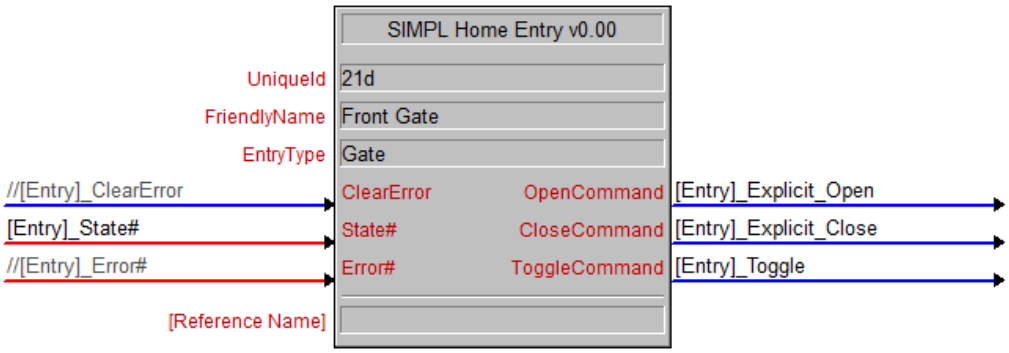
Output

Activate – Pulses when the scene is pressed whilst inactive in Crestron Home.

Deactivate – Pulses when the scene is pressed whilst active in Crestron Home.

COPYRIGHT © 2026 ULTAMATION LIMITED

Entry Example



Parameters

FriendlyName – The name the entry will be given in Crestron Home.

EntryType – The type of entry (Gate or Garage Door). This changes the icon shown in Crestron Home.

Inputs

ClearError – Pulse to clear the error value, setting it to none.

State# - This should be a value between 0-5. The possible values are as follows:

0	Closed
1	Open
2	Partially Open
3	Opening
4	Closing
5	Moving

This state will be shown on the entry tile in Crestron Home.

Error# - This should be a value between 0-3. The possible values are as follows:

0	Jammed
1	Object In Path
2	Manually Stopped
3	Unknown

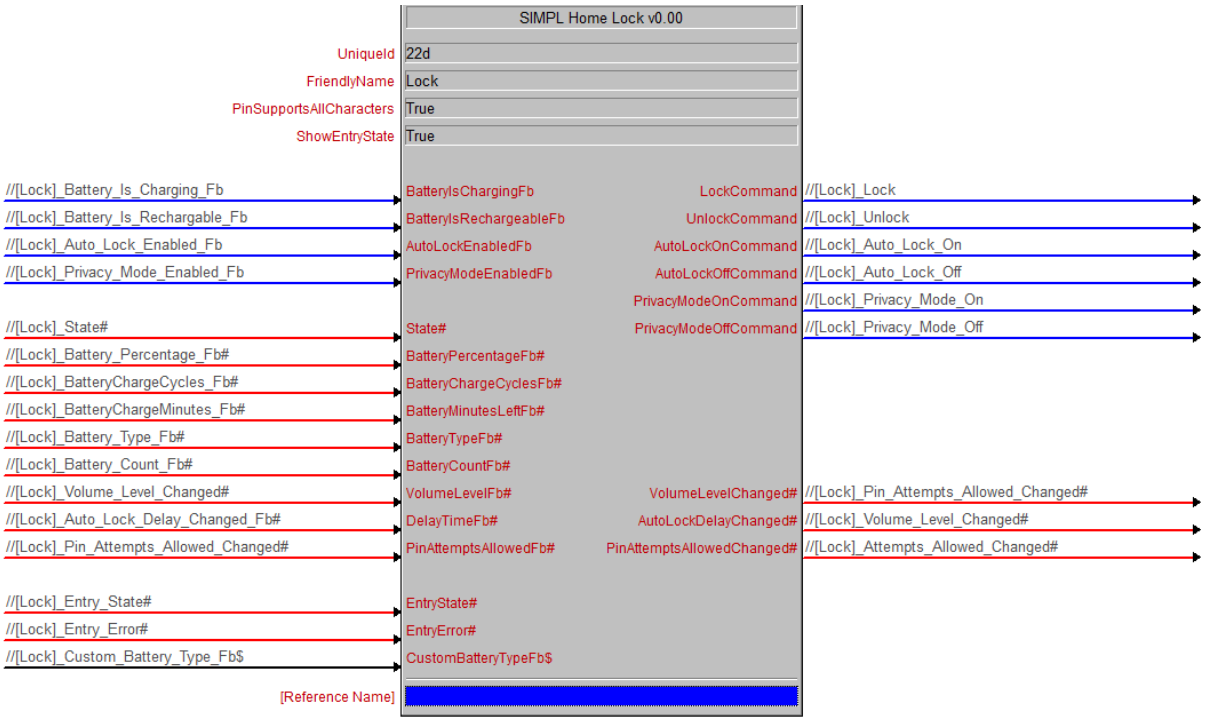
This error will be shown on the entry tile in Crestron Home.

Output

OpenCommand, CloseCommand, and ToggleCommand – When the corresponding command is pressed in Crestron Home, these signals will pulse. You can then send each signal to your entry system to action the command.

COPYRIGHT © 2026 ULTAMATION LIMITED

Lock Example



Parameters

PinSupportsAllCharacters – True or False depending on if the pin code being used supports all character types or only numbers.

ShowEntryState - True or False depending on if you would like to show the Open/Close state of the connected device, if supported by the device.

Inputs

BatteryIsChargingFb – Set high when battery is charging, low otherwise.

BatteryIsRechargableFb – Set high if the battery is rechargeable.

State# - A value between 0 and 4 that will be displayed on Crestron Home showing the status of the lock as follows:

0	Locked
1	Unlocked
2	Locking
3	Unlocking
4	Jammed

BatteryPercentageFb# - The battery percentage remaining on the device.

COPYRIGHT © 2026 ULTAMATION LIMITED

BatteryChargeCyclesFb# - The number of charging cycles the battery has completed.

BatteryTypeFb# - A number between 0 and 7 corresponding to the following battery types:

0	AAA
1	AA
2	C
3	D
4	N
5	NineVolt
6	Proprietary
7	Other

BatteryCountFb# - The number of batteries that the device uses.

VolumeLevel# - The volume of the lock when pressing on the keypad.

DelayTime# - The time after opening the lock will automatically lock itself, only applicable to devices which support Auto Locking.

EntryState# - One of the following values to show the state of the connected door:

0	Closed
1	Open
2	Partially Open
3	Opening
4	Closing
5	Moving

EntryError# - One of the following values to show any errors related to the door:

0	Jammed
1	Object In Path
2	Manually Stopped
3	Unknown

CustomBatteryTypeFb$ - A string which represents the type of battery being used if “Other” was previously selected.

COPYRIGHT © 2026 ULTAMATION LIMITED

Outputs

LockCommand, **UnlockCommand**, **AutoLockOnCommand**, **AutoLockOffCommand**, **PrivacyModeOnCommand** and **PrivacyModeOffCommand** will all pulse when the corresponding command is pressed in Crestron Home. You can use this command to send the signal to your Lock system to action the device.

AutoLockOnCommand and **AutoLockOffCommand** are only applicable to devices that support Auto Locking. If your device does not support this simply place a "//" into the signal name.

AutoLockOnCommand, AutoLockOffCommand, PrivacyModeOnCommand and **PrivacyModeOffCommand** are all programmable actions available in Crestron Home. Privacy Mode prevents the device from unlocking while enabled.

AutoLockDelayChanged# - An analog signal representing the most recent output from the programmable action which changes the Auto Lock time. Only available for devices which support Auto Locking.

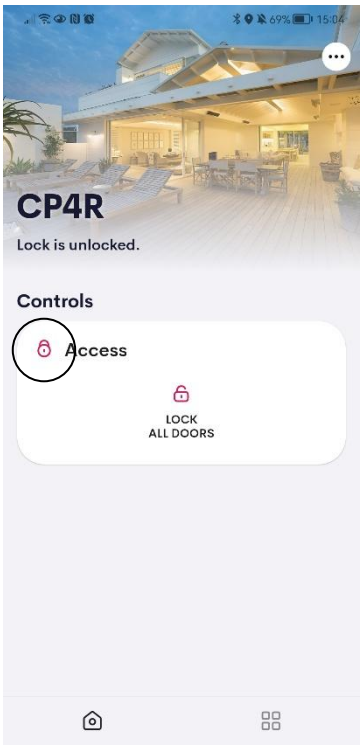
Settings

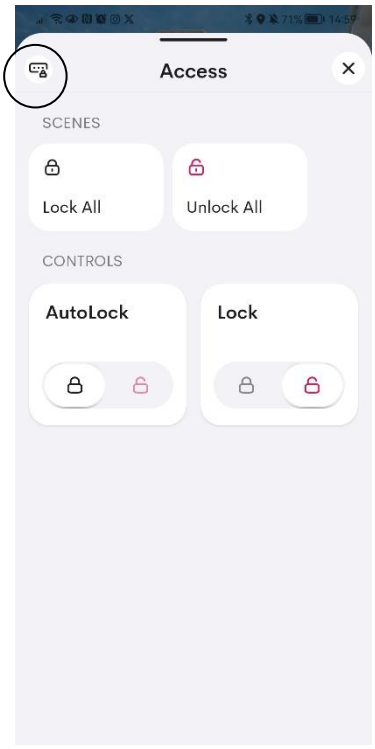
In order to make full use of this module and access all the lock settings you need to set up Users as shown in this guide: https://docs.crestron.com/en-us/8525/Content/CP4R/Operation/Settings/Settings-Users.htm

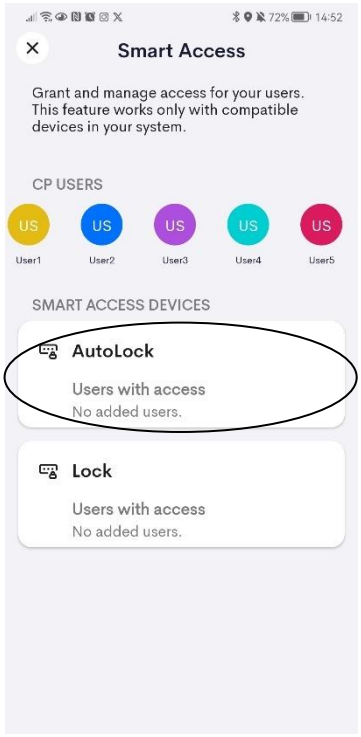
Afterwards you can click on the Access tile on the home screen to bring up the smart access section. Then click on the lock you want to edit the settings for and then the cog in the top right-hand corner.

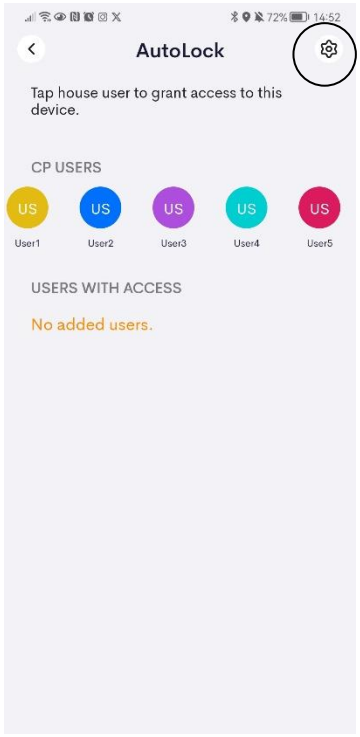
Users are not supported by our driver. You can add, remove and modify the users in the app but they will not be there on reboot and do not functionally do anything.

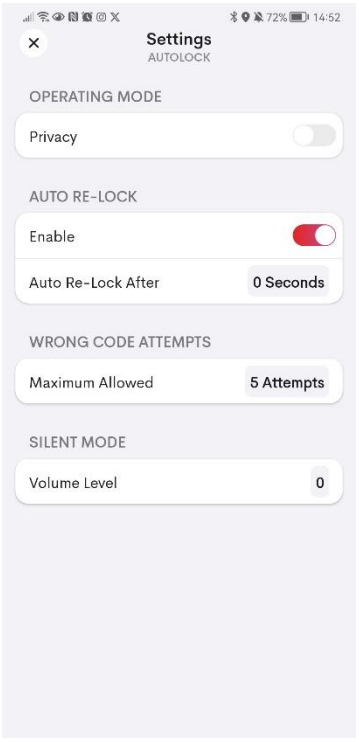
COPYRIGHT © 2026 ULTAMATION LIMITED











COPYRIGHT © 2026 ULTAMATION LIMITED

NVX MultiView Example

At the time of writing, Crestron Home doesn't offer a built-in way to manage solutions with multi-view video to support the new NVX capabilities offered by the NVX-384 and NVX-385 devices.

SIMPL Home offers a solution to making this powerful feature available to homeowners. With large format TVs and projectors being the norm, there are innumerable use-cases for displaying different video sources on a single screen, whether it be keeping an eye on the news whilst simultaneously enjoying a sporting event, or a gaming party for up to 6 consoles, or a CCTV pop-up when someone's coming up the driveway - the utility of multi-view is impossible to deny.

This document explains how Ultamation have devised a user-friendly way to exploit multi-view within Crestron Home.

The high-level solution is as follows:

- In the multi-view room, create multiple displays - these are the "windows" of the multi-view solution. The user will send sources to these videos as normal, and the system will then configure the multi-view solution automatically.
- Add a secondary custom processor running a custom SIMPL program with SIMPL Home.
- SIMPL Home presents two objects to Crestron Home - a "virtual" AV switcher and an extension tile for layout control.
- The sources are routed to the virtual switcher which then controls the NVX logic in the custom SIMPL program.
- The display control can either be handled on the SIMPL side, or as the display driver on Window A.
- Custom logic on the SIMPL side then configures the NVX layout, source routing and transitions.
- The extension tile is used to both manually and automatically set the layout selection. It's also used to set the transition for each window.
- The extension tile has programmable actions that means the layout and transitions can be changed via Quick Actions.

COPYRIGHT © 2026 ULTAMATION LIMITED

Recommended Prerequisites

The following details for setting up the DM-NVX cards were correct at the time of developing this integration and should be considered as guidance rather than followed verbatim. A DMF-CI-8 with 8 DM-NVX-384C cards was used. This setup worked for development purposes so will likely differ from real installations. Please refer to Crestron support for the latest details.

A Display that supports 4K 4:4:4 video signals must be used to support all the features of this module. Support for this may need to be enabled in the Display's settings. Text Overlay may not work without this.

- Connect the NVX cards to a switch that supports the necessary IGMP and Multicast settings
    - See DM NVX® Devices: Network Switch Configurations
    - Assign the ports connected to the cards to a new VLAN
    - Enable the necessary IGMP and Multicast settings on this VLAN
    - Assign this VLAN a static IP address. Setup the routing to this on the router
    - Choose a multicast range you want the NVX transmitter cards to use. Crestron recommend starting from 239.1.0.0. Each card will take three consecutive addresses (audio, video, multi-view)
- Login to each card's web portal
    - Go to Settings, Open "System Setup"
        - Set IGMP Support to the correct value for the network
        - Set the time zone
    - Open "Network"
        - Set the Primary and Secondary Static DNS
        - Disable DHCP
        - Set the IP address to one within the subnet defined for the switch
        - Set the Subnet Mask as 255.255.255.0
        - Set the Default Gateway to the static IP address of the new VLAN on the switch
    - Open "Stream"
        - Select the mode for this card (Receiver/Transmitter).
- For Transmitter cards
    - Open "Stream"

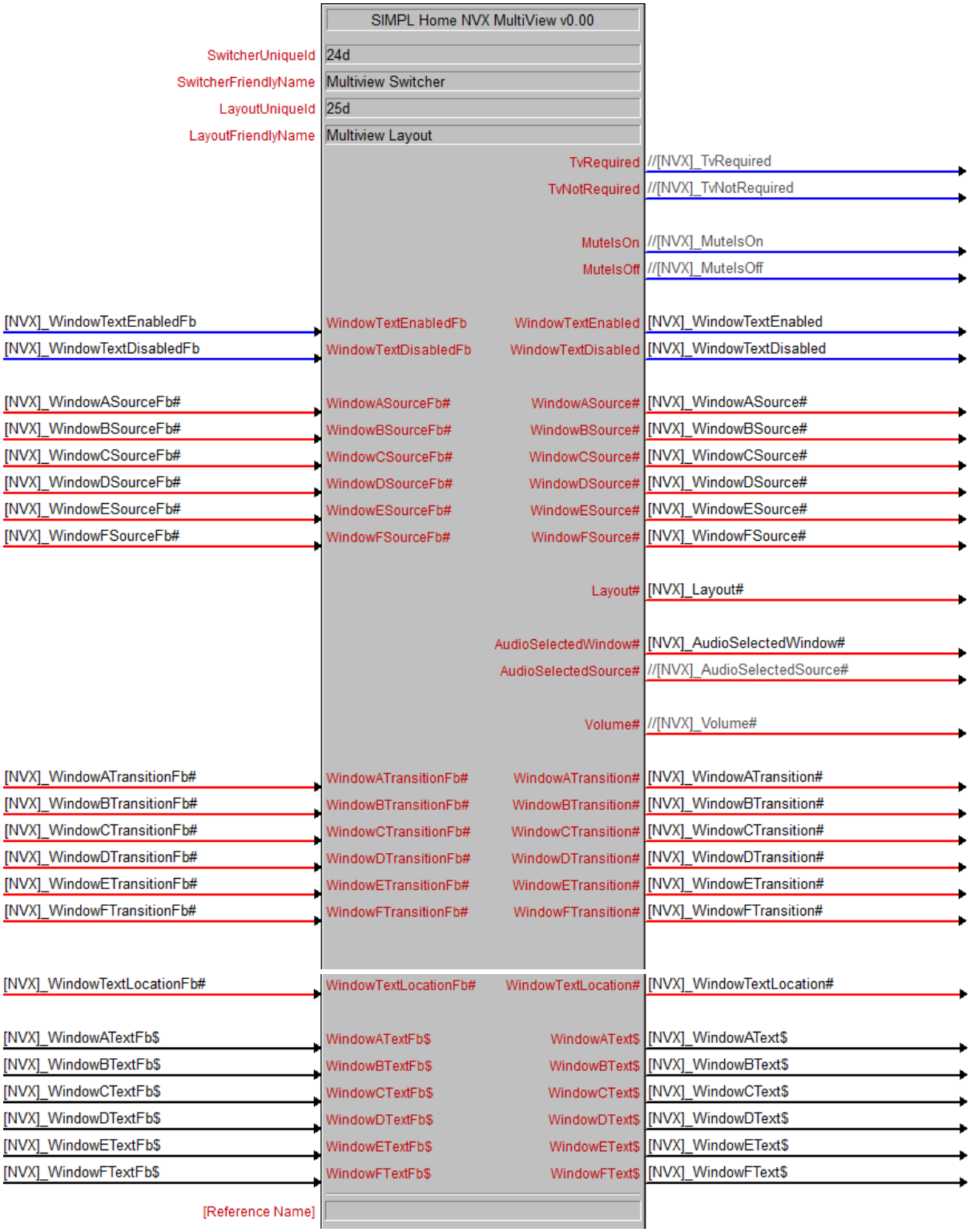
COPYRIGHT © 2026 ULTAMATION LIMITED

        - Set the Multicast Address to within the range that was selected previously, leaving space for the three addresses each card will use
        - Enable Multi-View
        - Enable Auto Initiation
    - Open “Routing”
        - Enable Automatic Input Routing
        - Set the other settings as desired
- For Receiver cards
    - Open “Stream”
        - Enable Preview Output.
    - Open “Routing”
        - Enable Automatic Input Routing
        - Set the Video Source to Stream
        - Set the audio settings as desired
        - Enable Multi-View
    - Open “Subscriptions”
        - Subscribe to the streams of the transmitters and then order them as desired. This order must be matched in Crestron Home later
    - Open “Routing”
        - Test the routing. Select the desired number of windows, the layout, and which transmitters you want connected to each window, and which transmitter you want the audio from
    - Open “Stream”.
        - You should now see a preview of the selected Multi-View setup. The Stream Location and Multicast Address should be populated with the corresponding values from the transmitter going to the primary window. The Status should be “Stream started”. The Preview window should show what will show on the screen connected to the receiver

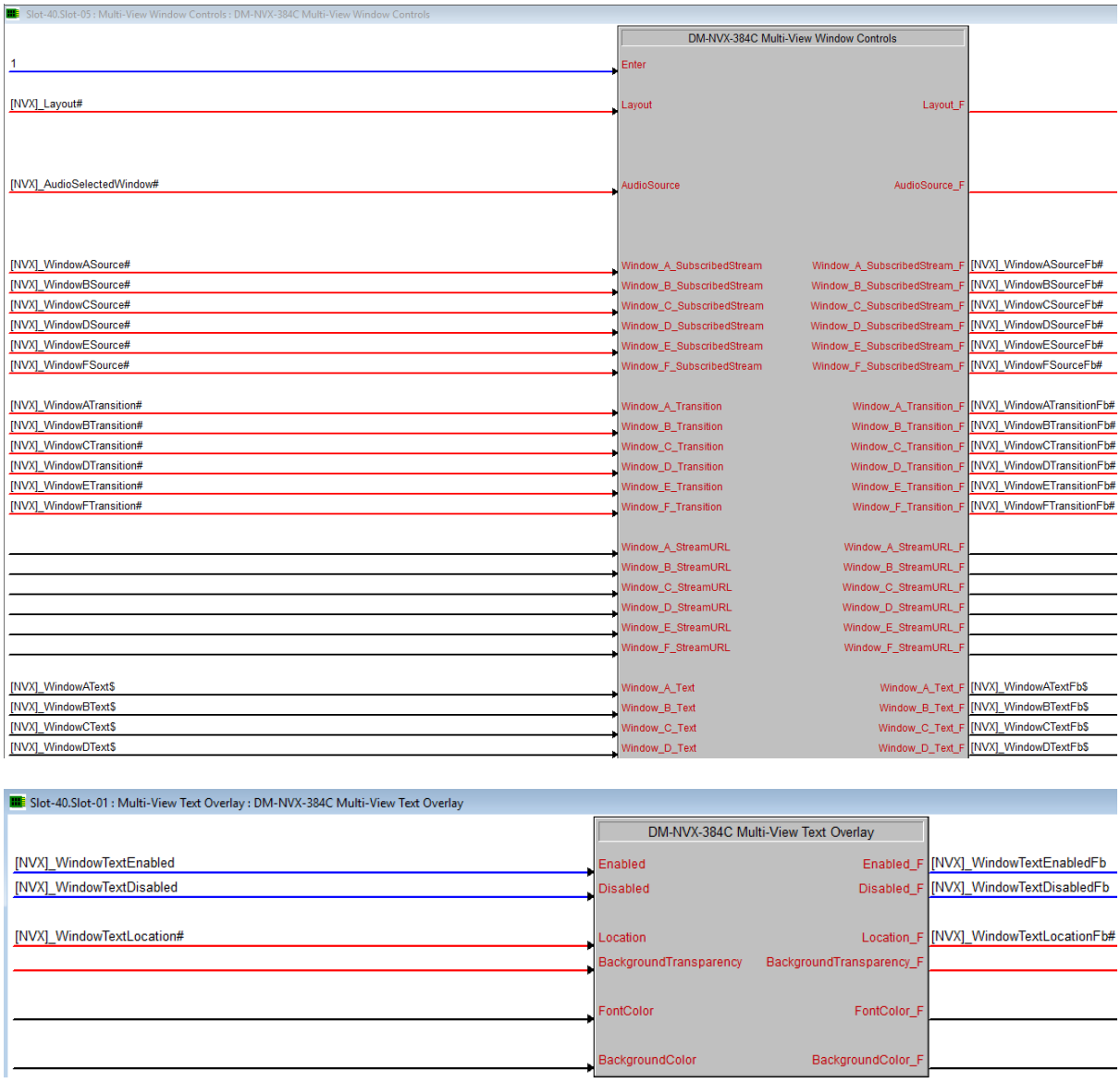
Setup in SIMPL

The SIMPL Home NVX MultiView module needs to be connected to the DM-NVX Multi-View Window Controls found under MultiviewControls Setup on the DM-NVX device in the hardware tree. See the inputs and outputs below for more details.

COPYRIGHT © 2026 ULTAMATION LIMITED



COPYRIGHT © 2026 ULTAMATION LIMITED



Parameters

SwitcherUniqueId – The unique ID that will be passed to the Switcher module.

SwitcherFriendlyName – The name that will be given to the Switcher device in Crestron Home.

LayoutUniqueId – The unique ID that will be passed to the Switcher module.

LayoutFriendlyName – The name that will be given to the Layout tile in Crestron Home.

Inputs

WindowTextEnabledFb – High indicates the text overlay is enabled. This should be connected to the Enabled_F output on the DM-NVX Multi-View Text Overlay.

COPYRIGHT © 2026 ULTAMATION LIMITED

WindowTextDisabledFb - High indicates the text overlay is disabled. This should be connected to the Disabled_F output on the DM-NVX Multi-View Text Overlay.

Window[X]SourceFb# - Feeds back the one-based index of the subscribed source for that window. A value of 0 indicates no source has been selected and that the window is inactive. These should each be connected to the corresponding Window_X_SubscribedStream_F outputs on the DM-NVX Multi-View Window Controls.

Window[X]TransitionFb# - Feeds back the code of the selected transition for that window. These should be connected to the corresponding Window_X_Transition_F outputs on the DM-NVX Multi-View Window Controls.

WindowTextLocationFb# - Takes the code for the current text position to update selection in the Layout Tile. This should be connected to the Location_F output on the DM-NVX Multi-View Text Overlay.

Window[X]TextFb$ - Takes the current text for the window to update text entry in the Layout Tile. These should each be connected to the corresponding Window_X_Text_F output on the DM-NVX Multi-View Window Controls.

Outputs

TvRequired – High whilst at least one window is active. Low when no windows are active. Can be used for controlling the TV displaying the MultiView.

TvNotRequired – Low whilst at least one window is active. High when no windows are active. Can be used for controlling the TV displaying the MultiView.

MuteIsOn – High when the switcher's audio output is muted in Crestron Home. Low when the audio is not muted.

MuteIsOff – Low when the switcher's audio output is muted in Crestron Home. High when the audio is not muted.

WindowTextEnabled – High if the text overlay is enabled and low otherwise. This should be connected to the Enabled input on the DM-NVX Multi-View Text Overlay.

WindowTextDisabled - High if the text overlay is disabled and low otherwise. This should be connected to the Disabled input on the DM-NVX Multi-View Text Overlay.

COPYRIGHT © 2026 ULTAMATION LIMITED

Window[X]Source# - Outputs the one-based index of the source selected for the respective window in Crestron Home. A value of 0 indicates no source has been selected and that the window is inactive. These should each be connected to the corresponding Window_X_SubscribedStream input on the DM-NVX Multi-View Window Controls.

Layout# - Outputs the code for the layout selected in the Layout Tile in Crestron Home. This should be connected to the Layout input on the DM-NVX Multi-View Window Controls.

AudioSelectedWindow# - Outputs the one-based index of the window selected for the audio in Crestron Home. This should be connected to the AudioSource input on the DM-NVX Multi-View Window Controls. A value of 0 indicates no window has been selected.

AudioSelectedSource# - Outputs the one-based index of the source selected for the audio in Crestron Home. A value of 0 indicates no source has been selected.

Volume# - Outputs the current full scale (0-65535) volume value of the Switcher's audio output in Crestron Home.

Window[X]Transition# - Outputs the code of the selected transition for the respective window in Crestron Home. These should each be connected to the corresponding Window_X_Transition input on the DM-NVX Multi-View Window Controls.

WindowTextLocation# - Outputs the code for the text position selected in the Layout Tile. This should be connected to the Location input on the DM-NVX Multi-View Text Overlay.

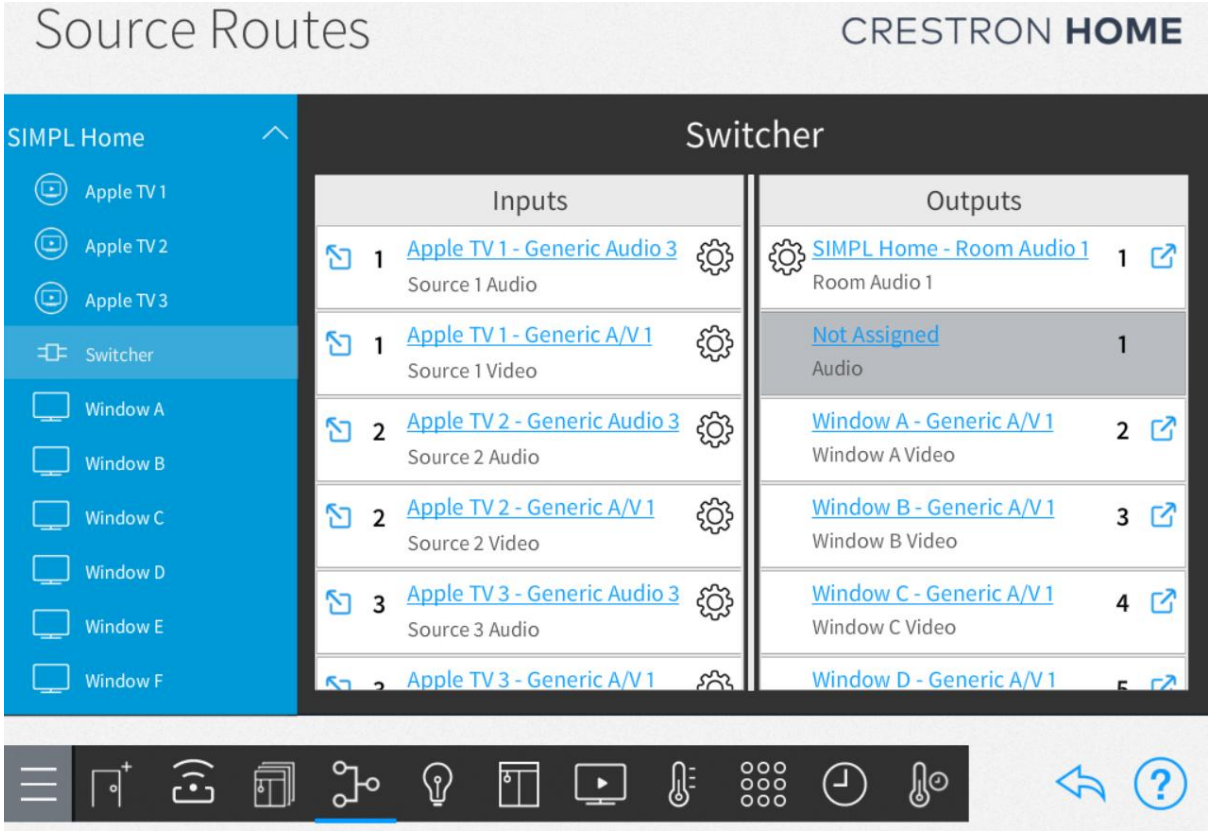
Window[X]Text$ - Outputs the text that has been entered into the text entry in the Layout Tile. These should each be connected to the corresponding Window_X_Text input on the DM-NVX Multi-View Window Controls.

Setup in Crestron Home

Add the "SIMPL Home NVX MultiView Layout" and "SIMPL Home NVX MultiView Switcher" to a room. In the routing menu, Connect the sources to the switcher **in the same order as the stream subscriptions are on the receiver.** Uncontrolled A/V Sources can be used in place of sources that won't be controlled by Crestron Home. Connect the Window Video outputs to Uncontrolled A/V Displays. If the

COPYRIGHT © 2026 ULTAMATION LIMITED

display is to be controlled in Crestron Home, then connect it to the Window A Video output instead of an Uncontrolled A/V Display. In the following example, these are named Windows A-F. All these displays should be in the same room.

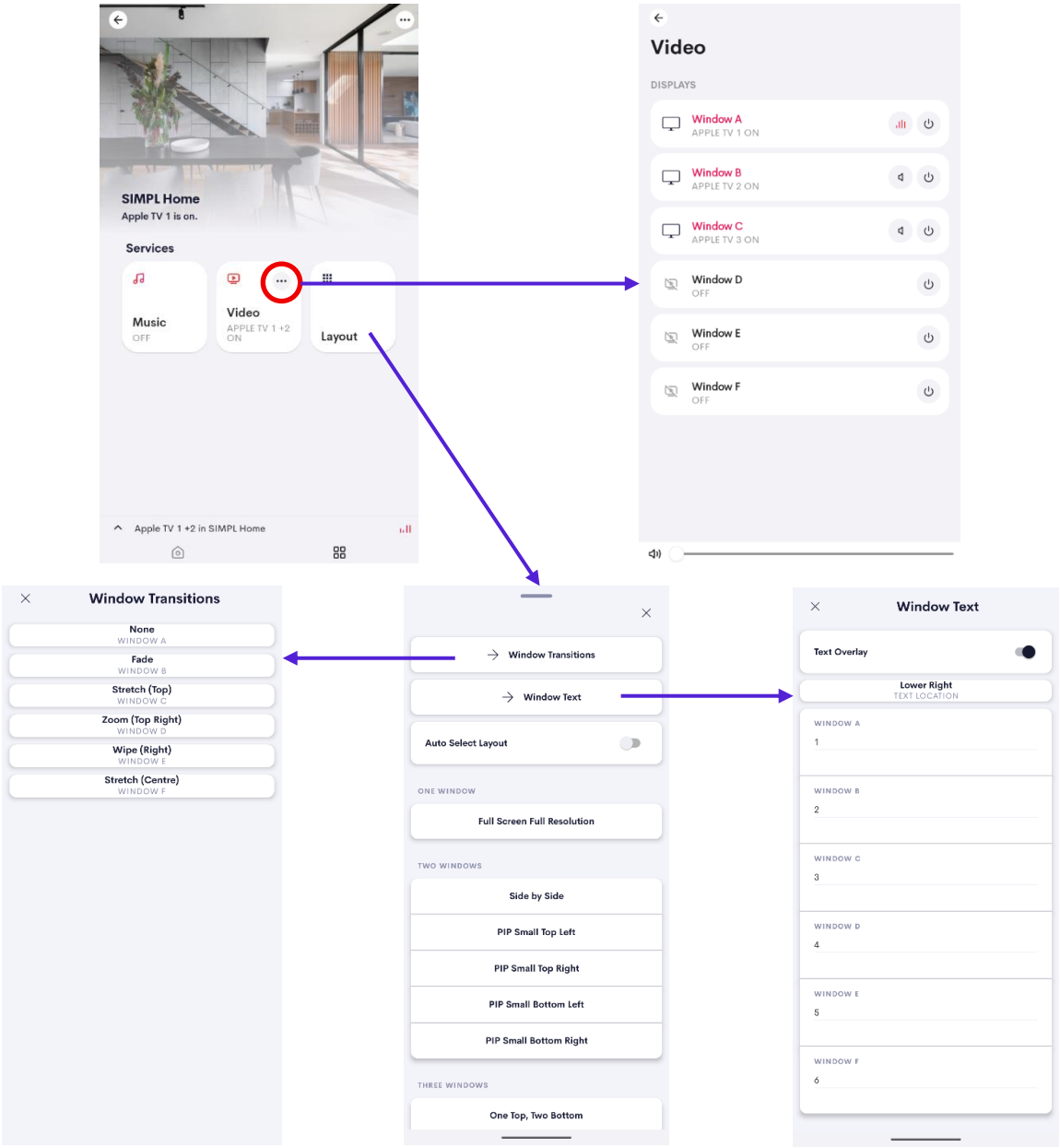


The user can then select the source for each window and turn them off as desired. Selecting the audio can also be done here. In the below example, Window A is connected to Apple TV 1 and has been selected as the audio source. A source can be selected for the audio output by pressing the speaker icon next to any window it is routed to.

The Layout Tile is used for changing the MultiView layout, transitions and text overlay for each window. It shows the layout options for the current number of active windows or less. For example, in setup shown above there are 3 active windows so the Layout Tile will only show options for 1, 2, and 3 window layouts. Routing a source to Window D would then make the 4 window layouts show on the Layout Tile. The “Auto Select Layout” option will automatically change the layout when a window is activated or deactivated. It will change to the first layout option for the number of currently active windows. In the setup above, activating Window D would result in the “Two Top, Two Bottom” option being selected. Deactivating Window D would then cause “One Top, Three Bottom” to be selected. This Auto

COPYRIGHT © 2026 ULTAMATION LIMITED

Select Layout option is enabled by default and can be turned on or off at the top of the Layout Tile. A layout can be manually selected regardless of whether Auto Select Layout is on or off. The top buttons open other menus where the transitions and text overlay can be set. See the screenshots below.



COPYRIGHT © 2026 ULTAMATION LIMITED

Crestron Console Diagnostic Commands

Should any diagnostics be required on the SIMPL Windows side, a couple of user commands have been provided.

To invoke a command in the Crestron Console, connect to the processor and determine which program slot SIMPL Home is running in. You will need to append the program slot number to the "ucmd" command to route it to the correct slot. For example, if SIMPL Home is running in slot 4, you would enter

`ucmd:4 "help"`

to invoke the SIMPL Home "help" command.

The available commands are as follows:

info	Provide a summary of the SIMPL Home host's state. At present this simply displays the number of active client connections.
clients	List the current state of all Crestron Home clients. This is primarily useful for SIMPL MDU solutions where multiple Crestron Home clients will connect into a single SIMPL Windows instance.
ep	List the registered endpoints (SIMPL objects). This may be useful in identifying duplicate endpoint ids or checking to see why an endpoint isn't shown in the Crestron Home instance. For example, in SIMPL MDU, each endpoint also has an "Instance ID".
orphans	For debugging only – this will list the internal state of SIMPL modules that share a parent/child relationship. Only useful when discussing issues with Ultamation.

SIMPL Home will also emit various log messages to the processor's error log giving helpful info about client connection changes and so on. Please provide any error log output if you need to report an issue to support.

COPYRIGHT © 2026 ULTAMATION LIMITED

Version Checking

To ameliorate problems with mismatched versions, starting with v2.00 SIMPL Home will now validate the executable versions between Crestron Home and the SIMPL Windows modules. Versions pre-dating v2.00 will simply report as v0.00.

Major versions of SIMPL Home will NOT be backwardly compatible with earlier major version (i.e. v1.06 Crestron Home endpoints are NOT compatible with v2.00 SIMPL Home SIMPL modules), however we will endeavour to retain backwards compatibility between minor releases (accepting that new features will not be available of course!)

After initial boot up, please check the processor's error log using Crestron Toolbox console for BOTH Crestron Home and SIMPL Windows ends of the solution to verify that no version warnings are displayed.

The "ERR" command will display this information. This is NOT the same as the internal Crestron Home logging system, which was dreamed up by someone with zero diagnostics experience.

COPYRIGHT © 2026 ULTAMATION LIMITED

Installation

The driver can be found under:

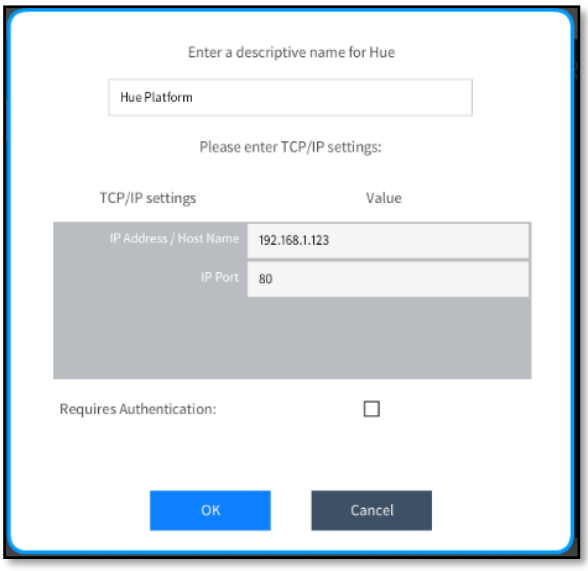
Drivers > Platform > Ultamation

Adding the Driver

Make sure you have the correct platform driver(s) for the SIMPL endpoints you are wanting to connect to. It may be the case that you need an instance of "SIMPL Home" and an instance of "SIMPL Home Entities" to access all the endpoints in one SIMPL program. Please refer to the section "OS4 onwards – Which platform driver to use" at the top of this document. Multiple platform drivers can connect to one SIMPL program.

Ensure a room is selected, and then click on the '+' against the appropriate device. The room can be ANYWHERE in the system and there will not be a visual tile added for the platform. To keep things organised you could add this device to a plant or equipment room.

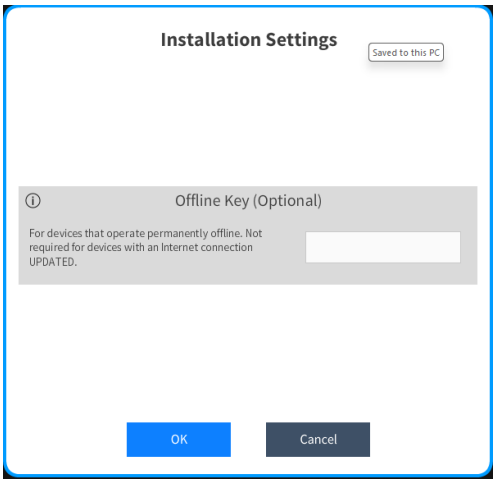
You will be asked to provide a descriptive name – this can be anything you wish and the IP address of the Crestron Processor running the SIMPL Program.



The port can be left at the default value of 33333 unless your configuration specifically requires something different (in which case you will know what it should be set to).

COPYRIGHT © 2026 ULTAMATION LIMITED

The driver is licensed via the cloud, but an optional "offline" key can be requested under special circumstances. Please FIRST refer to the "Licensing" section below and then contact support@ultamation.com if you REQUIRE offline activation. The same offline key can be used for multiple driver instances in the same system.



If you do not have a cloud licence and do not enter a valid offline licence key, the system will function for ONE hour and then control will be suspended. The offline licence key can be entered at any time after initial setup without having to reconfigure each tile.

COPYRIGHT © 2026 ULTAMATION LIMITED

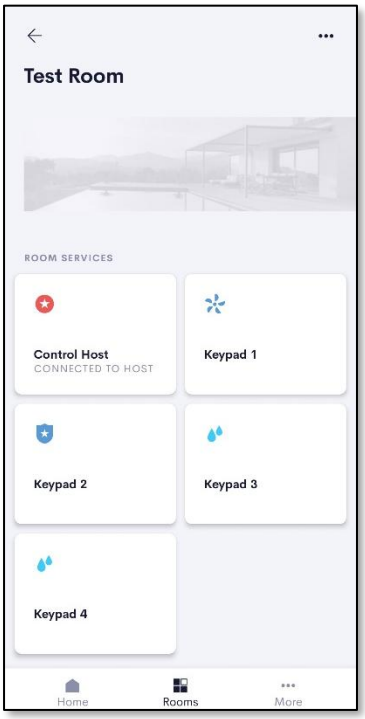
Adding Devices

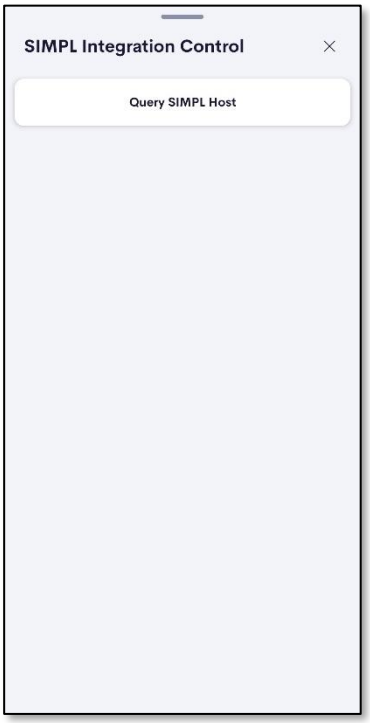
Once the SIMPL Home platform driver has been added to the system, it will start communicating with the SIMPL Windows Program and attempt to request information about the system.

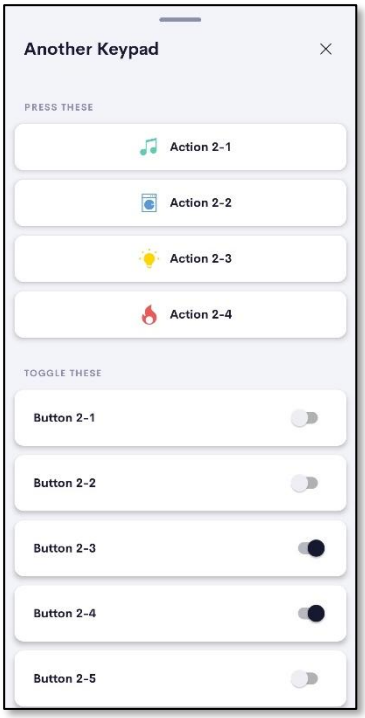
All the devices which have been setup within SIMPL should now be visible within the SIMPL Home folder. Before OS4, this would appear at the same location as the platform driver. In OS4, this has moved to Pair Devices > Managed Platforms.

Device Setup Demonstration

In the example below, 4 keypads were added to a room along with the Control Host device. The devices display as tiles within the UI as normal within Crestron Home. The keypads provide a wealth of functionality, including action functions, button toggles and sliders. The amount of customisation is determined by the user within SIMPL Windows.







COPYRIGHT © 2026 ULTAMATION LIMITED

Creating Translation Files

To create a translation file, connect to the Crestron Home processor using an FTP client and navigate to user/Data/UsedThirdPartyDrivers. There should be a folder named “ultamation.simplhome.tcpclient.ultamationlimited”, which will contain one or more folders with names corresponding to version numbers of the SIMPL Home. Open the folder with the version currently in use. Inside, open the folder for the device type you want to create a translation for, and copy the en-US.json file to your PC. You can then change the translation values and save the json file as a new json file with the correct language code as the name.

For example, to translate the thermostat to French (France), the following steps would be taken:

- Open the tstat folder and copy en-US.json.
- Change Humidity to Humidité, etc.
- Save the file as fr-FR.json.

Now, copy the new json file to the same folder that the en-US.json file came from, and repeat for all languages and devices you wish to create translations for. Reboot the Crestron Home processor, and the translations should then be usable.

COPYRIGHT © 2026 ULTAMATION LIMITED

Quick Actions & Sequences

Several functions are exposed to sequences via Crestron Home's quick actions for the basic and advanced tiles.

Trigger Action

- The user can trigger an action with a sequence, this will perform whatever action has been set within the Simpl application
- Actions which have been set will appear within the folder Trigger Action for the user to select

Set Value

- This allows for a discrete value to be set on a slider.
- The user can select the device or feature which they wish to control, for example a bulbs brightness or hue, from the drop-down menu. Then, a discrete value can be set for the slider. This is within the standard range of

Slider Toggle

- If enabled within Simpl Windows, the user can use this quick action to select a particular slider and set its state. There are 3 states, Set, Reset and Toggle.
    - Set will toggle the state high
    - Reset will toggle the state low
    - Toggle will toggle to the opposite of the current state

Toggle Button

- If enabled within Simpl Windows, the user can use this quick action to select a particular button from the drop down and set its state. There are once again 3 states, Set, Reset and Toggle.
- Set will toggle the state high
- Reset will toggle the state low
- Toggle will toggle to the opposite of the current state

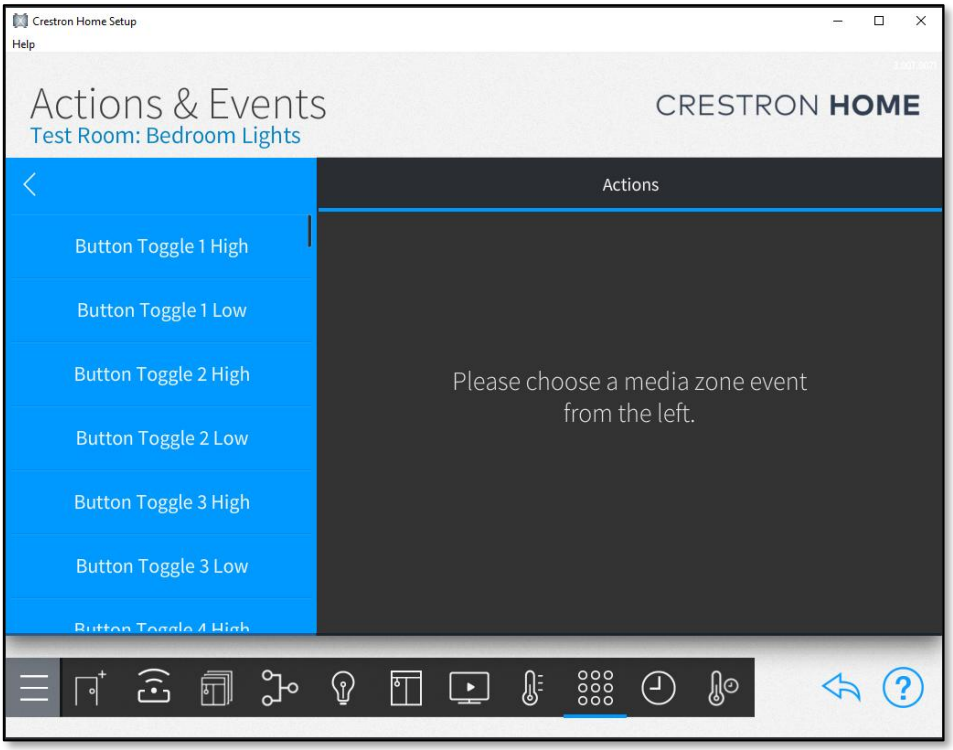
Trigger Tile

- This quick action will perform whatever function a tile press has been set to perform within the Simpl Windows Application

COPYRIGHT © 2026 ULTAMATION LIMITED

Events

As there is an abundance of possibility within SIMPL Home, there are several events accordingly. For each button state or slider toggle state, there is an event which can be triggered high or low. There is a maximum number of 20 buttons and 20 toggles, each has an event representing their two states (high or low). This means there is a total of 80 events which can be utilised within Crestron Home.



COPYRIGHT © 2026 ULTAMATION LIMITED

Scripted Tile Programming Guide

The scripted tile is one of two mechanisms, along with the “Modular Tile” we have provided to give SIMPL Windows programmers the ability to create bespoke Extension devices for Crestron Home without (too much!) exposure to the nuts and bolts of the Crestron Certified Drivers SDK.

Download all sample files from the Ultamation Website:

https://shop.ultamation.com/index.php/hikashop-category-information-menu-129/product/159-simpl-home-driver-for-crestron-home

Having said that, the Scripted Tile does require some familiarity with the way Crestron Home extension devices are defined for UI presentation. To that end, you will need to read the UI-Files topic in the Crestron CCD SDK documentation (link below).

https://sdkcon78221.crestron.com/sdk/Crestron_Certified_Drivers_SDK/Content/Topics/Driver-SDK-V1/Create-a-Driver/Device-Types/Extensions/UI-Files.htm

There are three essential components to using the Scripted Tile.

1. UI Definition
   Under the hood, all Crestron Home extensions require an XML definition of the user interface components to display and interact with. These elements, such as buttons, text boxes, sliders and object lists interact with the underlying system via “commands” (a simple trigger, possibly with an argument), and “properties” which are mapped onto state/signals in SIMPL Windows.

   As an example, the simple “Toggle Slider” in Crestron Home can make use of a number of properties – digital properties for the toggle, visibility or enabled states, an analog property for slider value itself (with a defined, unsigned, range) and a serial properties for the main and secondary labels of the control.
   These property “bindings” are specified in the XML UI definition file, and match up with corresponding property objects (described below) in the SIMPL Windows program.

COPYRIGHT © 2026 ULTAMATION LIMITED

The UI definition file is exactly the same as would be used in a traditional Crestron Home extension device, including support for multiple layouts, and as such can be used as a tool in the rapid prototyping of Crestron Home extensions as well as a way to bridge Crestron Home and SIMPL.

A malformed XML definition will often result in Crestron Home failing to display the extension tile. As such, we **STRONGLY** recommend you create the XML definition file using an editor which supports the referenced Xml Scheme Definition file provided by Crestron as this will validate much of the structure of the file before you start to debug any issues in Crestron Home/SIMPL.

This file, once verified for correctness, is copied to the SIMPL Windows processor to a location of your choosing. The filename is arbitrary. Our example program expects the file "UiExample.xml" to be present in the processor's NVRAM folder, though the actual file path is defined in the root SIMPL Scripted module.

2. Translation File
   At the time of release, Crestron Home does not properly support multi-lingual interface, though the CCD SDK does include support locale specific translation files. These files are essentially a table of look-up keys which map onto string literals for display in Crestron Home. E.g. "labelHeader" => "My Extension"

   The translation files are defined using JSON and, again, we STRONGLY recommend that you use an editor, or external "linter" tool, that will validate your JSON structure and highlight any errors. If you attempt to load an invalid translation file, Crestron Home may well respond in a sub-optimal way.

   As described in the Crestron documentation, you will refer to these translation keys using the "^" prefix. i.e. if you wish to display "My Extension" in your Scripted Tile, you would assign "^labelHeader" to a serial property which is bound to a string property in the UI XML. It sounds more complicated than it actually is!

COPYRIGHT © 2026 ULTAMATION LIMITED

As for the UI Definition file, this translation file is copied to the SIMPL Windows processor to a location of your choosing. Again, the filename is arbitrary. Our example program expects the file "TranslationExample.json" to be present in the processor's NVRAM folder.

At present, only one language file is supported – though this is in line with Crestron Home's own constraints.

We cannot stress sufficiently, please ensure that these files are well formed and valid before loading. Crestron Home consumes these files with **very little** to say if there are any errors. If your Scripted tile doesn't appear on the page it is most likely because the XML or translation files are invalid. Please check the logs for the Crestron Home processor, but remember some errors are silently supressed by Crestron Home.

3. SIMPL Windows Modules

   As described above, each interactive element of the UI definition can either trigger an event in SIMPL Windows or be bound to a property. Each Scripted Tile is defined by a root object, as for all other SIMPL Home objects with a unique Id. Not only is this Unique ID important so that Crestron Home knows which endpoint is being updated, it is also the "Parent ID" for all other properties, commands, actions and events that sit logically below (looking from upside down!) the root.

   The next section will briefly describe each module in the Scripted Tile tool kit.

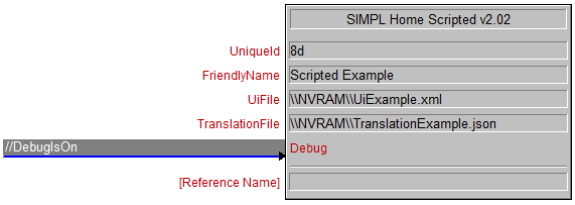
   Remember, the scripted tile does NOT describe UI controls themselves (in contrast to the Modular Tile) but simply described the properties used to drive elements described in the UI XML. This is how the Scripted tile is able to support any new UI component that is added to Crestron Home without an update, so long as the properties are compatible with the new UI element.

COPYRIGHT © 2026 ULTAMATION LIMITED

SIMPL Home Scripted

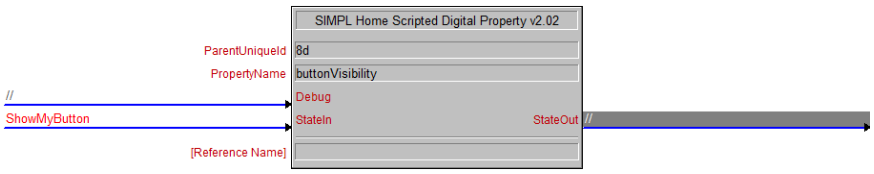
Although fairly simple, this object is required for each Extension Tile you wish to define in Crestron Home. For anything other than the most trivial or "one off" user cases, you will probably want to wrap the Scripted tile objects in a SIMPL User Module and parameterise the Unique Id so that multiple copies can easily be added to the program.

The SIMPL Home framework will ensure that where multiple tiles request the same UiFile or TranslationFile, the file operation is only carried out once to avoid excessive (slow) file I/O.



SIMPL Home Scripted Digital Property

The Digital property is used to drive or act on Boolean properties in the UI XML definition. This could be "read/write" properties such as a toggle button state, or "set only" properties such as the visibility or enable attributes of a control.



The example below is linked to the Scripted Parent ID 8, and binds a digital signal "ShowMybutton" to a Boolean property "buttonVisibility" such as the one defined in this XML snippet below:

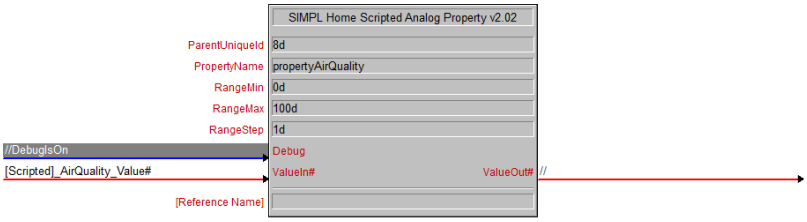
```
<toggle id="digitalExample" label="^labelToggle" value="{buttonState}"
visible="{buttonVisibility}" />
```

The button will be displayed with the text translation defined by "labelToggle".

SIMPL Home Scripted Analog Property

Similar to the Digital property, analog values can be bound to UI elements through the Scripted Analog Property. This object also defines the valid range for the property which is used for scaling in the Crestron Home UI.

COPYRIGHT © 2026 ULTAMATION LIMITED

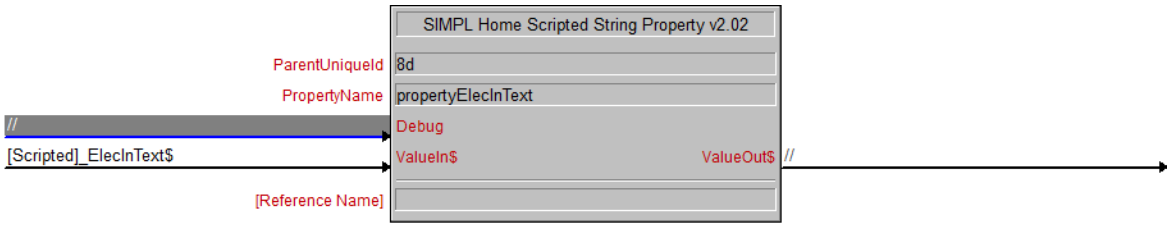


The example above sets a valid range of 0-100 with a granularity of 1 for a hypothetical air quality gauge. Please refer to the example program and files for more examples of how the bindings are mapped to Crestron modules.

Please note that not all UI elements in Crestron Home report property changes in the same way. The slider object only reports a change on release, while the toggle slider will track and report changes as the user makes adjustments.

SIMPL Home Scripted String Property

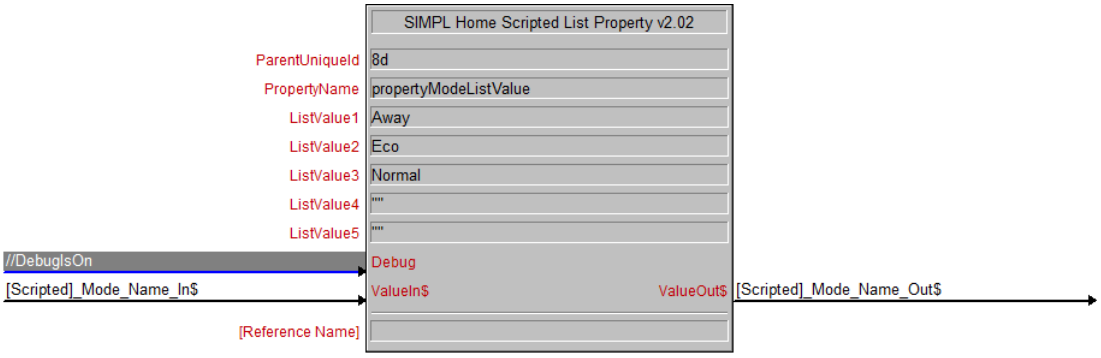
The String property is used to map serial strings onto text properties in Crestron Home. Usual Crestron Home rules apply to the string values sent – such that a ^ prefix will invoke the translation processing and icons must match the valid values defined in Crestron's built in iconography. (https://sdkcon78221.crestron.com/sdk/Crestron_Certified_Drivers_SDK/Content/Extension-Device-Icons.pdf)



SIMPL Home Scripted List Property

A close cousin to the Scripted String is the Scripted List Property. This module allows you to set a collection of list values (20 in total) for use with the segmented slider and selector button in Crestron Home. A segmented slider can only handle up to 5 values so use the 5 value version one for that. The example below shows 3 labels defined (blank labels mean "no value"). The ListValue[x] parameter is expandable to get to the number of values you require.

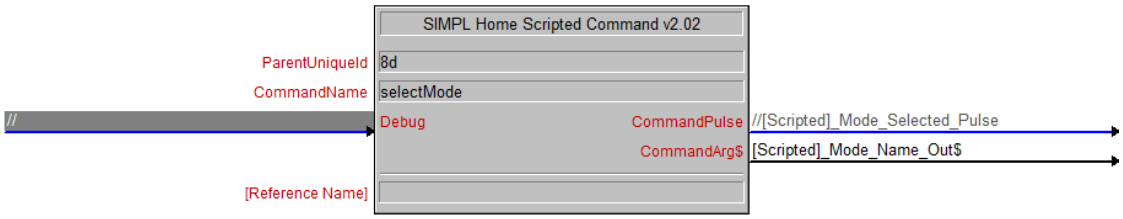
COPYRIGHT © 2026 ULTAMATION LIMITED



It appears that this control does NOT accept translation keys. Literal values must be used.

SIMPL Home Command

The Command "feedback only" module is provided to accommodate button actions.



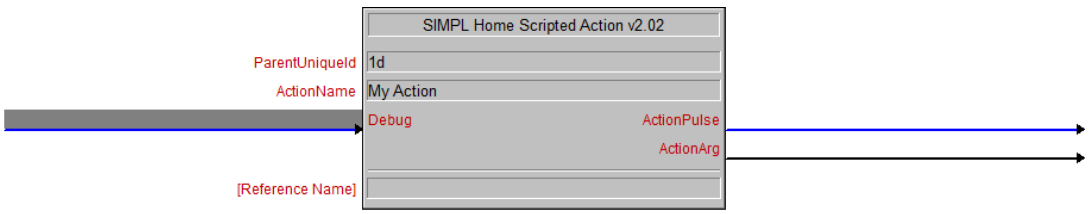
This example defines a command action called "selectMode" and here it is bound to a listbutton, one of the more complex object structure in Crestron Home extensions (more on this below).

An action can also include arguments. In SIMPL Home, we support a single argument.

```
<listbutton id="selectMode" … action="command:selectMode"
actionparameters="{.propertyModeName}" />
```

SIMPL Home Scripted Action

The Scripted Action provides a mechanism to programmable actions in Crestron Home to trigger events in SIMPL Home and as for commands, we support a single argument.



COPYRIGHT © 2026 ULTAMATION LIMITED

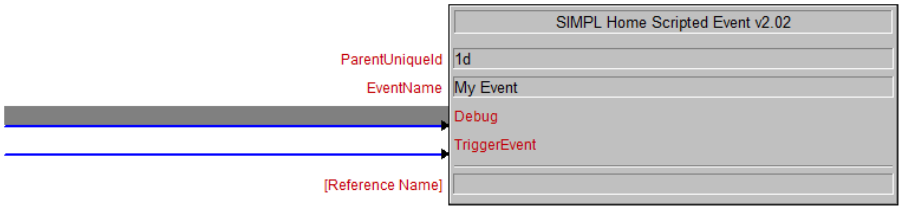
The action pulse will trigger for a single logic wave AFTER the ActionArg (if defined) has been set.

The ActionName parameter defines the action label argument that will appear in Crestron Home.

You may create as many actions as you wish.

SIMPL Home Scripted Event

The Scripted Event provides the reverse – i.e. triggering an event in Crestron Home from SIMPL Home. Crestron Home does not support event arguments.



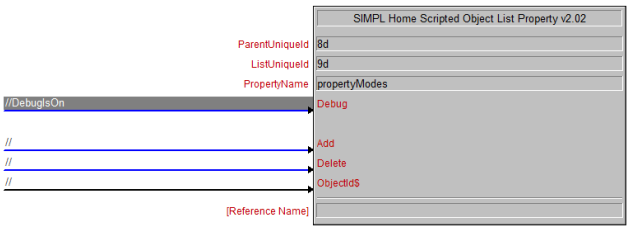
SIMPL Home supports a maximum of 10 Events per Scripted Tile.

SIMPL Home Scripted Object List Property

The Scripted Object List property is by far the most complex object in the suite as it requires:

- A root object, which binds the list to a property used as the source of list items in the Crestron Home UI. List items can be created statically, or dynamically, though this presents some challenges under the SIMPL Windows environment (though not in SIMPLSharpPro).
- A collection of properties that are defined for EACH item in the list.
- Optionally, one or more statically declared items for the list. There are some limitations placed on these statically declared items.

The main root object declares the name of the list for the binding, and also a "List ID" which should also be unique in the collection of object IDs.



COPYRIGHT © 2026 ULTAMATION LIMITED

The example object list shown above is bound to the “propertyModes” property in the sample UI definition.

```
<listbutton id="selectMode" … source="{propertyModes}" … />
```

If dynamic lists are required, this module provides a way to Add and Delete new items.

To achieve this, you must first set the “ObjectId$” value to a unique label which identifies the item in the list. This value is not displayed anywhere, and can simply be a string representation of a number (e.g. “1”, “2”, “3”) or a label (“off”, “heat”, “cool”, “auto”).

Once ObjectId$ is set, trigger Add or Delete to create a new object. When adding an object, you will also then need to set the property values of the new item.

When setting ObjectId$ to an existing item label, the sub-properties (see below) of object list items will display the current property values of the selected list item. You can also SET these values to update the list item’s properties.

Object List Properties

Just list the Scripted Tile properties already described, you will uses these same modules to define “class properties” of the list’s items. Please note that these properties only reflect actual item values when an item has been selected with the ObjectId$ signal.

Note: for object properties where the property is also the display label for the item, failing to set a value will prevent the list item from appearing and Crestron Home may report there at no list items at all, even though they do exist.

Commands, Actions and Events are all handled at the main Scripted object level and do not apply to Object List items.

Statically Declared List Items

Whilst it is possible to do so, SIMPL Windows is not an environment that lends itself well to dynamic lists. While this is an advanced topic and we do not provide support specifically for this, you can use SIMPLSharpPro with SIMPL Home in which case dynamic list management is fairly trivial.

COPYRIGHT © 2026 ULTAMATION LIMITED

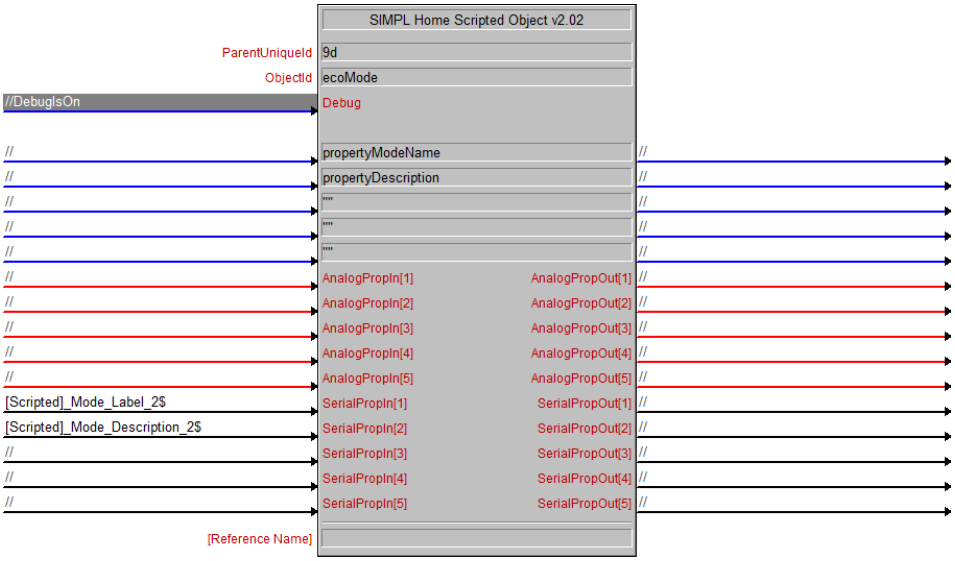
From the provided example program, you will see that we have defined 3 list items statically in the SIMPL Windows program. One for each of the hypothetical modes of our home dashboard.

Each item object declares the parent unique Id (the list object Id) and an object ID for the item – which corresponds to the values passed to the ObjectId$ input on the Object List module.

Next, we defined up to 5 property names which MUST match the properties defined as "class" properties (described above).

As we cannot dynamically cast the values for these properties at runtime, each property is represented by all three signal types – i.e. 5 digitals, 5 analogs and 5 serials, and you will need to connect your signals appropriately.

In this example, we have two properties "propertyModeName" and "propertyDescription" which are both string properties, so we connect signals to SerialPropIn 1 and 2 respectively.



These signals are simply driven by an MSEND and MSP.

List item selection is captured via the Scripted Command ("selectMode") and the value from the selected "propertyModeName" is passed as an argument.

At present only string properties are supported as arguments.

COPYRIGHT © 2026 ULTAMATION LIMITED

Support

If you have any issues with an integration solution please let us know by contacting Ultamation support on support@ultamation.com and please include as much detail about your issue as possible, such as a recent processor error log.

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

Licensing

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

A purchase should not be completed without correct information as refunds cannot be issued for errors or changes made to details following purchase.

This is an electronic product and there is no physical delivery.

The integration solution 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 integration solution's functionality and any bug fixes will be provided free-of-charge. Additional functionality may be released as a variation of this integration solution and this will be a separate, purchasable, product.

COPYRIGHT © 2026 ULTAMATION LIMITED

Cloud Licence

This integration solution contacts Ultamation's licensing server at startup. If the server finds a matching licence for the integration solution and processor then the integration solution will be licensed. Otherwise, the integration solution will check the offline licence key. If you purchase a licence **after** you have loaded the integration solution, please reboot the system to see changes take effect.

If you purchased a licence **before** it was migrated to the cloud service, i.e. you have a licence key already, you must enter this into the **Offline Key** user attribute. If you purchased the integration solution **after** it was migrated, and you don't have a licence key, no further action is required.

If no licence exists for the product/processor the integration solution will enter a short trial period (ONE HOUR) to allow for verification of correct control or evaluation.

To request an OFFLINE key, please contact support@ultamation.com with your order details and a brief explanation why you REQUIRE offline activation. Ultamation reserve the right to refuse offline activation.

NOTE: Once an offline key has been issued no further licence changes will be granted. Moving the integration solution to a new processor will require an additional licence purchase.

COPYRIGHT © 2026 ULTAMATION LIMITED