CRESTRON INTEGRATION MODULE

Revision: 4.01
Date: 21 January 2026

INTRODUCTION

Ultamation's Philips Hue Module allows a Crestron system to control Philips Hue lighting products and smart plugs. The modules provide control of an individual bulb's brightness, hue, saturation, and colour temperature (colour bulbs only) and control of groups of lights simultaneously. It also now supports the use of static and dynamic scenes, and Philips Hue Sensors.

This document is intended for Authorised Crestron Programmers and assumes you are familiar with Crestron control systems, networking, and programming.

To achieve this, you will need the following:

- A Philips Hue Hub/Bridge
- A Philips Hue White Bulb, Philips Hue White Ambiance Bulb, Philips Hue Colour Bulb, LED Strip, smart plug, or sensor

Note: This module is not compatible with 2-Series processors.

Install the Philips Hue Bridge, Lights, and Sensors as instructed by the Philips Hue user manual. Each device and group will require a unique name; these names are used for identification within the Crestron Module. These must match the names as shown in the Philips Hue smartphone application.

Download the module/example program from the Ultamation Shop. For each Philips Hue bridge in your system, add a single Philips Hue Bridge module. Then add a Philips Hue Light, and/or a Philips Hue Group module for each light and/or group you wish to control. Do the same for any sensors. Lights, sensors, and plugs are all represented by the device module. The device module has a device type field with these devices as options.

ESSENTIAL STEPS FOR FIRST USE

Before the control system can be used, it must first authenticate with each Hue Bridge. This is a simple task of associating the control system with the bridge. First, you must put the Hub into "link mode". To do this, press the LINK button on the hub. Within 30 seconds, you must *either* load and run (i.e., upload, or PROGRES) your Crestron program including the Philips Hue Bridge module or trigger the "Refresh" signal on the bridge (see below). This will create an entry in a file on the processor (\user\Ultamation\hue.cache) for future use. If you ever replace the hub, you will need to re-authenticate. To do so, delete this file and repeat the link process.

PHILIPS HUE LIMITATIONS

The Hue Developers site describes the following limitations on commands to a bridge:

"We can't send commands to the lights too fast. If you stick to around 10 commands per second to the /light resource as maximum you should be fine. For /grouped_light commands you should keep to a maximum of 1 per second. The REST API should not be used to send a continuous stream of fast light updates for an extended period of time"

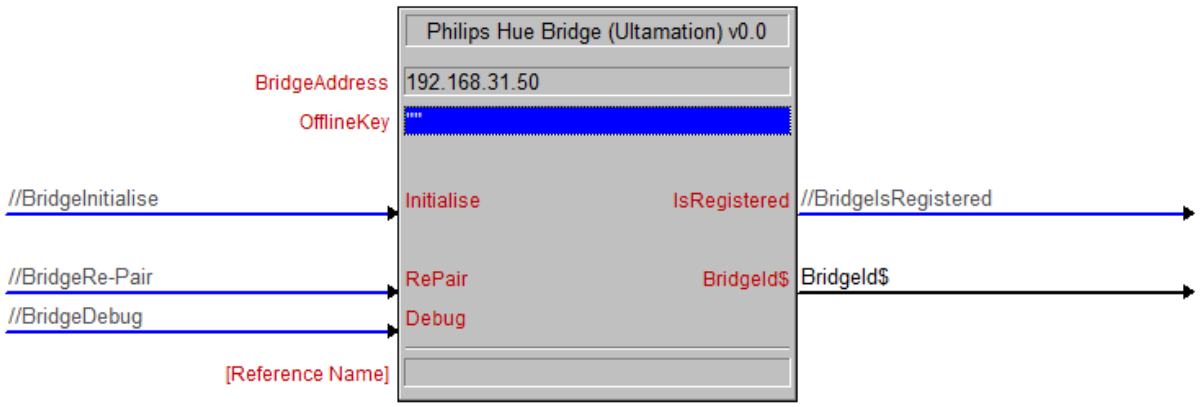
It also provides the following limitations on the number of devices connected to a bridge:

Resource/Device	Limit	Notes
# of ZigBee devices	126	This is the total number of ZigBee lights (max: 63) and sensors (max: 62). No error will be returned from the bridge upon reaching the limit.
# of ZigBee lights	50	Up to 63 is supported, but the system becomes less responsive going higher than 50
# of ZigBee sensors	50	Up to 63 is supported, but the system becomes less responsive going higher than 50. There is an additional limitiation depending on the number of rules created for each sensor.
# of sensors	64	Including the daylight sensor, CLIP sensors, Hue Tap, Hue Dimmer, etc. If the limit is reached a 502 error will be returned.
# of rules	200	A maximum of 400 conditions and 400 actions can be divided over a maximum of 200 rules. An individual rule can contain a maximum of 8 conditions and 8 actions. If the limit is reached a 601 error will be returned.
# of groups	64	If the limit is reached an 301 error will be returned. (Luminaire and LightSource groups aren't taken into account)
# of scenes in lamp	50	
# of scenes in bridge	200	It's a soft limit, because it's depending on the amount of lightstates per scene, see next row. Once the limit is reached, the least recently used scene is deleted.
# of lightstates used for scenes	2048	Once the limit is reached, the least recently used scene is deleted.
# of whitelist entries	100	If the limit is reached the least recently used whitelist entry is deleted.
# of schedules	100	If the limit is reached an 701 error will be returned.
# of resourcelinks	64	

There is a 1 second rate limit on when Philips hue sends updates about a device state, if the same device changes the same property twice within 1 second the value may not be sent correctly until it updates again or in the next second.

MODULE DETAILS

Philips Hue Bridge



Configure the signals/parameters of the bridge as follows:

BridgeAddress – Enter the IP address or hostname of the Phillips Hue Bridge as configured using the Philips Hue application. We recommend a static address is used for each Bridge.

OfflineKey – The bridge module is licenced 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 bridge modules 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 logs will display the status for the licence.

Initialise – This is to be used once during startup to allow staggered initialising of the bridges. If you only have one bridge you can set this to 1 and it will activate on startup.

RePair – This is used during initial commissioning. It creates the unique application key with a bridge. And reinitialises the internal model with devices, groups, and scenes. This is used at the beginning - after you have pressed the LINK button on the Philips Hue Bridge (and within 30 seconds).

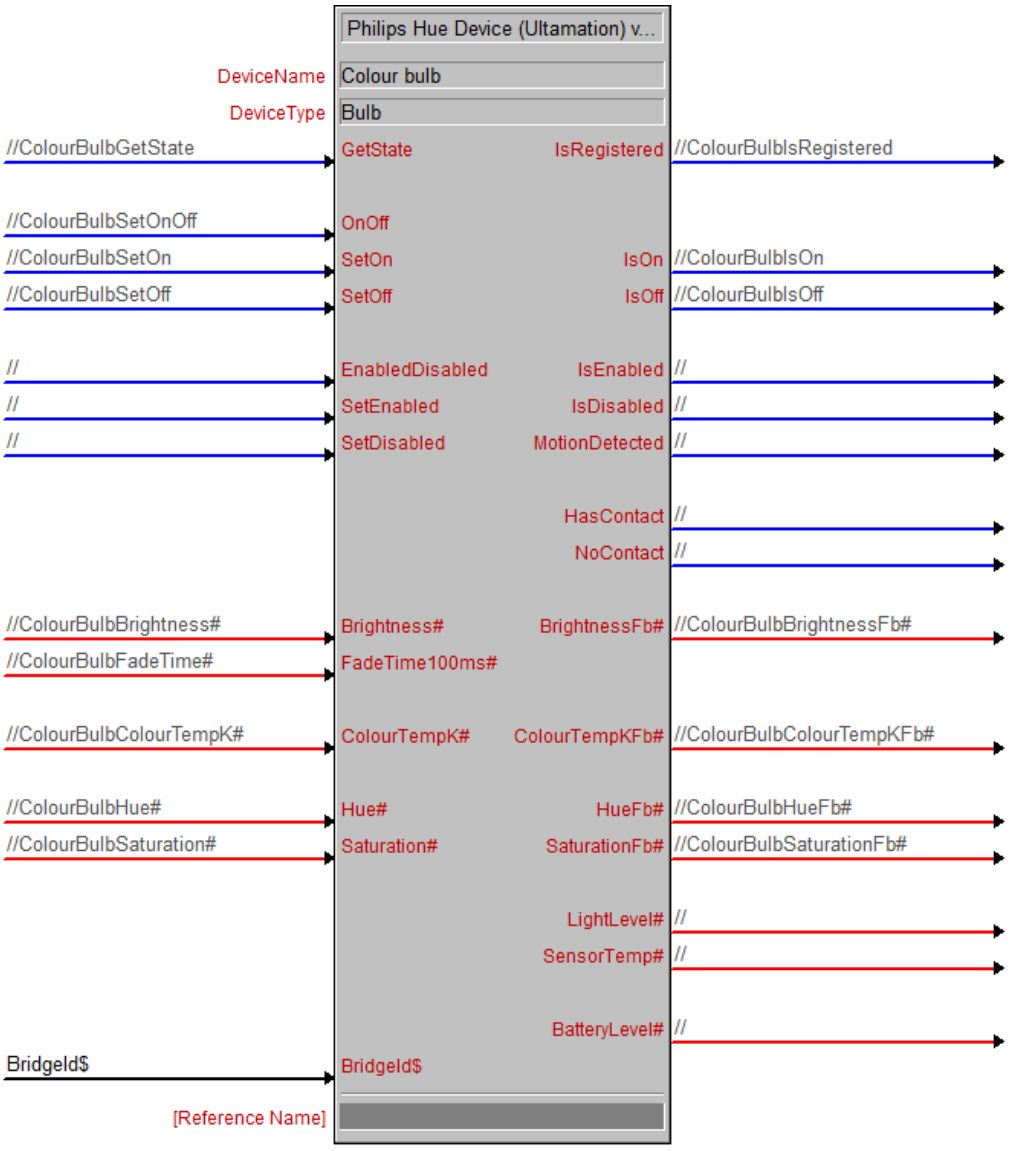
Note – If you change any devices, groups, and scenes after initialising, this will need to be triggered again to update the internal model registered on the bridge.

IsRegistered – This will go high when the bridge's IP address is registered with the program. This is for diagnostic purposes only and does not necessarily mean the IP address is valid.

BridgeId$ – This signal MUST be connected to the device and group modules that control devices and groups connected to this particular bridge. When the bridge initialises, the BridgeId$ signal is sent to the other modules to trigger initialisation of these elements.

Debug –This signal is for debugging purposes. When set high it will send extra details to the console and error log. If you are having issues with the module, you can send this information to support@ultamation.com so we can help solve the problem.

Philips Hue Device



Parameters

DeviceName – This must match the name (not case sensitive) used when setting up the Philips Hue Bulb within the Hue app. For devices with multiple lights the name should be in the format "Name SubName".

DeviceType – Choose either Light, Plug, or Sensor.

Inputs

GetState – Requests the current state of the device. The returned values are outputted on the relevant output signals.

OnOff – Lights and plugs only. This allows a device's on/off state to be controlled from a single signal. It is NOT a toggle. but reflects the state of the signal. i.e., HIGH = On, LOW = Off.

SetOn – Lights and plugs only. Discrete (rising edge triggered) to turn on the device.

SetOff – Lights and plugs only. Discrete (rising edge triggered) to turn off the device.

EnabledDisabled – Sensors only. This allows a sensor's enabled/disabled state to be controlled from a single signal. It is NOT a toggle. but reflects the state of the signal. i.e., HIGH = On, LOW = Off.

SetEnabled – Sensors only. Discrete (rising edge triggered) to enable the sensor.

SetDisabled - Sensors only. Discrete (rising edge triggered) to disable the sensor.

Brightness# – Lights only, Controls the intensity of light. Full scale analogue 0 – 65535 (0-100%)

FadeTime100ms# – Lights only. Defines the transition time between current value and the requested value, given in multiples of 100ms. i.e., a value of 20 is a 2 second fade.

ColourTempK# – Colour and ambiance lights only. Sets the light to the white colour temperature expressed in degrees Kelvin. Valid range is 2000 to 6500. Values outside this range will be clipped. Lower values represent "warm" light and higher values represent "cold" light.

Hue#– Colour lights only. Sets the colour hue of the light. Full scale analogue 0-65535 (0-100%). On the standard hue range, 0 is Red, 33% is Green, 66% is blue and 100% is Red again.

Saturation# – Colour lights only. Sets the colour saturation of the light. Full scale analogue 0-65535 (0-100%). Where 0% is no colour (i.e., white) and 100% is full colour.

BridgeId$ – Each device module MUST be driven by a signal from the appropriate bridge module.

Outputs

IsRegistered – Will return high when the device is "defined" on the Hue bridge. If this signal is low, there is likely a configuration issue in the device name, or BridgeId$.

IsOn – Lights and plugs only. High when device is on. Low when device is off.

IsOff – Lights and plugs only. Low when device is on. High when device is off.

IsEnabled – Sensors only. High when sensor is enabled. Low when sensor is disabled.

IsDisabled – Sensors only. Low when sensor is enabled. High when sensor is disabled.

HasContact – Contact Sensors only. High when sensor detects contact. Low otherwise.

NoContact – Contact Sensors only. High when the sensor detects no contact. Low when contact is detected.

MotionDetected – Sensors only. Pulses when motion is detected.

BrightnessFb# – Lights only, provides the current intensity of light. Full scale analogue 0 – 65535 (0-100%)

ColourTempKFb# – Colour and ambiance lights only. Provides the current white colour temperature, expressed in degrees Kelvin.

HueFb#– Colour lights only. Provides the current colour hue of the light. Full scale analogue 0-65535 (0-100%). On the standard hue range, 0 is Red, 33% is Green, 66% is blue and 100% is Red again.

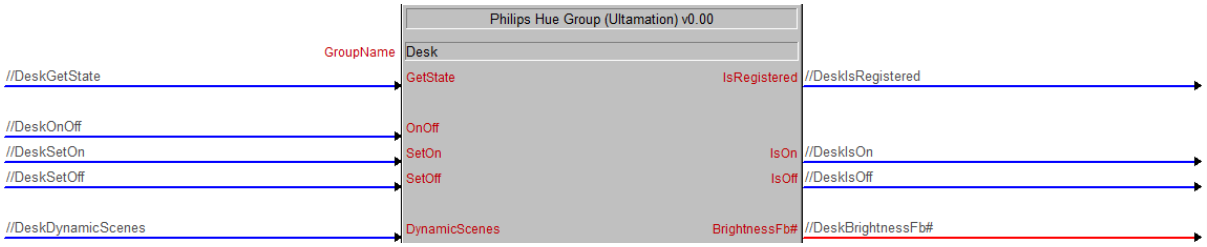
SaturationFb# – Colour lights only. Provides the current colour saturation of the light. Full scale analogue 0-65535 (0-100%). Where 0% is no colour (i.e., white) and 100% is full colour.

LightLevel# - Sensors only. Sensors can output the current light level when it passes a threshold defined in the Hue app. The value is given as 10000*log10(lux) +1. The Philip Hue API reference gives the following reasoning for this: "Logarithmic scale is used because the human eye adjusts to light levels and small changes at low lux levels are more noticeable than at high lux levels. This allows use of linear scale configuration sliders."

SensorTemp# - Sensors only. Sensors can output the current temperature readings at regular intervals. Given in degrees Celsius.

BatteryLevel# - Battery powered devices only. Such a device may output its current battery power level when it becomes low.

Philips Hue Group



The signals for the group module are like those of the device module. Our group module can be used for both room and zone type groups. Note that feedback for hue, saturation, and colour temperature will be output from the individual light modules.

GroupName – This must match the name (not case sensitive) used when setting up the group within the Hue app.

GetState – Requests the current state of the group (on/off state and brightness). The returned values are outputted on the relevant output signals.

IsRegistered – Will return high when the group is "defined" on the Hue bridge. If this signal is low, there is likely a configuration issue in the group name, or BridgeId$.

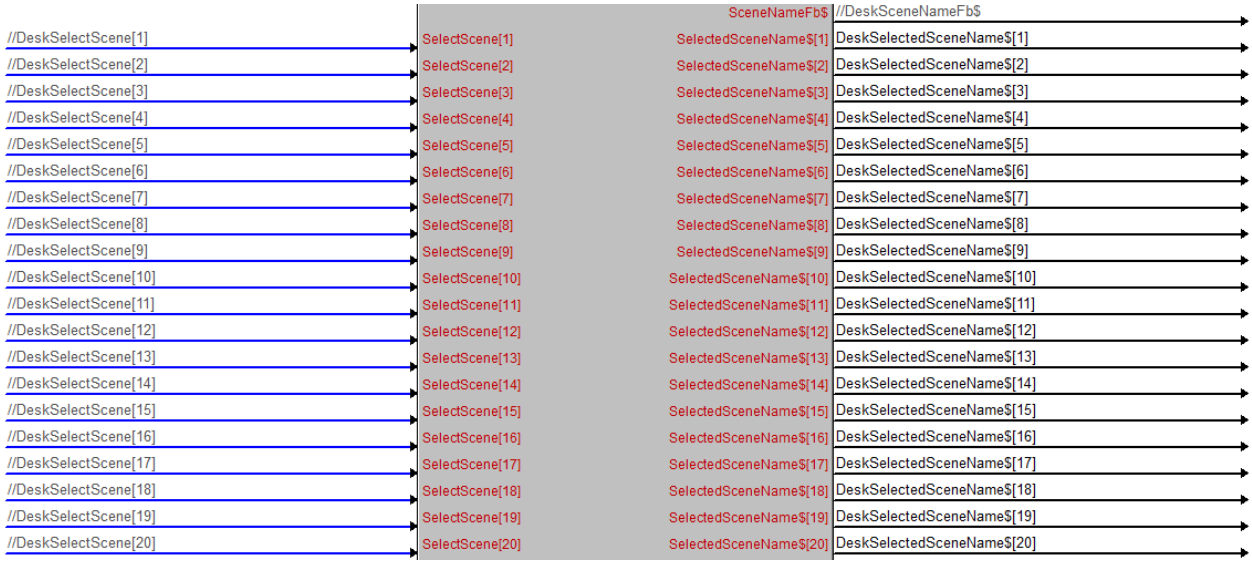
OnOff –This allows a group's on/off state to be controlled from a single signal. It is NOT a toggle but reflects the state of the signal. i.e., HIGH = On, LOW = Off.

SetOn –Discrete (rising edge triggered) to turn on the group.

SetOff –Discrete (rising edge triggered) to turn off the group.

DynamicScenes – Controls the static/dynamic state of scenes. HIGH – Dynamic, LOW – Static.

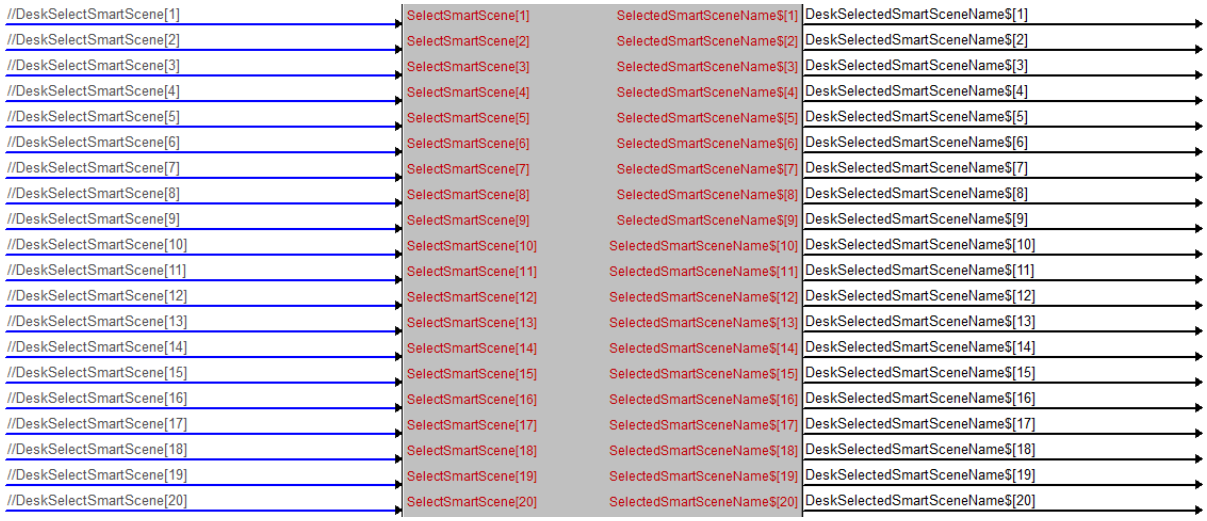
BrightnessFb# –Provides the current intensity of the group. Full scale analogue 0 – 65535 (0-100%).



SceneNameFb$ - Outputs the name of the group's current scene whenever it changes.

SelectScene[] – Will cause the chosen scene to be set on the current group. Scenes are ordered alphabetically.

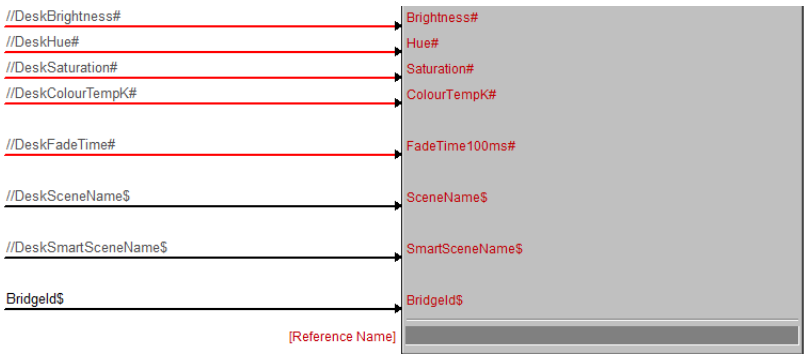
SelectedSceneName$[] - the name of the scene at the specified index of the scene list. Up to 20 scenes can be stored.



SelectSmartScene[] – Will cause the chosen smart scene to be set on the current group. Smart scenes are ordered alphabetically.

SelectedSmartSceneName$[] - the name of the smart scene at the specified index of the smart scene list. Up to 20 smart scenes can be stored.

There is **no name feedback for the selected smart scene**. This is because a smart scene works by transitioning through a collection of normal scenes over the course of the day. When it transitions to another scene, the name of the scene will be given via SceneNameFb$.



Brightness# – Controls the intensity of light. Full scale analogue 0 – 65535 (0-100%)

Hue #– Controls the colour of the light. Full scale analogue 0-65535 (0-100%). On the standard hue range, 0 is Red, 33% is Green, 66% is blue and 100% is Red again.

Saturation# – Controls the colour saturation. Full scale analogue 0-65535 (0-100%). Where 0% is no colour (i.e., white) and 100% is full colour.

ColourTempK# – Sets the light to the white colour temp expressed in degrees Kelvin. Valid range is 2000 to 6500. Values outside this range will be clipped. Lower values represent "warm" light and higher values represent "cold" light.

FadeTime100ms# – Defines the transition time between current value and the requested value, given in multiples of 100ms. i.e., a value of 20 is a 2 second fade.

SceneName$ - Sets the scene of the group. Value must match the name (not case sensitive) of a scene associated with the group within the Hue app.

SmartSceneName$ - Sets the smart scene of the group. Value must match the name (not case sensitive) of a smart scene associated with the group within the Hue app.

BridgeId$ – Each group module MUST be driven by a signal from the appropriate bridge module.

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 a recent processor error log.

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

LICENCING

This integration solution (including software, images and all other associated assets distributed as part of the purchased download package) is licenced 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.

CLOUD LICENCE

This integration solution contacts Ultamation's licencing server at startup. If the server finds a matching licence for the integration solution and processor then the integration solution will be licenced. 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** parameter. 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.