

## Lutron LEAP for RA3 and QSX

**This Auto-Config driver is designed for Integration Designer APEX only.**

Version 10.10.1 or higher is required

This driver is designed to use Lutron's LEAP protocol and tested with a Lutron HQP7-2 processor and a Radio RA3 processor. This driver is only intended for use with QSX and RA3, not Caseta.

### Revision History

- 1.0** Initial Release of the Driver
- 1.1** Updated runtime environment to use HTTPS, with certificates instead of login information.
- 1.11** Fixed an issue with new Erase Pairing Information function, and added additional name checking on sources to allow for special characters(< > , ' ").

### **Adding the Driver**

The driver is added via the Drivers tab in the Add Workspace Item dialog. The driver is named 'Lutron LEAP' and initially offers only the processor as a source to add to the system. It does not matter what room the processor is added to.

Once the processor is added you must add the connection information: IP Address, Port, and Login information, in the configurations. This is done by opening the Drivers section in the Global room and selecting the Lutron Leap driver, then selecting Driver Properties in the Driver Configuration section.

### **Adding the remaining sources**

While the Lutron Leap driver is selected, two additional buttons will be in the Driver Configuration. One will be labelled 'Connect to Lutron Processor', the other is labelled 'Driver Sync'. Once the connection information has been entered, the 'Connect to Lutron Processor' button must be used to connect the driver to the processor and collect the current system stored there. Once that has been done ID will know the configuration of the Lutron processor and will make those devices available to the driver. Any time that the Lutron project is changed via Homeworks, this button should be used to update the copy of the project that ID is using. The Driver Sync button runs the routines that are called as new sources are added to the project. You should not need to use this button but occasionally ID may not refresh the listed commands or variables. In this case pressing this button causes those routines to be run again. You may have to switch to viewing an actual source for the 'Macro Stepsd' to all be visible.

Returning to the Drivers tab of the Add Workspace Item, the driver should now be listed under the 'Add To Existing Driver', and should have up to 3 subsections: Lights/Zones, Shades, and Controls. Expanding these sections will show the devices that are currently programmed into the Lutron System. These devices will be removed from this list as they are added to rooms.

### **Adding Sources**

Initially, before devices are added to rooms in the workspace, there will be no commands listed under the Macro Steps pane on the right of the APEX window, and no commands to add in the Button Properties pane at the bottom. Commands, variables, and events will only be added for devices when they are placed into room using the Drivers tab of the Add To Workspace dialog.

*(APEX may not update the listing of macros in the Macro Steps pane until your focus in the Workspace has switched to a different source)*

### **\*\*NEW\*\* Connecting the Lutron Processor to your RTI Processor**

After pushing your APEX file to the RTI processor, it will establish an initial connection to the Lutron system, after this connection is established a System Variable for "Pairing Required" will flag true as well as a statement will print to the system log to push the button on the Lutron bridge, this will give RTI access to negotiate authorization with the bridge. This will only need to be done the first time you connect your RTI processor to the Lutron Bridge, subsequent connections should work from memory, if they do not there is a function "Erase Pairing Info" to call from a RTI panel which will reset the system to no longer hold onto the needed authorization information. After calling this function you will need to reset the RTI processor to reestablish the connection to Lutron and start over in the pairing process.

## Commands

Integration Designer will build the commands for devices as they are added to the system:

### Lights:

Dimming Loads: Lighting Loads have commands to set their level from 1-100, as well as turn them off, on, and toggle them. They do not remember their former level and will toggle between off and full brightness.

Switched Loads: Switched Loads have the same commands, without the Raise, Lower, and Stop. Sending a command to set the level will turn the light on if the level is greater than 0, and off if the level IS 0.

SpectrumTune (Ketra): These have the same commands again, but add commands to set the Hue, Saturation, Vibrancy, and White Tuning. These commands are only available for lights that support them.

WARNING: There is a bug in the Lutron system. If you set the saturation to 0, the hue may change. The sliders are set in the driver to only allow and adjustment from 1-100.

### Shade Groups:

Shades can be sent to a specific level (1-100). They also have a raise and lower command that will continue until the stop command is sent, or the shade reaches the top or bottom of its travel. If a shade group includes the ability to tilt, the command to set the tilt will be included.

### Control Stations:

The Control Stations will have commands to emulate the pressing of their buttons. The commands will appear in ID with both their button number and the engraving text that was entered in Homeworks to help identify them. The commands will default to Press and Release, but Press and Hold, Release, and Multitap are also available. These must be programmed correctly in Homeworks to work correctly

Lutron button numbers vary greatly from device to device and occasionally have two different numbers. The buttons appear in ID with a name

## Variables

Integration Designer will also build the variables for the devices:

*Note: When using Raise and Lower commands on Lights and Shades, the level will not be updated until the stop command is received. Even if the device reaches a top of bottom limit.*

### Lights:

Dimming Loads: Most loads will have a Name, a Level, and a State. The state will be true, or on, for any value greater than 0.

Switched Loads: Have the same variables and will show a level of 100 whenever they are turned on.

SpectrumTune (Ketra): These have level variables for Hue, Saturation, Vibrancy, and White Tuning. The Hue and Saturation levels will adjust for any change in White Tuning, but the Tuning Level does not adjust for changes in Hue or Saturation.

### Shade Groups:

Shades have a Name variable and can show their current level. They have a variable for when they are closed and two 'Open' variables: One any time they are not at bottom (0), and another for when they are at the top (100). These two variables also have different tags, as shown below. Tiltable shades will also include a Tilt variable.

### Control Stations:

The Control Stations themselves only have a Name variable. Each button on the station will have a

name of its own, which is automatically filled from the Engraving Text entered in to Homeworks. It will also have an LED state variable if the button has an LED associated with it.

#### **Others:**

Overall variables also exist for the connection state of the driver

## **Events**

Integration Designer will also build the variables for the devices:

#### **Lights:**

All lights have events when they turn on and off

#### **Shade Groups:**

Shades have a slightly more complicated set of events to help you know roughly where they are in their travel. They have events for when they hit the top and bottom of their travel, and events for when they leave those positions. As an example, if a shade is at the bottom and is sent to level 50, it will trigger the 'Not fully closed' event, but NOT the Open event. If it subsequently does move to the top, the event Open will be triggered. When it leaves the fully open position, the matching event will be triggered, with the Closed Event being triggered only when the shade is completely closed.

It is probable that the Closed, and Not Fully Closed events will be the most used. One thing to be careful of is that the Lutron system sends differing status messages to the driver, depending on how the shades are operated. If a Control Station button is programmed with a Raise command, the system will immediately send the driver a status message that says that the shade is fully open. Only when you stop the travel will the number be updated again. The same thing happens with a Lower command sending a fully closed status. If you use the Raise and Lower commands in the driver however, the Lutron system never sends any updates, and no events will be triggered, until an actual Stop command is received.

#### **Control Stations:**

The Control Stations will trigger events when any button is pressed, or when the status of any LED changes.

## **Tags:**

#### **Lights**

Instance ID: The name of the light

Light Level: Sliders and direct set commands

Light: Toggles light and shows on/off state (for button control)

Light On: Discrete on command and light state

Light Off: Discrete off command

Light Up: Fades the light up

Light Down: Fades the light down

Light Stop: Stops the fading (In this driver, the level and state variables are not updated while the light is fading, only when the fading is stopped)

Light Hue Level: Sets the Hue of the Ketra lights, and displays the Hue Level

Light Saturation Level: Set and displays Saturation

Light Temperature: Sets the Color Temperature of the lights (Changing Color Temp will update Hue and Saturation, but changing Hue or Saturation DOES NOT update Color Temp)

#### **Fans:**

Instance ID: The name of the fan

Fan Level: Slider control and status

Fan: shows on/off state

Light Off: Discrete Fan Speed command and boolean speed indicator

Light Low: Discrete Fan Speed command and boolean speed indicator

Light Medium: Discrete Fan Speed command and boolean speed indicator  
Light Medium High: Discrete Fan Speed command and boolean speed indicator  
Light High: Discrete Fan Speed command and boolean speed indicator

#### Shades:

Instance ID: The name of the shade  
Shade Level: Sliders and Direct set  
Shade Up: Starts the shade up (Does not update variable)  
Shade Down: Starts the shade down (Does not update the variable)  
Shade Stop: Stops the shade

Shade Tilt: Sets (and displays) Tilt level for tiltable shades

#### Control Stations:

Instance ID: The name of the Control Station  
Button Name x: Name of Button 1 (Will show the engraved text as programmed in Homeworks)  
Button x: Press and Release Button x, Shows the LED state of Button x

#### Events

##### Lights

On  
Off

##### Fans:

On  
Off  
Low  
Medium  
Medium High  
High

##### Shades

Open: Triggers when shade hits 100%  
Not Open: Triggers when Shade is no longer at 100%  
Closed: Triggers when shade hits 0 (Completely closed)  
Not Closed: Triggers when shade is no longer at 0

#### Control Stations:

*The event you receive is dependent on the programming of the button in Homeworks*

Button Press  
Button Release  
Button Multitap  
Button LongHold  
LED On  
LED Off