Nice Developer Partners	**Integration Note**
Manufacturer:	**Converging Systems, Inc.**
Model Number(s):	**ILC-x00 family of LED lighting controllers** **IMC-xx0 family of Motor controllers**
NICE Core Module Version:	Core Module Version: **8.6** (Schema 95 LUA driver) *
Models Tested	Front-end Comm Devices: e-Node, e-Node/dmx, IBT-100 Downstream Controllers: ILC-100sa, ILC-300, ILC-400, ILC-450 IMC-100, IMC-170, IMC-300, CVM
Driver Information	Converging Systems Inc. (licensed NICE LUA Developer Partner) Driver Version 1.0.397 for LED, Driver Version 1.0.391 for Motors
Document Revision Date:	03/17/2021

***Note**: This Schema 95 driver is only compatible with Core releases 8.6. and later. If using a Core version between 8.0.279 through 8.5 Build 510, please use the Converging Systems' Schema 60 driver 1.0.38 instead.

For compatibility with Core Module 7.2 versions, please refer to the *CSI Integration Note* for Core Module Version g!7.2 (Schema 3 LUA Driver) and separate driver (V1.019) drivers.

IMPORTANT NOTE-NEW LIGHTING FEATURES (click on links below for more info)

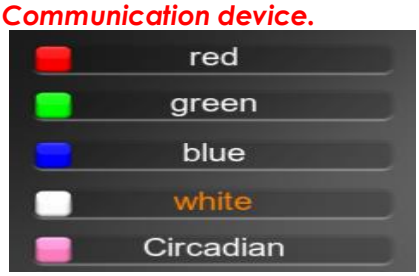
With the launch of NICE's new 8.6 core software, exciting changes to lighting control have occurred. Converging Systems is pleased to provide its new driver at the launch of 8.6!

Changes:

- **MULTI Device**. For Full color (RGBW/tunable White) applications, use the new **MULTI** device type (rather than the **MCH_RGB** device) which when a new "**Light Multi**" Control is populated within a Custom Page, an auto-expanding/autogenerated **Popup** automatically appears without programming each control within the UI. Four separate controls—**Brightness, Color Temp, Hue, & Saturation *all appear on one popup* (in that order)** rather than having to create four separate sliders.

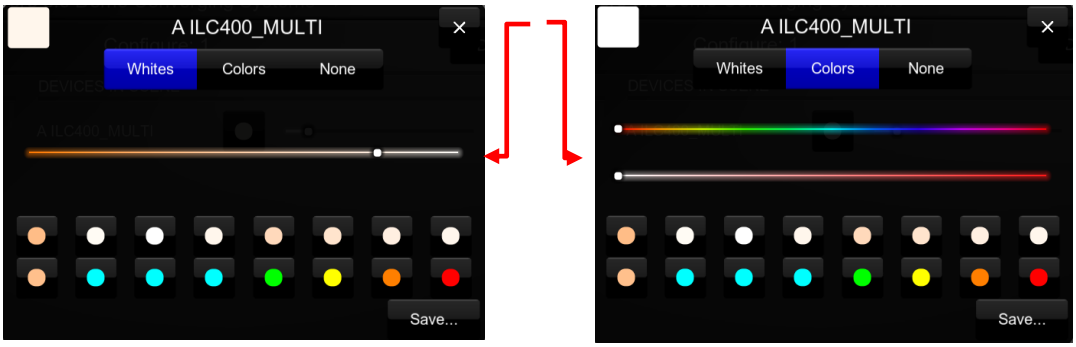
Converging Systems Inc.• 32420 Nautilus Drive• Rancho Palos Verdes, CA 90275 USA
Tech support 310.544.2628 x2 • Sales 310.544.2628 x 1 • email:info@convergingsystems.com

- **Additional adjustable controls**[1] for other color parameters (Red, Green, Blue, White, and Circadian) can still be exposed by adding **SLIDERS** (not Dimmers) to your Custom Page and activating them by using the **Properties/"Connect To"** pulldown to link to automatically generated controls (i.e., Red, Green, Sun (for Circadian) for the Zone/Group/Node (**ZGN**) address of your individual controller). ***All supported control parameters automatically populate with the "Connect To" scroll box without any dealer programming so long as our auto-generated MULTI device appears under the connected Communication device.***



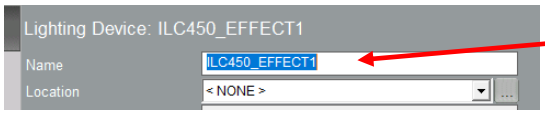
[1]Note: NICE engineers recommend that no two identical sliders be programmed on the same page to control the same function. If you have a Color slider for HUE on the Popup page, don't create another HUE slider elsewhere controlling the same ZGN address controller.

- **Customizable Scene** buttons have been dramatically turbocharged! Link a new Customizable Scene button to the newly populated **MULTI** device and have fun exploring. You will never guess when you hit the dot here what pops up!



- **Project Update.** We have got you covered, simply (i) download the latest CSI driver and place in the correct directory, (ii) select "Update" for your current device , and (iii) select "Discovery Device" and for applicable controllers we will automatically update your old MCH_RGB device type to the **Multi**-type without any intervention on your side. Just add a new GUI object called **Light Multi-Control**, link and you are done!

- **More neat lighting features** have been added including a new **Area/Location** selector within each CSI communication controller in the parent Lighting control properties box. Refer to new NICE documentation for more information.



Integration Note Table of Contents

Please the following table of contents to help you navigate through this Integration Note.

OVERVIEW AND SUPPORTED FEATURES

The Converging Systems ILC-x00/IMC-xxx family of LED lighting controllers (and motor controllers) are networkable devices which can provide support for

- Converging Systems' Flexible Linear Lighting Arrays (FLEX) RGB, RGBW, and monochrome LED devices powered by ILC-100/ILC-300/ILC-400 devices (using the front-end e-Node or the IBT-100)
- Specified certified third-party lighting fixtures powered by the ILC-450 controller (using the e-Node or IBT-100).
- Any third-party DMX device (using the e-Node/dmx).
- Specified third-party motor devices powered by the IMC-xx0 controllers (using the e-Node or IBT-100 or built-in IP functionality).

The NICE system is capable of receiving bi-directional communication data (color status in RGB, RGBW, or HSB color space) and updating NICE sliders (faders), toggle buttons, and other status indicators to indicate real time feedback of color state changes. The NICE system is also capable of receiving similar bi-directional information from Converging Systems' motor controllers.

Note: If IP connectivity is possible within your installation, this is the preferred communication choice given the new Auto-Discovery feature available within NICE/Converging Systems' software. This feature is supported only with the e-Node and dramatically reduces the programming time required for initial NICE programming (for more information click here). Without Auto-Discovery, individual entries for all specific sliders and controls (red/green/blue or hue/saturation/brightness as well as individual scenes and effects must be manually added). With Auto-Discovery (only available with the e-Node), nearly all of these manual processes are eliminated.

Theory of Operation-note on IBT-100 use. The NICE/Converging System's driver queries an XML database present within the e-Node to make intelligent decisions as to the type and quantity of Devices auto-populated. Therefore, if you wish to use the IBT-100 you must resort to manual Device entry which is quite acceptable for small installation. See Appendix 4 for step-by-step directions.

Regardless of the type of Converging Systems' lighting or motor controllers desired to be supported (i.e., e-Node with one or more ILC-xx0 controllers, e-Node/dmx with third-party DMX fixtures, or e-Node with any IMC-100 controller), a **single NICE driver is utilized for each** separate e-Node system or IBT-100 front-ended system.

THE FOLLOWING OPTIONS ARE SUPPORTED BY THE CONVERGING SYSTEMS CS-BUS (LIGHTING) DRIVER:

- Support of new NICE multi-control feature with the new Light Multi Channel Control UI for supported devices which offers RGBW/Kelvin functionality (***used with auto-populated sliders and popups)***

  (Schema 95 and later) 

- Support of new NICE Customizable Button feature pop-ups for Full Color control for supported devices)

  (***used with auto-populated NICE sliders and popups)*** (Schema 95 and later) 

- Support of new NICE Area feature that enables lighting controllers to be dealer assigned to particular Areas (Schema 95 and later) 

- Automatic upgrade support for existing NICE project from non-MULTI functionality to MULTI functionality by simply loading new CSI driver and re- "Discovering Device." 

- Support of NICE Dimmer (RGB-Multi-Ch) feature with the new Light Multi Channel Control UI for supported devices which offer RGB but no Kelvin functionality (***used with auto-populated Multi Channel Control as well as individual Sliders not Light Dimmer objects***) (Schema 95 and later)

- Support of NICE Light RGB Control GUI object for RGB devices (Schema 20 and later)
- Auto-discovery of ILC-100m, ILC-100c, ILC-300, ILC-400, and ILC-450 controllers previously identified and addressed through the e-Node Web Server (or Pilot application) -- This is a huge labor-saving feature.
- Auto-discovery of 1-channel, 2-channel, 3-channel and 4-channel DMX fixtures (using the e-Node/dmx) of devices previously identified and addressed through the e-Node Pilot application or the Web Server commissioning process -- This is a huge labor-saving feature.
- Discrete control of LED states (ON/OFF)
- Two-way control of Correlated Color Temperature (CCT) (or sometimes referred to as "Dynamic White") settings with RGBW devices using Converging Systems FLLA LED elements or certified third-party fixtures. Specific CCT settings can be selected as well as CCT UP/DOWN controls for CCT adjustments
- Two-way control of Circadian Rhythm (Sunrise to midday sun to Sunset dynamic settings) using Converging Systems RGBW FLLA devices and certified third-party fixtures.
- Support of communication utilizing Telnet with or without authentication (Port 23)
- Two-way control of color settings in the RGB, RGBW, or HSB color space.
- Ability to store and recall specific colors set by a user (using Customizable Scenes) stored within NICE controllers.
- Ability to store and recall specific colors set by a user within ILC-x00 controllers. (Schema 11 and later)
- Ability to recall specific Effects stored within specific ILC-x00 controllers. (Schema 11 and later)
- Ability to change Dissolve Rates (time it takes to transitions from one state to another) (i) for On and Off states, (ii) for Presets to other Presets (color) settings, and (ii) for state to state transitions within Effects. (Schema 11 and later)
- Ability to change Sequence Rates (time after any dissolve that a Preset color is maintained before transitioning to the next color in sequence) in Effects 1 and 4. (Schema 11 and later)
- Ability to store a Color Temperate or a Circadian Sun level setting within a Customizable Scene
- Control via all thin client interfaces (PC, NICE Touchscreen, Android, iOS,TS2, and HR2)

THE FOLLOWING OPTIONS are not supported by CS-Bus (lighting) driver**:**

- Auto-Discovery using the IBT-100 serial interface controller (manual discovery as described in Appendix 4 is still possible)
- Reliable feedback on Viewer Page for "All Devices OFF or ON" which is being deprecated by NICE

THE FOLLOWING OPTIONS ARE SUPPORTED BY THE CONVERGING SYSTEMS CS-BUS (MOTOR) DRIVER:

- Motor UP/Down/Stop
- Motor Position Feedback (for CS-BUS motor controllers that provide this level of functionality).
- Store and Recall of presets (for CS-BUS motor controllers that provide this level of functionality)
- Support of communication utilizing Telnet with or without authentication (Port 23)

THE FOLLOWING OPTIONS are not supported by CS-Bus (motor) driver:

- Auto-Discovery using the e-Node or the IBT-100 serial interface controller (manual discovery as described in Appendix 4 is still possible)
- Reliable feedback on Viewer Page for "All Devices OFF or ON" which is being deprecated by NICE

Tabular Summary of Supported Features

The following commands are supported by the current driver for the various lighting and motor control devices (except those that are grayed out).

LED Lighting Commands

Table 1

General CS-Bus Commands	NICE Naming Convention[1]	ILC-100 m	ILC-100c (sa) ILC-300	ILC-400 ILC-450 (RGBW mode	ILC-400 (4 ch Mono)	e-Node DMX
General LED Control Commands						
ON	eNode_On	✓	✓	✓	✓	✓
OFF	e-Node_Off	✓	✓	✓	✓	✓
EFFECT,n	Execute_Effect	✓	✓	✓	✓	✓1
STORE,#	Store Preset	✓	✓	✓	✓	✓
RECALL,#	Recall Preset	✓	✓	✓	✓	✓
DISSOLVE.1=XX	Set_Dissolve_Rate	**	**	**	**	**
DISSOLVE.2=XX	Set_Dissolve_Rate	**	**	**	**	**
DISSOLVE.3=XX	Set_Dissolve_Rate	**	**	**	**	**
DISSOLVE.5=XX	Set_Dissolve_Rate	**	**	**	**	**
SEQRATE=XX	Set_Sequence_Rate	2	2	2	2	2
SUN_UP	Sun_Up			✓	✓	
SUN_DOWN	Sun_Down			✓	✓	
SUN.S	Set_Circadian_Value			✓	✓	
NICE's Customizable Scene	Can program any CS-Bus command to operate with memory retained in NICE processor	NICE	NICE	NICE	NICE	NICE
HSB (HSL) Color Space Commands						
FADE_UP	Fade_Up	✓	✓	✓	✓	✓
FADE_DOWN	Fade_Down	✓	✓	✓	✓	✓
SET,L	Set_Brightness	✓	✓	✓	✓	✓

HUE_UP	Hue_Up		✓	✓		✓
HUE_DOWN	Hue_Down		✓	✓		✓
HUE,H	Set_Hue_Value		✓	✓		✓
SAT_UP	Sat_Up		✓	✓		✓
SAT_DOWN	Sat_Down		✓	✓		✓
SAT_S	Set_Saturation_Value		✓	✓		✓
STOP	STOP	✓	✓	✓	✓	✓
~~COLOR=H.S.L~~	~~Set_Preset_HLS Colorspace~~	~~✓~~	~~✓~~	~~✓~~		~~N/A~~
PRESETH.X=XXX.XXX.XXX	Set LED Presets/HLS Color spacer for preset x	✓	✓	✓	✓	✓
COLOR,H.S.L	Set_Preset_HLS Color space	✓	✓	✓		N/A
RGB(W) Color Space Commands						
RED,R	Set_RED_Value		✓	✓		✓
GREEN,G	Set_GREEN_Value		✓	✓		✓
BLUE,B	Set_BLUE_Value		✓	✓		✓
VALUE=R.G.B	???					
WHITE,W	Set_BLUE_Value	✓	✓	✓	✓	✓
RGB,R.G.B	Set RGB Value		✓	✓		✓
RGBW,R.G.B	Set RGBW Value			✓		✓
PRESET.X=XXX.XXX.XXX (3-color)	Set LED Presets/RGB Color spacer for preset x					
PRESET.X=XXX.XXX.XXX (4-color)						
STOP	Stop adjustment	✓	✓	✓	✓	✓
Correlated Color Temperature (CCT) Commands						
CCT,XXXX	SET_Correlated_Color_Temp			✓		✓
CCT_UP	Color_Temp_Up			✓		✓
CCT_DOWN	Color_Temp_Down			✓		✓
Bi-Directional Commands						
COLOR=?	Automatic polling within Driver. **Note**: Driver achieves same function with Notify ON	✓	✓	✓	✓	✓
VALUE=?	Automatic polling within Driver **Note**: Driver achieves same function with Notify ON	✓	✓	✓	✓	✓
PRESETH.X=?		*		*		*
PRESET.X=?		*		*		*

Accessory 3-Node Command/Setup Parameters						
Verbose Mode						
UDP Port 4000/5000						
Telnet Login with Authentication (with e-Node		✓	✓	✓	✓	✓
Telnet Login without Authentication		✓	✓	✓	✓	✓

Notes:

- With current LUA release, these can only be set within e-Node Pilot**.** Check back to see if any updates to the LUA driver have become available allowing these to be set directly.

** Integrated feature within LUA Dimmer Devices, LUA Scene Devices

1 Effect (1) only supported

2 Easiest to set within the ILC-xxx device using e-Node Pilot, or alternately you can use special driver field for adding two dissolve/seq rates into one device. See link for details.

NICE Feature is implemented through internal function within NICE programming rather than supporting this command.

Motor Commands

Table 2

General Commands	NICE Naming Convention	IMC-100	BRIC ("Bric Mode ")	IMC-300 (MKII) IMC-170
General Motor Control Commands				
UP		✓	✓	✓
DOWN		✓	✓	✓
STOP		✓	✓	✓
RETRACT		✓	✓	✓
STORE,#		✓	✓	✓
RECALL,#		✓	✓	✓
PRESET.X=XX.XX				
Bi-Directional Commands				
STATUS=?				
POSITION=?				✓
Accessory e-Node Command/Setup Parameters				
Verbose Mode		✓	x	✓
UDP Port 4000/5000				
Telnet Login with Authentication (with e-Node		✓	✓	✓

Telnet Login without Authentication		✓	✓	✓

INTEGRATION REQUIREMENTS-CONVERGING SYSTEMS CONFIGURATION

NOTE: Converging Systems LED and most Motor Controllers REQUIRE a communication device (either an e-Node for Ethernet connectivity or the IBT-100 for serial connectivity). It is not possible to connect CSI LED or most Motor controllers to an NICE controller in any other way (except if those motor controllers have on-board serial or IP connections). For more information, consult Appendix A and more detailed documentation available on the Converging Systems' website including

-e-Node Commissioning Guide (long version)
-ILC-x00 Intelligent Lighting Controller
-IMC-x00 Motor Controller Manual

IMPORTANT: Converging Systems LED and Motor Controllers REQUIRE a preliminary amount of initial setup/commission which requires the e-Node Ethernet adapter. This is required to set **Z**one/**G**roup/**N**ode addressing as well as to turn specific types of bi-directional communication necessary to have NICE dimmer sliders react to color state changes. For more information, consult Appendix A and more detailed documentation available on the Converging Systems' website including

-e-Node Commissioning Guide (long version)
-ILC-x00 Intelligent Lighting Controller
-IMC-x00 Motor Controller Manual

NICE Configuration

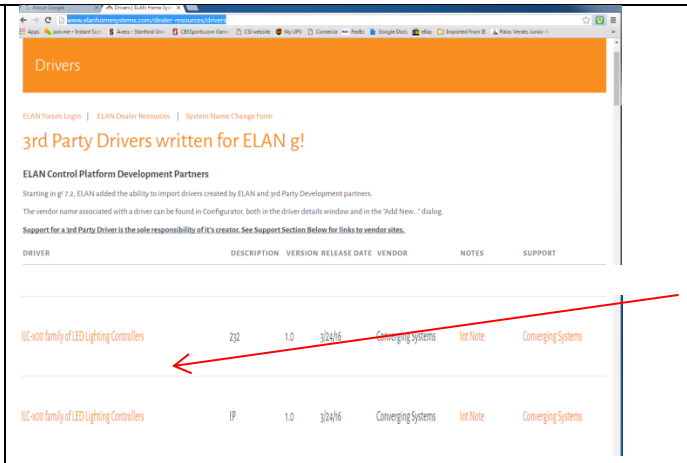
The configuration process will involve loading a lighting communication device (for the e-Node or the IBT-100) and one or more load devices (LED loads). Please follow the below steps to load one or more compiled EDRVC within NICE Configurator.

Installation Process

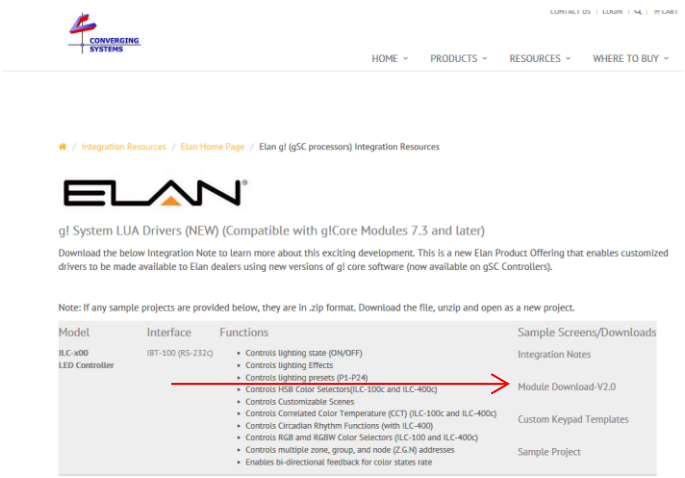
1. Import Converging Systems LUA driver into your project. (Ethernet or Serial as appropriate)

Step	Step	Detail
1a	Download the appropriate Converging Systems' **LUA** driver (see above) into convenient subdirectory below NICE in your Program Files (x86) directory or anywhere else that you can find them. **Note: Make sure that you first unzip the file (CRITICAL) and then**	-Select the appropriate **LUA** driver depending upon if you will be driving your systems through Ethernet using the Converging Systems' **e-Node**, or through RS-232C communication using the Converging Systems' **IBT-100** serial adapter.

consistently place that UNZIPPED .EDRVC file into the EXACT same directory or subdirectory where you placed any earlier such driver so that in the g! Configurator a subsequent driver update can be seamless.



or on the Converging Systems' website
http://www.convergingsystems.com/local_profiles.php



Note: always check on the Converging Systems website for the latest version.

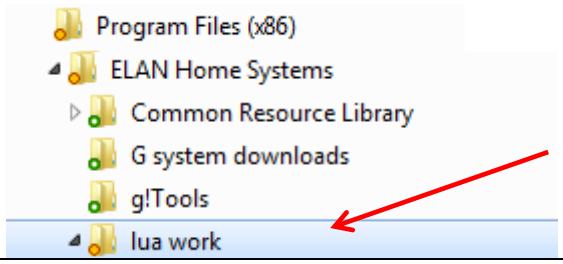
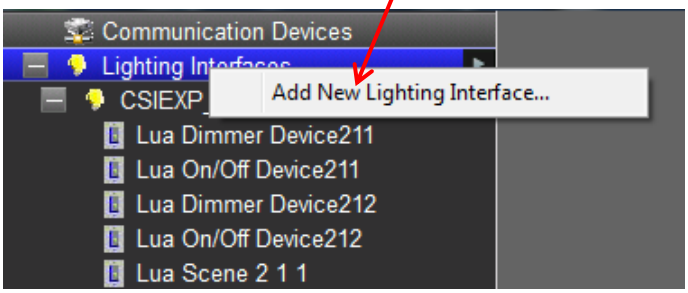
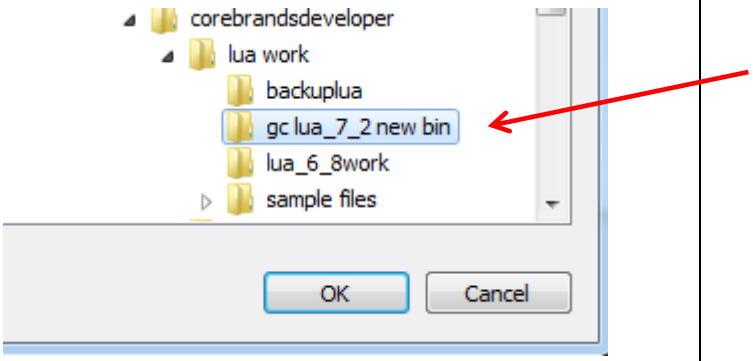
Select the appropriate file as below:

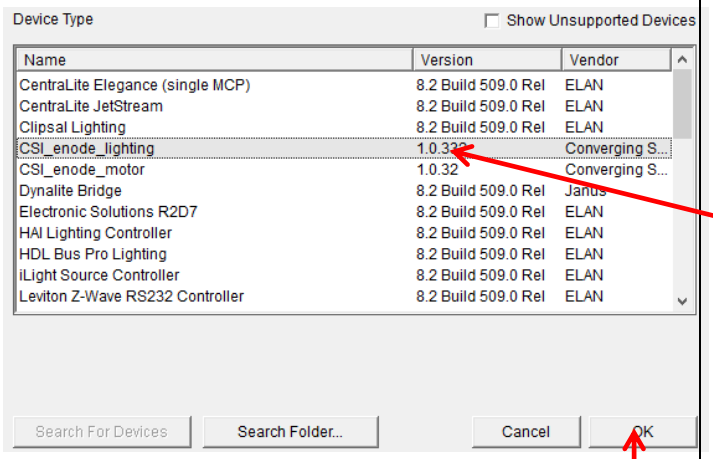
LED Lighting Control

Type of Connectivity	File Name
Ethernet connectivity	*CSI_enode_lighting.EDRVC* file
RS-232c Connectivity	*CSI_IBT_lighting.EDRVC file*

Motor Control

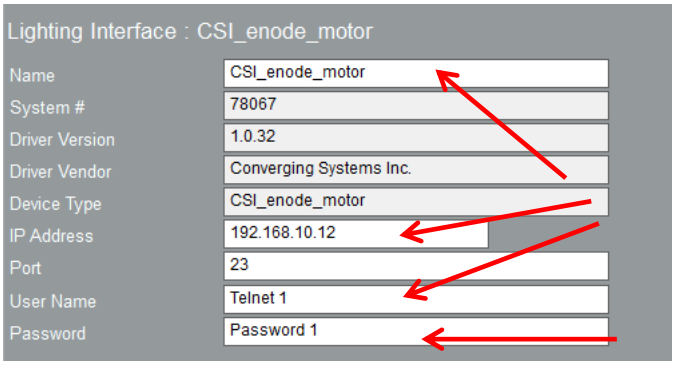
Type of Connectivity	File Name
Ethernet connectivity	*CSI_enode_motor.EDRVC* file
RS-232c Connectivity	*CSI_IBT_motor.EDRVC file*

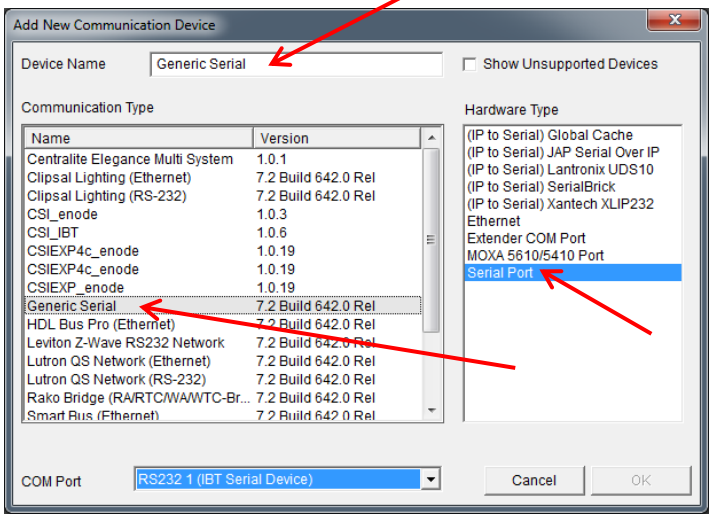
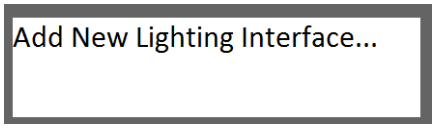
		-Place file within the NICE (NICE) directory on your computer or anywhere else you can consistently and reliably locate that file in the future. 
1b	Import the applicable **LUA driver** into your NICE Project **Note**: Make sure you download latest version from the Converging Systems' website or NICE's (if available) and ensure you know the location of the **extracted** EDRVC driver files on your computer's hard drive. **Note**: See the first page of the integration Note regarding compatibility between various Converging Systems' LUA drivers and particular NICE Core Modules.	-Within your project, go to the **Lighting** Tab, and right click on the **Lighting Interfaces** category to expose the "Add New Lighting Interface…" dialog box.  -Next, select the **Search Folder**. button and navigate to the directory where you placed the .EDRVC file in **Step 1a** above and select that directory. (In this case, the file is located in the ***corebrandsdeveloper*** folder but on your computer this location will vary.)  Hit **OK** to continue. -You will now see a dialog box appear which will show the device driver found. Select the driver name (**CSIEXP_enode** in this case) to continue.

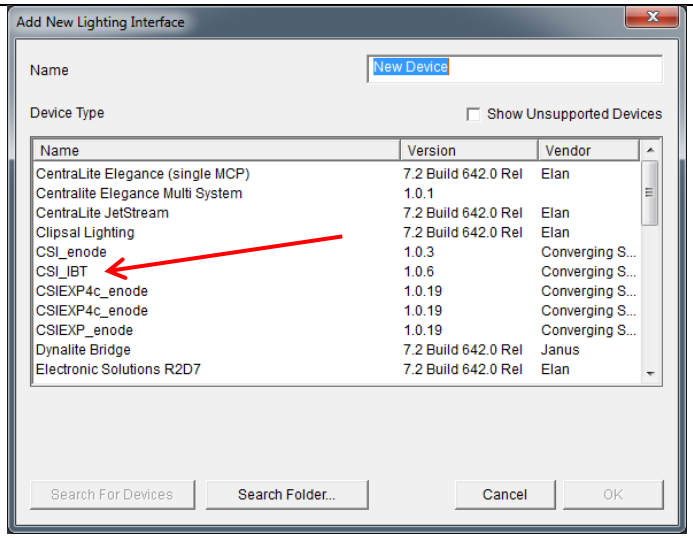
		 Hit **OK** to confirm -Your new **LUA** Driver has now been updated to your NICE Controller.
1c	Updating previous versions of the Converging Systems driver	-Place driver in the same directory as the previous version of the driver was location. -Hit the **Update Driver** within Configuration for you particular communication device being upgraded.  ***Note: if you desire to update to newer versions of the one of the Converging Systems' drivers, it is important that you remember where you placed the original unzipped file on your computer so that you can simply download and unzip the any newer CSI driver into the exact same directory (again unzipped) so that the NICE Driver UPDATE function will work properly. Otherwise, you will have to create a new lighting device and start all over again with each new update.***

2. Set-up communication device for the Converging Systems Communication Device (Ethernet or Serial)

Step	Step	Detail
2a	Set-up communication parameters for the Converging Systems interface (**e-Node IP device** or **IBT-100 serial device**) that will be used with one or more	Determine what will be the communication linkage that you will use to connect to the Converging Systems' device. -Refer to **Step 2b** if you will be using IP Communication and the **e-Node**.

	Intelligent Lighting Controllers (ILC-xx0) or Motor Controllers (IMC-xx0).	-Refer to **Step 2c** if you will be using RS-232c Communication and the **IBT-100**.
2b	Communication Setup for **Ethernet** connectivity (e-Node). This will set up both (i) a Communication Device as well as (ii) a single Lighting Interface (through which lighting controllers will be added in **Section 3** below).	**-**Select the applicable device (e-Node) for which you have loaded the driver in **Steps 1a** and **1b** above. The following data entry box will appear for our example of the **CSIEXP_enode** found.  **Currently, the NICE's LUA development program is ongoing and therefore user interfaces and data fields are subject to change. Certain data fields that may be pictured above may not need to be programmed. See below documentation for current information.** **Name**: This is name of the particular (communication) device loaded. Should you have multiple e-Nodes (for large systems for where you may have one standard e-Node and one e-Node/dmx or multiple standard e-Nodes), make sure you utilize different names for each e-Node to be supported. If you only have one e-Node in your system, just leave the default name unchanged. **User Name**: This is e-Node's Telnet User Name for login authentication. The factory default is **Telnet 1** for latest version e-Nodes (e-Node MKIII) and **E-NODE** for older version (MKII e-Nodes). Note the MkIII has 3 RJ-type ports in a row, while the MkII has just two RJ-25 ports. Unless you have changed the **User Name** within the e-Node Pilot application, simple use the default name provided. **Password**: This is e-Node's Telnet Password for login authentication. By default, from the factory, the Password is **Password 1** (for MKIII e-Nodes) and **ADMIN** for (MKII versions). Unless you have changed the Password within the e-Node Pilot application, simple use the default name provided. **IP Address**. This is IP address for the particular e-Node being used as the communication device. The IP address can be determined by either using the e-Node Pilot application or

		by discovering the e-Node using Windows' UPnP discovery mechanism with Windows. Consult the e-Node manual for more information. **Port**. By default, Telnet communication utilizing Port **23** is supported by this driver. Therefore, you do not need to change this field.
2c	Communication Setup for RS-232c connectivity (**IBT-100**). This will set up both (i) a Communication Device as well as (ii) a single Lighting Interface (through which lighting controllers will be added in **Section 3** below).	- Select the **Lighting** tab and right click on **Add New Communication Devices** and scroll down to pick a **Generic Serial Type.** Under **Hardware Type** pick **Serial Port** and under **Device Name** provide a unique name for the serial port that will be utilized for the IBT-100. In this example, it will be **called IBT Serial Interface. Select the COM port that will be used to connect to the IBT-100.**  Click **OK** to continue. -Next **right click on the Lighting Interfaces** tab to expose the following pop-up.  -Select this pop-up and the following screen will appear enabling you to establish communication parameters.

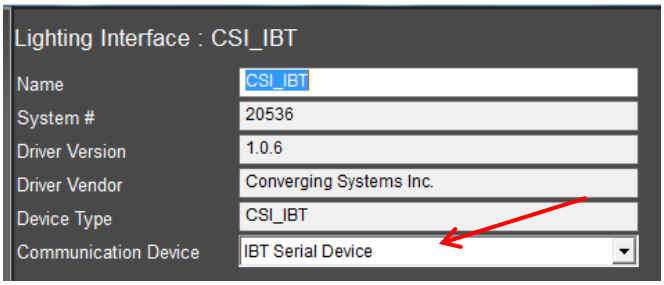


-You may have to select **Search Folder** button and navigate to the location where the Converging Systems applicable .EDRVC file is located. Select **the CSI_IBT** as the Device Type.

-Enter a name in **Name** field to help you identify which device will be controlled

-Select the **CSI_IBT** driver. Select **OK** to proceed.

-Left click on your new **Serial Lighting Interface**. This page will appear.



-Select the Communication **Device**, and select the **IBT-100 Serial Device**

-Select **Apply** to continue**.**

3. Set-up Lighting/Motor Controllers (i.e., ILC-x00 or other similar CSI controller) for the Converging Systems driver.

New Developments within the NICE/Converging Systems driver technology (V 1.25 or later) now allow nearly seamless and instantaneous discovery of all lighting controllers and their internal feature sets* (when initially discovered and connected with the e-Node), regardless of whether those devices are (i) monochrome (ILC-100m or IMC-400/monochrome mode, (ii) RGB (ILC-100c/ILC-300) or (iii) RGBW (ILC-400/ILC-450 RGBW mode). For documentation related to this new driver feature, follow the instructions within this section.

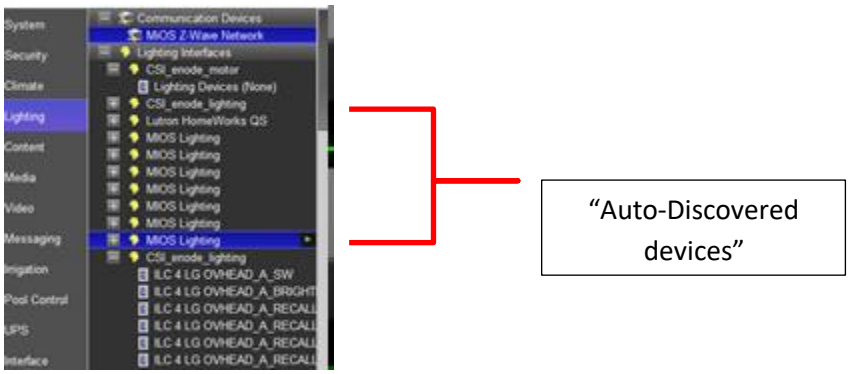
For the support of DMX fixtures using the e-Node/dmx (where the number of potentially supported devices and the resulting auto-generated device list would be extremely long) as well as for the support of motor control devices, see the separate directions in Appendix 4.

***Note:** Because the feature set available within the ILC-x00 family of LED controllers is always expanding, it may be necessary for an integrator to add new command(s) (NICE calls this **Add New Devices**) manually even after an Auto Discovery is performed. That insures that the NICE/Converging Systems interface is future proofed even without new NICE driver updates. For directions on how to add new NICE Lighting Devices (i.e. Add New Devices) that might handle an extra **Effect**, or **Preset,** or **Store,** or an extra unsupported command, please refer to Appendix 4 for directions on how to add specific types of features and how to map those to particular NICE GUI objects.

A. Background on Auto Device Discovery (Discover Devices)

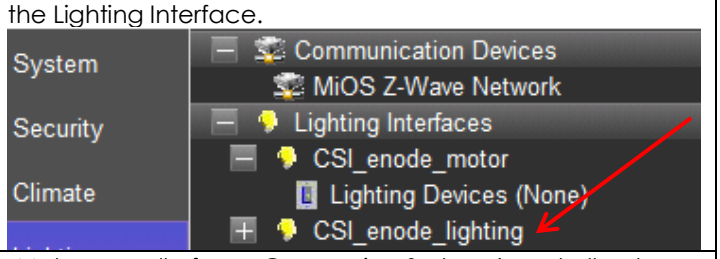
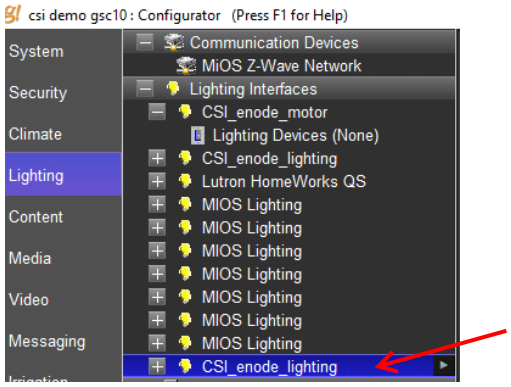
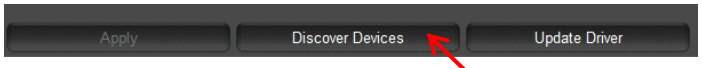
The **Discover Devices** button permits the automatic discovery of a generous set of" NICE Lighting Devices" than can be quickly mapped to NICE sliders, buttons or other user controls within NICE Configurator. After initiating a **Discover Device** operation*, a number of "Auto-Discovered Devices" will AUTOMATICALLY appear under the **CSI_enode lighting** interface (see "Auto-Discovered Devices" below).

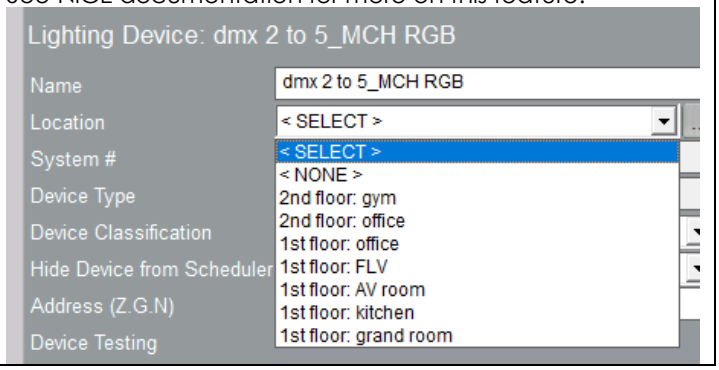
Figure 1

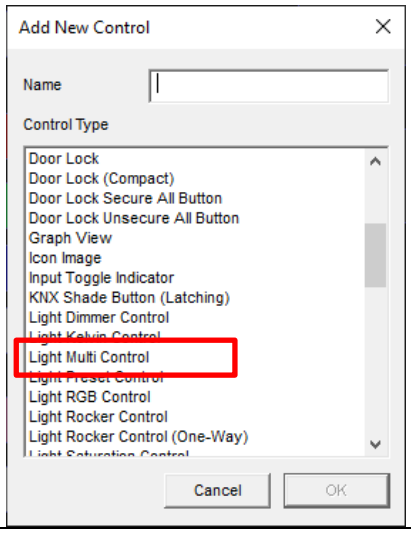


*Note: this is provided the **CSI_enode_lighting** interface is first discovered in Step 2 above-t***this Auto Discovery does not work with the IBT-100 serial interface adapter because there is no active XML database from which to query using serial communication***.

B. Step by Step Directions to Discover Devices

Step	Step	Detail
3a	Make sure the **CSI_enode_lighting** lighting interface is populated under the Lighting Tab under Lighting Interfaces	If you do not see this entry, go back to Step 2 to discover the Lighting Interface. 
3b	Discover Devices	-Make sure all of your Converging Systems' controllers have been properly discovered with the e-Node Pilot application (within the Component Software Setup section), and -Make sure all controllers have been assigned unique Zone/Group/Node ("ZGN") addresses again within the Component Software Setup section and as additionally detailed within Appendix 3. -Next Highlight the **CSI_enode_lighting** entry to reveal these buttons on the bottom of Lighting Interface page   **THEN AND ONLY THEN**, select the **Discover Devices** button on the bottom of the Lighting section. Please be patient—depending upon the number of devices and their type, this Auto-Discovery process could take 20 or more seconds. **Why is this important**: Depending upon your set-up you may have dozens of controllers with 10 or more entries (features)

		auto-populated all with factory default address (containing a zero) or improper addresses (not relevant for your particular installation perhaps) that would take an extremely long period of time to manually correct when in fact the Discover Device function will auto-generate all correct entries provided good information was initially available. Remember the old adage "Garbage In-Garbage Out." **We cannot stress how great the Discover Device feature is but only when it is used properly as documented above.** **Note: After you have Auto Discovered Devices, do not re-Auto-Discover devices again without first highlighting all previously Auto-Discovered Devices and first deleting those. The system will not selectively update entries—it must start from a clean slate.**
3c	Auto Discovery will occur	Underneath the **CSI_enode_lighting** will appear a number of "New Devices" that can be used in the next section to build GUI pages. **In general, if the above steps have been carefully followed, no changes to these auto-generated devices will be need to be made.** Note all the relevant and necessary fields will have been populated automatically from original settings set with the e-Node Pilot application. You can make any changes as necessary after the auto-generation process.
3d	Create any new Lighting Devices as required	The Converging Systems' software architects have made some general determinations as to the type and quantity of Devices that are auto-discovered. For instance, we have established 1 or 2 **STORES** (scenes) and 1 or 2 **EFFECTS (sequences of colors with varying dissolve rates)** while many more entries are possible. You can simply examine the model from which you wish to duplicate the entry and carefully make copies with new entries as required. For more information on creating new Devices, see Appendix 4.
3e	Add new **Area** parameter within any motor or lighting device	-Open a device that you wish to link to a specific area. Select either None or create your own new area. See NICE documentation for more on this feature. 

3f	Convert old Lighting Types to New Types (with advent of NICE 8.6)	The latest Converging Systems drivers intelligently review pre-existing drivers already in a project and once "Discovery Device" is selected, existing MULTI_CH_RGB device are automatically converted to the new MULTI device thus saving the installer time to manually add the new device type. **IMPORTANT**. This auto-conversion only occurs when merited (that is the device to be supported is of the correct type and present). You will still have to: -add a new LIGHT CONTROL MUTLI into your custom page -and link that UI control to the applicable **MULTI** device (newly auto-created). 

4. Create (or Modify) Various User Interface (UI) Controls for (i) Hue/Sat/Brightness or Red/Green/Blue adjustments, (ii) ON/OFF adjustments, and (iii) Scene adjustments.

Depending upon the specific Converging Systems' software command desired, the matching NICE lighting/dimmer/scene/on/off/etc. control when selected has an autogenerated drop down table to **ALL** Auto-Discovered devices to which the UI control can be linked. As NICE increases the range of operations possible, Converging Systems' driver can be enhanced to automatically support those new operations.

The Table below are a summary of the currently supported NICE UI controls. These summary tables include two distinct types of UI types available.

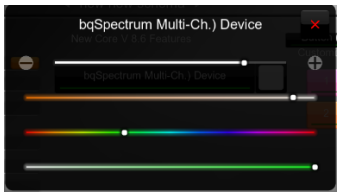
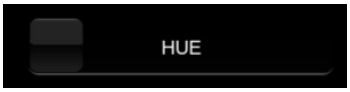
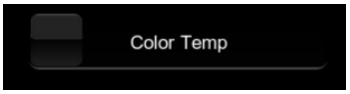
-"**Connect To.." Types (Table 5a)**. Those User Interface controls which can be used to link features through the "Connect To" pulldown. **Note**: For ease of programming and setup try to use as many Connect To.. features as possible in lieu of the below Event Map types.

-"**Event Map" Types (Table 5b)**. Those User Interface controls which can be used to link features only through Event Maps

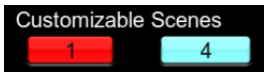
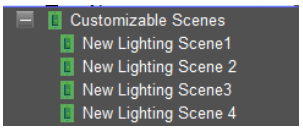
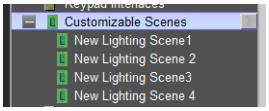
A. Background on UI Types

Table 3

Recommended UI types that are able to be auto-selected with "Connect To.." feature within Properties

NICE UI Library Choice	Entry within NICE Configurator (under Lighting Interface)	NICE Lighting Device	Application
	**Controller alias** **__MULTI** **(for RGBW devices with Kelvin)**	Light Multi Control	**Popup Slider** adjustment for -**Hue**, -**Sat** -Brightness (a.k.a. **SET**) -**Kelvin**
	**Controller alias** **__MCH_RGB** **(for RGB devices without Kelvin)**	Light Multi Control	**Popup Slider** adjustment for -**Hue**, -**Sat** -Brightness (a.k.a. **SET**)
 ....  ...   **SLIDER** **Note: Do not select** *Light Dimmer Control which was the case in Pre-Schema 59 CSI drivers*	**Controller alias** **__MCH RGB: SET** **Controller alias** **__ MCH RGB: HUE** **Controller alias** **__ MCH RGB: CCT** **Controller alias** **__ MCH RGB: WHITE** **Note:** Besides the above, other parameters (found	(RGB Multi-Ch.) Device But use SLIDER object to control this device –not LIGHT DIMMER **Note**: The innovative device dynamically displays all available choices (shown in right-most column) under Connects To.. box within Properties without having to individually enter each type manually- Cool new Feature.	**Slider** adjustment for -**Hue**, -**Sat** -Brightness (a.k.a. **SET**)* -**Red** -**Green** -**Blue** -**White** -**CCT** (Color Temp) -**SUN** (Circadian Rhythm) *Note: **Dim Level** may also be used here in lieu of **SET**.

	in 4th column here) are auto populated and available as ___ **MCH RGB**:xxx and will appear as separate selectable devices for each Z/G/N address discovered.		
 **Light RGB Control** **Note: With Core 8.6 this is available but it is recommended to use the MULTI control instead now**	**Controller alias__MCH RGB** **Note:** When using this control, it is recommended to also use a separate SLIDER for Brightness **(SET**-see above row). When using with RGBW/tunable white environments, you may also want to add a SLIDER for **WHITE** to control this 4th channel.	(RGB Multi-Ch.) Device Use LIGHT RGB object to surface this control **Note**: The innovative device works here as well with the **Light RGB Control** GUI element to combine Red, Green and Blue parameters into a unified transmission packet to act as a surrogate for our preferred HUE (Slider). **Note**: with Core 8.6 we translate this control's output to HUE, rathe than the bus intensive and not as accurate RGB format which had no control over Saturation.	**Light RGB Control color picker** for -Red/Green/Blue color selection
 **Light Dimmer Control**	**Controller alias__Brightness** **Note:** This control is only used with monochrome LED controllers.	Dimmer Device	**Slider** adjustment for -Monochrome Brightness (a.k.a. **SET**)*
 **Light Toggle Control**	**Controller alias__SW** **Note:** A separate device must be installed for	On/Off Device Or Scene (optionally)	**Button (Standard**) -On -Off

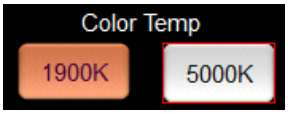
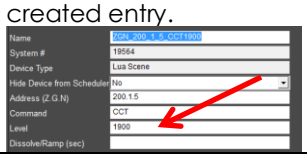
(with capability for dissolve setting)	ON/Off button set for each Z/G/N address	**Note**: Auto Discovery creates this device only as a Lua ON/OFF device.	
 **Light Scene Button (Customizable)** Scene button	Customizable Scenes entries must be created by dealer (see right for more information)  No Additional Devices Required here (as long as these are created) **Note:** New NICE UI populates all available Devices (seen elsewhere within this table) for user selection	Can point ot - Light Multi Control -(RGB Multi-Ch.) Device --Scene	**Customizable Scene buttons** -Scene 1 to n  **Note**: Installer adds sufficient number of entries here to allow selection of any -Light Control Multi -Dimmer (RGB Multi-Ch.) Device -Scene" Devices, This is a very useful UI control that can simplify lighting control for the system owner. Following are examples that can be selected through Customizable Scenes which allow the end-user to make scene selections on the fly from available choices including: -**New MULTI Control** -**Recall** (1 to 24) -**Store** (1 to 24) -**Effect** (1 to n) -**Hue Levels** (1 to 240) -**Sat Levels** (1 to 240) -Brightness (**SET**) **Levels** (1 to 240) **-Combined RGB Levels** (red, yellow, green, cyan, etc.) -**Red Levels** (1 to 240)

			-**Green Levels** (1 to 240) -**Blue Levels** (1 to 240) -**White Levels** (1 to 240) -**CCT Levels** (1700 to 7000K) **-Circadian Levels** (sunrise "0" to midday sun "240")

Table 4

Alternate UI Button types that can be used with Event Maps for alternative control.

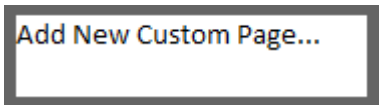
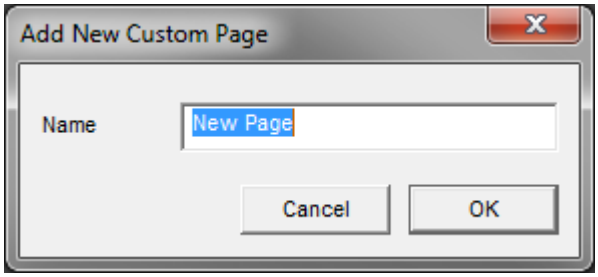
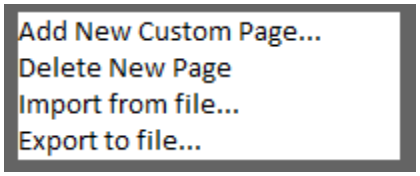
NICE UI Library Cholce	Entry within NICE Configurator (under Lighting Interface)	NICE Lighting Device	Application
 **Button Standard** Scene Select-recall Preset	**Controller alias__RECALL1** **Note:** A separate device must be installed for each Recall desires. The Auto-Discovery process poplates a small number which can be manually duplicated by the Installer.	Scene	-**Recall 1… Recall n**
 **Button Standard** Scene Store (store Preset)	**Controller alias__STORE1** **Note:** A separate device must be installed for each Recall desires. The Auto-Discovery process poplates a small number which can be manually duplicated by the Installer.	Scene	-**Store 1… Store n**
 **Button Standard** Recall Effect buttons (with capabiilty for dissolve)	**Controller alias__EFFECTn** **Note:** n reflects one of the various Effects auto-generated.	Scene **Note**: A single device must be installed for each Z/G/N address to be supported as well as **for each** discrete index references (i.e., Effect1, Effect 2, etc.)	-**Effect 1 ..Effect n**

 **Button Standard** Fade Level controls	Provided a **Controller alias__BRIGHTNESS** entry is populated (Auto-Discovery generates it), this type of control can be used to control brightness of monochrome devices <u>Quick Primer on Event Maps.</u> -Right click on targeted button to **Edit Event** Map. -Create new mapping (**Add**) and select appropriate library (System Family-Lighting System -**Brightness** (entry for appropriate Z/G/N) -**Set Level** -**Options** (and pick %)	Relies on Dimmer Device **Note**: A **single** Dimmer Device is utilized to create any number of % buttons using Edit Event Map for each ***<u>monochrome</u>*** controller with a unique Z/G/N address.	- **Monochrome Brightness Level** to pick a Particular level setting
 **Button Standard** Color Temperature Selection **Note: With Core 8.6 this is available but it is recommended to use the new Customizable Screen popup to select up to 16 presets that can be specific colors or CCTs**	**Controller alias__CCTxxxx** **Note**: Auto Discovery creates a placeholder value of 2700K. Additional Devices can be created specifying other Color Temperature Values (CCT) by simply inputting the desired CCT value into the Level field for each newly created entry. 	Scene **Note**: A single device must be installed for each Z/G/N address to be supported at a specific color temperature.	- **Color Temperature Setting** to pick a particular level setting

B. Step by Step Directions to Create UI

Prior to the development of the Auto Discovery feature (**Discover Devices**) within the Converging Systems driver for NICE, the above User Interface (UI) features and the required programming was a bit complicated, but now with Auto Discovery, the creation of these UI objects is extremely SIMPLE and QUICK. Please follow the directions below to automatically discover all of the above Device types which can be

then linked or connected to UI objects to quickly build your user interface with Converging Systems LED or Motor control technology.

Step	Step	Detail
4a	You can create a UI for your system that is suited to your customer's requirements. This **Integration Note** references some pre-programmed UI pages that you may find useful. They contain sliders and buttons which are uniquely developed to control Converging Systems' loads (LEDs in this case). This step will show how to import Converging Systems pre-programmed pages that you can edit and re-use for your own project.	-Go to the **Lighting** Tab and right click on **Custom Pages** , The following popup will appear  Select this task and the following popup will appear.  Select an appropriate name and hit **OK**. -Hover over the **New Page** now listed under Custom Pages and right click to expose this popup.  Select **Import from File** and browse for the ILC Ethernet Control LUA.ECV file available from the Converging Systems website. Click **OK** to import. http://www.convergingsystems.com/local_profiles.php Here is an example of a sample on which you can now begin working

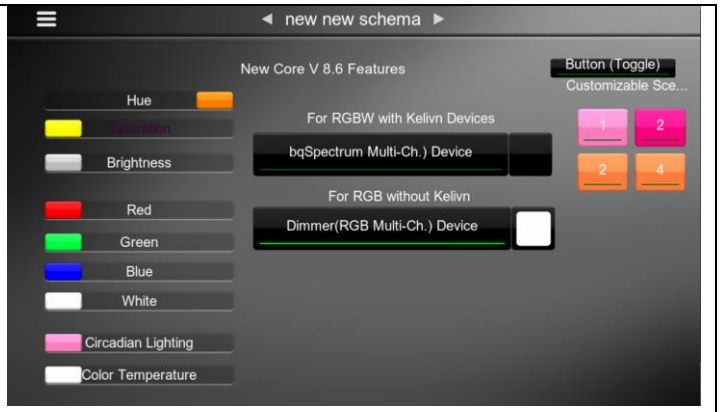
<table>
<tr>
<td></td>
<td></td>
<td> 

Figure 2</td>
</tr>
<tr>
<td>4b</td>
<td>Now let us understand how buttons and sliders are generally created and programmed to trigger specific events.

Note: for more detailed information consult the tables to the right.</td>
<td>Currently, there are specific types of Lighting Devices that are relevant for lighting control user interfaces and specfic types of other Devices that are relevant for motor control user interfaces. Refer to the Tables below which identifies these types.

Table 5 (for Lighting Devices)
<table>
<tr><th>User Interface Type (see Table 5a and Table 5b for more information)</th><th>NICE Control Type</th></tr>
<tr><td>RGBW control (with Kelvin)</td><td>MULTI Device</td></tr>
<tr><td>RGB Control</td><td>MULTI Device</td></tr>
<tr><td>Additional sliders to additional control (i.e., Circadian or RGB or W control)</td><td>SLIDER (which points to avaialbe controls)</td></tr>
<tr><td>On/Off buttons (with capability for dissolve setting)</td><td>Light Switch Control</td></tr>
<tr><td>On/Off Toggle</td><td>Light Toggle Control</td></tr>
<tr><td>(Customizable) Scene button</td><td>Light Scene Button (customizable)</td></tr>
<tr><td>Slider for Brightness Control of Monochrome lights</td><td>Dim Level</td></tr>
<tr><td colspan="2">Legacy Controls (probably best to use above controls with new installations using 8.6 or later)</td></tr>
<tr><td>Light RGB Color (older color picker)
Note: with Core 8.6 R/G/B color</td><td>Dimmer (RGB Mulit-Ch.) Device</td></tr>
</table></td>
</tr>
</table>

picker now transmits HUE color information. It is necessary to set Saturation and Brightness separately.		
To be Used with Event Maps only		
Recall/Store/Effect buttons (with capabiilty for dissolve)	**Button (Standard)**	
% Set button	**Button (Standard)**	
Color temperature setting button (not slider)	**Button (Standard)**	

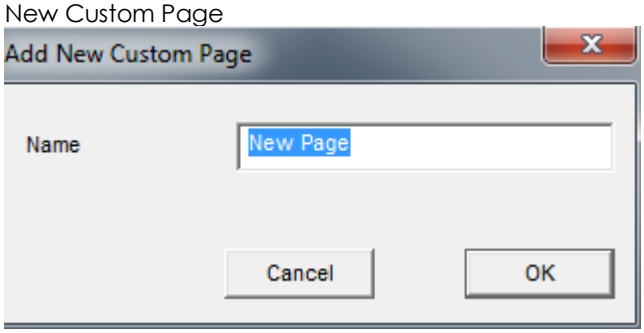
Table 6 (for Motor Devices)

User Interface Type	NICE Device Type
Slider (for motor position)	**Light Dimmer Control**
UP/Down/Stop buttons	**Button (Standard)**
Store Position	**Button (Standard)**
Recall Position	**Button (Standard)**

Note: Currently only the above Device types are relevant to the Converging Systems family of LUA drivers. ***Over time additional type devices may become available which may increase the functionality of choices available to the installer.***

Provided you crated the requisite number of Lighting (or Motor) Devices, then all you have to concern yourself here is to make sure the **Address Tag** is accurate and when required you create an Event Map joining available commands to programmed devices.

NOTE: IF YOU DID NOT CREATE THE REQUISITE NUMBER OF DEVICES IN SECTION 3 ABOVE THROUGH DEVICE DISCOVERY, YOU WILL NEED TO CREATE AS MANY DEVICES (of the three or more Device Types available) FOR THE NUMBER OF SLIDERS OR BUTTONS REQUIRED RELATED TO A SPECFIC Zone/Group/Node ADDRESS. See Appendix 4 for more information.

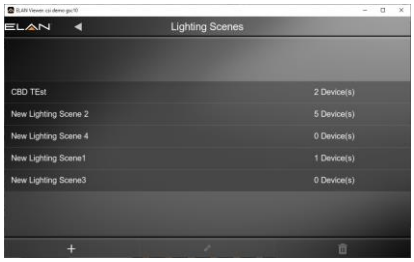
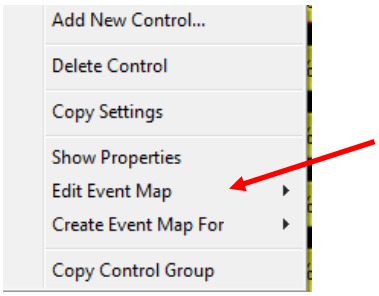
<table>
<tr>
<td>4c</td>
<td>Create applicable UI controls to control targeted operations</td>
<td>-Right click on the Custom Pages entry, to expose “Add New Custom Page
 

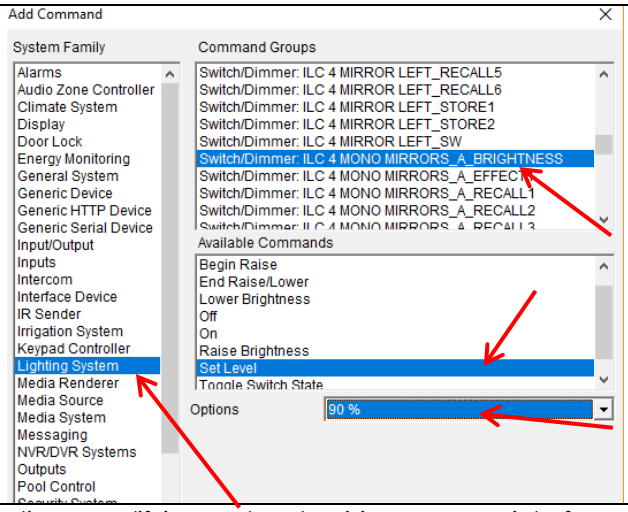
-Name the new Page and begin entering UI controls applicable specfied in the above two Tables.

-Continue entering controls until you have completed the current New Page</td>
</tr>
<tr>
<td>4d</td>
<td>Connect Controls (where applicable) to previously programmed Devices.</td>
<td>-There are two ways by which a UI control is programmed to control a Device programmed within Section xx of this Integration Note. The first of which is through the “Connect To” box within the Properties pop-up within NICE Configurator for the UI control. The second of which is through the Event Map feature within NICE Configurator. In cases where the Connect To box is not exposed (i.e. Button (Standard)), only the Event Map method is applicable.

-Refer to the Table below for a subset of currently supoorted UI types and the method by which those UI types are programmed to interact with previously programmed Devices.

Table 7
<table>
<tr><th>UI Control Type</th><th>Connect To.. compatible</th><th>Event Map</th></tr>
<tr><td>Light Switch Control</td><td>Yes</td><td>Available</td></tr>
<tr><td>Multi Control</td><td>Yes</td><td>N/A</td></tr>
<tr><td>Dimmer(Multi-Ch) Device</td><td>Yes</td><td>N/A</td></tr>
<tr><td>Light Toggle Control</td><td>Yes</td><td>Available</td></tr>
<tr><td>Light Dimmer Control (monochrome)</td><td>Yes</td><td>Available</td></tr>
<tr><td>Light Scene Button</td><td>Yes</td><td>N/A</td></tr>
<tr><td>Button (Standard)</td><td>N/A</td><td>Available</td></tr>
<tr><td>Customizable Scene Button</td><td></td><td>Using the new Lighting</td></tr>
</table></td>
</tr>
</table>

		<table><tr><td></td><td></td><td>**Scheduling NICE feature***</td></tr></table> *Note: control of preset colors and CCT can be accessed now though the NICE Lighting Scene feature  -For each UI Control specified above which has a "**Connect To"** data entry location, right click on that control to generate the **Properties** box. -Customize each of your controls as desired and where presented with a "**Connect To"** box, select from the pull down menu the applicabel Lighting (or Motor) device programmed for that UI. In this example, we are selecting the **SAT** slider previsously progarmmed with **Z/G/N** address of 2.1.1 to be tied to the targeted slider. -Continue programming all UI controls
4e	Program **Event Map** information for UI controls that do not support the **Connect To** function	-Right click on any UI control for which there is not a **Connect To** data field available to exposer this pop-up  -Either select the **Create Event Map For** option if there is not a **Edit Event Map** showing, Within the **Event Map** popup, program the desired operation to the previously programmed Device and to its specific operator. In this case, for a 90% percent fade button for following data fields are selected/entered.

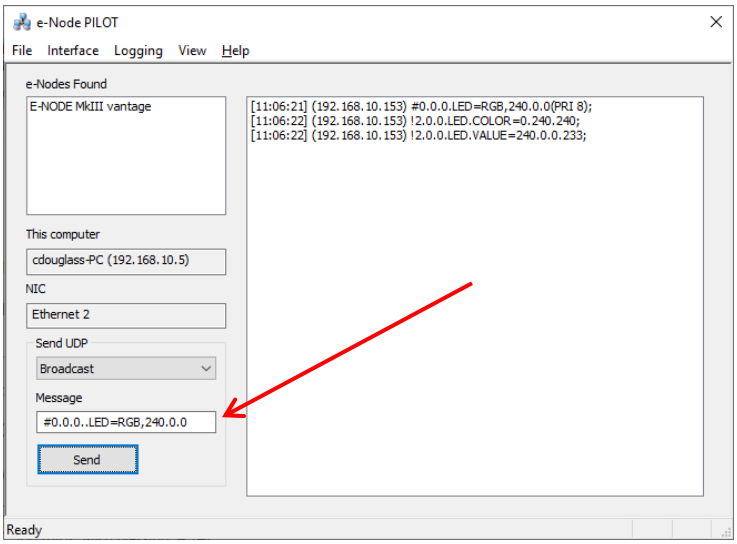
		
4f	Finish up your User Interface	Continue modifying and customizing your user interface as required. When you are done just hit **Apply** to upload all code changes to your NICE processor.
4g	Examples	See Appendix 4 for examples of new UI controls

5. Test

5a	Launch the NICE Viewer and select a programmed button to operate.	Make sure your eNode/IBT-100 and connected controllers are properly working and tested using e-Node Pilot. Observe your connected LEDs (or motors) and see if they operate properly. If so, you have successfully interfaced Converging Systems' controllers. If they do not operate, proceed to the next section.

6. Troubleshooting

6a	Launch the Converging Systems' Pilot application which communicates with the Converging Systems' e-Node Ethernet bridge. This application can be downloaded here. https://www.convergingsystems.com/downloads_library.php (see first entry)	Within the Pilot application, select the **View Map** Tab and **Discover** e-Nodes and Devices. Then go to the **Traffic** Tab, and enter the following command to see if your e-Node and connected LED controllers are properly functioning. **#0.0.0.LED=RGB,240.0.0** **Or** **#0.0.0.LED=RGBW,240.0.0.0**

		 The connected LEDS should turn RED Consult the e-Node documentation or see Appendix 8 for more troubleshooting information (if supplied).

NICE CONFIGURATION DETAILS

The following table provides settings used in Configurator …Please refer to the Configurator Reference Guide for more details. The first table indicates IP settings for the e-Node Ethernet device. The next table shows RS-232c settings for the IBT-100. The final table shows settings for various supported Device Types.

Note: Currently only four (4) types of Lighting devices are available with the current release of LUA tools. These are as follows:

-Lua On/Off Device

-Dimmer (RGB Multi-Ch. Device) Note: for non-monochrome devices

-Lua Dimmer Device Note: for monochrome devices only

-Lua Scene Device

Accordingly, no other functions other than those available in these four devices are currently available.

In the table below:

- "<User Defined>", etc. Type in the desired name for the item.
- "<Auto Detect>", etc. The system will auto detect this variable.

Table 8 e-Node Ethernet Communication

Devices	Variable	Setting	Comments
Communication (Lighting Interface)	**Name**	<User Defined> (Typical CSIEXP_enode)	
	System #	<Auto Detect>	
	Driver Vendor	Converging Systems Inc.	
	Device Type	CSIEXP_enode	
	User Name	**Converging Systems e-Node**	
	Driver Version	<Auto Detect>	
	Driver Vendor	Converging Systems	
	IP Address	<User Defined>	
	Port	<Auto Detect> (Default 23)	The field is discovered automatically.

Table 9 IBT-100 (Serial Communication)

Devices	Variable Name	Setting	Comments
Communication (Serial Port)	**Name**	<User Defined> (Typical IBT Serial Device)	

	Device Type	<Auto Detect> (Default Serial Port / Standard Configuration)	
	COM Port	<User Defined>	
	Protocol & Other Serial settings	<User Defined> (RS232, 57600,None, None, 8,1)	

Table 10 Device Type (regardless of Communication Device Selected above)

Device Type Supported	Variable Name	Setting	Comments
Dimmer (RGBW) Device for each ILC-400 or ILC-450 load or 4 ch DMX fixture with CCT)	**Name**	<User Defined> (Default **MULTI Device**)	
	System #	<Auto Detect>	
	System #	<Auto Detect>	
	Command	<User Defined> **Note** Depending upon type of dimmer/slider you must customize the entry as appropriate. See **Dimmer Device Parameter Table** below for choices.	
Dimmer (RGB) Device for each ILC-100sa/ILC-300 or 3ch DMX fixture)	**Name**	<User Defined> (Default **MULTI_CH_RGB Device**)	
	System #	<Auto Detect>	
	System #	<Auto Detect>	

	Command	<User Defined> **Note** Depending upon type of dimmer/slider you must customize the entry as appropriate. See **Dimmer Device Parameter Table** below for choices.	
Lua Scene (for each ILC-xxx load)	**Name**	<User Defined> (Default **Scene**)	
	System #	<Auto Detect>	
	Device Type	<Auto Detect> (Default **Scene**)	
	Address Tag	<User Defined> **Note** Enter in format **Z.G.N** (with **periods** between the Z & G & N entries,	
	Level	<User Defined> **Note** Enter reference number for specific index related to command (i.e. Preset **1**, Effect **1**, etc.) device	
	Dissolve/Ramp (sec)	<User Defined *dissolve rate*> **Special Case For Effect=1 and Effect=4 only:** A secondary data value for Sequence Rate (Seq Rate) can be entered ***after a comma*** following the initial <user defined dissolve rate> entry as follow: <user Defined dissolve rate>,<user Defined Seq Rate> **Note** Enter integer value from 0 to highest supported value (in	**Dissolve Rate** is the time in seconds to transition from one state to another for a particular Dissolve feature (X) **Seq Rate** (which is used with Effect(1) and Effect(4)) specifies the time (after any dissolve) that the preset color is maintained before transitioning to the next color in sequence.
Lua On/Off Device (for each ILC-xxx load)	**Name**	<User Defined> (Default **On/Off Device**)	
	System #	<Auto Detect>	
	Device Type	<Auto Detect> (Default **On/Off Device**)	

	Address (Z.G.N) Tag	<User Defined> **Note** Enter in format **Z.G.N** (with periods between the Z & G & N entries)	
	Command	No required entry	
	Level	No required entry	
	Dissolve/Ramp (sec)	<User Defined> **Note** Enter integer value from 0 to highest supported value (in seconds)	

Table 11 **Dimmer Device Command Table**

Dimmer Type	Command
Hue	HUE <entry for a HUE slider in HSB color space>
Sat	SAT <entry for a Saturation slider in HSB color space>
Brightness	SET <entry for a brightness/fader slider in HSB color space>
Red	RED <entry for a RED slider in RGB color space>
Green	GREEN <entry for a GREEN slider in RGB color space>
Blue	BLUE <entry for a BLUE slider in RGB color space>
White (only for RGBW device-not for RGB device or monochrome device)	WHITE <entry for a WHITE slider in RGB color space with the ILC-400c controller>
CCT (for Color Temperature)	CCT <entry for a Correlated Color Temperature slider>
SUN (for Circadian rhythm)	SUN <entry for a Circadian Tuning slider with the ILC-400 controller>

COMMON MISTAKES

1. Forgetting to set TELNET credentials for Converging Systems e-Node device within the Lighting Interface page. Typically, Telnet sessions require a LOGIN ID. Typically, you should use Telnet 1 for the username and Password 1 for the password. Pay attention to the initial cap and the space in the credentials.

   Also, within the NICE setup, Telnet is used with LOGIN. IF the LOGIN setting within the e-Node is set to **DISABLE**, the NICE processor will be unable to establish a Telnet session with the e-Node. Make sure it is set to ENABLE to enable this feature. If you have changed this feature within e-Node Pilot, you must hit the **RESTART** button in order for this change to become valid.

2. Forgetting to update Zone/Group/Nodes addresses within the default serial or IP driver for specific controllers. The default driver from Converging Systems is set to **2.1.0** for lighting devices, and **1.1.0** for motor devices. The "0" in the last location refers to a wildcard setting which causes all devices with a Node address from 1 to 254 to respond. If you have a setup with uses specific addresses other than **2.1.1** for instance (i.e. **2.1.2** for the second controller, **2.1.3** for the third controller, etc.) you must update the serial or IP driver accordingly.

3. Forgetting to enter or select the Command entry for suffix for sliders (RED, GREEN, etc., or spelling them wrong).
4. Using commas between the Zone/Group/Node entries instead of periods (within the Address Tag)
5. Forgetting to enter a numerical entry within the Level Tag for Effects, Recalls and Presets.
6. Forgetting to enter a numerical entry within the Dissolve/Ramp Tag for Device types which support Dissolve.
7. Forgetting to enter a secondary numerical entry within the Dissolve/Ramp Tag for Effect 1 and Effect 4 if you desired to vary both the Dissolve Rate as well as the Sequence Rate.
8. Make sure that you do not use the Communication Device created by more than one Generic Serial Device or Generic Ethernet Device.
9. Forgetting to create a Generic Serial Port when utilizing the IBT LUA driver for communication with the IBT-100.
10. You must update the driver, post 8.6, to use the new NICE Lighting UI
11. You must hit the Discover Devices after updating the Converging Systems driver.

Appendix 1

Converging Systems System Setup/Configuration

Before proper operation between the Converging Systems' controllers and the NICE system can begin, it will be first necessary for most applications to setup the Converging Systems' products and then commission them for integration with a third-party platform. In case you have not previously configured a Converging Systems controller product, please refer to these instructions below.

INTEGRATION REQUIREMENTS-CONVERGING SYSTEMS CONFIGURATION

The Converging Systems e-Node is an Ethernet communication device which can be used to connect the NICE Host to one or more Converging Systems motor and/or lighting controllers. Alternatively, the Converging Systems' IBT-100 serial interface device can be used alternatively to connect the same number of Converging Systems' controllers to an NICE processor in situations where Ethernet communication is not desired (but where bi-directional feedback is still required).

However, regardless of whether you desire to interface ***more than one*** lighting controller (or motor controller) each with its own controllable operation (i.e. its own **Z**one/**G**roup/**N**ode or **Z/G/N** address) with either the e-Node (Ethernet) or the IBT-100 (RS-232c communication), and/or you desire ***bi-directional communication/feedback*** between your user interface (UI) and a particular motor or lighting controller, **you must still follow the directions below under (i) e-Node Programming** and **(ii) Device Commissioning** in order to establish **unique ZGN address(es) for connected loads** and **turn on the NOTIFY command** which provides for that bi-directional communication.

Note: If you plan on utilizing the IBT-100 for serial communication and (i) ***you will not need*** more than one address other than the factory default **ZGN** address of 2.1.0 for lighting controllers or 1.1.0 for motor controllers, and (ii) ***you do not need bi-directional communication*** between the lighting load or the motor load and your User Interface, then you can proceed to the IBT-100 Set up Section and you may skip the (i)

NOTE: It is recommended that the Converging Systems controllers as well as the e-Node Ethernet gateway) are running the latest version of firmware available at the time of installation. Directions for uploading new firmware on contained on the Converging Systems website.

WIRING DIAGRAM (for IP connection)

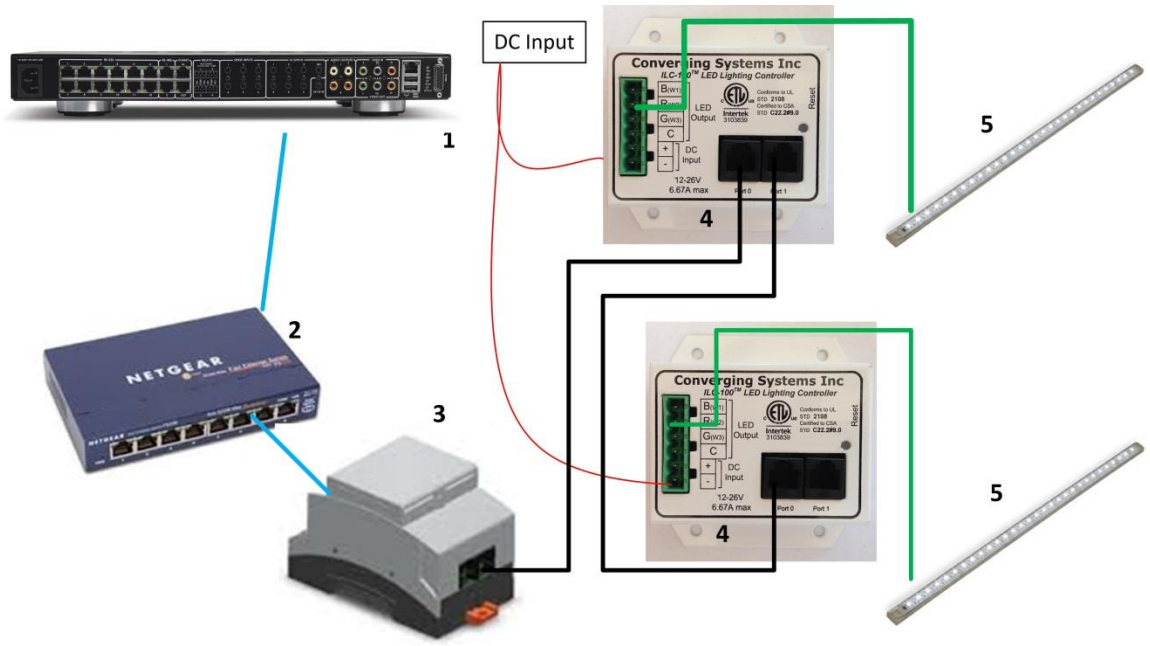


Figure A1

Wiring/Configuration Notes:

1. Maximum length of CS-Bus cabling from e-Node to the last ILC-xx0/IMC-xx0 using CAT5e or better cabling (and obeying the 1-1 pin-out requirements for the RJ-25-RJ25 cable) = 4000 feet
2. Maximum number of ILC-xx0/IMC-xx0 controllers and Converging Systems' keypads (if provided) that can exist on a single network connected to a single e-Node device = 254
3. Maximum number of e-Nodes that can exist on an NICE system = 254

BILL OF MATERIALS (for IP control)

Table 12

#	Device	Manufacturer	Part Number	Protocol	Connector Type	Notes
1	NICE Host Processor (gSC-n or similar)	NICE Home Systems	gSC-n or similar	Ethernet/Serial/IR	various	
2	Network Switch	Various	Various	Ethernet	RJ-45	
3	e-Node	Converging Systems	e-Node	Ethernet	RJ-45 (for Ethernet) RJ-25 for local bus	

4	Lighting Controller (or Motor Controller)	Converging Systems	ILC-100 or IMC-100 or (Stewart BRIC)	CS-Bus protocol	RJ-25 for CS-Bus communication	Must terminate beginning and end of bus with 120-ohm resister on pins 3/4
5	Flexible Linear Lighting (FLLA) RGB or RGBW luminaries	Converging Systems	FLLA-RGB-xxx FLLA-RGBW-xxx		3-color 4 pin 4-color 5 pin 1-color 4 pin	

WIRING DIAGRAM (for RS-232 serial connection)

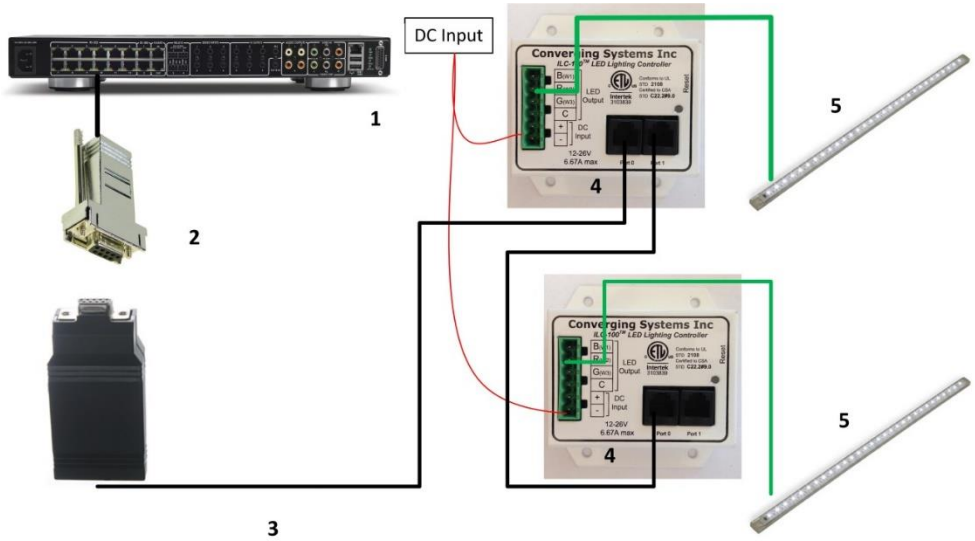


Figure A2

Wiring/Configuration Notes:

1. Maximum length of CS-Bus cabling from e-Node to the last ILC-x00/IMC-xxx using CAT5e or better cabling (and obeying the 1-1 pin-out requirements for the RJ-25-RJ25 cable) = 4000 feet
2. Maximum number of ILC-ss0/IMC-xxx controllers and Converging Systems' keypads (if provided) that can exist on a single network connected to a single e-Node device = 254
3. Maximum number of e-Nodes that can exist on an NICE system = 254

BILL OF MATERIALS (for RS-232c connection)

Table 13

#	Device	Manufacturer	Part Number	Protocol	Connector Type	Notes
1	NICE Host Processor (gSC-n or similar)	NICE	gSC-n or similar	Ethernet/Serial/IR	various	

<table>
<tr><td>2</td><td>RJ-45 to DB-9 dongle</td><td>NICE</td><td>RJ-45 to DB-9 straight dongle (CB-307 Male)</td><td>RS-232c</td><td>Pinouts<table><tr><th>RJ45</th><th>DB9</th></tr><tr><td>1</td><td>9</td></tr><tr><td>2</td><td>1</td></tr><tr><td>3</td><td>4</td></tr><tr><td>4</td><td>5</td></tr><tr><td>5</td><td>2</td></tr><tr><td>6</td><td>3</td></tr><tr><td>7</td><td>8</td></tr><tr><td>8</td><td>7</td></tr></table></td><td></td></tr>
<tr><td>3</td><td>IBT-100</td><td>Converging Systems</td><td>IBT-100</td><td>RS-232c</td><td>DB-9 (for Serial)
RJ-25 for local bus</td><td></td></tr>
<tr><td>4</td><td>Lighting Controller (or Motor Controller)</td><td>Converging Systems</td><td>ILC-100 or IMC-100 or (Stewart BRIC)</td><td>CS-Bus protocol</td><td>RJ-25 for CS-Bus communication</td><td>Must terminate beginning and end of bus with 120 ohm terminating resister on pins 3/4</td></tr>
<tr><td>5</td><td>Flexible Linear Lighting (FLLA) RGB or RGBW luminaries</td><td>Converging Systems</td><td>FLLA-RGB-xxx
FLLA-RGBW-xxx</td><td></td><td>3-color 4 pin
4-color 5 pin
1-color 4 pin</td><td></td></tr>
</table>

COMPONENT HARDWARE SETUP

Min requirements for this operation

- Computer running Windows XP or later OS, preferably with a wired Ethernet connection to a local router using CAT5 type cabling
- Converging Systems E-Node Ethernet adapter connected using CAT5 cabling to the above router.
- Download of the latest version of e-Node Pilot application, unzipped and operating on your computer platform
- Powered up and connected ILC-x00/IMC-xx0 controller using straight thru (1-1) wiring using a 6-pin RJ-connector (**Do not use 568A or 568B wiring and simply chop off the browns because this does not preserve twisted pairs on pins 1 / 2, 3 /4 , and 5/ 6 which is required**).

Recommended RJ-25 6P6C connections 6 wires		
e-Node Side	**ILC-x00 side**	**Color of wire**
Pin 1	**Pin 1**	**blue**
Pin 2	**Pin 2**	**Blue/white**
Pin 3	**Pin 3**	**Orange**
Pin 4	**Pin 4**	**Orange/white**
Pin 5	**Pin 5**	**Green**
Pin 6	**Pin 6**	**Green/white**

Suboptimal RJ-11 4P4C connection 4 wires		
e-Node Side	**ILC-x00 side**	**Color of wire**
Pin 1	**Pin 1**	**Orange**
Pin 2	**Pin 2**	**Blue**
Pin 3	**Pin 3**	**Blue/white**
Pin 4	**Pin 4**	**Orange/white**

Note: For the purposes of commissioning if *you do not have* 6P6C RJ-25 connectors, you can use standard 4-pin RJ11 connectors, but follow the wiring directions above preserving twisted pairs on Pin 2/3 and Pins 1 /4. **This cable will not work for keypad communication or IBT-100 communication**.

NOTE: The CS-BUS uses standard RJ-25 (RJ-11 6P6C) connectors available at Home Depot or Amazon and all electrical distributors). **Failure to follow the CS-BUS wiring standard will void your warranty**. If you return a unit to Converging Systems with its communication chip destroyed this is a telltale sign that you used Telephone cablina. **REPEAT--DO NOT USE TELEPHONY CABLE**.

Step	Setting
HW-1	Connect each LED lighting controller (and/or Motor controller) sequentially using Port **1** of the previous device to Port **0** of the next sequential device. Use **CS-BUS Color Standard** for your wiring.
HW-2	Connect an available CS-BUS port on the first or last LED Lighting or Motor Controller to an available CS-BUS port on the e-Node or the single CS-BUS port on the IBT-100. Power on all units.

Note: The CS-BUS by design is a modified IEEE-485 bus which requires termination on the beginning and the end of the CS-Bus. Please be advised that in most cases, termination is not required but if you do experience communication issues, it would be wise to turn on termination (in software using the Pilot software) on the first unit of the chain. If the e-Node or the IBT-100 is used as the last item in the chain, those units have built-in termination. It is important, however, not to turn on any other termination features on any other unit.

COMPONENT SOFTWARE SETUP (using e-Node and e-Node Pilot app)

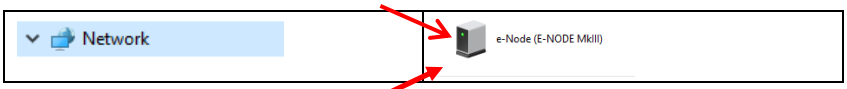
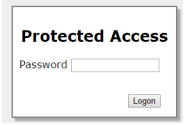
Please follow the below steps under "**e-Node Programming**" when using the e-Node for Ethernet communication and to set-up specific loads (lighting or motor) with unique, non-zero, **Z**one/**G**roup/**N**ode or **Z/G/N** addresses.

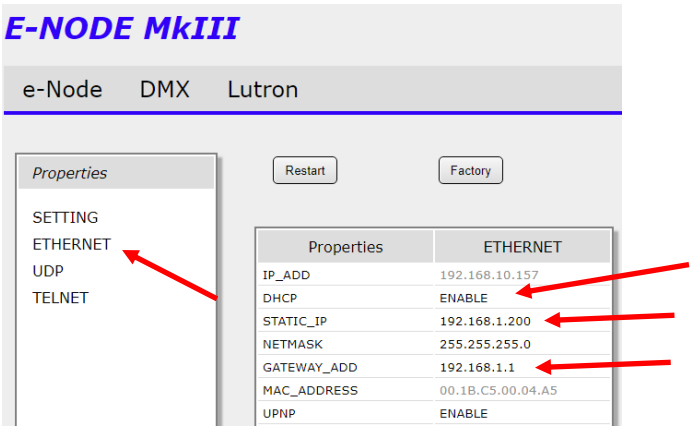
Note: there are currently two methods of performing this process

-***e-Node/web page commissioning***. Programming/commissioning using the built-in web server application of the e-Node. (The documentation below describes this new method.)

- **e-Node Pilot (PC) application commissioning**. Programming/commissioning using the download Pilot (PC) application. If you wish to perform this process using Pilot, refer to the separate "e-Node Quick Start Guide."

e-Node Programming (new uPnP Discovery Mechanism)

Step	Setting	Choices
EN-1	**e-Node IP Address** setting Set up the e-node with an appropriate Static or Dynamic IP address.	Static or Dynamic Addressing (**use Chrome as your recommended browser here**). -Use a Windows computer and open File Explorer and search for the Network tab to expand to see available uPnP* devices. Any connected e-Nodes should appear.  - Double click on the e-Node icon to expose its webpage (depending upon the model a different UI may appear).  -Click on the triple dash menu icon and you **may** be asked for a **Password**. Unless the Password has been changed or blanked out, enter **Admin** and select **Logon**  *__Note on uPnP__. You may have to turn on you PC's Discovery service (temporarily if it is turned off) to load the uPnP service within Windows to enable this type of Discovery

		-Select the **Ethernet** Tab to expose this menu where DHCP/Static addressing can be set (by default DHCP is Enabled).  -It is recommended that you set a STATIC IP address for the e-Node when using this gateway with any third-party automation system. Here, enter a static IP addresses under **STATIC_IP**. Then, enter gateway IP address under **GATEWAY_ADD**. Next, select **DHCP** DISABLED and hit **Restart** to reboot the e-Node to establish the new parameters.
EN-2	**TELNET Port** (transmit and receive)	Depending upon your network requirements and you own personal preferences, authentication can be turned on or off. As a standard, the e-Node uses Telnet / Port 23 communication (with or without authentication). ***For use with 3rd party automation systems, we recommend that you set Telnet to be enabled and turn on authentication (LOGIN-ENABLE)***. -Verify within the **Telnet** tab, that **Telnet Server** is set to Enable. -If you to use authentication, verify that LOGIN is set to Enable. Review the pre-programmed Telnet usernames/password listed under User 1~4. Select one pair (Username/Password) to enter into your automation system. If you desire to change these defaults, make the changes within this page. **Note**: You will notice that you have the capability of having 4 separate/concurrent Telnet sessions. Either accept the provided Username and Passwords or enter your own. The use this information later on when setting up your automation system's connections to the e-Node.

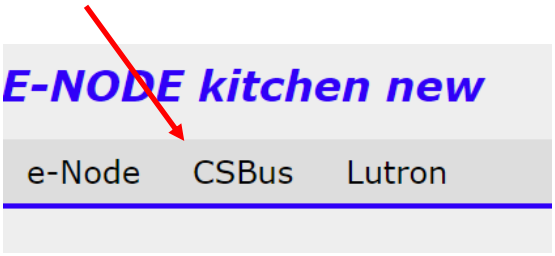
IBT-100 Programming

Auto-Discovery (Discover Devices) is not available using the IBT-100 because there is not an available XML file generated that can be used for this purpose. If you wish to use the IBT-100 within your installation, it will be necessary to perform manual Device data entry as per Appendix 4 of this document.

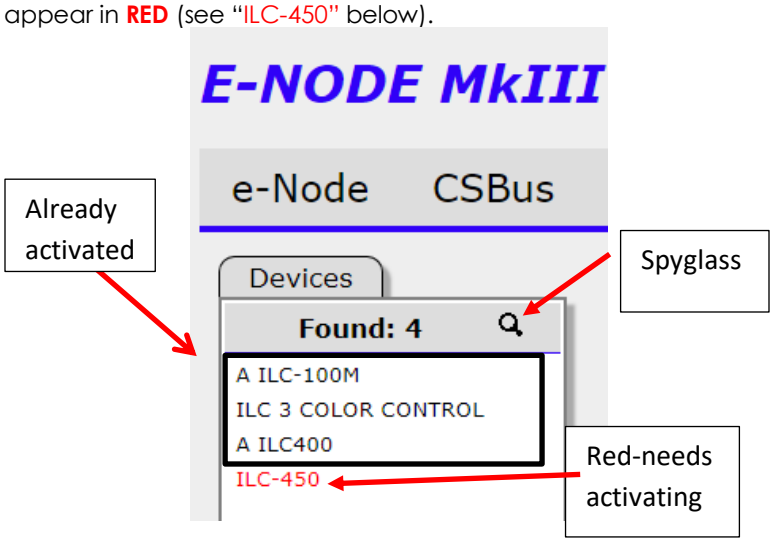
All of the communication parameters to support the IBT-100 are built into the NICE LUA driver and therefore no special programming is required of the IBT-100 serial adapter. However, certain features of the ILC-100/ILC-400 with respect to **NOTIFY** (which permits automatic signaling of color status upon color state changes) described above will need to be programmed using the e-Node. But in this case, after the specific lighting controllers are programmed, the e-Node will no longer be required for NICE to Converging Systems communication using the IBT-100.

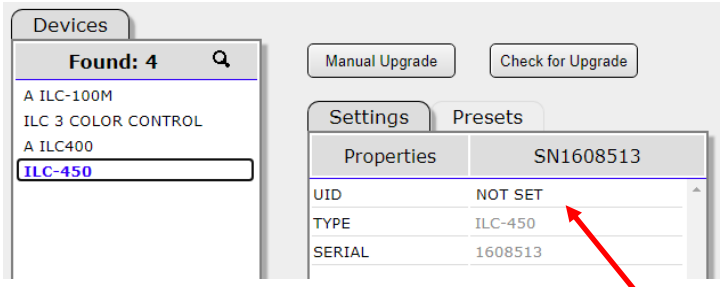
RS-232C Interfacing Note: If you plan on simply using the IBT-100 for serial communication and desire to have multiple lighting loads (more than one ILC-100) with a unique **Z**one/**G**roup/**N**ode address you must set up your system using the e-Node as specified above as well as the particular lighting load as specified below. However, if you do not care about bi-directional feedback or support of multiple controllers' address, no further set-up is required. However, this is not recommended.

ILC-xx0/IMC-xxx Device Discovery ("Commissioning")

Step	Setting	Choices
DV-1	ILC-xx0/IMC-xx0 Discovery (UID setting)	Background. From the factory the **ILC-x00/IMC-xx0 controllers generally do not** have an assigned UID (unique ID) address from the factory. -Lighting controllers come equipped with a factory default address of **Zone**=2, **Group**=1, and **Node**=undefined or a 0. -Motor controllers come equipped with a factory default address of **Zone**=1, **Group**=1, and **Node**=undefined or a 0 If you set up your automation system to communicate with a controller with an address of **2.1.0** (or 1.1.0 for motors), the controller will react but it will **not** provide feedback data which is required for automatic slider and toggle updates within automation systems. ***Therefore, it is advisable to set up a non-zero address for each ILC-x00/IMC-xx0 controller that is connected to either an e-Node (or IBT-100 using an e-Node to set it).*** The directions below indicate how to perform this operation. (See **Step 2b** below as well as **Appendix 2** for more information on **Z**one/**G**roup/**N**ode addressing.) Process. - Make sure all lighting and motor controls are properly connected to the e-Node and the e-Node's webpage described in step EN-2 is opened. -Select the **CSBus** tab within the e-Node webpage. 

-Select the **spyglass** and wait while the system auto-discovers all controllers. Any controllers previously discovered will appear in **BLACK** do not have to be activated again (see those within the black box below), while those which need to be activated will appear in **RED** (see "ILC-450" below).



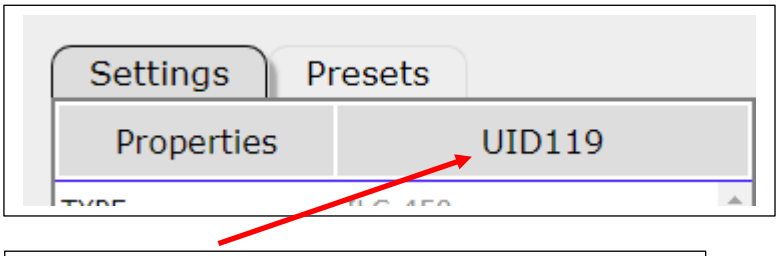


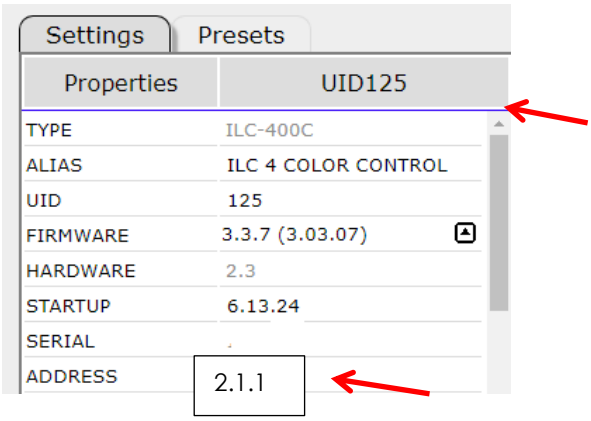
-You must ACTIVATE each controller displayed in RED in order to enable that device within your systems. To do so, hover over the device name in RED (i.e., ILC-450) with your mouse and select the Settings tab on the right to expose the UID entry area. Next to the **UID** window on the right, enter a unique (unused) UID number/address (good to start with 1 and work upwards) **but never use a duplicate number of another IMC-xxx or ILC-xxx controller anywhere within your installation).**

-After entering the UNUSED UID number, select 

on your keyboard to program. The display will change from **NOT SET** to a valid **UIDxxx** number (which will remain intact for that device in the future indicating that it has been **ACTIVATED**).

-Hit the **spyglass** icon once again to rediscover and verify the setting.

		 SNxxxxxx disappears and programmed UID### appears here. **Note**: In the unlikely case that you controller does not appear under the **Devices** window after selecting the spyglass, this may be related to the following -improper wiring -controllers not turned on -controllers may be older models of ILC-xx0 controllers or IMC-100 controller that did not originally support this function. In this case, follow the directions within the separate document "CS-Bus Controllers…Models: IMC/SMC-100x/IRC200" within its Section 5. **NOTE**: If you by chance enter duplicate UIDs for two controllers, the system will fail to work. In this case since you may not know which unit had the correct (no duplicated UID number assigned to it) and which unit had the duplicated UID number assigned to it, you must **reset both units** according to documentation found for the respective controller on the Converging Systems website and then assign unique UIDs to each one again (i.e., "Unique" IDs).
DV-2	ILC-xx0/IMC-xx0 Addressing (setting a non-zero ZGN address for all devices)	Background. Third party automation systems communicate with Converging Systems' controller through a sophisticated **Z**one/**G**roup/**N**ode addressing scheme (**ZGN**). From the factory the **ILC-x00/IMC-xx0 controllers generally do not** have a non-zero assigned **Z**one/**G**roup/**N**ode address. -Lighting controllers come equipped with a factory default address of **Zone**=2, **Group**=1, and **Node**=undefined or a 0. -Motor controllers come equipped with a factory default address of **Zone**=1, **Group**=1, and **Node**=undefined or a 0 If you set up your automation system to communicate with a controller with an address of **2.1.0** (or 1.1.0) the controller will react but it will **not** provide feedback data which is required for automatic slider and toggle updates within automation systems. ***Therefore, it is advisable to set up a non-zero address for each ILC-x00/IMC-xx0 controller that is connected to either an e-Node (or***

		IBT-100 using an e-Node to set this value.) The directions below indicated how to perform this operation. (See **Appendix 2** for more information on **Z**one/**G**roup/**N**ode addressing.) Process. - Within the e-Node webpage **CSBus** tab, select the first controller that you wish to address. Next to the **Address field**, type in that address separately by periods. For lighting controllers, we recommend the first address you use is: **2.1.1**  -Continue through all Discovered controller by entering a unique address in to each controllers' **Address** field. **Note**: Please refer to **Appendix 2** for how subgroups, larger groups, and entire floors, or buildings can be programmed innovatively when groups of controllers (rather than individual controller) need to be controlled within an automation system—this will save you many hours of customization downstream.
DV-3	Notify Mode	**Background**. Should you be implementing Color and Dimmer sliders, toggle buttons with status indicators or similar indicators within your project, automation systems need to receive bi-directional feedback from our controllers to update their UI. In order to activate the transmission of this bi-directional (Change of Value) information, the **NOTIFY*** feature is the enabled. You must turn on NOTIFY (as described below) for each controller where you want such data reported. -By default, from the factory, for lighting controllers, the **NOTIFY** function is set to **OFF** to reduce the amount of bus traffic. -By default, from the factory, motor controllers provide feedback which cannot be altered by the installer. For lighting controllers, it is recommended that **NOTIFY** should generally be set to **BOTH** (except as described below). The types of NOTIFY settings are described below:

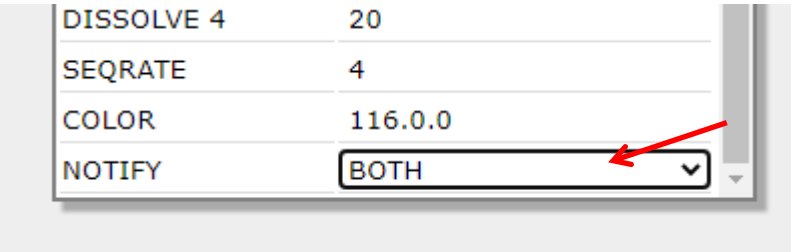
Type of Feedback Data	NOTIFY Setting
HSB color data	NOTIFY=COLOR
RGB color data	NOTIFY=VALUE
HSB and RGB color data	NOTIFY=BOTH*

*__Note__: this feature is newly added in V3.14 of ILC-100 firmware and is standard on all other controllers. In some very large installations, if is recommended that in order to reduce bus traffic, **NOTIFY** might be turned to **OFF** on controllers where feedback is not required. And even though the choice of **VALUE** (RGB or RGBW or W**)** and **COLOR** (**HSB**) and **BOTH** are all available, if you are populating buttons or sliders which need to respond to particular feedback data (W, RGB, or RGBW for instance) you must at minimum set NOTIFY to **VALUE**. And if you are expecting feedback of **Hue/Saturation** or **Brightness** values, then you need in this expanded case to also have **COLOR** selected. So, in this expanded feedback case, NOTIFY should be really set to **BOTH**.

In summary, BOTH is generally the recommended setting.

Process. Within the e-Node webpage **CSBus** tab, select the first controller that you wish to update. Next to the **NOTIFY** field, select the desired setting. For lighting controllers, we recommend:

BOTH



-Continue through all Discovered controller and update the **NOTIFY** field as appropriate.

Note: Prior to V 3.15 of the ILC-100 firmware, it is necessary to reboot the ILC-100 for this new setting to become active after it is changed. For versions 3.15 or later, simply changing this value here is sufficient.

Legacy Firmware Note: Earlier version of Converging Systems' color controllers did not support the **NOTIFY** function. In those cases, it will be necessary to either update those controllers or accept having no bi-direction control from NICE. Contact Converging Systems for more information.

Appendix 2

Background on Addressing

Topic	Reference
Background on ZGN Addressing	Section 1
Background on Bi-Directional Feedback	Section 2
How to Setup Control for a Group Load	Section 3

This information is only relevant for when you ***start*** adding buttons and sliders within the UI section of your NICE project. All Converging Systems' devices (loads or controllers as opposed to communication devices) that are connected to a communication device (e-Node or IBT-100) will be addressed using a unique **Z**one/**G**roup/**N**ode addressing scheme (**Z**/**G**/**N**). Those addresses are referred to within NICE Configurator as **Zone, Group and Node Addresses.**

1. **Background on ZGN Addresses**: The largest group is referred to as the **Zone**, which might be associated with a floor of a building. The next smaller group is referred to as the **Group**, which might be associated with a room on that floor of a building. Finally, the smallest entity is referred to as the **Node**, or the particular unit in that Room or Group, and within that Floor of Zone. From the factory, all lighting devices have a default address of **Zone**=2, **Group**=1, **Node**=0 where "0" refers to an undefined unit.

   Range of Z/G/N Addresses: Enter a number between 1 and 254 for **Z**one numbers, **G**roup numbers, and **N**ode numbers.

   Please note -- no two controllers should be assigned the same Z/G/N address.

2. **Background on Bi-Directional Feedback**: Once a load device (CS-Bus controllers) is programmed using the e-Node Pilot application or the E-Node web server tool to a non-zero value, then **AND ONLY THEN** can those devices can be queried or monitored for state data (color or motor position) which is quite useful in auto-updating sliders and numerical readouts.

   The figure below describes this hierarchy.

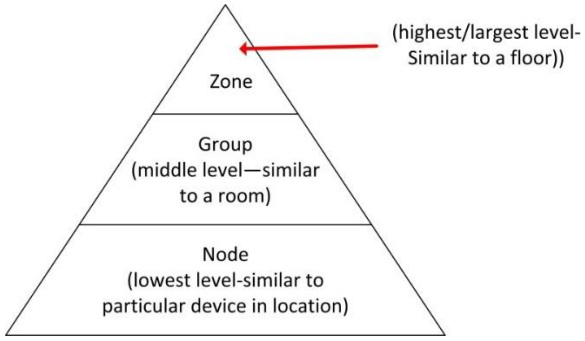


Figure A2-1

YOU MUST HAVE PRE-ASSIGNED Z/G/N ADDRESSES TO ALL LOADS BEFORE PROCEEDING WITH NICE PROGRAMMING. See the Converging Systems' documentation on the e-Node Pilot application for more information here.

At this point after you assigned **Z/G/N** address to all loads (ILC-xx0 or IMC-xx0 controllers) it would be useful to write down a "map" of all interconnected loads and their re-assigned **Z/G/N Addresses** for use when programming within NICE Configurator.

Example: If you have a device with a **Z/G/N** address of **2.1.1**, then the NICE system can monitor that device to determine its current lighting status. If you choose to enter a wildcard address of a 2.1.**0** (that is a broadcast **to** all units with **Z/G/N** addresses between **2.1.1** and **2.1.254**), only the unique color settings available from the device with an address of **2.1.1** or the first **Z/G/N** unit in the series will be queried. See **Appendix 5** for more information.

Example: If you have a device with a **Z/G/N** address of **2.1.1**, then the NICE system can poll that device to determine its current lighting status. If you choose to enter a wildcard address of a 2.1.**0** (that is a broadcast to all units with Z/G/N addresses between 2.1.1. and **2.1.254**), only the unique color settings available from the device with an address of **2.1.1** (even though there may many other units with other addresses starting with 2.1.xx) will be queried. This simplifies programming of banks of controllers which have been formed into a larger group.

Specifically, if you had more than one ILC-xx0/IMC-xx0 controller, you could give them (through the e-Node Pilot application) addresses as follows:

Table A2-2

ILC unit	Zone/Group/Node Address
First Unit	2.1.**1**
2nd unit	2.1.**2**
nth unit	2.1.**3** or some other number up to **254**

3. **How to set up group control of loads using sliders with feedback available to sliders**.

Addressing Background CS-Bus controllers can be addressed with a unique **Z**one/**G**roup/Node (ZGN) address. Up to 254 entries can be used for each field. The first field is the **Zone** (or largest range), the middle field is the **Group**, and the last field is the **Node**. No two loads can share the same **Z/G/N** address. As an example, if you will be populating a pair of two controllers within each of two rooms on two floors of a building here would be the suggested addressing that could be used.

Table A2-1

	Floor One	**Floor Two**
Room 1	2.1.1 for first controller in room. 2.1.2 for second controller in this room	
Room 2	2.2.1 for first controller in room. 2.2.2 for second controller in this room	
Room 3		3.1.1 for first controller in room. 3.1.2 for second controller in this room
Room 4		3.2.1 for first controller in room. 3.2.2 for second controller in this room

Group Addressing. In certain cases, it is desirable is simply send a wildcard address for a group of controllers to all respond in unison rather than programming each individually to respond through macros. There are two problems with macros in general. One is that often they are executed serially which means that if you had two hundred loads referenced within a macro, the timing of the execution of the last command sent out might be delayed from the first command sent out. In this case, not all LEDs would turn on or OFF at the same time, potentially. The second issue involves the actual programming time required to program scores or even hundreds of commands for a simple ALL OFF button.

Within the CS-Bus software protocol is the concept of utilizing a "0" within any address field as a surrogate for defined numbers ranging from 1 to 254 within that same field. Thus, if you issued a command of #2.1.0.LED=ON:<cr> , all units with addresses of 2.1.1 to 2.1.254 would immediately respond. Please see the table below for an example of how various wildcards could be used.

Table A2-2

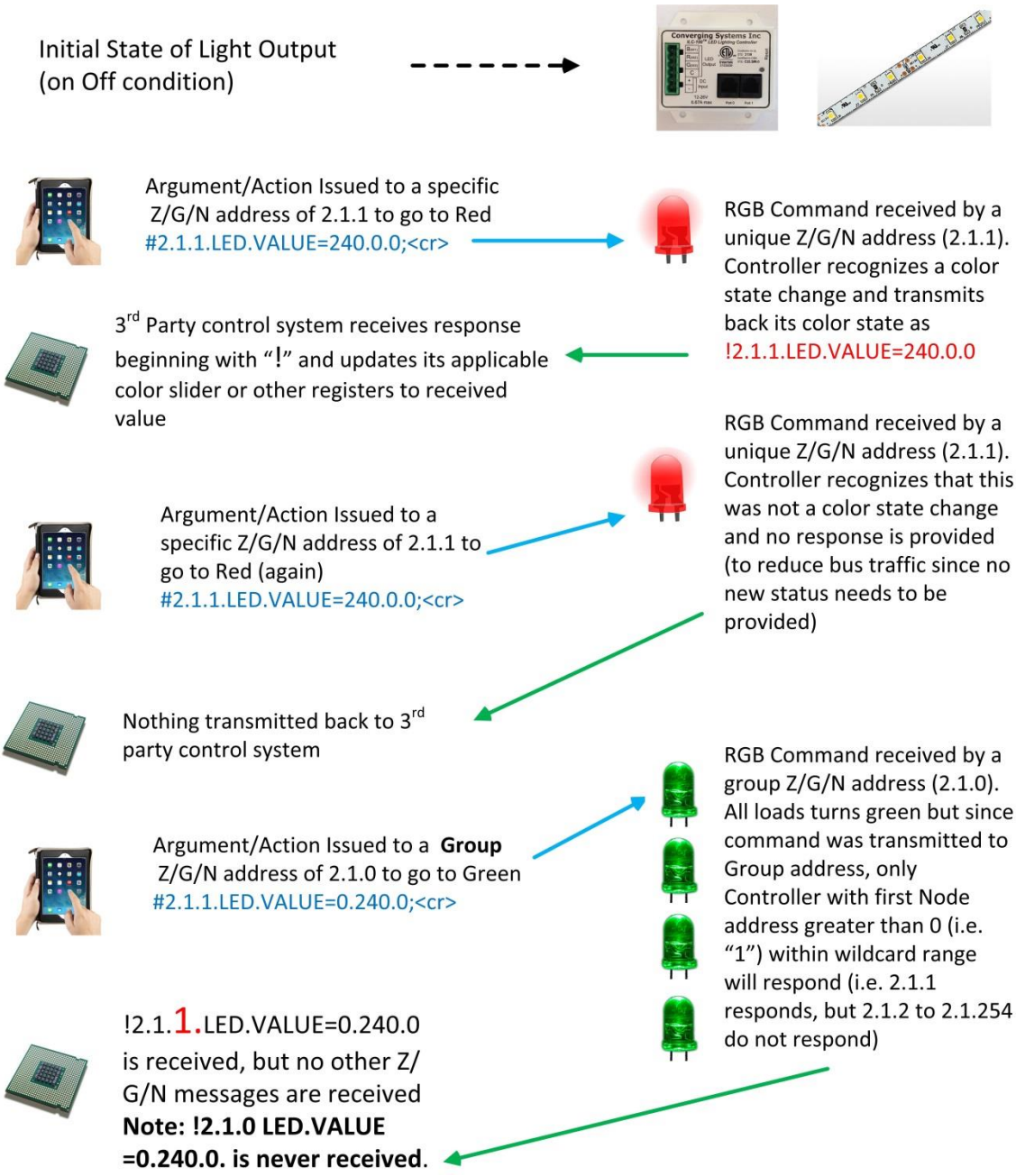
Specific controller address	**Specific command that will trigger targeted controller**
2.1.1	2.1.0 or 2.0.0 or 0.0.0
2.1.2	2.1.0 or 2.0.0 or 0.0.0
2.1.3	2.1.0 or 2.0.0 or 0.0.0
2.2.1	2.2.0 or 2.0.0 or 0.0.0
2.2.2	2.2.0 or 2.0.0 or 0.0.0
2.2.254	2.2.0 or 2.0.0 or 0.0.0
5.254.4	5.254.0 or 5.0.0 or 0.0.0

NOTIFY Command Background Converging Systems has a **NOTIFY** function which automatically provides color state feedback (from the targeted controller) provided a unique **Z**one/**G**roup/**N**ode (**Z/G/N**) address is provided with an action/argument payload to that specific controller. Specifically, if a command to invoke a color change is directed to a controller that has a **Z/G/N** address of 2.1.1, that

specific controller with that address will respond back to the automation system as to its specific color state if and only if there is a color state change impacted on that specific controller.

In some cases, as has been discussed above, there might be a requirement to send a group command or all hail command to more than one controller. In this case, the group command would be directed not to a single controller or load but to a series of controllers. To reduce bus traffic when a series of controllers is given the same command, ***only the first controller whose node number is 1 greater than the wildcard command of "0" will respond*** (which reduces bus traffic by up to 243 messages). The logic here is that if 254 controllers are all told to turn **Red**, only the surrogate for that group of controllers will respond and within the CS-Bus messaging logic that surrogate is the controller with a node of "1." So, for example, if a **#2.1.0.LED.VALUE=240.0.0:<cr>** command is transmitted to 254 controllers, they will all turn to **Red**, but only the controller with an address of **2.1.1** will respond with its new color status. In this case, a command on the bus from that surrogate controller would come back as follows: **!2.1.0.LED.VALUE=240.0.0** (the exclamation mark indicates that it is a message from CS-Bus device rather from an automation controller). Please see the diagram below for the theory of operation here.

Figure A2-2 Theory of Operation- Change of Value Feedback



Appendix 3

COLOR SPACE ISSUES

Note on Color Space.

Converging Systems recommends that only the HSB (Hue, Saturation and Brightness color space is used for it is infinitely more accurately and user friendly to control color. Although **Figure 4** below shows both HSB and RGB on the same UI, this is probably more confusing for the typical user than the simple subset of HSB (hue, saturation, brightness) controls**. Since there is no concept of dimming within the RGB color space, having RGB sliders only frustrates the user who may just want to dim an existing colored output. However, if the user is intent on having RGB sliders, we would recommend adding a separate Brightness (using Dimmer(RGB Multi-Ch-SET) Device slider to get accurate dimming.**

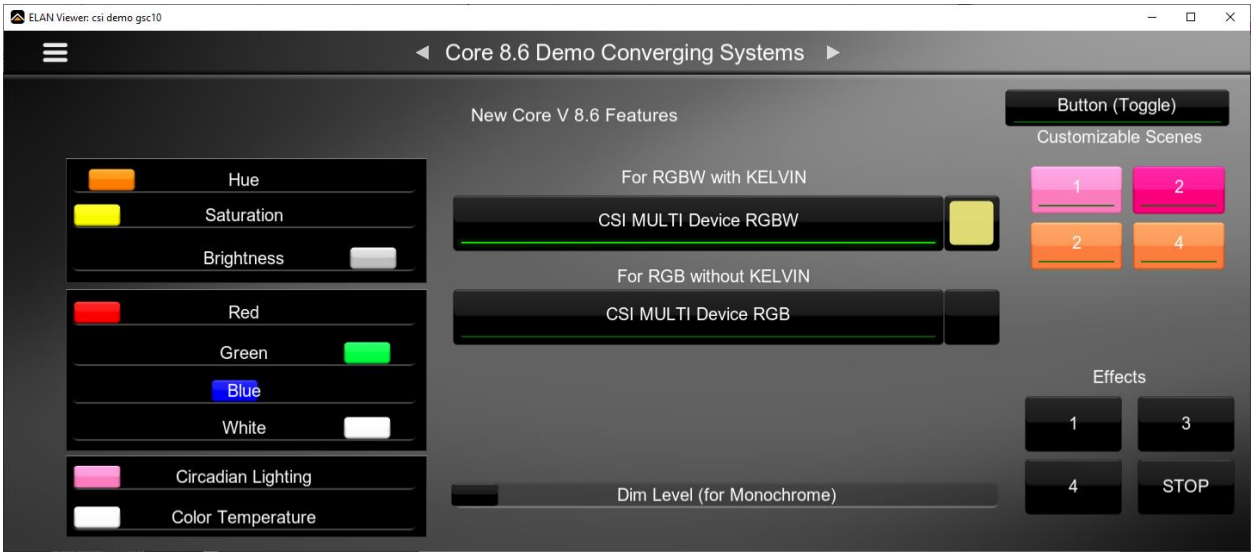


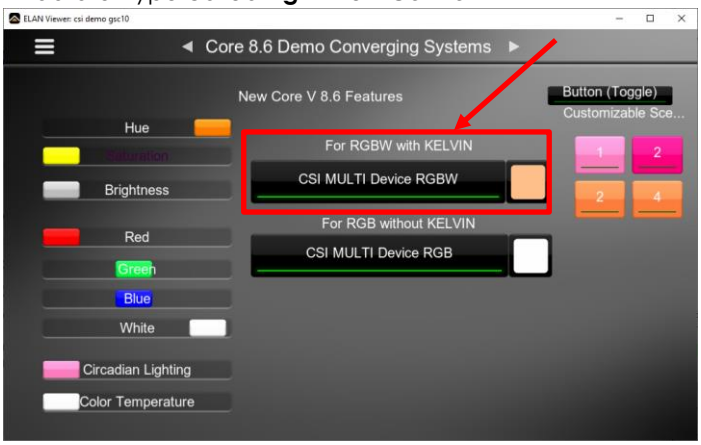
Figure A3-1

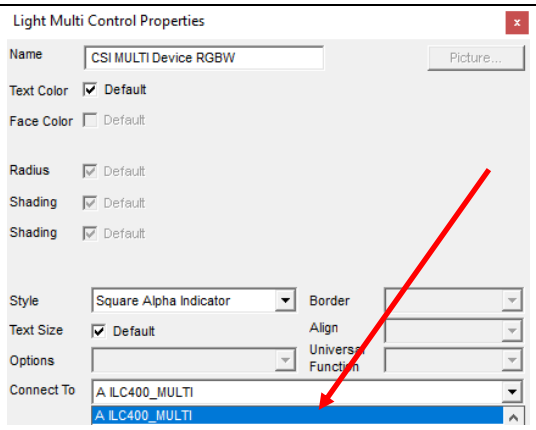
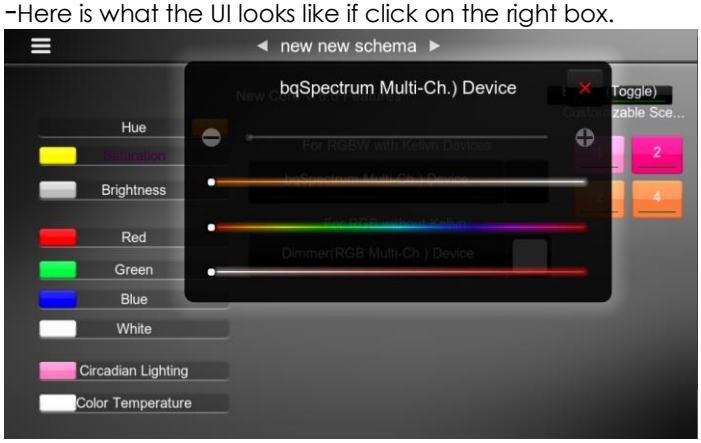
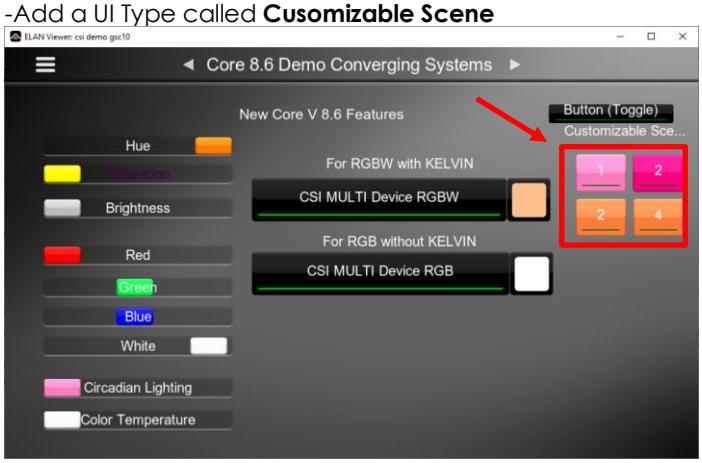
Appendix 4

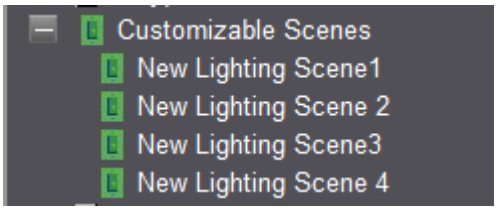
Examples/Special Interest Topics

This Appendix provides some examples of recent NICE Core features as well as some additional Special Interest Topics which might be of interest.

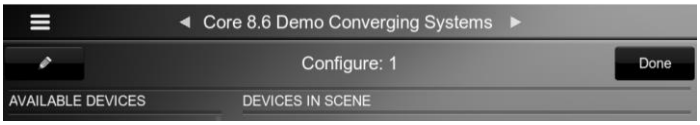
Topic or Example	Link
Examples	
Create a new Light Control UI for Core 8.6 Release	4X1
Try Out the New Customizable Scene UI	4X2
Adding a Circadian Slider	4X3
Custom Effect Examples with Dissolve Rates	4X4
Adding Effects and Customizable Scene to a HSB and CCT setup	4X6
Special Interest Topics	
UI Type Quick Reference Guide	Topic 1
Manual Discovery--Adding a Custom Lighting Device that was not Discovered automatically using DISCOVERY DEVICES & What Various Parameter Tags Mean	Topic 2

Topic	Step	Detail
EXAMPLES		
4X-1	Create use the new Light Multi Control UI for the innovative new popup **Note**: Make sure you have properly updated your Converging Systems driver (if you are upgrading an old project) to the latest version and you are using Core V 8.6 or later.	-Add a UI Type called **Light Multi Control**  -Click on new UI and seelct the "Connect To" box and link this control to an apporpritae MULTI device.

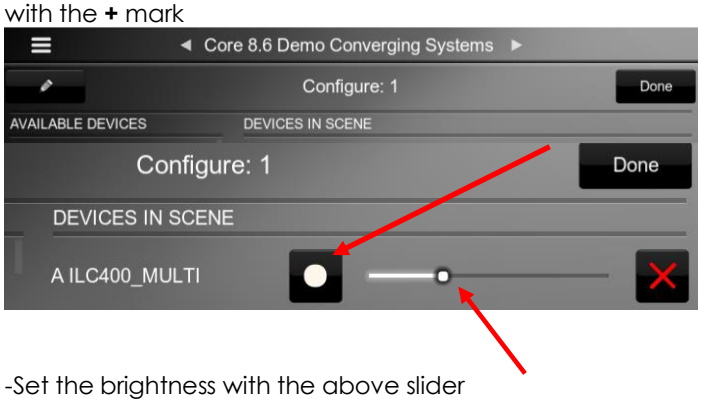
		 -Here is what the UI looks like if click on the right box.  **Note**: for Full color device with Kelvin this will appear. For RGB devices without Kelvin a similar popup will appear but without the Kelvin slider.
4X-2	Try out the new Customizable Scene UI **Note**: Make sure you have properly updated your Converging Systems driver (if you are upgrading an old project) to the latest version and you are using Core V 8.6 or later.	-Add a UI Type called **Cusomizable Scene**  -Connect that to a dealer created Customizable Scene entry (see below) for each button desired.



-Within the **Viewer**, hold the Customizable Scene button for a few seconds to reveal this new UI.

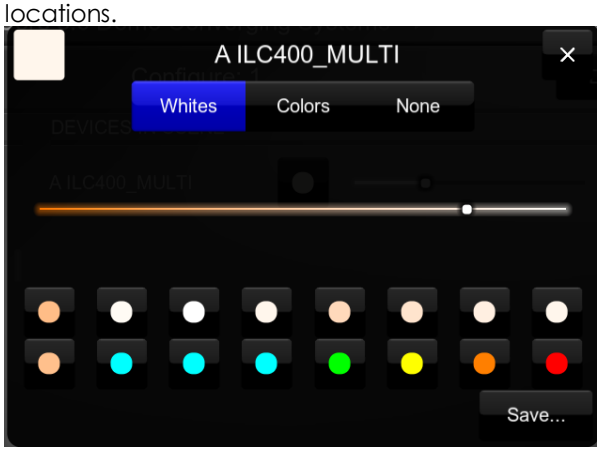


-Scroll down and find the applicable **MULTI device** (without any trailing suffixes like Red, Green, Set, etc.) and Add it with the **+** mark



-Set the brightness with the above slider
-Then hit the square with inner dot to make further adjustments.

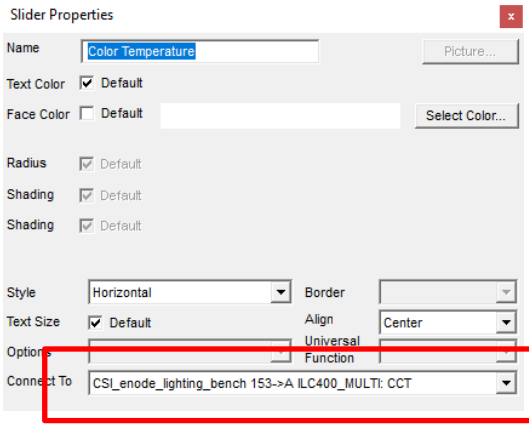
Here is what the UI olooks like if click on the right box. You can save colors, color temperatures into 16 discrete locations.



<table>
<tr>
<td>4X3</td>
<td>Adding a Circadian Slider</td>
<td>-Add a UI Type called Slider (not dimmer)

-Within Properties, set the “Connect To” to a MULTI with the appropriate suffix for the specific function you wish to control. In this case, since we want to make it control a Circadian function, select
-MULTI_CCT

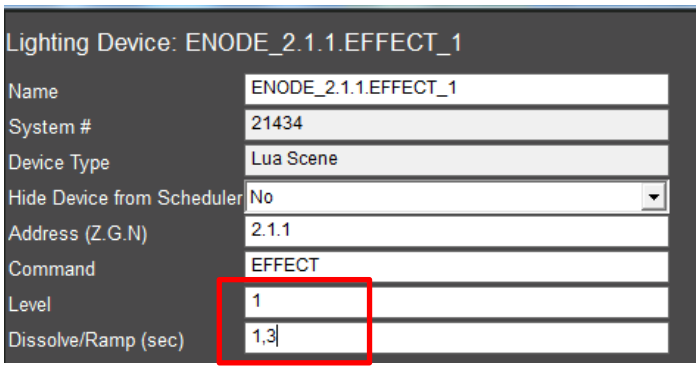
-Here is a table for other connections that you might wish to make

<table>
<tr><td>Red</td><td>_MULTI_Red</td></tr>
<tr><td>Green</td><td>_MULTI_Green</td></tr>
<tr><td>Blue</td><td>_MULTI_Blue</td></tr>
<tr><td>White</td><td>_MULTI_White</td></tr>
<tr><td>Circadian (i.e., SUN)</td><td>_MULTI_Sun</td></tr>
<tr><td colspan="2">Although these additional functions below are still available, we advise that for these items you use the new MULTI Device which automatically generate these sliders.</td></tr>
<tr><td>Hue</td><td>_MULTI_Hue</td></tr>
<tr><td>Saturationj</td><td>_MULTI_Sat</td></tr>
<tr><td>Brightness</td><td>_MULTI_Set</td></tr>
<tr><td>Dim Level</td><td>_MULTI_Dim Level</td></tr>
<tr><td>CCT (color temp or Kelvin)</td><td>_MULTI_CCT</td></tr>
</table>
</td>
</tr>
<tr>
<td>4X4</td>
<td>Custom Effects Examples with Dissolve Rate enhancements</td>
<td>In two special cases, for Effect(1) and Effect(4), an additional concatenated sub-TAG can be entered to</td>
</tr>
</table>

<table>
<tr><td></td><td></td><td>change the SeqRate from the factory default as well. The SeqRate specifies the time (after any dissolve) that the preset color is maintained before transitioning to the next color in sequence.

In the example below, for Effect(1), a Dissolve rate of 1 second is specified as well as SeqRate of 3 seconds. The format for this entry is:

<Dissolve Rate, Sequence Rate>

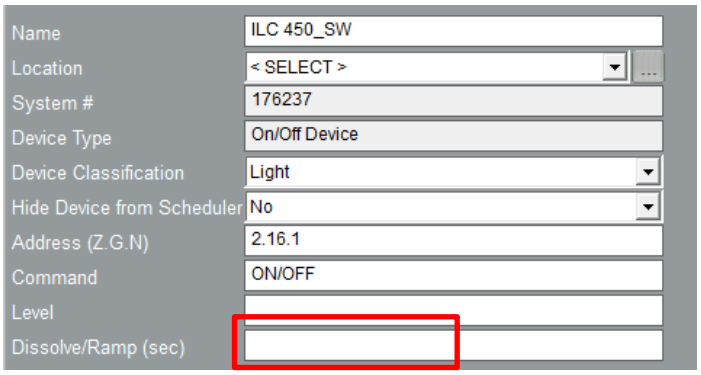
 

SUMMARY NOTE: YOU WILL NEED TO CREATE AS MANY DEVICES (of the three types available) FOR THE NUMBER OF SLIDERS OR BUTTONS REQUIRED RELATED TO A SPECFIC Z/G/N ADDRESS.</td></tr>
<tr><td>4X4</td><td>Create a Custom Page that allows two controllers with two different ZGN addresses to be controlled along with a number of special features.</td><td>If you have one ILC-400 LED controller with a Z/G/N address of 2.1.1 and a second ILC-400 LED controller with a Z/G/N address of 2.1.2 AND you wanted a Hue/Sat/Brightness/CCT set of sliders, an ON/OFF control, a Customizable Scene (Scene 1) button for each controller, as well as a Standard Button that can select Effect 1 (which cycles through Preset Colors 1 through 8 in an infinte loop with a Dissolve Time of 1 second and a Seq Rate of 3 seconds) for both controllers in unision, you would need to create this following:

<table>
<tr><th>Deisred button or slider</th><th>Device Type</th><th>Add. (Z/G/N)</th><th>Com-mand Tag entry</th><th>Level</th><th>Diss/ Ramp (sec)</th></tr>
<tr><td>HSB+ CCT Slider</td><td>MULTI Devcie</td><td>2.1.1</td><td>autopo pulate d</td><td></td><td></td></tr>
<tr><td>Cust. Scene 1</td><td colspan="5">Add Customizable Scene button and from Viewer add applicable devices created elsewhere throughout this example</td></tr>
</table>
</td></tr>
</table>

ON/Off control (with dissolve capability)	ON/OFF Toggle	2.1.1	(Set through Event Map ON— Off-	(not req'd)	See below
HSB+ CCT Slider	MULTI Devcie	2.1.2	autopopulated		
Cust. Scene 1	Add Customizable Scene button and from Viewer add applicable devices created elsewhere throughout this example				
ON/Off control (with dissolve capability)	ON/OFF Toggle	2.1.2	(Set through Event Map ON— Off-	(not req'd)	See below
Effect 1	LUA Scene	**2.1.0**	EFFECT	1	1,3

*Note the Toggle On/Off and the Discrete On/Off do not permit the addition of a dissolve rate. Therefore, you either need to set that Dissovle.2 (the dissolve rate that impacts ON and Off rates) either as a setup entry within e-Node web server setup, or create a new Lighting ON/OFF Device as shown below and insert your dissovle rate in seconds within the box shown.



Special Interest Topics

4S1	UI type Quick Reference Guide	The following information details the type of **Lighting Devices** that would need to be added in order to support the functionality of the UI pictured below. The letter references are explained below.

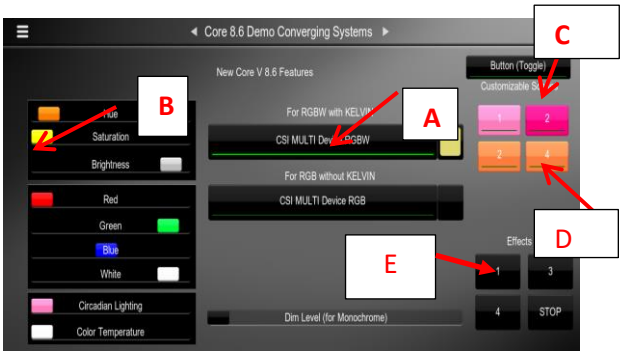


Figure 3

Currently, the available functions supported by these Device Types relevant to Converging Systems LED products are as follows:

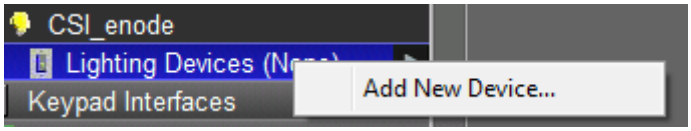
Table 14

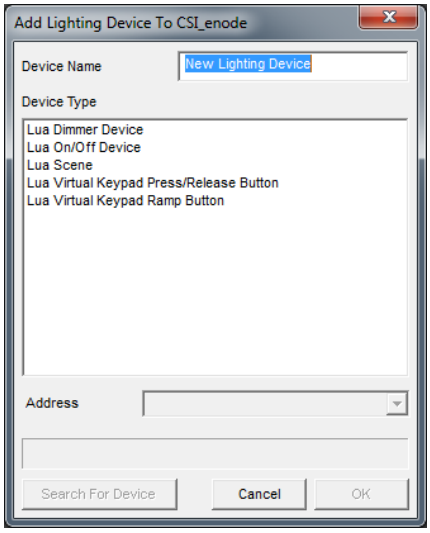
Ref. to above Figure	UI Type	NICE Lighting Device	Application
A	Light MULTI control (Hue,Sat, Brightness, CCT) for Kelvin devices And Hue,Sat. Brightness) for RGB devices	MULTI Device **Note**: Individual devices are automatically made available after a device discovery.	**Slider Control** for -Hue, -Sat -Brightness --Color Temp-erature
B	Slider Control	Lua Dimmer (RGB Multi-Ch.) Device	**Slider Control** for -R/G/B, Circadian, Control
C	On/Off button (with capability for dissolve setting)	Lua Scene Or Lua On/Off Device **Note**: A separate device must be installed for	**Button (Standard)** -On -Off

				ON/Off button set for each Z/G/N address		
		D	(Customizable) Scene button	LUA Scene **Note**: A single device must be installed for each Z/G/N address to be supported regardless of the number of scenes to be supported.	**Customizable Scene buttons** -Scene 1 to n (which allows you Pick from nearly Any control Available)	
				Event Map Buttons		
		E	Recall/Store/ Effect buttons (with capabiilty for dissolve)	LUA Scene **Note**: A single device must be installed for each Z/G/N address to be supported as well as for each discrete index references (i.e. Effect 1, Recall 1, etc.)	**Button (Standard)** -On -Off	
		E (not shown)	% Set button	Lua Dimmer Device **Note**: A single device must be installed for each Z/G/N address to be supported regardless of the numbet of	- **Button (Standard)** to pick a Particular level setting	

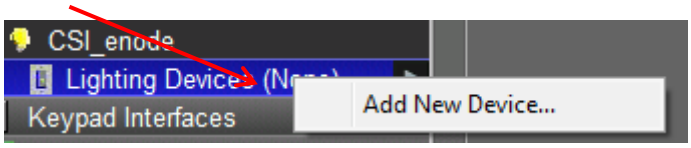
				% set buttons to be populated.		

These choices are available by right clicking on the **Lighting Devices (None)** entry or any programmed entry under the **Lighting Interface** and selecting **Add New Device**…

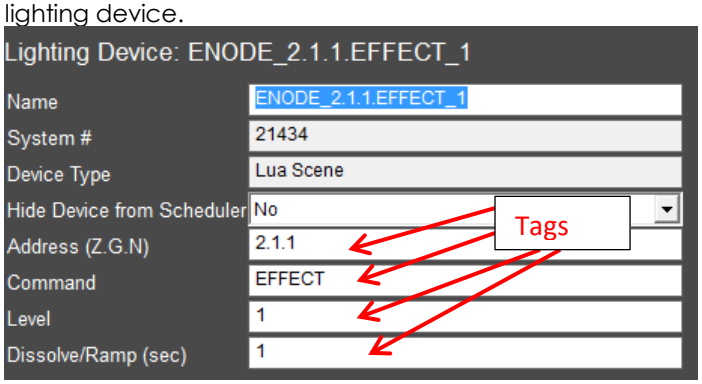




Note: Additional devices may be displayed above that either (i) may be undocumented within this current Integration Note/Driver set, or (ii) may not be functional with the current revision level of the Converging Systems' LUA driver.

4S2	Adding a Custom Lighting device that otherwise was not discovered using "Discover Devices)	- Right click on the auto-populated (generic entry) **Lighting Devices (None)** found below the **Lighting Interface** established in **Step 1c** above. A pop-up Add New Device… will appear  -Left click on the **Add New Device...** button to begin adding the applicable Lighting device to be supported. Depending upon your control needs, you will need to select a specific Device Type specified in the table above to match your requirements. Following is an example of

the data entry window that may appear for your particular lighting device.



Note: Depending upon the lighting device to be added, 2 or more data entry **Tags** will appear in addition to a field for a user-entered **Name**. Occasionally additional *Tags* may appear than are not currently required to be filled out.

Following are the **Tags** that may be required to be filled out depending upon the nature of particular user interface and the lighting device selected. Refer to the table below for required and non-required **Tags** for each lighting device.

Table 15

	TAGS			
User Interface Type	**Address (Z.G.N)**	**Command**	**Level**	**Dissolve Ramp**
Dimming Device				
Slider	Req'd	Req'd for type		
On/Off button (without dissolve feature)	Req'd	Not req'd		
Scene Device				
Recall	Req'd	RECALL	Value	Optional
Store	Req'd	STORE	Value	Optional
Effect	Req'd	EFFECT	Value	Optional
On/Off button (with dissolve feature)	Req'd	Not req'd	Not req'd	Optional

Next are the descriptions for all Tags (which may or may not need to entered depending upon the type of User Interface required).

-**Name**. This is an alias name that should be entered to easily identify the Device. Typically, a **Z/G/N** (**Z**one.**G**roup.**N**ode) reference can be used to facilitate

device identification especially when there are many devices to be programmed (see example below for more information).

-**Address Tag.** This is an addressing reference this is read by the NICE Core software and is bundled in all outgoing command strings sent to Converging Systems controllers. The address must be accurately entered or no control of a specific device will be possible. **It is critical that each number (between 0-254) is entered with Periods (not Commas) separating those numbers:**

Address (Z,G,N) **Z•G•N**

For example, for a device with the following **Z/G/N** address:

Zone	2
Group	1
Node	1

you would enter the following with NICE Configurator (exactly as shown):

Address (Z.G.N)	2.1.1

Note: The **Z.G.N** entries refer to the **Zone** number, **Group** number and **Node** number previously programmed into each CS-Bus controller (**see Appendix 1** for more information).

Command Tag. This is the type of slider for which control and feedback is desired. **The tag must be accurately entered using upper case letters spelled correctly.** Refer to the following table for the **Command Tag** information that must be entered for each Device Type to enable the operation of these types of controls.

Table 16

UI Type	Command Tag
Red Slider	**RED**
Green Slider	**GREEN**
Blue Slider	**BLUE**
White Slider (for ILC-400 controllers only)	**WHITE**
Hue Slider	**HUE**
Saturation Slider	**SAT**

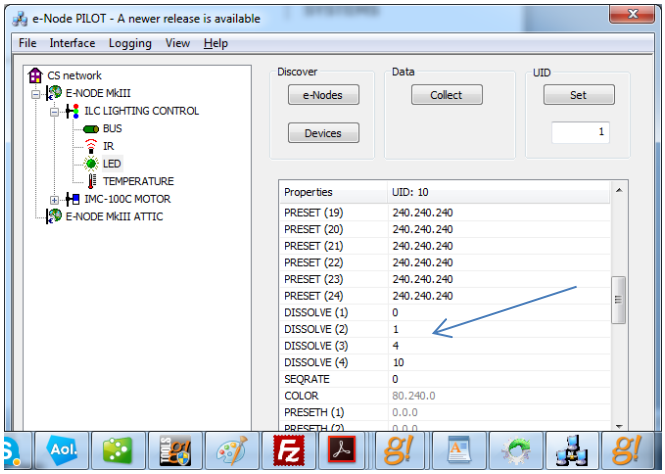
Brightness (Fade) Slider	**SET**
CCT (correlated color temperature-RGB and RGBW devices)	**CCT**
SUN (circadian rhythm ILC-400 only)	**SUN**
Standard On/Off buttons	**(No entry)**
Recall (Preset within CS-Bus controller)	**RECALL** (where n is the scene or recall number)

Level Tag. For applicable Command Tags (i.e., Recall, Store, Effect) this is the field for the numerical entry or index to be either Recalled, Stored or activated through the applicable command. **The tag must be accurately entered using numbers or no control of a specific device will be possible.** Refer to the following table for the **Level Tag** information that must be entered for those Command Tags requiring such additional information.

Table 17

Command Tag Class	Level Tag
Effect	**0,1,2,3 (see controller documentation for all supported Effects)**
Store	**1-24**
Recall	**1-24**

Dissolve Tag. For applicable Dissolve/Ramp Tags (i.e., all supported UI controls other than sliders) this is the field for the numerical entry of a **Dissolve Rate** to be entered (if desired) in seconds. If the field is not entered, the factory default for the applicable **Dissolve Rate** will be utilized instead or the **Dissolve Rate** that was last entered through a command will be utilized. **The Tag therefore is optional and if not set through the Dissolve Tag will be maintained as the value originally set from the factory.** See below where the current setting of the Dissolve Rates can be seen using e-Node Pilot software and

		a connected e-Node. Refer to the Device Driver Toolkit for more information on Dissolve Rates. 

Appendix 5

DMX Functionality (using e-Node/dmx) and the new e-Node Web Pilot Setup Application for MKIII e-Nodes

Backgrounder on DMX Lighting Devices. There are many third-party lighting devices available in the marketplace that support the DMX512 lighting standard ("standard for digital communication). The DMX 512 protocol is based on the EIA/TIA-485 standard (commonly known as Recommended Standard 485 or RS-485) which uses asynchronous, differential data transmission. This standard supports 32 devices or fixtures on one network at a distance of up to 4000 feet. DMX devices were originally utilized for theatrical interior and architectural lighting application only, but recently their adoption rate has grown in other areas where colored lighting is desired. Popular DMX 3-channel lighting fixtures utilizing Red, Green, Blue (RGB) illuminants (and 4-color derivatives utilizing Red, Green, Blue, White (RGBW) illuminants), which although practical for theatrical applications by the trained lighting designer are often impractical for general lighting and general automation adaptations because of interfacing, compatibility and basic functionality issues. Specifically, most DMX fixtures with channels dedicated to particular colors (i.e. Red, Green, Blue, etc.) lack a slider or control for dimming and through this inherent structural weakness lack the capability for hue accurate dimming without color shifts (because linear movement of color sliders cannot dim accurately). ***But that has all changed now...***

Converging Systems' e-Node/dmx. Converging Systems has developed an adaptation of its lighting/dimming technology currently available within its ILC-x00 line of LED controllers and has re-purposed that technology into a separate product known as the e-Node/dmx. Existing third-party automation and lighting control software drivers for Converging Systems' product line also enable support for the e-Node/dmx (color engine/dmx translator) controller. Unique to the e-Node/dmx is its ability to perform color adjustments within its own processor to enhance hue-accurate dimming without colors shift along with the added benefit of light level stores and recalls as well bi-directional communication. In addition, the robust color engine embedded within the e-Node/dmx offloads DMX support from the lighting or automation platform. (See the listing of commands that are supported with the e-Node/dmx device within the supported LED command section within this document or within any specific Integration Note for a third-party platform.)

NOTE: DMX cannot be split reliably by making T-connectors or Y-cables. Third-party splitter/repeaters typically use optical isolation to protect each segment from electrical faults or reflections on other branches. These can be used to increase the number of devices on one network beyond the limit of 32. However, each branch of a splitter/repeater can support only 32 devices. Converging Systems maintains that limitation of 32 devices or fixtures per e-Node/dmx.

Please follow the directions which follow to drive DMX fixtures utilizing the e-Node/dmx and a host automation or lighting controller.

DMX Channels/Compatibility and Interfacing Issues

The e-Node/dmx has been designed to adapt to a tremendous breadth of DMX interfacing scenarios. It is important to understand however, the methodology on how interfacing works. Device drivers from third-party platform manufactures are based upon two kinds of models- one is the single channel monochrome type scenario (1-channel) and the other is a multi-channel RGB or RGBW scenario (3-channel or 4-channel). For both of these types of devices, setup and programming is quite straightforward. In some cases, however, DMX fixtures may come with upwards of 7 or 15 or even 57 channels of control (that is they occupy 7 or 15 or 57 channels of the 512 possible DMX channels in a universe controlling pan, tilt, other motion and even 16-bit color addressing). No standard automation or lighting systems would have ever imagined driving 57 different parameters within a lighting fixture and that is why specialized theatrical control devices have come into existence.

Converging Systems' recognizes though that from time-to-time that some of these fully functional DMX fixtures may by necessity or convenience be desired to be interfaced with traditional lighting and automation systems. It is here that Converging Systems has become creative and has developed procedures thus enabling this type of adoption of a theatrical device (i.e., DMX fixture) into a traditional lighting or automation system.

Steps to plan out next steps for interfacing.

<table>
<tr><th>Step</th><th>Topic</th><th>Detail</th></tr>
<tr><td>1</td><td>- Document (i) all channels numbers of the DMX fixture available and (ii) their channel name (i.e. RED, GREEN, PAN, etc.). Transcribe this information on the table on the right in the first and second column.

-Determine which channels of the N-channel DMX device that you wish to actually control and which channels that you wish to bypass (and not control). Note that information on the table in the third column.

Note: Think about Red, Green Blue and White as virtual placeholders understood by the automation system but which could have varying meaning to the outside world. Specifically, if your device has a PAN mode that Pan mode could be driven by a virtual RED slider regardless of its functionality.</td><td>
<table>
<tr><th>DMX channel assignment on fixture</th><th>Channel name or functionality (i.e. Red, Green, Pan, etc.)</th><th>Control (Y) or Bypass (N)</th></tr>
<tr><td>N (base DMX address for that fixture)</td><td>(fill in)</td><td></td></tr>
<tr><td>N+1</td><td></td><td></td></tr>
<tr><td>N+2</td><td></td><td></td></tr>
<tr><td>N+3</td><td></td><td></td></tr>
<tr><td>N+4</td><td></td><td></td></tr>
<tr><td>N+5</td><td></td><td></td></tr>
<tr><td>N+6</td><td></td><td></td></tr>
<tr><td>N+7</td><td></td><td></td></tr>
<tr><td>N+8</td><td></td><td></td></tr>
<tr><td>N+9</td><td></td><td></td></tr>
<tr><td>N+10</td><td></td><td></td></tr>
<tr><td>N+11</td><td></td><td></td></tr>
<tr><td>(expand this table as</td><td></td><td></td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td></td><td></td><td>appropriate to any length)</td><td></td><td></td></tr>
<tr><td>2</td><td>Review the table created in Step 1 above,

-If you have anywhere from 2-4 channels to be supported (with variable control) and your DMX fixture has no more than 15 channels available, proceed to Step 2a

-If you have more than 5 channels to be supported (with variable control) or if you DMX fixture has more than 15 channels available regardless if you want to support more than 5 of those channels, proceed to Step 2b.</td><td colspan="3"></td></tr>
<tr><td>2a</td><td>Background: Automation and lighting systems currently support up to 4 controls within their Converging Systems drivers. The existing names for these controls are Red, Green, Blue and White. Think about these name as virtual names which could be mapped to anything (i.e. Pan, Tilt, Zoom, Move CCW, Move CW, Vibrate, etc.)

Here for environments where you wish to provide variable control for up to four controls on the DMX fixture (for any fixture than has up to 15 discrete DMX channels), map each channel to one of the following variables (V) or binary (B) options:

-RED (V)
-GREEN (V)
-BLUE (V)
-WHITE (V)
-MONO (V)
-FULL (B)
-OFF (B)

Note: From time to time additional placeholder names may be added for convenience, however, regardless of the virtual names added, any supported (variable) operational name can be used for variable control (V) (Red, Green, White, etc.) in addition to any binary operational name can be used for binary control (B) (i.e. Full ON or OFF).

-Given the above, map all channels to be controlled to the supported Variable and Binary names

-See the example to the right for more information here.</td><td colspan="3">
<table>
<tr><th>Actual DMX channel assignment on fixture</th><th>Channel mapping (from available mapping choices)</th></tr>
<tr><td>N (base DMX address for that fixture)</td><td>(choose)</td></tr>
<tr><td>N+1</td><td></td></tr>
<tr><td>N+2</td><td></td></tr>
<tr><td>N+3</td><td></td></tr>
<tr><td>N+4</td><td></td></tr>
<tr><td>N+5</td><td></td></tr>
<tr><td>N+6</td><td></td></tr>
<tr><td>N+7</td><td></td></tr>
<tr><td>N+8</td><td></td></tr>
<tr><td>N+9</td><td></td></tr>
<tr><td>N+10</td><td></td></tr>
<tr><td>N+11</td><td></td></tr>
<tr><td>(expand this table as appropriate to any length)</td><td></td></tr>
</table>

Example. Here is an example where there is a dimmer channel (low-tech channel that will not be used) on DMX Channel 1, and R, G, B, W controls on Channels 6,7,8,9 on a 10 channel DMX fixture. Other channels although available on the DMX fixture are not relevant here and will be disabled (bypassed).

<table>
<tr><th>Actual DMX channel assignment on fixture</th><th>Channel mapping (from available mapping choices)</th></tr>
<tr><td>1 (base DMX address for that fixture)</td><td>FULL (to keep brightness on full such that the R/G/B/W components</td></tr>
</table>
</td></tr>
</table>

<table>
<tr>
<td></td>
<td>-When completed refer to this programming information when programming in Step DMX-WP3 below</td>
<td>
<table>
<tr><td></td><td>can be controlled separately</td></tr>
<tr><td>2</td><td>OFF (for this is an irrelevant channel for our example)</td></tr>
<tr><td>3</td><td>OFF (for this is an irrelevant channel for our example)</td></tr>
<tr><td>4</td><td>OFF (for this is an irrelevant channel for our example)</td></tr>
<tr><td>5</td><td>OFF (for this is an irrelevant channel for our example)</td></tr>
<tr><td>6</td><td>RED</td></tr>
<tr><td>7</td><td>GREEN</td></tr>
<tr><td>8</td><td>BLUE</td></tr>
<tr><td>9</td><td>WHITE</td></tr>
<tr><td>10</td><td>OFF (for this is an irrelevant channel for our example)</td></tr>
</table>
</td>
</tr>
<tr>
<td>2b</td>
<td>In the event that you wish to control more than 4 channels with variable output on a single DMX fixture, this step provides a procedure to link together more than one virtual e-Node/dmx “fixture” to provide more than 4 channels of variable output and up to a possible128 channels of variable control.

-Follow the procedure in Step 2a for the first set of 4 DMX channels to be controlled (Variable). Then, add up to 4 additional DMX channels to be variably controlled on that same DMX fixture to a second e-Node DMX Fixture,

-Continue until all variable controls have been allocated to subsequent e-Node DMX Fixtures

-See the example to the right for more information

-When completed refer to this programming information when programming in Step DMX-WP3 below</td>
<td>Example
A DMX fixture with 25 channels (base DMX channel 1) is desired to be supported with offers the following relevant channels that are desired to be controlled

-RED (Channel 1)
-GREEN (Channel 2)
-BLUE (Channel 3)
-WHITE (Channel 4)
-PAN LEFT (Channel 21)
-TILT (Channel 22)
-ZOOM (Channel 23)

There is also a DIMMER channel available (DMX Channel 11) that only moves R/G/B/W sliders on a proportional basis and which we want to disable so that the e-Node/dmx’s Pure Mode HUE ACCURATE DIMMING color computer is utilized alternatively for dimming.

All other channels for this example are irrelevant and will be set to 0 (not controlled or bypassed). Bypassing them (setting them to 0) in this case will not cause any negative impact on the remaining channels to be controlled.

<table>
<tr><td>DMX Fixture 1</td><td>DMX Fixture 2</td></tr>
<tr><td>ALIAS Virtual DMX Fixture A-1</td><td>ALIAS Virtual DMX Fixture A-2</td></tr>
<tr><td>ADDRESS 2.1.1</td><td>ADDRESS 2.1.2</td></tr>
</table>
</td>
</tr>
</table>

		NOTIFY BOTH	**NOTIFY** BOTH
		CHANNELS 13	**CHANNELS 11**
		BASE DMX CH 1	**BASE DMX CH 14**
		ASSIGN CH 1 (1) RED	**ASSIGN CH 1 (14)** OFF
		BASE DMX CH 2 (2) GREEN	**BASE DMX CH 2 (15)** OFF
		BASE DMX CH 3 (3) BLUE	**BASE DMX CH 3 (16)** OFF
		BASE DMX CH 4 (4) WHITE	**BASE DMX CH 4 (17)** OFF
		BASE DMX CH 5 (5) OFF	**BASE DMX CH 5 (18)** OFF
		BASE DMX CH 6 (6) OFF	**BASE DMX CH 6 (19)** OFF
		BASE DMX CH 7 (7) OFF	**BASE DMX CH 7 (20)** OFF
		BASE DMX CH 8 (8) OFF	**BASE DMX CH 8 (21)** OFF
		BASE DMX CH 9 (9) OFF	**BASE DMX CH 9 (22)** OFF
		BASE DMX CH 10 (10) OFF	**BASE DMX CH 10 (23**) OFF
		BASE DMX CH 11 (11) FULL	**BASE DMX CH 11 (24)** OFF
		BASE DMX CH 12 (12) OFF	**BASE DMX CH 11 (25)** OFF
		BASE DMX CH 13 (13) OFF	

Now that you understand the breath of control available with the e-Node/dmx you are ready to proceed.

WIRING DIAGRAM (for DMX control using e-Node/dmx and Internet Protocol-IP)

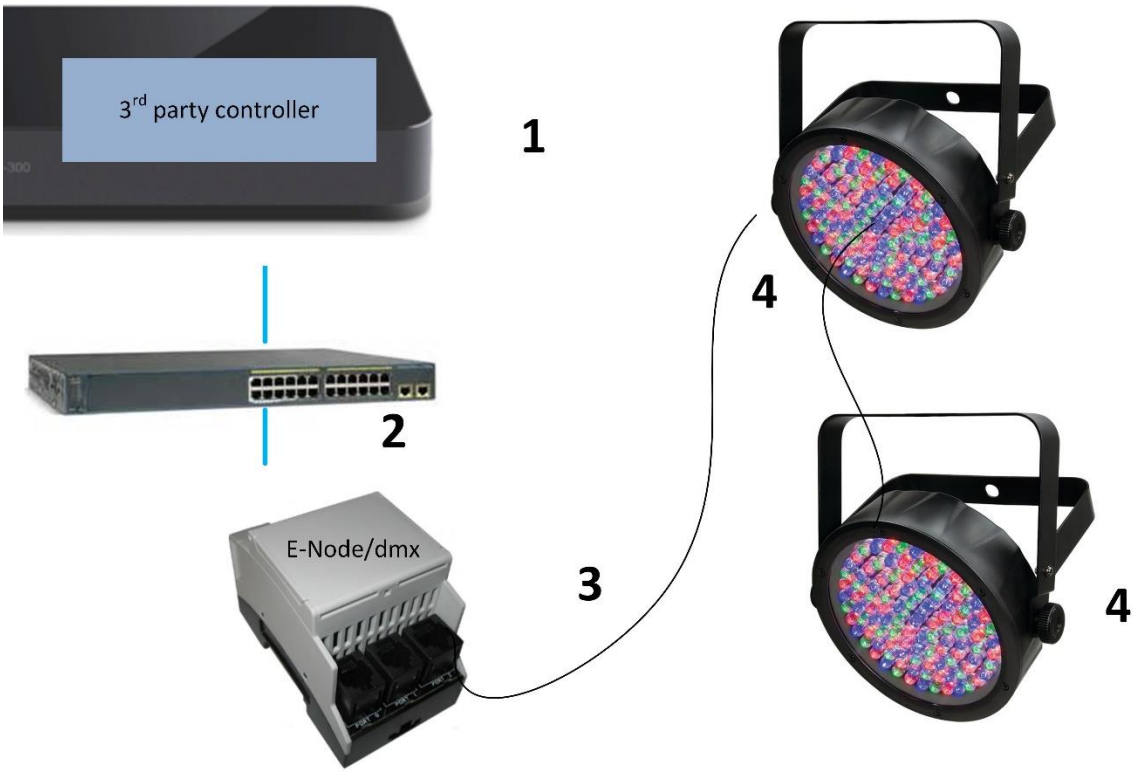


Figure 4

Wiring/Configuration Notes:

1. Maximum length of CS-Bus cabling from e-Node to the last DMX fixture using DMX cabling = 1200 meters (3,900 feet). This theoretically limit may be reduced with some fixtures. Consult individual fixture documentation for more information here
2. Maximum number of DMX fixtures connected to a single e-Node/dmx device = 32. If more than 32 fixtures are required, implement additional e-Node/dmx devices.
3. Maximum number of e-Nodes that can exist on a typical lighting or automation platform = 254 generally. Consult automation platform documentation for more information here.

BILL OF MATERIALS (for IP control)

#	Device	Manufacturer	Part Number	Protocol	Connector Type	Notes
1	Automation family processors	Various	Various	Ethernet	various	
2	Network Switch	Various	Various	Ethernet	RJ-45	
3	e-Node/dmx	Converging Systems	e-Node/dmx	Ethernet	RJ-45 (for Ethernet)	

					RJ-25 for local DMX bus	
4	Third party DMX fixtures	Various	Various	DMX512	RJ-25 for DMX communication	Must terminate final OUT or THRU connector on last DMX fixture using a 120-ohm resistor

e-Node Programming/Device Programming

Minimum requirements for this operation.

-e-Node/dmx with power supply. (If using power supply not provided by factory, DC voltage provided should be between 12v and 24v with output current of at least 90ma.)
-CAT cable to connect e-Node/dmx to local switch or network.
-Necessary cabling to connect e-Node/dmx **PORT 2** to first DMX fixture (see "e-Node Interfacing with DMX Guide"). For reference the pin-outs for **PORT 2** on the e-Node/dmx as well as popular pin-outs for DMX fixtures are included below as well.

Note: The Data**+** signal on one connector should be connected to the Data**+** signal on the other connector. Similar connections for Data**-** and Signal Ground should also be made.

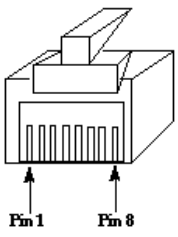
E-Node RJ-45	XLR connector	Std RJ-45 DMX*	Philips RJ-45
Pin 1 (485+)	Pin 3	RJ-45 Pin 1	RJ-45 Pin 2
Pin 2 (485-)	Pin 2	RJ-45 Pin 2	RJ-45 Pin 1
Pin 7 (Gnd)	Pin 1	RJ-45 Pin 7	RJ-45 Pin 7
(all other pins)	N/C	*Std wiring	N/C

e-Node/dmx (MkIII) PORT 2
RJ-45 connector

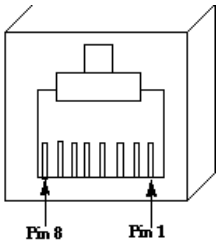
Pin	Signal
1	Data +
2	Data -
3	Do not connect
4	Do not connect
5	Do not connect
6	Do not connect
7	Signal Common/Ground
8	Do not connect

Typical DMX 3-pin XLR connector

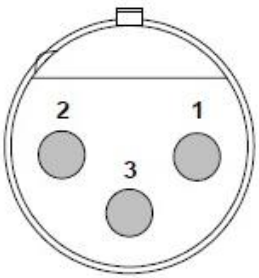
Pin	Signal
1	Signal Common
2	Data -
3	Data +



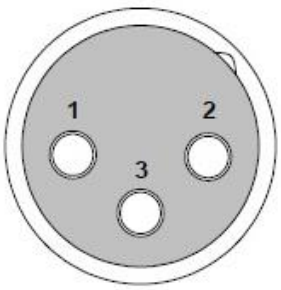
RJ-45 Jack Plug



RJ-45 Jack Face



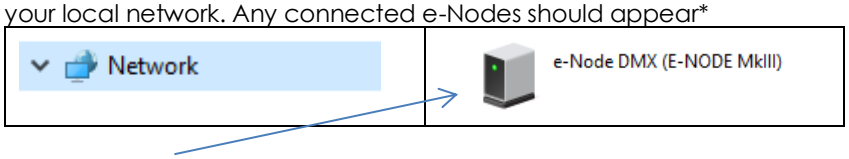
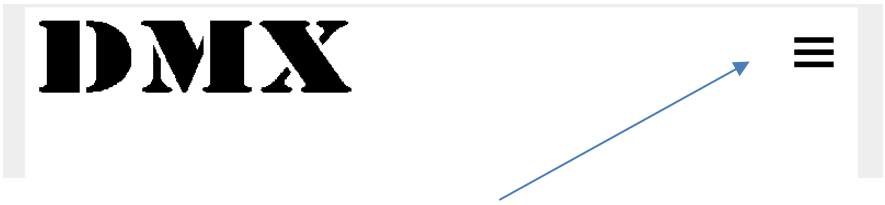
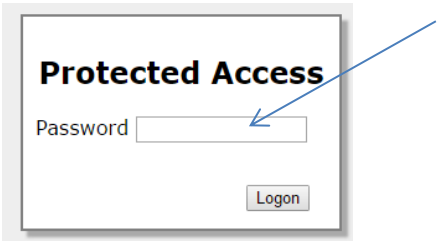
FEMALE

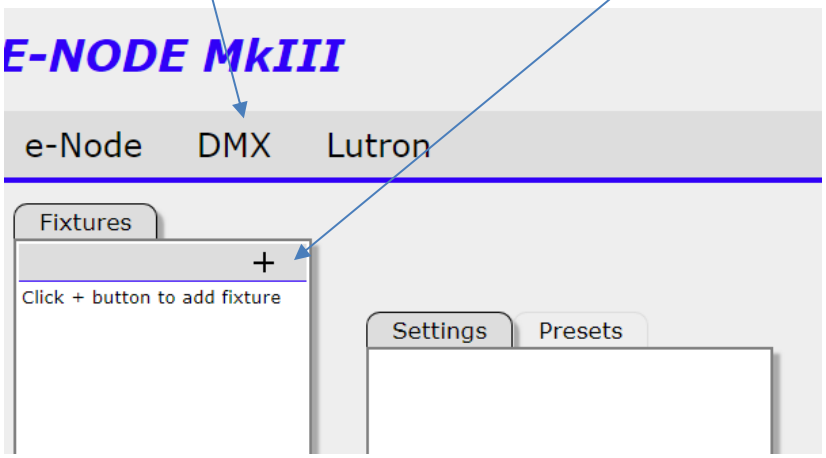
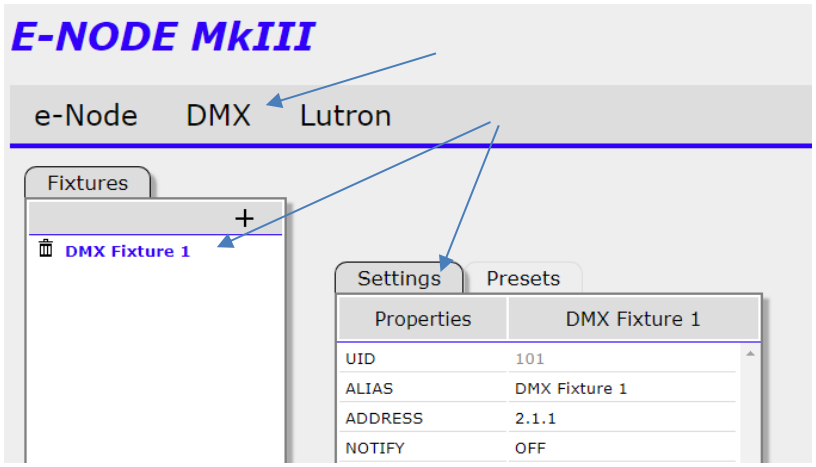


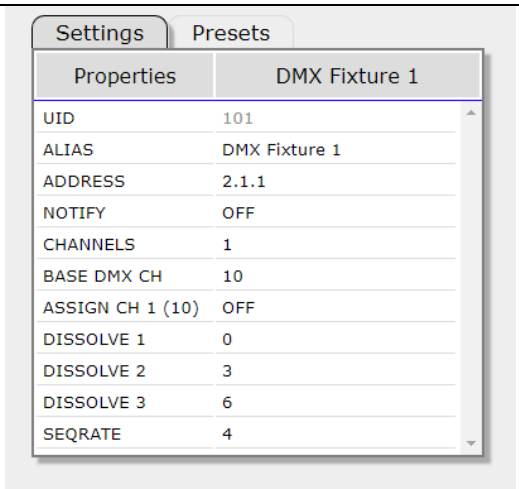
MALE

e-Node/dmx Programming (using new Web Pilot application)

Note: The MK**III** hardware release of e-Node/dmx is required for this level of functionality operating with e-Node/dmx firmware versions 2.01.14 or later. If you have a previous version of your e-Node MKIII hardware please review to legacy directions for "e-Node Interfacing with DMX (MKIII version) version 1.04"

Step	Setting	Choices
DMX-WP1	e-Node/dmx setup	-Power on e-Node/dmx and connect its Ethernet cable to your switch. -Use a Windows computer connected to the same switch and open your Microsoft **File Explorer** (or **Windows Explorer)**  and search for the **Network** tab to expand it to see available uPnP devices on your local network. Any connected e-Nodes should appear*  -Double click on the icon representing your newly discovered e-Node/dmx and the Web Pilot Home Page will appear (as picture below).  -Click on the triple dash menu icon on that home page (above) and you will be asked for a **Password**. Unless this Password has been changed, enter ADMIN and select **Logon**.  ***Note on uPnP Troubleshooting**: You may have to turn on (Microsoft) Discovery or load the **uPnP** service on your respective computer depending upon the version or settings of Windows loaded. Make sure that your router or switch has **UPnP** turned on which in some cases is turned OFF by default. Before you waste too much time resolving **uPnP** issues on your computer, you can always load the standalone e-Node Pilot application and follow the steps (WP5) below to find the IP address of your e-Node dmx. After you have found that address, simply type that address into your address bar on your browser and continue onto the next step.

DMX-WP2	Opening the DMX Fixture Wizard	-Select the DMX tab on the top to access the DMX Wizard. By default, no DMX fixtures are initially installed or present under **Fixtures.** Click on the + button to add first fixture.  -After first (or subsequent) DMX Fixture(s) are installed, each will populate under the **Fixtures** tab. Select that fixture and all of its properties will be displayed using **Settings**. 
DMX-WP3	Enter settings for DMX Fixture	-A number of programmable fields appear that are necessary to fill out in order to establish connection with any connected DMX fixture(s). The entries and available choices are presented below.

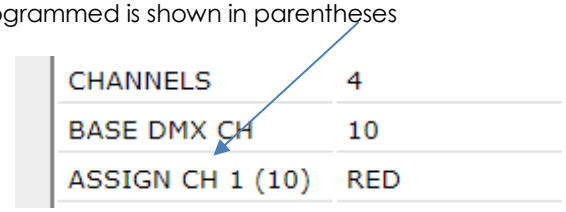


UID	This is an auto-programmed unique ID for each fixture being added. The number cannot be altered. **Note**: no two fixtures can have the same UID.
Alias	You may enter a description for the DMX fixture to make it easier to identify especially if you will be adding multiple fixtures. Click on field to change.
ADDRESS	-Enter a Zone/Group/Node (**ZGN**) address separated by periods that will be used to control this particular Fixture from any supported third-party automation and lighting systems. See information on ZGN Addressing found in any Converging Systems Third-Party Integration document under "Background for Addressing."
NOTIFY	This is a pulldown selector that selects the type of bi-directional feedback which will be provided from the e-Node/dmx back to a supported automation or lighting system. ***Any value other than OFF is required to enable the integration system to able to update its sliders or status buttons when there are color state changes.*** -Mouse select on the current entry and right click to expose available choices. Select desired entry from choices below.

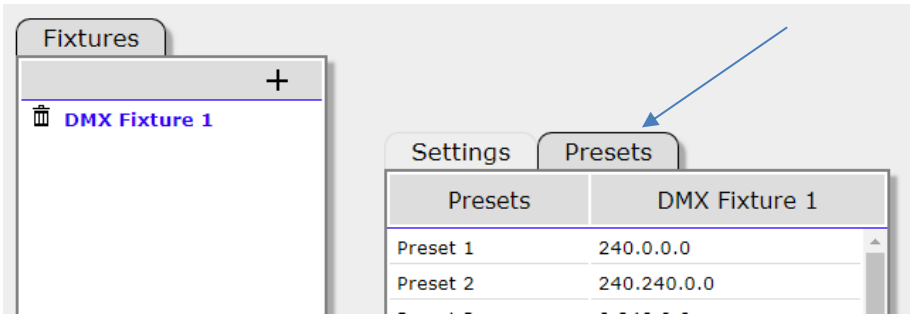
COLOR	This enable Hue/Saturation/ Brightness (HSB) feedback (Preferred for most systems)
VALUE	This enables Red/Green/Blue (RGB) or RGBW feedback. (Useful for legacy systems where just RGB or RGBW control is desired.)
BOTH	This enables both HSB and RGB (RGBW if selected) feedback
OFF	This turns off bi-directional feedback **Note:** for large networks sometimes OFF is preferable to reduce amount of back-

	channel data (but in such case no feedback is provided).
CHANNELS	-Select the number of channels that the e-Node/Pilot wizard will help you program. **Note:** For each **UIDn/DMX** Fixture it is important to select the appropriate # of Channels in order for the embedded software to be able to properly adapt itself for the target output device. Channels 1 to 15 are currently supported. See note on DMX Channels/Compatibility and Interfacing Issues for cases where you require more.
BASE DMX CH	-Select the starting DMX Universe address for the targeted fixture. Typically fixtures have 1, 3, 4 or n-channels available. If your DMX fixture has more than 4 channels available that you wish to control, see DMX Channels/Compatibility and Interfacing Issues.
ASSIGN CH 1(n)	Depending upon the number selected under **Channels** above, the operation of that channel can be programmed here. Drop down boxes permit various "commands" to be assigned to each Channel. Below is a table showing standard Commands available

Option	Notes
RED	Standard for 3- and 4-color devices which provide Red as an available color
GREEN	Standard for 3- and 4-color devices which provide Green as an available color
BLUE	Standard for 3- and 4-color devices which provide Blue as an available color
WHITE	For 4- channel RGBW fixtures, use White for the W channel
MONO	For monochrome DMX fixtures (single color) select MONO
FULL	Typically use this setting if the fixture has a simple dimming channel (that just varies the RGB(W)sliders on a linear basis). Set this channel to **FULL** and use the e-Node/dmx's embedded HUE ACCURATE DIMMING enhanced dimming function available to dim the fixture properly.

<table>
<tr><td></td><td></td><td></td><td>
<table>
<tr><td>OFF</td><td>For DMX fixtures that typically have more than 4 channels, set each non-applicable channel to OFF that you choose not to control (the e-Node will simply send out a “0” to each set channel). For more information see DMX Channels/Compatibility and Interfacing Issues.</td></tr>
</table>
Note: For convenience, the DMX Universe channel associated with the specific channel being programmed is shown in parentheses


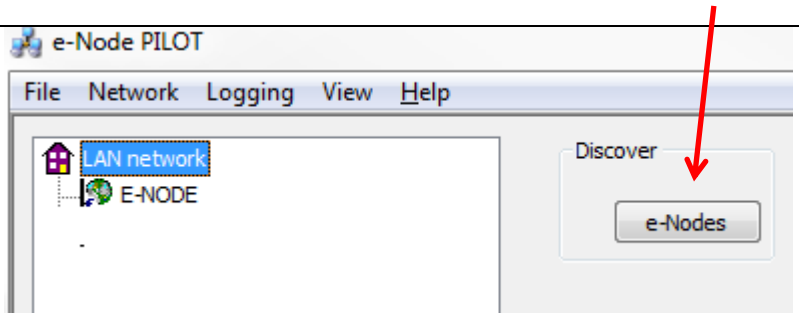
</td></tr>
<tr><td>DMX-WP4</td><td>Continue adding all required DMX Fixtures using the e-Node web application</td><td colspan="2">Once completed with all DMX Fixture additions, you are now ready to start integrating the e-Node/dmx into your third-party automation system.

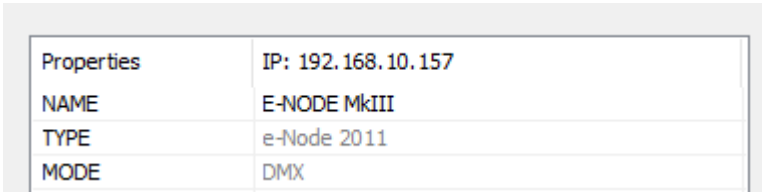
The relevant items that need to be considered are as follows:
-IP address and any passwords for e-Node/dmx
-The Zone/Group/Node address for each DMX fixture
-The type of control available for each DMX fixture (i.e. monochrome using the FADE command, RGB/RGBW commands or HSB brightness commands)</td></tr>
<tr><td>DMX-WP5</td><td>Additional (optional) settings available</td><td colspan="2">Typically, automation and lighting systems support the e-Node/dmx’s built-in STORE and RECALL functions. Some system designers prefer to pre-set their own presets (i.e. RECALLS) and not make the storing of such presets available to the end-user. In such event, you can enter information within the Preset tab to overwrite the factory settings (and simply not expose the saving of presets t to the end-user through your automation platforms UI pages).
 

Depending upon the Number of Channels setting (previously made, the syntax for entering color or brightness data dynamically changes.</td></tr>
</table>

<table>
<tr><td></td><td></td><td>Available color space choices are shown below:
<table>
<tr><th>If Channel is set to</th><th>Color Space</th><th>Settings available)</th></tr>
<tr><td>1</td><td>Monochrome Mode (brightness is only option)</td><td>Level of brightness from 0 to 255 for DMX can be entered.</td></tr>
<tr><td>3</td><td>RGB Mode (3-color mode)</td><td>Separate Red, Green and Blue entries separated by periods
(i.e. 255.255.255 for all colors on. The 1st entry represents Red, the 2nd entry represents Green, and the 3rd entry represents Blue)</td></tr>
<tr><td>4</td><td>RGBW Mode (4-color mode)</td><td>Separate Red, Green, Blue and White entries separated by periods
(i.e. 255.255.255.255 for all colors on. The 1st entry represents Red, the 2nd entry represents Green, the 3rd entry represents Blue and the 4th entry presents White)</td></tr>
<tr><td>>4</td><td></td><td>It is assumed that the Channels is set to >4, that some unused or non-varying channels will be set (see DMX Channels/Compatibility and Interfacing Issues). Regardless, the maximum variable color space (RGBW) is still preserved (regardless of what those RGB and W entries control). Therefore, enter preset color data in the RGBW virtual format with whatever is in the first field controlling the virtual “Red” channel, and whatever is in the 2nd filed controlling the virtual “Green” channel and so on.</td></tr>
</table></td></tr>
<tr><td>DMX-WP6</td><td>Using Pilot application (on a PC) to determine IP address of e-Node/dmx</td><td>Note: In the unlikely event that your computer’s network discovery (uPnP) is not functional and your e-Node/dmx cannot be found with Network Scanners, download and unzip the e-Node Pilot application from the Converging Systems website
http://www.convergingsystems.com/downloads_library.php

-Launch the desktop Pilot application and from the View e-Node tab select the Discover e-node button. Any e-Node(s) connected on the same network will appear as shown. Simply click on the targeted e-Node/dmx and you will find its IP address under the Properties window</td></tr>
</table>





-If e-Node Pilot cannot find your targeted e-Node/dmx, it may have been given a static IP address outside your existing Subnet. In such event, you can reset the e-Node/dmx to Dynamic DHCP Addressing such that Pilot will once again Discover the device. To do so, remove the shroud to the right of the 2-pin power connector and depress the reset button and hold it until **three** sets of flashes on the on-board LED are observed. Immediately, release the button and the on-board PCB LED will go out and then start flashing fast. If it secures a DHCP address in a short period of time, it will start flashing slowly. Then once again try to discover it with the **Discover e-Node button** within Pilot.

Appendix 7

Sample User Interfaces

NICE Programming-User Interfaces

The individual installer typically designs the User Interface (UI) for the particular needs of the end-user. Converging Systems may add from time-to-time new UIs with advanced functionality. Sample UI screens are pictured below.

LED CONTROL ENVIRONMENTS

The following illustrations provide some sample UI for LED control interfaces.

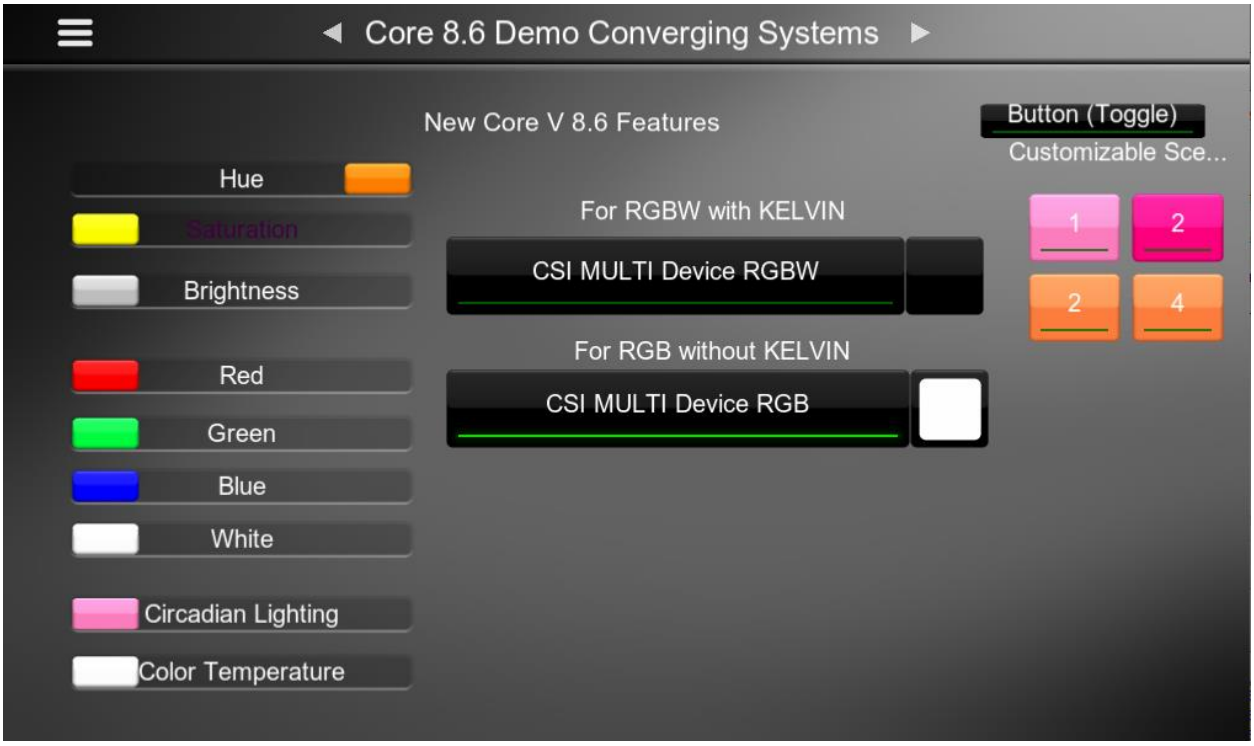


Figure 5

Note: The CSI MULTI Device controls for RGBW devices exposes full control popups. The individual Sliders on the left are often redundant with Core 8.6 new features but are presented to show the capabilities that exists. Customizable Screen buttons eliminate the former concept of having multiple buttons to set Color Temperature and various Recalls.

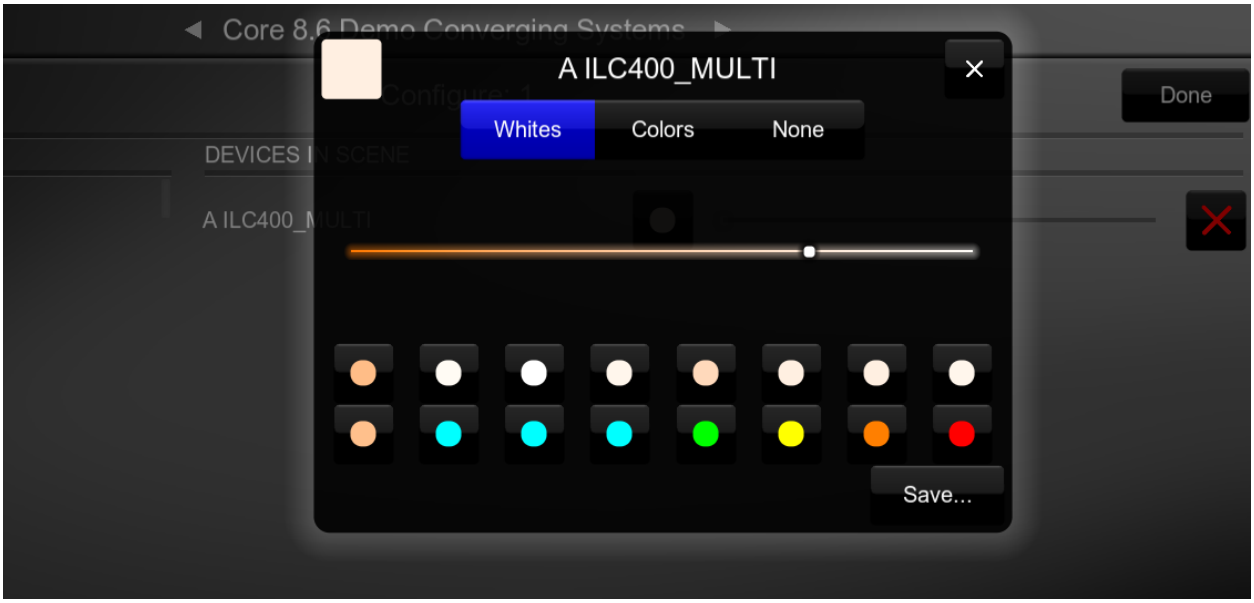


Figure 6

Note: Customizable Scene Pop-ups now enable up to 16 presets to be stored and recalled.

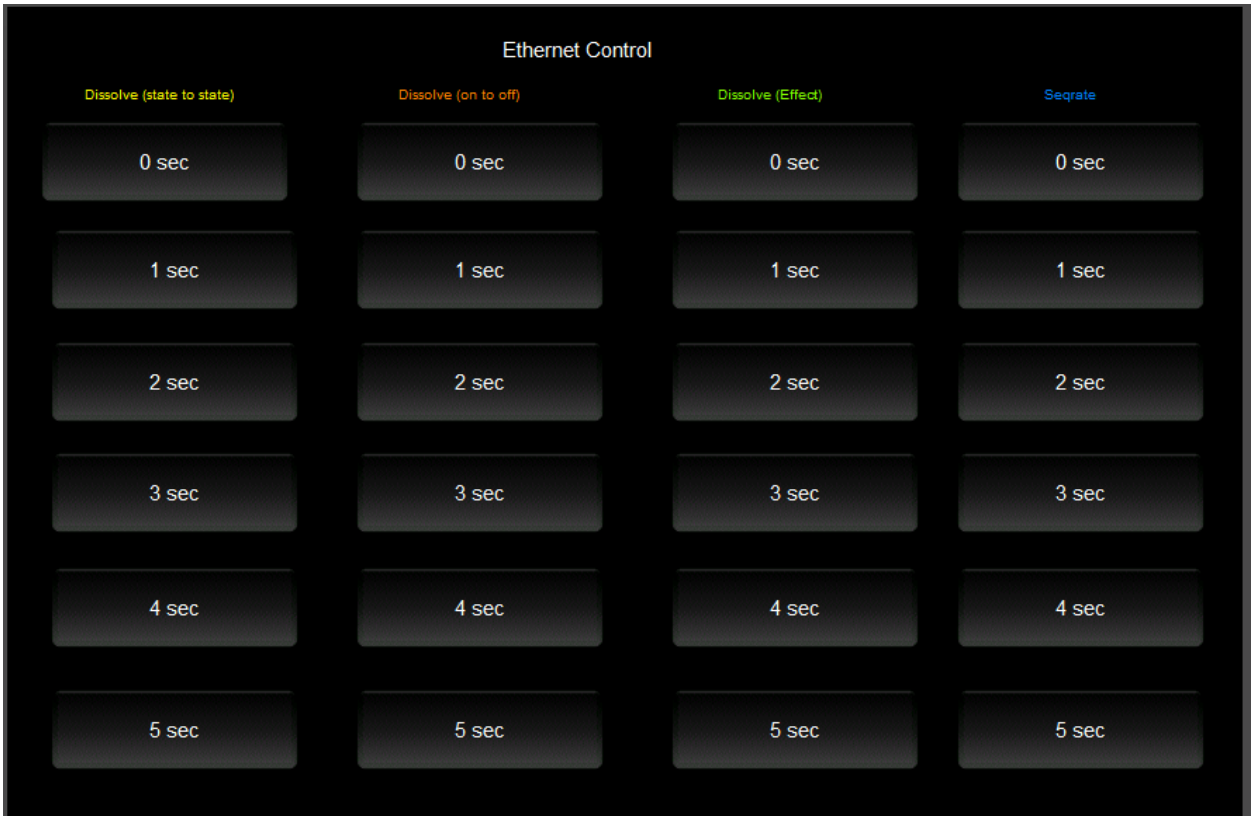


Figure 7

MOTOR CONTROL ENVIRONMENTS

The following illustrations provide some sample UI for motor control interfaces. Future updates to the LUA drivers will be made available supporting these screens.

Figure 8

Note: (Motor Control UP/Stop/Down for up to 4 motors. Preset Recall positions for up to 10 presets. Store Preset positions for up to 9 presets.)

Figure 9

Note: Motor Control UP/Stop/Down for up to 5 motors. Preset Recall positions for up to 3 presets for each motor. Store Preset positions for up to 3 presets for each motor

Appendix 8

Troubleshooting/System Monitoring

See Troubleshooting