It's Under Control®

Integration Designer

User Manual

Version: **11.5**

Copyright © 2023 Remote Technologies Inc. All rights reserved.

Remote Technologies, the Remote Technologies logo, It's Under Control, and Integration Designer are Remote Technologies Incorporated (RTI) trademarks registered in the United States and other countries. Third-party products and brand names are the trademarks of their respective owners.

While RTI believes the information in this document to be accurate, RTI makes no warranty or representation, express or implied, concerning the quality, accuracy, or completeness of the information contained herein.

Remote Technologies Incorporated
5775 12th Avenue East
Suite 180
Shakopee, MN 55379 USA
952-253-3100

Table of Contents

1

FEATURES **9**

HARDWARE/SOFTWARE REQUIREMENTS **10**

SOFTWARE INSTALLATION **10**

SOFTWARE PREFERENCES **10**

UNDERSTANDING THE LAYOUT **13**

CUSTOMIZING THE LOOK **13**

DIFFERENT SECTIONS **15**

TOOLBAR GLOSSARY **16**

RTI Terminology Glossary **21**

THE WORKSPACE AT A GLANCE **26**

STARTING A NEW PROJECT **29**

1. Adding a Main Processor **30**

2. Adding Rooms **31**

3. Adding RTI Controllers **32**

4. Add Expansion Devices **35**

5. Adding Source Devices **38**

ADDING A LOCAL DISPLAY DRIVER SOURCE DEVICE ***39***

ADDING A LOCAL AUDIO DRIVER SOURCE DEVICE ***42***

ADDING A LOCAL AUDIO DRIVER SOURCE DEVICE WITH MULTIPLE ZONES ***44***

ADDING A THIRD-PARTY DRIVER ***45***

ADDING AN IR SOURCE DEVICE ***46***

ADDING A SOURCE DEVICE TO GLOBAL ***50***

ADDING AN RS-232 SOURCE DEVICE ***52***

ADDING A PAGESET ***54***

ADDING A SOURCE AS A BUNDLE ***56***

CONFIGURING PROPERTIES **58**

CONFIGURING SYSTEM PROPERTIES **58**

Configuring RTI Authentication **63**

Processor Configuration **66**

Controller Configuration 75
KX4 Keypad Properties *77*
T4x Controller Properties *85*
RTiPanel Properties *88*
T1-B+ Properties *90*
RK1+ Properties *91*
WK2 Properties *93*
ADDING A KX4 MAIN PROCESSOR 93
Adding a KX4 Expansion Processor 95
DRIVER CONFIGURATION 97
Basic Driver Configuration 97
Configuring a Driver with Dynamic Naming 101
Configuring Auto-Discovery Drivers with Multiple Zones and Sources 103
CONFIGURING ACTIVITIES IN SYSTEM MANAGER 106
Programming Selection Macro Events 108
Programming Deselection Macro Events 118
Benefits of System Manager 125
NAVIGATING THE WORKSPACE 125
Selecting Items in the Workspace 128
Identifying Items in the Workspace 134
Renaming Items in the Workspace 135
Selecting a Source Device in the Controller Configuration 136
Selecting a Source Device Page in the Source Configuration 141
Using Pages and Frames 156
USING LAYERS IN INTEGRATION DESIGNER 158
Understanding the Layer Window 159
Creating a Layer 160
Inserting a Shared Layer 163
Changing the Layer Properties 165
Changing the Layer Priority 166
Deleting a Layer 168

Moving Buttons to Layers 169
WORKING WITH BUTTONS 172
Drawing Buttons 172
Changing Border and Font Colors 174
Changing the Icon Color Overlay 176
Active and Normal State Buttons 177
Aligning Buttons 178
Moving Buttons 179
Arranging Buttons 180
Aligning Buttons 181
Resizing Buttons 183
Equalizing Spacing and Distributing Buttons 185
Centering Buttons 187
Deleting Buttons 188
INTRODUCTION TO BUTTON GRAPHICS 189
Included Bitmap Libraries 189
Using Personal Bitmap Libraries 191
Understanding Graphic Objects 192
Embedding Icons in Buttons 193
Changing Buttons 195
Changing Icon Colors 195
Editing Button Text 196
Button Properties 198
Button Styles 203
MACRO PROGRAMMING 206
Introduction to Tags 206
Entering and Viewing Tags *207*
Auto-Programming Tag Usage *208*
Tag Facts *209*
Tag Exercises *210*
Edit Tags Function *212*

PROGRAMMING BUTTON MENUS 214
INTRODUCTION TO MACRO LEVELS 219
GLOBAL MACROS 221
SOURCE MACROS 222
ROOM MACROS 223
CONTROLLER MACROS 225
Programming Controller Standalone Macros 226
PROGRAMMING MACROS 228
MACRO STEPS EDITOR 228
System Mode Macro Steps 228
Standalone Mode Macro Steps 239
PROGRAMMING VARIABLES FOR FEEDBACK 242
BUTTON STATE FEEDBACK 243
Button Text Feedback 243
Reverse State Feedback 253
Inactive State Buttons 254
Visible State Variables 255
GAUGES AND SLIDERS (GRAPHS) 257
TOGGLE BUTTONS 262
DRAWING GRAPHS AND TOGGLES 264
ITEM LISTS 265
DYNAMIC IMAGES 272
IMAGE LISTS 274
SPINNER (XP-8V OSD) 276
ADVANCED MACRO PROGRAMMING 277
CREATING SYSTEM MACROS 277
GLOBAL SYSTEM MACROS 278
ROOM SYSTEM MACROS 280
SOURCE SYSTEM MACROS 281
MACRO LIST PROGRAMMING 283
CONTROLLER SYSTEM MACRO 286
VARIABLE LISTS 287

CALL MACRO 290
MULTI-ROOM CONTROLLER PROGRAMMING 295
GLOBAL CONTROLLER ROOM MACROS 299
PROGRAMMING SYSTEM VARIABLE TESTS 302
FLAG PROGRAMMING 307
ADVANCED LAYER PROGRAMMING 310
SYSTEM MANAGER PROGRAMMING 318
PROGRAMMING MULTI-ROOM BUTTON MENUS 324
RTI BEST PRACTICES 330
LEARNING TIPS 330
USER INTERFACE DESIGN 331
PROGRAMMING BEST PRACTICE 332
EVENT PROGRAMMING 333
PROCESSOR EVENTS 333
SENSE EVENT 333
PERIODIC EVENT 337
DAILY EVENT 339
STARTUP EVENT 341
DRIVER EVENTS 343
SOURCE EVENTS 347
ROOM SELECTION EVENTS 348
EVENT CONTROL 349
RTIPANEL MULTI LOCATION 353
VIEWPORTS 361
PROGRAMMING HYBRID MACROS 367
PROGRAMMING RELAYS 370
INTERNAL VARIABLES 378
VOIP 378
VOIP AUDIO 380
RTSP 381
REMOTE FINDER 381

FRAMES 381
RTIPANEL CONNECTION 382
VOLUME 383
ZIGBEE 383
OBJECTS 383
ANALOG VIDEO 384
AXIS CAMERA VIEWER 385
GENERIC MJPEG VIDEO 387
GENERIC RTSP STREAMING 387
HDMI AND COMPOSITE VIDEO 389
HTTP REQUEST 390
ICREALTIME MJPEG CAMERA 391
PANASONIC MJPEG CAMERA 392
RTIPANEL ANDROID OPEN URL 394
RTIPANEL MULTIPLE CONFIGURATION COMMANDS 394
RTIPANEL OPEN URL 395
SNAPAV MJPEG CAMERA 398
TWO-WAY ITEM LISTS 399
VIRTUAL PANEL 400
VIVOTEK MJPEG CAMERA 402
VOIP 404
VOIP AUDIO 408
ZIGBEE 412
CREATING A SEPARATE ZIGBEE NETWORK 412
CREATING A SEPARATE RS-485 NETWORK 415
UPDATING FIRMWARE 419
VIEWING FIRMWARE VERSIONS 419
UPDATING FIRMWARE VIA ETHERNET 422
UPDATING THE PROJECT FILE 426
IR LIBRARY MANAGER 429
RS-232 LIBRARY MANAGER 445

Chapter 1. Introduction

Integration Designer™ is a powerful Windows-based software package that allows custom installers to program an RTI control system. It is a visual application with various tools to create intelligent professional programming solutions. Integration Designer is the fastest custom installation and continues the evolution of the industry-leading CEDIA Hall of Fame control software package.

The software can completely customize the user interface and experience, which is ideal for commercial solutions and projects requiring special interface considerations. No graphic design skills are required, and Integration Designer features stock templates and assets to give your projects a professional look. Installers may import their graphics for a higher level of customization.

For those interested in contracting and deploying RTI control systems quickly, there is a fast track that allows for quick project builds and several productivity features, such as auto-programming and a simple workflow.

FEATURES

- Allows you to access a user interface to define all control system parameters without needing command line code.
- Fully customizable and stock templates are available, which can be mixed and matched for the ultimate user interface.
- Features a wizard-type function allowing you to build your project quickly.
- Auto programming and quick maintenance are made possible in the software using a special tag feature combined with different command levels.
- Room, Activity, and Power menu options are automatically built into your interface, saving valuable time and effort.
- A full graphics suite is included, and other graphics can be easily imported.
- Graphics are fully scalable in the software and can be customized with minimum effort.
- Buttons and other graphics can be easily aligned, colored, and made to match the attributes of other graphics.
- A built-in driver store, a licensed IR, and an RS-232 library are regularly updated.
- Powerful Macro Steps provide precision programming while offering solutions to issues.
- A layer feature allows you to create and share graphics and maintain them in an individual location.

- A bundle feature exists to export source devices, preserving programming, graphic design, and the entire code database.
- System Manager is a powerful feature that tracks what rooms, activities, and devices are currently used. Activity macros are programmed in an individual location, saving time and effort.

HARDWARE/SOFTWARE REQUIREMENTS

- Windows 10 PC or the latest version
- 64-bit operating system, x64-based processor
- 8GB of RAM or more
- 1920x1080 screen resolution or greater
- Ethernet or Wi-Fi
- USB and Mouse Connection

SOFTWARE INSTALLATION

- Ensure there are no open programs.
- Download the latest Integration Designer version on the RTI Dealer website.
- The first time the software runs, select "Check for Updates Automatically."
- You may need to enter your dealer site login credentials the first time you use the software.

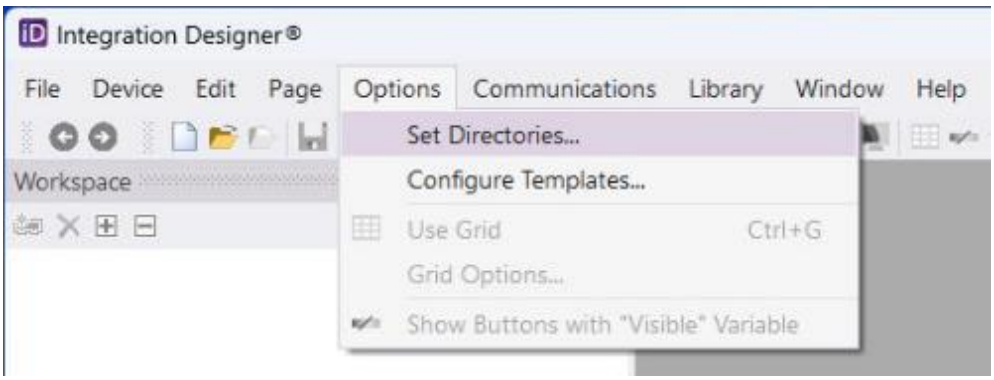
The software installation will also include the following in the installation:

- **IR Library Manager** creates and maintains a personal IR library database.
- **RS-232 Library Manager** – used to create and maintain a personal RS-232 library database.
- **RTI Music Configuration** – used to configure options for RTI Music products.
- **USB Drivers** – Used to synchronize and upload firmware to RTI controllers.

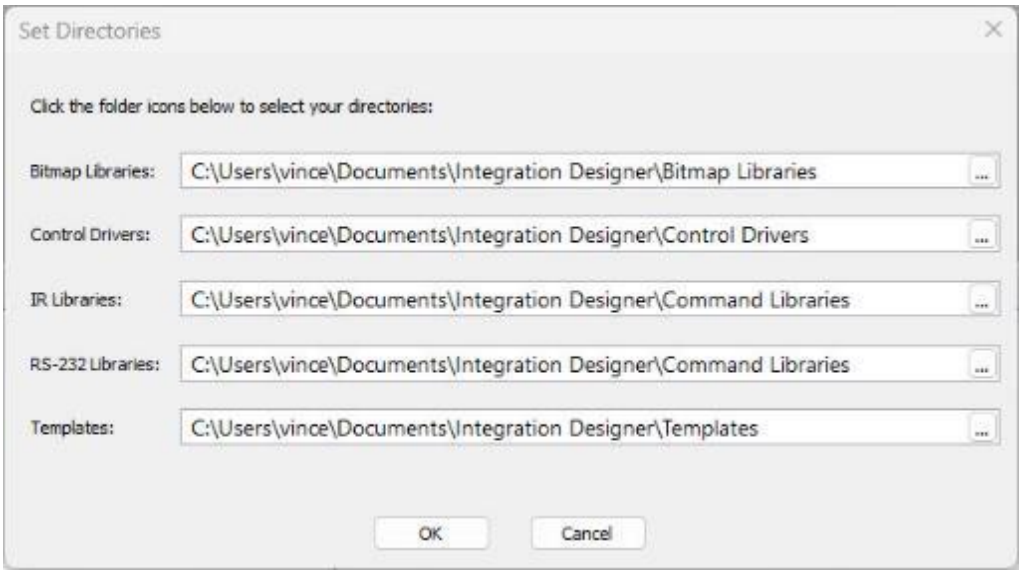
SOFTWARE PREFERENCES

When opening the software for the first time, you may configure personal preferences depending on where you want to store your graphics, IR libraries, RS-232 libraries, and RTI Templates.

- On the top toolbar of the software, select **Options**, then select **Set Directories**.



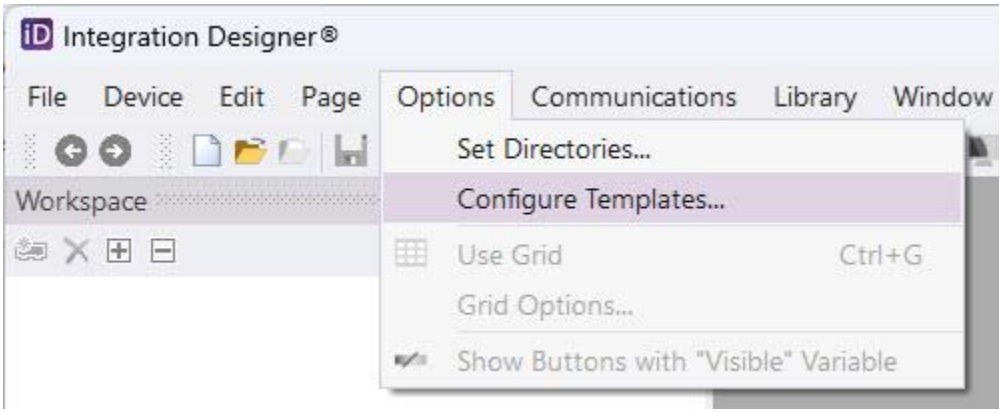
These are the default directories that the software will point to by default. While the bitmap library will be changed at various times depending on where personal graphics are located, the rest of the directories can be left with the default value and utilized for personal drivers, IR libraries, RS-232 libraries, and RTI templates.



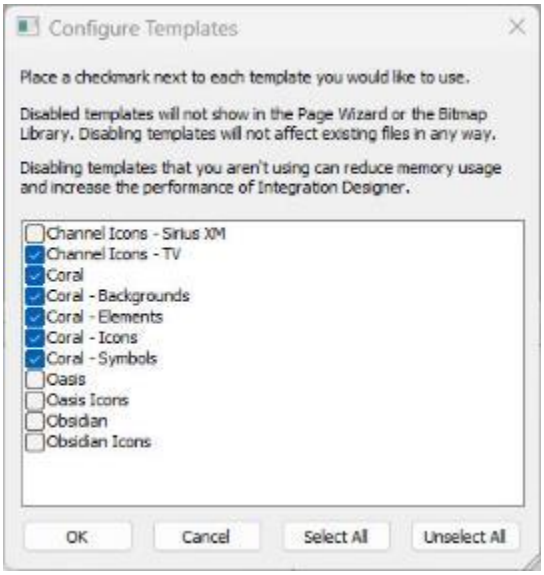
Please note that the Templates directory is where bundles are exported. Bundles allow the installer to export source devices configured for future use. Graphics, code libraries, and programming are preserved.

Another feature worth noting is the ability to suppress templates from displaying in the bitmap library. Templates may be acquired through third-party graphic developers, other installers, or self-creation. Suppressing templates unrelated to your project will make it easier to work with the graphics.

- On the top toolbar of the software, select **Options**, then **Configure Templates**.

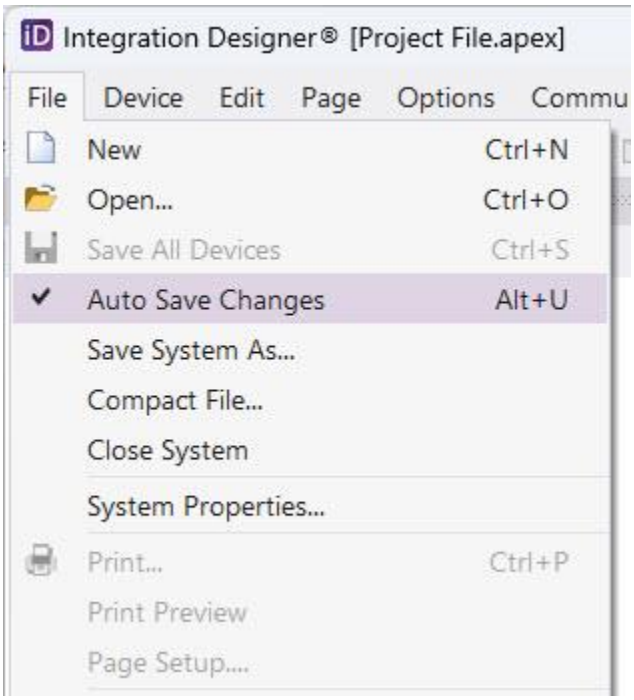


We can deselect all the templates except for the Coral template assets and Channel TV Icons.



Another option we can choose as a matter of personal preference is to enable the auto-save function in Integration Designer. Turning on this feature will automatically save the project file regularly, preventing data loss during a power outage, computer crash, or interruption.

To enable the auto-save feature, select **File** from the toolbar and select the **Auto Save Changes** option. Selecting it again will toggle the feature off. A check next to the Auto Save Changes feature indicates that it will perform autosaving functionality.

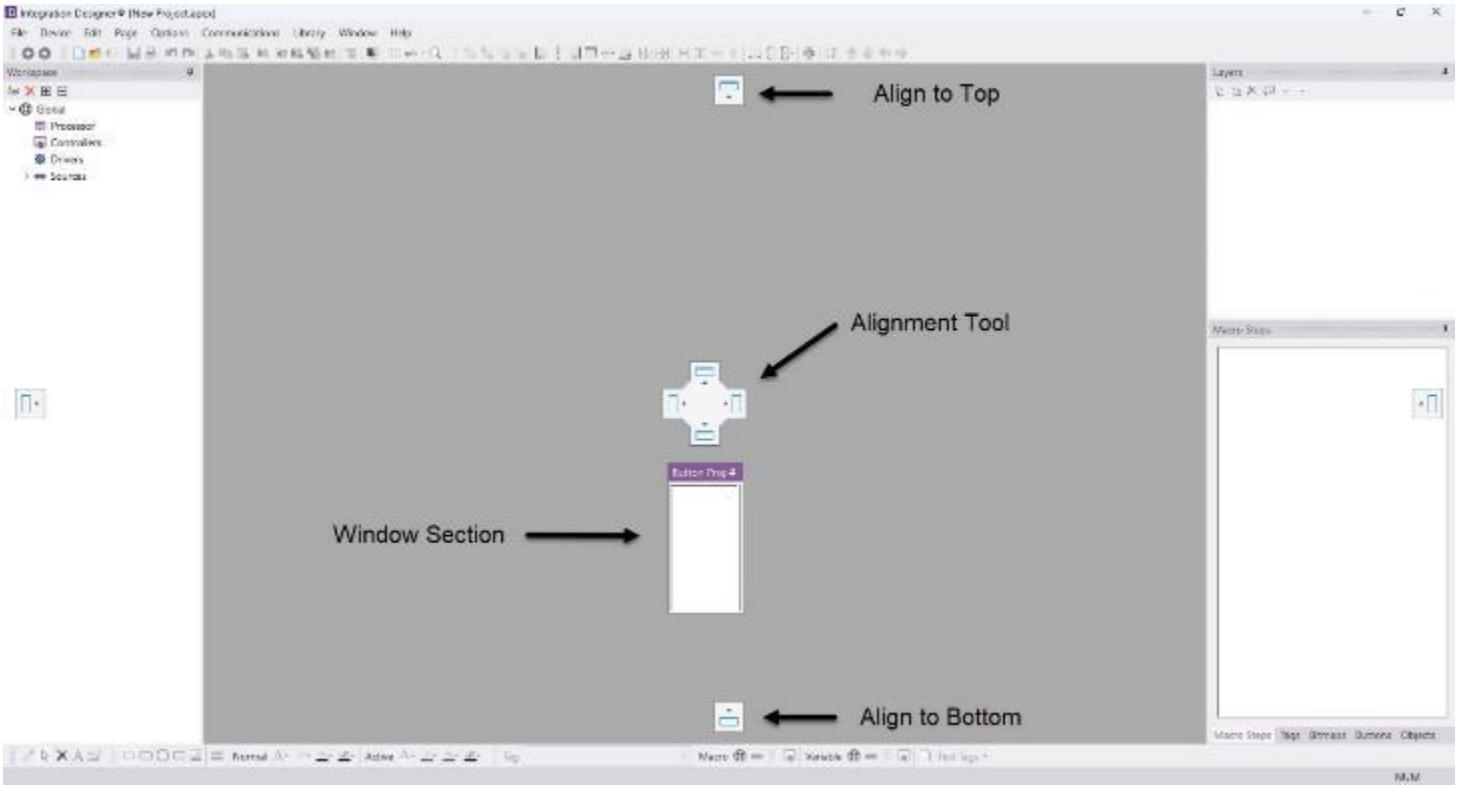


Alternatively, the installer can select the Alt + U function to toggle the autosaving mode in the software.

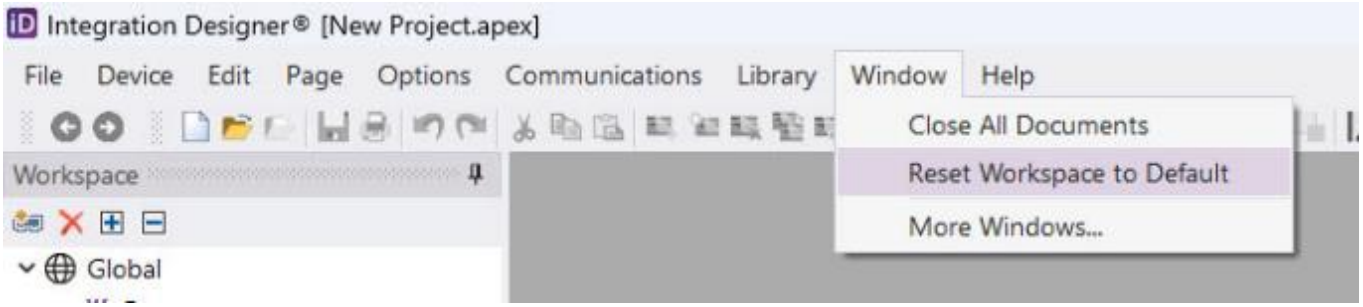
Understanding the Layout

CUSTOMIZING THE LOOK

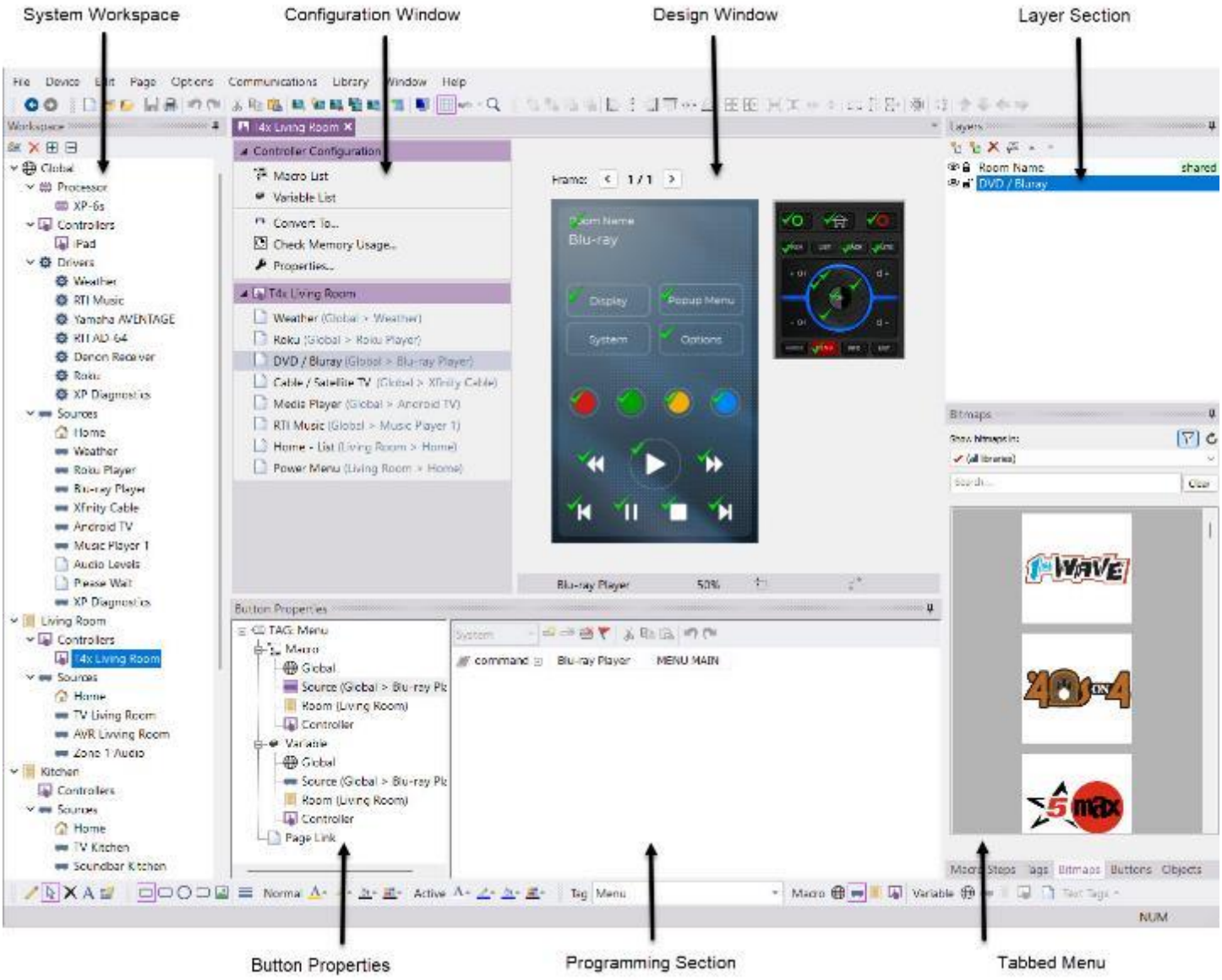
The different sections of the software can be snapped into various positions on the screen using the align tool. It will snap into the selected region by dragging the window by the border to an area of the alignment tool.



You can configure your sections by snapping them to any screen area. To get back to the default layout of the software, select the **Window** option in the top toolbar, then select **Reset Workspace to Default**.



DIFFERENT SECTIONS



Integration Designer is divided into seven sections, each with a unique function. Here are the sections that form the software from left to right:

System Workspace: An organized system tree including rooms, controllers, source devices, and drivers in a project file.

Configuration Window: Configure the options, properties, and features of anything selected in the system workspace.

Design Window: Displays the selected device for editing the user interface and programming.

Button Properties – Define the type of macro or variable in the programming section.

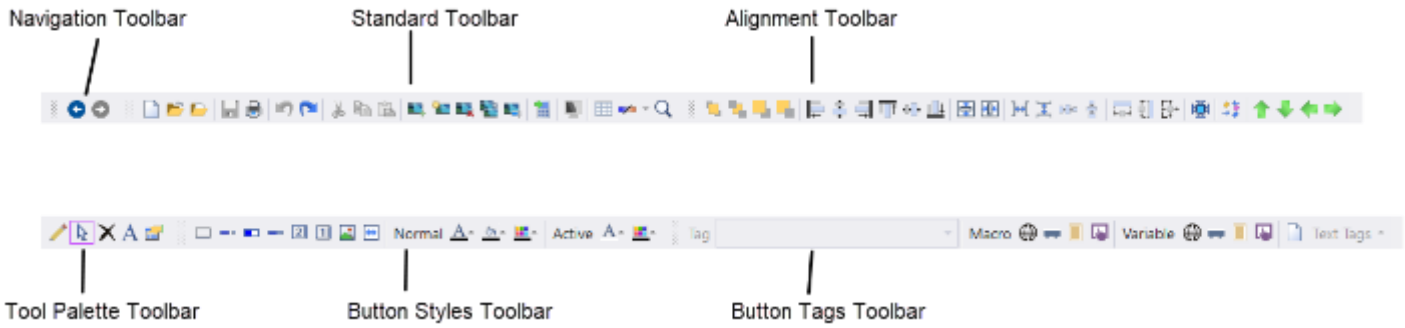
Programming Section – View, create, and edit macros and variables for feedback.

Layer Section – Create, manage, and share layers on the user interface.

Tabbed Menu – Access the bitmap library, tag window, buttons, objects, and Macro Steps.

TOOLBAR GLOSSARY

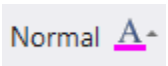
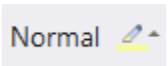
Integration Designer features a variety of toolbar icons for quick action selections.

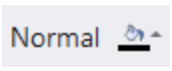
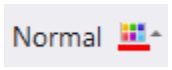
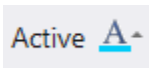
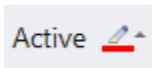
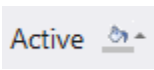
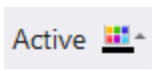
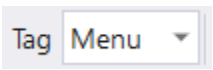
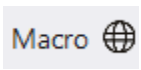
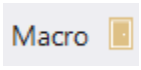
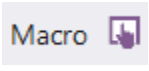
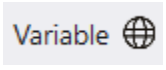
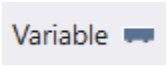
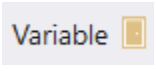
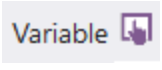
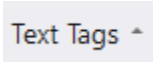


	Back	Alt + Left Arrow	Back to the previous menu.
	Forward	Alt + Right Arrow	Forward to the next view.
	New System	Ctrl + N	Create a new system file.
	Open	Ctrl + O	Open an Existing Document.
	Merge		Merge select pages into this file from another one.
	Save	Ctrl + S	Save all currently open files to disc.
	Print	Ctrl + P	Print the active document
	Undo	Ctrl + Z	Undo the last action
	Redo	Ctrl + Y	Redo the previously undone action

	Cut		Cut the currently selected Macro Step.
	Copy		Copy the currently selected macro step.
	Paste	Ctrl + V	Paste the clipboard contents.
	New Page	Insert	Add a new page to the client file.
	Page Wizard	Ctrl + W	Insert a pre-defined page into this file.
	Delete Page	Ctrl + K	Remove the currently displayed page.
	Duplicate Page	Ctrl + L	Make a duplicate copy of the current page.
	Rename Page	Ctrl + E	Rename the current page name.
	Send File	Ctrl + T	Send the file to devices.
	Preview Mode	F8	Preview the applicable device in standalone mode only.
	Use Grid	Ctrl + G	Toggle the alignment grid on and off
	Show Visible Buttons		Toggle the visibility of buttons with a variable attached to the "visible state."
	Zoom Percentage		Change the zoom percentage of the current device.
	Bring to Front	Ctrl/Shift + F	Bring this button to the front
	Send to Back	Ctrl/Shift + B	Send this button to the back
	Bring Forward	Ctrl + F	Bring this button forward
	Send Backward	Ctrl + B	Send this button backward
	Align Left		Line up all selected buttons on their left side.
	Align Center		Line up the horizontal centers of selected buttons.
	Align Right		Line up all selected buttons along their

		right side.
	Align Top	Line up all selected buttons along their top side.
	Align Middle	Line up the vertical centers of all selected buttons.
	Align Bottom	Line up all selected buttons along their bottom side.
	Center Vertical	Center selected buttons vertically on the page. Hold SHIFT to center on the selection anchor.
	Center Horizontal	Center selected buttons horizontally on the page. Hold SHIFT to center on the selection anchor.
	Equalize Spacing Across	Make all selected buttons have the same amount of horizontal space between them.
	Equalize Spacing Down	Make all selected buttons have the same amount of vertical space between them.
	Distribute Buttons Across	Distribute the selected buttons evenly across the page. Hold SHIFT to distribute across the selection anchor.
	Distribute Buttons Down	Distribute the selected buttons evenly down the page. Hold SHIFT to distribute down the selection anchor.
	Make Same Width	Make all selected buttons the same width.
	Make Same Height	Make all selected buttons the same height.
	Make Same Size	Make all selected buttons the same size.
	Size to Bitmap	Size the button to fit the bitmap.
	Make Same Style	Make all selected buttons the same color and style.

	Up	Shift + Up Arrow	Move the selected button(s) up one pixel
	Down	Shift + Down Arrow	Move the selected button(s) down one pixel
	Left	Shift + Left Arrow	Move the selected button(s) one pixel to the left.
	Right	Shift + Right Arrow	Move the selected button(s) one pixel to the right.
	Draw Button		Draw buttons in the currently selected style.
	Select Button		Select buttons to move, copy & program.
	Delete		Delete button or button criteria.
	Edit Text		Edit button text
	Edit Properties		Edit the button properties.
	Rectangle Button		Draw a button with square corners.
	Rounded Rectangle Button		Draw a button with rounded corners.
	Circle Button		Draw an elliptical button.
	3D Button		Draw a three-dimensional button.
	Dynamic Image		Draw a dynamic image.
	Line Width		Set the line width of the border on the button.
	Text Color (normal)		Set the text color on the normal state button.
	Line Color (normal)		Set the borderline color on the normal state button.

	Fill Color (normal)	Set the fill color on the normal state button.
	Icon Overlay Color (normal)	Set the icon overlay color on the normal state button.
	Text Color (active)	Set the text color on the active state button.
	Line Color (active)	Set the borderline color on the active state button.
	Fill Color (active)	Set the fill color on the active state button.
	Icon Overlay Color (active)	Set the icon overlay color on the active state button.
	Tag Window	The name of the tag on the button
	Global Macro	Program or edit a global macro type
	Source Macro	Program or edit a source macro type
	Room Macro	Program or edit a room macro type
	Controller Macro	Program or edit a controller-level macro type
	Global Variable	Program or edit a global-level variable
	Source Variable	Program or edit a source-level variable
	Room Variable	Program or edit a room-level variable
	Controller Variable	Program or edit a controller-level variable
	Page Link	Program or edit a page link on a button
	Text Tag	Program or edit a text tag on a button

RTI Panel Devices

	Slider Object	Draw a two-way slider.
	Switch Object	Draw a two-way switch.
	Gauge Object	Draw a two-way gauge.
	Scrolling List Object	Draw a two-way scrolling list.
	Static List Object	Draw a local, one-way scrolling list.
	Viewport	Draw a Viewport

Please note that toolbar functions are specific to the RTI controller type. Certain toolbar functions will not be available on all controllers.

RTI Terminology Glossary

RTI terminology is important and may differ from other control manufacturers.

Active State Button – A button will display the active state when pressed. An active state should be programmed on all buttons because the button appears to have been pressed.

Activity – An activity is a set of commands programmed to set up a specific task. "Watch Cable," "Listen to Music," and "Control Lighting" are examples of activities.

Add Workspace Item – In Integration Designer, the Add Workspace Item will guide the installer by adding rooms, controllers, and various source device types to a project file.

Bitmaps – Bitmaps are buttons that are created to execute commands. Bitmaps may have a normal and active state, and an icon can be embedded inside a Bitmap.

Bundles – Bundles provide a way to export configured source pages for multiple controllers in the software. They preserve programming, graphics, and the entire code set and can be imported into future projects.

Button – A button is a graphical element for programming commands, feedback, page links, and assigning names. Buttons may be drawn directly in the software, imported, or provided in the bitmap library. They can be active or inactive, depending on the functionality.

Controller – A controller is an RTI device with a user interface. All RTI remote controls, keypads, RTiPanel devices, and virtual panels are considered controllers.

Controller Macro – A specialized macro ensuring a tag will share the same macro programming on the controller it was conceived. Typically, the commands are available in the room in which it was created.

Controller Variable – A specialized variable ensuring a tag will share the same feedback programming on the controller it was created. Typically, the feedback is available in the room where it was created.

Driver – A control program developed for a specific device using RTI's SDK and API. A driver can use different control protocols, feature two-way communication, and have a more comprehensive code library than an IR database. Typically, drivers allow the processor to update different variables or states.

Driver Events – Driver events can trigger automated tasks based on certain device statuses. For example, whenever the processor detects an OFF status for a device, it will turn the lights off. Each driver has different event types depending on the device feature set.

Ethernet – Ethernet is a control protocol typically used on applicable keypads that will use the wired internet wiring for communication to the processor.

Expansion Device – An expansion device may be a processor or accessory device connected to the internet to communicate with the main XP processor. It may expand ports, relays, and control points in larger projects.

Global Area – Each project file automatically creates the global area in the Workspace. Shared resources such as multiroom controllers and source devices that require control interfaces on two or more local room controllers are added to the global area.

Global Macro – A specialized macro ensuring a tag will share the same macro programming in all rooms, on all controllers, and any source device.

Global Variable – A specialized variable ensuring a tag will share the same feedback programming in all rooms, on all controllers, and on any source device.

Icon – An icon is a graphic with no configured attributes, typically embedded in a button. Icons have a single state and should not be used for buttons. They may also be used for display purposes.

Layer – An interface can be divided into different layers depending on functionality. These layers can be shared throughout a project or made to be visible under certain conditions. Layers include status alerts, pop-up menus, toolbars, and sub-menus.

Macro – A macro is a command or multiple commands programmed to accomplish a task. Macros can be programmed on buttons, events, and other places in Integration Designer.

Normal State – A button's natural appearance without being pressed. A button will always have a normal state. A contrasting active state will also be created so the user knows the button is pressed.

Output Tab – In most controller properties, the installer can select the communication protocol used to communicate with the processor in the project file and determine if one-way or two-way communication is needed.

Pageset – A pageset is a source device added to a project not backed by a true code set. An example of a source device without a code set would be a "Welcome Page," "Please Wait," and a blank source device that can be used for a landing page.

Processor – The processor is the central part of an RTI control system and coordinates communication between controllers in a project file. The processor features an astronomical clock for time-based events and stores critical project information.

Processor Events – Processor Events are specialized events using features of a processor such as its sense inputs, time-based events using the built-in astronomical clock, and timed-based events.

Room Macro – A specialized macro ensuring a tag will share the same macro programming in the room it was created, all controllers assigned or selecting that room, and any source device placed in that room.

Room Variable – A specialized variable ensuring a tag will share the same feedback and programming in the room it was conceived on all controllers assigned or selecting that room and any source device placed in that room.

Rooms – A room is an area added to the Workspace used to designate where a control session may occur on the project site. An equipment closet used to centralize equipment would not be a room since no interaction occurs there.

RTiPanel – RTiPanel includes iOS or Android-based mobile devices for wireless and portable control. RTiPanel devices require licenses tied to the MAC address of a processor in a project file.

RTIQ – RTI's intelligent monitoring service that lives on the control processor. RTIQ can be configured to monitor devices and trigger actions remotely. Installers can configure options to send emails or text alerts if a condition is met.

Source Macro – A specialized macro ensuring a tag will share the same macro programming on the source and Room it was conceived on all controllers.

Source Variable – A specialized variable ensuring a tag will share the same feedback programming on the source and room conceived on all controllers assigned or selecting that room.

Sources – Sources are devices added to the project file for control. Source devices tend to be used for activities. An example of a source device is a music streamer used for the activity "Listen to Music."

System Manager – Intelligence built into Integration Designer to track various statuses in the control interface. The system Manager simplifies programming and can track the room's power on or off, audio on or off, selected sources, and source in use statuses.

Tags are a logical naming convention developed by RTI assigned to buttons and feedback to assign specialized macros and variables. Tags are responsible for auto-programming and maintenance and are used with software features.

Variable – A variable is assigned when two-way feedback is required. With a Boolean value of available data points, programmers can utilize button-state feedback and system variable tests.

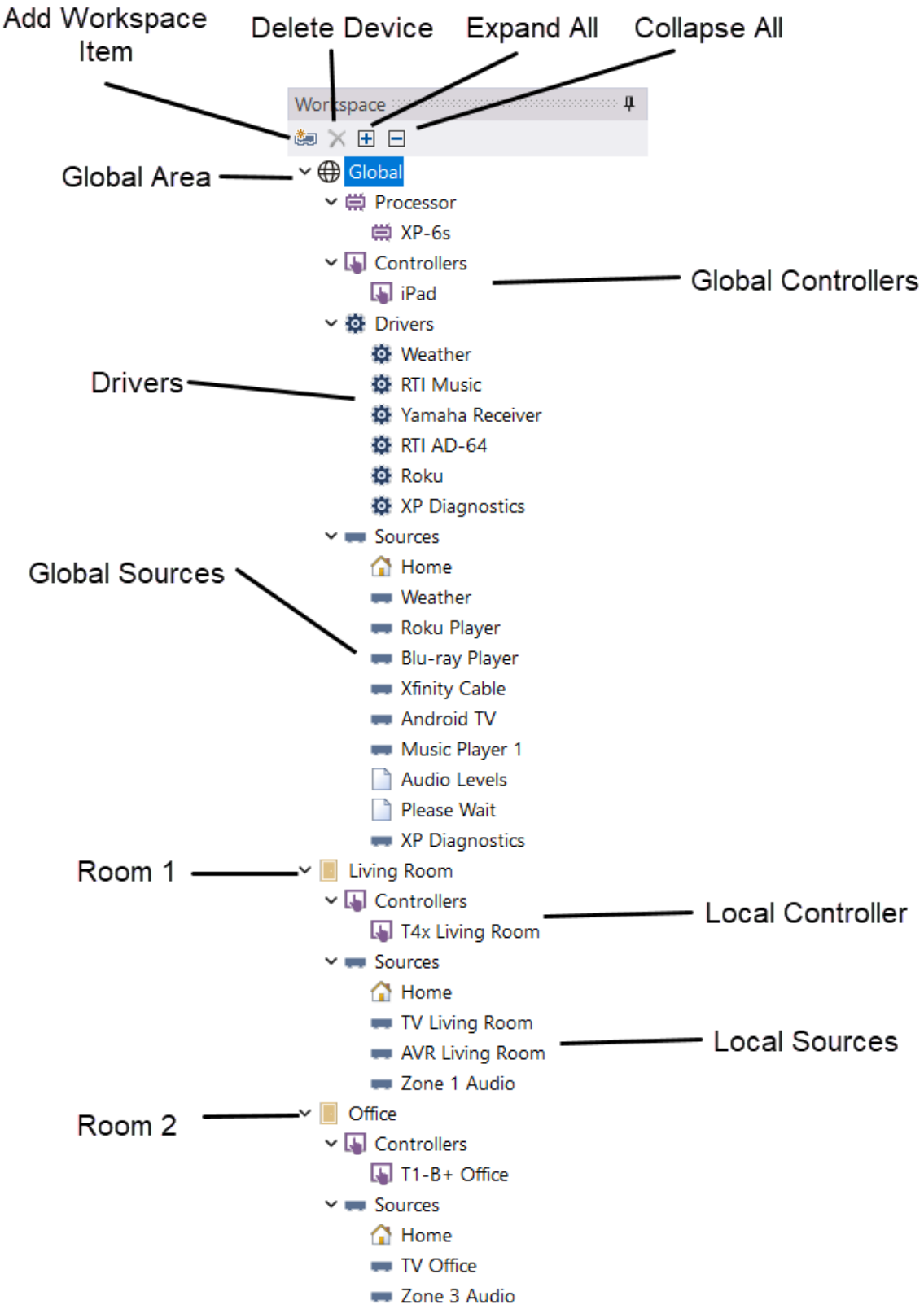
Virtual Panel – A virtual panel is a control interface provided for a Windows-based PC that can be used as an installer tool or a client control interface. While a single local virtual panel assigned to the local area network does not require a license, RTI installers must purchase a license if using it on the WAN or if additional virtual panel devices are required.

Workspace – The Workspace is an area of the software where the programmer places the main processor, rooms, controllers, drivers, and source devices.

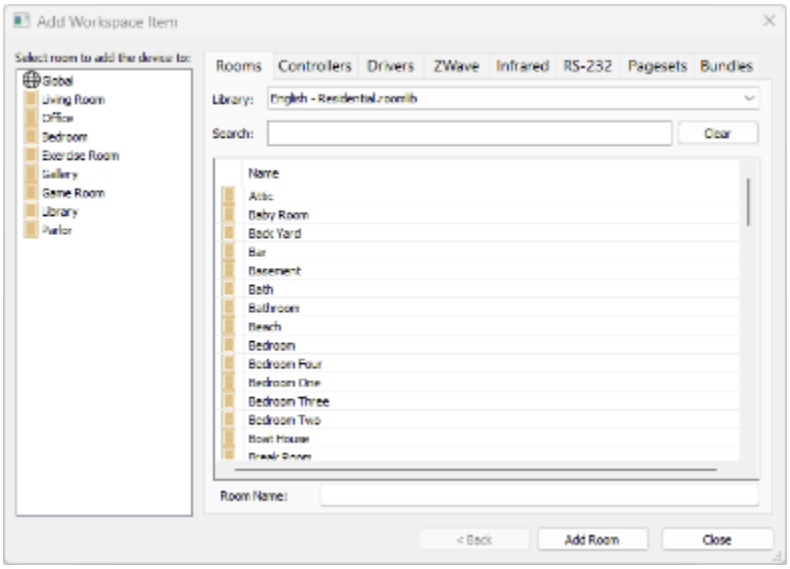
Zigbee – Zigbee is a wireless standard utilized by RTI handheld controllers. Zigbee is low cost, low latency, and low powered, which is an ideal solution for RTI handheld controllers.

THE WORKSPACE AT A GLANCE

The Workspace in Integration Designer is organized by the main processor, the project's drivers, rooms, controllers, and source devices. Viewing the Workspace will provide a deep understanding of what devices and rooms comprise the project and how they function.


Add Workspace Item
Delete Device
Expand All
Collapse All
Workspace
Global Area
Global
Processor
XP-6s
Controllers
iPad
Global Controllers
Drivers
Weather
RTI Music
Yamaha Receiver
RTI AD-64
Roku
XP Diagnostics
Drivers
Sources
Home
Weather
Roku Player
Blu-ray Player
Xfinity Cable
Android TV
Music Player 1
Audio Levels
Please Wait
XP Diagnostics
Global Sources
Room 1
Living Room
Controllers
T4x Living Room
Local Controller
Sources
Home
TV Living Room
AVR Living Room
Zone 1 Audio
Local Sources
Room 2
Office
Controllers
T1-B+ Office
Sources
Home
TV Office
Zone 3 Audio

Add Workspace – Select the add workspace icon, and a menu will display to add rooms, controllers, and source devices to your project.



Delete Device – Select an item in the Workspace, then select the delete button. Once you confirm your selection, the item will be deleted.

Expand All – Display the entire Workspace, controllers, and source devices in each Room.

Collapse All – Fold up the Workspace so you can only see the rooms in the project.

Global Area – Automatically included with every project file, this is where multiroom controllers and shared source devices that require programmed user interfaces on more than one Room's controllers are located.

Global Controllers – Controllers that control more than one Room in a project file are placed in the global area. Global controllers are considered multiroom controllers. When adding a controller to the global area, you can create a programmed user interface for all sources in the project file.

Global Sources – These sources are placed in the global area if shared between multiple local room controllers.

Drivers – Drivers in the project file may be found and configured under the driver tab in the Workspace.

Room – A location or zone in the project where a control interaction occurs.

Local Controller – A controller placed in a local room is confined to that Room. When adding a local controller to a room, you can create programmed user interfaces on that room's controllers and any multiroom controller.

Local Sources – Similar to local controllers, local source devices are placed in a room when confined to that Room and do not require programmed user interfaces on other local room controllers. Adding a source to a local room will allow you to create programmed user interfaces on all the controllers in that room and any multiroom controllers in the global area.

Additional Rooms – Each room added to the project contains a placeholder for controllers and source devices.

Starting a New Project

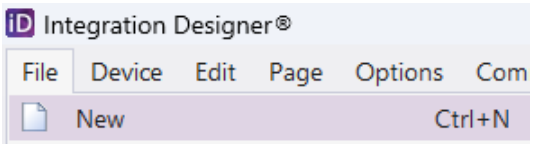
When starting a new project using Integration Designer, it is important to ensure you have collected all the essential information on the types of devices that comprise the system and their role.

Here is a general list of information needed before the programming phase. This List will be a checklist to ensure the system is designed before programming.

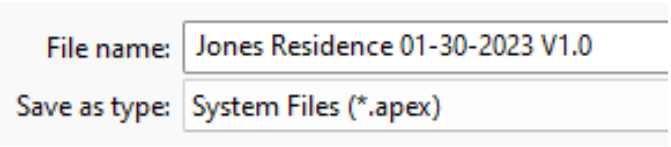
- Client Information (Name, Address, Cable, or Satellite Provider)
- RTI Main Processor & All Communication Protocol
- Expansion Devices (Location, Purpose, & Name)
- Room Information (Determine if the Room is used for Video, Audio, or both)
- Controllers and the Room and devices they need to control. Secure Licenses for RTiPanel devices.
- Video and Audio Source Devices – what controllers need to control these devices?
- Shared Devices (Lighting, Climate, Shades, Security, Surveillance, etc.)
- Relay & Sense Inputs
- Activities required by the end-user on a room-by-room basis.
- Driver & Network Information
- Communication (Zigbee, RS-485, Ethernet, 433Mhz, IR) – Determine how your controllers communicate and what provisions need to be made.
- Design Considerations (The client experience.)
- Automated Event Control (What types of events would suit the client's needs and lifestyle?)
- RTIQ Monitoring and contingency equipment (IP Power Management)
- Is Remote Access needed?

Once this information is determined, you are now ready to program!

1. Select **File** on the top toolbar, then select **New**.



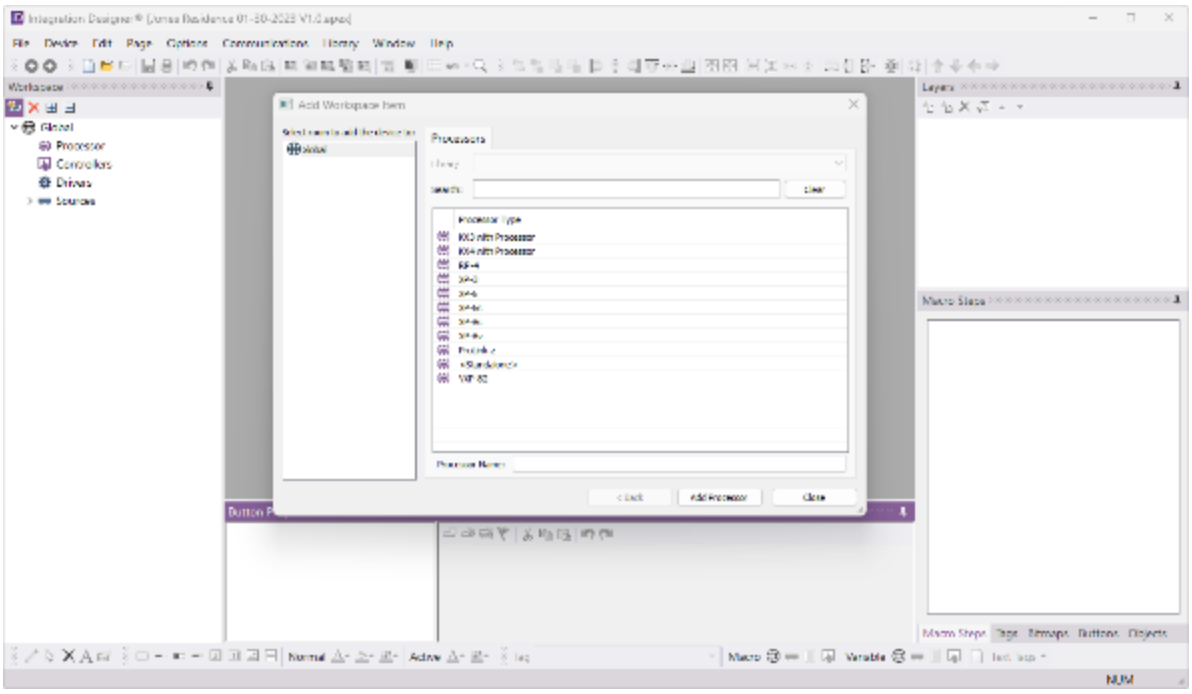
2. Navigate to an appropriate directory and enter a logical file name to help you identify your client program files.



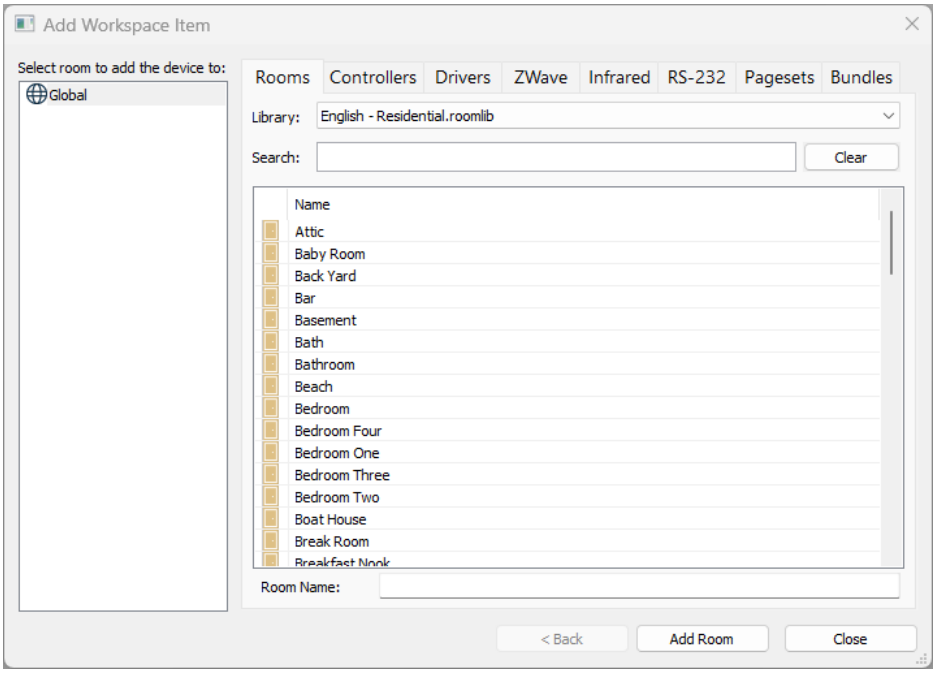
Once you establish a new file name, the software will create a global area in the Workspace with subcategories for the processor, control, drivers, and source devices. The **Add Workspace Item** will display and guide you through adding all the essential building blocks of your system.

1. Adding a Main Processor

Select the main processor for your project. If your project does not include an RTI processor, select the **<Standalone>** option from the List of processors.



Once the main processor is added to your project, additional options will be displayed in the **Add Workspace Item**.

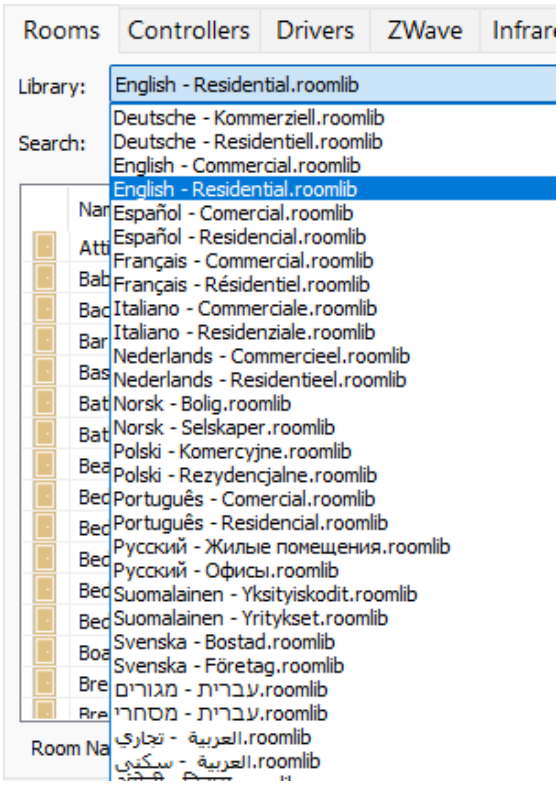


2. Adding Rooms

Add the Rooms that comprise the project. A room is an area where control interactions are made. There are three ways to enter rooms:

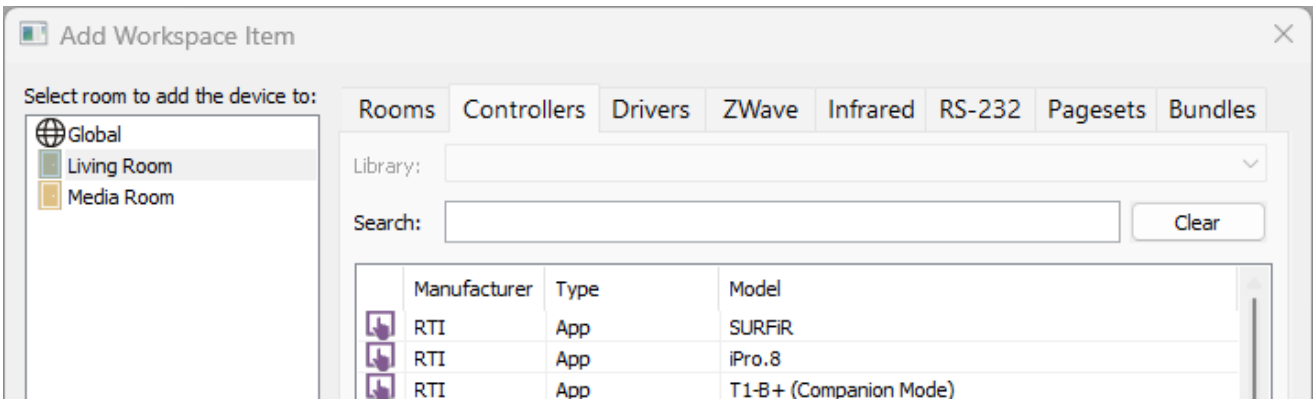
a. Select a room from an alphabetized list by double-clicking the room name.
b. Search for a room in the List of alphabetized rooms by typing text in the search box.
c. Enter a customized room name by typing text in the **Room Name** field.

Selecting the applicable library from the library dropdown menu, you may set the default room view to residential or commercial room names. Additional languages are supported. Once you make a new selection, the software will default to that choice in future projects.



3. Adding RTI Controllers

After rooms are entered, select the controller tab and add the controllers to your project.



Integration Designer software features current and legacy RTI controllers. When adding a controller, you must select the area where the controller lives. It may be placed in the Global area or assigned to a local room.

Global Area: Controllers are added to the global area if they need to control two or more rooms in the project. In situations with a single room in the project, place the controller in the global area if you expect the need for multiroom control should the client want to expand the system.

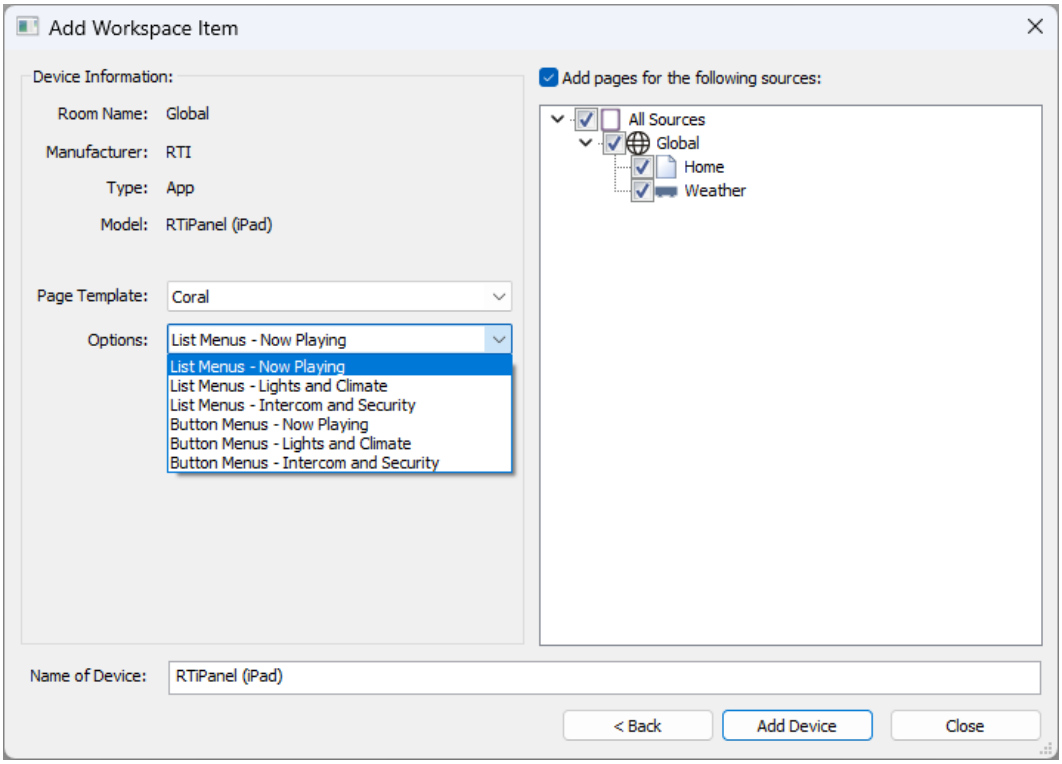
Local Room: Controllers are typically placed in local rooms when designed to control devices in that Room and any shared devices in the home. They are not designed for multi-room access. A controller must be placed in the global area if it is needed to control the attributes of two or more local rooms.

Pro Tip: RTIPanel devices that typically roam on Wi-Fi are ideal for multiroom control and should almost always be placed in the global area. Handheld controllers are ideal for local room control.

There are two ways to enter controllers:

a. Scroll through the List of controllers and select the controller. Be sure to select the area where the controller is placed depending on the project's needs.
b. Search for a controller by entering text in the search box. When selecting a controller, select the area where the controller is placed depending on the project's needs.

Select the controller type, then select the area to the left to place the controller. Select **Next**.



The Add Workspace Item window will now display some options that can be selected when adding the controller to the project.

Device Information: The device information confirms the Model, Type, Manufacturer, and Room Name where the controller will be placed. Verify that the information is correct before selecting **Add Device**.

Page Template: The template selected for this project. Additional choices may be provided in the dropdown menu.

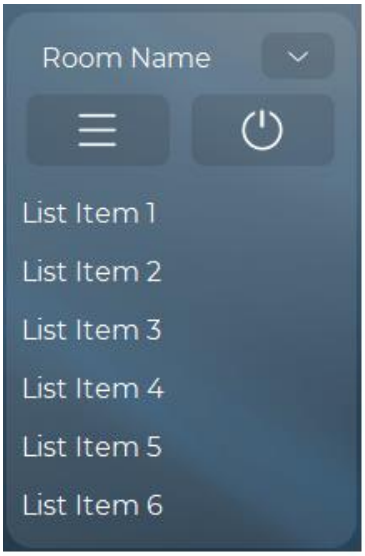
Options: A menu type such as List Menus or Button Menus can be selected, along with a dashboard feature if available for that controller. Note: Smaller controllers do not have dashboard selections.

See the figure below on the list menu and button menu types.

Name of Device: The default name of the device you are adding. It is best to personalize the names of your controllers. For example, "Joe's iPad" or "Living Room T4x" make them easy to identify.

Add Pages: Integration Designer will add a default home page and a weather page if desired. You may check or uncheck this option based on your preference. You can always add or delete pages later if you unintentionally forget to add or add a page.

After assigning these options to your controller, select **Add Device** to apply your controller. Repeat this step by adding the controllers to your project.



List Menu

Button Menu

Pro Tip: This figure to the left illustrates list menus and button menus. List menus are auto-populated with text selections. Rooms and sources are automatically generated.

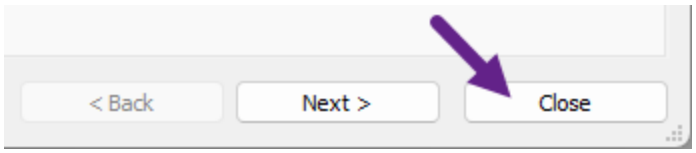
Button menus may be customized by changing the text and icon on the button and placing activity tags on each button.

4. Add Expansion Devices

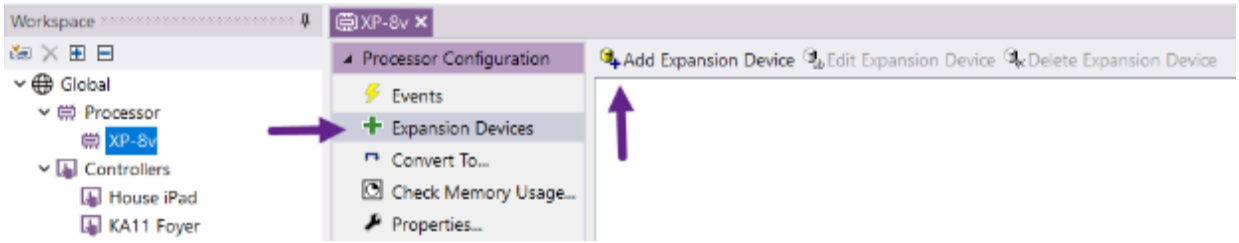
Once the main processor, rooms, and controllers have been added to your project, you can add the source devices using the **Add Workspace Item**. Before doing so, you must add any RTI expansion devices needed in your project.

If your project has no expansion devices, you may skip Step 5, "Adding Source Devices to the project file."

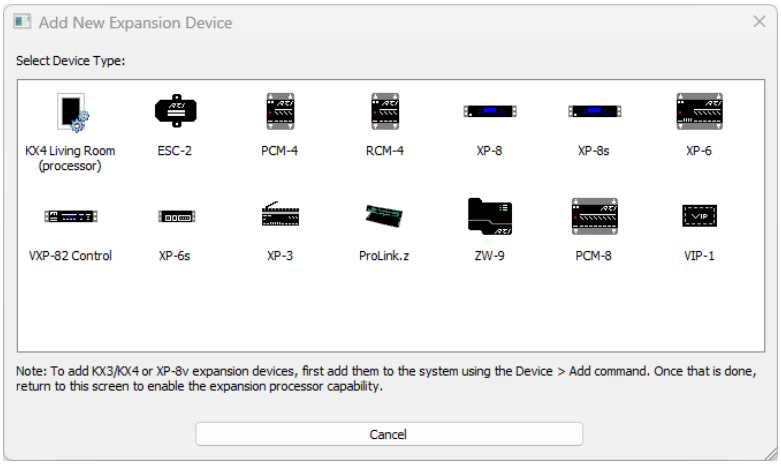
Select **Close** on the **Add Workspace Item**.



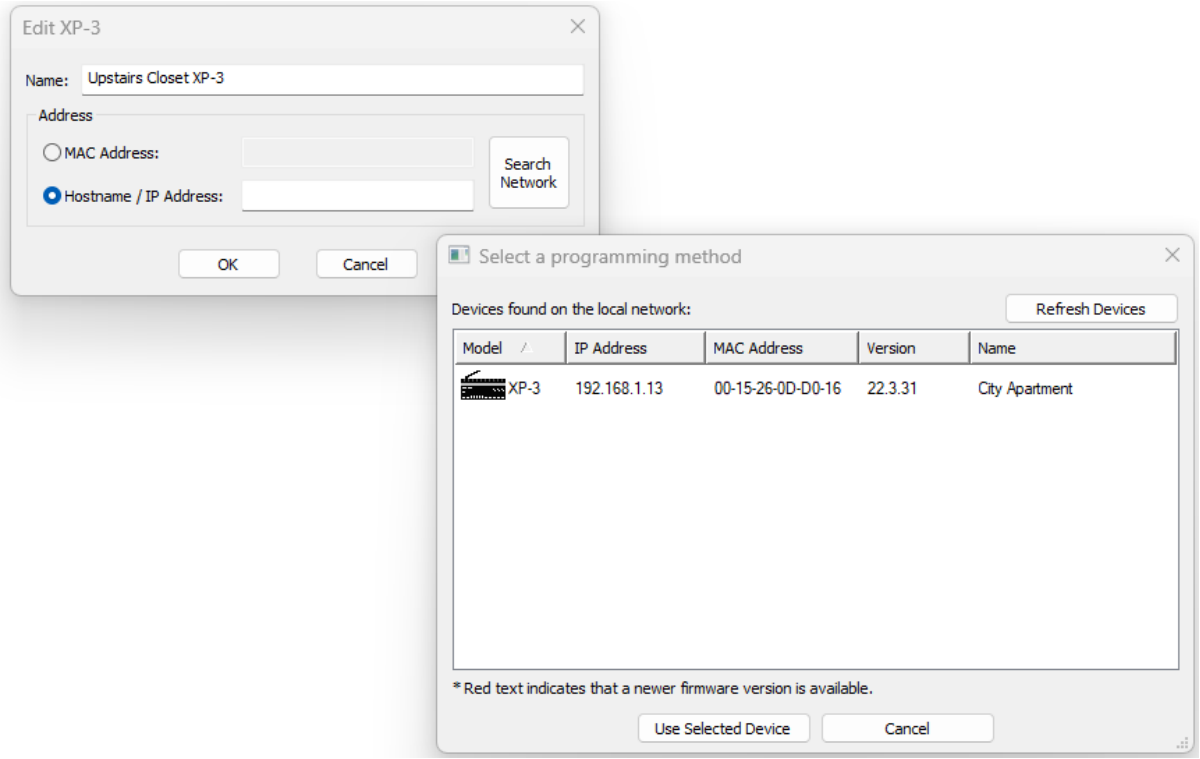
Select the main processor in the Workspace, then select **Expansion Devices** in the Process Configuration Section.



A list of expansion devices will be displayed in a new window. Expansion devices can communicate with the Main XP processor by connecting the device to the network and discovering it here.



Select the expansion device you wish to add to your project. A configuration window will display the following:

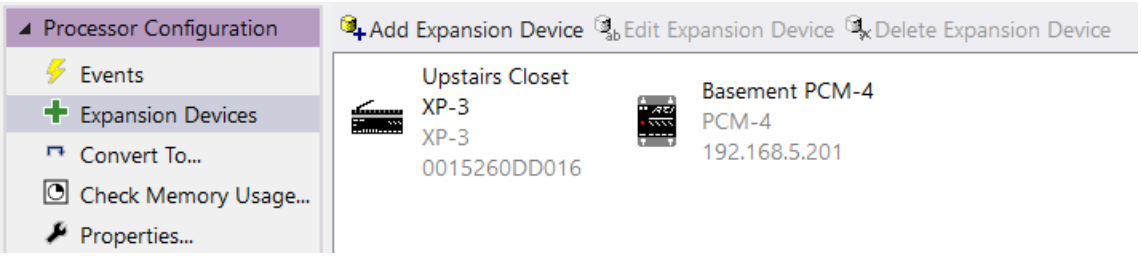


Name: Give the expansion device a descriptive name designation.

Address: Discover and communicate with your processor via MAC or IP Address. You may manually enter your device's MAC address or IP address, but selecting **Search Network** and discovering the device is the most reliable way of populating correct information.

Select the **Search Network** button, locate the device, and select "**Use Selected Device**."

Repeat these steps to add additional expansion devices to your project.

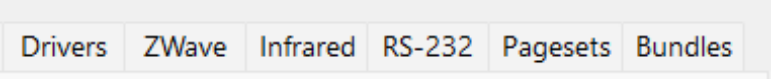


You may edit the properties of any expansion device by selecting the device and **Edit Expansion Device**. You can also delete the device by selecting **Delete Expansion Device**.

Return to the Add Workspace Item by selecting the Add Workspace Item icon in the Workspace toolbar.

5. Adding Source Devices

The next seven tabs will add source devices to the Workspace. Sources must be added using these tabs in Integration Designer. At this point, no specific order must be followed.



Drivers: Drivers are control programs developed by a driver developer using a manufacturer SDK. In most cases, drivers offer several advantages, such as multiple control options, two-way feedback, a more comprehensive command list, and the ability to test driver variables due to processor polling. It is important to understand the equipment you specify and read the driver information included for vital information.

Z-Wave: If you use Z-Wave devices, add a ZW9 expansion device to include and import Z-Wave control devices into the software. A training video on how to use Z-Wave manager can be accessed after adding a ZW9 Z-Wave gateway.

Infrared: RTI includes a vast, comprehensive, consolidated, and accurate infrared third-party database with the software. The IR database is updated often and available through software releases. IR code commands can be previewed before adding an IR device to the project. In addition to the IR code databases, RTI installers can maintain their libraries using the included IR Library Manager. IR Codes can be learned, or data can be entered directly into the IR Library Manager.

RS-232: RTI provides a library of one-way RS-232 devices updated in software updates. In addition to the RS-232 one-way database, RTI installers can maintain their own RS232 devices using the RS-232 Library Manager, which is included in the software.

Pagesets: Pagesets are source devices not backed by an underlying driver, IR, or RS-232 library. RTI Installers can add a pageset when a blank source is needed, such as a setting, scenes, or welcome screen. There are several pageset options available to help build your user interfaces.

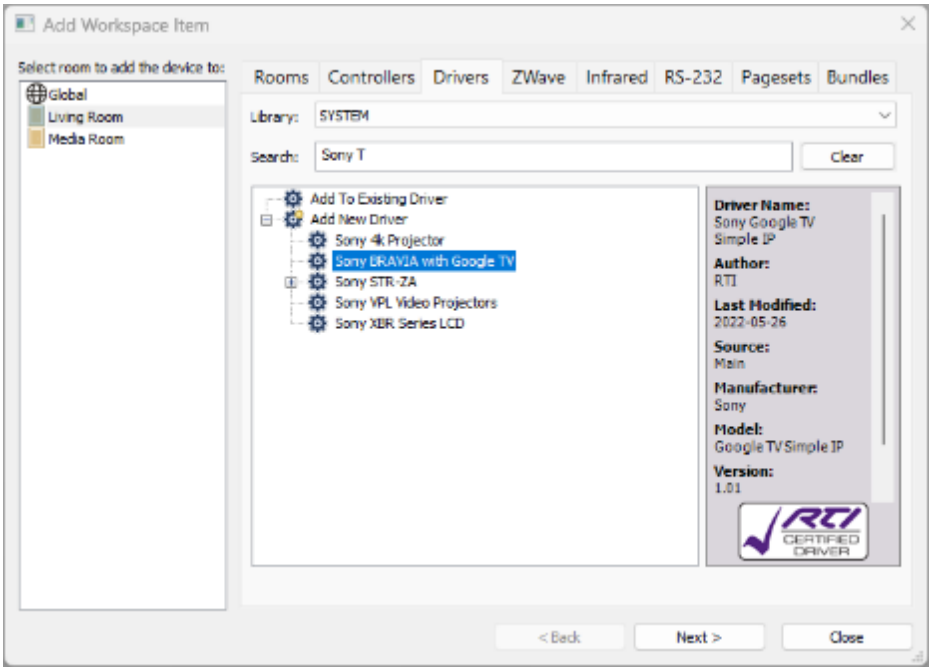
Bundles: Bundles are source devices previously exported from Integration Designer. Bundles preserve custom graphics, UI arrangement, programming, and device code set. RTI Installers can save time by exporting and importing bundles so that work is not duplicated.

ADDING A LOCAL DISPLAY DRIVER SOURCE DEVICE

When adding sources to your project, start with the local devices in the first room of the project. Typically, video displays and devices responsible for audio are added to a local room. While there are always exceptions, it is more important to familiarize yourself with the concepts and apply them in your projects.

Add a video display driver to the first room in the project.

1. Select the **Drivers** tab in the **Add Workspace Item** window.

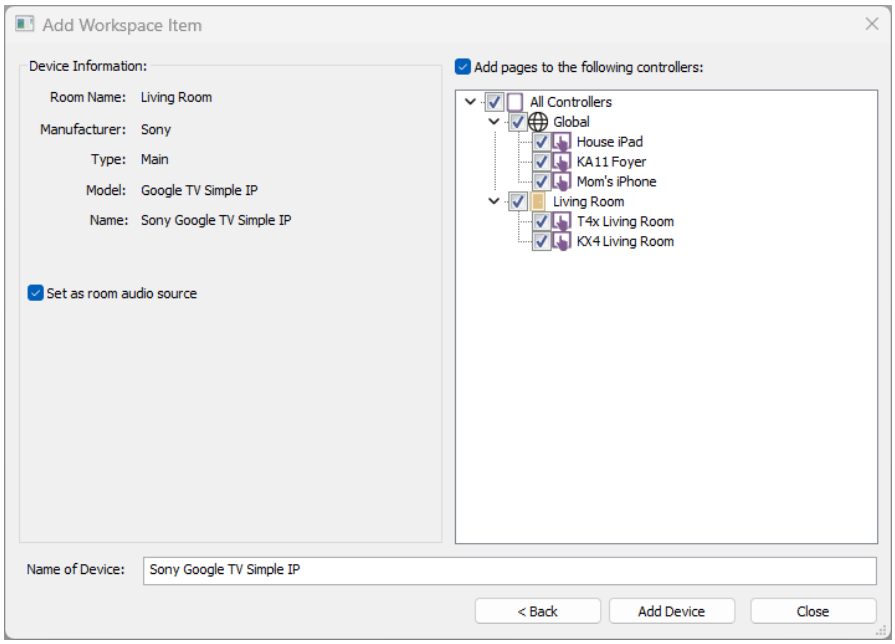


2. In the **search** box, begin typing in the brand and device type until you see the device needed.

3. Select the device, then select the room to add the device to on the left side.

   The criteria required to determine if the device should be placed in a local room or the global area is as follows:
   - Does the device exist in the local room and require control?
   - Do controllers in other rooms require a user interface for this device? If yes, the device belongs in the global area. If not, the device is placed in the local room.

4. Add the device to the local room as it is not shared in other rooms. Select **Next** to continue.



5. Confirm your selection and configure all settings.

   Device Information – Confirm the device information and the room where the device is located.

   Set as Room Audio Source – Checking this option will ensure the volume will be programmed using the device's volume commands when controlling the room on applicable controllers. In this example, checking this option will ensure the TV volume is used for all activities in the Living Room.

   Name of Device – Give the device a simple descriptive name. For example, "Living Room TV."

   Add Pages to the Following Controllers – Integration Designer provides a pre-programmed user interface for any checked controllers. Since the device was added to the local room, options are available to add user interfaces to any global (multiroom) controllers and local controllers in the room.

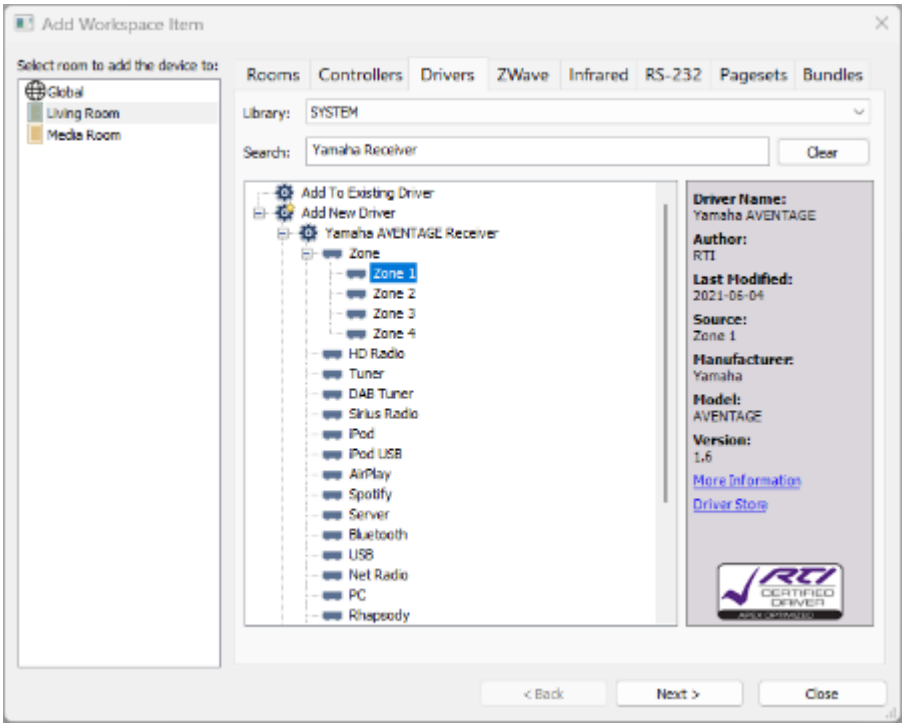
   Once the options have been configured, select **Add Device** to continue.

Pro Tip: It is sometimes necessary to give client-controlled user interfaces for every device. Consider the following – Will the interface cause problems for the client? Do the features of this device overlap with other devices we are providing? Does the client understand how to use the settings or control options for the device? Avoid service calls and provide only required user interfaces.

ADDING A LOCAL AUDIO DRIVER SOURCE DEVICE

Add an audio receiver to the first room in your project.

1. Select the **Drivers** tab in the **Add Workspace Item** window.



On the right side of the driver screen, you can view valuable information on the driver.

Select **More Information** to access the driver integration notes.

You can also jump to the RTI Driver store by selecting the **Driver Store** link.

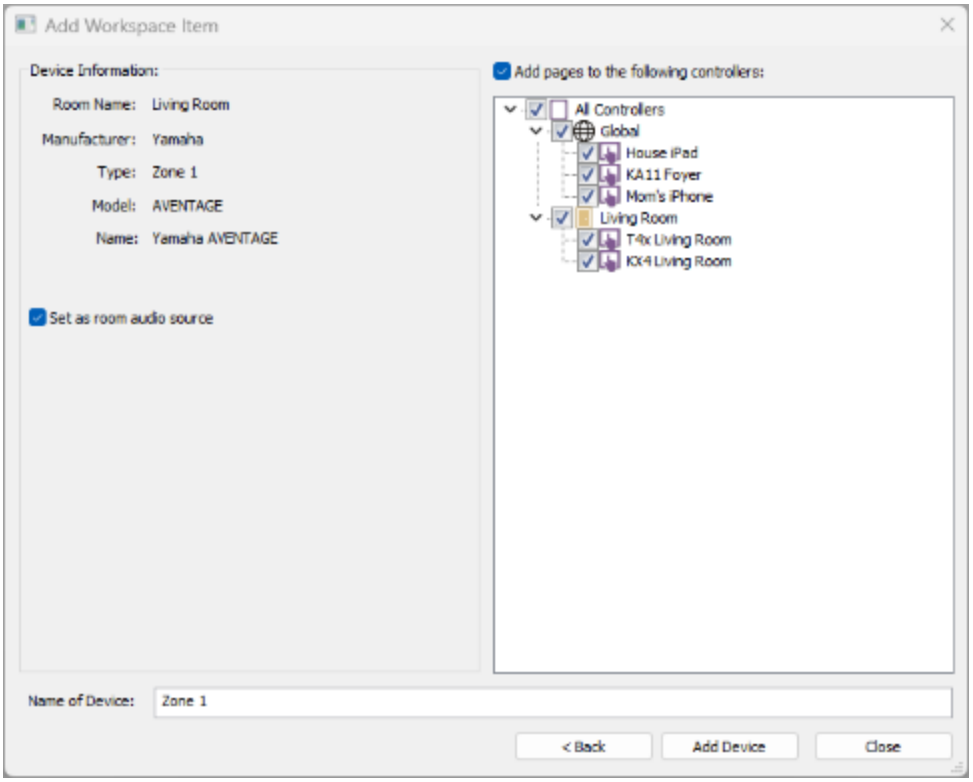
2. In the search box, begin typing in the brand and device type until you see the device needed.

3. You will see several options if you add a driver for a device with multiple zones and music services.

4. You should initially always add Zone # to your working room. Once a zone has been added, you can add subsequent zones using the same driver instance. For example, if Zone 2 is in the Media Room, you may use the same driver for additional receiver zones.

5. Select Zone 1, then select the room to add the device on the left side. Refer to the criteria to determine if a device should be added to a local room or the shared global area.

   Add the device to the local room as it is not shared in other rooms. Select **Next** to continue.



6. Confirm your selection and configure all settings.

   Device Information – Confirm the device information and the room where the device is placed.

   Set as Room Audio Source – Checking this option will ensure the volume will be programmed using the device's volume commands when controlling the room on applicable controllers. In this example, checking this option will ensure the AVR volume is used for all activities in the Living Room.

   Name of Device – Give the device a simple descriptive name, for example, "Living Room AVR."

Add Pages to the Following Controllers – Integration Designer provides a pre-programmed user interface for any checked controllers. Since the device was added to the local room, options are available to add user interfaces to any global (multiroom) controllers and local controllers in the room.

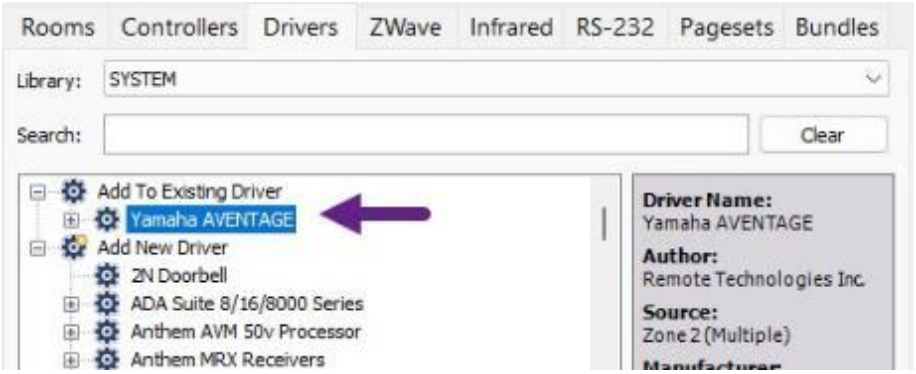
Once the options have been configured, select **Add Device** to continue.

Pro Tip: RTI installers should understand the equipment they control. Carefully read driver notes and test devices off-site before delivering a control experience to clients. Doing so will ensure that there are no surprises or wasted time onsite.

ADDING A LOCAL AUDIO DRIVER SOURCE DEVICE WITH MULTIPLE ZONES

If you have additional rooms utilizing another zone for the same device, add the other zones to the applicable rooms.

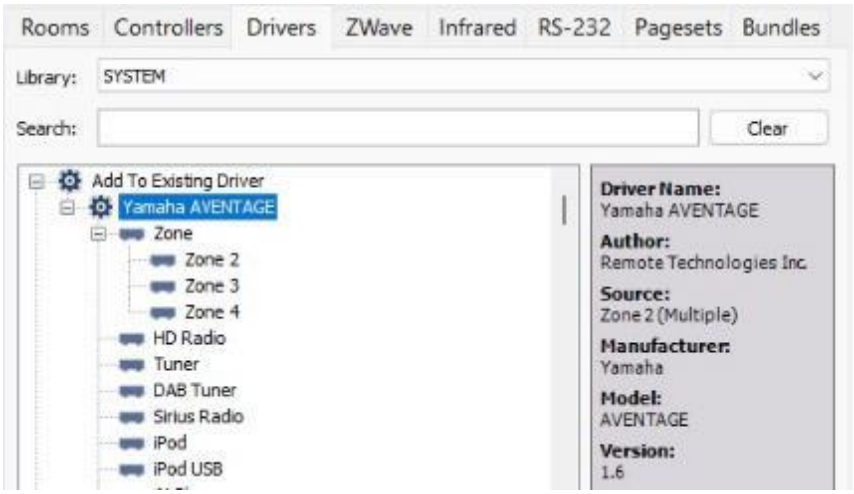
Notice that an existing driver entry for the AVR is added in the last step.



The term "Add to Existing Driver" indicates that a part of the driver is being used in the existing project file.

1. Select the + to the left of the existing driver.

You will notice that the zone 1 entry has been used in the project for another room.



2. Add the subsequent zones to the room they are designated.

Note that you may add music activities if you require programmed user interfaces. Always add the zone information first before adding any other driver services. If you have another AVR, you must add a new driver. Do not use existing drivers for additional zones.

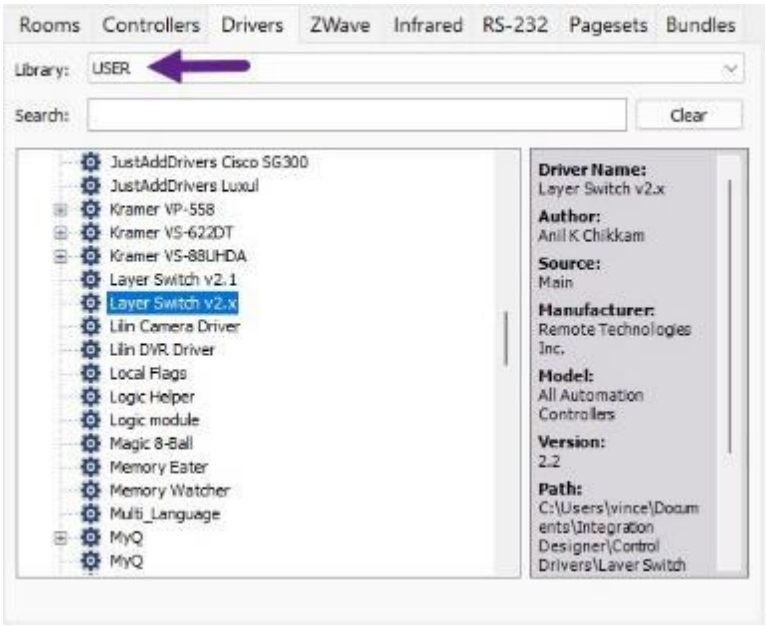
ADDING A THIRD-PARTY DRIVER

Certain drivers created by a third party may be unavailable in the Driver store. Drivers may be placed in the Documents/Integration Designer/Drivers area and accessed in the software. Third-party drivers may be downloaded on manufacturer websites and driver development stores and developed in-house using the RTI software development kit.

1. Select the **Driver** tab in the **Add Workspace Item** menu.

2. Select the dropdown menu and change it from **SYSTEM** to **USER**.

3. Select the driver from the driver list.

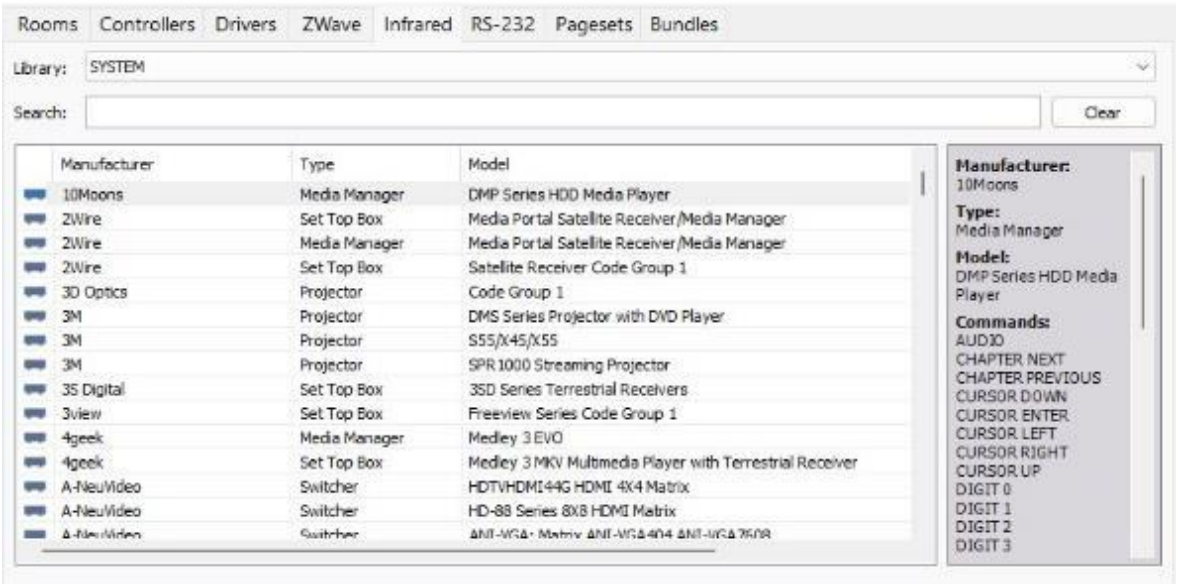


Set the driver library to **SYSTEM** after adding the user-based drivers.

ADDING AN IR SOURCE DEVICE

Integration Designer features a built-in IR database that is comprehensive, consolidated, consistent, and accurate.

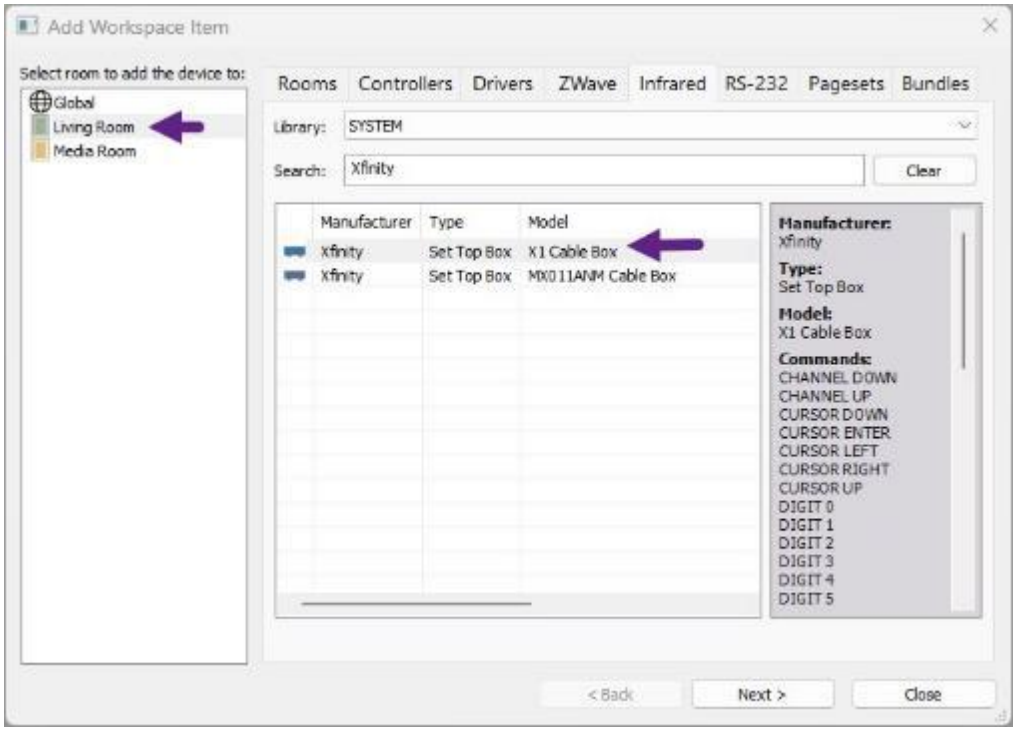
1. Select the **Infrared** tab from the **Add Workspace Item** menu.



2. Ensure the library is set to **SYSTEM**. It will ensure that the integrated database is accessible. You may use personal infrared libraries by setting the library to **USER**.

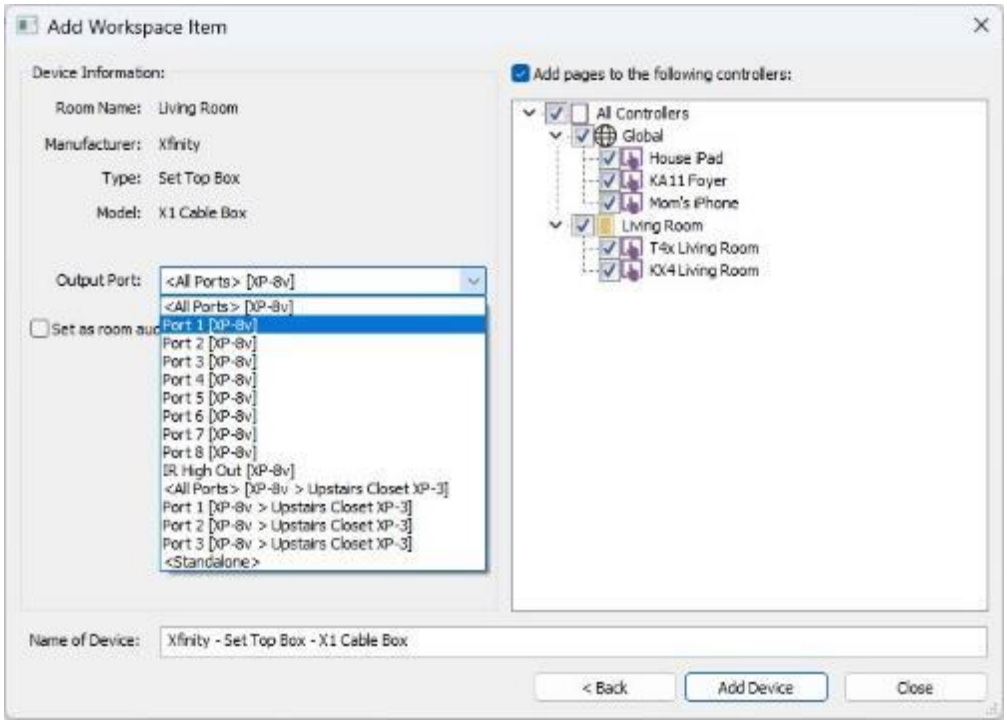
3. Search for the infrared device to add to the project file.

4. Select the source device to add, then select the room or the global area if the device is shared. If the source device requires a pre-programmed user interface on controllers in other local rooms, it belongs in the global area.



A preview window on the right side features a preview of the infrared codes, device information, and any noteworthy information, such as workarounds for discrete codes.

5. Select **Next** to continue.



6. Verify the **Device Information** to ensure the correct device and room is selected.

7. Select the processor or expansion device **Output Port** to route the infrared signal. Once the port is set, all commands will be routed to that port.

   *Set the device to **standalone** if the device communicates directly from the controller via line-of-sight infrared.*

8. Rename the **Name of the Device** to something logical and easy to understand. Renaming the device will also designate a name to the infrared port on the processor.

   Pro Tip: It is a good habit to name source devices with a room name attached. For example, “Living Room Cable” will make identifying your devices easier when programming macros later.

9. Ensure the pages are selected for the global and local room controllers. You may deselect any controller if you want the controller to bypass the programmed source interface.

Select Add Device to add the source device to the workspace.

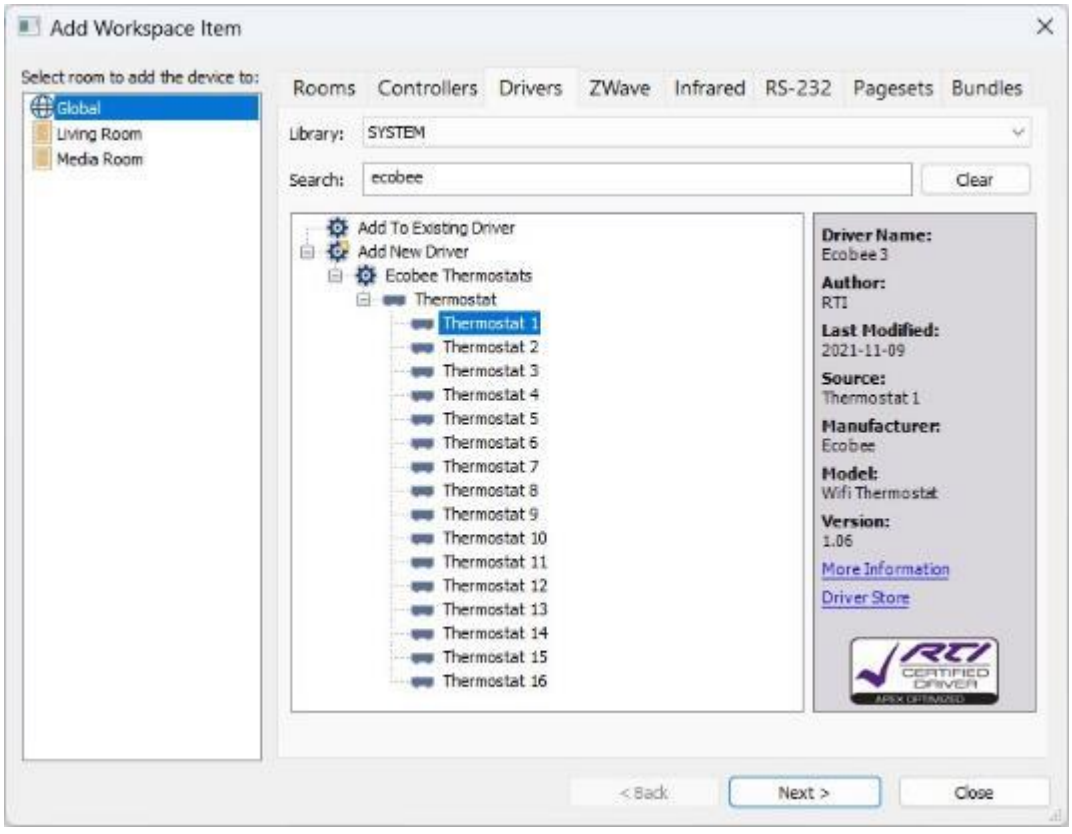
Note that the SYSTEM library can be changed to USER for personal IR libraries.

ADDING A SOURCE DEVICE TO GLOBAL

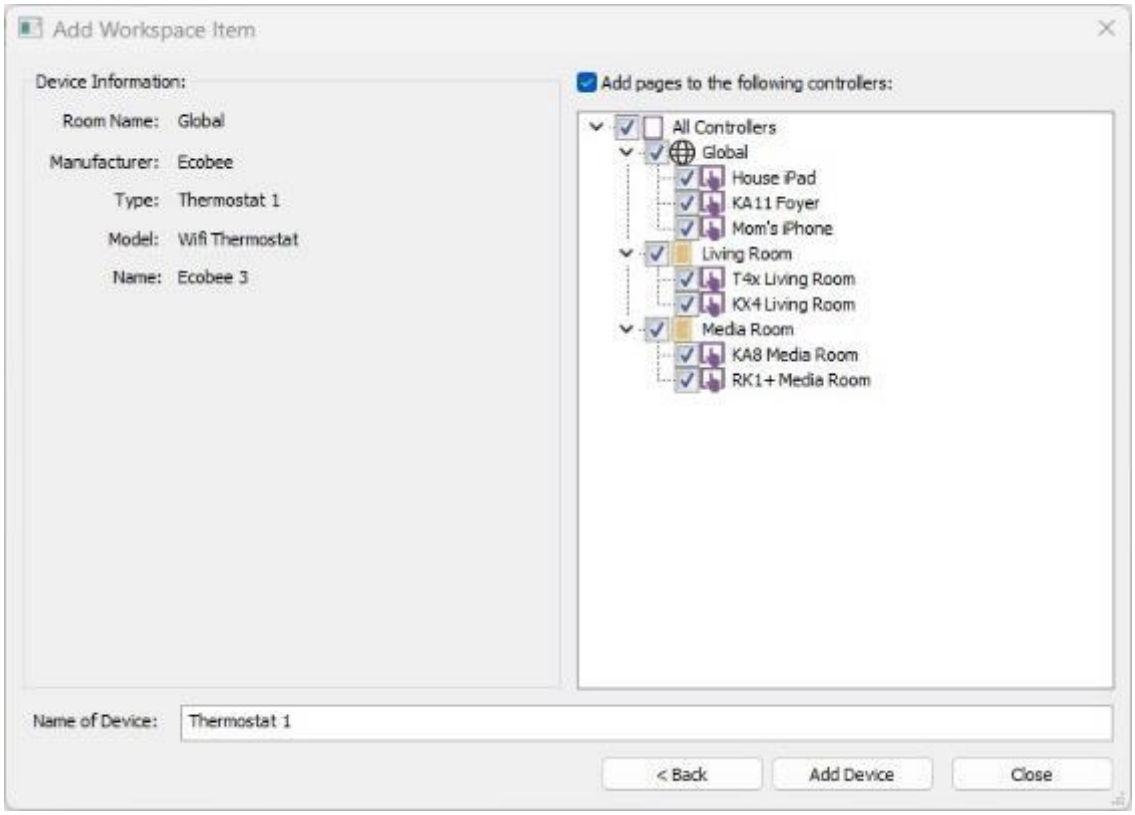
Integration Designer features a global area automatically built into each project. The global area is where multiroom controllers are placed, and source devices are shared in the project file in two or more local rooms. Typically, you will place devices such as lighting, security, climate, and shard sources in the global section of the workspace.

1. Select the driver tab and enter text in the search box to locate the shared device. In this example, a thermostat will be selected.

2. Select the sequential thermostat and add it to the global area.



3. Optionally, you may add thermostats to the global area if the project has multiple climate controls. Be sure to use the **Add to Existing Driver** option for additional controls.



4. Verify the information in the upper left corner to ensure you are adding the correct device to the global area. You may select **Back** if you need to adjust.

5. For the **Name of the Device**, give the device a logical and descriptive name.

6. Ensure the controllers that require a pre-programmed user interface and full control of the device are selected on the right side.

   Because you are adding the device to the global area, you can add the user interfaces to all the controllers in the project. You may deselect controllers that do not require a user interface, such as a guest room or child's bedroom controller.

7. Select **Add Device** when complete.

> **Pro Tip**: Always ensure that devices are placed properly based on the project's needs. There are advantages to placing devices in local rooms when local control is needed and the global area where shared control is required. It will make programming and utilizing the features of Integration Designer much easier.

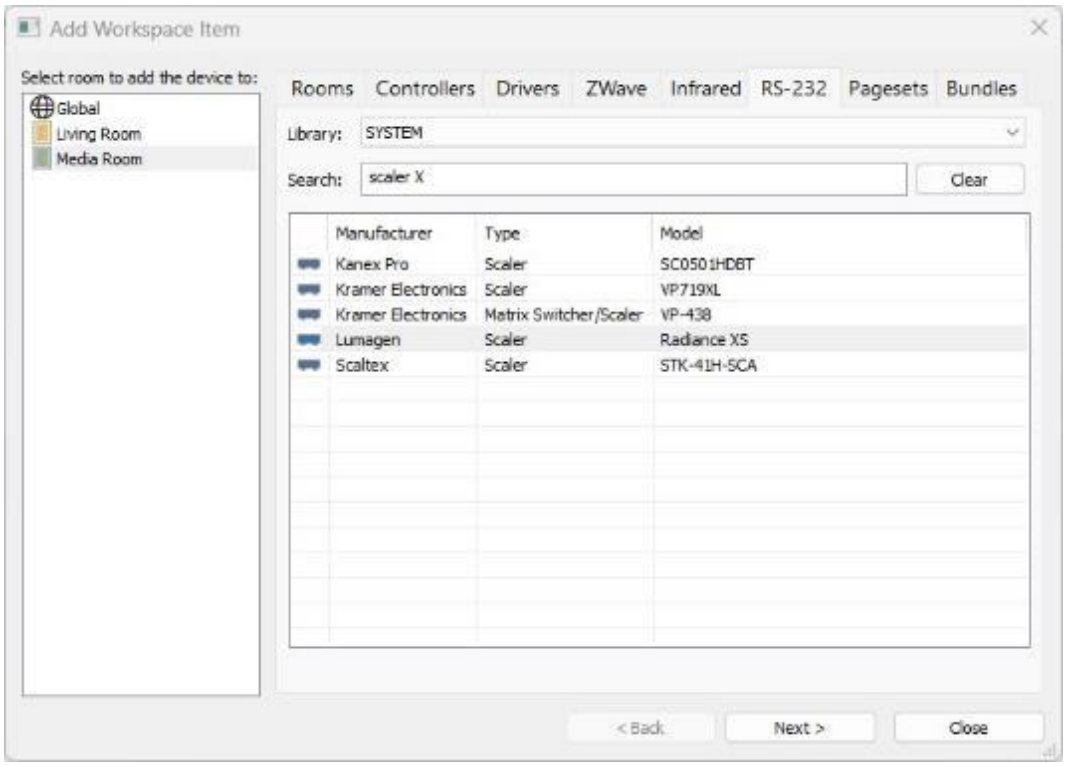
ADDING AN RS-232 SOURCE DEVICE

Although it is common for drivers to allow for RS-232 control, there are cases when a driver is unavailable. Integration Designer features an integrated RS-232 database for controlling devices. These devices offer one-way control and do not create pre-programmed user interfaces. However, they can offer many advantages of infrared devices and a robust control option.

1. Select the RS-232 tab in the **Add Workspace Item** menu.

2. Ensure the **SYSTEM** library is selected. You can create, manage, and use personal RS-232 libraries by setting the library to **USER**.

3. Enter text in the **Search** area to find the device required.

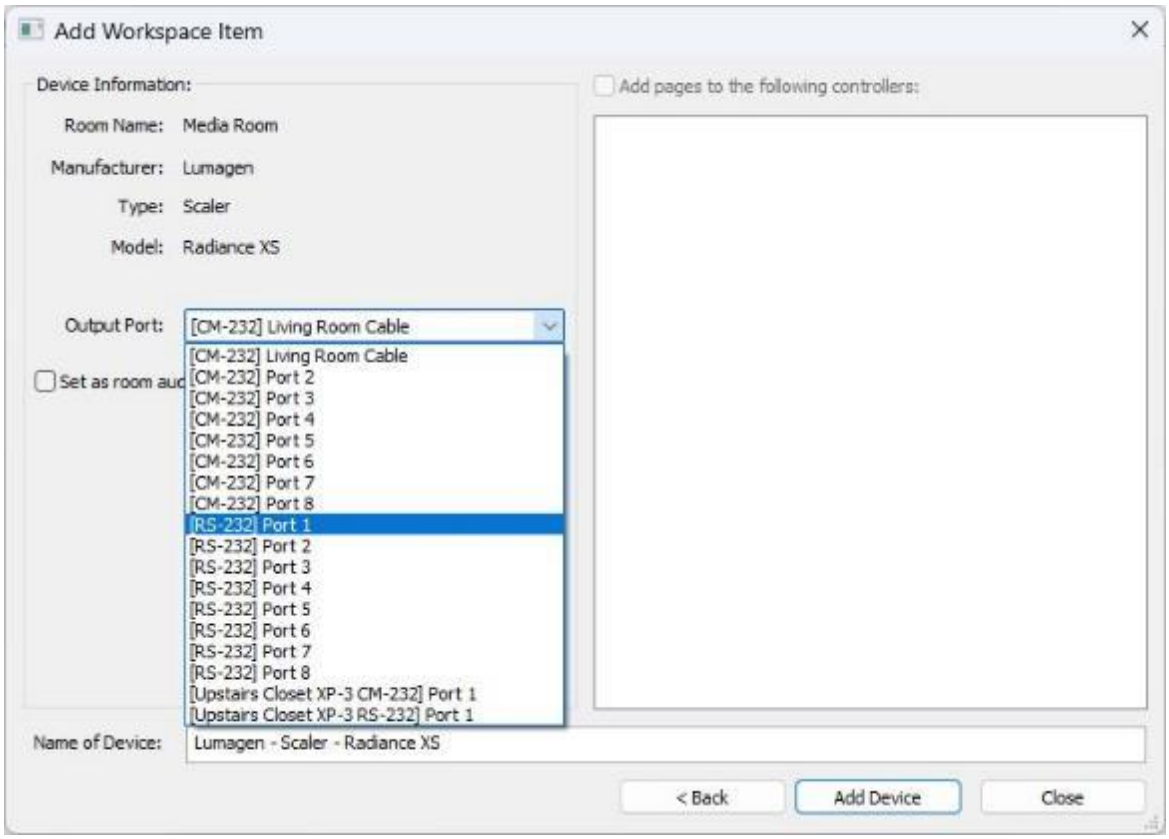


4. Select the device and the room or global, depending on the project requirements.

5. Select Next to continue.

6. Verify the device information on the upper right and ensure the room is correct.

7. Select the MPIO processor MPIO, RS-232, or expansion device RS-232 port. Note that you will require a CM-232 adapter when using multi-purpose input-output ports.



8. You may check **Set as Room Audio Source** if you want the volume for the device on all source pages for this room.

9. Give your device a logical and descriptive name.

10. Since no pages will be added, these must be added in a future step if required.

Select **Add Device** to continue.

ADDING A PAGESET

Pagesets are source devices without an infrared, driver, or RS-232 command set. They can be used for many different purposes, such as the following:

- A generic source such as a settings or scenes page
- A page to warn the client is processing a macro and to wait patiently.

- It can be used as a source for an activity that does not require control, such as an HDMI port for multiple game consoles. Different control commands and feedback could be programmed on a pageset.

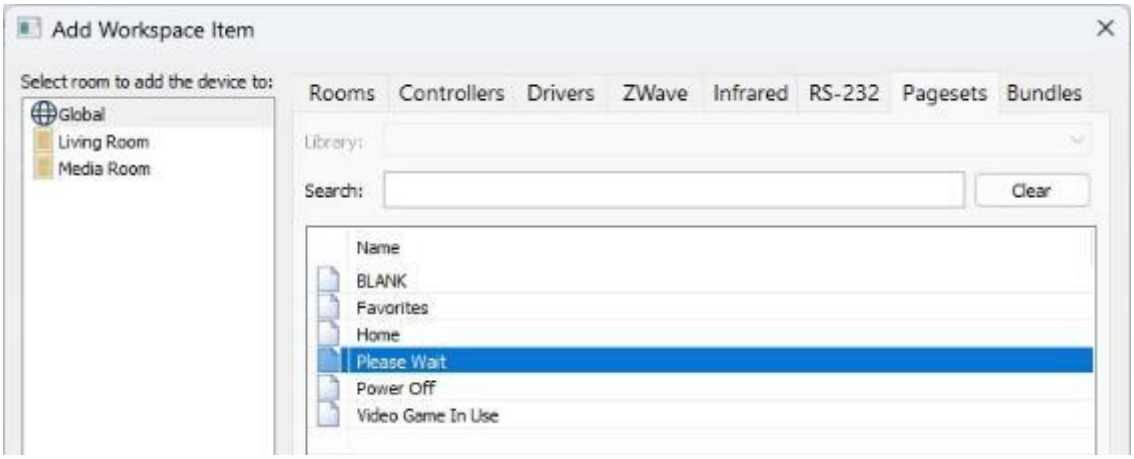
- It may also be used as a landing page for an activity that does not have an underlying device.

In this example, we add a wait page for the controllers in the project.

1. Select **Pagesets** from the Add Workspace Item menu. You will notice several examples of pagesets, including a blank source.

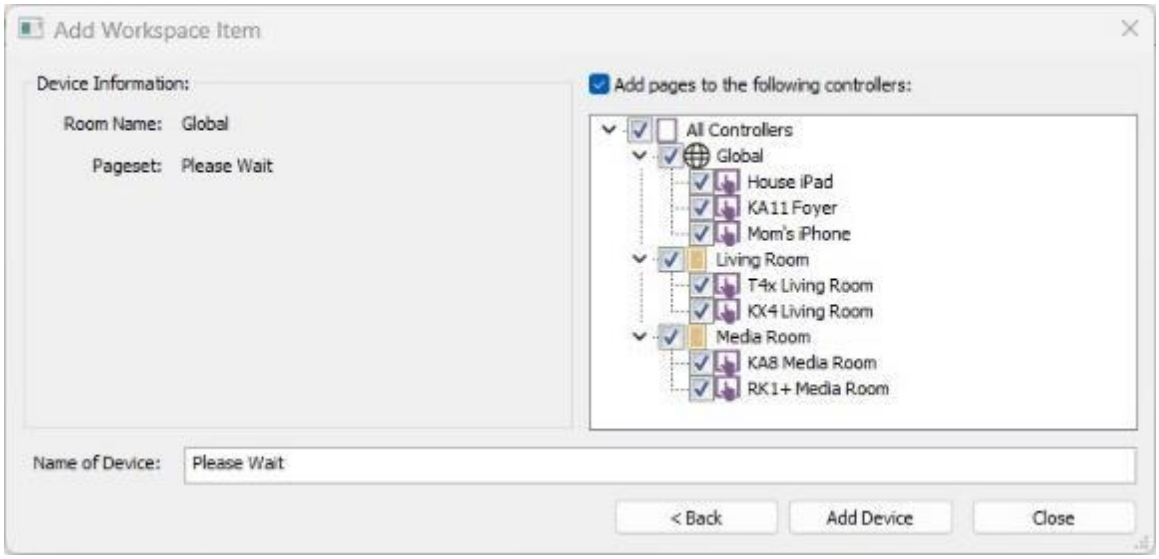
2. Select "**Please Wait"** from the List of pagesets.

3. Since this page is required on all controllers, it is added to the global area. Select the pageset and the global area on the room list on the left side.



4. Confirm the controllers that require the pageset in the List.

5. Select a different name if desired.



Select **Add Device** to continue.

Integration Designer handles pagesets in the same manner as a source device.

ADDING A SOURCE AS A BUNDLE

Many RTI installers custom program user interfaces and change the programming in many cases depending on the device features, programming style, and client needs. Once a source device has been programmed and graphics altered, you may export the source as a bundle to a personal device library available for importing.

Bundles preserve the following:

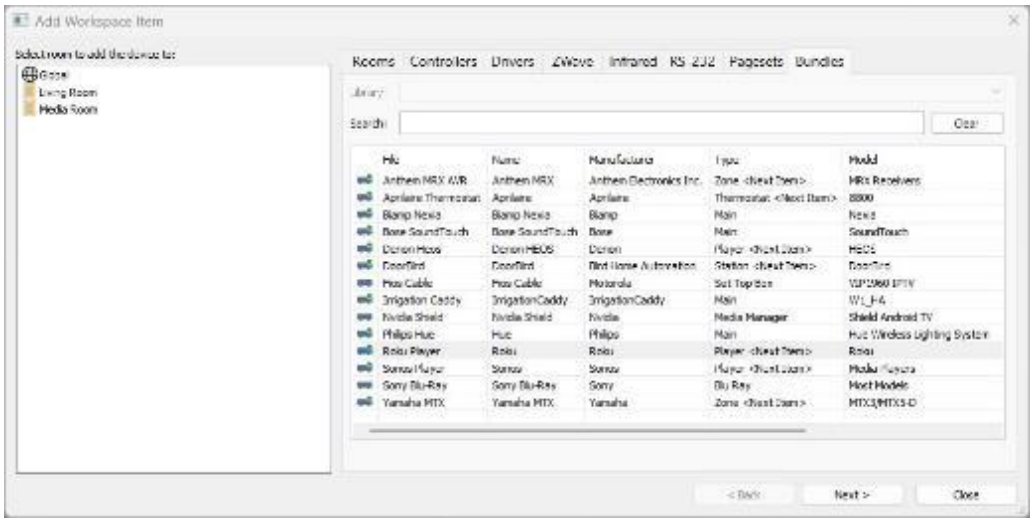
- The entire driver or infrared command library.
- Any graphic changes for a specific source device.
- Any programming related to that source device.
- Any graphical framework, including layers and toolbars.

Bundles will be stored in the **Templates** directory as defined in the **Options/Set Directories** area.

Bundles are an alternative way of adding a driver or infrared database, except that prior graphical and programming work is preserved.

Pro Tip: Use bundles to save time by exporting all refined graphical and programming development related to a specific source device. Bundles are particularly useful if you create a fully functional, well-developed user interface for your clients.

1. Select the **Bundles** tab in the **Add Workspace Item** menu.

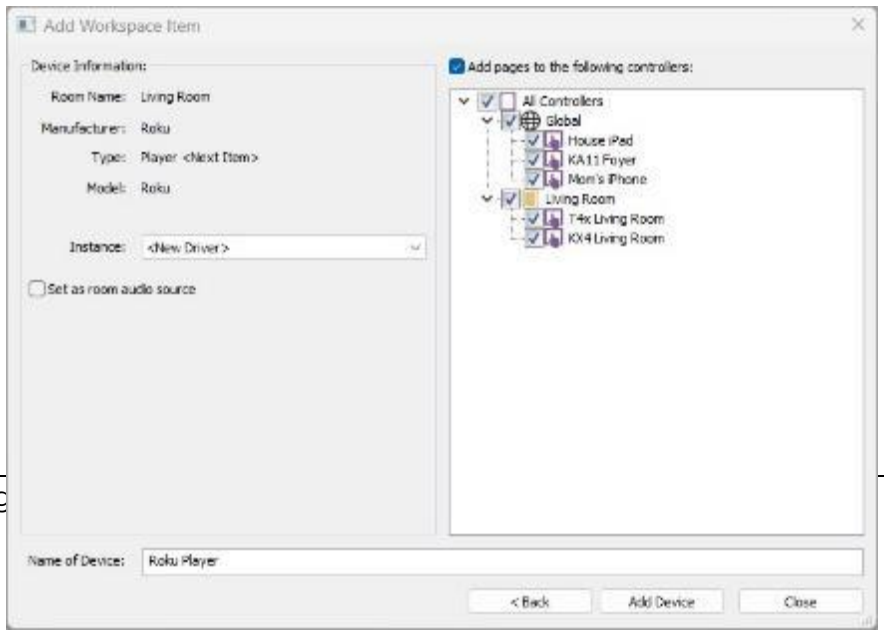


   A library of devices that have been previously exported will be displayed. Adding a bundle will have the same procedure as adding a driver or infrared source device.

2. Search for your device by entering text criteria in the search box.

3. Select and highlight the bundle you want to add to the project file.

4. Select an area (global or local) depending on the device's use.



5. Select **Next** and configure any options.

Follow the same procedure for adding drivers or infrared devices depending on what type of device the bundle originated.

Configuring Properties

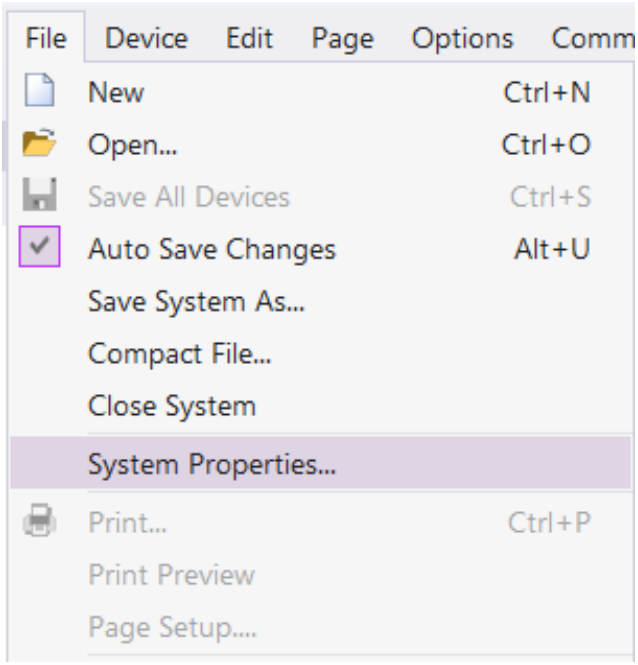
Items in the workspace, including the main processor, controllers, and drivers, require quick configuration settings. You will also need to configure the overall system properties for your project.

It is good practice to review the settings for your devices if you are utilizing only a few options.

CONFIGURING SYSTEM PROPERTIES

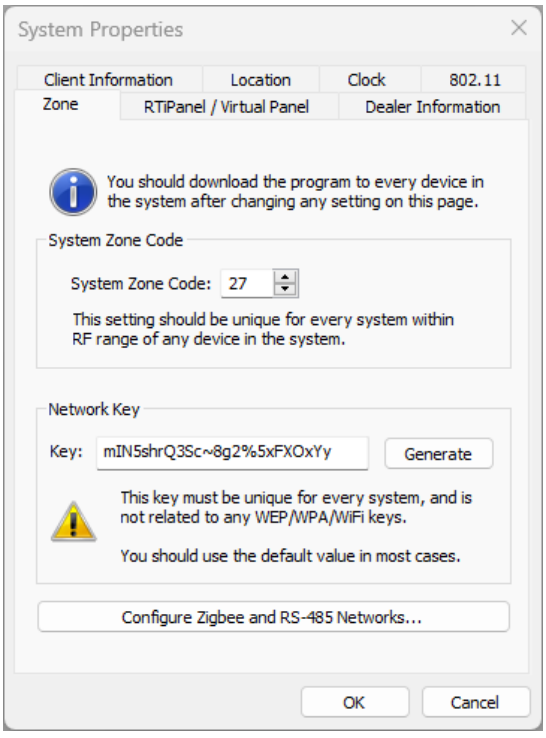
Configuration of the system properties is required for every project. Many of the preferences created in the system properties will populate on most controllers in the project file. If you are using RTiPanel devices, this is where licenses are attached to your project.

Select **File** from the toolbar, then select **System Properties**.



The software will save periodically if the Auto Save Changes feature contains a checkmark. Select or toggle this feature on or off or use Alt + U.

Select each tab and fill in the required information.

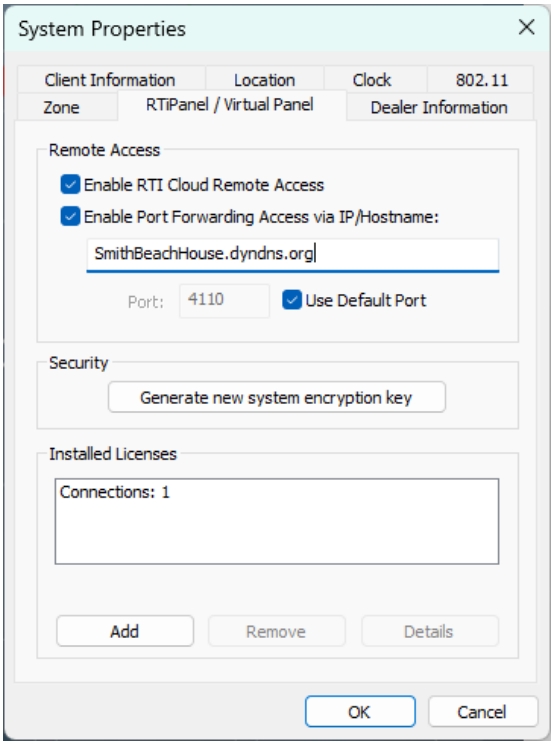


System Zone Code – Change the code to any value from 1 to 255. Doing so will prevent collusion with other RTI control systems in the area.

Network Key – The network key should not be changed unless directed by tech support.

Configure Zigbee and RS-485 Networks – This option is used when you use expansion

processors requiring multiple RS-485 or Zigbee networks.



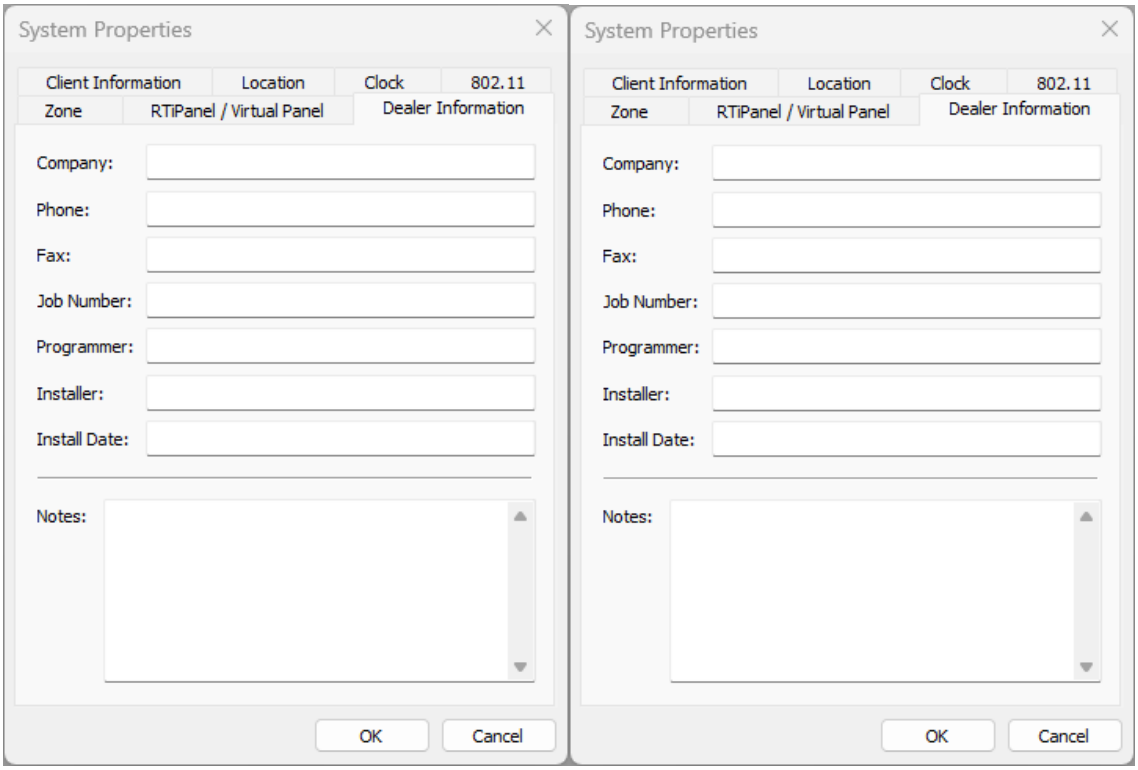
Remote Access

Enable RTI Cloud Remote Access- Tick this option if remote access via RTI Cloud is desired.

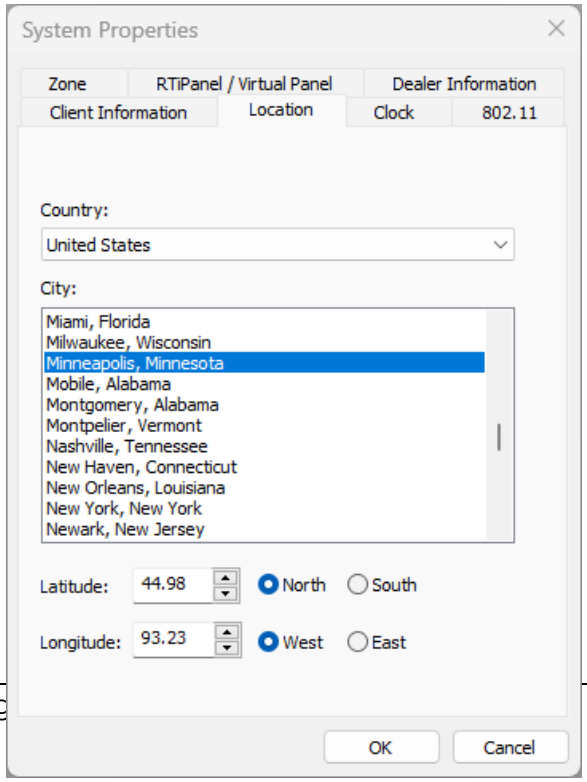
Enable Port Forwarding Access via IP/Hostname: If RTI Cloud remote access is not used and remote access is required, tick this option and enter an external IP or hostname.

External IP/Hostname – If using remote access, enter a **static IP** or **DNS** address. Note that the default port (4110) must be forwarded to the IP address of the processor in the router settings, but this can be changed to another port by unchecking the default checkbox and entering a new port number.

Licenses – Select **Add** to upload your RTiPanel license file(s) to the project.



The **Client Information** and **Dealer Information** tabs are optional. Entering contact information, user accounts and special reminders about the system could save you time.



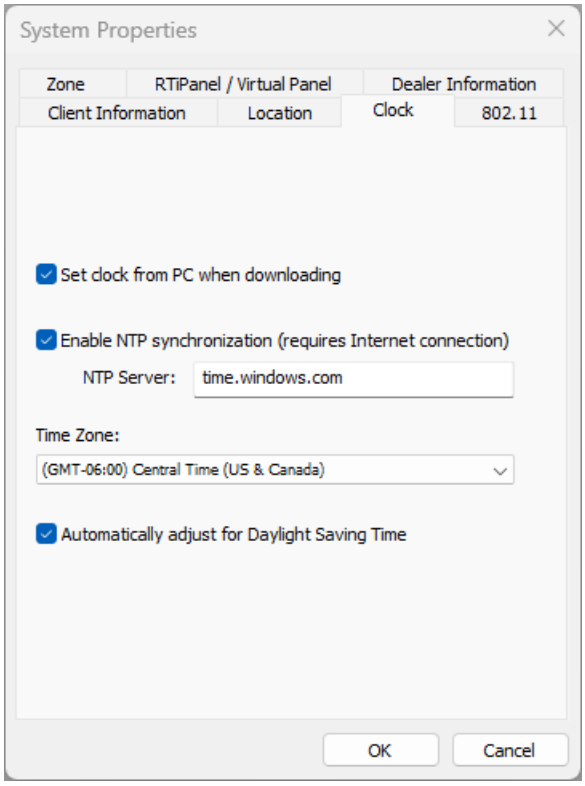
For **Country**, select the country the project resides in from the dropdown menu.

Select the **City** from the List of cities. If the city is not on the List, enter the location's latitude and longitude.

https://www.findlatitudeandlongitude.com

It is important to enter the correction information for accurate time-based events using the astronomical clock in the XP Processor. This option will populate the

project file's location tab on applicable RTI controllers.

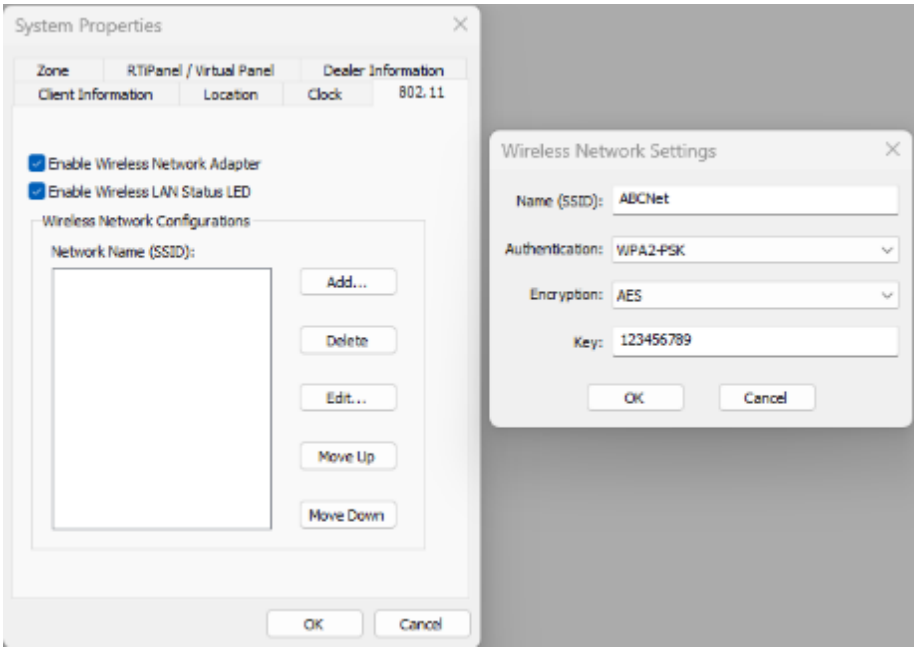


Set clock from the PC when downloading will be checked by default. You can uncheck the option if you upload the project from another time zone. In most instances, this option should remain selected.

Enable NTP synchronization requires an internet connection and is used to synchronize with computer clock time. This option should remain selected.

Time Zone – Select the time zone from the dropdown menu where the project resides. Doing so will ensure the astronomical clock is correct. This option will populate controllers with a clock setting.

Automatically adjust for Daylight Saving Time should remain selected unless DST is not observed.



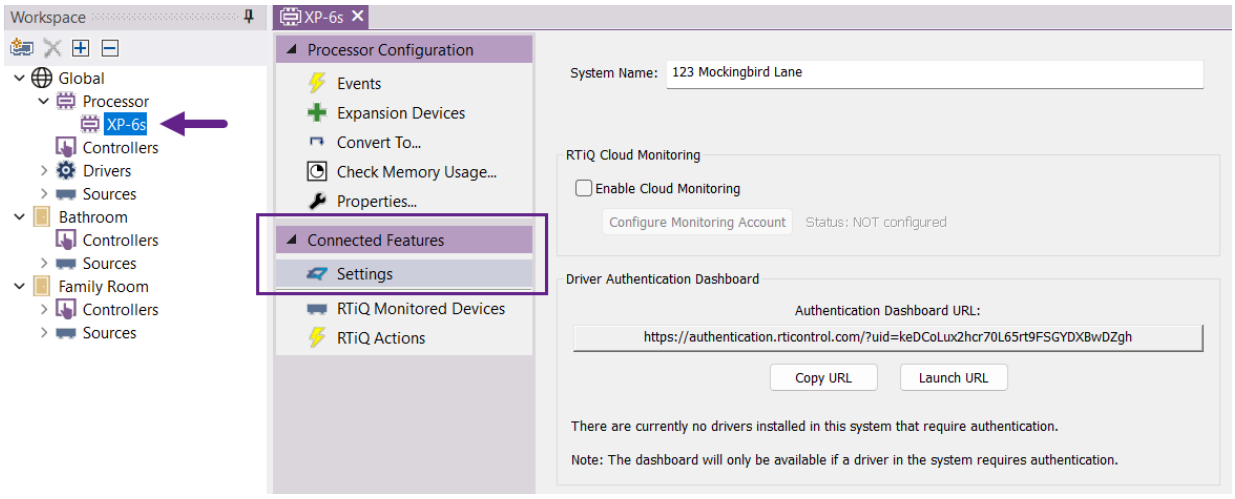
If you wish to use Wi-Fi on applicable RTI controllers, select **Add** under the network name and enter in the SSID Name, Authentication, Encryption Type, and Key. Please note that these are case-sensitive values.

If you are using Wi-Fi, ensure the **Enable Wireless Network Adapter** and **Enable Wireless LAN Status LED** are enabled.

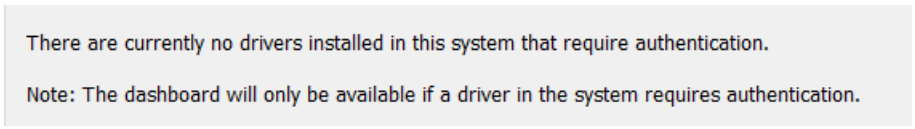
Configuring RTI Authentication

RTI Authentication provides a secure method of communicating with 3rd party devices using the client's credentials for drivers that support authentication. It allows the RTI integrator to send the client a URL link, which will open a web browser dashboard, where they can enter the login and password credentials. This method results in a secure method allowing clients to manage their passwords without requiring integrators to update project files each time. RTI will continue to develop drivers that support authentication when possible.

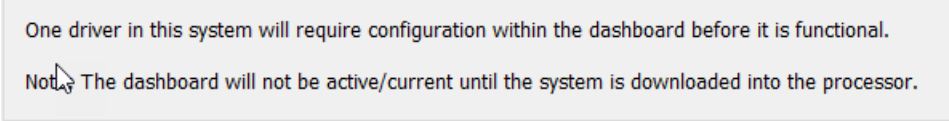
1. Select the **main processor** in the workspace. In the **processor configuration**, under the **connected features** option, select **settings**.



2. In the authentication dashboard, **confirm** that any drivers require authentication settings. If you see the message below, you have not added any drivers that support this method and should not continue.



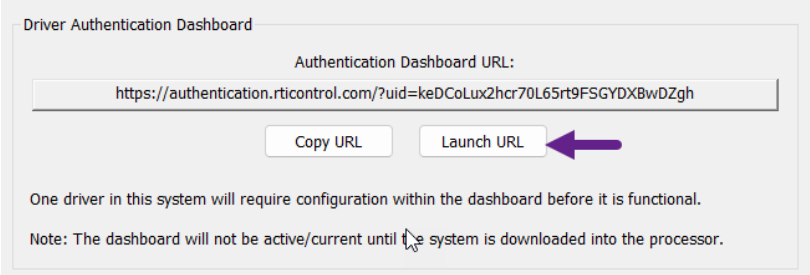
If any drivers support authentication, there will be a message in the window indicating the number of drivers that require authentication before being functional.

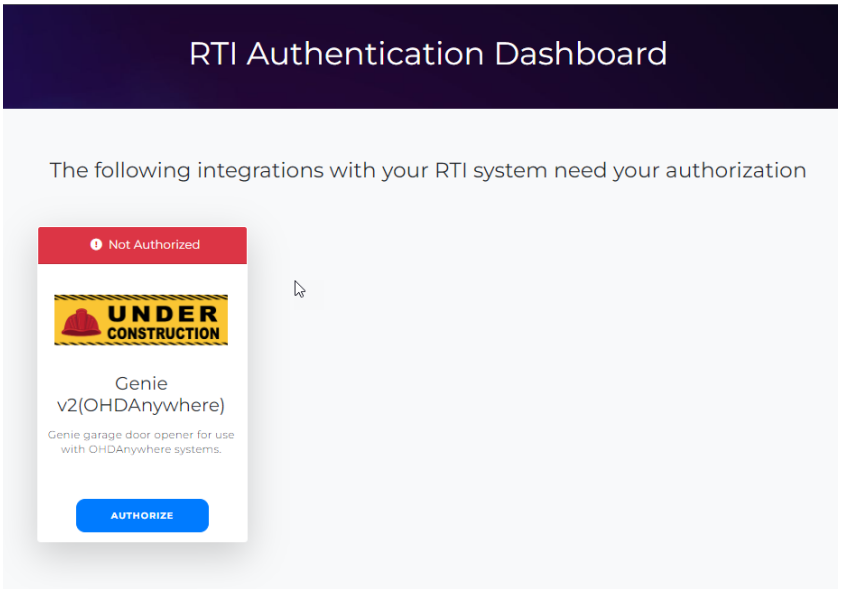


3. **Update the processor configuration** (not RTI devices) to establish the authentication communication.

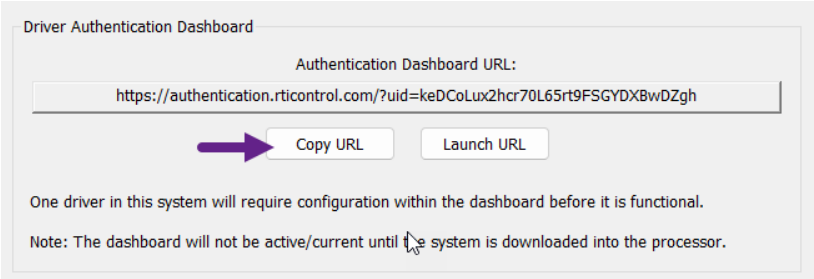
4. Continue programming the project. Once complete, upload all RTI devices as required. Return to the authentication page in the processor configuration window.

5. Select “Launch URL.” Doing so should launch a window displaying the RTI authentication dashboard with any applicable devices that need to be authorized by the end-user.

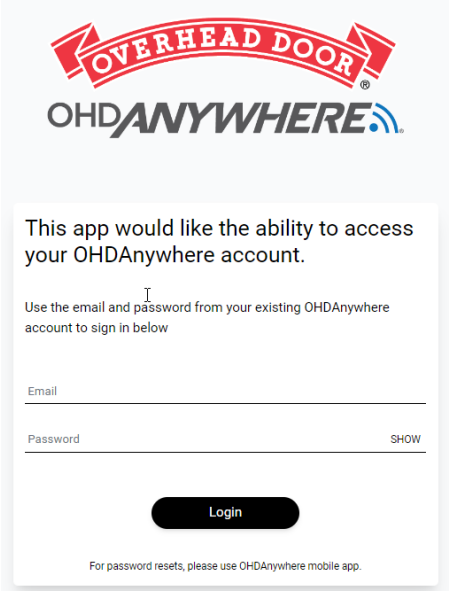




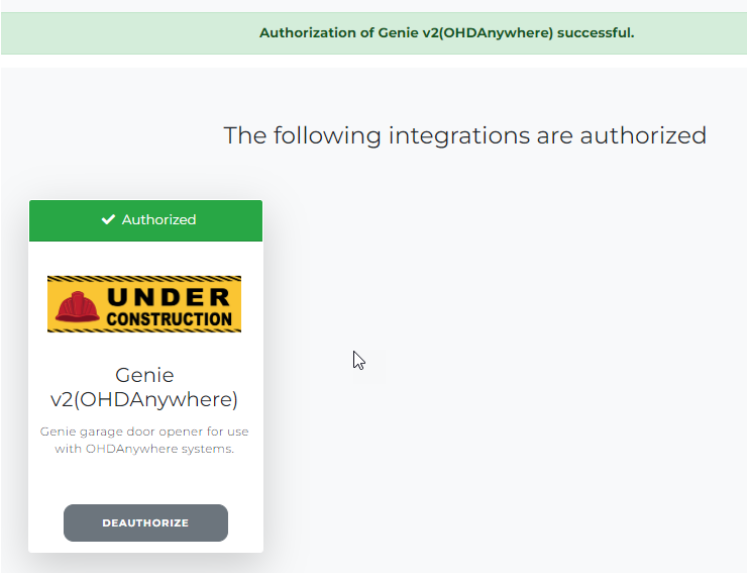
6. Once the authentication dashboard and available devices have been confirmed, select “Copy URL” and send it to the client to authorize any devices requiring authorization.



7. The client should select the URL and access the RTI Authentication Dashboard. Direct them to select “authorize,” which will take them to the manufacturer authorization website. The end-user should enter the login credentials obtained by the product manufacturer.



8. Once the end-user enters the correct credentials, the RTI Authentication Dashboard will display the device(s) that are authorized and authenticated.

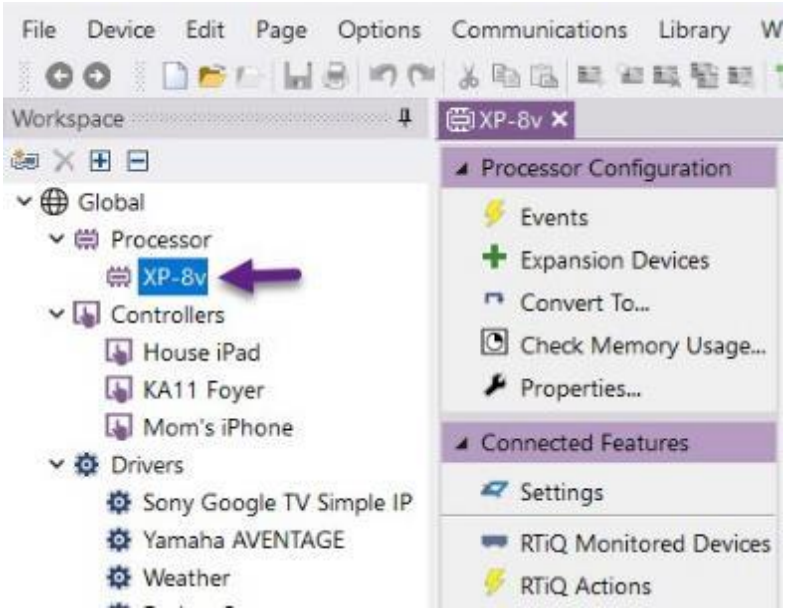


The RTI processor will effectively communicate and control all devices that have been programmed and authorized. If the client changes login criteria, a link can be sent anytime for reauthorization.

If more information is required, a technical bulletin and demonstration video are available on the RTI dealer site.

Processor Configuration

1. Start by selecting the main processor in the workspace.



2. The processor configuration section to the right of the workspace will display a tab featuring the options and settings available for the main processor.

   Events- The Events tab features the available trigger events based on the main processor's features, including sense inputs, and an astronomical clock,

   Add Expansion Devices – Select this tab to add any expansion devices on the network that communicate with the main processor.

   Convert To – The convert option allows you to convert the processor to another model. Please note that not all processors have the same features. Conversion could remove certain features and ports.

   Check Memory Usage – This option will check the memory utilization with items stored on the processor. RTiPanel devices, processor data, and any UI data can impact the processor memory.

   Properties – The properties selection will allow the RTI installer to configure settings and options on the main processor.

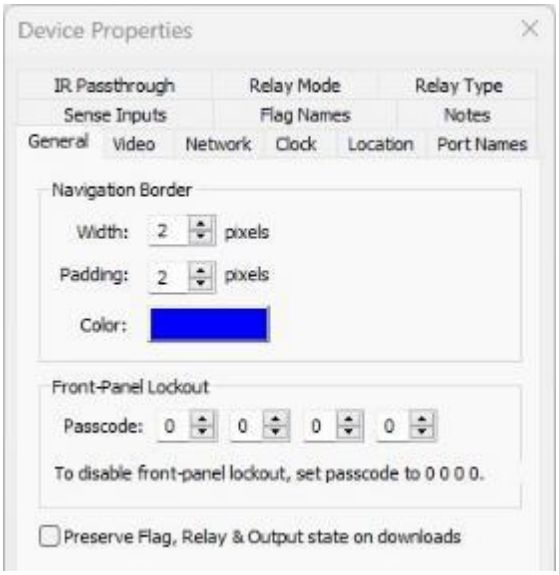
   Connected Features include the settings and options available for RTIQ and RTI Authentication. This topic is reviewed in a later chapter.

3. Select the Properties tab in the processor configuration.

4. Configure each processor tab that applies to the project.

Please note that processors have different configuration options and features; not all tabs and settings will be available on every processor.

General Tab



Navigation Border – The cursor's width, padding, and color on an on-screen display when an option is selected. You may skip this option if you are not using the HDMI output on the processor to create a user interface on a video display.

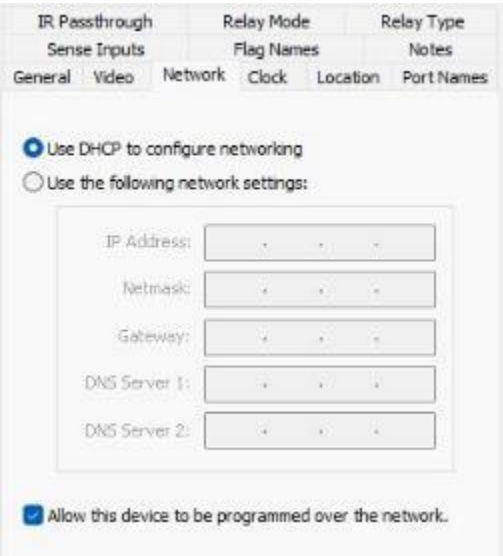
Front-Panel-Lockout – Setting a four-digit code will ensure the processor settings will not be controllable through the front display on the hardware. Set the passcode to 0 0 0 0 to turn off front-panel lockout.

Preserve Flag, Relay, & Output state on downloads – This option, when selected, will ensure the processor properly tracks these states after a download.

Video Tab

Video Output Resolution – You may set the output resolution on the processor to 1080p or 720p with various refresh rates. The default is 1080p @ 60Hz. This option is only referenced when using the processor HDMI output to a video display.

Network Tab

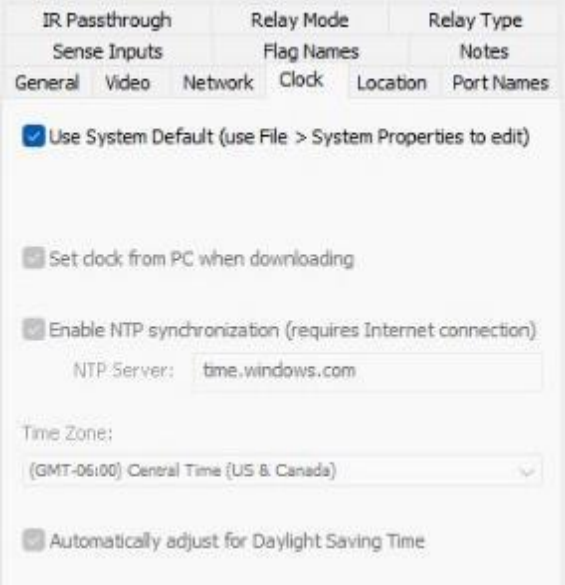


Use DHCP to configure networking – Select this option if using the DHCP server on a network router to assign DHCP.

You must assign static network settings to your processor if not using DHCP. Enter an IP Address, Netmask, Gateway, and DNS servers.

Allow this device to be programmed over the network – When selected, uploading to a processor is possible with a network connection instead of a USB cable.

Clock



Use System Default – If this option is selected, the processor will inherit the settings defined in the system properties, and the other settings will be greyed. If unchecked, fill in the settings manually.

Set clock from PC when downloading – This option will synchronize the settings using the PC clock after each download.

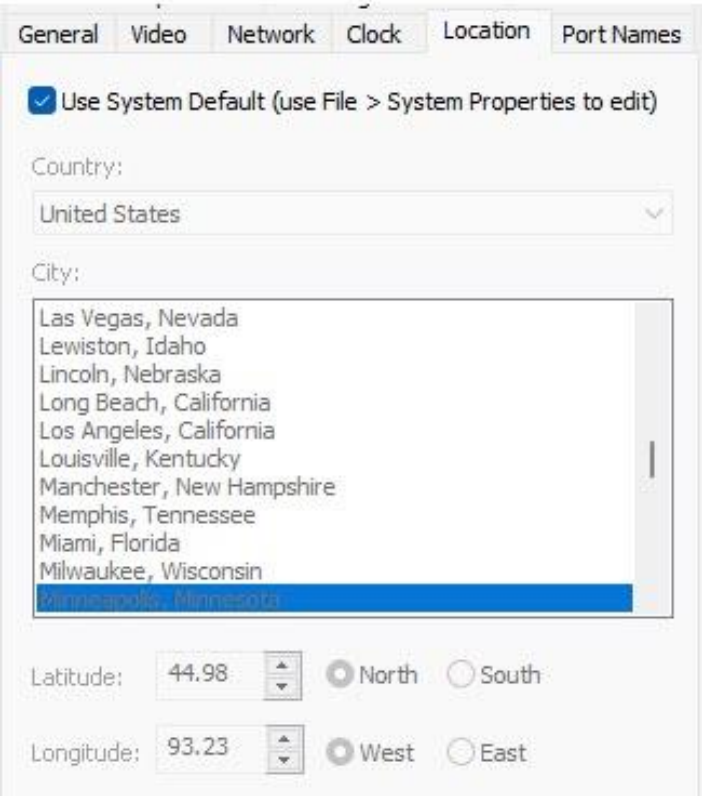
Enable NTP Synchronization – NTP, or The Network Time Protocol, is a networking protocol for clock synchronization. Selecting this option will use the NTP and requires an internet connection.

NTP Server – The NTP server name is used for synchronization. The default URL is time.windows.com.

Time Zone – Select the project's time zone using the available dropdown menu options.

Automatically adjust for DST - The process involves advancing the clocks by one hour during the warmer months and then reverting to Standard Time during the fall.

Location



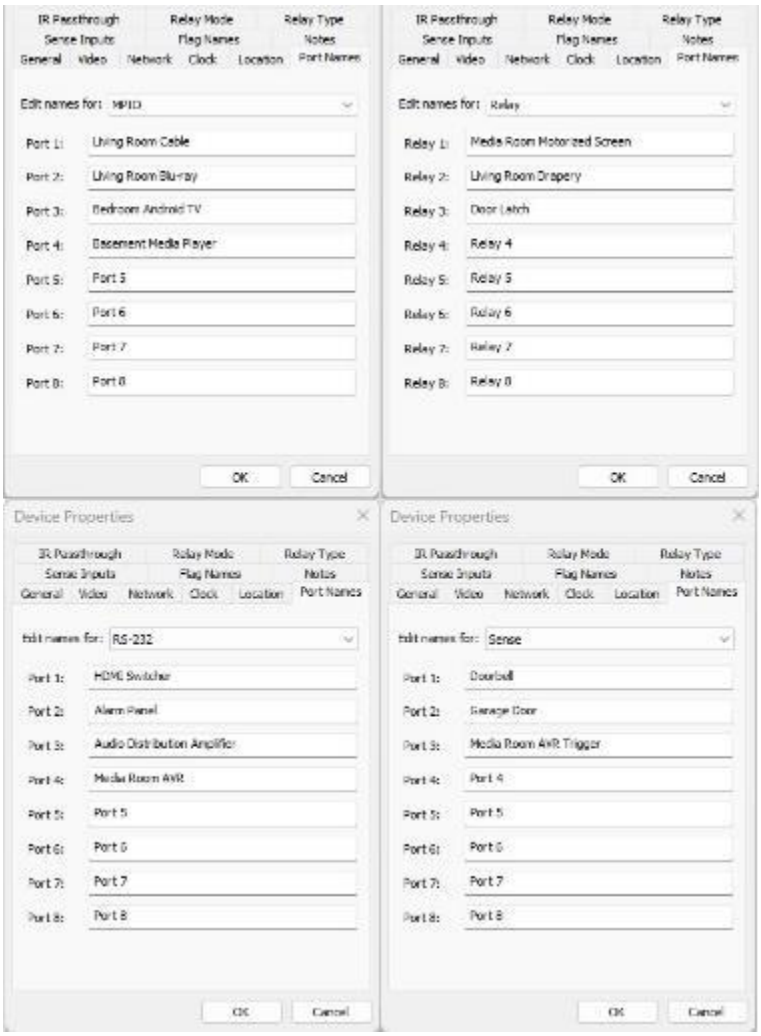
Use System Default – If this option is selected, the processor will inherit the settings defined in the system properties, and the other settings will be greyed. If unchecked, fill in the settings manually.

Country – Select the country from the dropdown menu where the project is located.

City – Select the project city and state. If the city is not populated, enter **Latitude** and **Longitude** coordinates. You can use a website such as https://www.latlong.net for precise information.

Port Names

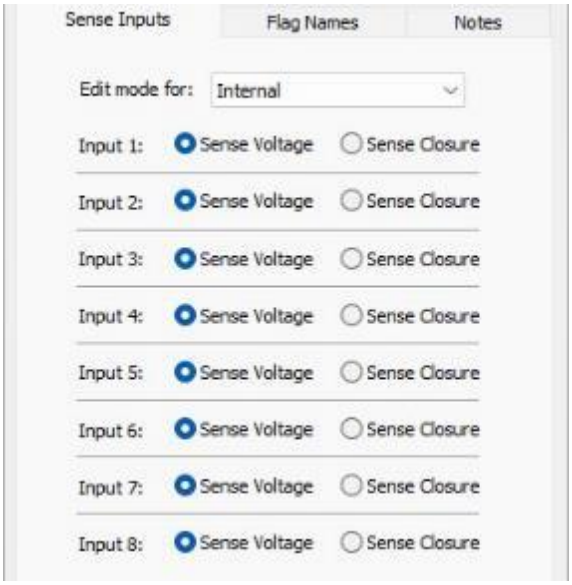
Processor MPIO (multi-purpose input-output) Relay, RS-232, and Sense Inputs may be named using the Port Names feature on a processor. When adding an IR device in the Add Workspace Item, IR ports are named.



It is important to understand that renaming a port will not change the address of the port settings. However, if a port is readdressed in a separate step, it must be changed here.

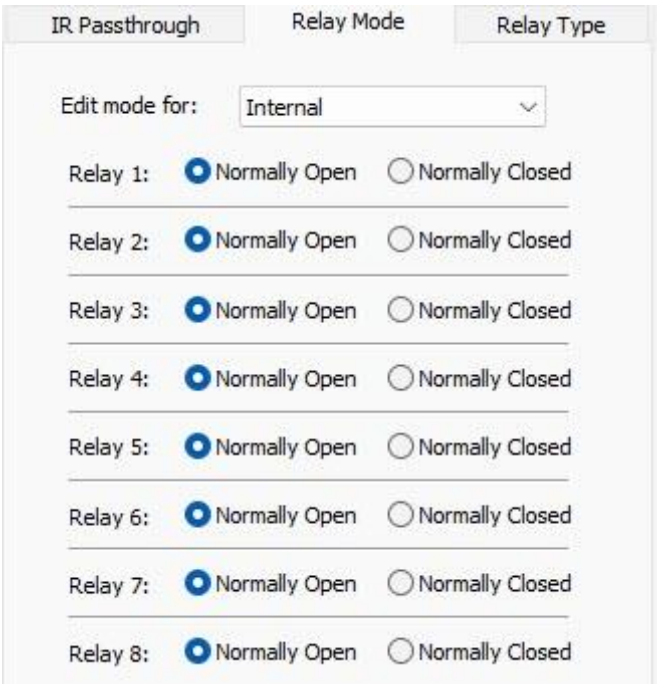
Edit Names For – Select the dropdown and rename any ports with port routing capabilities for the main processor and expansion devices.

Sense Inputs



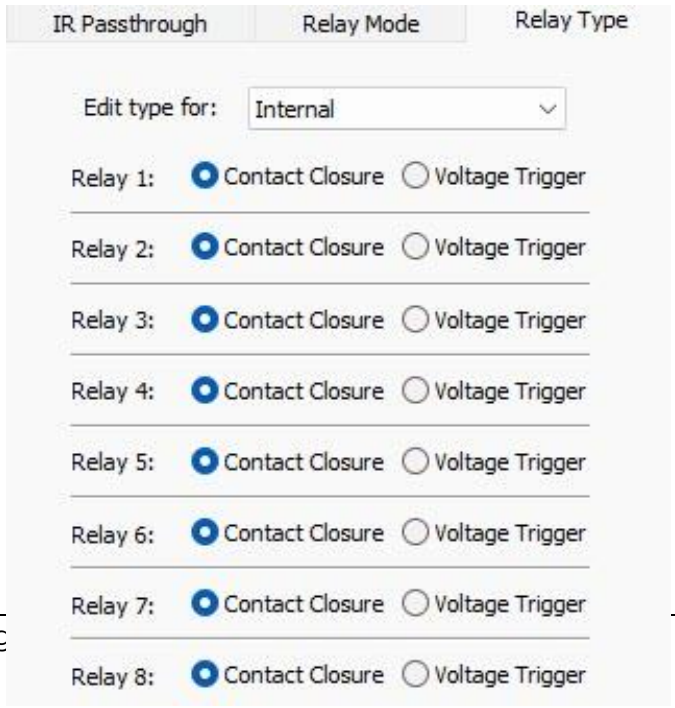
Configure the internal sense inputs on the internal processor by setting the required inputs to **Sense Voltage** or **Sense Closure**. Expansion devices with sense inputs can be accessed using the dropdown menu.

Relay Mode



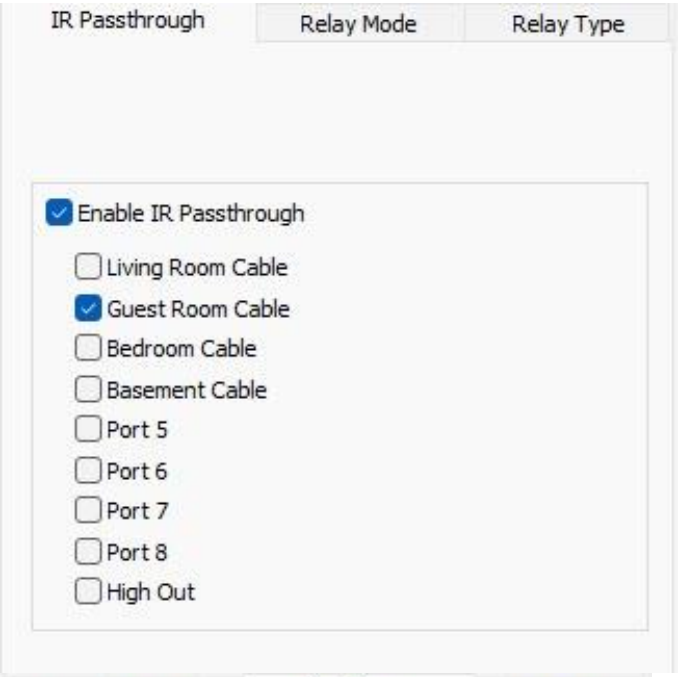
Set each relay to **Normally Open** or **Normally Closed** in the internal processor or expansion devices with relay control options.

Relay Type



Configure relays on the main processor or expansion devices as a contact closure or voltage trigger under **Relay Type.**

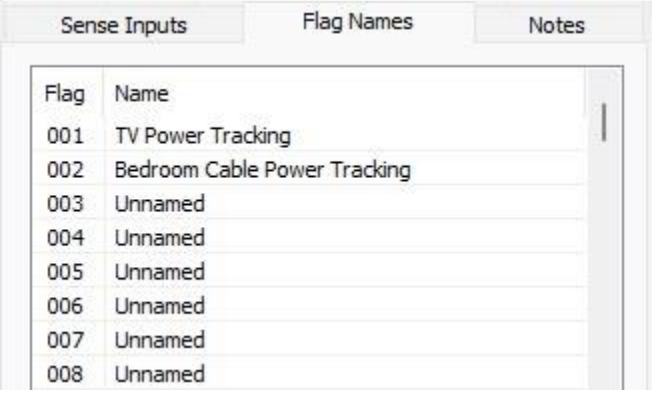
IR Passthrough



Connecting an IR receiver to the main processor and having IR pass through to any designated ports is possible.

Select the port(s) you wish the IR receiver to pass infrared signals. Untick this feature if it is not being used.

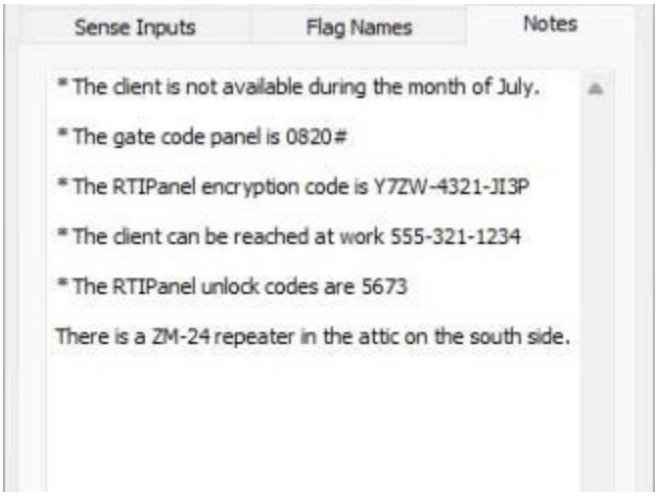
Flag Names



Used with **flag** and "**if flag set"** Macro Steps options, up to 256 flags can be named, set, cleared, or tested. While flags can be managed in the macro window, they may also be edited in the **Flag Names** properties.

Pro Tip: Flags are used to track the statuses of devices, button presses, and other items of interest in the software. Since the RTI installer sets and clears flags in the software, it is not a true status that can be obtained by polling or using sensing equipment. Use flags with a plan in case they get unsynchronized.

Notes



Notes can be optionally used to put essential information based on the project and space.

Controller Configuration

RTI controllers have a variety of functions and may differ between models. Configuring controller properties is vital to the client's experience and ensuring proper communication.

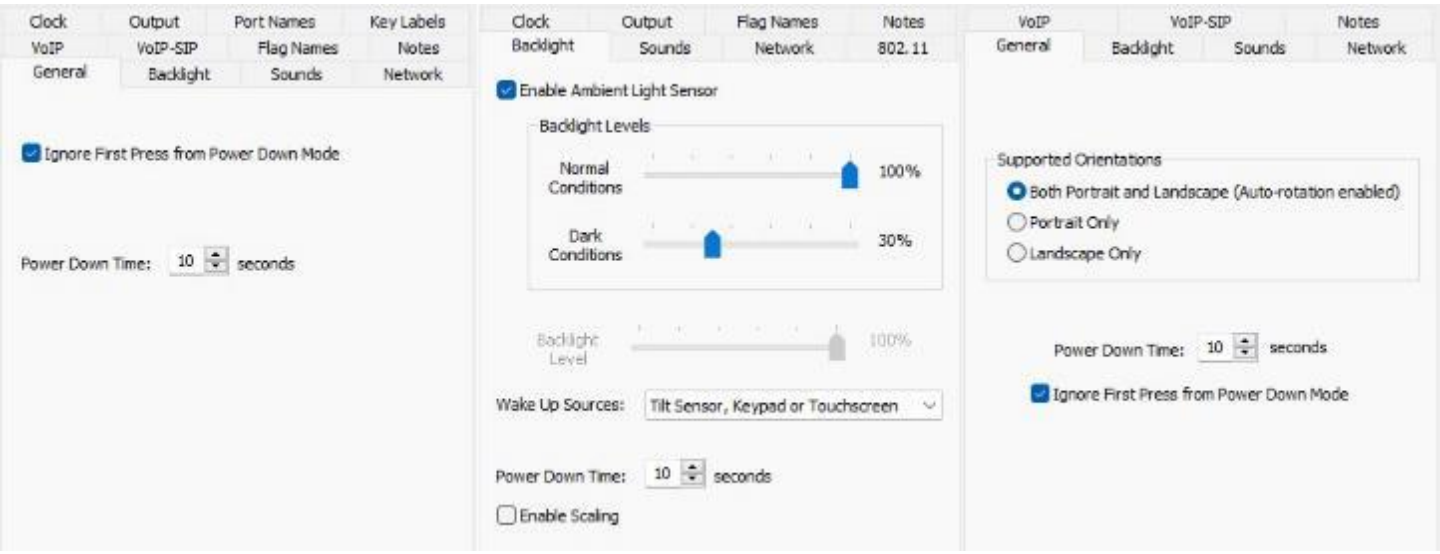
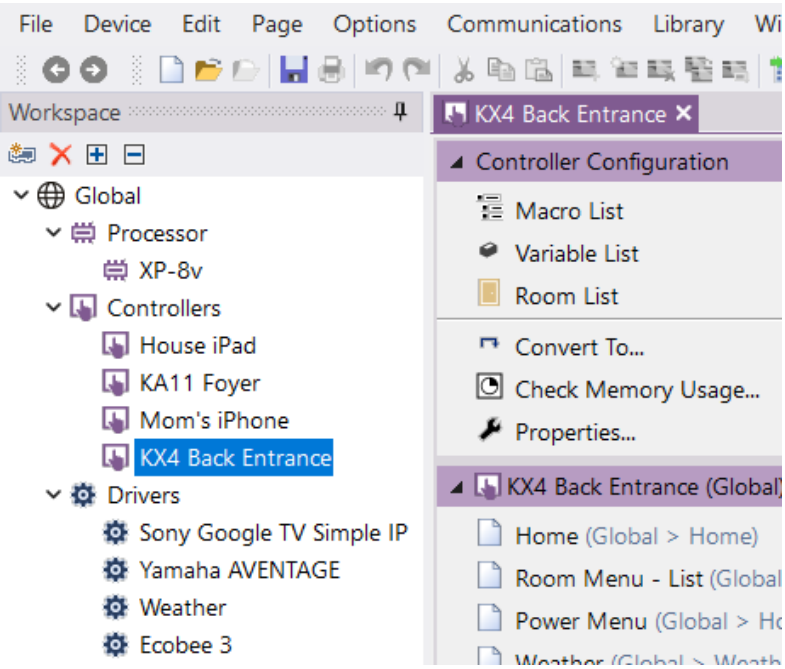


Figure 1 From Left to Right: Keypad, Handheld, and RTiPanel

1. Select the controller in the workspace.

2. On the right side, the controller configuration options will display.

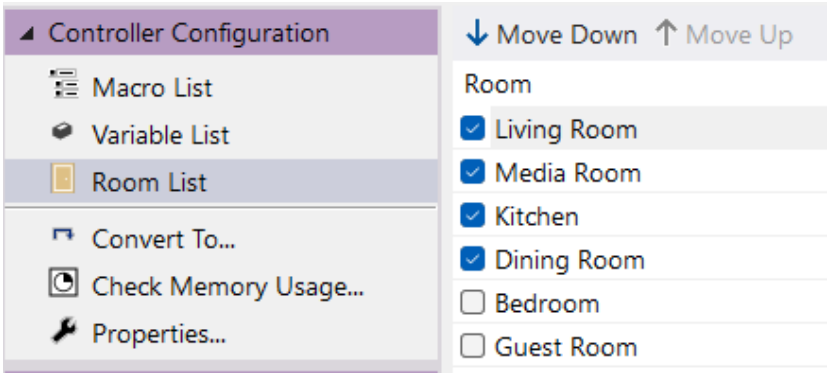
   The controller configuration tab will display options available for configuration and an overview of all the pages that comprise the source devices.



The **Macro List** displays the tags comprising the source device buttons on the selected controller.

The **Variable List** displays the tags that comprise the source device feedback on the selected controller.

Room List – The Room List will display a list of rooms available on the controller. When using list menus, rooms can be removed on the selected controller.



Rooms can be selected or deselected using list menus on the selected controller.

Use the **Move Down** and **Move Up** functions in the toolbar to change the room order in the list menus.

Convert To will convert a controller to another controller with a like resolution.

Check Memory Usage will display a pie chart of available and used memory on the controller selected.

3. Select the **Properties** option in the controller configuration.

KX4 Keypad Properties

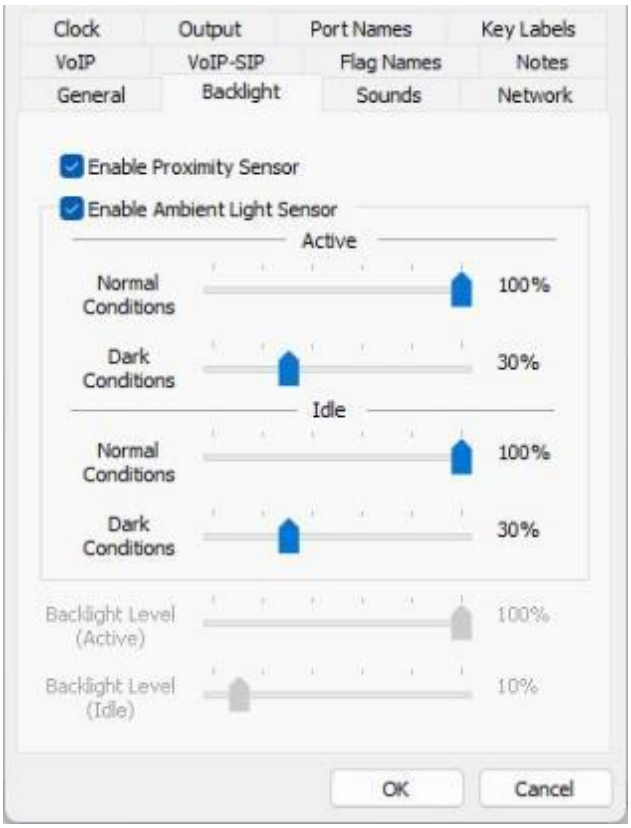
GENERAL

Ignore First Press from Power Down Mode – Ignore the first press after power down selection, which will awaken the keypad from a low light mode.

Power Down Time – In seconds, the time it takes for the screen to go dark after no button presses are made.

Pro Tip: Always adjust the Power Down to a value in the 30 to 60-second range. Doing so will improve the client experience.

BACKLIGHT



Enable Proximity Sensor – Will awaken the screen if selected if direction motion is detected.

Enable Ambient Light Sensor – This option will awaken the screen if any light is detected.

Active Normal Conditions – Set the screen brightness when the device is used.

Dark Conditions- the screen brightness when the device is used in a dark situation.

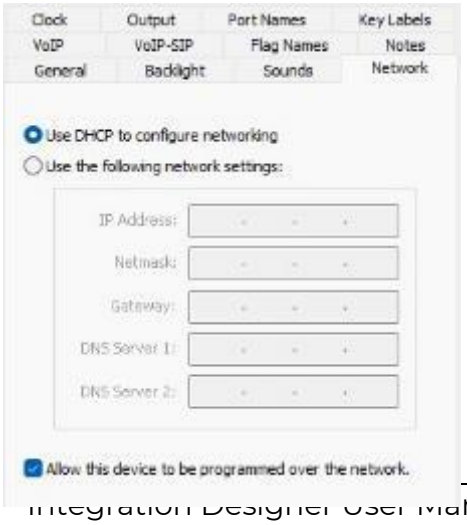
Idle Normal Conditions – the screen brightness when the device is not in use (idle.)

Idle Dark Conditions – the screen brightness when the device is not used (dark.)

Backlight Level (Active) – When used without the **Ambient Light Sensor**, the screen brightness when in use.

Backlight Level (Idle) – When used without the **Ambient Light Sensor**, the screen brightness when not in use (idle.)

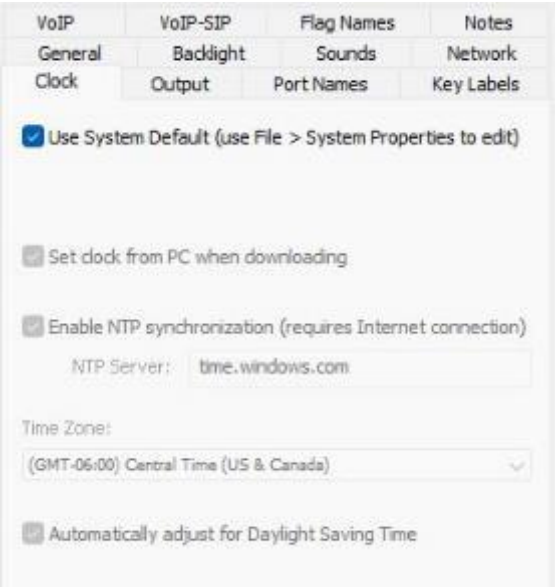
NETWORK



If **DHCP** is not required, deselect this option and enter the device's IP Address, Netmask, Gateway, and DNS Servers.

Allow this device to be programmed over the network should be checked to allow for synchronization without a USB cable.

Clock



Use System Default – Obtain the controller properties from the system properties.

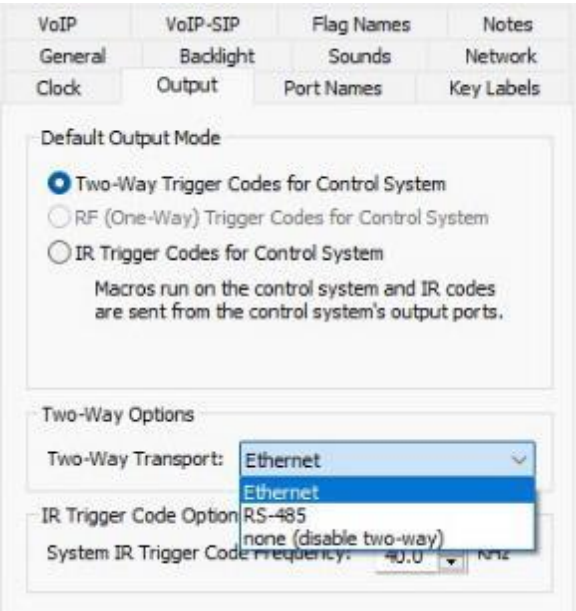
Set Clock from PC when downloading will set the time in the settings after each download.

Enable NTP synchronization requires an internet connection and uses NTP (network time protocol) to obtain time information.

Time Zone – Select the project site's time zone from the dropdown menu.

Automatically Adjust for DST will reflect the hour change based on DST.

OUTPUT



Setting the output tab based on the communication method used for the project is critical for operation.

Two-way Trigger Codes for Control System

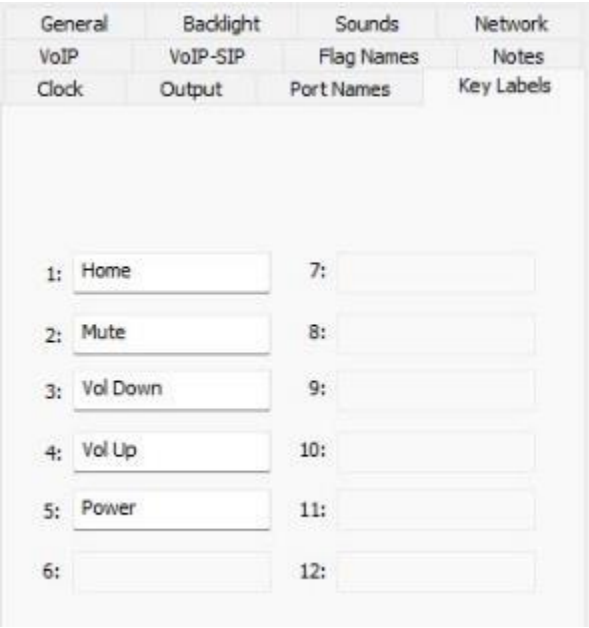
The Two-Way Transport will use a two-way trigger from the device to the processor. Set the type to Ethernet, RS-485, 802.11, Zigbee, or none.

IR Trigger Codes for Control System

In this mode, devices send IR trigger codes to the control processor, which outputs IR commands and macros. IR triggers are not commonly used.

Two-Way Options – Select the Protocol used in the dropdown menu. Depending on the type of controller, you will have different options available.

Ethernet will use the network for communication, while **RS-485** will require a connection to the processor or via a CB8 communication block. You can also turn off two-way communication by setting the option in the dropdown to **None**.

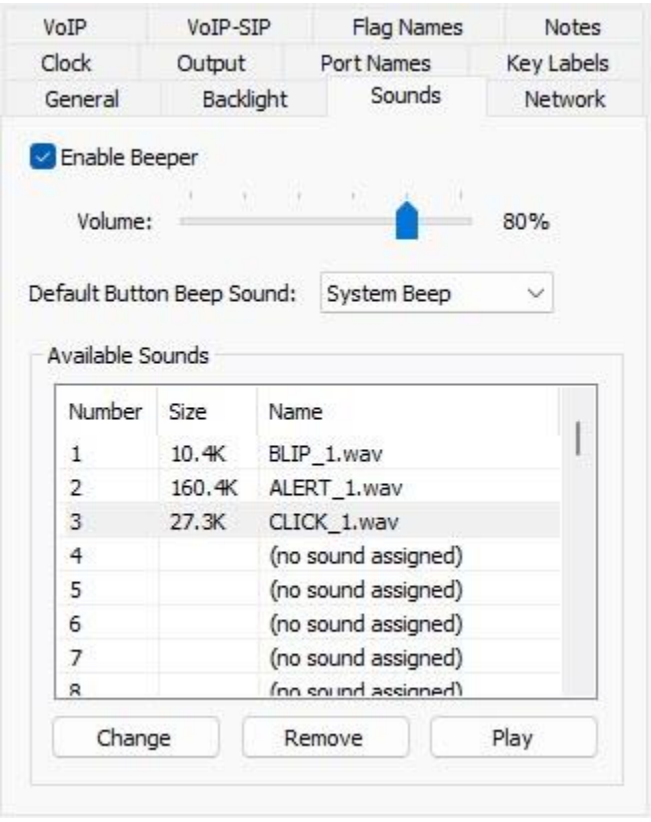


KEY LABELS

Key Labels are available for devices with customizable keycaps or buttons using RTI's **Laser Shark** engraving service.

You may change the names to match the custom keycaps. Doing so will only affect the display on the software user interface.

SOUNDS



Enable Beeper

Check this box to enable the beeper. If the box is unchecked, the controller will play the default button beep sound every time a button is pressed.

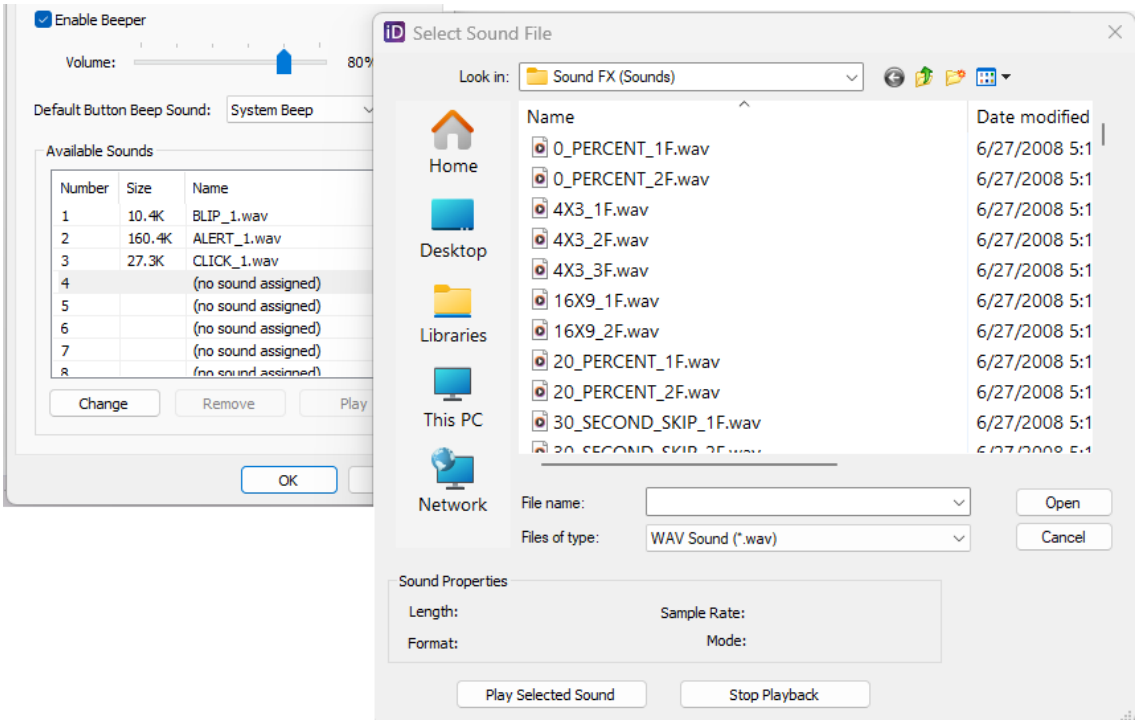
Volume

Set the volume slider to set the volume level of the beeper.

Default Button Beep Sound

This setting determines the default sound for all buttons on the control device. System beep is the built-in beep sound. Choose the available

sounds in the dropdown list to change the device default.



Add to the List of available sounds by selecting **Change**. An Open Sound File dialog box allows you to select a *.wav file. You may add up to fifty sound files.

Remove a file from the List by selecting **Remove**.

Test an added sound file by selecting **Play**.

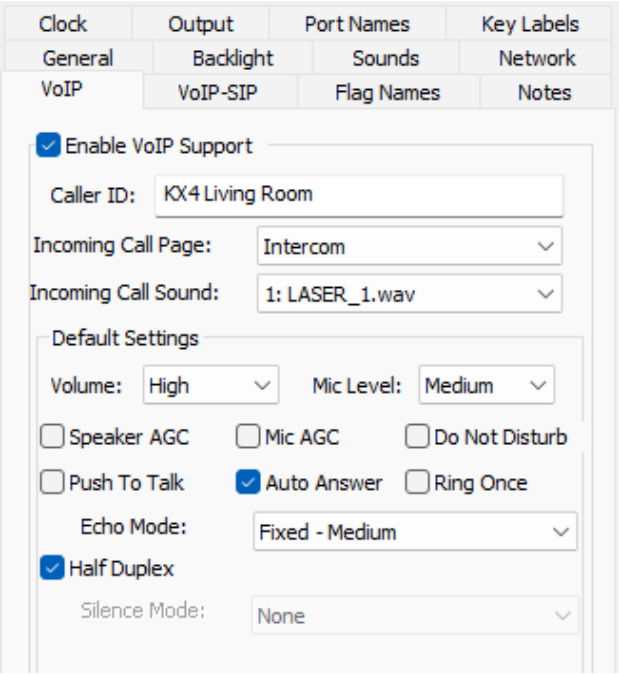
Sound length is limited to the available memory on the controller.

Choose **OK** to confirm settings.

You may trigger sounds on the controller using macro programming as well.

Pro Tip: Sounds can be used in various situations, such as arming a security panel and hearing a confirmation sound or in a situation where someone is visually impaired. Avoid using sounds with long durations or high bit rates for best performance.

VOIP (Voice over IP)



Enable VoIP Support will allow VoIP options to be configured.

Caller ID: The controller's name appears on the intercom interface of the recipient during an incoming call.

Incoming Call Page: Set a page for incoming calls with VoIP communication options.

Incoming Call Sound: Optionally add a sound configured in the sound tab to play when an incoming call is received.

Under **Default Settings**, several options exist to allow customization of VoIP.

Volume: The speaker volume level can be set to High, Medium, or Low.

Mic Level: The microphone level can be set to High, Medium, or Low.

Speaker AGC: allows the internal speaker to adjust automatically to compensate for variations in volume from different talkers or variations due to a single talker moving relative to the mic. You may select or deselect this option.

Mic AGC: allows the microphone signal to adjust automatically to compensate for variations in volume from different talkers or variations due to a single talker moving relative to the mic. You may select or deselect this option.

Do Not Disturb: Select the DND feature if incoming calls should not be received.

Push to Talk:

Auto Answer: Automatically answer a call without selecting an answer button.

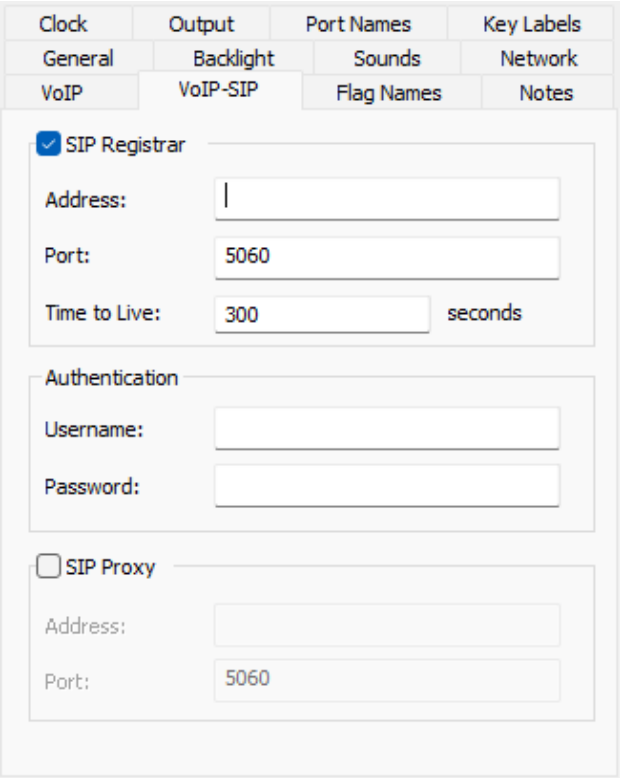
Ring Once: This option will make the controller ring once instead of repetitive rings.

Echo Mode: Set the echo suppression mode to none, automatic or fixed (low, medium, and high.)

Half Duplex: Setting this option will ensure that only one device can transmit simultaneously.

VOIP - SIP

A SIP server is highly recommended for best practice when using multiple controllers using an intercom feature. Configuring the VoIP (Voice over IP) SIP options for applicable controllers is necessary when using a Session Initiated Protocol (SIP) server.



Enable the **SIP Registrar** to enter the required SIP information.

Address: The address of the SIP server.

Port: The port utilized by the SIP server.

Time to Live: In seconds, the amount of time a data packet will exist before it is discarded.

Authentication

Username: The SIP username

Password: The SIP password

Enable **SIP Proxy** if using a proxy server.

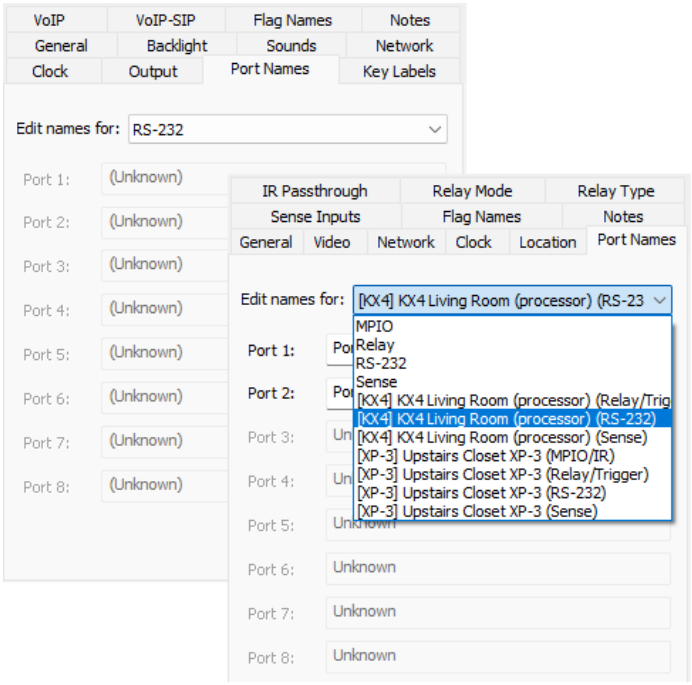
Address: The address of the proxy server.

Port: The default port of the proxy server.

PORT NAMES (KX4)

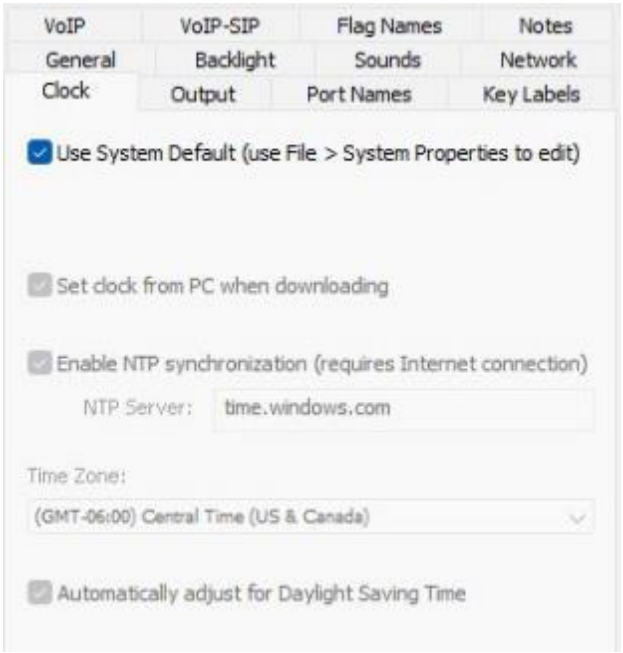
The KX4 features a built-in control processor with two serial ports. The two RS-232 ports may be used as a processor or expansion device for two-way communication.

To change the port names, you must go to the main processor properties when using the KX4 as an expansion device. Renaming ports do not address signal routing and are only meant for naming purposes.



Access the main processor to assign port names when using the KX4 as an expansion device.

When using the KX4 as a main processor, the port names may be changed in the processor or keypad port name settings.



CLOCK

Use System Default – Obtain the controller properties from the system properties.

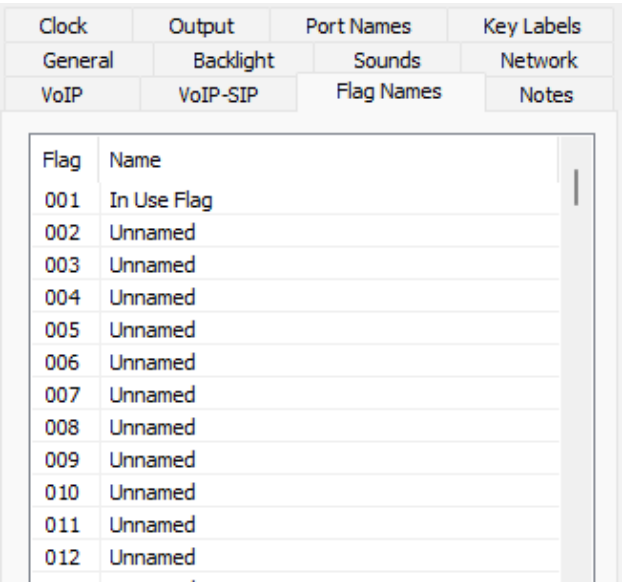
Set Clock from PC when downloading will set the time in the settings after each download.

Enable NTP synchronization requires an internet connection and uses NTP (network time protocol) to obtain time information.

Time Zone – Select the project site's time zone from the dropdown menu.

Automatically Adjust for DST will reflect the hour change based on DST.

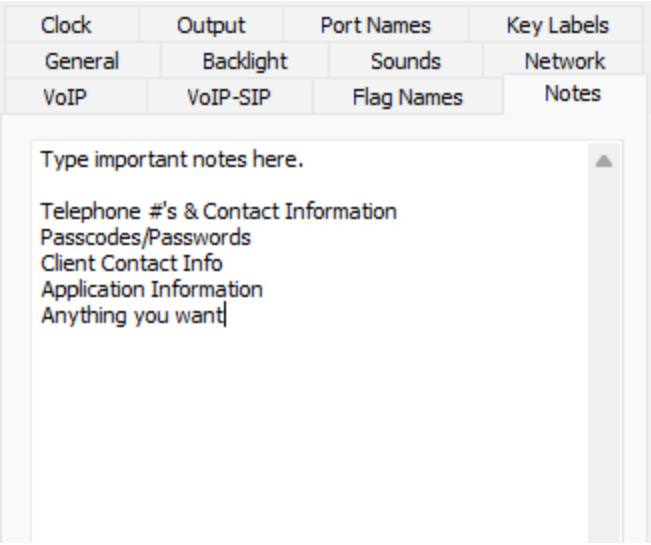
FLAG NAMES



Two hundred fifty-six flags are available on a controller to track power states, button presses, and other criteria. The Macro Steps **flag** commands may be set, cleared, and tested.

A **controller flag state** is exclusive to the controller and unavailable on other controllers or the control processor.

NOTES



Use the **Notes** tab to enter any essential information about the project.

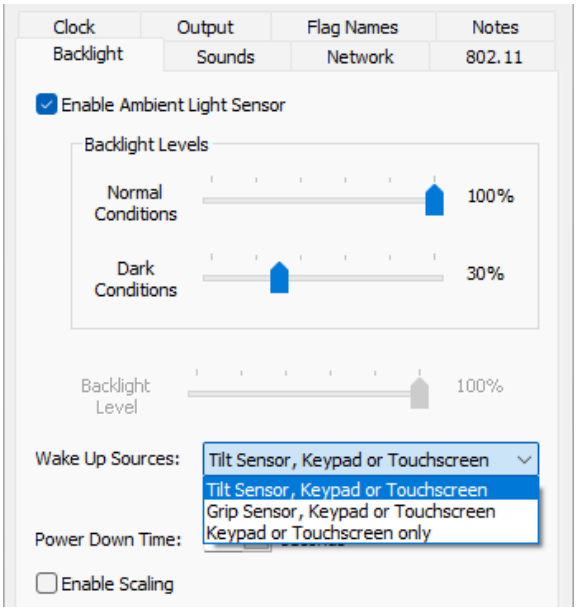
T4x Controller Properties

Controllers with the ability to use Wi-Fi, Zigbee, and 433Mhz one-way radio frequency will have different or additional properties in most cases.

1. Select the controller in the workspace, then select **Properties** in the controller configuration window.

   BACKLIGHTING

2. Change the backlighting to a desirable level.



To enable the **Ambient Light Sensor**, select the option.

Backlight Levels can be set for **Normal** and **Dark Conditions**.

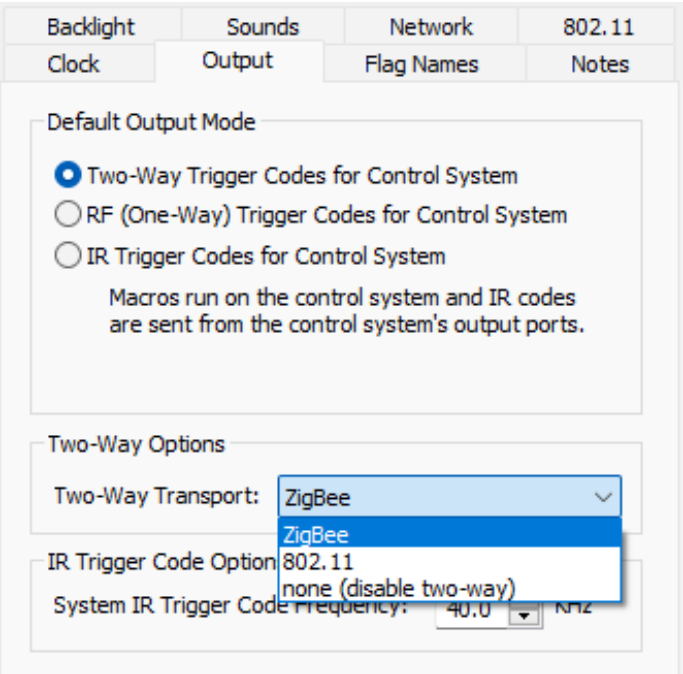
You may adjust the **Backlight Level** by turning off the Ambient Light Sensor.

Certain handheld controllers have a feature that will wake up the remote using a combination of options.

3. Select an option under the "**Wake Up Sources**" dropdown menu option:
   Tilt Sensor, Keypad, or Touchscreen
   Grip Sensor, Keypad, or Touchscreen
   Keypad or Touchscreen only

4. Select the Enable Scaling option.

OUTPUT



Setting the output tab based on the communication method used for the project is critical for operation.

Two-way Trigger Codes for Control System

The Two-Way Transport selection option will use a two-way trigger from the device to the processor.

IR Trigger Codes for Control System

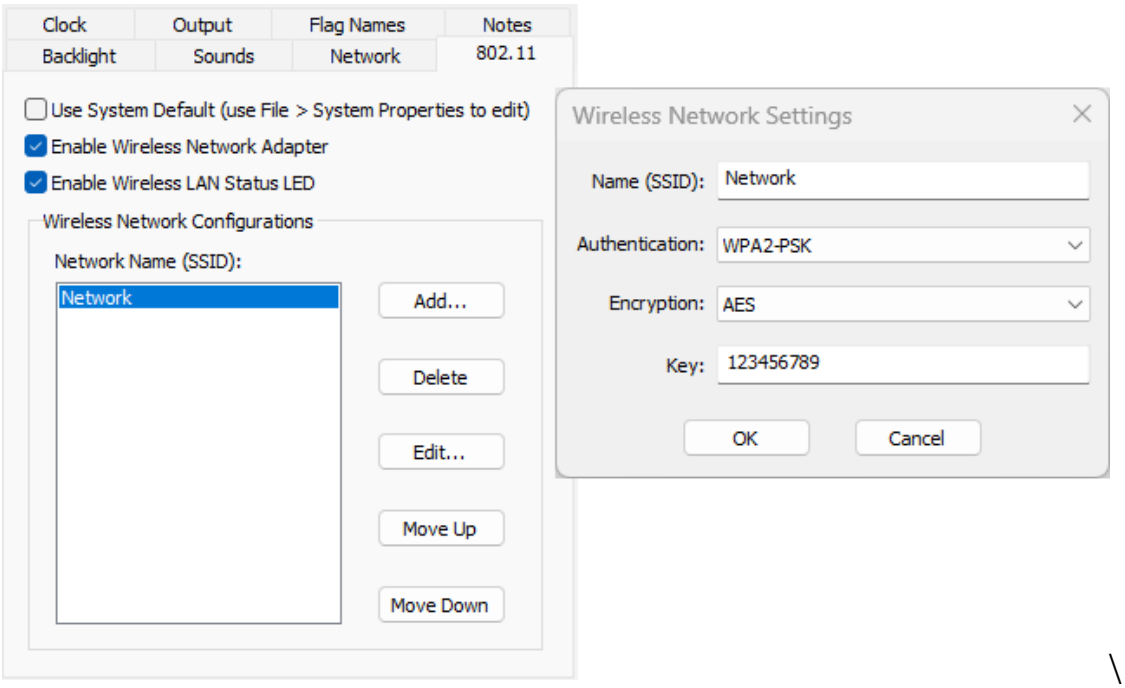
In this mode, devices send IR trigger codes to the control processor, which outputs IR commands and macros. IR triggers are not commonly used.

Two-Way Options – Select the Protocol used in the dropdown menu. Depending on the type of controller, you will have different options available.

Ethernet will use the network for communication, while **RS-485** will require a connection to the processor or via a CB8 communication block. You can also turn off two-way communication by setting the option in the dropdown to **None**.

Pro Tip: To create the best client experience, avoid using 802.11 as a primary two-way transport protocol. Instead, select Zigbee and add 802.11 settings on applicable controllers. Doing so will take advantage of the quick connection Zigbee offers by sending commands instantly. Feedback will return to the controller via Wi-Fi so the client can enjoy camera feeds and cover art, which Zigbee does not support.

802.11



\

Use System Default: Selecting this option will inherit the settings created in the 802.11 section of the system properties.

You may deselect "Use System Default" and add multiple Wi-Fi networks if you have a different network for a certain controller.

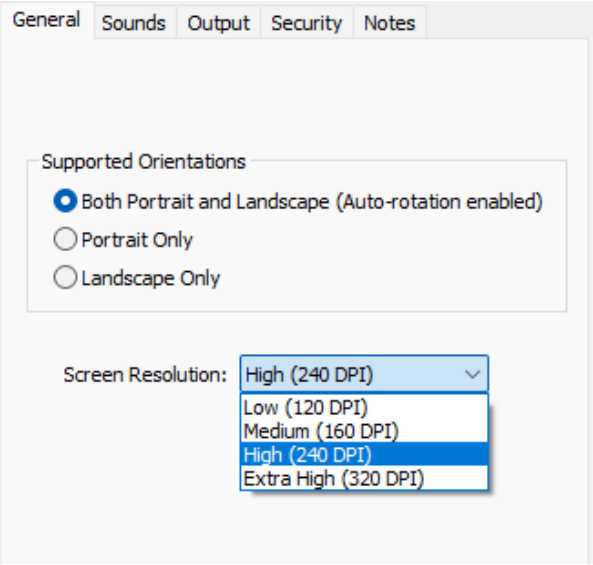
1. Select **Add**.
2. Enter the **name** or SSID of the network.
3. Enter the **Encryption type**.
4. Enter the **Network Key**.

You may **add** primary and secondary Wi-Fi networks while **editing** or **deleting** existing networks. Changing the priority order of the Wi-Fi network is possible using the up and down arrows.

RTiPanel Properties

An RTiPanel device is an iOS or Android phone or tablet or a Windows-based Virtual Panel that can be used as an installer tool or customer interface. The first virtual panel has a free license included in the software for local area network (LAN) use.

1. Select the RTiPanel controller in the workspace, then select **Properties** in the controller configuration window.



Supported Orientations

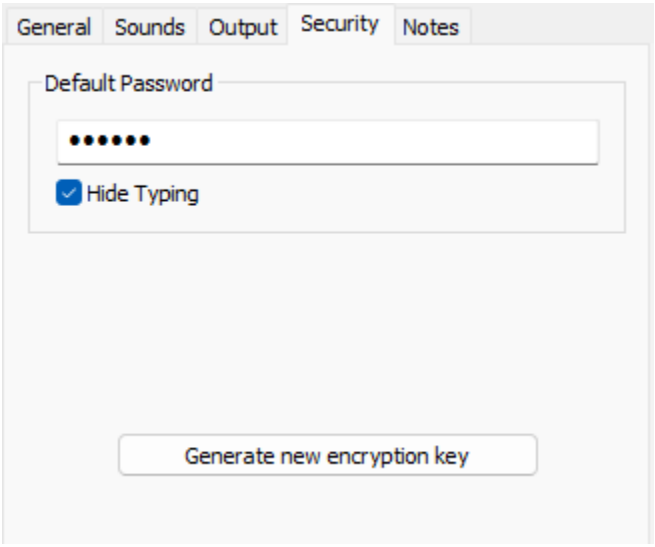
2. Select **Portrait**, **Landscape** Only, or **Both Portrait and Landscape**.

If using Portrait and Landscape, ensure auto-rotation is enabled on the device.

Screen Resolution (Android Only)

3. Set the **Screen Resolution** based on the device to Low, Medium, High, and Extra High.

SECURITY



RTI Installers can set a Password for end-users who wish to prevent others from downloading their control interface on a separate device.

Enter a **Default Password**. The password will be required when downloading the RTiPanel user interface from the control processor.

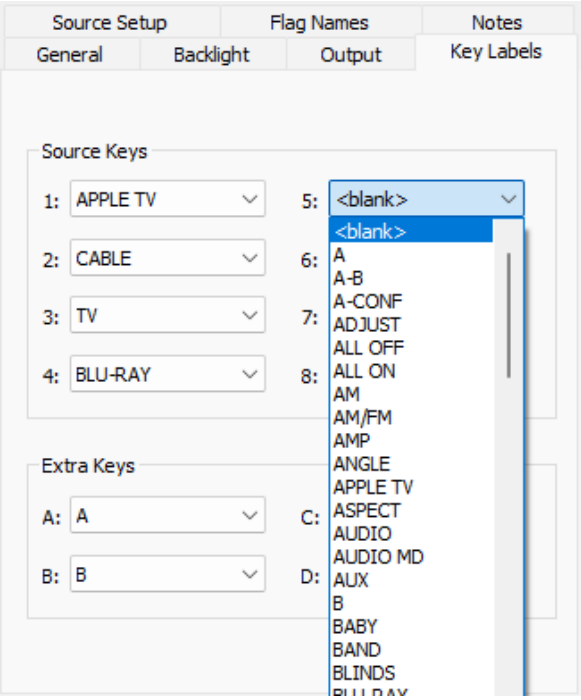
The **Generate New Encryption Key** button should only be selected at the direction of RTI Technical Support.

Pro Tip: Consider providing passwords on RTiPanel devices if they access sensitive functions such as unlocking a liquor cabinet, gun safe, or doors.

T1-B+ Properties

Certain hard-button controllers require different property configurations, which are necessary for programming.

KEY LABELS



For the **Source Keys** one through eight, select the preconfigured labels available in the software for each source button.

Keys 1-4 are on the top row, left to right.

Keys 4-8 are on the bottom row, left to right.

These keys are used for activity labels.

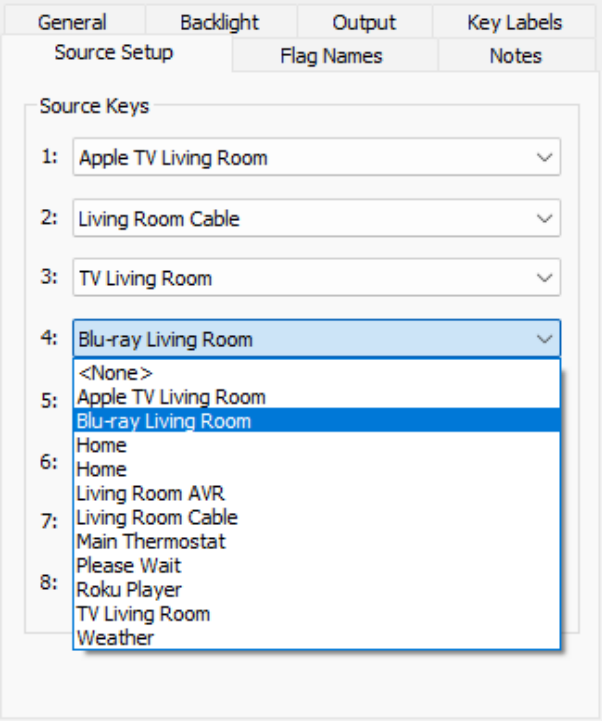
Four extra keys can be used for regular button commands and are labeled A, B, C & D by default.

These labels will name the buttons in the software for viewing purposes and do not impact the programming.

SOURCE SETUP

The source setup in a T1-B+ controller will create the pre-programmed user interfaces based on the sources designated to the activity buttons.

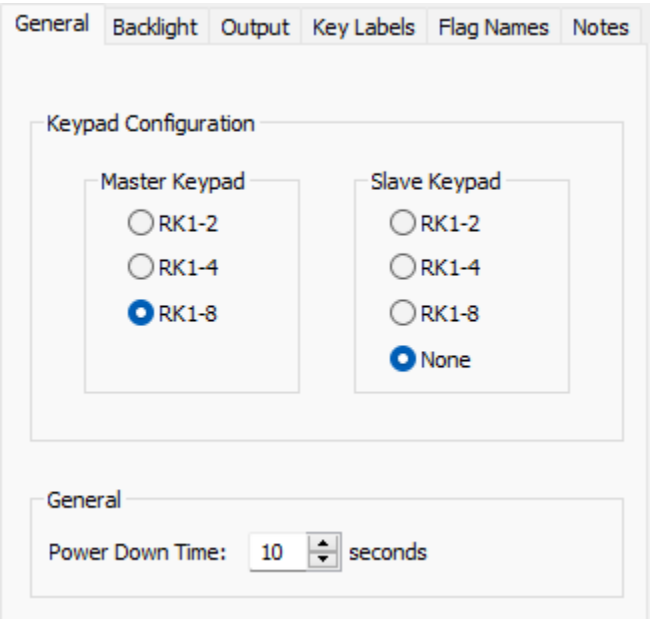
Select the dropdown menu and designate source devices added to Integration Designer to the controller activity buttons. After selecting **OK** from the source properties, pages will be automatically generated under each source in the workspace.



Pro Tip: Adding custom buttons using the RTI Laser Shark service will help give your controller a personalized look.

Laser Shark software can be downloaded on the RTI dealer website.

RK1+ Properties

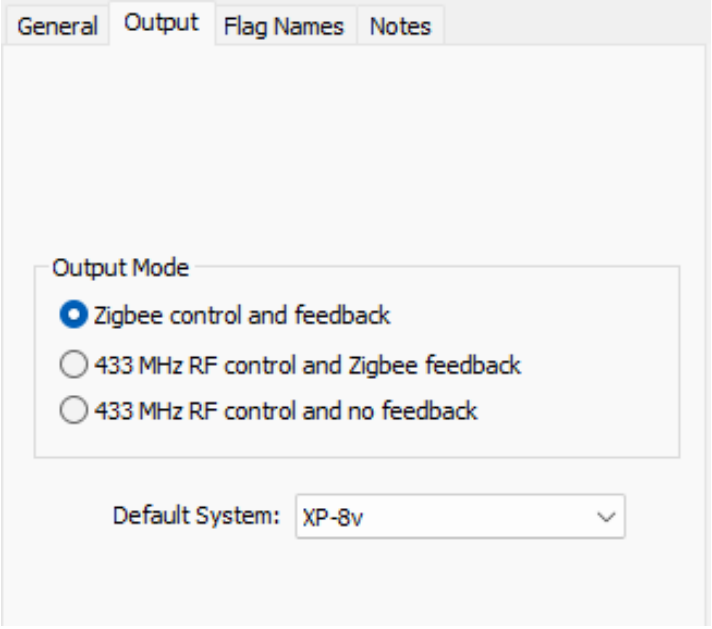


1. Select the **Master Keypad** model based on the number of buttons.

2. If using a **Slave Keypad** connected to the Master Keypad, select the model type.

General

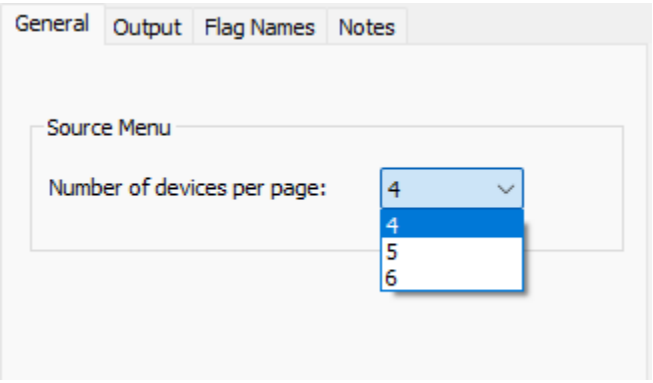
Set the **Power Down Time** to the number of seconds to turn off the backlighting.



U3 Properties

GENERAL

You may set the number of devices per page. Select an option from 4-6



OUTPUT

The U3 communication methods differ from other controllers.

Output Mode

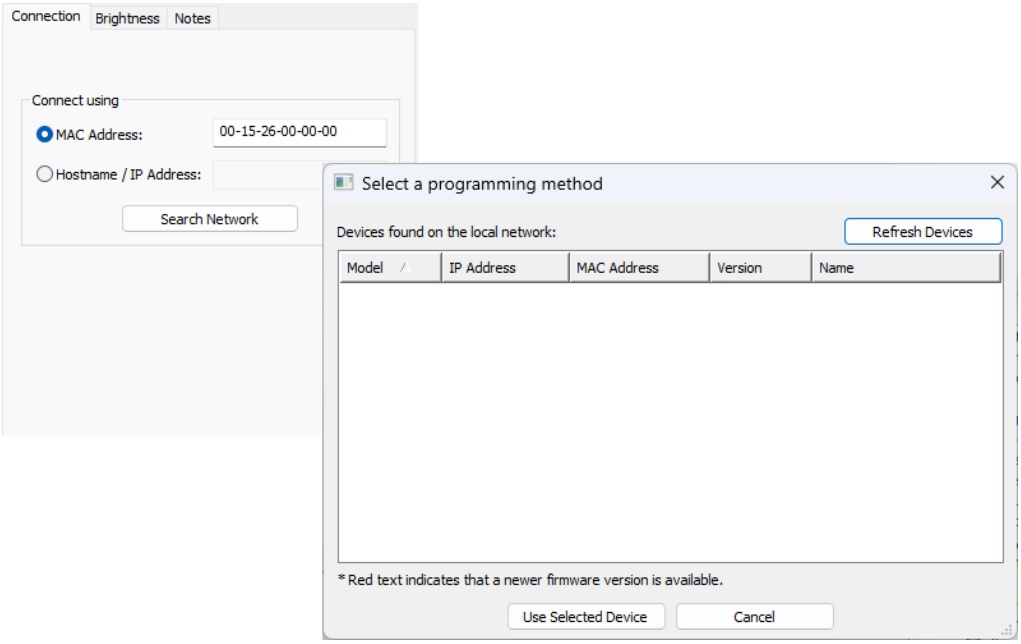
Zigbee control and feedback will use Zigbee for control, requiring a Zigbee transceiver.

433 MHz RF Control and Zigbee Feedback send commands one-way via 433Mhz and receive feedback via Zigbee. This method requires a Zigbee Transceiver and a 433Mhz RF antenna.

433 MHz RF Control and no Feedback sends commands via 433Mhz and does not feature two-way feedback. This method requires a 433Mhz RF antenna.

WK2 Properties

CONNECTION



1. For **Connect Using**, select a MAC Address or Hostname/IP Address.
2. Select **Search Network** to discover the device and ensure proper communication.
3. Select the device in the list, then select **Use Selected Device**.

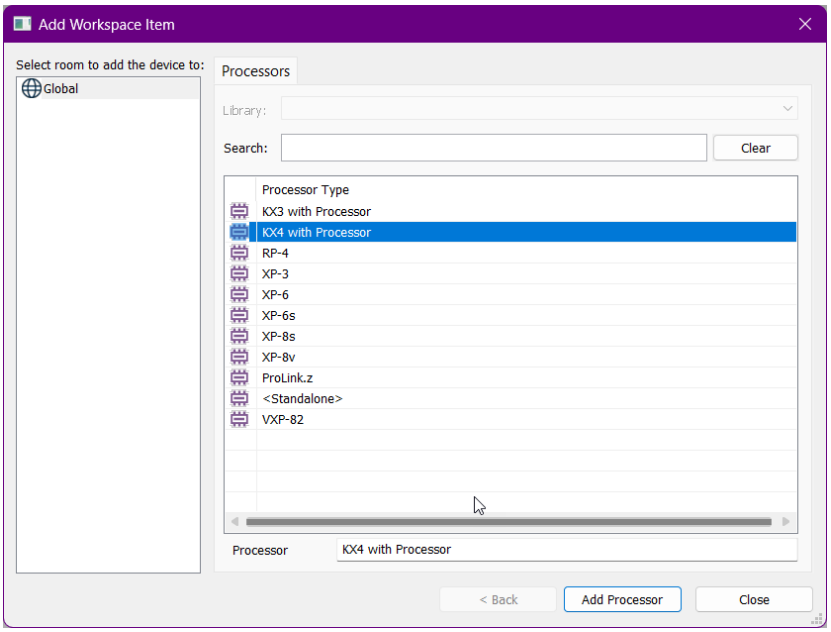
Adding a KX4 Main Processor

The KX4 in-wall keypad features a built-in control processor, ideal for single-room commercial or residential solutions. In addition to supporting two-way drivers, RTIPanel applications, and expansion configurations, the KX4 can control devices directly with built-in IR, RS-232, sense inputs, and relays for an all-in-one control option. It also has a built-in ZigBee transceiver, serving as a control point for RTI ZigBee-based controllers.

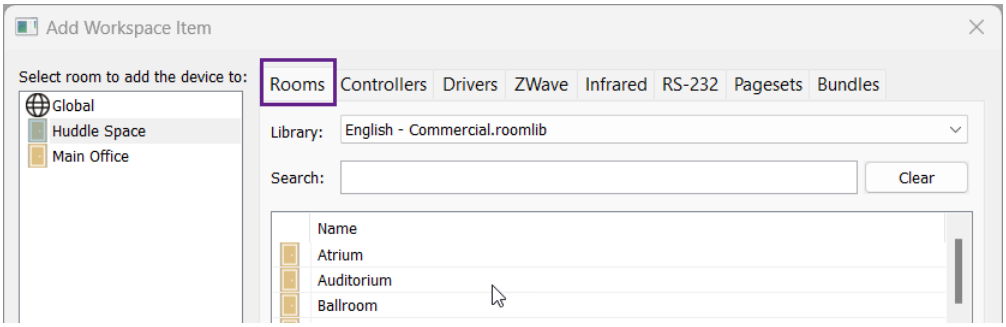
To add a KX4 as a processor to a project:

1. Open Integration Designer and select **File**, then **New System**. Navigate to an appropriate directory and create a name for the project file.

2. The Add Workspace Item will display a list of processors, including a standalone option. Select “KX4 with Processor.” In the field “Processor,” you may provide the KX4 processor with a unique name or use the default name.

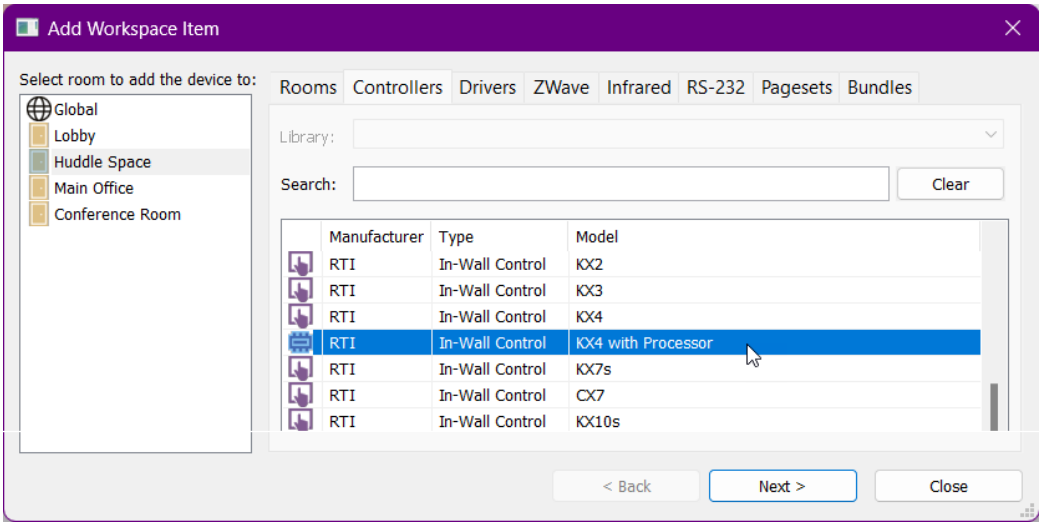


3. Select the **Rooms** tab, then add a room or rooms to your project.

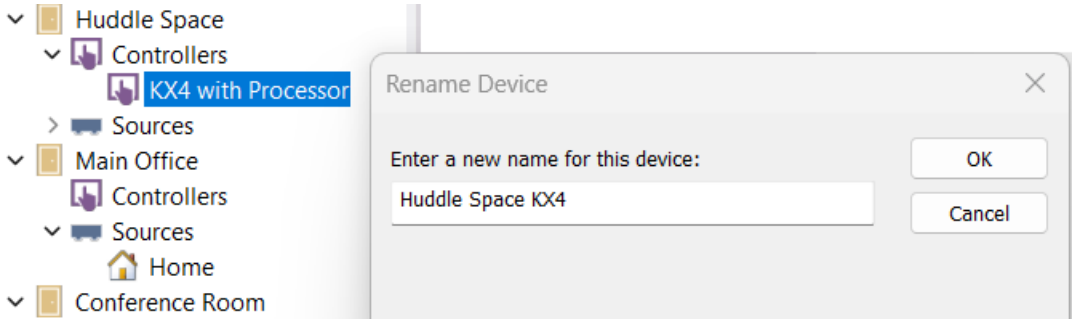


4. Select the **Controllers** tab. The KX4 processor keypad must be added to the workspace as a controller since its primary function is a keypad. Select the controller named “**KX4 with Processor**,” then select the **room** you wish to add the controller. Select **Add Device**.

   Note: If required, the KX4 with Processor may be added to the global area. Please note that the processor is best for single-room projects.



5. Once the controller has been added to the workspace, it can be renamed. Select the controller in the workspace. Right-Click on the controller's name, then select "Rename." Enter a name for the KX4 controller. The name change will also be applied to the KX4 processor in the processor tab.



6. Once a "KX4 with Processor" is added as a controller to the workspace, it will no longer appear under the controller tab. You may continue to add additional controllers, including KX4 controllers, to the project.

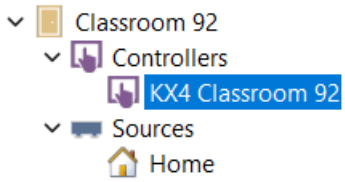
7. Add source devices using the remaining tabs. When controlling components via IR, RS-232, Relay, or requiring sensory inputs, select the available port on the KX4. Please consult the product manual for additional information.

Adding a KX4 Expansion Processor

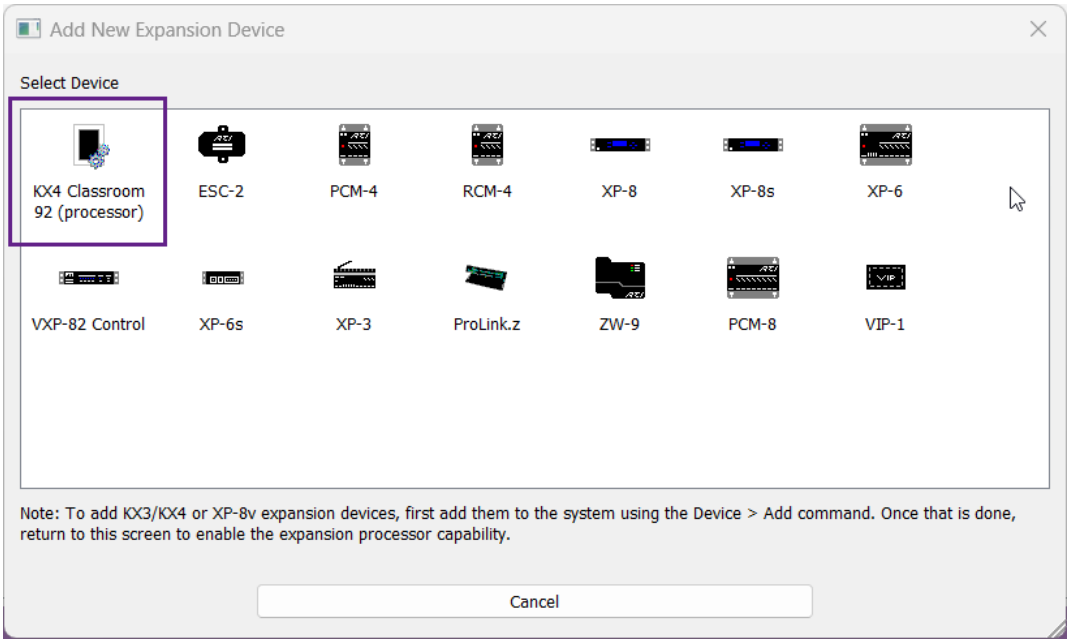
In addition to using the KX4 as a main processor, it may also be used as an expansion processor. When adding the KX4 as an expansion processor, it must first be added as a controller.

1. Add the **main processor** to the **workspace**. Next, add any rooms that are required in the project.

2. Add a **KX4 controller** to the project. Designate a name that is easy to identify and distinguish from other KX4 controllers in the project. Renaming controllers to specific names is a good habit because it makes future processes easier.



3. Select the **main processor** in the workspace, then select **expansion devices**.



4. Any KX4 added to the workspace will be eligible for use as an expansion device. **Select the KX4** from the list of expansion devices required.

   Note: When using a KX4 as a main processor, expansion devices will be limited to other KX4 processors and non-processor expansion devices.

5. You may now route IR and RS-232 through the KX4 expansion processor and the main processor ports. Available sensory inputs and relays for events and macro commands are also possible.

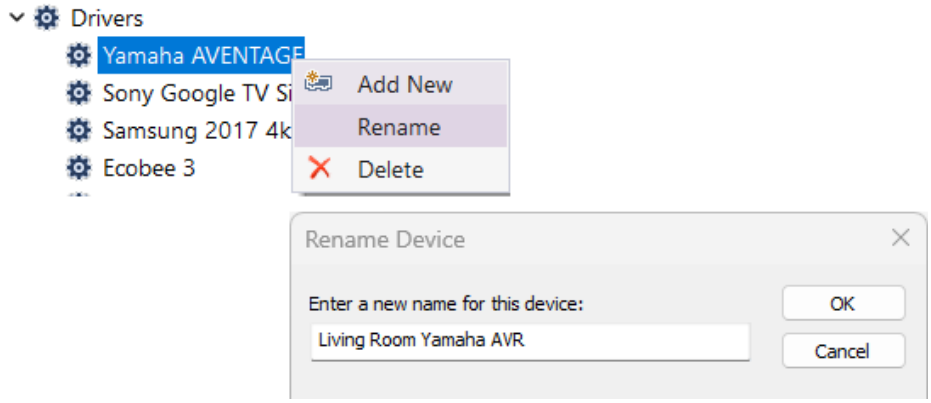
Driver Configuration

Basic Driver Configuration

The driver tab in the workspace details the driver devices added to the Add Workspace Item. When adding a driver, you can designate a source name to the driver, but the original driver's name will be preserved. For example, if you add a driver named "Yamaha Aventage" and name it "Living Room AVR," the source device will reflect the name change, but the driver will still be named "Yamaha Aventage" in the driver tab.

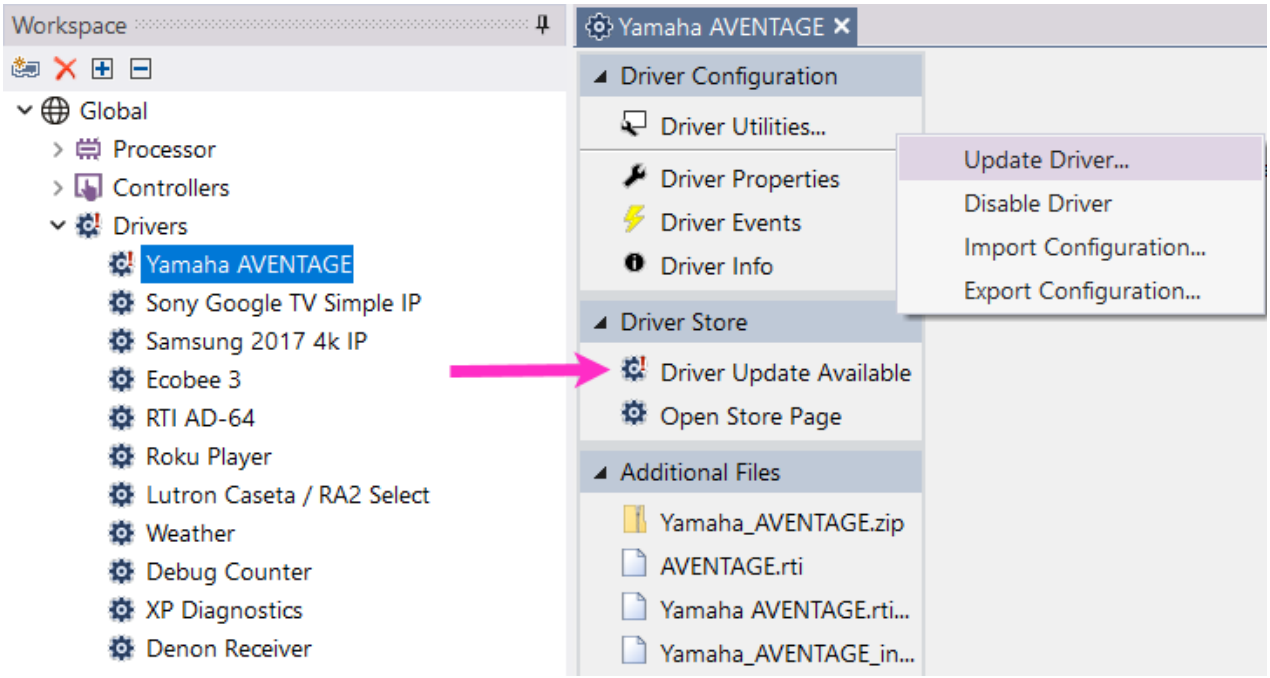
1. Ensure the drivers are exposed by selecting the + expand button or the > button to the left of the driver tab.

2. Since driver names are preserved and not renamed when adding, you may right-click the name and rename it to make it easier to identify. Be aware that some drivers may control multiple items in different rooms.



3. It may be necessary to group drivers by type or preference. Hold the right mouse button and drag the drivers to the preferred order.

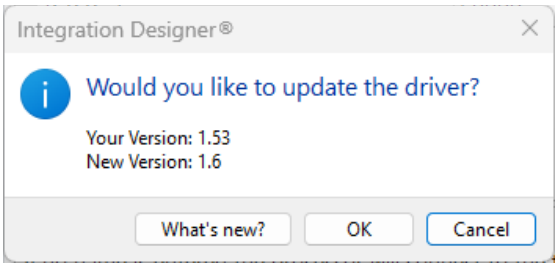
4. Select the first driver on the list. The driver configuration window will display to the right of the workspace.



5. In certain situations, a driver may be out-of-date. Under **Driver Store**, a message "**Driver Update Available**" will appear if a driver update is required.

   Driver Utilities

   Update Driver – Select the "Driver Update Available" message under the Driver Store. A confirmation will be displayed. You may select "**What's New**" for details or **OK** to confirm the update.



   Disable Driver – Select this option if drivers in the project file are temporarily in use or for testing only.

   Import Configuration – You may import driver properties using a previously exported properties configuration.

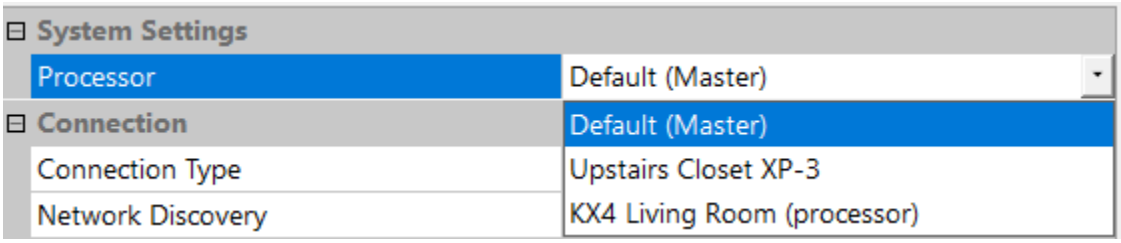
   Export Configuration – To preserve the driver properties, select this option.

6. Select Driver Properties to configure the different driver settings available.

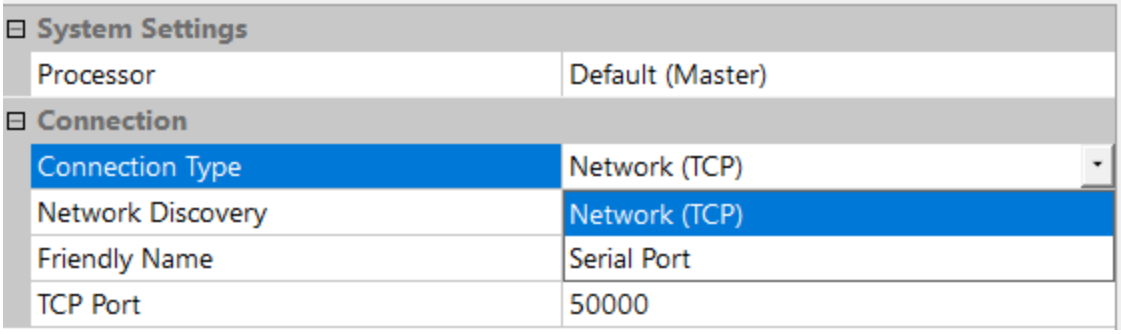
   Each driver may be unique in the configuration options available. Drivers may have the following property types:
   - **Multiple Protocol** features drivers that support IP and RS-232 capability.
   - **Multiple IP Protocol** options are available for certain drivers and can communicate using an IP Address or Discovery method. Discovery options include the serial number, MAC Address, friendly name, or another identifier.
   - **Dynamic Naming** fields store hard-coded information about the different criteria when using a device. For example, lighting load names, the zone(s) name, the available presets, and many others. The dynamic name will make programming easier as it will become part of the command set.
   - **Auto-Configured** drivers can import driver configurations based on setting up the device options in another application.
   - **Login Information** may also be entered for certain drivers that require a username and password.

     Other settings and options may be required. Check the driver information for configuration information.

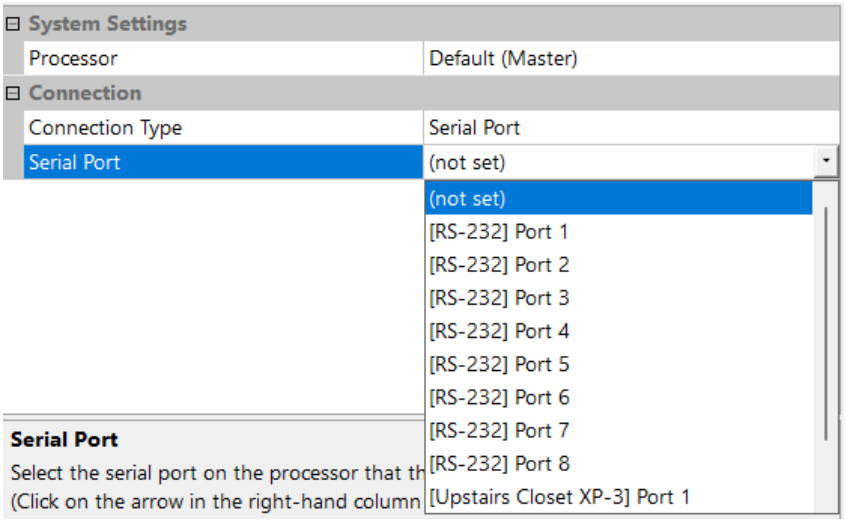
7. Select a **Processor** that runs the driver. Only valid if the system has one or more expansion processors.



8. Select the **Connection Type**. In this example, **Serial** or **Network TCP** may be used.

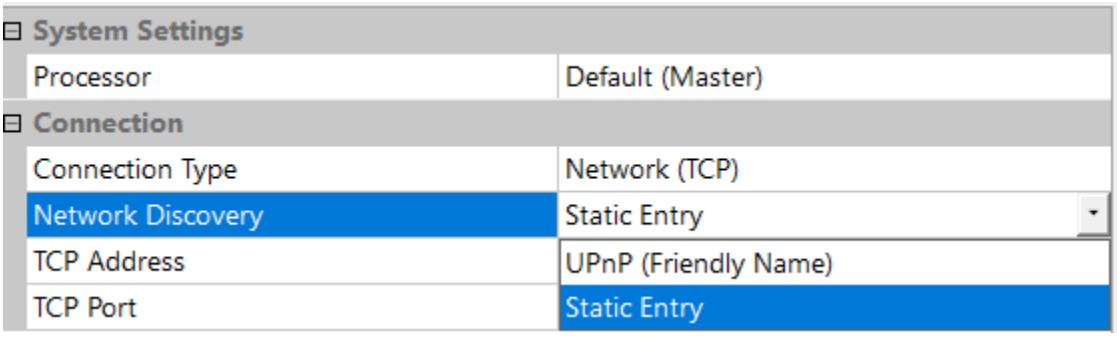


When selecting serial control, you must define the RS-232 processor port to send and receive communication. Select a serial port from the available list.



Selecting the processor in the workspace and selecting Properties in the processor configuration window is good practice. Select the port names tab and change the name of the RS-232 port.

If Network control is required, set the **Connection Type** to **Network (TCP)** and select a **Network Discovery Method**. In this example, a friendly name configured in the device settings or a static IP address entry is available.



9. If using a **Static Entry**, enter the IP address in the IP Address field. If using a **Friendly Name**, enter the device's name as configured in the settings.

System Settings	
Processor	Default (Master)
Connection	
Connection Type	Network (TCP)
Network Discovery	UPnP (Friendly Name)
Friendly Name	MyAVR
TCP Port	50000

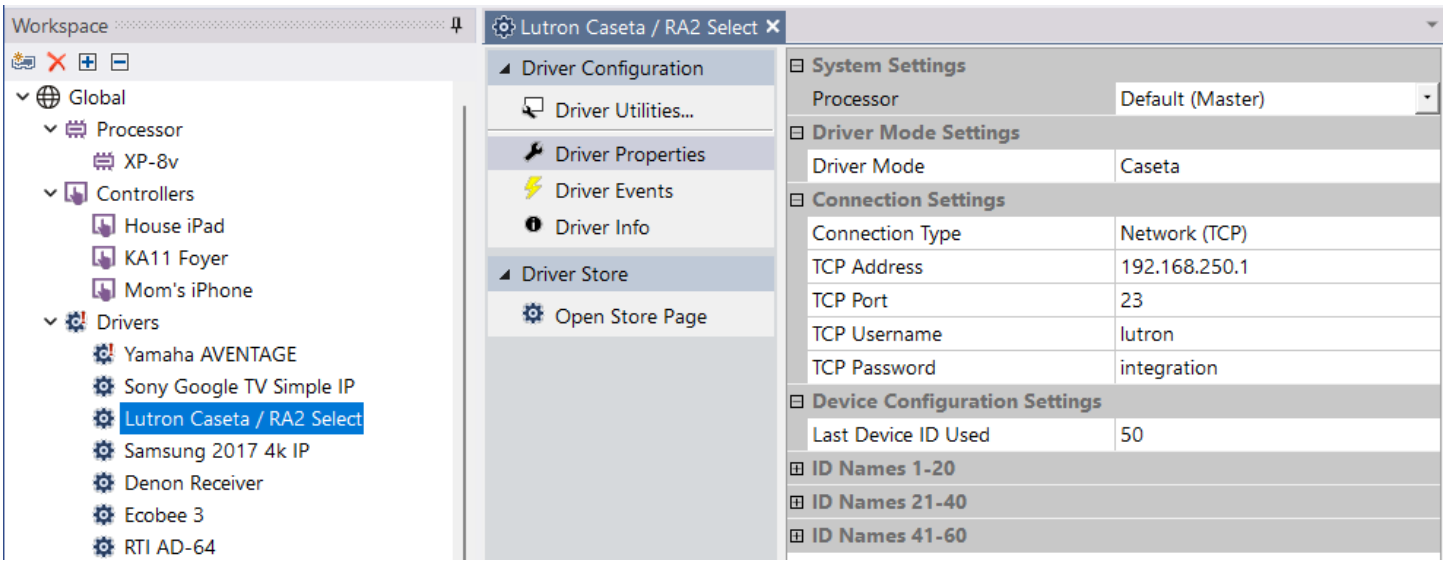
10. Verify the default **TCP Port**. RTI makes every effort to set the default port based on the documentation from the manufacturer. The port may change if the network is administrated or due to a change by the device manufacturer.

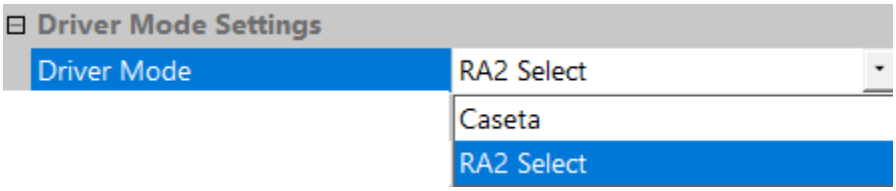
Pro Tip: Read the **Driver Info** included with each driver. Important settings may need to be set on the device to communicate properly with a control system. In addition, there may be features that need a detailed explanation. Driver notes may change with new driver updates, so following the release notes is important.

Configuring a Driver with Dynamic Naming

In certain scenarios, the configuration for drivers will be more complex. For example, you may have to enter dynamic names and telnet information in certain drivers.

1. Select the lighting driver from the workspace and select **Properties** in the driver configuration window.





2. For **Driver Mode**, select the type of lighting control system.

3. For the **TCP Address**, enter the IP address of the lighting controller device. The port name should not need to be changed.

4. Enter the login credentials created on the lighting controller for the **TCP Username** and **Password**. Do not use the default login credentials.

5. Set the **Last Device ID** Used based on the last integration ID. To find the integration IDs, consult the lighting manufacturers' integration report, usually accessible in the lighting controller programming software.

6. Select the **+ plus sign** to the left of the bank of **ID Names** and enter the names of the lighting loads or devices comprising the system. Match the correct ID to the correct lighting load or device.

Device Configuration Settings	
Last Device ID Used	20
ID Names 1-20	
ID 1 Name	ID 1
ID 2 Name	Foyer Chandelier
ID 3 Name	Garden Lights
ID 4 Name	Entrace KEYPAD
ID 5 Name	ID 5
ID 6 Name	Living Room Sconces
ID 7 Name	Living Room Ceiling
ID 8 Name	Kitchen Fixture
ID 9 Name	Office Lamps
ID 10 Name	ID 10

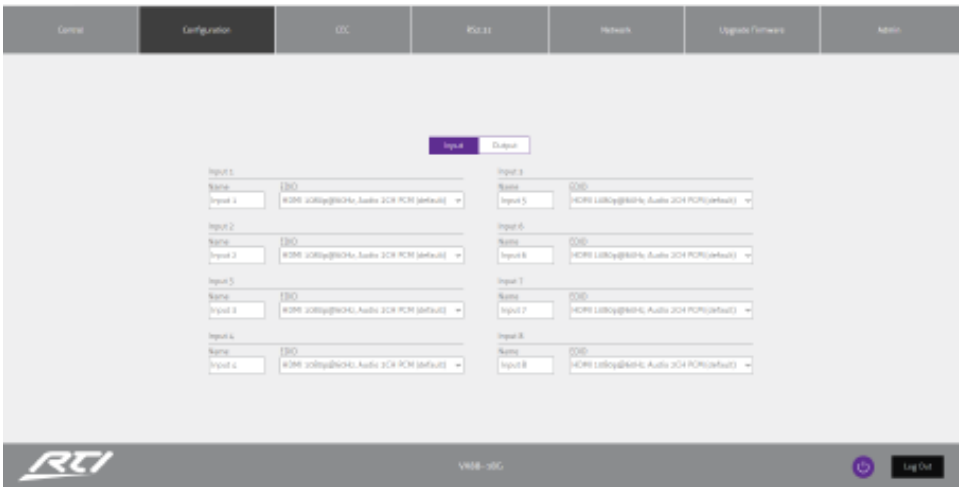
Enter all the lighting loads and devices and leave the default ID name for any non-assigned IDs.

Pro Tip: Dynamic naming in driver properties requires up-front work but makes programming macros, feedback, and events much easier.

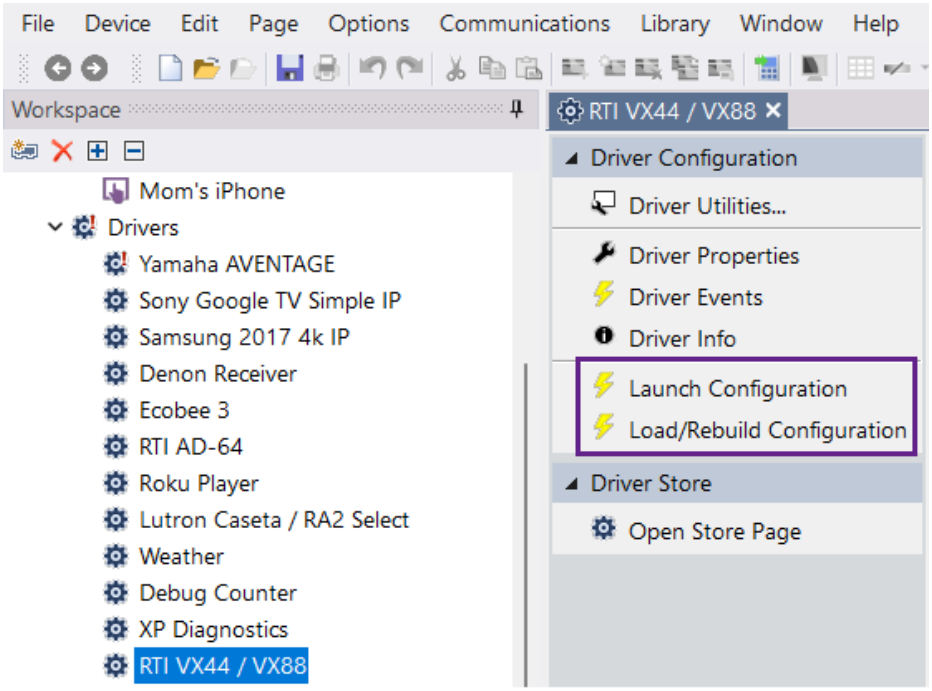
Configuring Auto-Discovery Drivers with Multiple Zones and Sources

Drivers may have auto-configuration capabilities. In these situations, settings and requirements are preconfigured in a separate application and auto-imported into the programming software. In this example, auto-discovery will populate the driver settings, and multiple outputs and inputs will be added manually to the workspace.

1. Configure the web interface or configuration software of the device.



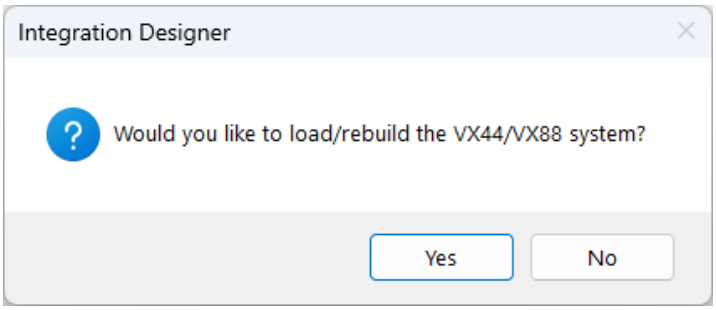
2. Go to the Add Workspace Item and add the driver to initiate communication between the software and the device platform.



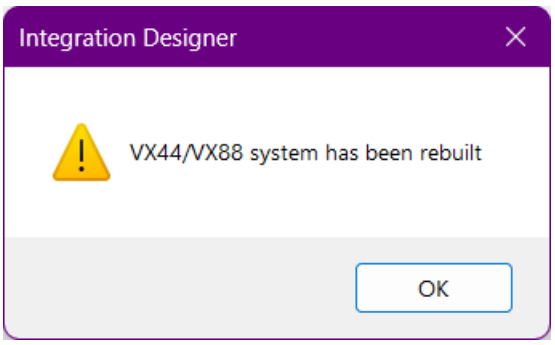
The matrix switch driver is added to the global area as a shared source device.

3. For drivers with auto-discovery and configuration options, you will notice **Launch Configuration** and **Load/Rebuild Configuration** in the driver configuration. Access to the device setup is possible by selecting Launch Configuration. Once the device is set up properly, **select Load/Rebuild Configuration** to import the outputs and inputs into the driver tab in the workspace.

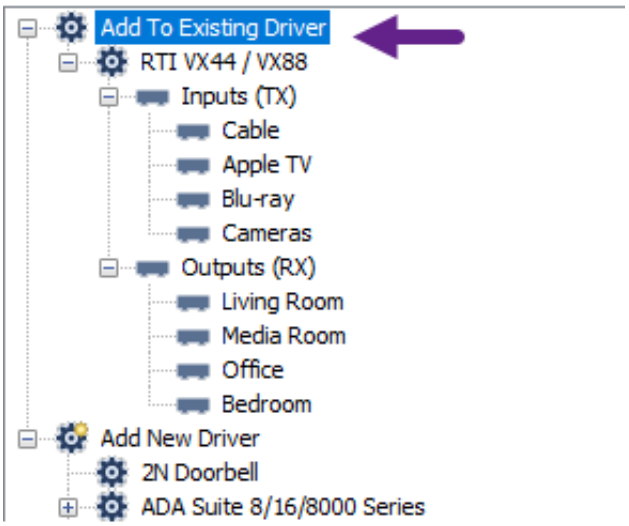
4. A confirmation window will display. Select **Yes** to confirm.



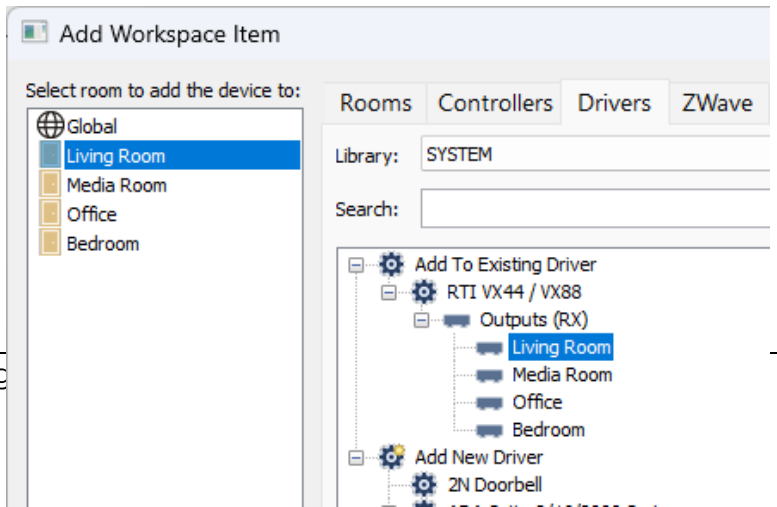
5. An alert window will display after successfully rebuilding the driver by importing the matrix inputs and outputs. Select **OK** to continue.



6. Return to the **Add Workspace Item** and review the driver configurations added under the Add to Existing Driver section in the driver tab. Open the **Inputs (TX)** and **Outputs (RX)** menus to ensure the devices are populated and named properly.



7. Since each input or source device is shared and interfaces are required on several local room controllers, they will be added to the global area. Add each input source to the global area until no more inputs are available.



8. Add the **Outputs** for the matrix to the local rooms in which they are designated.

After all inputs and outputs are added, the driver should have no other options.

You can make changes on the device and reload the driver options at any point.

Pro Tip: When adding the inputs and outputs, you build the driver commands available in Integration Designer. In many cases, pages need only be created if you use the matrix for CEC control and require those interfaces.

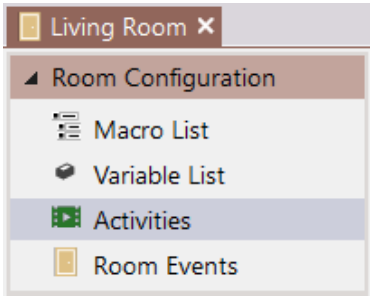
Configuring Activities in System Manager

With its adaptive auto-programming intelligence, the System Manager adds feedback for rooms, devices, and automation, allowing the project to share programming and saving valuable time. Once activities are programmed, the RTI installer can design powerful user interfaces with abundant tracking feedback.

Pro Tip:

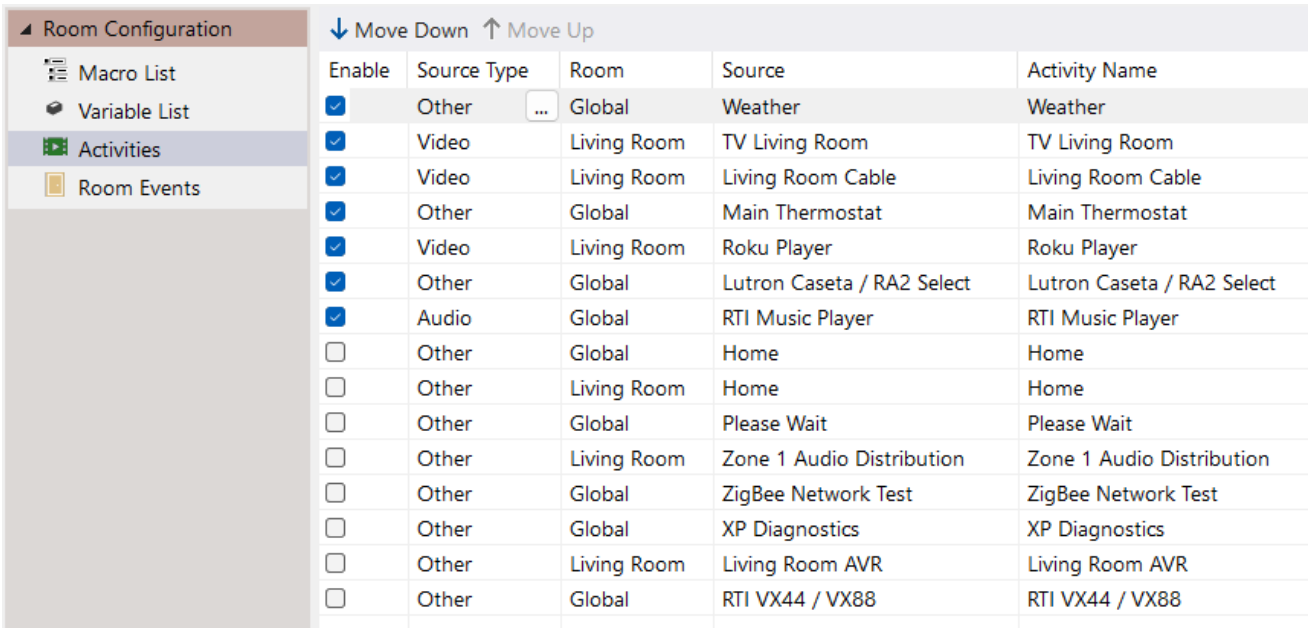
Integration Designer features the "Activities" feature to help program Macros or Events efficiently. Although there are substantial benefits to using the System Manager, some projects may call for a different approach. The System Manager may be bypassed when it is not an ideal project solution. RTI Installers may forego some intelligence and time-saving features but gain additional flexibility.

1. Select any room in the workspace to program the activities in a project. Try to start with the first room in the workspace and work your way down.



2. Select **Activities** from the **Room Configuration** menu.

A view of the activities and devices available in the **System Manager** for the selected room will be displayed to the right of the room configuration window.

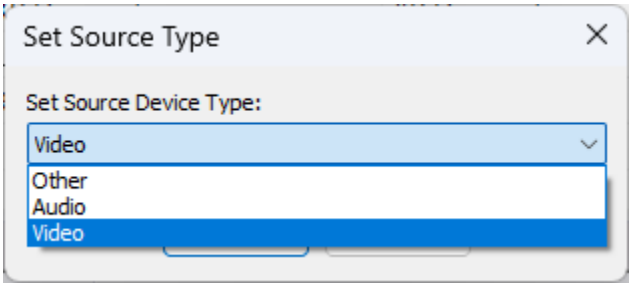


Enable	Source Type	Room	Source	Activity Name
☑	Other	Global	Weather	Weather
☑	Video	Living Room	TV Living Room	TV Living Room
☑	Video	Living Room	Living Room Cable	Living Room Cable
☑	Other	Global	Main Thermostat	Main Thermostat
☑	Video	Living Room	Roku Player	Roku Player
☑	Other	Global	Lutron Caseta / RA2 Select	Lutron Caseta / RA2 Select
☑	Audio	Global	RTI Music Player	RTI Music Player
☐	Other	Global	Home	Home
☐	Other	Living Room	Home	Home
☐	Other	Global	Please Wait	Please Wait
☐	Other	Living Room	Zone 1 Audio Distribution	Zone 1 Audio Distribution
☐	Other	Global	ZigBee Network Test	ZigBee Network Test
☐	Other	Global	XP Diagnostics	XP Diagnostics
☐	Other	Living Room	Living Room AVR	Living Room AVR
☐	Other	Global	RTI VX44 / VX88	RTI VX44 / VX88

3. In the **Enable** column, select the activities required in that room. Selecting an activity will move it toward the top with other selected activities. If you use list menus or button menus, enabling the activities required for that room is good practice. If an activity is disabled, there is no need to configure other steps.

   Special Considerations: *Enabling multiple home pages is necessary if the room you selected has local controllers and a global controller that accesses that room. The controllers will jump to their respective pages when multiple home pages are enabled.*

4. Select the type of device for the **Source Type** column from the dropdown menu. A device is either used for video, audio, or other.



Video devices are devices that require a video display.
Audio devices do not require a video display and are mostly music activities.

Other devices are source devices that are not activity-based and require page jumps.

5. The **Room** and **Source** columns are informational only and display the source's name and the location where it was placed in the workspace. This information can help differentiate between local and shared devices.

6. The **Activity Name** column is where the Activity name can be viewed and changed. Rename each Activity when appropriate to a user-friendly name. Doing so will not impact the source name in the workspace. When using list menu options when adding controllers, the activity name will appear in the list.

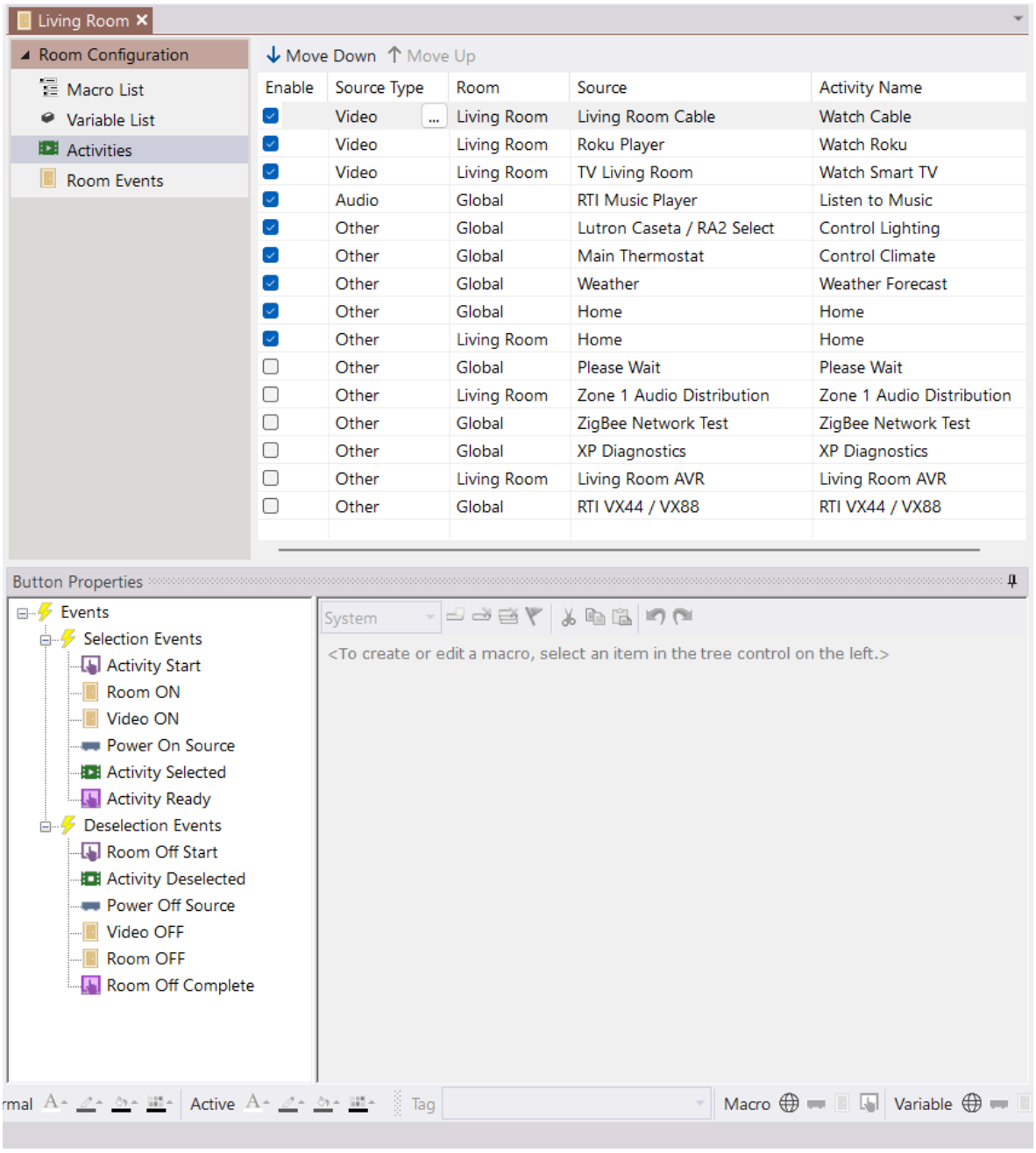
7. You may use the **Move Up** and **Down** buttons on the top of the activity page to change the order of the activities in the list. It is best practice to sort the activities from top to bottom based on the client's preference.

   Pro Tip: The system manager's activities and settings will benefit list menu controllers more than button menus. However, it is best practice to set up the activities regardless of your selected menu. All necessary work is done if you add a controller with a list menu later.

Programming Selection Macro Events

Once you have enabled activities, set them to the correct source type, rename them logically, and order them according to preference, the activity macros may be programmed. Start with the first Activity in the list of enabled devices.

Selection Events will run when an activity is selected in the controller's list or button menu. **Deselection Events** will run when another activity is selected or the room is powered off. Both event types have different steps that run in the order they appear and will be described next.



ACTIVITY START

1. Select the first Activity in the list of enabled activities. You will now begin programming the necessary Macro Steps for each Activity in the form of selection events.

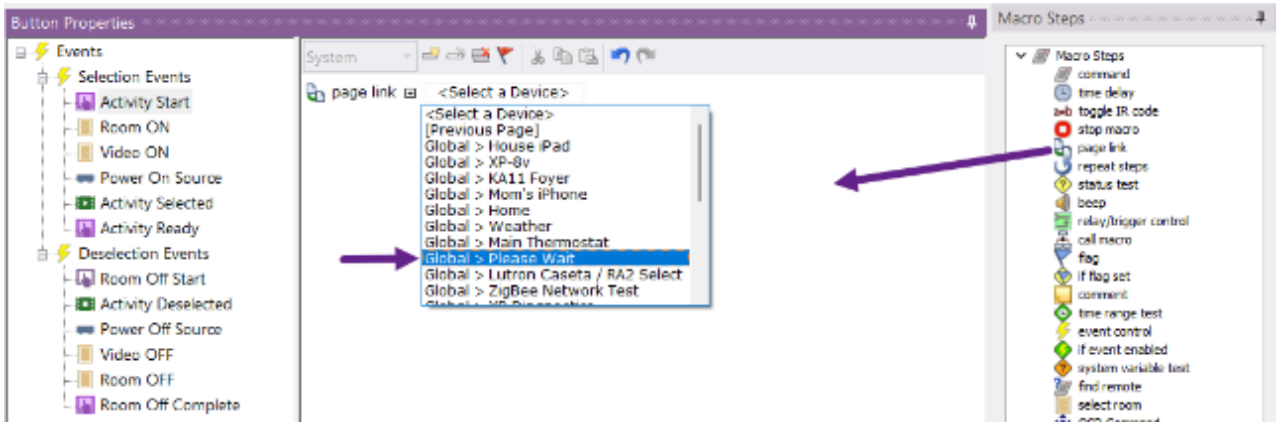
2. Select the first selection event, **Activity Start**. Like all events, if there are no commands in the macro window, the yellow text will display instructions for using this event. Activity Start is the first step run reserved for any "Please Wait" pages. If you have created this page in the workspace, create a page link for it.

<Drag macro steps here>

This macro is run whenever any activity is selected, but before any changes are made.
You can use it to send the controller to a "Please Wait" page.
The macro is common to all activities in the room.

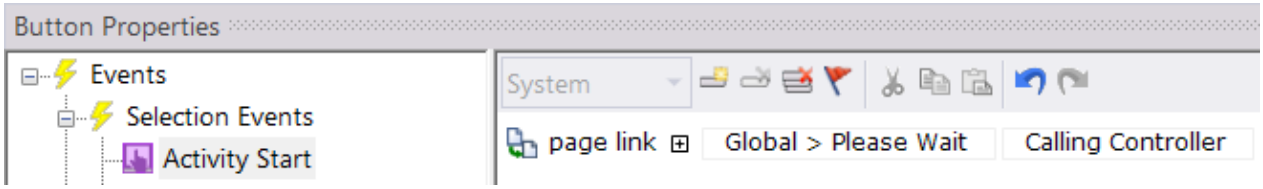
Since this macro is common to all activities in the room, once you program it for the first activity, it will populate the rest of the activities with the wait page.

3. Double-click or drag a **page link** from the macro step tab in the tabbed menu on the bottom right of the screen. Select the **<Select a Device>** section of the page link and select the "Please Wait" page from the available page list.



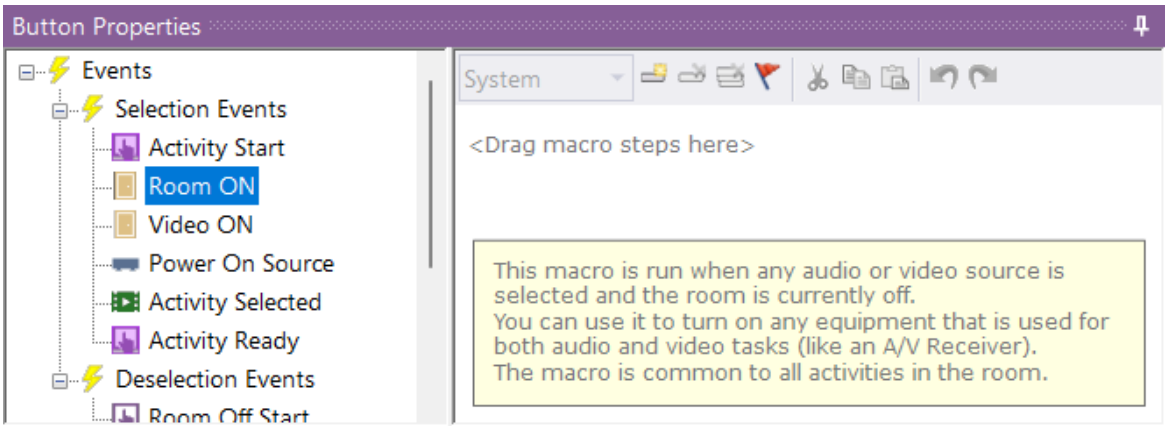
The page link will now jump to the page selected based on the **Calling Controller**. The Calling Controller is the controller that is accessing this event.

If no pageset is available, you may skip this step.



ROOM ON

1. Select the next event, Room ON, which will run after the controller jumps to the "please wait" page.

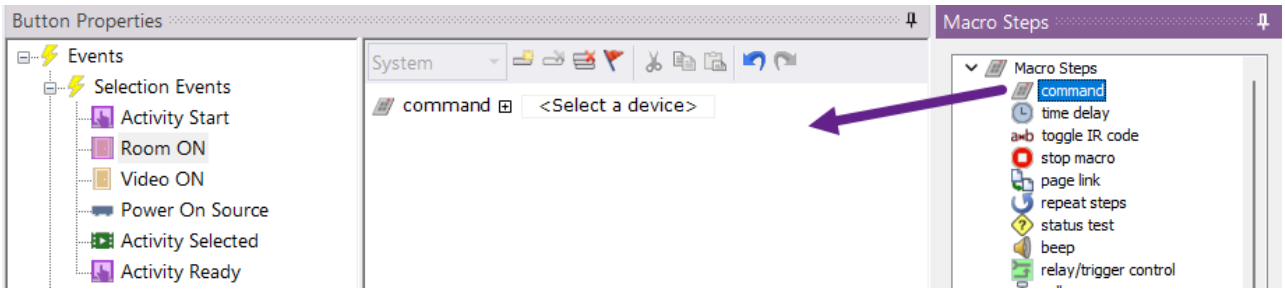


2. The **Room ON** event will run when the end user selects a **video** or **audio** activity and the room is currently **off**.

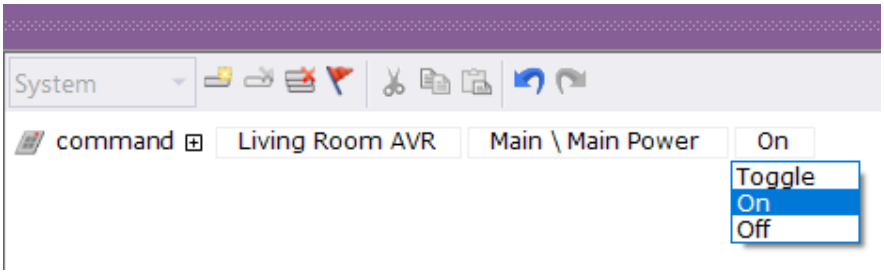
   Identify any devices in the room necessary for both audio and video. Turning on an AV Receiver is common, but only if audio and video activities require the receiver to be on. The **Room ON** event is common to all activities, so programming for one audio or video activity will populate it on the rest of the activities in this room.

   Since the audio and video activities require the receiver to be on in this room, we will add a command to turn it on in this step.

   Drag the Command macro step to the Room ON event macro window.



3. Select the **<Select a Device>** next to the command and select the Living Room AVR from the dropdown menu. Then select the main power commands, then the **on** command from the dropdown menu.



This step may be skipped if there are no common audio and video devices in this room.

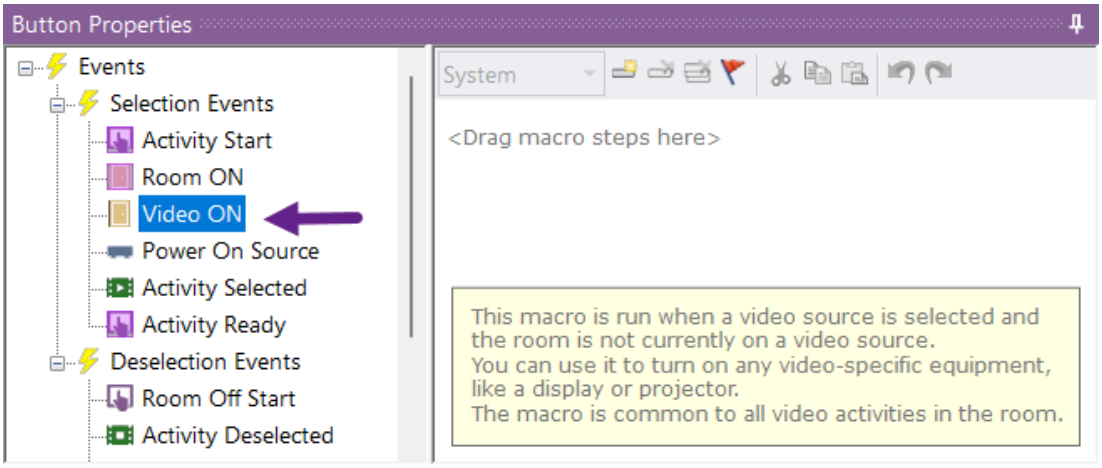
Pro Tip: Each section of this command will have a dropdown menu allowing you to change the device, command, and arguments. Some driver commands may have a longer command tree. In this example, renaming the driver to "Living Room AVR" from "Yamaha AVENTAGE" in the workspace made it easier to identify the device.

VIDEO ON (Video Devices Only)

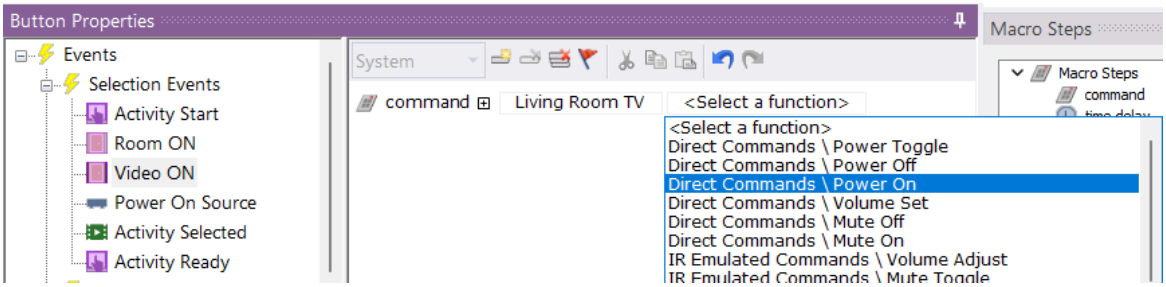
The Video ON event is available on video devices and will run when the room is not using a video source. The room may be off, or the end user could use an audio activity.

Use this to turn on video-specific devices like a television or projector. The Video ON event is common to all devices, so once you program it for the current activity, all video activities will not need to be programmed with the same command.

1. Select the **Video ON** event in the button properties window.

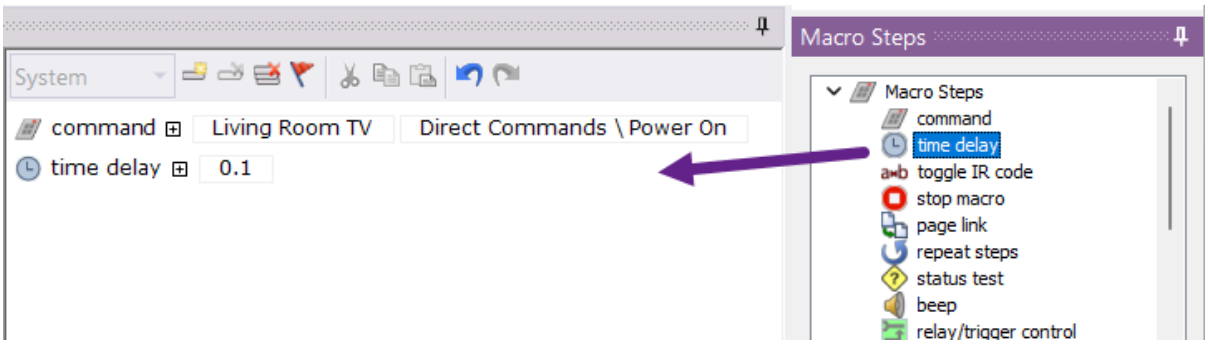


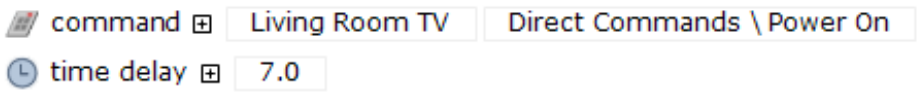
2. Drag or double-click the **command** Macro to the macro window. Select the Living Room Television driver from the dropdown menu. From the function section of the macro, select **Power On** from the direct commands category.



3. Some devices require a brief delay after they are turned on to give them time to warm up. If the video device requires a delay, you may enter it after the command.

   Select the time delay in the Macro Steps window. Set the delay interval to the appropriate time. The delay can be set in seconds and tenths of a second.





The time delay stop is optional and may differ depending on the device.

Pro Tip: The video ON event will only run if the room was previously off or using an audio activity, so the delay will only be sent once. If the end user selects another video activity, the Video ON event will not run and will not subject the end user to an unnecessary delay.

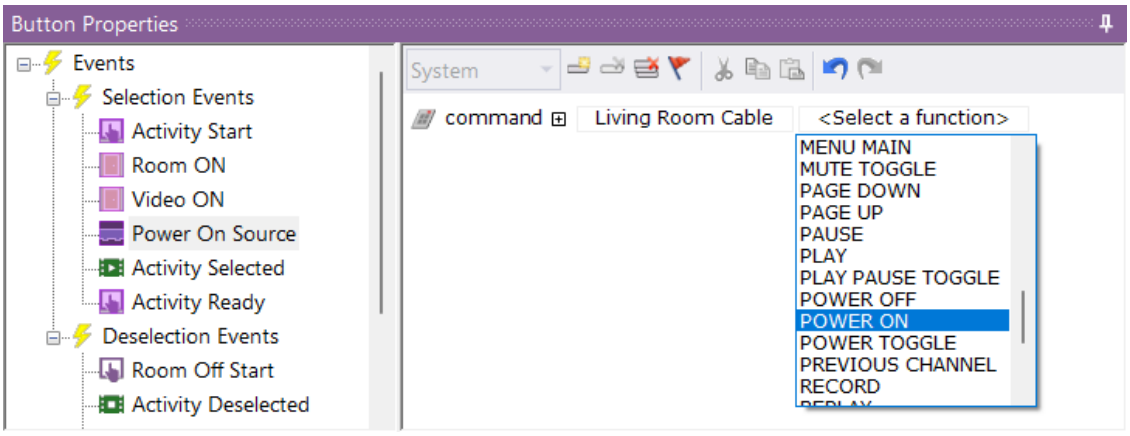
POWER ON SOURCE

The **Power on Source** event will turn on the source device when the first room in the project selects that activity. Since activities may be shared, such as a cable activity utilizing a matrix switch, the Power on Source will track the device's power status internally. If the device for the activity has power features, you may take advantage of this event.

1. Select the **Power on Source** event type in the button properties window.

2. Drag or double-click the **command** step in the Macro Steps window.

3. Change the device to the Living Room Cable, then select the function command from the alphabetical list of power commands.



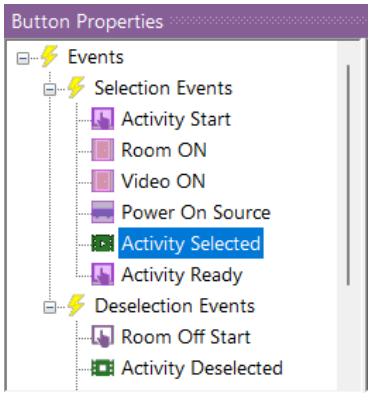
You may skip this step if the device does not require turning on or off.

Pro Tip: IR library commands are usually less extensive than driver commands and are listed in alpha order. Select the command's first letter in the macro window's function section to navigate quickly to a command.

ACTIVITY SELECTED

This type of macro event configures the equipment in a room to use the source device. This macro will run after all the ON events have run. Use this to switch inputs on video displays, receivers, matrix switchers, audio distribution devices, or anything with an input selection.

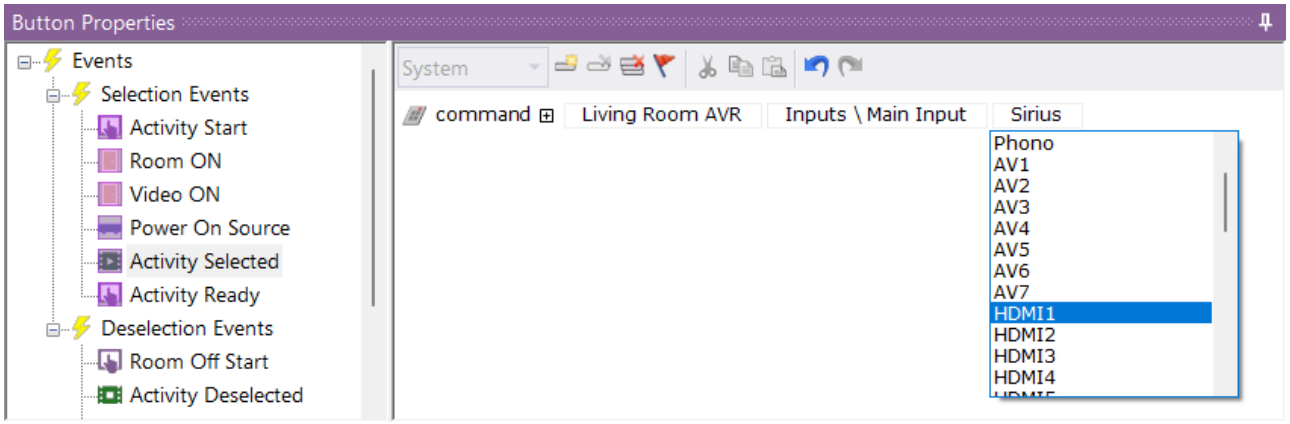
1. Select the **Activity Selected** Event type in the button properties window.



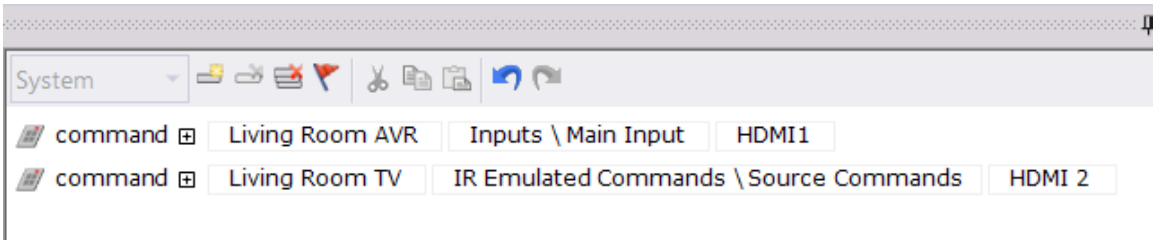
Notice that the populated macro events have a purple square border, indicating they are populated.

2. Drag or double-click the command in the Macro Steps window. Select the first device that needs to switch to an input. In this example, it will come first since the receiver warms up quicker than the video display.

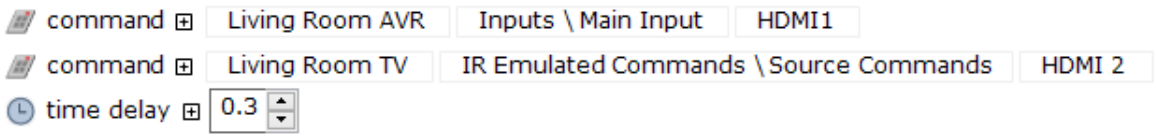
   Select the first function to the input command, select the next function, and select the proper input.



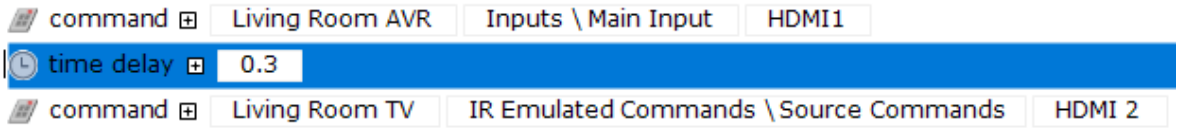
3. Select the next device and input for the next input change using the same procedure.



4. Putting a short delay between the commands in certain situations may be necessary. If you require a delay, drag or double-click the time delay Macro Step. Change the delay by selecting the time interval and either using the up and down arrows or typing in the delay manually.

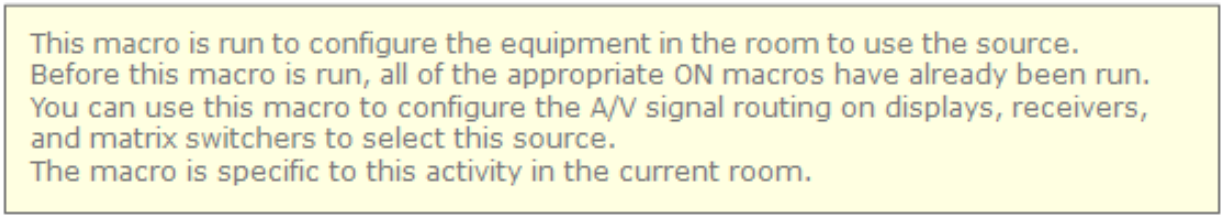


5. You may reorder the macro by selecting the command icon on the left side and dragging it to the right order.



6. Add any additional commands for configuring the source device.

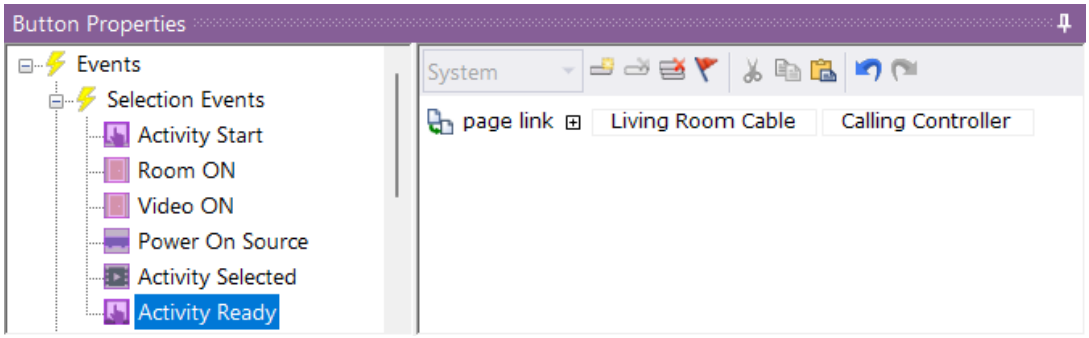
Pro Tip: It is best practice to use the macro event types as they were designed. Do not deviate from the instructions in the yellow highlighted text at the bottom of the macro window. Following the instructions will create a better client experience.



ACTIVITY READY

The last event macro, Activity Ready, will change the controller to the user interface for the device to be controlled. All other macro event types have been run to this point, and the system is properly calibrated to use the activity.

1. Select the Activity Ready macro event and ensure the page link has been automatically populated.



   If the page is not auto-populated, you may not have enabled the activity when configuring the system manager. Verify that the activity is enabled.

2. Make any changes to the page link destination page if desired.

Pro Tip: Selection Events are essential to ensuring the correct macro-order to optimize the end-user experience and only to send the commands necessary when selecting activities. In addition, Integration Designer software will be able to identify the type of device and make feedback available to track several scenarios.

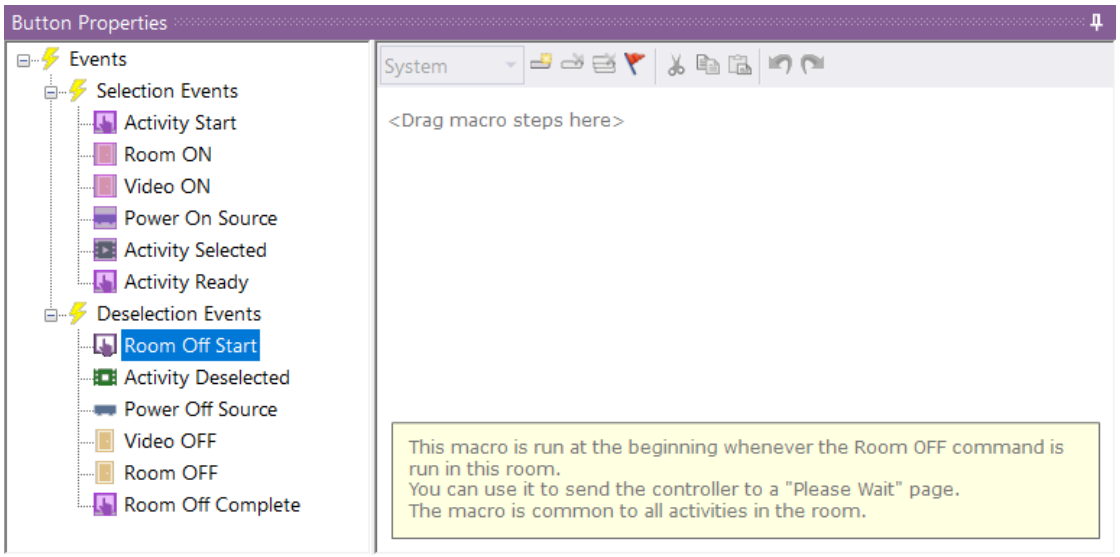
Programming Deselection Macro Events

Deselection macro events occur when the end-user powers off the room or selects another activity. Deselection macros handle the Room Off button built into the Coral template and any accommodations needed when switching activities.

ROOM OFF START

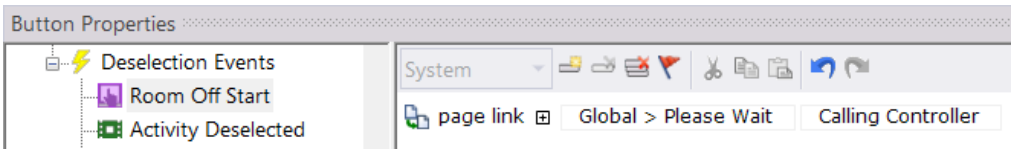
1. Select the first Deselection macro event, **Room Off Start**.

   Room Off Start is the first macro event to run when the Room OFF command is run in this room and, like Activity Start, is reserved for a "Please Wait" page. It is common for all the activities in the room, so you only need to program it once, and the rest of the activities will inherit the page jump.



2. Double-click or drag the **Page Link** step from the Macro Steps window.

3. Select the "Please Wait" pageset from the list of available pages in the dropdown menu.



The end-user will now press the power off button in the room, and the page will display on the screen while the rest of the deselection macros run. You may skip this step if you do not have or plan to use a "Please Wait" pageset.

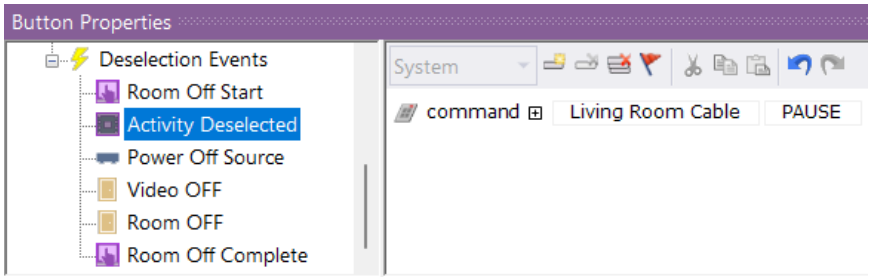
Pro Tip: Creating a "Please Wait" pageset for clients may be necessary to prevent them from pressing buttons before the system is ready. They are only sometimes necessary and should be used based on the client and project specifications.

ACTIVITY DESELECTED

The **Activity Deselected** macro event will run when another activity is selected or the system or room is turned off. While this step is rarely used, it may serve a practical purpose, such as pausing or stopping a player.

1. Select the **Activity Deselected** deselection event type in the button properties window.

2. Drag a command from the Macro Steps window and select a device and functions should you use this event.



In this example, the Cable DVR will pause a show if the end-user selects another activity or turns off the system.

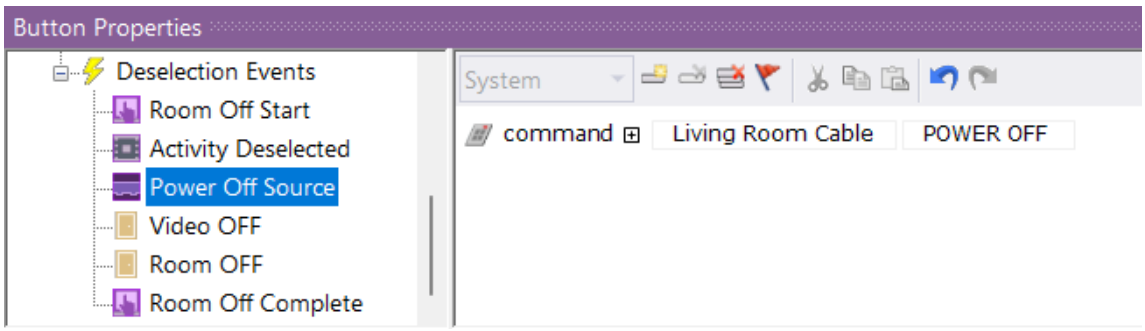
POWER OFF SOURCE

The Power Off Source macro event will run when that last room is not using this source. For example, if the Living Room and Kitchen use the same cable activity,

the last room to turn off will turn off the cable box. The system manager's built-in tracking will avoid conflicts and interruptions with shared devices.

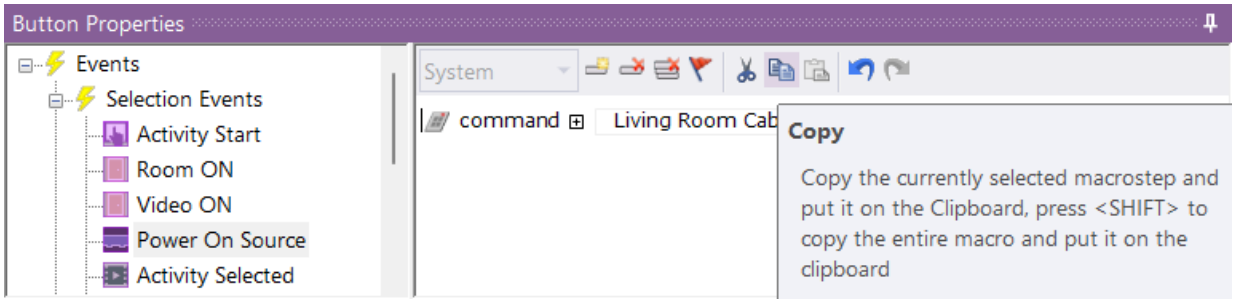
1. Select the Power Off Source deselection event macro.

2. Double-click or drag a command function into the macro window and change the device and function to power off the source device.



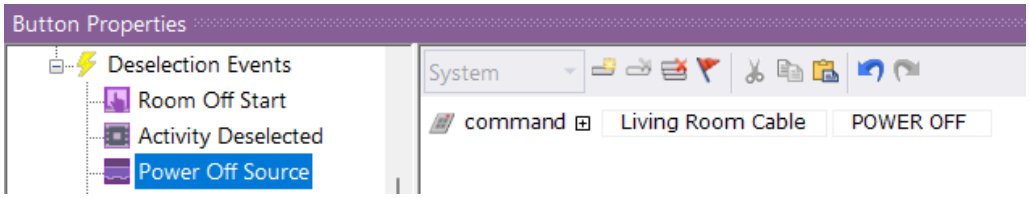
Pro Tip: Since the Power Off Source deselection macro is typically the opposite of the Power on Source selection macro, you can copy and paste the Power On Source macro to the Power Off source macro and make any necessary changes.

- Go to the **Power On Source** selection event macro command.
- Click and ensure the cursor is to the left of the command icon.
- Select the copy icon in the macro window toolbar. You can select a single command or hold down **SHIFT** to copy the entire macro set of commands.



- Go to the Power Off Source deselection event macro command.
- Select the Paste icon in the macro window toolbar.

- Change the command to power off the source device.

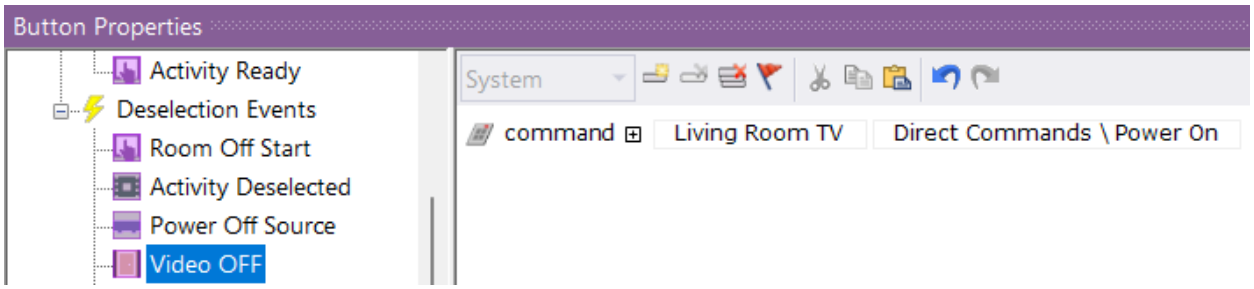


VIDEO OFF (Video Devices Only)

The **Video OFF** event macro is run when a room is powered off, or an audio activity is selected. You can use this step to turn off the power to any equipment turned on in the Video ON event macro. The **Video OFF** macro is common to all activities, so it is unnecessary to program it more than once.

1. Select the **Video OFF** deselection event macro in the button properties.

2. You may double-click or drag a command into the macro window and change it to turn off the video display or copy and paste the command from the **Video ON** selection event macro and change the function.



This macro event will run when an audio activity is selected. If you do not want the video display to turn off when switching to a video source, you can turn it off in the **Room OFF** event macro.

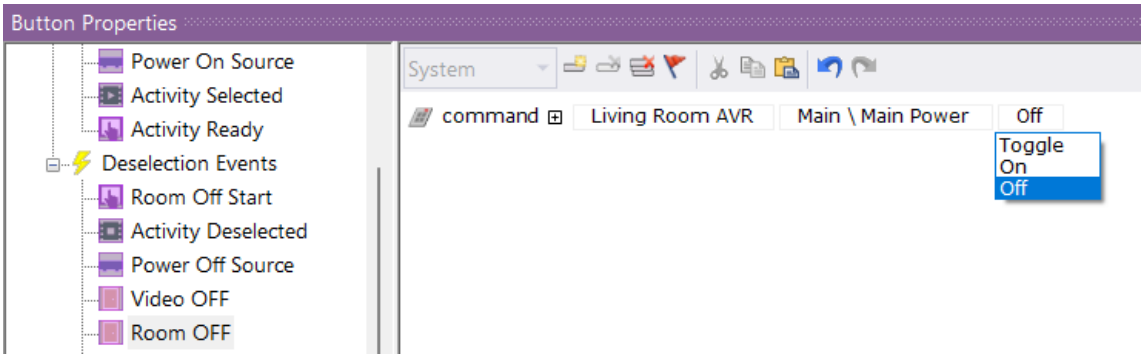
ROOM OFF

The **Room OFF** deselection event macro will turn off any equipment shared with audio and video activities. At this point, the **Video OFF** and the **Audio OFF** macro have already run. Use this macro to turn off the shared equipment powered on in

the Room ON selection event macro. The **Room ON** macro is common to all activities, so it is unnecessary to program it more than once.

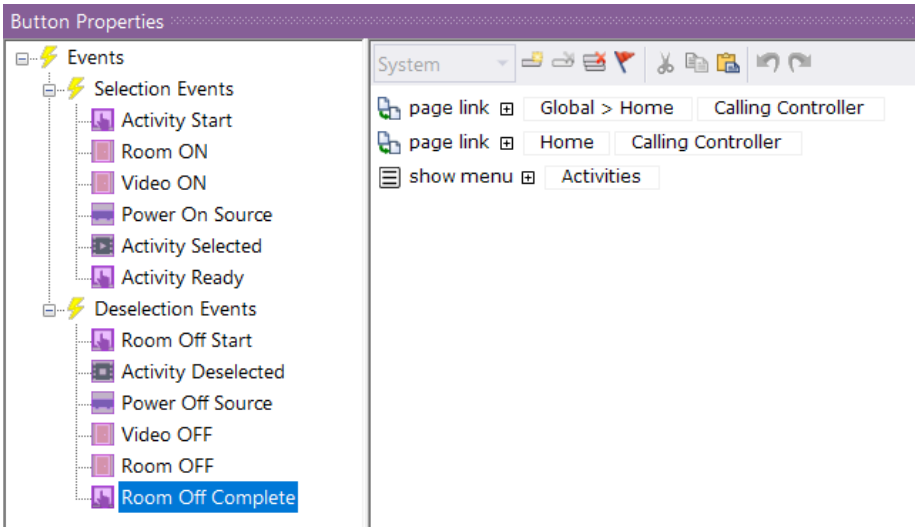
1. Select the **Room OFF** deselection macro event in the button properties window.

2. You may double-click or drag a command into the macro window and change it to turn off the shared equipment or copy and paste the command from the **Room ON** selection event macro and change the function.



ROOM OFF COMPLETE

The **Room OFF Complete** deselection event macro is the last step when powering off the room. It will automatically populate with commands to return the controllers (both local and global) to their respective home screens. A **Show Menu** step will also display the activities for controllers with multiple menus.



These commands can be customized if you wish the controller to jump to another page other than the home page. You can also alter the show menu command or remove it completely. It is recommended to maintain these settings as a beginner.

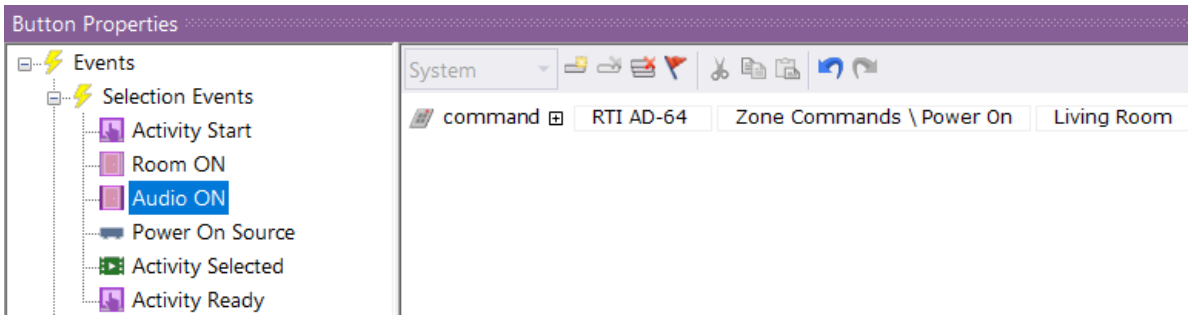
AUDIO ON (Audio Devices Only)

Audio ON is available on audio devices instead of the **Video ON** selection event macro. This macro event will run when an audio source is selected and the room is currently not on an audio source. Use this macro event to turn on any audio-specific equipment. If the audio equipment is shared with video, use the **Room ON** macro. Since this macro is common to all audio devices in the room, you only need to program it once.

1. Select the **Audio ON** selection event macro from the button properties window.

2. Double-click or drag a command from the macro step window.

3. Change the device and the function in the macro window to turn on the audio device.



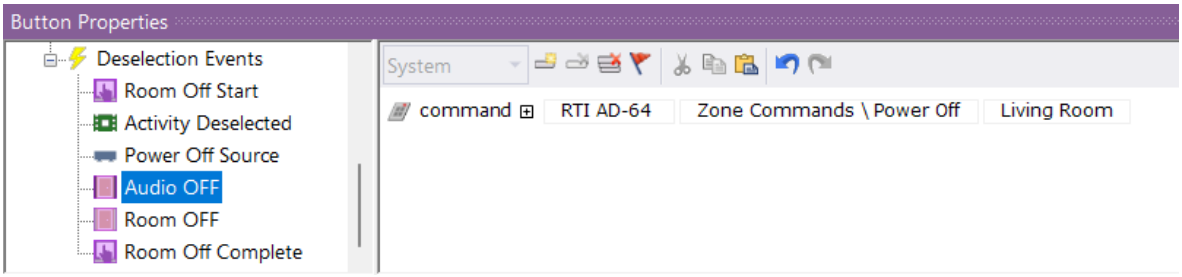
If you wish to turn off a video display when audio is selected, there is no need to program it here. The Video OFF deselection event macro will run when an audio activity is selected and sets the display to off.

AUDIO OFF (Audio Devices Only)

The **Audio OFF** deselection event macro will run when a video source is selected or the room is powered off. You can use this macro event to turn off audio-specific devices powered on with the **Audio ON** selection event macro. Since this macro is common to all audio devices in the room, you only need to program it once.

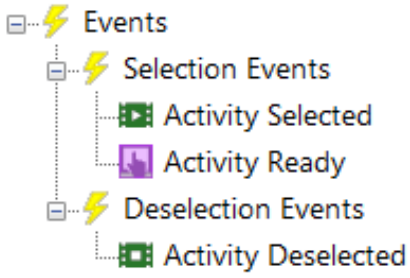
1. Select the **Audio OFF** deselection macro event in the button properties window.

2. Copy and paste the Audio ON selection event macro command to the **Audio OFF** macro or use the command function to turn off any audio-specific devices.



OTHER DEVICES

"Other" source-type devices will have fewer selection and deselection event macros since they do not need audio, video, or source device power commands. Activities such as lighting, climate, shades, security, and others not considered video or audio activities fall in this category. Usually, these macros require a page jump only, but other options are available.



1. Select the **Activity Selected** macro event type if input switching is required. This event type works the same way as audio and video source types.

2. **Activity Ready** should have a page link auto-populated, so there should be no need to program this event type. If the page link is not populated, ensure the activity is enabled for that room in the system manager.

3. **Activity Deselected** is available if you want to run a macro when another source is selected or the room is turned off. Usually, this deselection event macro is not used.

Benefits of System Manager

The system manager's built-in intelligence adds another level of control and convenience to RTI's award-winning Integration Designer software.

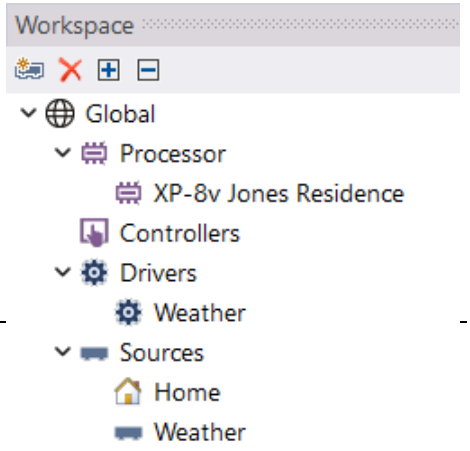
- Integrated Menu Configuration & Navigation Options
- Centralized Activity Macro Programming & Configuration
- "Smart Macros" is designed to send when needed.
- Tracking Variables and Additional Feedback Options
- Time-saving Features & More
- Improved auto-programming features.

Navigating the Workspace

Understanding how to navigate the workspace and find the options you need to make changes or customize your user interface is important. While adding the controllers and source devices to the project, source devices are created in the workspace. RTI installers can add underlying pages to these devices optionally, creating a pre-programmed interface automatically. RTI has pre-configured these pages based on the type of device being added. While not all source devices have page/auto-programming creation options, the RTI installer can manually add and create those interfaces.

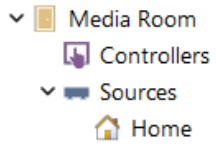
Here is a summary of how sources and underlying pages are automatically created in Integration Designer software:

1. A main processor is added. A global area is created in the workspace, reflecting the main processor. An area for drivers is created, and a weather driver is added to the software. In addition, a receptacle for controllers and sources assigned to the global area is created. A

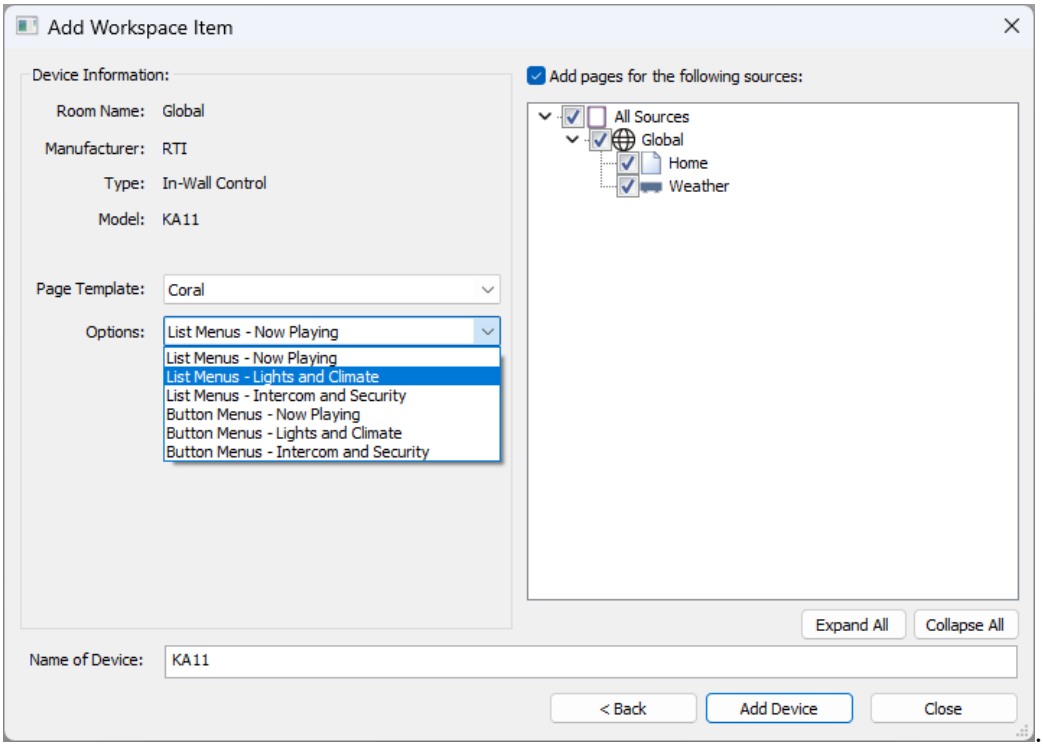


home source and a source for the weather driver are created.

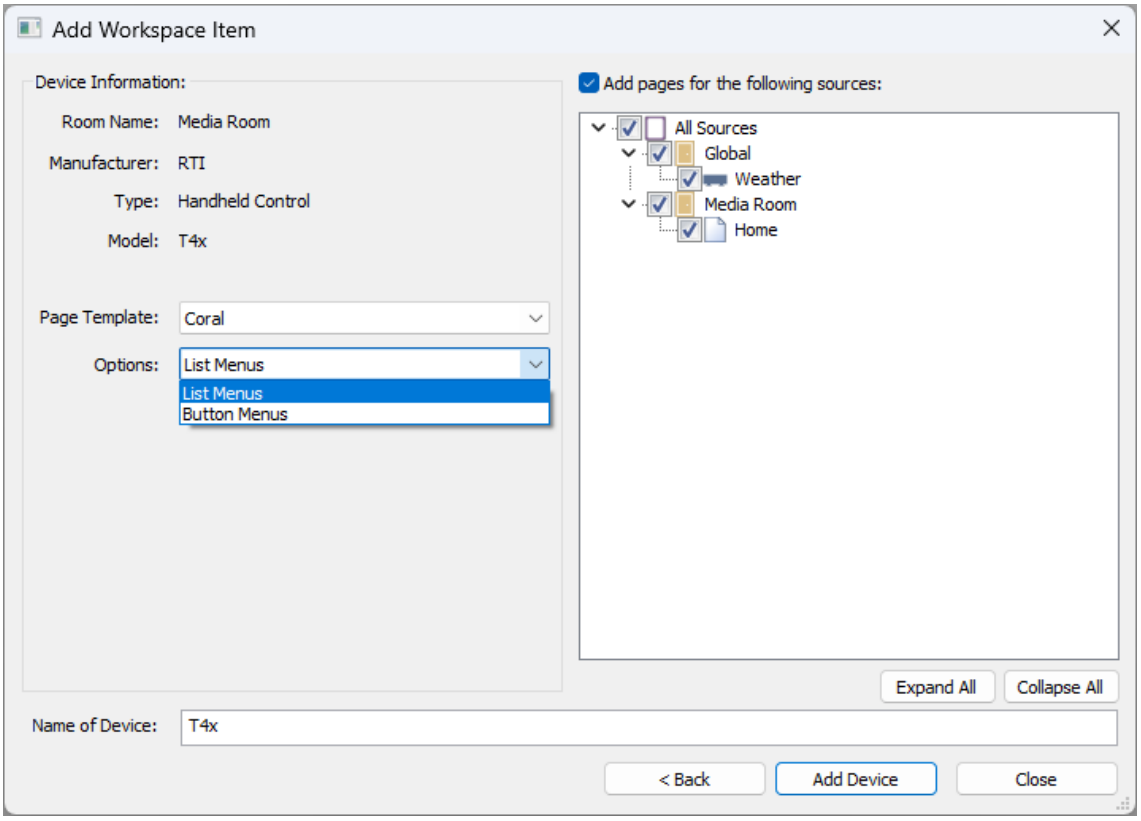
2. A room is added to the software. A receptacle for sources and controllers assigned to that room is created. An additional room has been added.



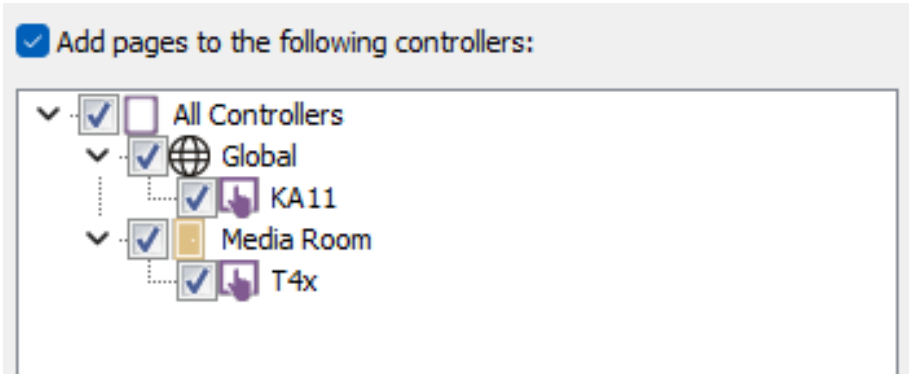
3. A controller is added to the global area as a multiroom controller. The template type and dashboard available on the controller may be selected. In addition, pages for the controller type can be added for the existing sources available in the global area, such as “Home” and “Weather.”



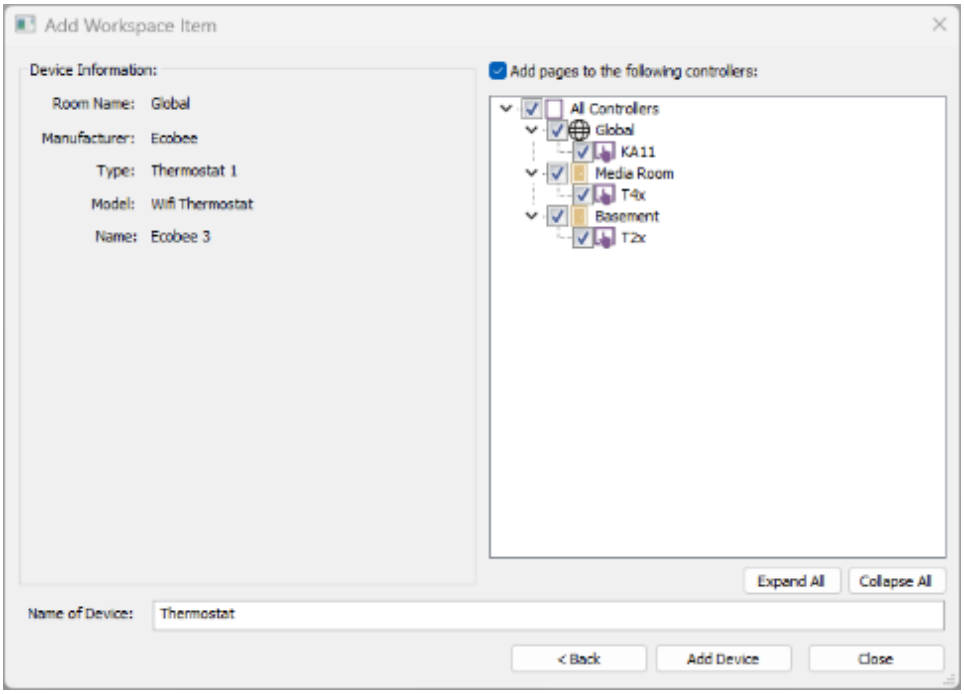
4. A controller has been added to the local room. A template name and menu type available on the controller may be selected. In addition, pages for the controller type can be added for the sources available in the global area (weather) and the sources available in the local room (Home.)



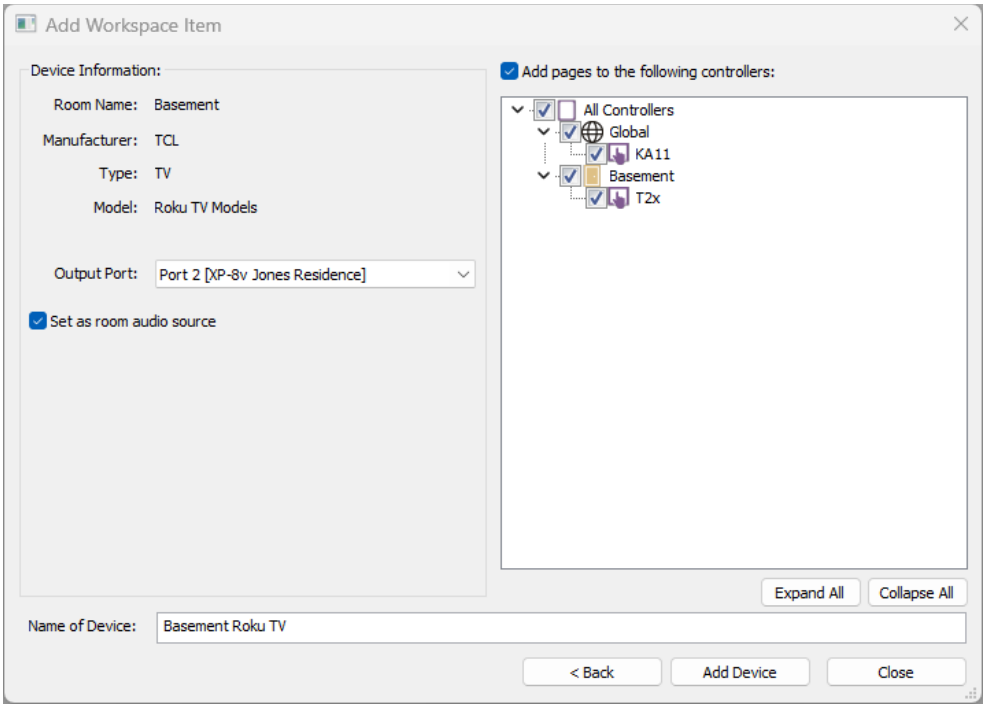
5. A source device is added to a local room. Pages may be added for both the local room controller and the controller in the global area.



6. A source device is added to the global area. In this situation, all controllers are available for page creation.



7. A source device is added to the other local room, and pages are selected for the local room controller and the global controller.



Selecting Items in the Workspace

The workspace now has a collection of rooms, controllers, and source devices available on different controllers based on where the controller is placed.

A glance at the workspace can reveal several key details about the project:

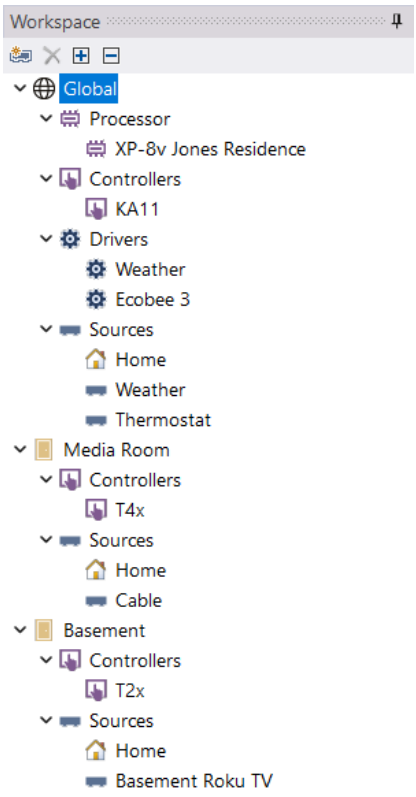
- The main processor can reveal the project's needs, growth potential, and future expandability.

- **Rooms** indicate where the client utilizes the control system in the space.

- **Source devices** placed in the **global** area indicate that it is shared, or may be shared, between two or more rooms.

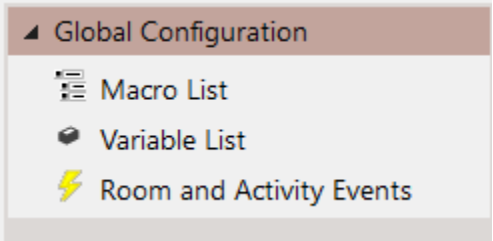
- **Source devices** placed in a **local** room indicate the source device is being used in that room and available on the multiroom (global) controller.

- The types of **controllers** used in the project indicate potential signal management provisions that must be made.



The following options are available upon selection in the workspace:

GLOBAL

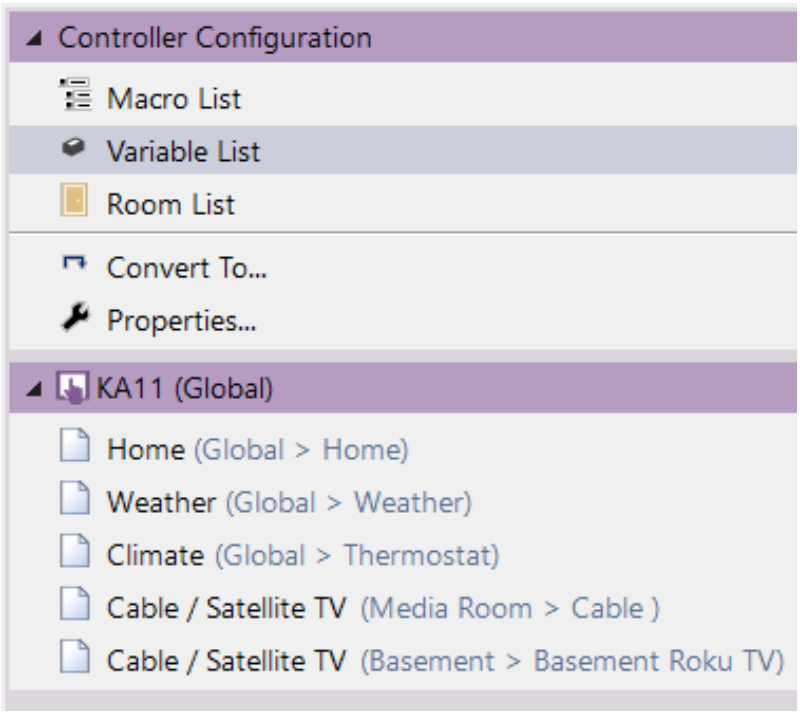


1. **Macro List**: Create global macros that can be called from buttons or events. It also provides an overview of the tags and macros that have been programmed on a global level.

2. **Variable List**: View a list of tags and variables programmed on a global level.

3. **Room and Activity Events**: A full list of the system manager selection and deselection event macros. These can be programmed for all rooms in an individual location.

SELECTING A CONTROLLER



1. **Macro List**: Create global macros that can be called from buttons or events. It also provides an overview of the tags and macros programmed on a controller level.

2. **Variable List**: View a list of tags and variables programmed on a controller level.

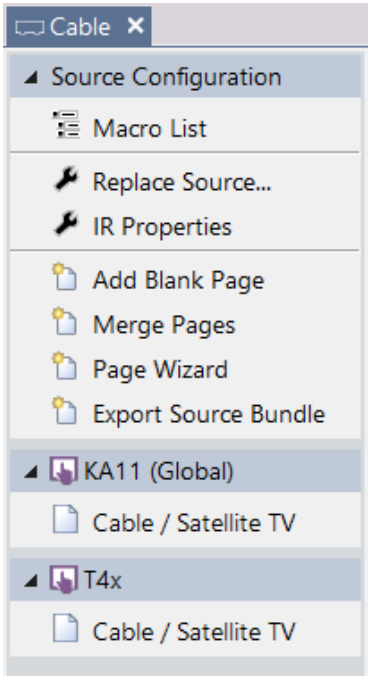
3. **Room List**: Rooms can be removed from the list and not appear when using list menus. This option is only available on controllers in the global area.

4. **Convert To**: Convert the controller to a controller with a scalable resolution.

5. **Properties**: Configure the properties of the controller.

6. **Controller Pages**: A full list of the pages for source devices available to the controller may be accessed. The source name and location are in parentheses next to the page name. Source pages can be selected, programmed, and customized. Additional maintenance options can be found by selecting the source and the applicable page.

IR SOURCE DEVICE



1. **Macro List**: Provides a list of the tags and source-level macros available for that source device. Source-level macros may be added and programmed.

2. **Replace Source**: A utility for swapping a source device for another and mapping commands properly.

3. **IR Properties**: Change the IR properties of a device, such as the port routing, default minimum repeats, and default delay after a command is sent. The device's original name can be viewed before being renamed in the workspace.

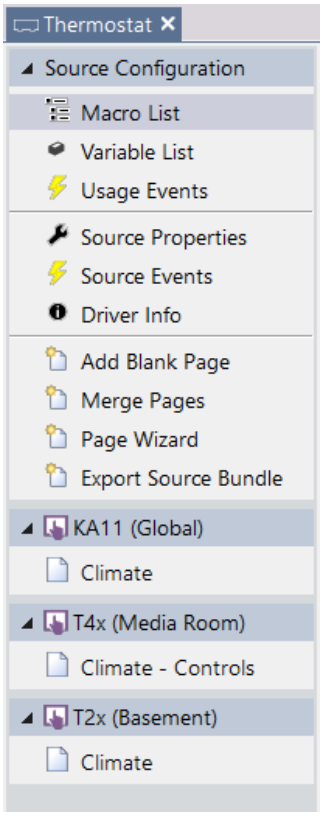
4. **Add Blank Page**: Add a blank page to this source on a selected controller.

5. **Merge Pages**: Merge a page from this project or another project for this source on a selected controller.

6. **Page Wizard**: Pick a template from a list of available template source pages and add them to this source on a specific controller.

7. **Controller & Page Name**: Displays a list of pages available for this source device by the controller. Pages may be selected and edited using the source configuration options. These options are not available when selecting the controller. Pages may be added by adding a blank page, merging pages, or using the page wizard.

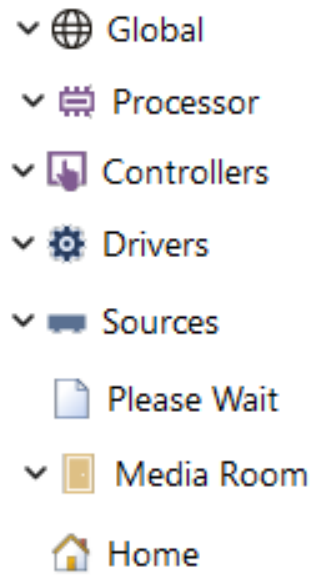
DRIVER SOURCE DEVICE



1. **Macro List**: Provides a list of the tags and source-level macros available for that source device. Source-level macros may be added and programmed.
2. **Variable List**: Provides a list of the tags and programming information for any source-level variables.
3. **Usage Events**: Macros can be programmed based on the source powering on and off.
4. **Source Properties**: This option can view and change certain driver properties.
5. **Source Events**: Driver events created for this source may be viewed.
6. **Driver Info**: Information related to the operation and feature options available for the driver. It may also contain a list of updated revisions and descriptions of changes after the initial driver release.

Identifying Items in the Workspace

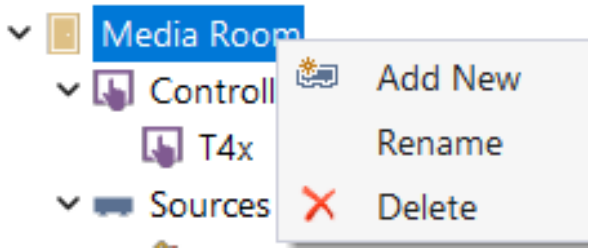
Identify items in the workspace by the symbol:



While the workspace tree categories cannot be renamed, the main processor, controllers, drivers, and source devices, including pagesets, may be renamed.

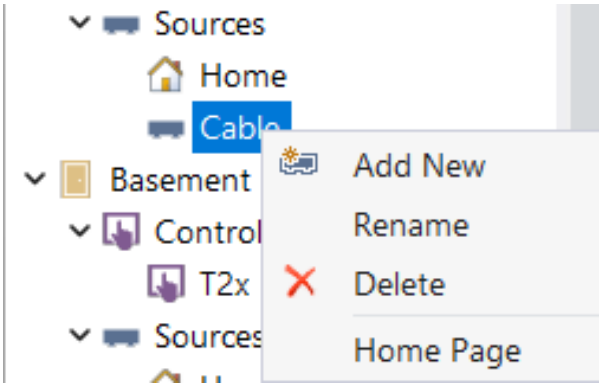
Renaming Items in the Workspace

Right-click on any item for a list of options:



You may rename or delete the item from the workspace. Selecting **Add New** will return to the **Add Workspace Item** window. Please note that deleting a workspace folder will remove any items below it.

There will be an additional option when right-clicking on source devices in the workspace.



You may designate a source device as a **Home Page** source type. Doing so will make the source device page the initial device page when synchronizing the controller. The previous home source will be converted to a pageset. Use this feature with care, as you may lose some functionality.

Pro Tip: The only way to add source devices to the workspace is by using the Add Workspace Item. Try to avoid falling into the habit of creating pages on an existing source device for another source. For example, add a lighting page to a home source. Doing so will impact the loss of timesaving features available in the software.

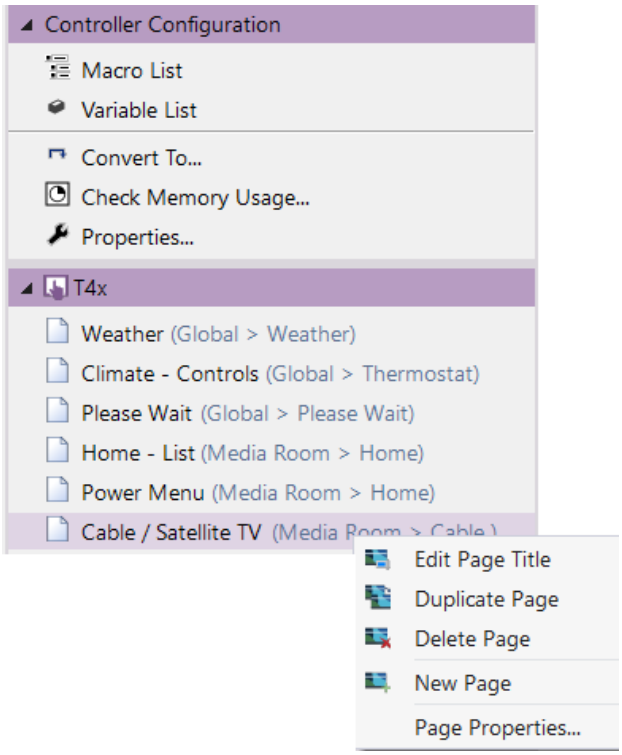
Selecting a Source Device in the Controller Configuration

There are two ways to access a source device page:

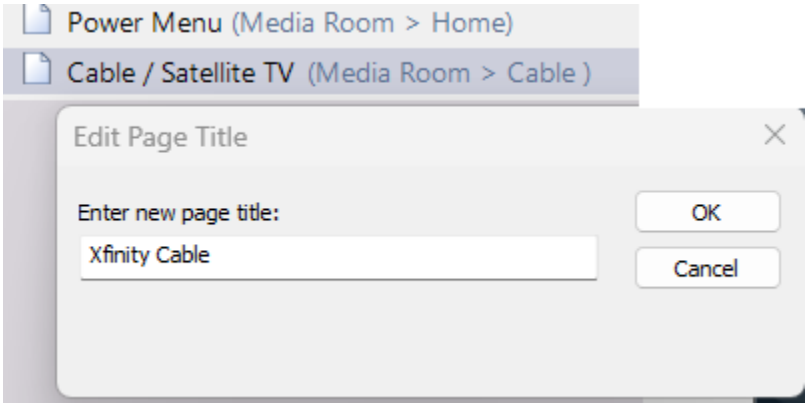
1. Select a controller in the workspace. Several options are related to the controller in the controller configuration window to the right of the workspace. In addition, a list of pages is available on that controller for the sources added to the workspace with a page selected.

2. Select the source page for the device you want to view. The user interface will show up in the centered user interface window.

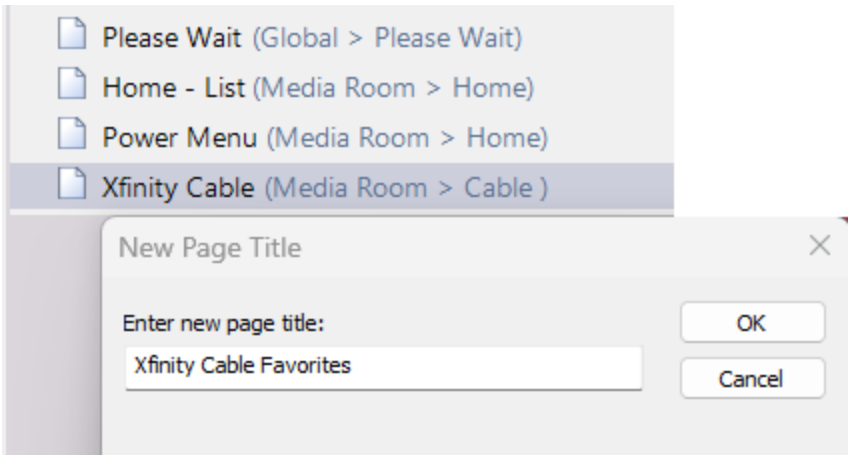
3. Right-click on the source page and view the available options:



Edit Page Title: Change the name of the underlying source page. This name is for software viewing purposes only and will not impact the user interface. Enter a new page title.

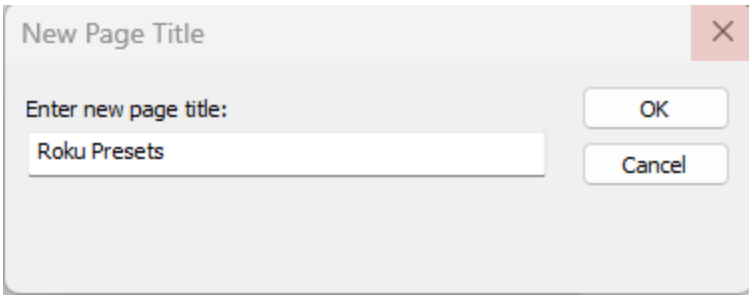


Duplicate Page: Duplicate the source device page. The software will prompt the new page name.



Delete Page: Remove the source device page. Integration Designer will not prompt you before deleting the page.

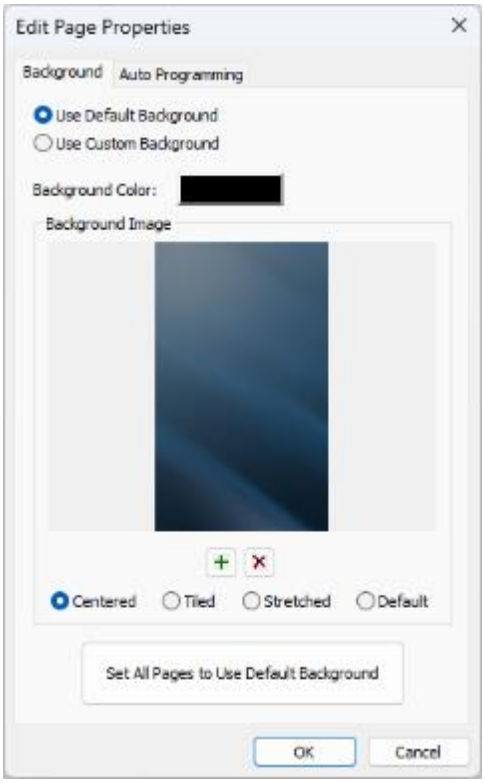
New Page: Name a new blank source page for the current source. The software will prompt a new blank page name. Select OK to rename the device.



Page Properties: View the page properties of the source device.

The page properties for a source device page enable the RTI installer to set a default background for the entire template or a custom background for the selected source page.

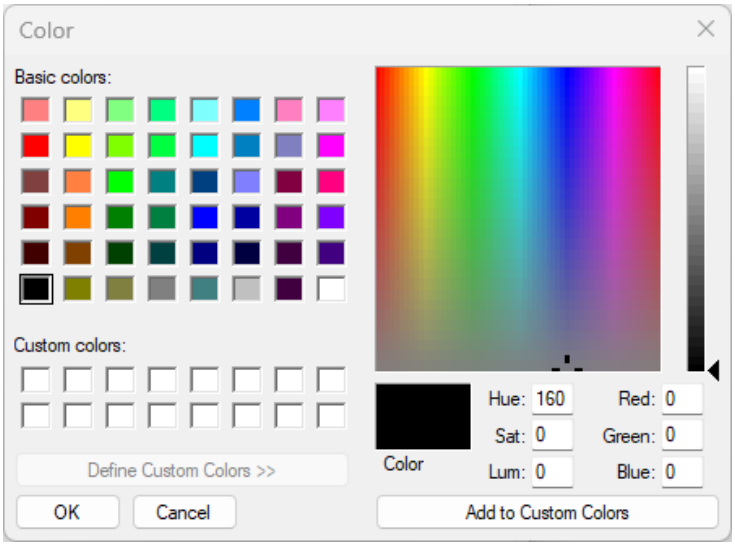
1. Select a source page from the list of pages.



2. The page properties indicate if the page uses a default or custom background.

3. Depending on the selected option, the page properties will display the current default or custom background in the Background Image section.

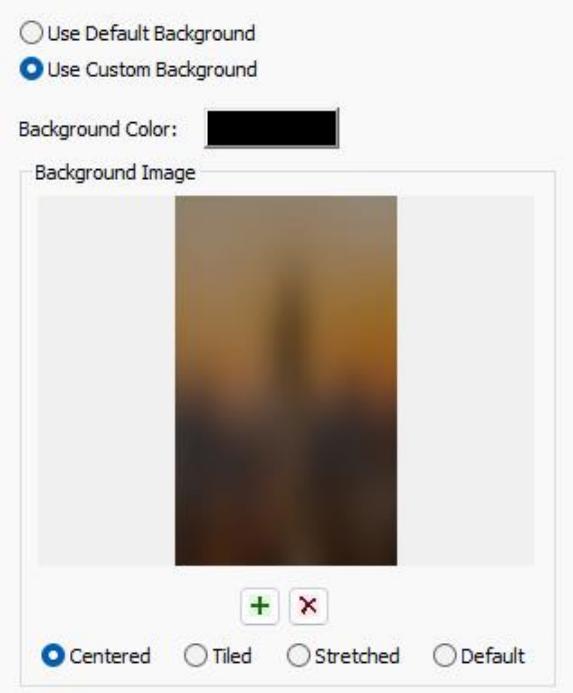
4. If using a custom background, select the background color.



The color selection window will display, allowing basic colors for backgrounds. Selecting **Define Custom Colors** will display a custom color palette for assorted color shading options. Select a color and raise or lower the level on the right side to adjust the shade. In addition, Hue, Saturation, Luminance, and red, green, and blue levels can be adjusted manually.

Once a color is selected, use the **Add to Custom Colors** button to save the color under the **Custom Colors** section.

5. Select the Use Custom Background option if using a custom background. Select **OK** to apply the change and drag a new background into the user interface. Doing so will change the existing source but not the rest of the pages in the project.



6. Selecting the **+** sign will allow navigation to a local PC directory where an image can be replaced with a personal background image. Selecting the **X** will remove the background completely from the page.

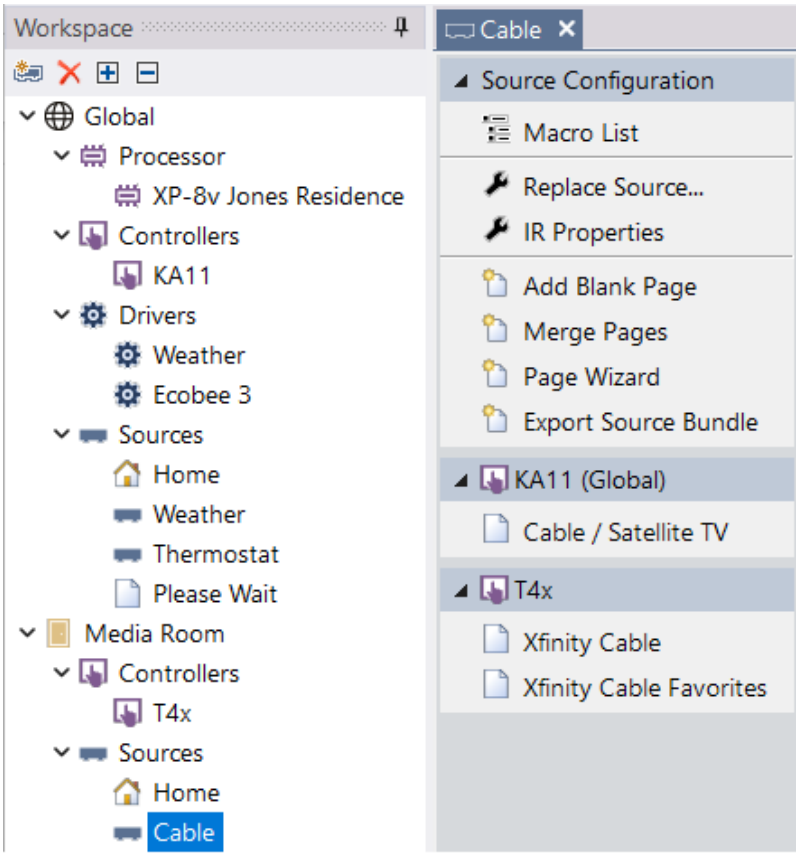
Depending on the image, it can be **centered**, **Tiled**, **Stretched**, or use the **default** type.

Selecting a Source Device Page in the Source Configuration

A source page can also be selected by selecting the source device in the workspace. When selected, the source will show the pages that exist for that source on the controllers. This method is better for source maintenance since options will be geared toward source maintenance.

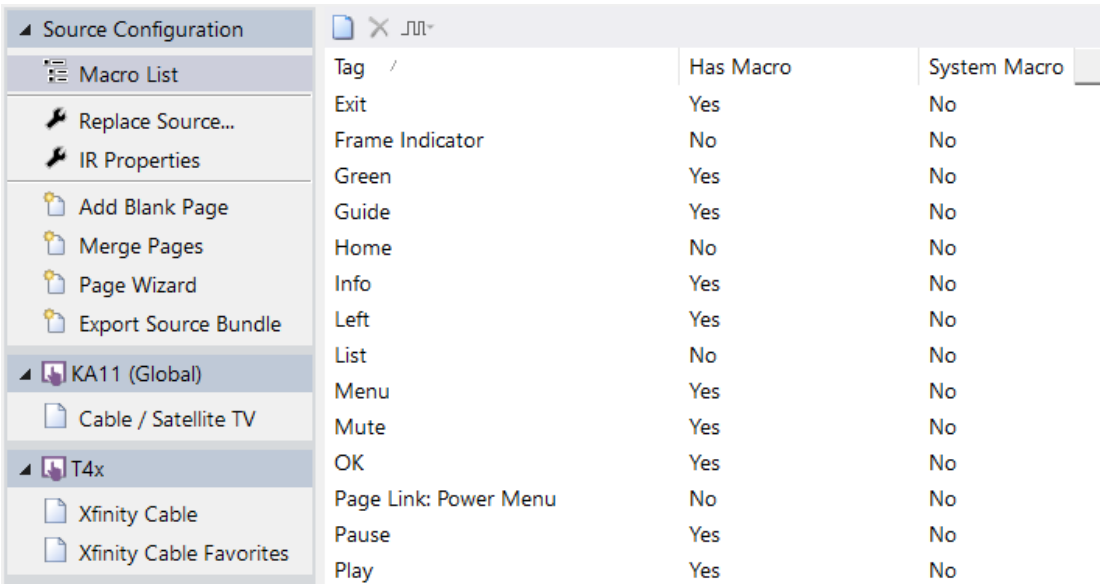
1. Select an IR source device in the workspace.

A list of pages will display for any selected controllers when adding the source to the workspace. This way, you can see all the available pages for a source and the controllers that house the source pages.



2. Several options in the **Source Configuration** window will be available to perform source maintenance.

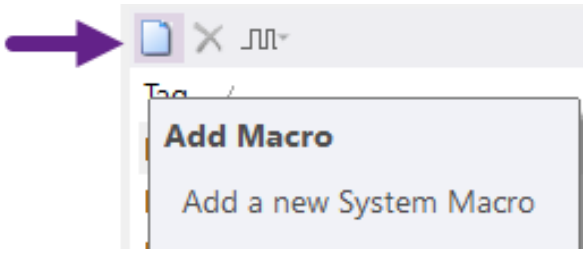
Macro List: The macro list will display all the tags created on the source device. RTI installers can create tags and macros available on the source device interface. These macros can also be called from other macros, even in other rooms, or used for events.



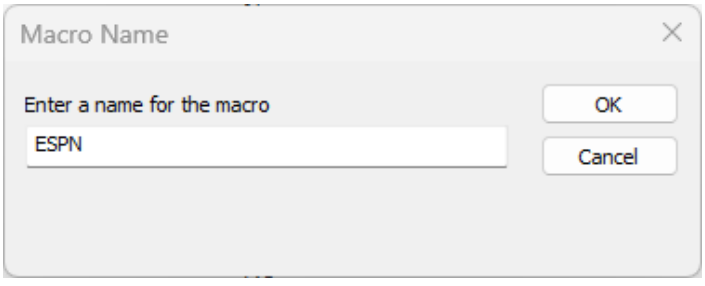
Tag	Has Macro	System Macro
Exit	Yes	No
Frame Indicator	No	No
Green	Yes	No
Guide	Yes	No
Home	No	No
Info	Yes	No
Left	Yes	No
List	No	No
Menu	Yes	No
Mute	Yes	No
OK	Yes	No
Page Link: Power Menu	No	No
Pause	Yes	No
Play	Yes	No

Each tag has a column indicating if a macro has been programmed or a system macro has been created and programmed manually.

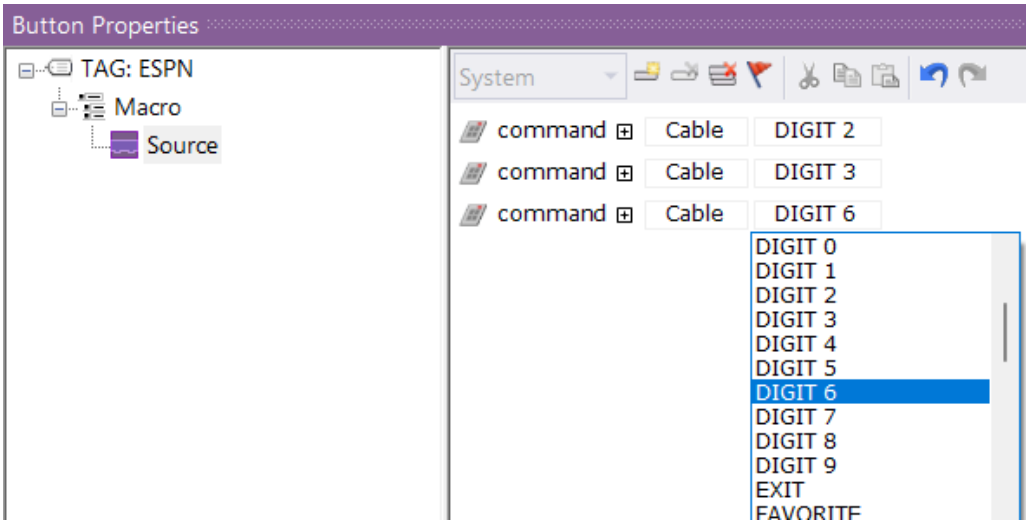
1. Select the **Add Macro** icon in the upper left to create a tag and macro.



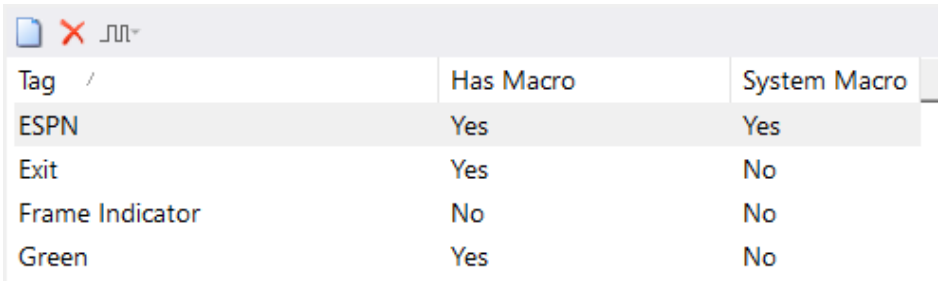
2. Create a tag name that best describes the macro programmed.



3. Select the command function and add a command from the Macro Steps tab in the tabbed menu. Repeat any additional commands.



4. The tag will appear at the top of the list and be identified as a system macro. Placing this tag anywhere on the source interface for any controller will auto-populate the macro programmed.



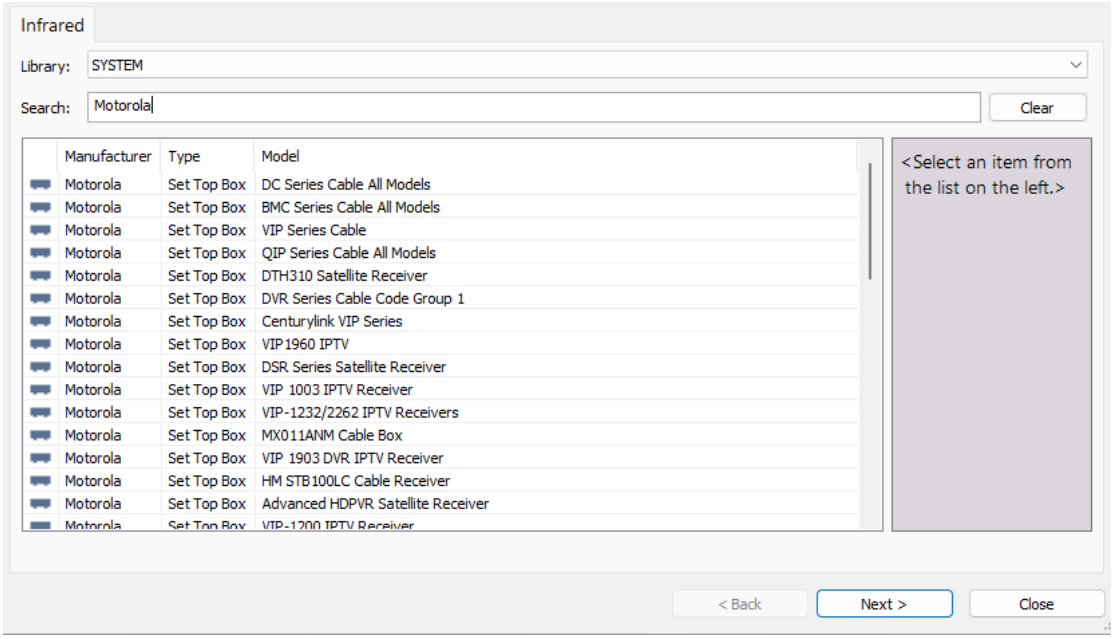
Tag	Has Macro	System Macro
ESPN	Yes	Yes
Exit	Yes	No
Frame Indicator	No	No
Green	Yes	No

5. Tags can be **deleted** by highlighting the tag and selecting the red delete command in the toolbar.

Replace Source: selecting this option will prompt a new IR device to swap with the existing IR source device. Commands may be mapped to existing commands, and any fallout can be reconciled.

1. Select the **Replace Source** command in the source configuration window. The IR library

2. Type in the search criteria for the new model that replaces the existing source.

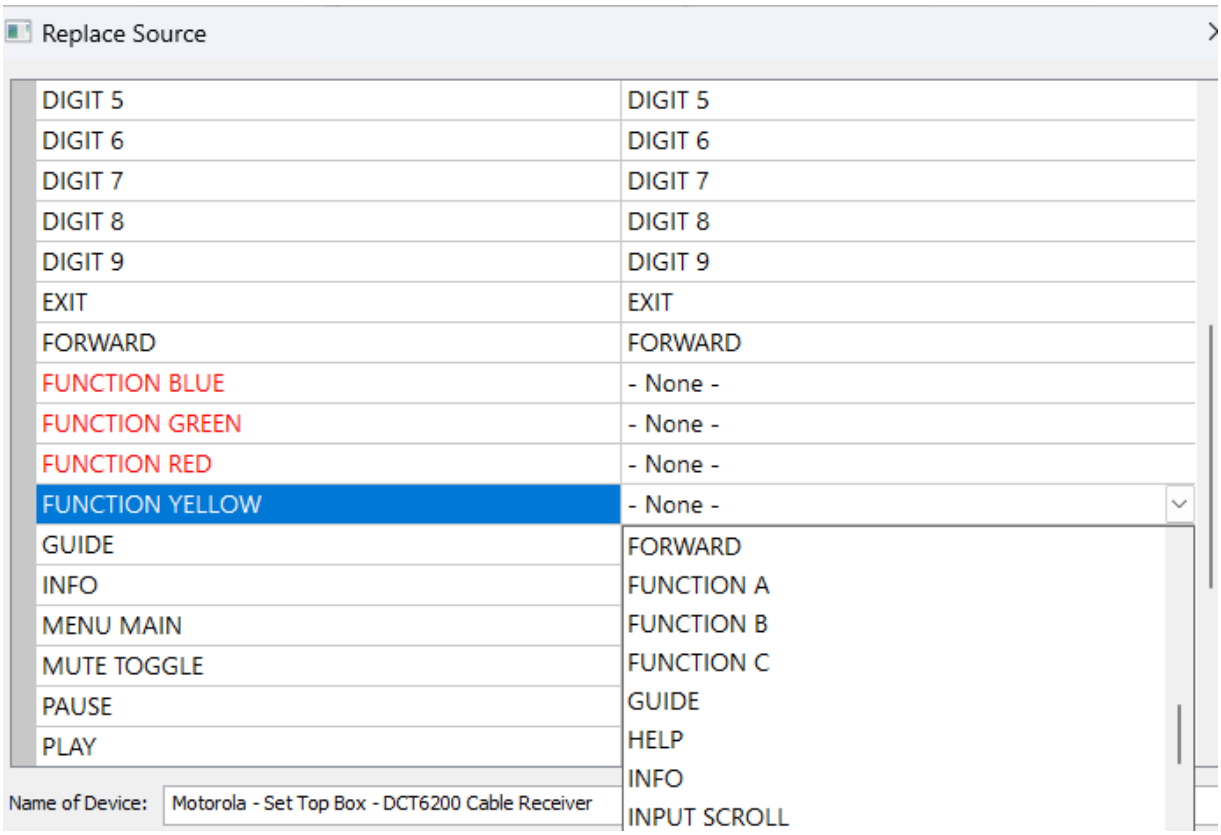


3. Select the new model and select the **Next** button.

4. Some commands may be available in the existing interface but cannot be mapped to commands in the new interface. Scroll through the list and identify commands in red that are set to map to "**none**."

   Function Blue, Green, Red, and Yellow could not identify a match in the new code set in this example.

5. Select a command from the dropdown menu and map accordingly. An option is available to map the commands to **none** or to **keep the original** command in place.

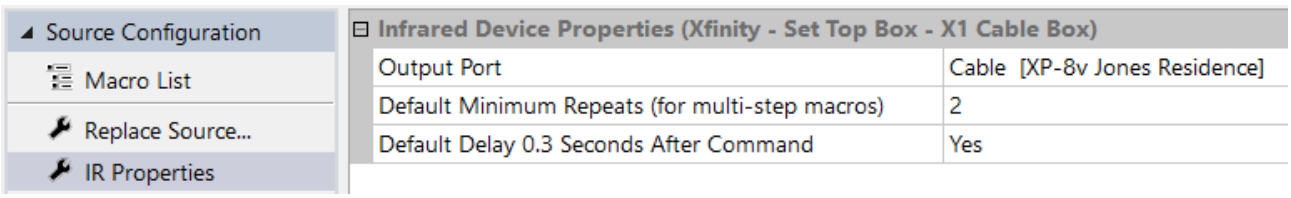


6. Give the device a new naming convention, then select **Replace Source**.

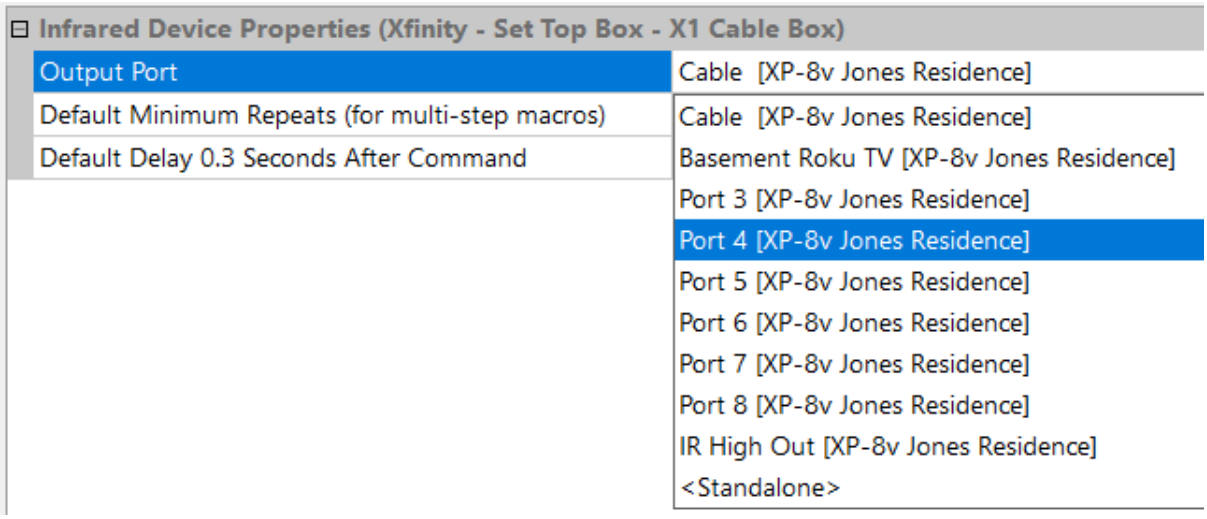
Pro Tip: If you decide to use the existing name, the name in the workspace may change to the same name with a "#2" after the device name. Rename the device in the workspace should this occur.

IR Properties: The IR properties option can change IR routing and other characteristics. If you decide to re-route the existing IR port of the device, you must use the IR properties option in the source configuration.

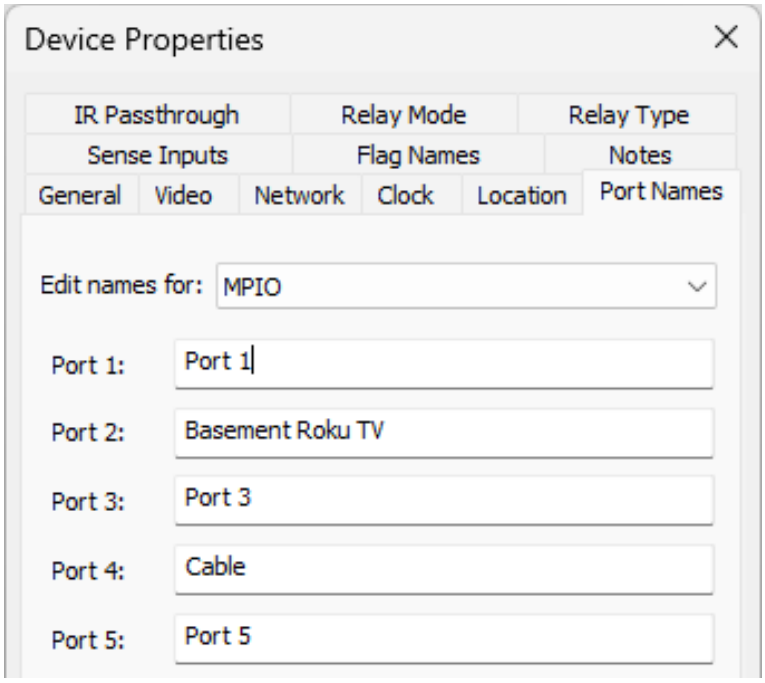
1. Select the **IR Properties** option in the source configuration window.



2. The new **Output Port** can be selected using the dropdown menu and routed to any main or expansion processors in the project file. It may also be routed to **Standalone,** and the controller will send IR directly.



3. Since rerouting a port does not rename it, it must be renamed in the port settings of the processor. Select the **main processor** from the **workspace**, then select the properties option in the processor configuration window. Rename the ports to correspond to the IR routing changes in the previous step.

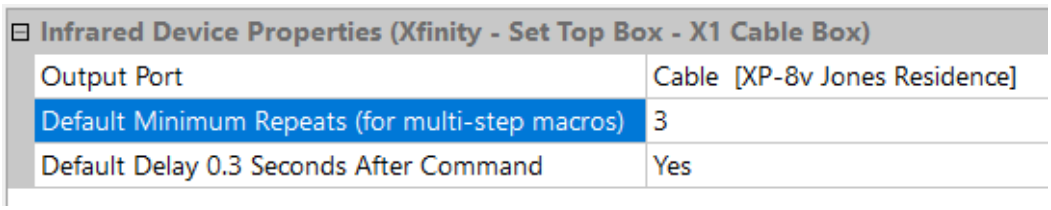


Pro Tip: IR ports are automatically routed and labeled with the source device name when added to the workspace. For relays, sense inputs, and RS-232 commands, it is best practice to manually label those ports in the port settings tab in the processor properties.

Default Minimum Repeats (for multi-step macros)

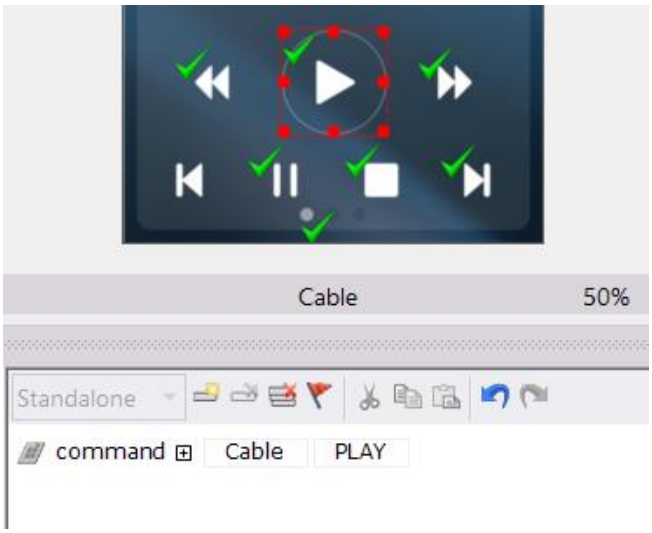
The **minimum repeats** setting guarantees that a Pulse and Hold IR command will be transmitted the specified number of times, even if the button that triggers the command is released quickly. By default, the minimum number of repeats for single macro commands is **one** repeat. This setting can be adjusted for individual buttons if the device does not respond properly on a button level. For multiple macro commands, the default minimum repeat is **two** and can be changed here.

1. Set the **default minimum repeats** to the desired setting.

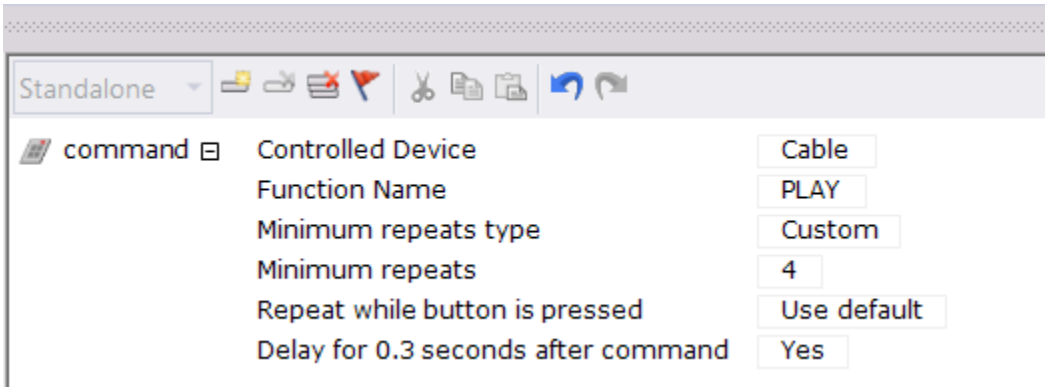


2. If the default minimum repeat needs to be changed for a specific command on a button, navigate to the button in the user interface and select the button.

   Select the **+** next to the command in the macro.



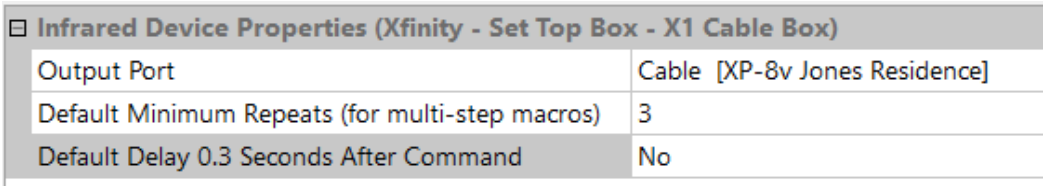
3. Please change the **default minimum repeats type** to **custom**, then set the desired value for **minimum repeats**.



Default Delay 0.3 Seconds After Command

It is customary for a small delay between IR commands to improve transmission. Sometimes, delays between commands must be increased if commands are lost.

1. Select the argument next to the Delay for 0.3 seconds. You can change this from **Yes** to **No**.



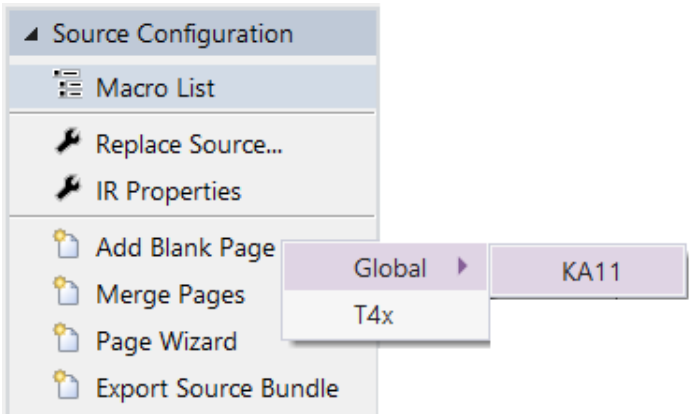
Once set to **No**, you must insert delays after the macro command in the macro window if required.

Add Blank Page

This option will add a blank page to the selected source on a selected controller. In this scenario, the page will be completely blank and require at least one layer.

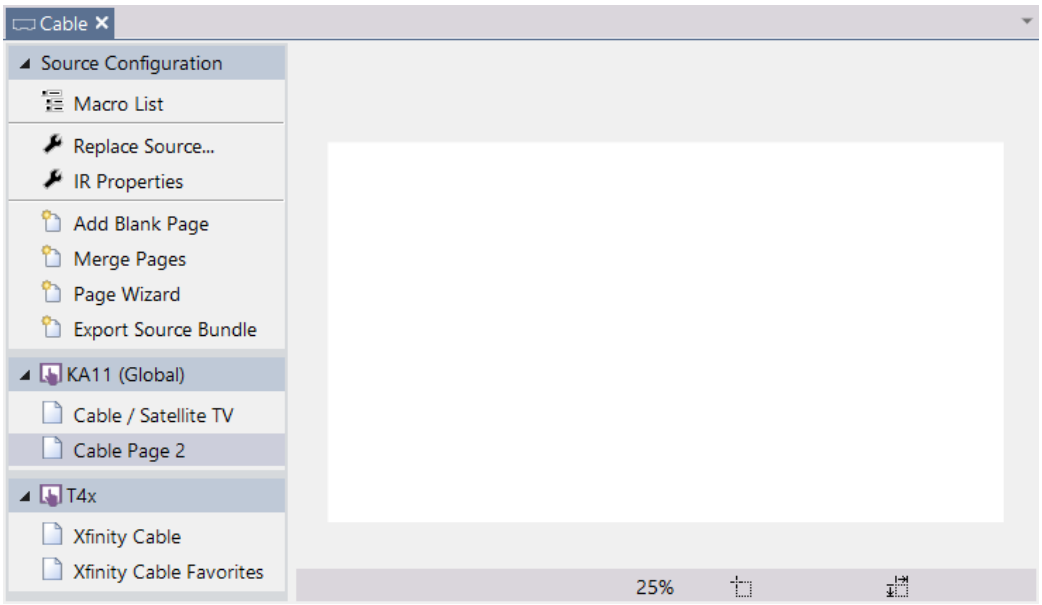
1. Select Add Blank Page in the source configuration properties.

2. Select the room and controller where the blank page should be added.



3. Enter a name for the new blank page.

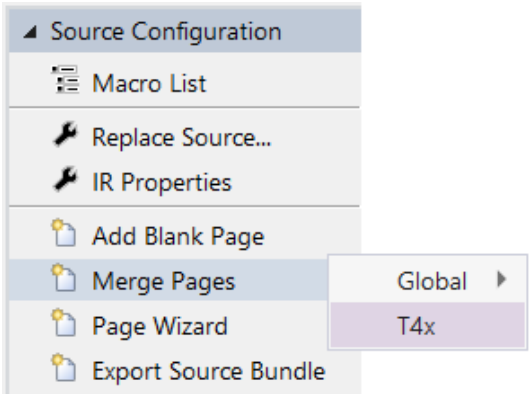
4. Add elements to the page that include at least a single baseline layer.



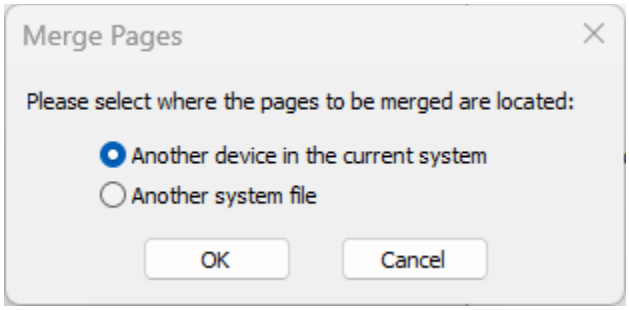
Merge Pages

Pages can be added to the source by merging them from an existing or another project.

1. Select the **Merge Pages** option and select the room and controller to import the merged page.

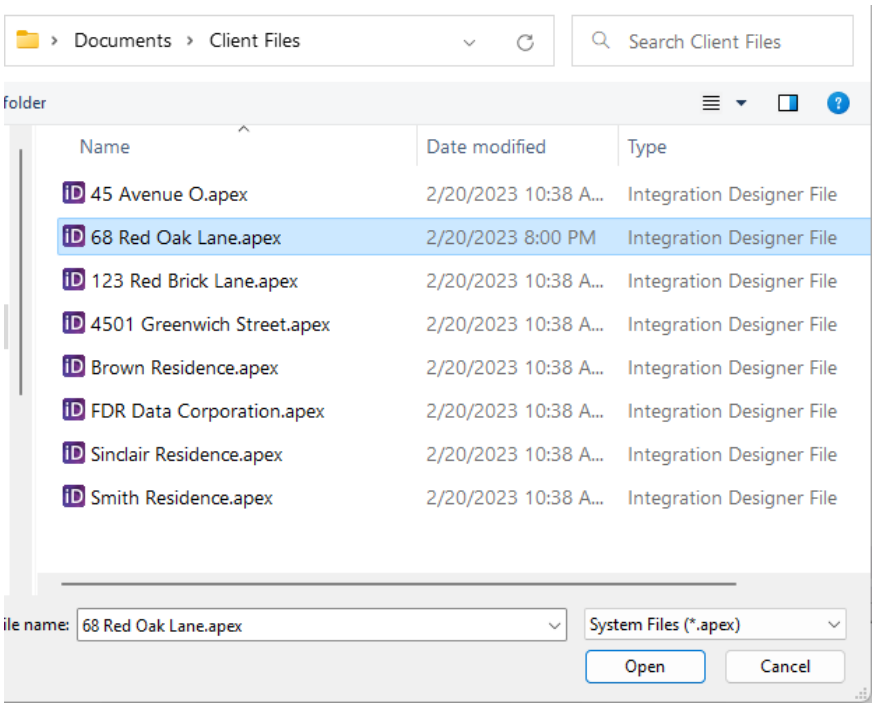


2. Select where the pages are to be merged or located. Pages can be merged from the existing or another system file.

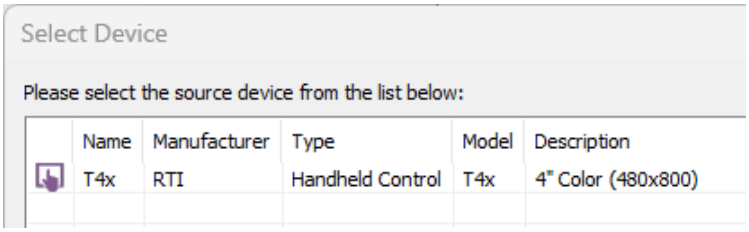


3. If selecting another system file, navigate to the directory where the file is stored and select it.

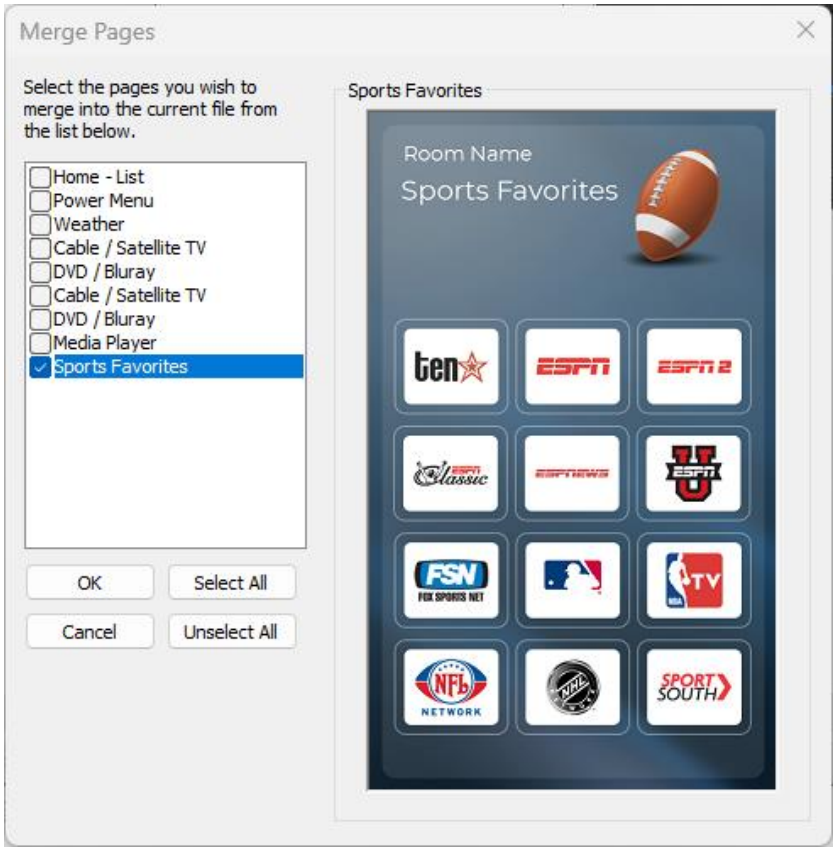
   Note: To find a compatible page, it must exist in the existing or another project for the controller selected. If nothing shows up when a project file is selected, it could mean there is no controller to match the selection.



4. Select the controller where the page is located.



5. Select the pages to merge into the project. Multiple pages may be merged. Selecting a page will preview the user interface on the right side.



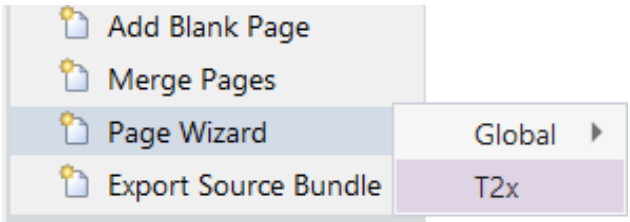
6. Once the page is merged into the project, it can be renamed and customized.

Page Wizard

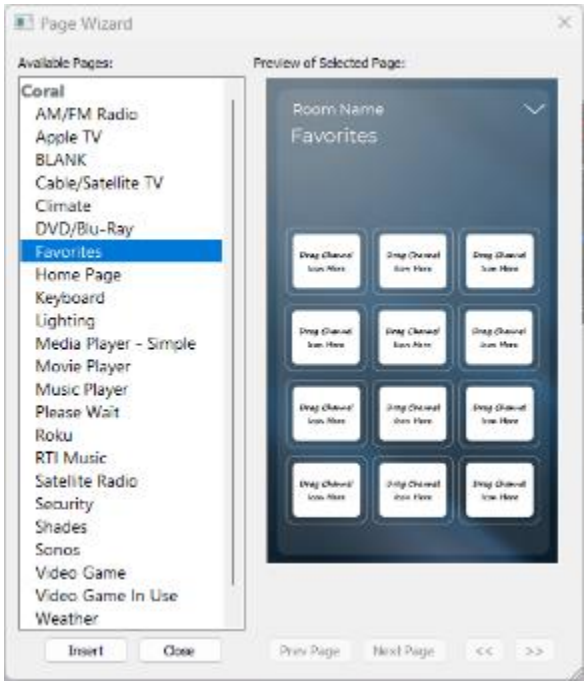
Importing template pages is possible with Page Wizard. The page wizard can easily add any page to a project by featuring all the generic template pages for source devices.

1. Select a source, then the **Page Wizard** in the source configuration window.

2. Select the room and controller where the page will be placed.



3. The pages that comprise RTI templates are provided in the page list. Next to the page wizard window is a preview of each selected template.



Some templates may include multiple pages or frames. To preview pages, select the **Next Page** or **Previous Page** buttons. Scroll through available frames using the **>>** and **>>** buttons.

4. Select the template page and select **Insert** to add the page to the source device.

Exporting and Importing Bundles

RTI offers full customization and programming options, allowing RTI installers to control the user interface and experience. Installers may build custom templates or alter existing templates to match the client's needs and lifestyle.

This convenience can cost time and effort. RTI Installers can use the bundles' feature to export source devices while preserving programming, graphics, and customization. These can be exported and imported to future projects, avoiding repetitious programming while making RTI installers more productive and profitable.

The following criteria are preserved in bundles:

- Macro Programming (source device)
- Feedback Programming (source device)
- Tags, both standard and custom
- All graphics
- Layers
- Full device library

Exporting a Bundle

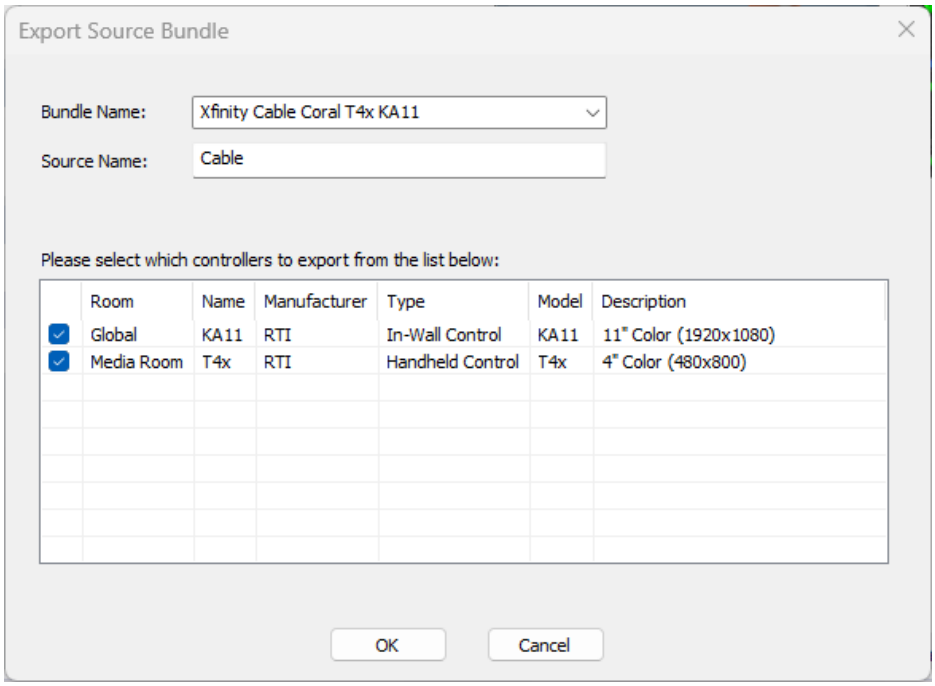
1. Select a source from the workspace.

2. Select Export Bundle from the source configuration window.

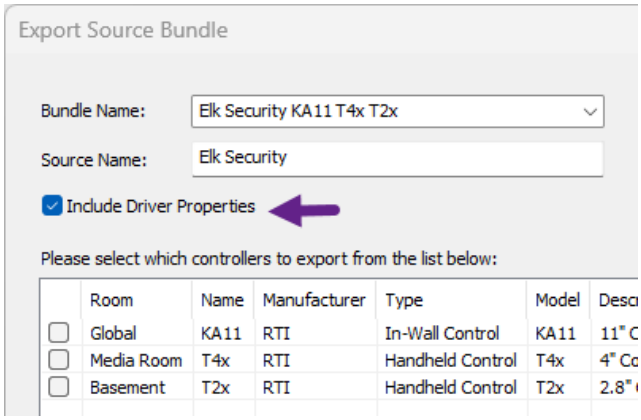
3. Please enter the **Bundle Name** in the export bundle window, as it will appear when importing the device. Use descriptive naming conventions for best practice.

4. For the **Source Name**, the name selected will be the name of the source device when it is imported into a project.

5. Select the controllers to export the user interface.



An extra option to include driver properties in the export will be included when exporting driver bundles.



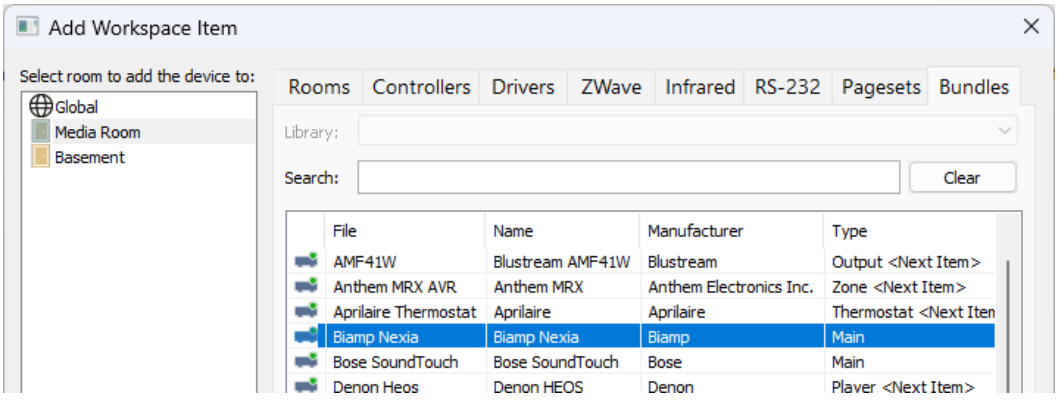
Bundles will be saved in the directory defined in the **Options/Set Directories/Templates** area.

Importing Bundles

1. Select the **Add Workspace Item** button in the workspace toolbar.

2. Select the **Bundles** tab.

3. Select the source device bundle you wish to import to the project. Adding a bundle is an alternative to adding a source device. Adding a separate driver or IR library is unnecessary when using a bundle is desired.



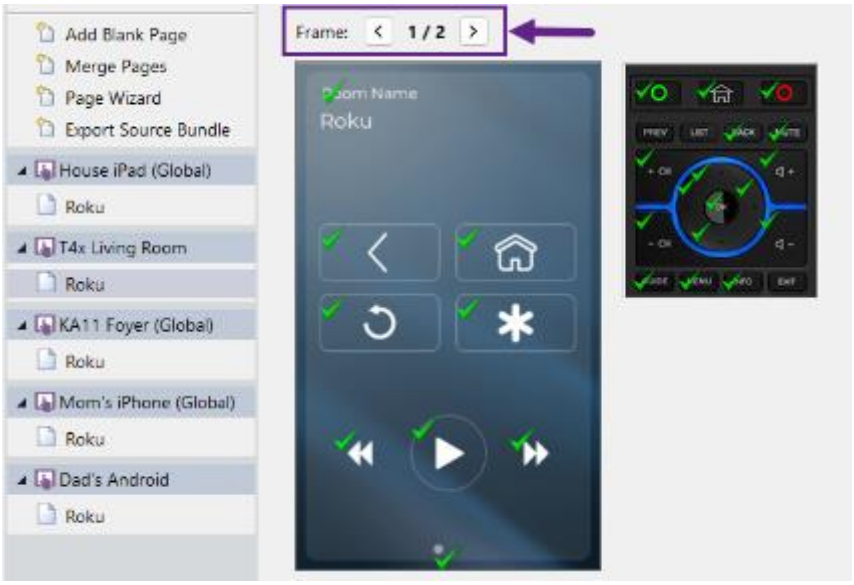
4. Add the device to the workspace, following the same procedure as adding any other source device.

Pro Tip: Saving your source templates to the bundle's area will make for an easier time building your next project file. If there are controllers in the project file that are not available in the bundle, they will not receive a user interface. These need to be added manually in the source configuration options.

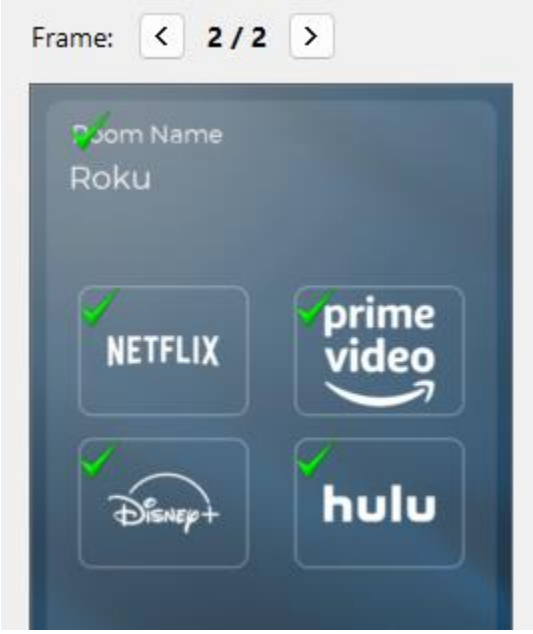
Using Pages and Frames

Additional source **pages** may be added in the source configuration window when creating activity user interfaces to provide more comprehensive control options. Select RTI controllers are compatible with a **frames** feature. Each page has up to ten frames accessed by scrolling left and right with the arrows on the user interface window above the controller view.

In the below example, the selected page source on the T4x controller has two frames. A **frame indicator** graphic object at the bottom of the screen will indicate the current frame, allowing the end-user to navigate.



1. Select the **right arrow** to scroll to the next frame available.



2. If additional frames are required, advance the right arrow to frame three.

Screen elements may be carried over from frame to frame if desired. In this example, the device, room information, and transport buttons will be displayed in each frame.

Buttons can be removed from each frame view by right-clicking the button or text box and selecting **Show Button on All Frames**.

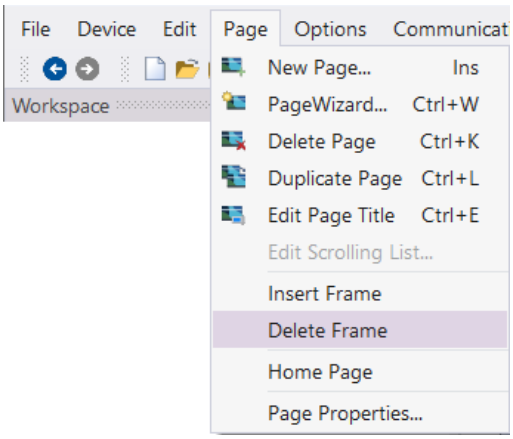
3. Removing the existing frame indicator graphic will be necessary when adding additional frames and swapping it with a frame indicator that matches the number of frames.

   In the tabbed menu**, frame indicators** are found in the bitmap library by searching for "**frame indicators**."

4. Some controllers have gesture navigation features, enabling the end-user to jump forward and back to each frame. If the controller does not support gestures, give the end user a forward and back button to navigate to each frame.

   Pro Tip: It is important to design user interfaces using pages and frames when applicable. Channel Favorites, presets, scenes and other repetitive functions are good examples of when to use frames. To separate functionality, use pages when necessary.

Deleting Frames and Inserting Frames



1. Select the source page, then navigate to the frame you want to remove.

2. Select the **Page** option in the main toolbar, then select **Delete Frame**. Alternatively, you may use the **Insert Frame** option to add a frame in the selected position.

Using Layers in Integration Designer

Layers are a powerful feature that allows the RTI installer to separate areas of the interface, manage popups and status messages, and share common graphics throughout the user interface, maintaining it in an individual location.

Here are some examples of layers already built into the Coral template:

Toolbars	Buttons to Return to the Current Source
Source, Room, and Power Menus	Volume Buttons
Keyboards	Dashboard Views
Text Identifying Room Names	Framework Graphics

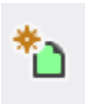
Important Layer Information:

- At least one layer is required for every page.
- Layers can be made visible under certain conditions based on feedback.
- Changing the conditions when a shared layer is visible without affecting the layer in other interface areas is possible.
- If anything is changed on a shared layer, it will change anywhere that layer appears.
- Deleting a layer on a page without deleting it on other pages is possible.
- The overuse of layers can impact screen performance based on available memory.

Understanding the Layer Window

The layer window provides useful information to the RTI installer. Information such as the layer name, source location, and if the layer is shared is noticed by glancing at the window. The layer's name is informative since layers can be originated and shared to different locations in the user interface and pointed to sources in any room for auto-programming.

Insert a blank layer.



Insert a shared layer.

Delete a layer from the page.

View the layer properties.

Move the selected layer up and down in priority order.

The layer is visible in the layer window.

The layer is not visible in the layer window.

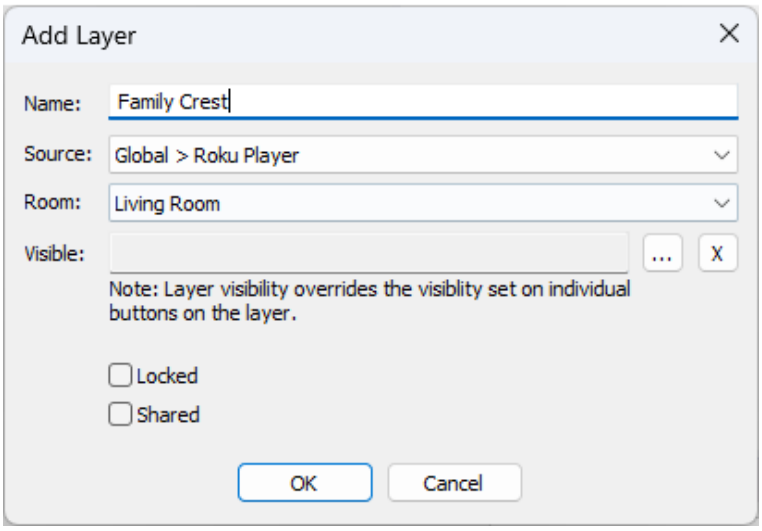
The layer is locked in the layer window.

The layer is unlocked in the layer window.

A shared layer that may be shared anywhere in the UI

Creating a Layer

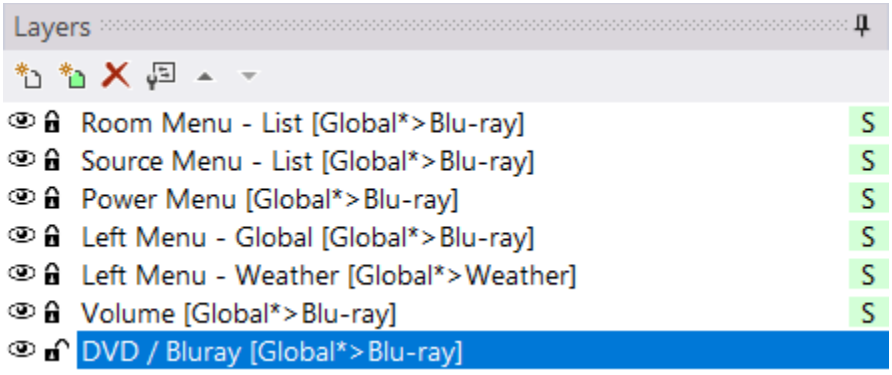
1. Select **the insert a new blank layer icon** in the upper left toolbar in the layer window, and the **layer properties** window will display.



2. In the **name** field, identify the layer by assigning it a name.

3. The **Source** field will allow you to derive macro and variable programming on the layer based on tags. Since this layer will feature a family crest logo and does not require programming, it is fine to leave this field as is. A further explanation will be provided in the **Advanced Layer Programming** section.

   A layer's source and room can be identified in the name of the layer in the layer window. The layer room and source in the layer name and properties indicate where the programming resides.



4. The **Room** field is connected to the source field and will allow you to point the layer to the room where the source is. A further explanation will be provided in the **Advanced Layer Programming** section.

5. The **Visible** field will allow you to select a Boolean state variable that, when true, will display the layer on the user interface. A further explanation will be provided in the **Advanced Layer Programming** section.

6. Select the **Locked** field to lock the layer and prevent accidental edits.

7. Select the **Shared** field to make the layer eligible for sharing in other or all areas of the user interface.

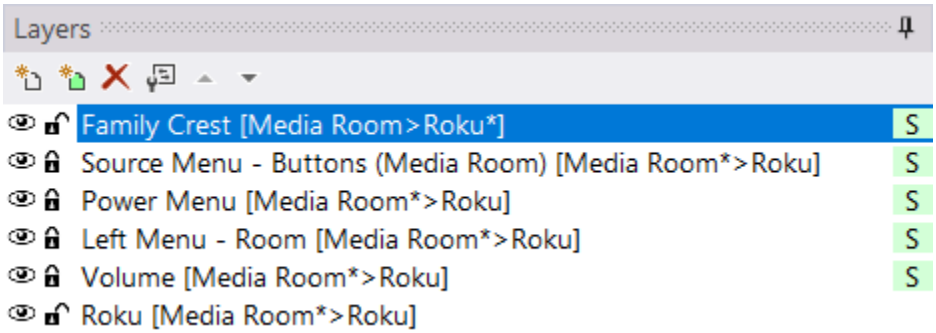
    - Selecting the shared layer option will not copy it to all user interface areas. It is eligible to be shared with other user interface areas.

    - Once a layer is marked shared, it is not reversible.

8. Select **OK** to apply the layer properties.

Next, place a logo on the new layer.

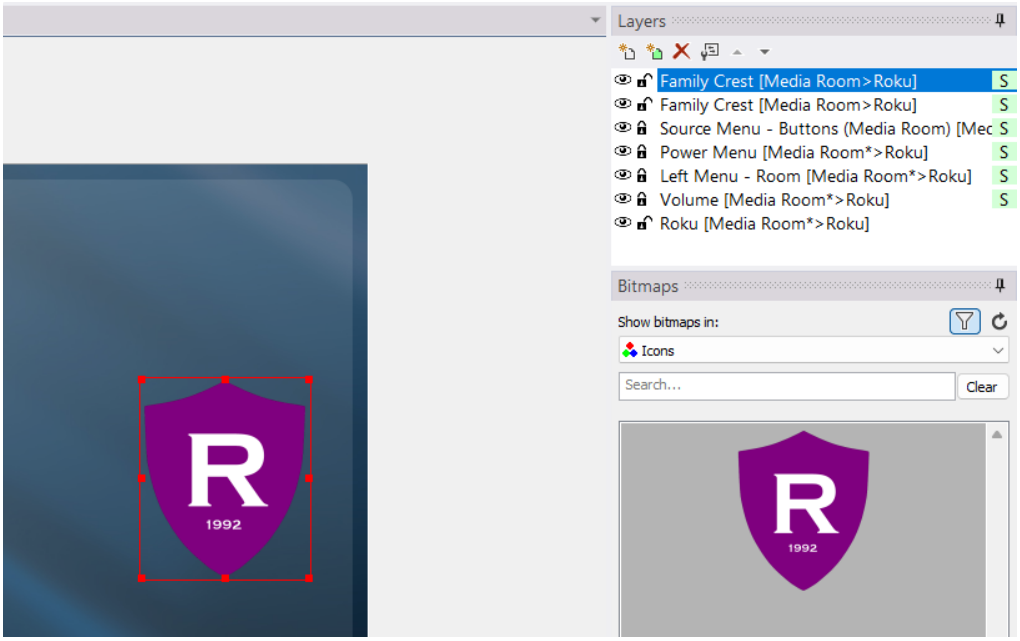
1. Ensure the new layer is selected, visible, and unlocked in the layer window.



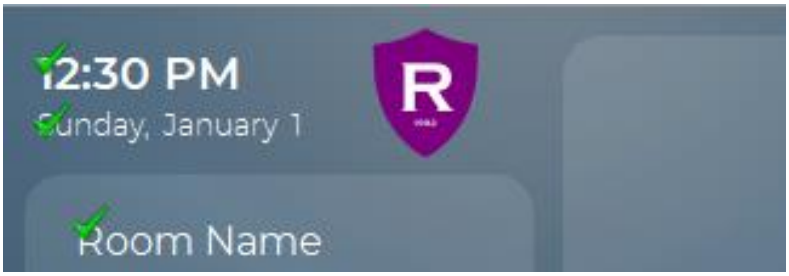
2. While the new layer is highlighted and selected, drag a graphic from the bitmap library to the user interface.

    If the other areas of the screen are distracting or in the way, you can lock or hide them to prevent accidental changes. Toggle the **hide/unhide** button or the

visible/hide button, but ensure the correct layer is selected when sending graphics to the interface.



3. Resize the graphic by selecting the frame handles and dragging the corner. Place the logo in an area of the screen available on the pages you wish to place it.



4. You may **lock** the new layer and unhide the layers that comprise the user interface.

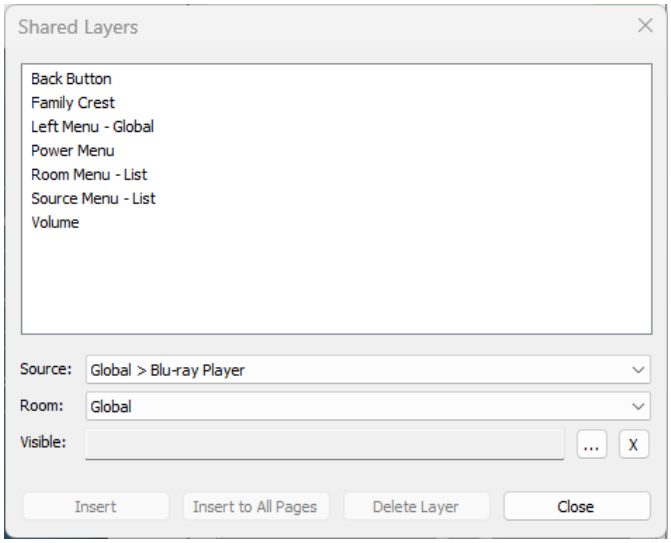
Inserting a Shared Layer

Once a layer is set to a shared layer, it can be copied to any other areas of the user interface.

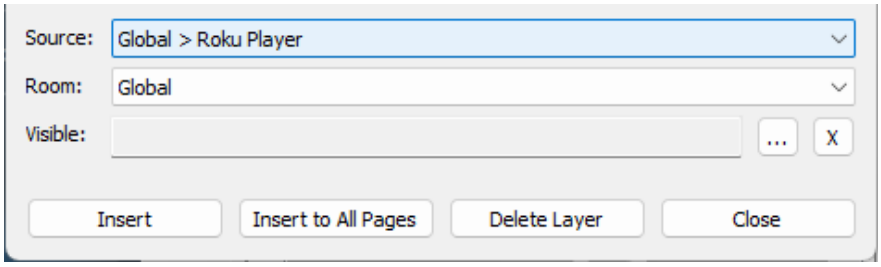
1. Select the next available page or where you want to insert a shared layer.

2. Select the green **Insert a Shared Layer** button in the layer window toolbar.

3. A list of **shared layers** will be displayed in the shared layer window.



4. Select the **Family Crest shared layer** from the list.



In the **source** field, it is possible to derive macros and feedback from any source device in the project.

For room, use this feature to set the **room** field where the source is located.

In the **visible** field, you can select the ellipsis menu and select a variable or flag. If the value is true, the layer will display. Since this layer will always be visible, skip this step.

You have the following options available:

- Insert the shared layer in the selected page.

- Insert the shared layer into all other pages.
- Delete the layer (this will remove the layer everywhere.)

5. Select **Insert to All Pages** so that the layer appears on all pages in the interface.

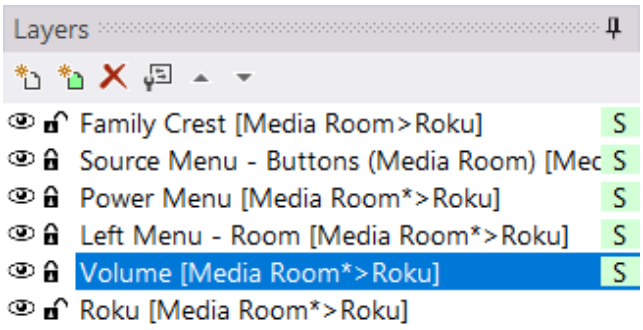
6. View the various pages in the controller configuration window and ensure the layer placement is correct.

   Since the shared layer is centralized, any cosmetic changes applied to the layer's contents will be available anywhere the layer is located.

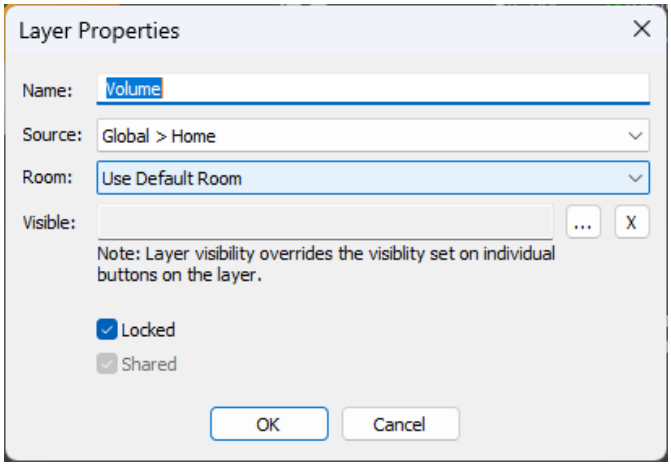
Changing the Layer Properties

Layer Properties allow the RTI installer to revise the characteristics of a layer. Select a layer in the layer window, then select the **Layer Properties** button in the toolbar.

1. Select a layer and **Layer Properties** in the layer window toolbar.



2. Make changes to the layer properties as desired.



Changing the **Name** field will reflect on the layer everywhere it is shared.

Changing the **Source** and **Room** fields will reflect on the shared layer on the current page. It will not impact the other shared layers.

Changing or adding a **visible** state will impact the selected layer, not any other shared layer.

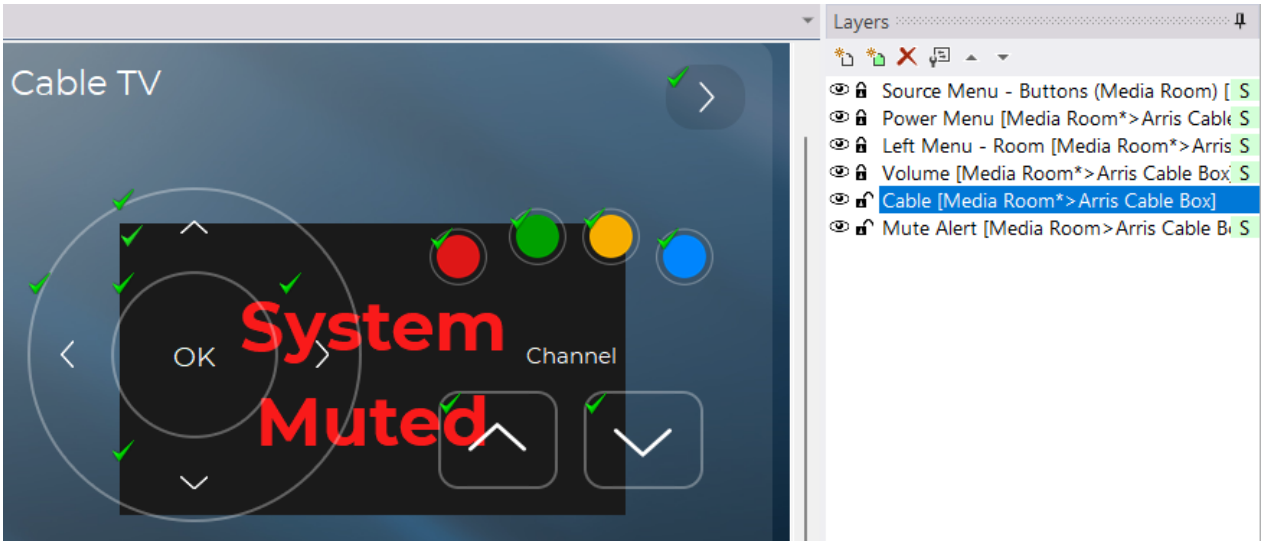
Locking the layer will only impact the selected layer, not any other shared layer.

Although reversing a **shared** layer is impossible, marking a layer shared makes it eligible for sharing anywhere in the project file for the current controller.

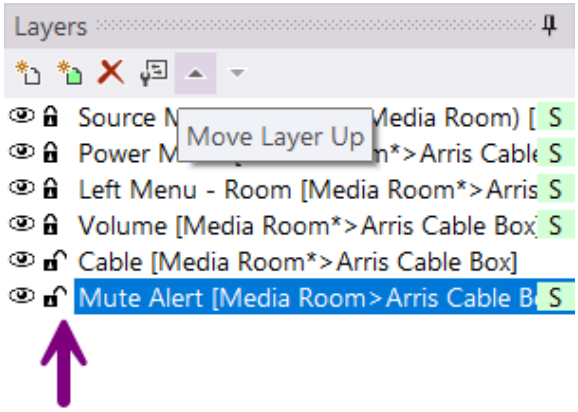
Changing the Layer Priority

The order determines the priority of the layer it appears in the layer window. Any other layer cannot overlap layers on the top of the list. Layers on the bottom of the list have the lowest priority and may be overlapped by all other layers.

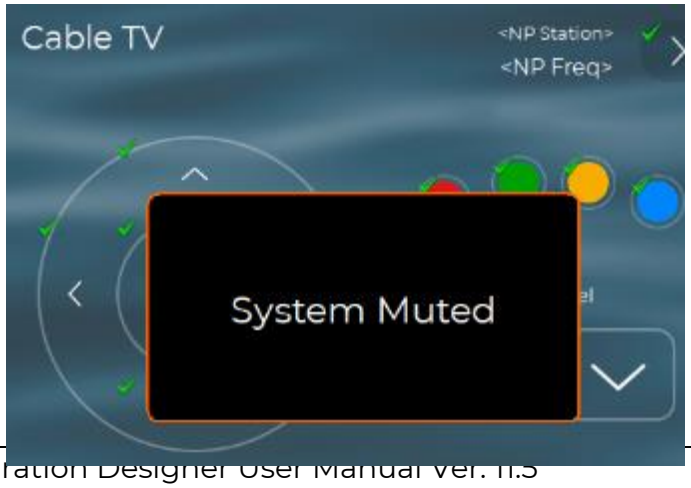
In this example, the mute status shared layer is the lowest priority and is overlapped by other layers on the interface, making it difficult to see.



1. Select the layer you wish to reprioritize in the layer window.



2. Select the **Move Layer Up** button in the layer toolbar to move the layer up in the list order.



No other layer contents can overlap as the layer moves up in the list.

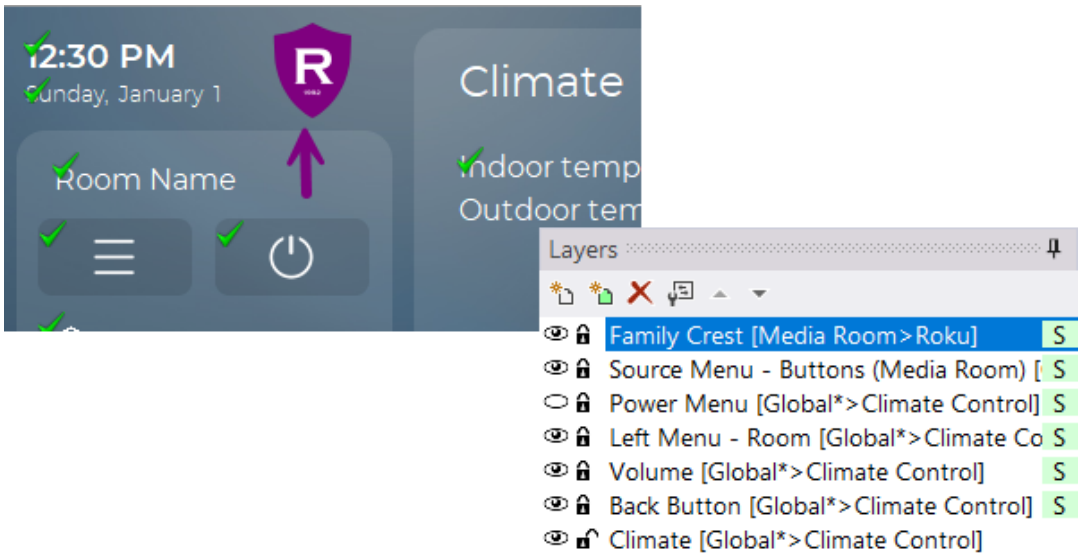
3. Change the priority of the shared layer on other pages in the user interface. Changing the priority of the layer does not change the priority of that layer anywhere else it is shared.

   Pro Tip: When working with shared, visible state layers, hide them on the screen so they do not distract or get in the way of programming the main interface.

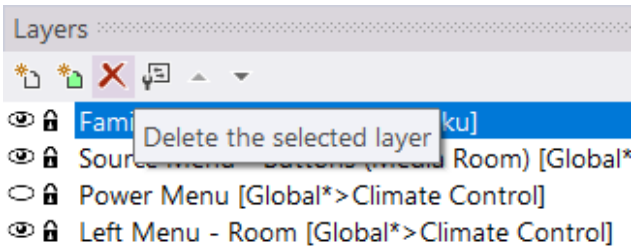
Deleting a Layer

There are two options for **deleting** a layer: deleting the layer on the **current** page only or deleting the layer from **all** the pages it exists.

In this example, the shared family crest logo appears on many other pages in the interface but is not wanted on the climate control interface.



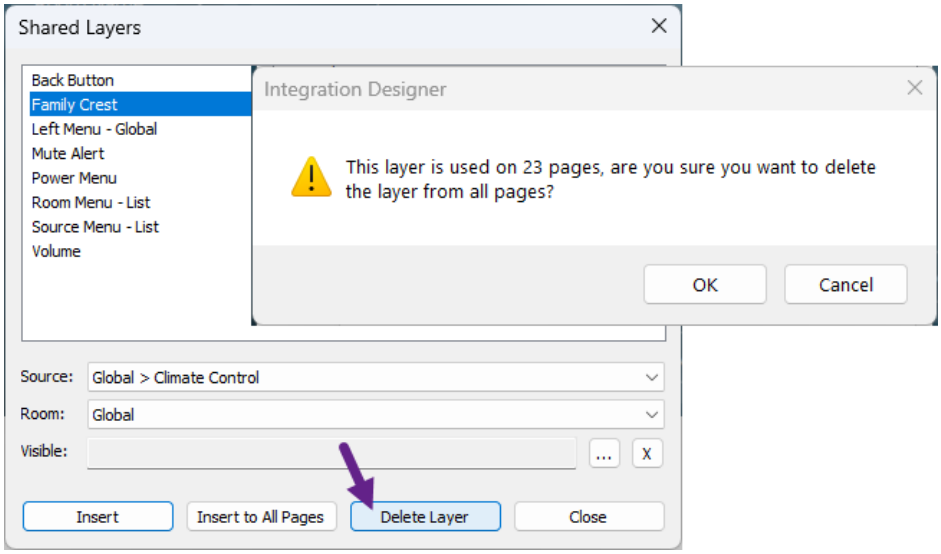
1. Select the red delete button in the layer toolbar to delete the layer from the current page only.



2. Repeat the same process on other pages where the layer is unnecessary.

A layer may be **permanently** deleted and removed **everywhere** it exists on the controller,

1. Select the **Insert a Shared Layer** button in the layer toolbar.



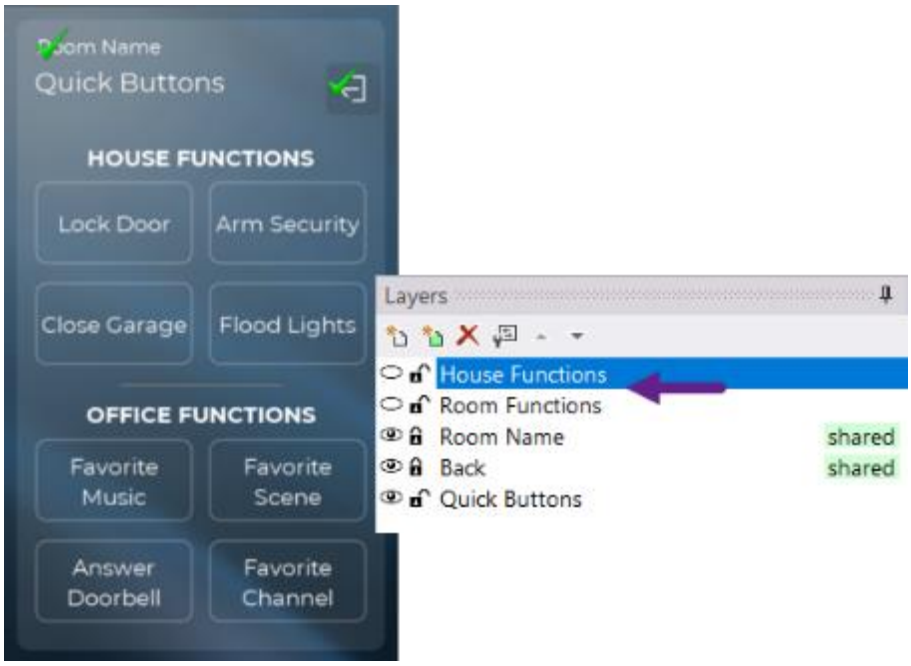
2. Select the layer and select **Delete Layer** in the **Shared Layer** window.

3. An alert will indicate that the layer is used on other pages. Select **OK** to confirm the selection, or select **Cancel**.

Moving Buttons to Layers

Dragging and dropping buttons or graphics to the wrong layer can happen if the proper layer is locked or not selected. In situations where you want to move a button or a series of buttons to another layer, there are two methods.

Scene buttons were placed on the baseline layer in this user interface. Two new layers were created to separate the functionality of the house and the room, and buttons must be moved to the corresponding layer.



Method 1: Cut and Paste

1. Select the first set of buttons that need to be moved by holding down the **SHIFT** key and selecting the buttons with the mouse key.

Selected buttons will display white handles. The last button selected will display red handles and is considered an **anchor button**.

2. Right-click anywhere on the screen and select the **Cut** function or **CTRL-X**.

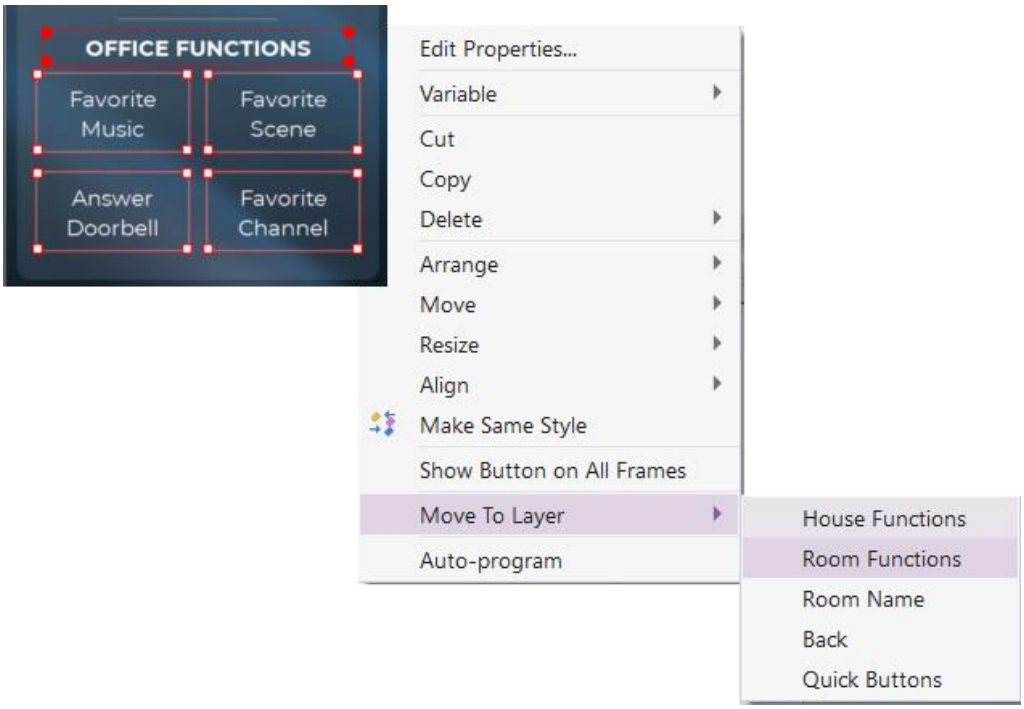
3. Select the layer to which you want to transfer the buttons, ensuring the layer is **visible** and **unlocked**.

4. **Paste** the buttons by selecting an area of the screen not occupied by a graphic or **CTRL-V**.

Method 2: Move to Layer Command

The **Move to Layer** command provides an easier way of moving screen elements to other layers.

1. Select the layer buttons that need to move to another layer.



2. Right-click and select the **Move to Layer** option on the menu. Select the layer from the list of available layers to move the selected buttons. Destination layers do not need to be unlocked or visible to move buttons.

Pro Tip: It may be easier to **hide** the other layers when copying and pasting between or creating layer content. **Lock** layers when you are finished creating them to avoid accidental changes.

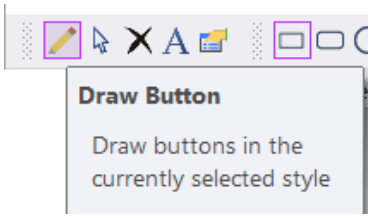
Working with Buttons

RTI features the fastest custom installation in the market today. RTI installers do not need to be experts in expensive graphic suites that require a steep learning curve. Buttons are the centerpiece of delivering the end-user a functional, cohesive, convenient, and enjoyable interface. Buttons may be drawn and imported from the included bitmap libraries. Third-party graphics and personal graphic files could provide a more personalized user interface and experience.

Drawing Buttons

Buttons may be drawn using the **Draw Button** mode in the palette toolbar.

1. Select the **draw button** selection mode.

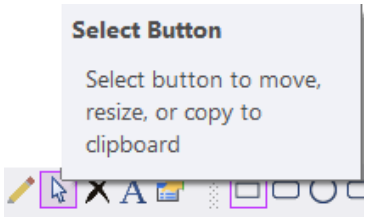


2. Select the **shape** for the button you want to draw. In this example, an option exists for a rectangle, a rounded-corner rectangle, a circle, and a 3D button.

   Drawing features may be different on RTiPanel and other device types.

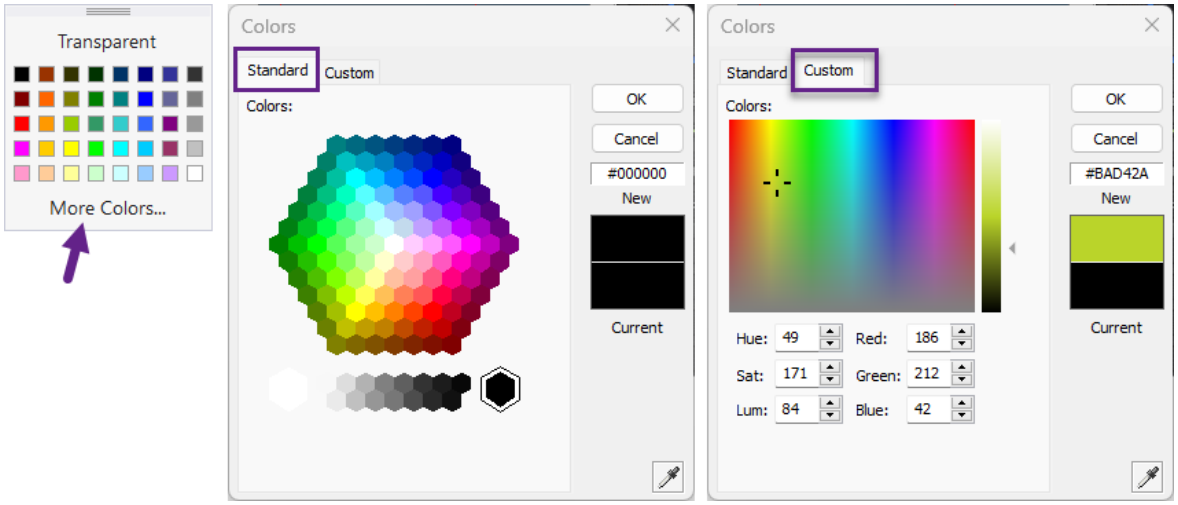
3. Holding down the left mouse, **draw** the button on the user interface selected.

4. The button may be transparent or contain a color that is not desirable. Once the button is drawn, return to selection mode by selecting the **Select Button** mode.



5. In button selection mode, the button may be selected or **resized**. Grab the frame handles by pressing the left mouse button and pulling the button by the red axis points to resize a button.

6. Select the **Active State Fill Color** to give the button a different fill color.



Pick a color from the **color palette** or select the **transparent** option if the button color is not desired. If additional colors are required, select **More Colors** from the button palette.

Pick a color from the color selection tool on the standard tab, then select a shade on the bottom. You can view or enter a color **hex code**. A **preview** of the color is available on the right side and will display the current and new color selections. Use the **eyedrop tool** to select a color in the software if color matching is desired.

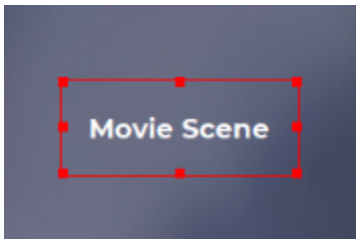
Selecting the **custom** tab will allow further customization of the color selection. Selecting a color from the spectrum and adjusting the **level** is possible. Hue, Sat, Luminance, Red, Green, and Blue levels may be entered manually. Use the **eyedrop** tool to select a color in the software if color matching is desired.

7. Select **OK** to confirm your color selections.

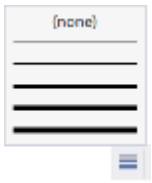
Changing Border and Font Colors

Adding a border to a drawn button on certain controllers is possible in Integration Designer software. The color palette toolbar may change borders, text, and icon colors.

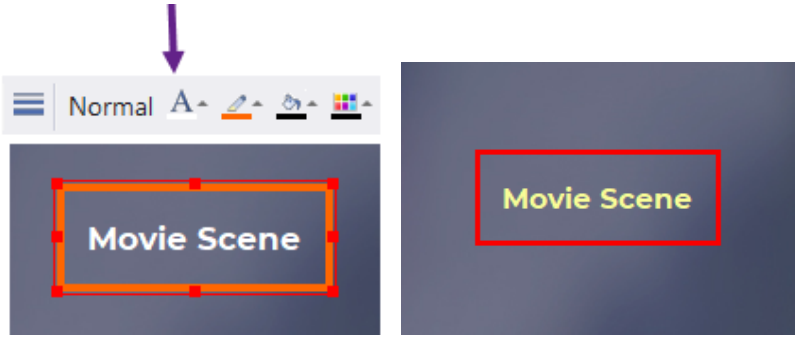
1. Select the drawn button you want to change in **select button** mode.



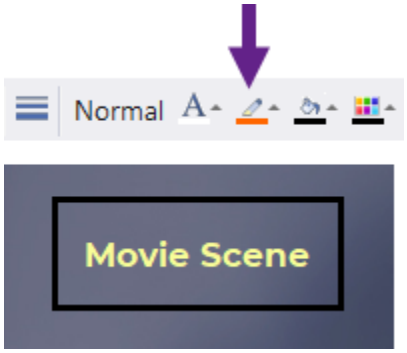
2. Select the **Line Width** icon to set the border line width. The border color will use the last selected color. If no border is desired, select **(none).**



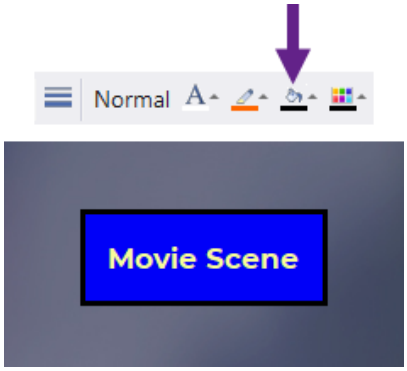
3. Select the **text** color and change it to a desired color.



4. Select the **border** color and change to a desired color.



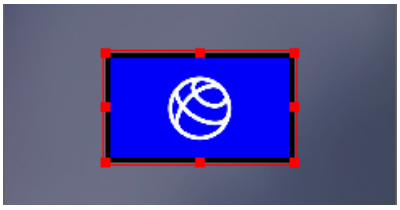
5. Apply a **fill** color to the button and evaluate the results.



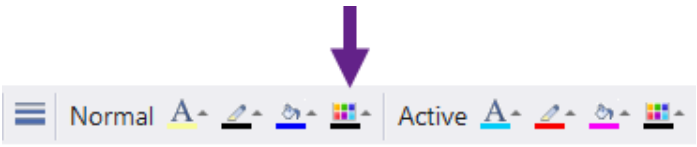
Pro Tip: Color a complimentary but contrasting look on the active state button when creating buttons for an improved client experience. Buttons can also be drawn to report essential information, titles, and feedback. If drawing buttons or text boxes for informational purposes, disable them to the touch in the button properties.

Changing the Icon Color Overlay

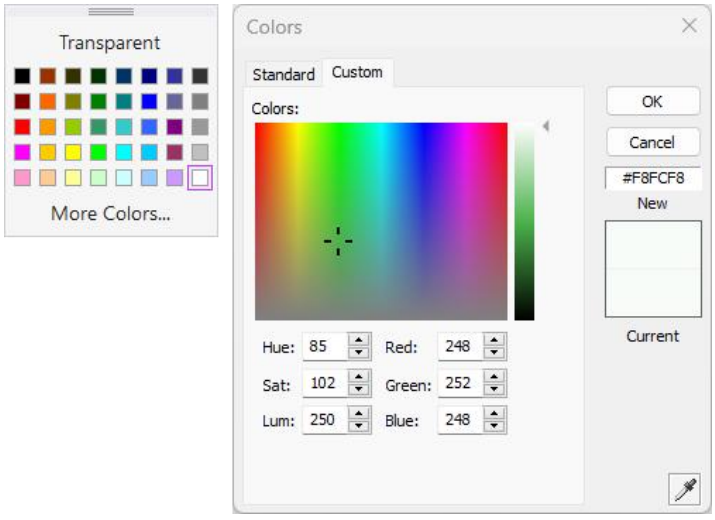
Once a button is drawn, it is possible to drag and drop an icon from the available bitmap library to the button. The icon will embed inside the button and take on some of the button properties. If the icon is a white greyscale, change the color using an overlay.



1. Select the Icon Overlay Color in the color tool palette and select a desired color.



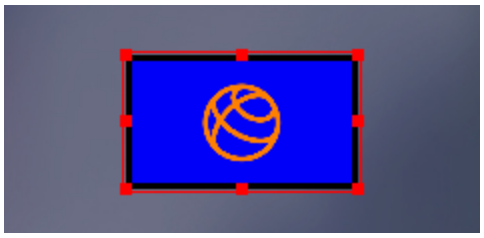
2. Evaluate the new color and make any changes using the **custom color palette**.



You may use the **eyedrop** tool to select a matching color in the user interface.

In addition, you can adjust the **color level** if the shade needs to be adjusted up or down.

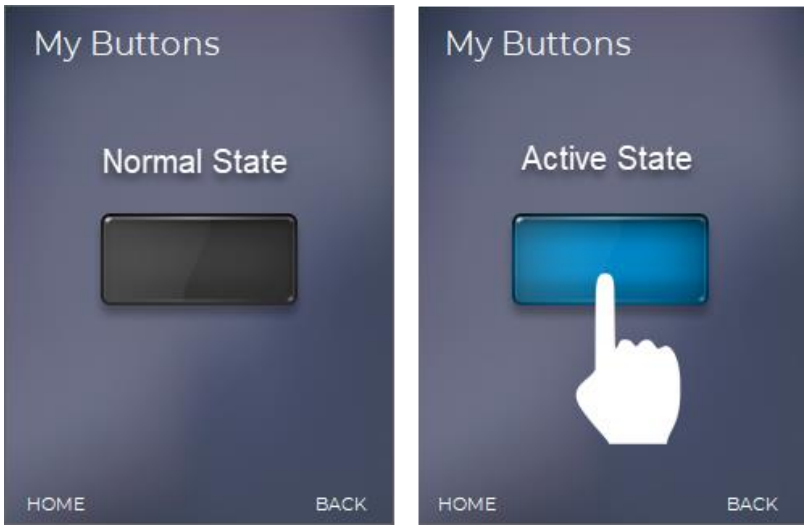
3. Change the active state's color level, which the icon will change to momentarily when pressed.



Pro Tip: Try to collect and download white icons from the internet. You will not have to carry assorted color collections. They can be changed in the software.

Active and Normal State Buttons

Ensuring a normal and active state is the best practice when adding buttons to a user interface. In the example below, a normal and active state button is illustrated.



In the **normal** state, the button looks like the picture on the left. When the end-user presses the button, it changes color to the **active** state until it is released. The brief color change is a visual confirmation that the button was pressed. If the button had a normal and active state with the same color, it would not appear to the end-user that the button was pressed.

Ensure a normal and active state when drawing or placing buttons in the software. The text, border, and fill color on a drawn button may be observed in the **color palette** toolbar.

When drawing a button, you can view and change the normal and active state color combinations using the tool palette.

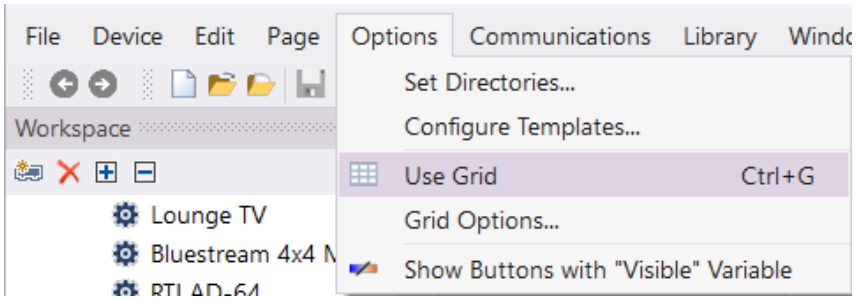
Certain controllers, such as RTiPanel, will have limited color palette functionality.

Buttons provided in the bitmap libraries or imported into the software can be viewed in the **button properties.**

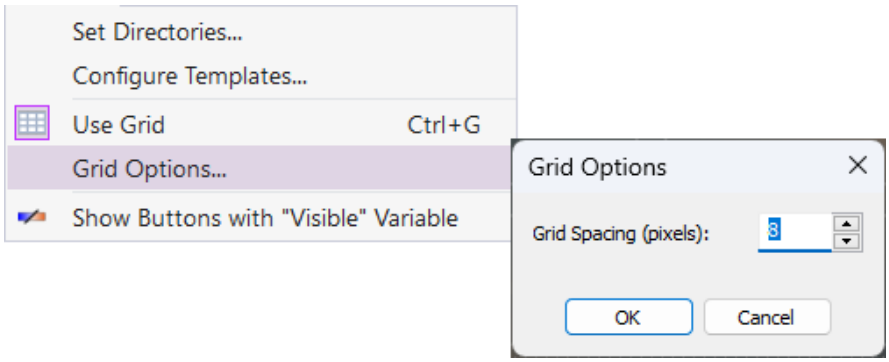
Aligning Buttons

The **Use Grid** command on the **Options** menu toggles the alignment grid on and off when editing color device files. With the alignment grid off, you can drag buttons to any position on the screen. With the grid on, the buttons automatically snap to the nearest grid position as you drag them. You can adjust the grid spacing using the **Grid Spacing** commands on the Options menu.

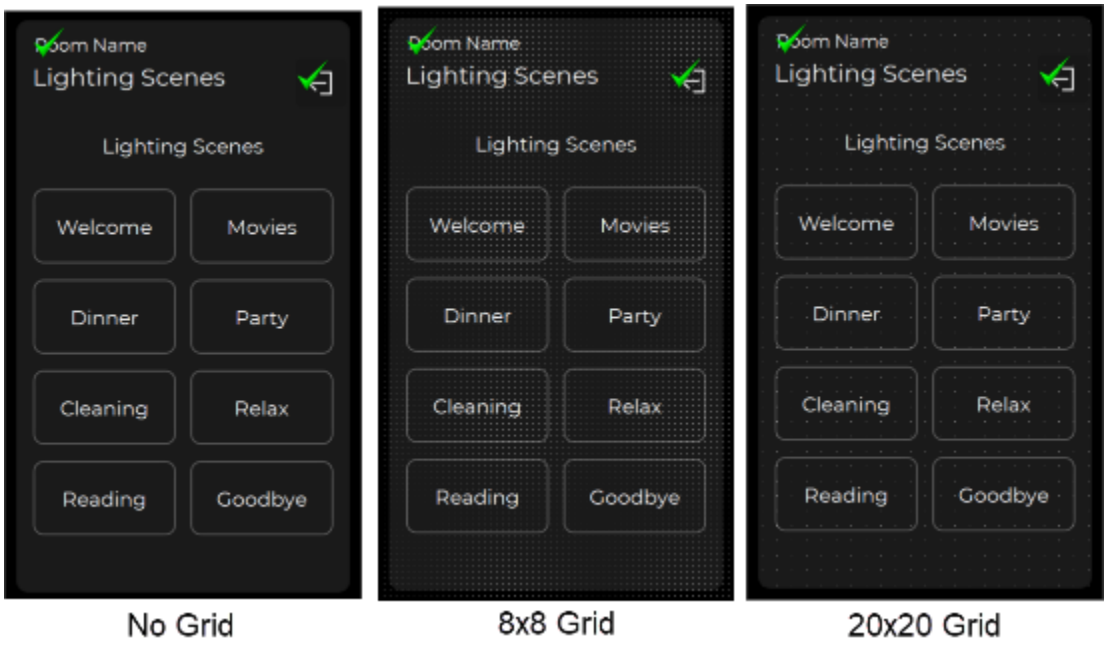
1. Select Use Grid from the options toolbar.



2. Select Grid Options and change the grid spacing in pixels depending on preference. The default grid size is 8x8.



3. Turn the grid on and off as needed.



Moving Buttons

1. Select the button tool in the tool palette.

2. Select a button or a group of buttons by holding down the SHIFT key and selecting the left mouse button.

3. Drag the buttons to the location and release the mouse button to place the buttons.

4. Use the **Arrow buttons** from the **Alignment Toolbar** to move a selected button group of buttons one pixel at a time.

Arranging Buttons

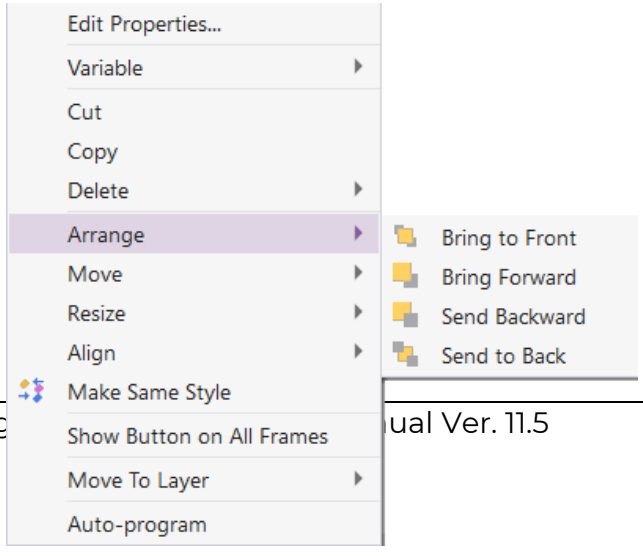
The selected button(s) can be arranged on the touchscreen of a controller using the tools in the alignment bar. You may also adjust a button's alignment.

Bring to Front – Click this button to make the selected button(s) appear before all other buttons, avoiding overlap.

Send to Back – Click this button to make the selected button(s) appear behind all other buttons.

Bring Forward – Click this button to move the selected button(s) closer to the top of the bottom order. Since the button order includes all buttons on the page, you may not see any change after clicking this button if the next button in the order does not overlap the selected button.

Send Backward – Click this button to move the selected button(s) closer to the bottom of the button order. Since the button order includes all buttons on the page, you may not see any change after clicking this button if the next button in the order does not overlap the selected button. Click this button repeatedly until the selected buttons are at the desired position in the button order.



You may also arrange buttons by using the right-mouse button context menu.

Select the button(s) you want to arrange, then right-click for the context menu.

Aligning Buttons

For touchscreen buttons, you can also use the tools on the Alignment Bar to arrange many buttons with a single operation. You must have more than one button selected to enable these tools. In addition, the tools for the Alignment

Bar uses the concept of an anchor button when multiple buttons are selected. To change the anchor button among the selected buttons, hold down the SHIFT key and click on the button that you wish to make the selection's anchor. The anchor button is indicated by filled red rectangles in the corners of its selection rectangle, while all other buttons are painted with white and red rectangles in their corners.

The three buttons on the bottom right row must be aligned with the bottom right button.

Select the buttons you want to align, then select the button you want to use for alignment. The last button you select is the **anchor button** and will have red frame handles.

Select the Align Top icon from the alignment toolbar to align the three selected buttons to the anchor button.

The buttons now have a symmetrical alignment.

Align Left - Select this button to move all the selected buttons so that their left edges align with the left edge of the "anchor" button. The buttons will be moved horizontally but not vertically by this command.

Align Center - Select this button to move all the selected buttons so that their horizontal centers line up with the horizontal center of the "anchor" button. The buttons will be moved horizontally but not vertically by this command.

Align Right - Select this button to move all the selected buttons so that their right edges align with the right edge of the "anchor" button. The buttons will be moved horizontally but not vertically by this command.

Align Top - Select this button to move all the selected buttons so that their top edges line up with the top edge of the "anchor" button. The buttons will be moved vertically but not horizontally by this command.

Align Middle - Select this button to move all the selected buttons so that their vertical centers line up with the vertical center of the "anchor" button. The buttons will be moved vertically but not horizontally by this command.

Align Bottom - Select this button to move all the selected buttons so that their bottom edges line up with the bottom edge of the "anchor" button. The buttons will be moved vertically but not horizontally by this command.

You must have more than one button selected to use this command.
These features are also available from the right-button context menu.

Resizing Buttons

Buttons and other graphics can be resized directly in Integration Designer without having to do so using third-party software. Integration Designer uses a 9-slice scaling method when developing template graphics. RTI installers can create 9-slice graphics for personal use by viewing online tutorials.

In this example, the buttons are too small. The height and width of the buttons need to be increased.

1. Select the **Select Button** tool from the tool palette.

2. Select the button you wish to re-size by clicking on it with the left mouse button. Note that to resize a button, only one button can be selected. If the selection can be resized, small boxes will appear on the red border on all four sides.

3. Position the mouse over the edge of the button to resize, click the left mouse button, and drag the mouse to change the size.

4. Release the mouse button when you have the desired size.

You can use the **Resize Tools** on the **Alignment Toolbar** for touchscreen devices to resize many buttons with a single operation. In addition, the Resize Tools use an **anchor** button when multiple buttons are selected.

To change the anchor button among the selected buttons, hold down the SHIFT key and click on the button that you wish to make the anchor button of the selection. The anchor button is indicated by filled red rectangles in the corners of its selection rectangle, while all other buttons are painted with white and red rectangles in their corners.

Make Same Width - Click this button to make all selected buttons the same **width** as the "**anchor**" button, leaving their heights unchanged.

Make Same Height - Click this button to make all selected buttons the same **height** as the "**anchor**" button, leaving their widths unchanged.

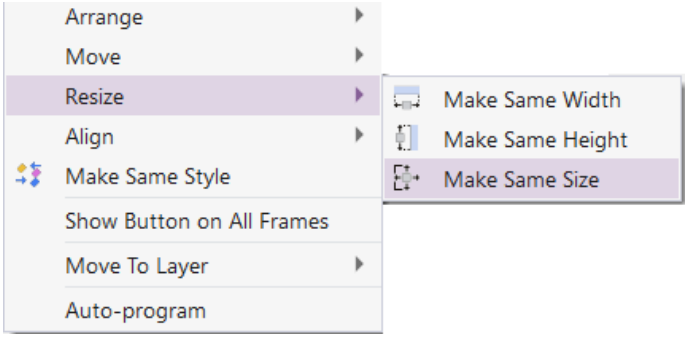
Make Same Size - Click this button to make all selected buttons the same **width** and **height** as the "**anchor**" button.

Make Same Style - Click this button to make all selected buttons the same **style** and **color** as the "**anchor**" button.

These features are also available from the right-button context menu.

1. Select the buttons you want to resize, selecting the anchor button last. You will resize the buttons selected using the proportions of the anchor button.

2. Right-click the selection or select the preferred resize tool in the alignment toolbar. The button can be resized to the anchor button's width, height, or size.



3. Once the buttons are resized, use the alignment tools to realign the buttons on the screen.

Equalizing Spacing and Distributing Buttons

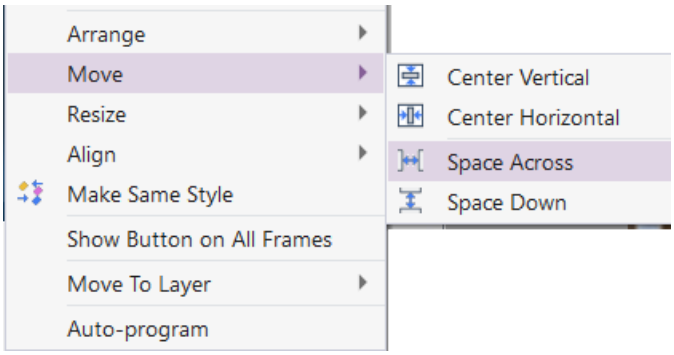
When working with a user interface, it is important to ensure that buttons are equally spaced or distributed across the screen.

In this example, the buttons must be evenly spaced across the screen.

1. Select each button by holding down the **SHIFT** key and selecting it with a left-mouse click.

2. Right-click and select **Move**, then **Space Across**. Alternatively, you may use the **Equalize Spacing Across** tool in the alignment toolbar.

   If buttons are **vertically** spaced, use the **Move**/**Space Down** context menu, or you may use the **Equalize Spacing Down** tool in the alignment toolbar.



There may be situations where horizontally or vertically placed buttons require equal spacing between the borders and buttons.

In this example, the buttons are too close together, and extra spacing between the right and left border.

1. Select each button by holding down the **SHIFT** key and selecting it with a left-mouse click. Selecting an anchor button is unnecessary if you distribute buttons

across the entire screen width. If the page has a border and you want to distribute them within the border, select the border as an **anchor** selection.

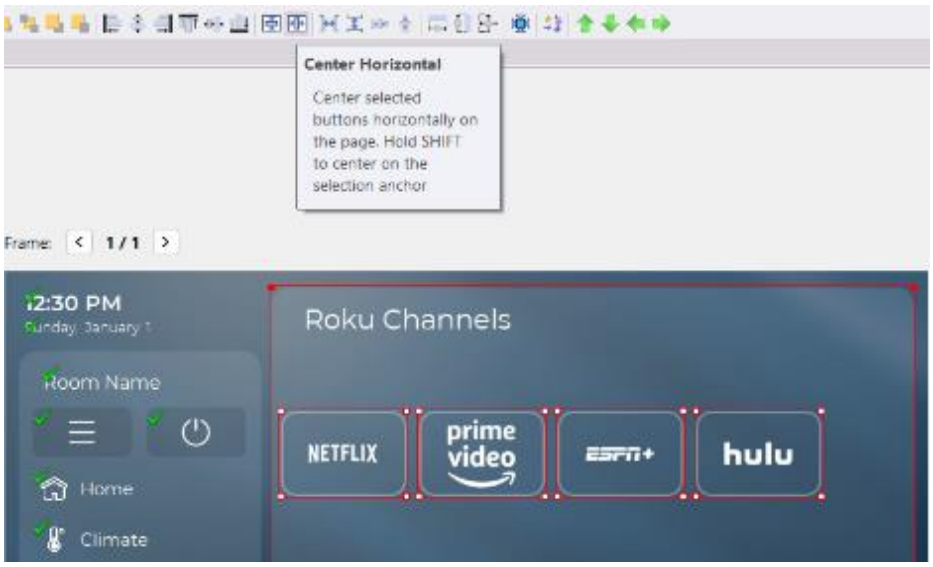
2. Select the **Distribute Buttons Across** tool in the alignment toolbar. Identify your client program files later.

Use the **Distribute Buttons Down** tool in the alignment toolbar if buttons are placed **vertically**.

Centering Buttons

Centering buttons horizontally and vertically on the screen is often necessary. In this example, buttons are not centered on the screen horizontally or vertically.

1. Select the buttons that you want to center. The buttons must be centered on the page border, not the full screen. Select the page border as the anchor button.



2. Select the **Center Horizontal** tool in the alignment toolbar while holding down the **SHIFT** key to center on the selection anchor.

3. If vertical centering is desired, select the **Center Vertical** tool in the alignment toolbar while holding down the **SHIFT** key to center on the selection anchor.

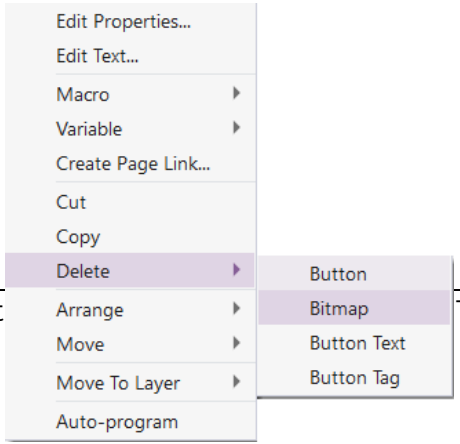
Deleting Buttons

1. Select the **Delete** tool from the Tool Palette.

2. Place the mouse cursor over the button you want to delete.

3. Click the left mouse button.

4. Select the **Delete Button** from the pop-up menu.



Several button characteristics may be deleted.

Button – Delete the entire button and all contents.

Bitmap – Delete the bitmap but preserve the button footprint and other elements.

Button Text – Delete the text on the button.

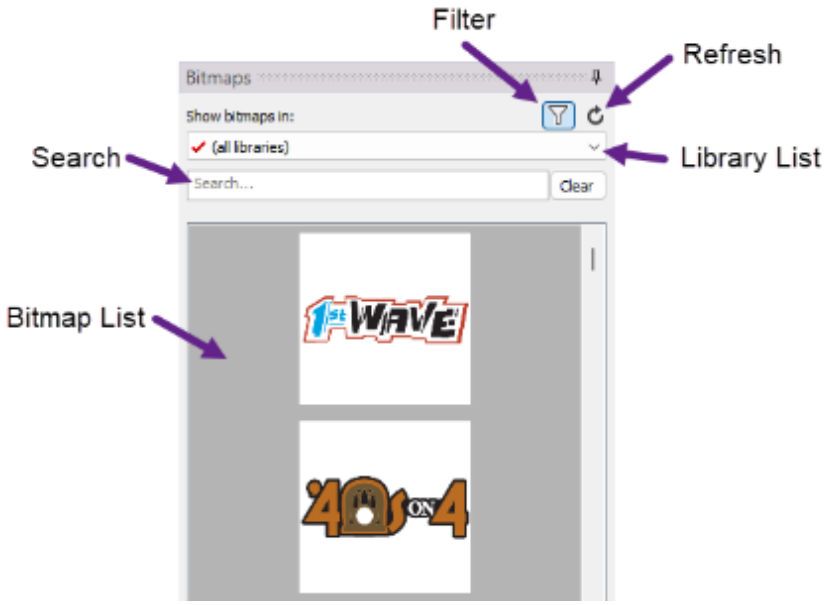
Button Tag – Delete the tag name on the button.

Introduction to Button Graphics

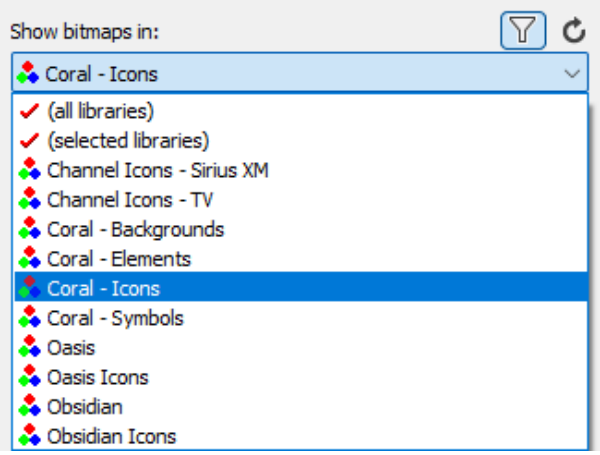
Integration Designer features templates and assets to help build personalized interfaces without the need for mastering a graphic editor application. Designing interfaces with custom buttons, text, and other elements is possible using a drag-and-drop approach.

Included Bitmap Libraries

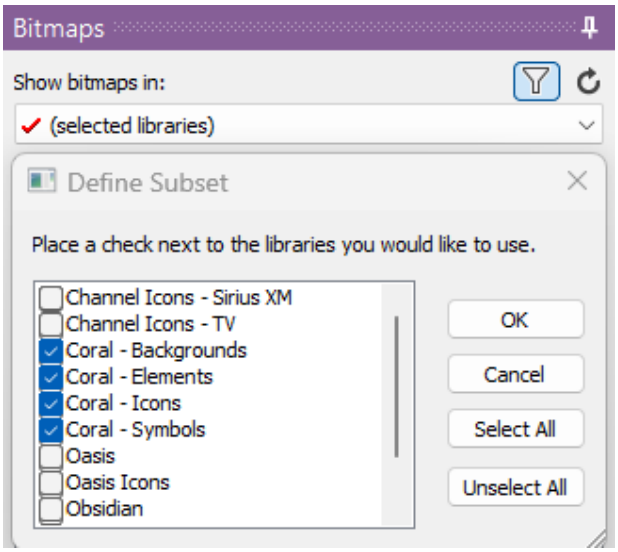
Select **Bitmaps** from the **tabbed menu** in Integration Designer.



Library List: A dropdown menu where a template category can be selected. Select a specific template to make searching and navigating easier. Selecting **All Libraries** will search for criteria collectively through all templates.



Selected Libraries may be selected to search and view multiple bitmap libraries. Select Selected Libraries and select the templates that will be viewed.



Select All will select all available bitmap libraries. **Unselect all** will deselect all bitmap libraries.

Click **OK** after making selections.

Search: Enter text criteria to search and return bitmaps matching information in the title.

Bitmap List: Displays all the graphic collections for the current library selection. Graphics may be dragged and dropped into the user interface onto the selected controller page.

Refresh: Refreshes the bitmap list after adding new bitmaps to a search directory or template.

Filter: Filter out images based on the current controller screen size.

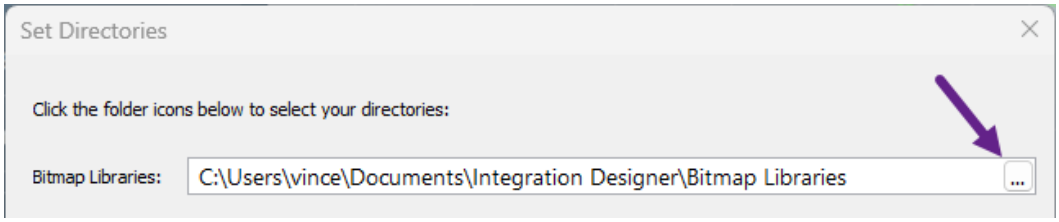
Pro Tip: Browse through the different bitmap libraries and become familiar with some of the graphics and naming conventions available. Drag and drop some graphics to the interface for practice.

Using Personal Bitmap Libraries

The Image Tab allows image files (*.bmp, *.jpg; *.jpeg, *.png, *.gif, *.tif; *.tiff, *.ico) to be imported directly to the color touchscreen controller. RTI installers can use personal graphics downloaded from the internet, purchased through a graphic developer, or created in-house. These graphics may be placed on a computer and retrieved in the bitmap library.

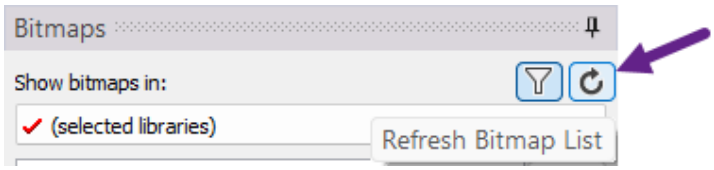
1. Select the **Options** tab, then **Select Directories** in the main toolbar of Integration Designer.

2. Select the **menu** icon to the right of the Bitmap Libraries directory and navigate to the directory where personal graphics are stored.

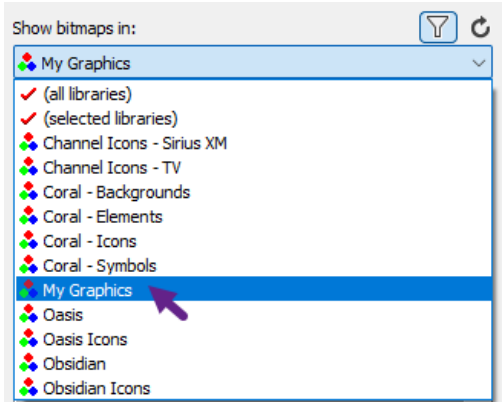


3. Select **OK** to save your changes.

4. Go to the bitmap tab and select **Refresh** to refresh the bitmap libraries.



5. Select the personal folder in the **library list** drop-down menu.



6. Drag and drop graphics to the controller interface.

Understanding Graphic Objects

RTI installers can use included graphic objects for special functionality in the interface. These have been created to provide normal and active state buttons, special wallpapers, and feedback such as item lists, sliders, gauges, dynamic images, and more. These graphics can be identified as a symbol in the bitmap library.

	Background	The graphics are embedded in the background and treated like controller wallpaper.
	Resizable Images	Resizable images are scaled up and down to match the button size.
	Dual State Bitmap	The graphic contains a normal and pressed active state.
	Bitmap Groups	Bitmap Groups allow a set of related bitmaps, such as a navigation pad, to be dragged and placed on the interface in a single unit.
	Image List	Image Lists provide a sequence of images where only one image is shown at a time based on the value of an integer variable.

	Item List	Item Lists are a two-way list of data entries, such as a list of song selections that may be highlighted and selected.
	Graphs	Graphs represent integer system variables graphically, as a gauge or bar graph object.
	Toggle Buttons	Toggle buttons consist of multiple images and show feedback based on a Boolean system variable.
{None}	**Icons**	Icons may be embedded inside scalable or dual-state bitmaps and do not have a symbol. They can be resized and are not primarily used as a button.

RTI installers can create these bitmap types. Please consult the **Image Group File Format 2.0 document** for instructions.

Pro Tip: Controllers may be incompatible with certain objects. Ensure you are using the correct bitmap for the controller selected.

Embedding Icons in Buttons

Buttons marked with a **Resizable Image** or **Dual State Bitmap** may contain icons not represented with a symbol in the bitmap library.

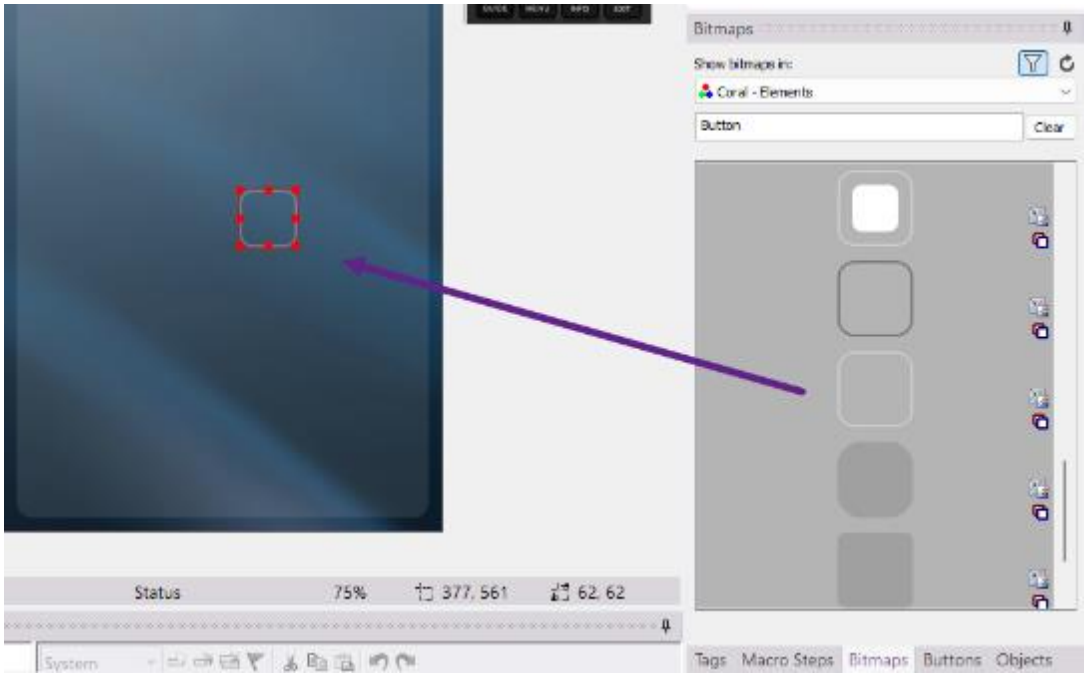
To create a button with an icon:

1. Select a **controller** and **page** that can be used for practice purposes. Creating a blank pageset and naming it “practice” is recommended.

2. Select the **Coral - Elements** bitmap library in the bitmap library list.

3. In the **Search** box, enter “Button” to find a button.

4. Browse through the list of available buttons. Choose a **resizable** button with a **normal** and **active** state indicated by the object icons. Drag and drop the button to the interface, ensuring the correct layer is selected, visible, and unlocked.



5. **Resize** the button appropriately using the frame handles.

6. Select the **Coral - Icons** library in the bitmap library. Browse through the library or search for certain icons as preferred.

7. **Drag and drop** the **icon** directly onto the button added in the previous step.

The icon will be embedded inside the button and take on the button's properties. Since a button is a special object and the icon does not have an object symbol, it is treated as such.

8. Continue to drag and drop different icons on the same button, noticing how the icon is seamlessly replaced.

Changing Buttons

Buttons may also be replaced while preserving the embedded icon.

1. Select the **Coral - Elements** bitmap library in the bitmap library list.

2. Find another **resizable** button in the list with a **normal** and **active** state.

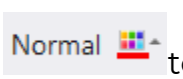
3. Drag and drop the new bitmap into the user interface, overlaying the existing bitmap containing the icon.

   Notice that the new button will replace the existing button, preserving the icon. If the button has different properties, the icon will adopt them.

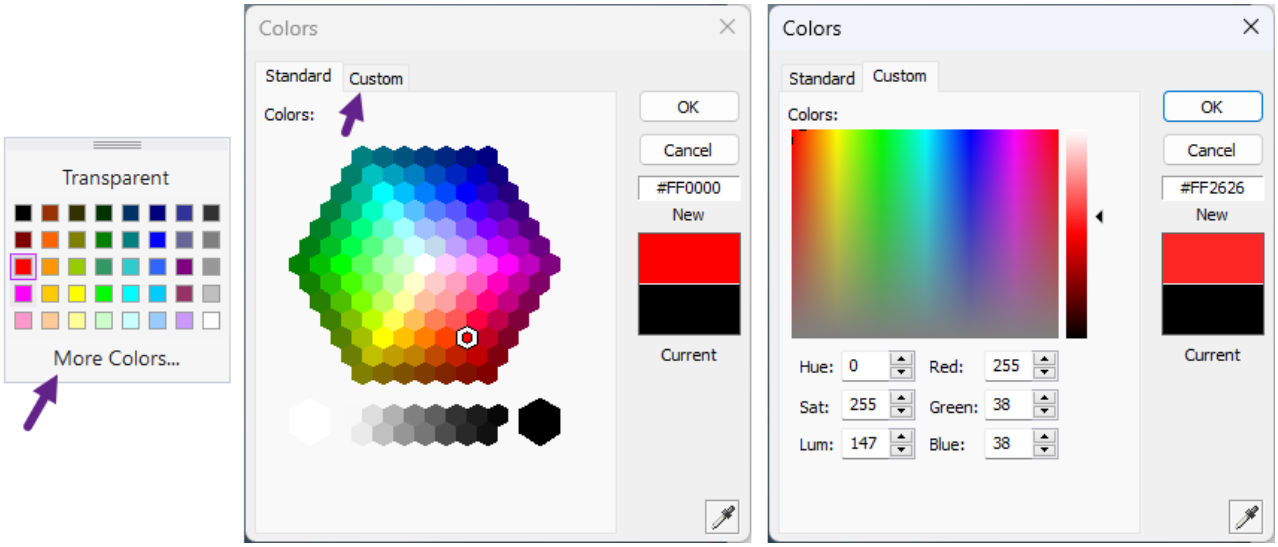
Changing Icon Colors

It is possible to change the color of a white, greyscale icon.

1. Select the icon using the **select button** mode in the selection toolbar.

2. Select the **icon overlay** 

tool for the **normal state** in the palette toolbar.



3. Select the **color** in the color palette. Select **More Colors** for the standard color palette if color matching or customization is required. Select the **Custom** option in the toolbar if shading or color coordinates require further tweaking.

4. Select **OK** to apply the color changes.

   The icon will reflect the new color overlay.

5. To change the color of the active state when pressed, select the Active State Icon Overlay

tool in the toolbar and repeat the same process.

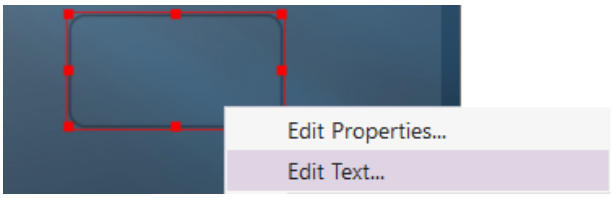
Editing Button Text

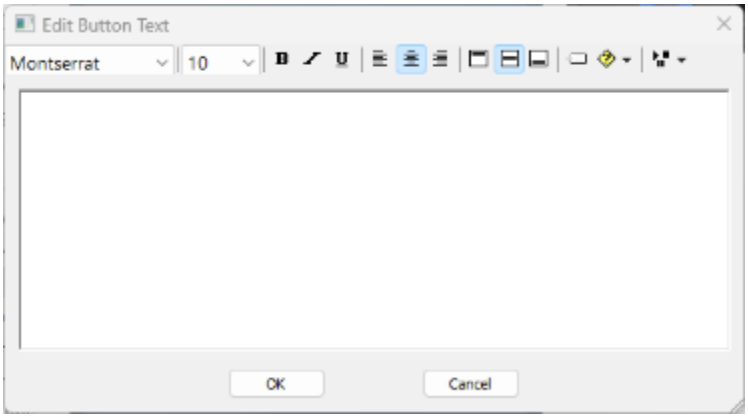
Text may be added to any button as an alternative or in addition to icons.

1. Select the button in **select button** mode.

2. If an icon is inside the button, **right**-**click** and select the **delete**/**Button Icon** option in the context menu.

3. **Right-click** on the button and select **Edit Text**.



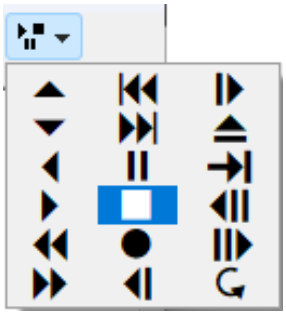


Text may be entered and formatted using the provided toolbar options.

4. Type in text into the Edit Button Text window. Browse the toolbar options and customize the font type, size, and style.

Montserrat	**Font Selection**	Select from a list of fonts in the dropdown menu.
10	**Font Size**	Select from a list of font sizes in the dropdown menu.
B I U	**Style**	Select Bold, Italic, or Underlined
	Text Tag	Insert a text tag for button text feedback.
	Control Variable	Insert a control variable for quick text feedback.
	Symbols	Select from a library of transport symbols as an alternative to graphics.

When using buttons with transport commands, use the symbols option instead of looking for graphics. An assortment of symbols is available for common functions.



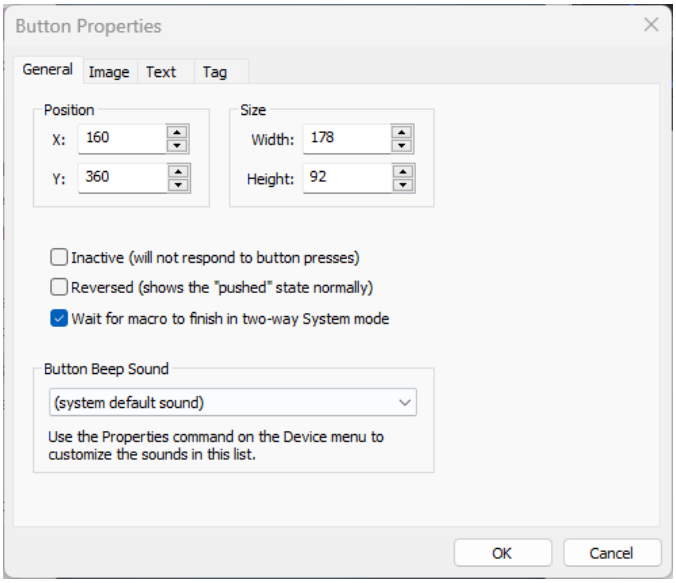
Pro Tip: Select the **Edit Text** tool in the **tool palette** toolbar to edit the text on several buttons. Selecting each button will take you to the **Edit Text** window.

Button Properties

The Button Properties window contains several options to impact the look and functionality of your user interface.

1. Select a button using the **select button** mode in the selection toolbar. Ensure the button contains an **icon** to realize the features of the button properties.

2. Right-click on the button and select **Edit Properties.**



GENERAL TAB

Position: To move a button to specific screen coordinates, enter the **X** and **Y** coordinates for the screen position on the controller.

Size: Change the button to a specific size in pixels by entering the **Width** and **Height** of the button.

Inactive: Select the **Inactive** option if you do not want it to respond to presses. They were used for informational text boxes or buttons.

Reversed: Selecting this option will invert the normal and active state. The active state will display normally, and the normal state will display when pressed. You can use this option to temporarily view what a drawn button will look like in the reversed state.

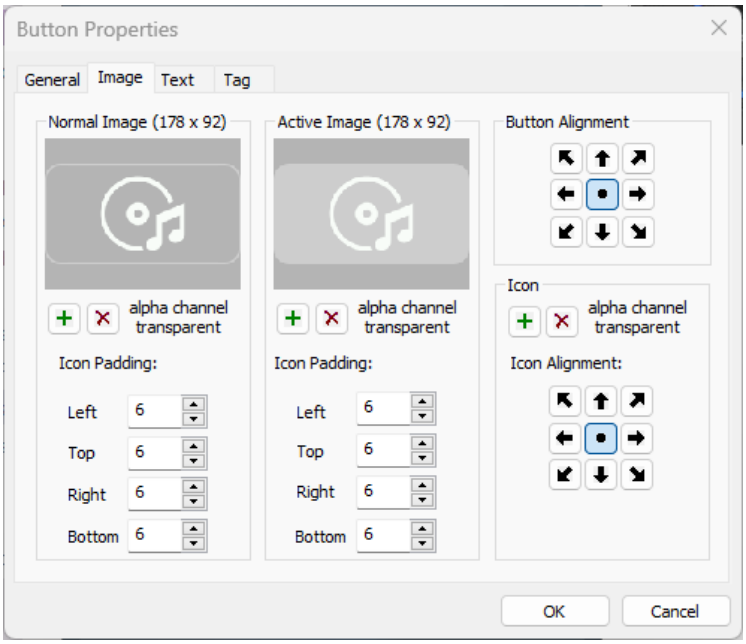
The **Wait for a macro to finish in two-way system mode** is selected by default and will ensure a macro will run before sending additional commands to a processor.

Button Beep Sound: Select a sound previously added in the sound properties of the controller if you want the button to play the sound when pressed. Select a sound from the available list of sounds in the dropdown menu.

3. Select OK to apply your changes after selecting any options above.

IMAGE TAB

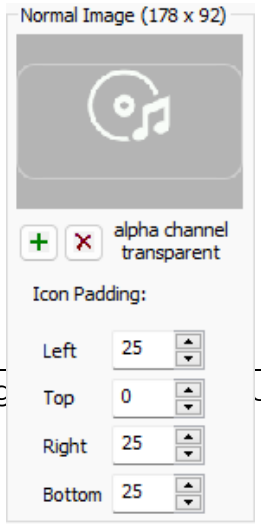
1. Select the **Image Tab** in the **Button Properties** window.



This window will display how a button will appear in the normal and active state. The icon's appearance in the button can be changed without requiring icons of varied sizes and padding.

Add or **remove** a normal or active state bitmap. Selecting the Add button will navigate to a directory on a PC to add another bitmap.

2. Use **Icon Padding** to increase or decrease spacing in pixels to the image's left, right, top, and bottom. Adding pixels will decrease the icon's appearance within the button, while decreasing the pixels will make the icon appear larger.

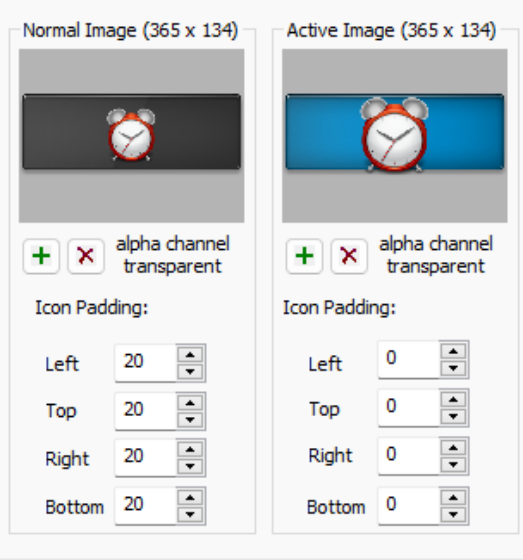


When changing padding, a **preview** can be seen in the normal or active state.

Making some room for text on the bottom or top of the button is possible using **Icon Padding**. In this example, spacing has been added to the button except for the top region. There is room for text to be entered below the icon.

Icon Padding may be adjusted if the icons occupy too much of the bitmap, creating an unappealing user interface.

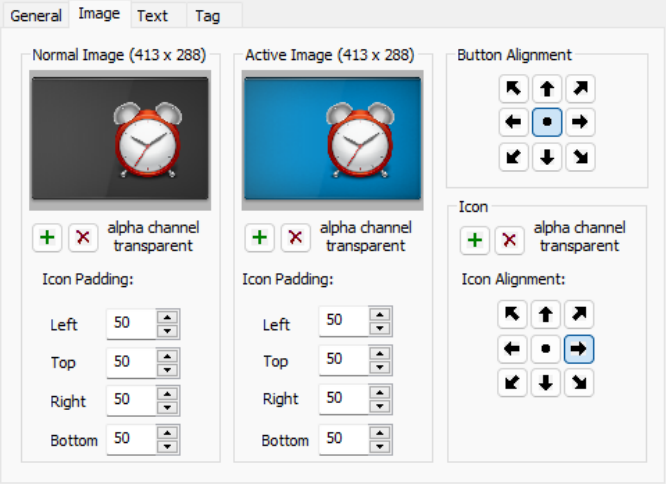
Increasing the padding on each side will make the icon appear smaller inside the button.



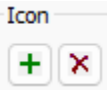
In this example, padding has been added to the **normal state** only. When the button is pressed, the active state will display with an icon "exploding" effect.

This method is useful when the normal and active state buttons do not contrast enough.

It is also possible to align both the button and the icon inside of the button if sized properly. The bitmap is centered in this example, but the icon is placed to the right.



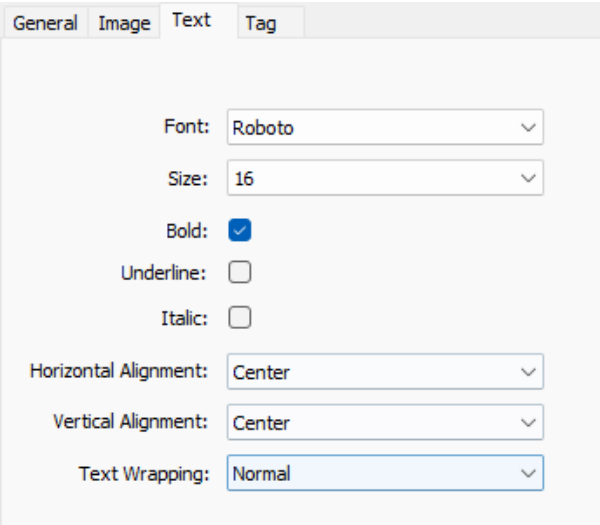
3. Select **OK** to apply any changes.



Select **Add** to navigate to a directory on your PC and import a new icon. Select the **Delete** button to remove the icon from inside the button.

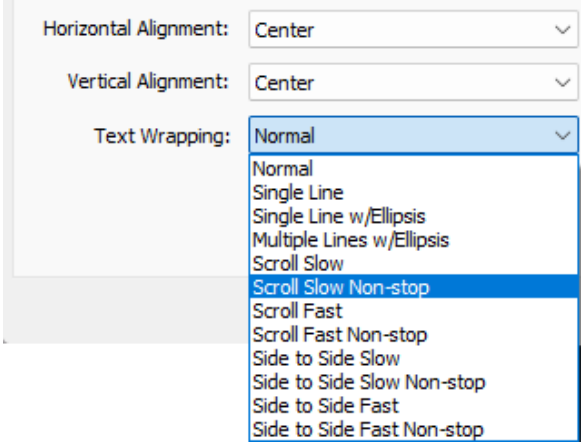
TEXT TAB

Select the text tab to revise some of the text attributes on the button.



Select a font type, size, style, and alignment.

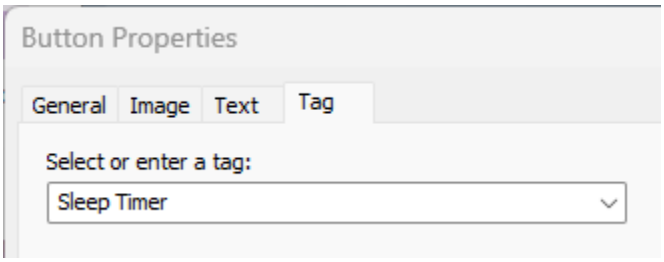
Text Wrapping is available for certain controllers. Select the Text Wrapping dropdown menu to display the different wrapping effects.



Use this option if the text is cut off due to the length of the text string. This option is useful for feedback on long song titles, albums, or other meta-data.

TAG TAB

You may **view**, **add**, **delete**, or **edit** the tag name of the button in the **tag tab** section.



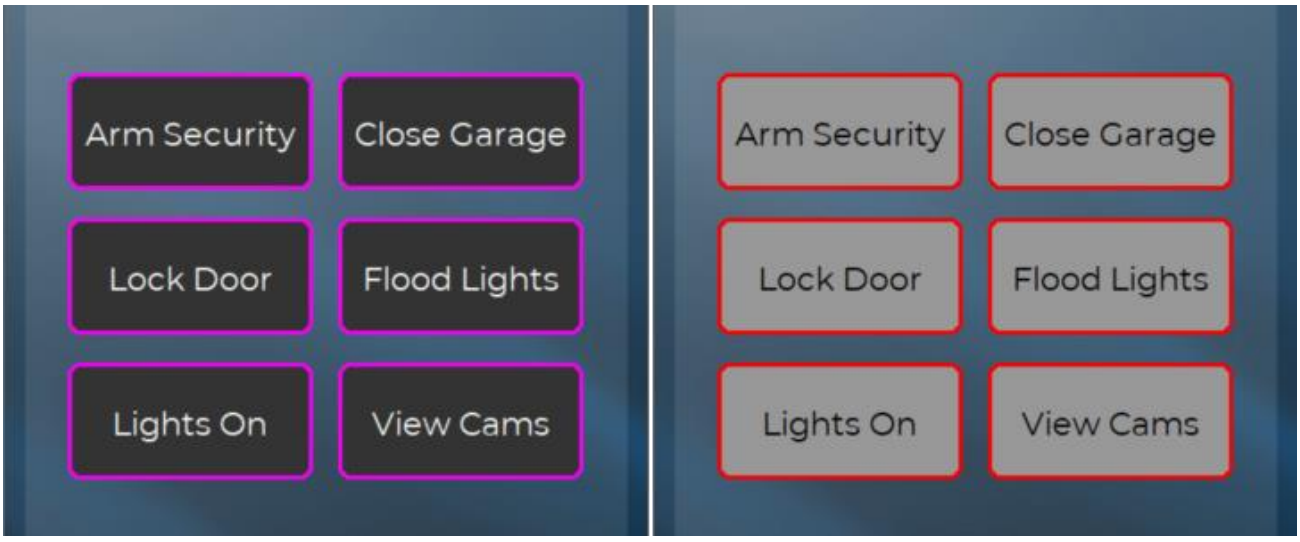
Use the dropdown menu to select an available tag name or enter the tag name manually.

This function is more commonly performed using the **tag window** in the **Button Tags Toolbar** in Integration Designer.

Button Styles

When making changes to multiple buttons, it is possible to copy the style of a single button and transfer the properties to other buttons in the interface.

In this interface, buttons have been drawn with matching font styles, sizes, fill colors, and borders for the normal and active states.



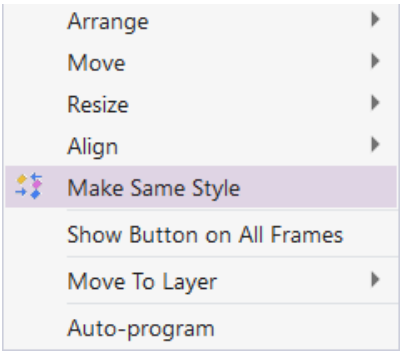
1. Select a button using the **select button** mode in the selection toolbar.
2. Right-click on the button and select **Edit Text**.
3. Change the **font style** and **size**.
4. Make the font **bold**.
5. Select **OK** to apply the changes.
6. Change the **text**, **border**, and **fill color** of the button selected**.**
7. Next, change the **border thickness**.
8. Repeat the **color** selections for the **active button state**.

The button has been changed, but all other buttons need to match. Use the **Make Same Style** tool to apply the changes made to this button to all the other buttons on the interface.

1. Holding down the **SHIFT** key, select all the buttons you want to change by pressing the left mouse button.

2. The **last** button selected is the button you want to use to make all other buttons the same style. Select the button that was changed as the **anchor** button.

3. Right-click with the buttons selected, then select the **Make Same Style** option in the context menu.



You may also use the **Make Same Style** tool in the alignment toolbar.

The following attributes will be sent to the other selected buttons from the anchor selection:

Font Type
Font Size
Font Style
Border Color and Width
Horizontal and Vertical Alignment
Text Wrapping
Button Properties (Inactive and Reverse State, Padding, Position, etc.)

Button size, **sounds**, and **tags** are not impacted.

Macro Programming

Integration Designer features automated macro programming, allowing RTI installers to program using solution-based macro tools. Macros can be programmed precisely to accommodate the most stubborn components.

Introduction to Tags

Tags are a logical naming standard developed by RTI to name buttons and feedback. A macro can be programmed on distinct levels once a tag name has been assigned to a button. The levels decide how the functionality attached to the tag is used in the interface.

Tags have several advantages, such as auto-programming templates when added to the workspace, quick maintenance without needing revisions in multiple locations, and faster programming. In addition, many features built into Integration Designer software are built around using tags.

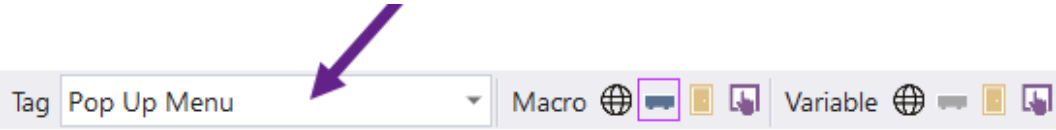
Tags are organized by device type and functionality categories.

- General
- HVAC
- Input Selection
- Keypad
- Lighting
- Media Player
- Miscellaneous
- Navigation
- Input Selection
- Presets
- Security
- Transport Buttons
- General
- Weather

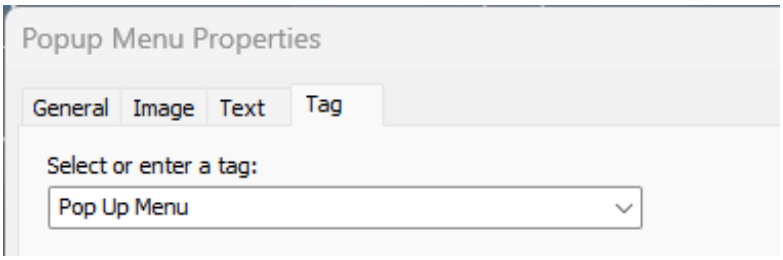
A complete list of **standard tags** can be found here: https://developer.rticorp.com/tags/taglist

Entering and Viewing Tags

Tags can be viewed, entered, or changed in the tag window of Integration Designer. When selecting a button, the tag window will display the tag name.



Tag names can also be viewed, entered, or changed in the button properties. Select the button, right-click, and select **Edit Properties**.



In both situations, tags may be named or changed using the drop-down menu. When entering tag names, there are three options:

- **Clear** the tag window.
- Enter a **standard** tag name.
- Enter a **non-standard** tag name.

Clearing the tag window will not remove the relationship between the tag name and the macro if a macro exists.

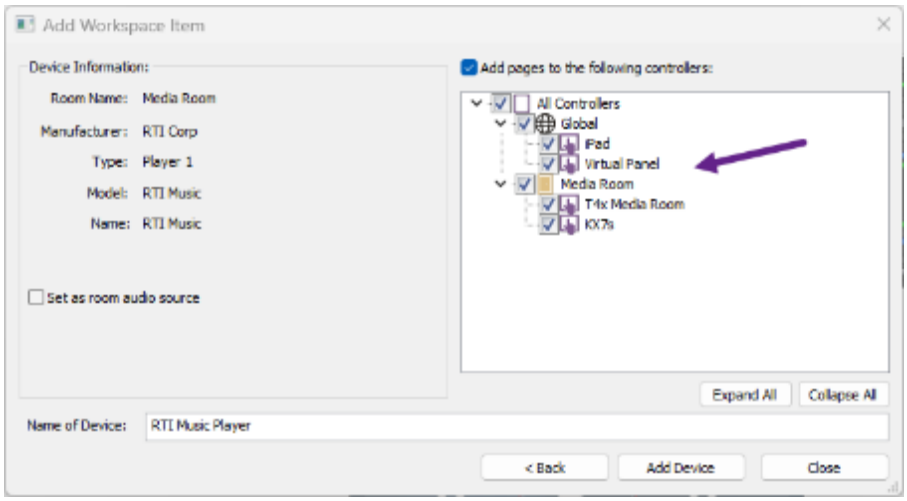
Integration Designer will cross-reference the device library when entering a **standard tag** name and populate the command if there is a match. In addition to creating the standard tag name, a cross-referencing process exists to match the tag name to an infrared or driver command. Tag names do not have to match the command name, and several match criteria can exist.

When entering a **non-standard** tag name, commands will not be automatically populated, but RTI installers may use some of the other advantages of tags.

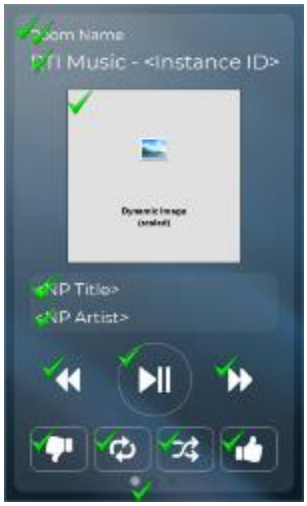
Auto-Programming Tag Usage

It is important to understand the auto-programming process, which uses standard tag names to auto-populate the commands in a user interface.

1. An infrared or driver-based source device is added to the workspace. Pages are selected for the device, and Integration Designer software supplies an available template for that device type.



2. A pre-tagged template is matched with the device added.



3. The standard tag names are matched to commands and feedback using a cross-referencing process inside the software.

4. Commands matches are found for the standard tags, populating macros, and variables to the template. Anything without a match will leave the tag populated but will not have an underlying macro or variable.

5. The programmer can use the unprogrammed buttons for other commands not populated in the template using standard or non-standard tag names.

Tag Facts

1. Standard tag names always begin with a **capital** letter. If other words are in the command, each first letter is capitalized. For example, “Pop Up Menu” is a standard tag.

2. Tags are **logically named** and come naturally to the RTI installer after some practice. For example, “Menu,” “Back,” and “Play” are examples of tag names.

3. Programming commands or feedback on a button is impossible without a tag.

4. **Drivers** are developed to match RTI’s standard tag names. RTI’s built-in **IR** database supports consistently named commands, making auto-programming possible.

5. When using tags, several advantages are realized other than auto-programming. Once a tag has been established, different macros and variable types may be programmed to determine how the tag/command relationship is used in the project file.

6. Without tags, a revision or addition to a source device command would have to be made on every controller with a user interface for that device.

7. Tags are not only linked to commands and feedback. They can also populate **page links** as well.

8. Tags will populate once a source device is added to the workspace. It also includes home pages, page sets, and devices without device libraries.

9. There are standard tag names for special functionality in Integration Designer. Some of this functionality includes menu selections, room selections, activities, and several different variables used for reporting feedback on the interfaces.

10. Tags can only be removed using the **Edit Tags** function found in the **Edit** tab in the main toolbar.

Tag Exercises

The following exercises will enforce how tags are used when programming.

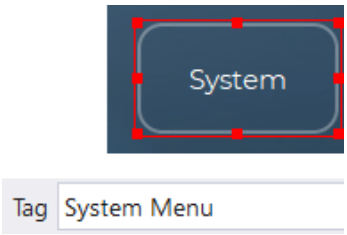
1. Add a K7s controller to a new project file after adding a processor and room.
2. Add a Sony Blu-ray All Models infrared device to a project file.
3. Select the Sony Blu-Ray under the source tab in the workspace.

4. Select the “Return” button on the interface. Note the tag name is “Back.”
5. Erase the tag in the window and change it to “Power.”

   Notice how the power command is auto-populated in the macro window.
6. Change the tag to “Back” and notice the return command populate.

This exercise illustrates how tags can be changed without disrupting the relationship between the command and the tag. The “Back” tag was erased and substituted with another tag name. Once the original tag name was reentered, the command was populated again.

1. Select the “System” button with the tag “System Menu.”



2. There is no command on the button because the standard tag name, “System Menu,” did not find a match in the Sony Blu-ray IR device library.

3. Repurpose this button for another useful function by erasing the tag in the tag window and entering “Menu Home.”

4. The Menu Home command will auto-populate a macro for the command.

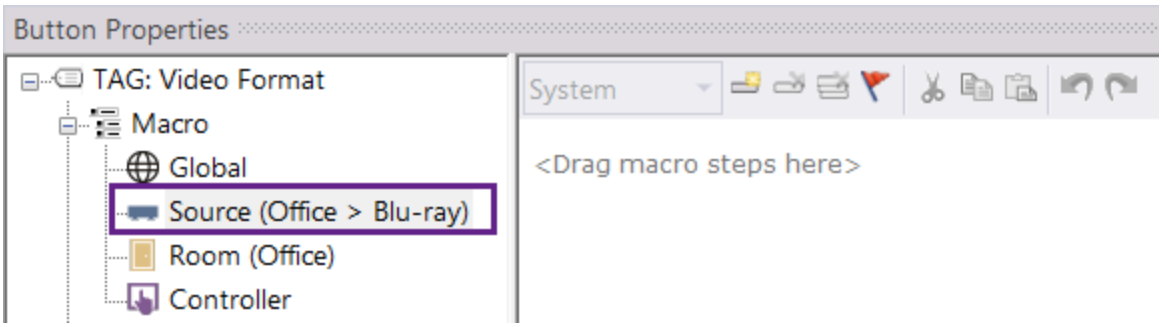
5. Change the text or use an icon to represent the new command.

Not every standard tag will auto-populate a command, and not every command has a standard tag. Make a tag name when a command is needed but a standard tag is unavailable.

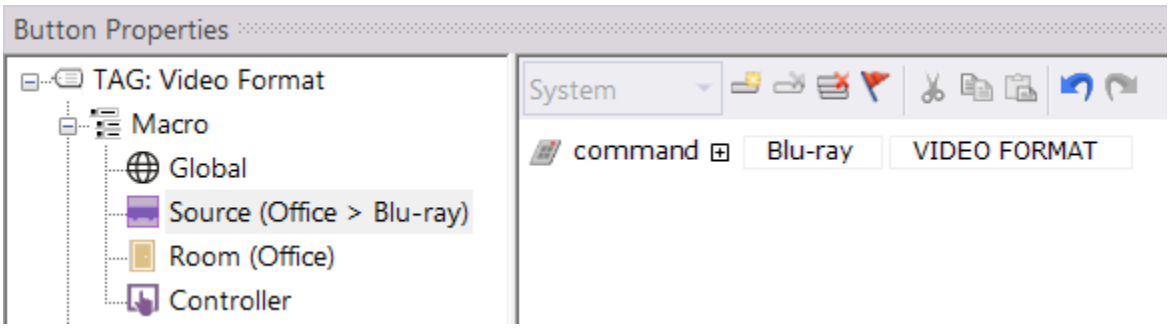
1. A command in the Sony Blu-ray – All Models code set is not populated in the template but exists in the infrared device library. The command is named “Video Format.”

2. Repurpose a button or bring in a button from the bitmap's library. Please select the button and give it a logical tag name. Since tag names should match the function as closely as possible, we will tag the button “Video Format.”

3. The button will not populate with a macro since the “Video Format” tag is not standard. It is necessary to add a command to connect to the tag. Select **Source** in the button properties window under the **Macro** section.



4. Drag a command to the macro window and change it to the Video Format command in the Sony Blu-ray device library.



5. The Video Format tag is now linked to the “Video Format” command in the device library. A source-level macro was programmed to ensure the functionality extends on a user interface for the Blu-ray in the controlled room on all controllers. If the command changes, it will change across multiple controllers controlling the Sony Blu-ray device.

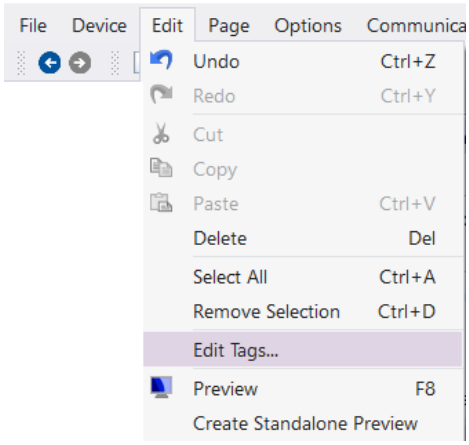
6. Make cosmetic changes to the button by editing text or using an icon.

   This exercise illustrates how non-standard tags can create a relationship between the tag and the macro command(s). While this is useful for programming, the benefit is realized when changing commands on multiple controllers.

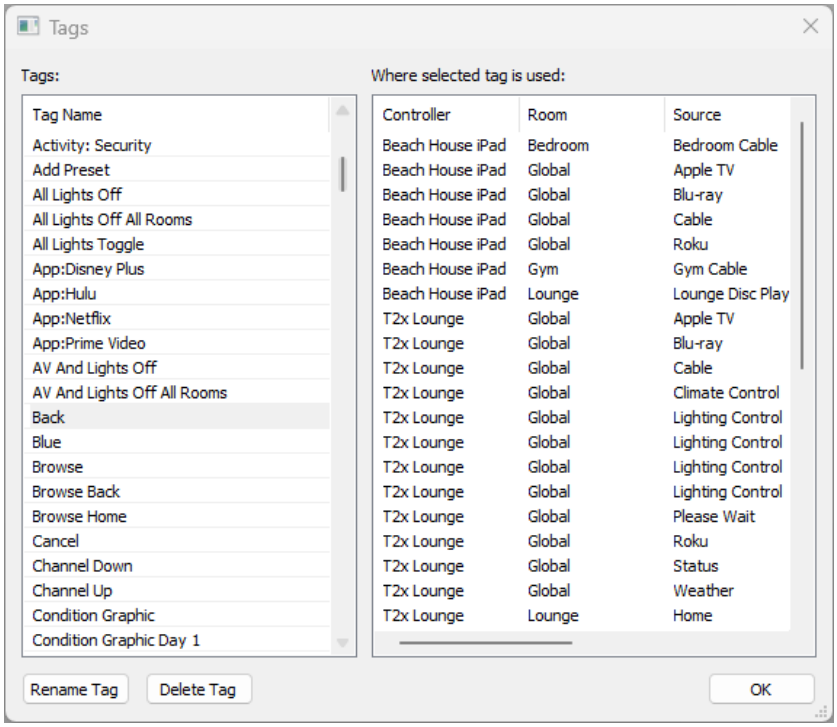
Edit Tags Function

The **Edit Tags** function allows the RTI installer to view every tag in the project file and see the room, controller, and source device where it resides. A tag can be **renamed** and **deleted** exclusively using the Edit Tags option.

1. Select **Edit Tags** in the **Edit** tab in the main toolbar of Integration Designer.

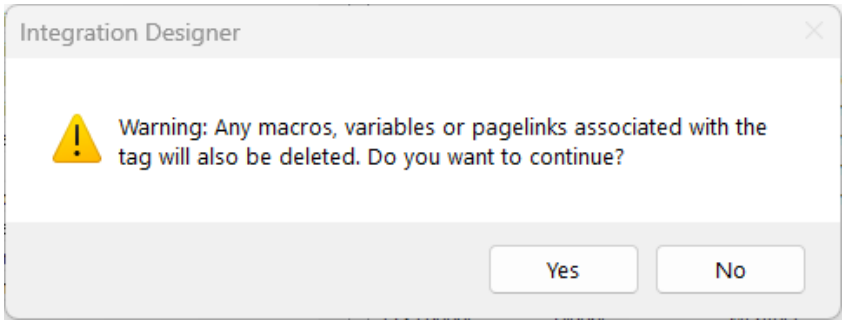


2. In the tags window, select a tag on the left and view where the selected tag is used in the project file. A controller, room, and source will be displayed if the tag exists on the user interface.



3. Tags can be renamed or permanently deleted using the options at the bottom of the menu.

4. A warning will appear when a tag is deleted, letting you know any macros, variables, and page links will also be deleted.



Press **Yes** to confirm or **No** to cancel.

Pro Tip: Always name tags logically to make finding them easier. Tags can only be **renamed** or **deleted** in the **Edit Tags** window.

Programming Button Menus

Integration Designer saves time and adds efficiency features to make macro programming less repetitive. Auto-programming, maintenance, and the System Manager allow for a quicker and more intelligent experience within the software.

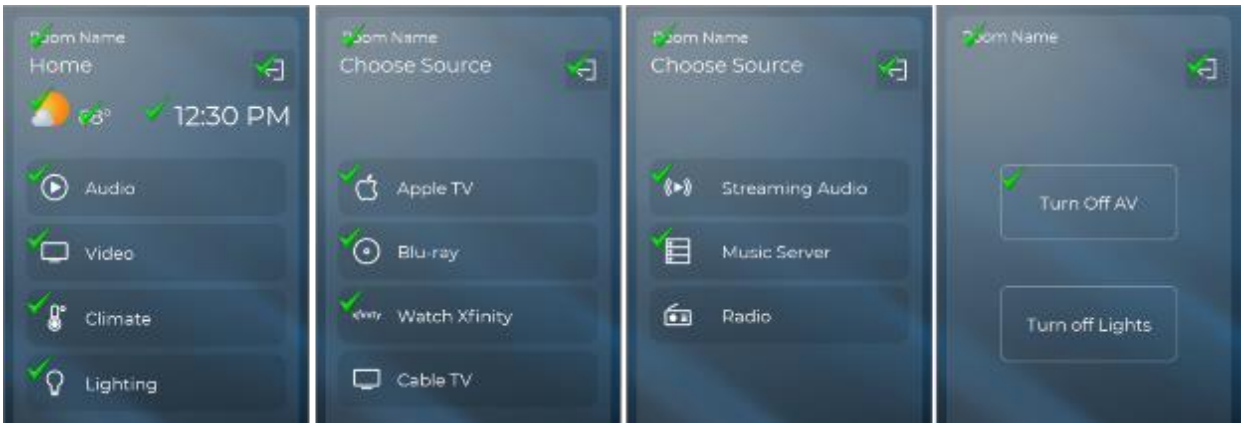
When choosing a **list menu** option when adding a controller to the workspace, it will auto-generate a text-based list of rooms and activities. These menus contain built-in tag functionality and tell the **System Manager** what macros to run. Choosing the Button Menu option requires the RTI installer to properly build the menu of rooms and sources for a more customized approach.

Adding a Button Menu Controller

1. Add a handheld **controller** to a **local** room in the **Add Workspace Item** window. Select any pages for source devices already added or add source devices next.

2. Select the **Home** source device in the room where the controller was added.

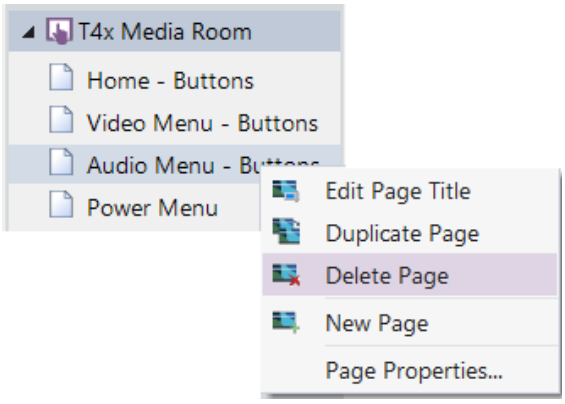
3. The software creates a multi-page home source device. The RTI installer can decide how to present the home page to the end user.



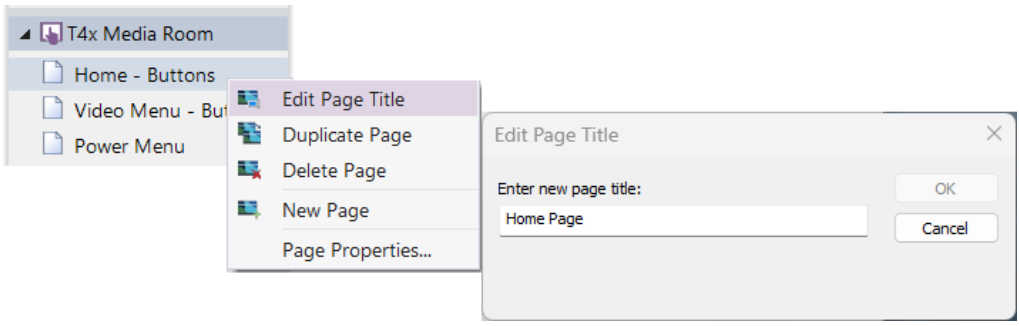
4. The pages designed for this controller are as follows:
    a. **Home Buttons** – Used to select Audio, Video, or Other device types.
    b. **Video Menu** – Used to select video devices after the video button has been selected on the home buttons menu.
    c. **Audio Menu** – Select audio devices on the home buttons menu after selecting the audio button.
    d. **Power Menu** – Used to turn off the room and lighting, if applicable.

    The RTI installer may decide to keep the functionality as is, or they may opt to present a menu of activities in a mixed fashion. Doing so will preclude the end user from selecting a category first.

5. Delete any unnecessary pages by right-clicking the page and then selecting Delete Page.



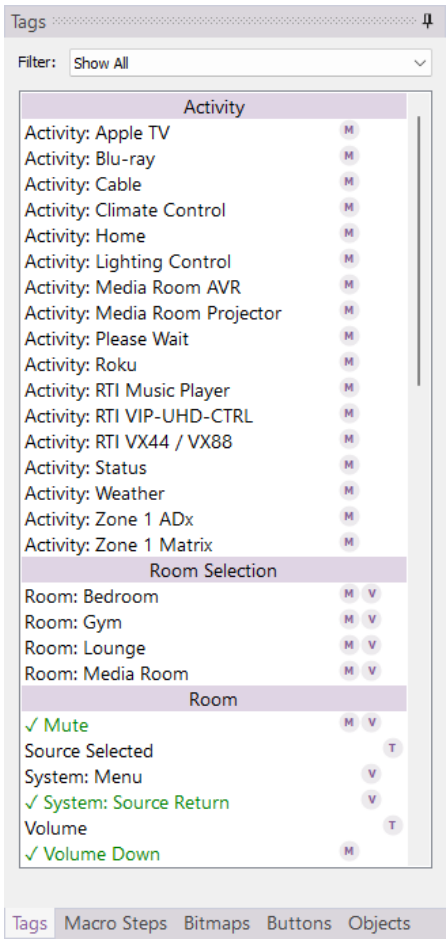
6. Rename **home** source pages by right-clicking and selecting **Edit Page Title**.



Do not delete the **power page;** a page jump has already been programmed on the controller's red power hard button.

7. Once the home source pages have been configured, the button menu on the controller may be configured.

Copying and pasting for additional buttons, resizing, adding pages and frames, and aligning buttons using the alignment tool is possible.



8. Drag and drop the icons needed for the activities to the buttons using an icon bitmap library. Commonly used functions should appear at the top.

9. Change the text on the buttons by right-clicking each button and selecting the Edit Text option in the context menu.

Pro Tip: Use names for our activities as “Watch Cable” or “Watch Roku.” Avoid using model numbers or brands when possible. Ensure naming conventions are used that the end user and family members can identify.

10. Once the button menu is created, they require specialized commands to ensure the macro will run in the System Manager. Fortunately, when devices are added to the software, specialized activity tags are created with built-in functionality that tells the System Manager to run the activity in the selected room.

Select the Tag tab in the tabbed menu to see a list of tags created in the software. Many of these tags have built-in functionality.

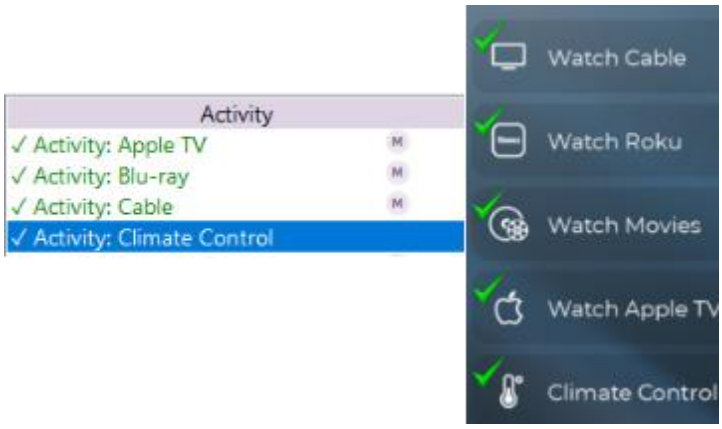
Activity tags have been created for every device added to the workspace. While some devices are not intended to be activities, they are created if an activity is required.

Special symbols to the left of the tag indicate specialized functionality built into the tag.

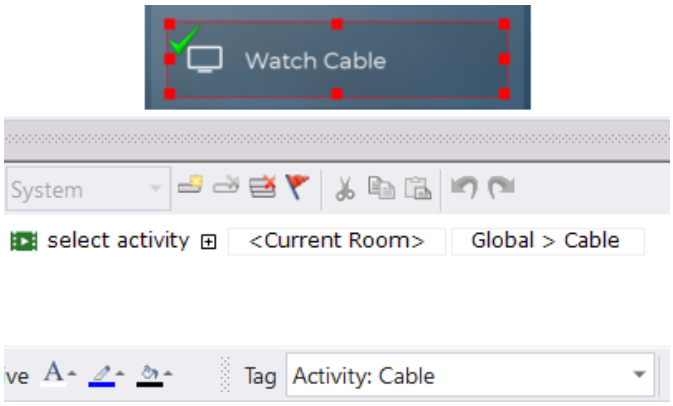
M = A macro is built into the tag
V = A variable is built into the tag
T = A text tag is built into the tag

Some activities may have a combination of **macros**, **variables**, and **text tags**.

11. Drag each Activity Tag to the appropriate button on the user interface. Once a tag has been applied to a button, it will turn green in the tag window, indicating that it has been added. Activity buttons on the interface that have been tagged will have a checkmark, indicating that a tag and macro have been programmed.

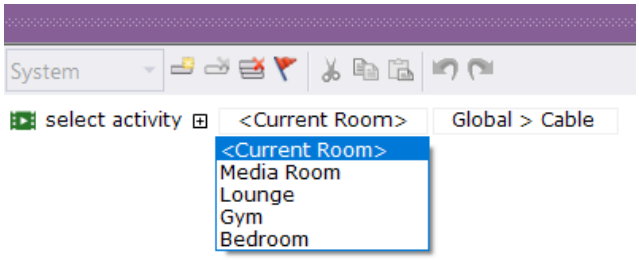


12. Selecting each button will reveal the tag name in the tag window and display the macro.



The activity tag will program a macro to tell the System Manager what room activity to run. Since the source is in the global area, it is intended to be shared. Integration Designer software tracks the room selection and will run the activity for the current room.

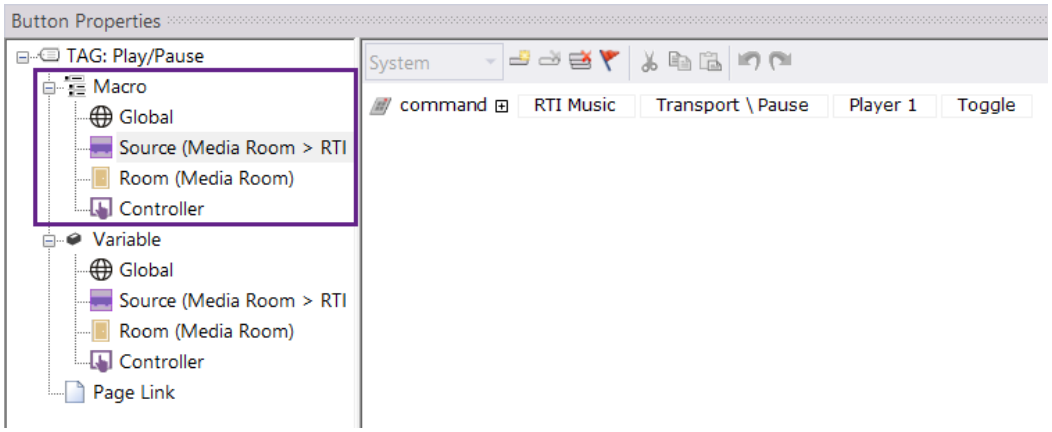
If programming assigned button menus for each room, a hard-coded room may be selected in the <Current Room> drop-down menu.



Introduction to Macro Levels

Once a button or feedback is tagged, a macro or variable level can be assigned. Macro and Variable levels allow the tag to maintain functionality in different interface areas. Macros and Variables can be designated using four levels: global, source, room, & controller. The RTI installer can determine which to use depending on the function. In this lesson, we will explain macro levels.

Macro levels can be assigned using the button properties once a button is tagged.

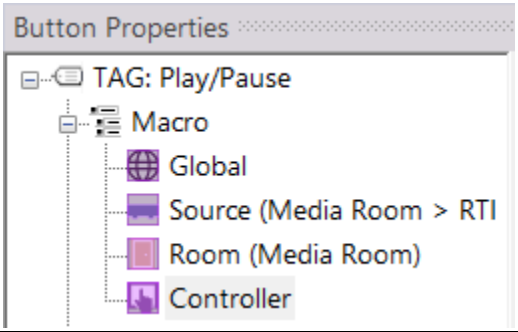


In addition to the four **macro** and **variable** levels, **page links** can also be programmed using the **Button Properties** window. A purple box will appear on a macro or variable that has been programmed. They are also indicated in the tag toolbar in Integration Designer.

These are the definitions of the **four different macro levels** in the order of **priority**.

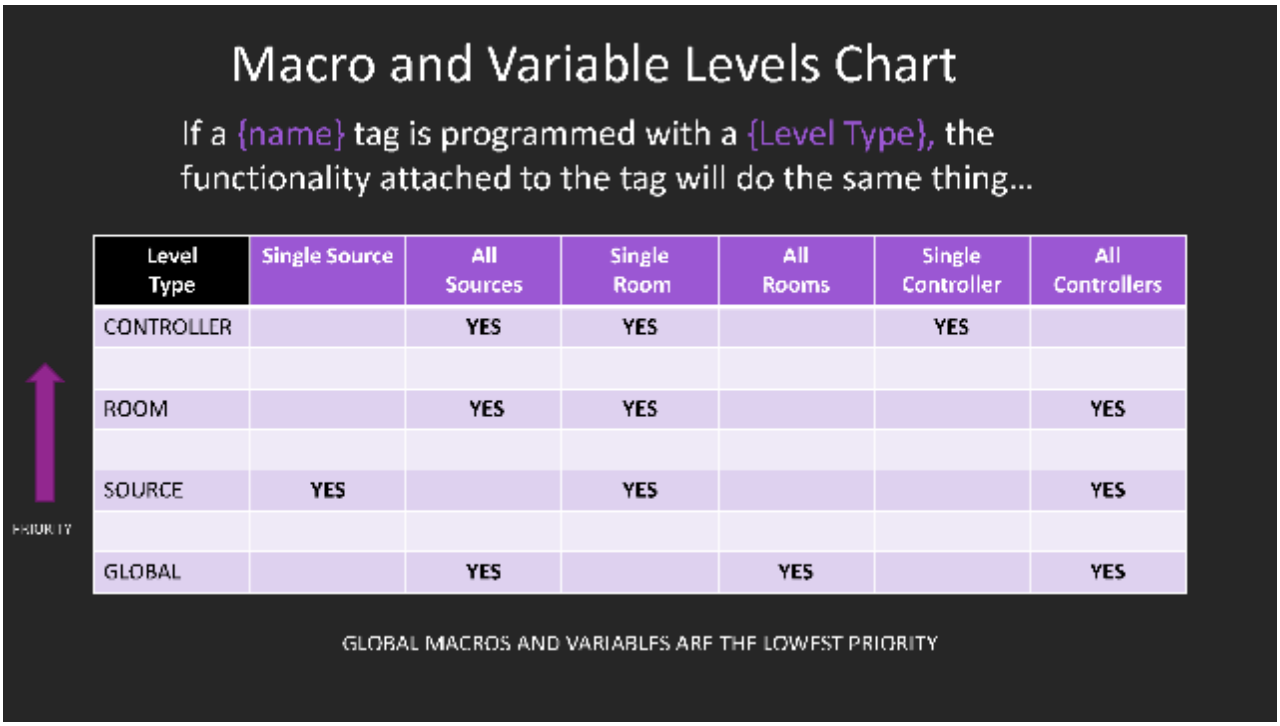
Macro	Global	The tag function will be the same in all rooms, all sources, and all controllers.
Lowest Priority		**Example: "Lock Front Door" or "Arm Security"**
Macro	Source	The tag function will be the same in the room it was created, on the source where it was created, and on all controllers. **Example: Play, Menu, and source-related commands.**
Macro	Room	The tag function will be the same in the room it was created on all sources and controllers. **Example: AVR Volume Up, Down & Mute**
Macro	Controller	The tag function will be the same in the room where it was created, on all sources, and on the controller where it was created.
Highest Priority		**Example: Favorite DSP**

Multiple macro types may be programmed on a button. The **priority level** determines which macro will run. A **global** macro is the lowest priority, followed by the **source**, **room**, and highest-priority **controller**-level macro.



In this example, all four macro levels have been programmed. The controller macro will run, but the rest of the commands are maintained.

Here is a chart that demonstrates how macro levels work:



Level Type	Single Source	All Sources	Single Room	All Rooms	Single Controller	All Controllers
CONTROLLER		YES	YES		YES	
ROOM		YES	YES			YES
SOURCE	YES		YES			YES
GLOBAL		YES		YES		YES

GLOBAL MACROS

A global macro is used for general commands not reliant on a room, source, or specific controller. For example, system-wide commands are usually candidates for a global macro level since the tag's functionality reaches the furthest in the interface. For example, there is only one front door in the home, so a tag labeled "Lock Front Door" should do the same thing no matter what room, source, or controller sends the command. However, this macro level has the lowest priority, so any other macro level created will supersede the global macro.

Exercise

1. Create a tag on a button called "Example."

2. Program any command on the button using a global macro level.

3. Create a button on any other controller in another room and source page.

4. Tag the “Example” button and notice how the command initially programmed is auto-populated.

5. Add or change the command on the button with the tag “Example.”

6. Return to the original location where you created the button and notice how the change carried over to the button.

   This example illustrates how functionality attached to a tag with a global macro can be used to program and revise programming throughout the project file.

SOURCE MACROS

A source-level macro is used for commands you would find on a source device. When devices are auto-programmed and added to the workspace, source macros are created and, thus, the most common type. A tag with a source macro will have the same functionality on that source, in the room where the source exists, on all controllers that control that source device. Source macros are useful for adding or revising commands on a source device, especially when multiple controllers exist. Changing a “Menu” command on a controller controlling a cable box in a family room will reflect that change on all controllers and avoid programming repetitively.

Exercise

1. Create a room labeled “**Family Room**” and add several different touchscreen controllers to the Family Room.

2. Add an infrared source device, “**Xfinity X1 Cable Box**,” to the Family Room.

3. Select the “Xfinity X1 Cable Box” source in the workspace. You should see all the source pages for the various controllers in the Family Room.

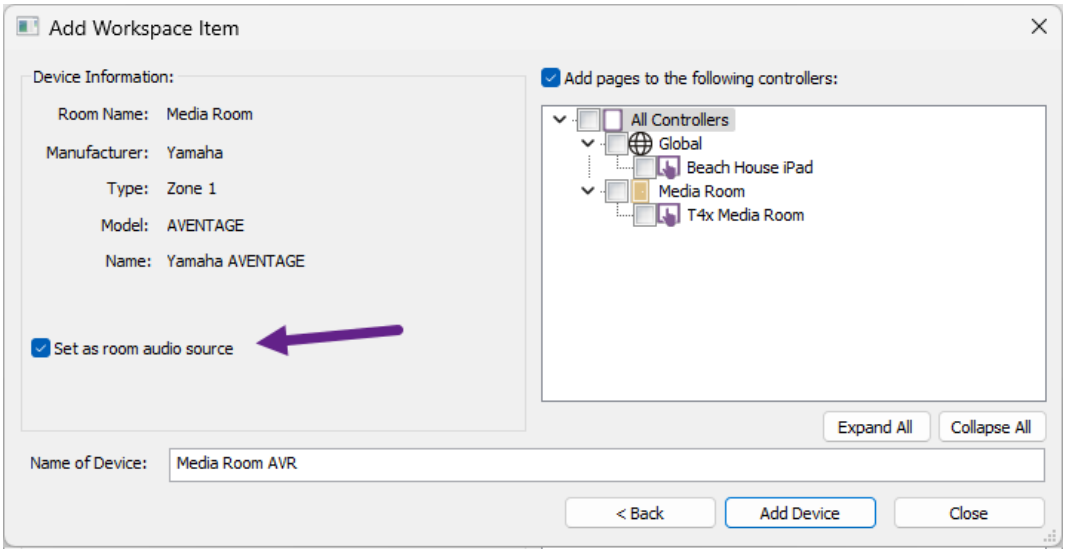
4. Select the “**Menu**” button and add a command (or two) to the source macro.

5. Select the same source page for another controller and note how the menu command reflects the same changes made on the other device.

This example illustrates how a source macro can be used for maintenance or programming buttons without the need to do it multiple times on different controllers.

ROOM MACROS

Room-level macros are used when a tag functionality needs to be the same on all sources and controllers in that room. Using a room-level macro will ensure that no matter where that tag appears in that room, any source or controller will have the same functionality. Room-level macros are assigned to the volume up, down, and mute commands when a source device is set as a room audio source.



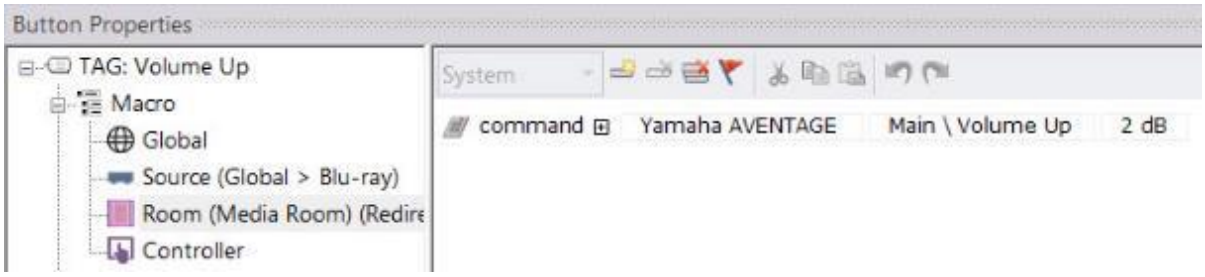
Setting a device as a room's audio source will populate room macros for the volume and mute commands, ensuring all the sources and all the controllers will use that device for volume control. Other examples of room macros include scenes, a sound program, or anything that is room-dependent but not source-dependent.

Exercise

1. Add a **Yamaha Aventage receiver** driver to the workspace for an existing project. Ensure the **set as room audio source** is selected.

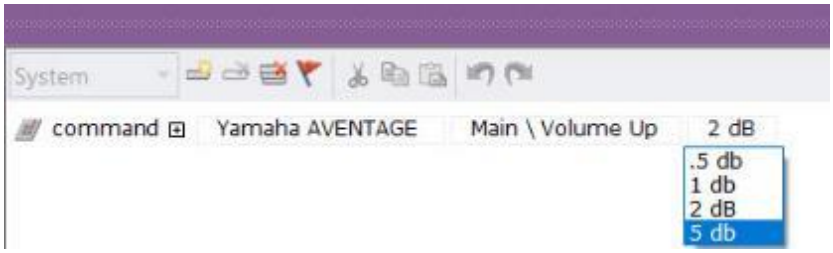
2. Add additional controllers to the room where the receiver driver was added.

3. Select a **controller** in the room where the receiver was placed. Select a **Volume Up** on the controller or screen user interface on any source device.



When the set as room audio source option was checked, this created room macros on all the sources in the room. The higher-priority **room macro** will supersede any volume commands programmed on a source level for the devices in the room. Each source device will run the room macro when the volume and mute keys are pressed.

4. Adjust the decibel increments in the drop-down menu in the macro window to -5db.



5. Next, please select any other **controller** and **source** device page in the room and notice how the **Volume Up** tag functionality will be the same on each source and controller in the room it was created.

   This exercise demonstrates how a **room macro** will allow RTI installers to create a macro based on the tag name that will do the same thing on **all sources** and **controllers** in the room. In this example, the button tagged **Volume Up** will do the same thing in one room, and other rooms will have their room macros.

   Examples of room macros are commands used in a room on all sources and controllers—volume, room scenes, favorite DSP settings, and anything that is room-dependent and not source-dependent.

CONTROLLER MACROS

In some situations, a tagged function needs to be controller-dependent. A controller-level macro is used when a tagged function needs to be the same in a room, on all sources, but only on a single controller. It is also the only macro level that is used for standalone commands. Standalone commands are macros that run on the remote instead of the processor. A device must be set to **Standalone** when added to the workspace to use standalone commands.

Exercise

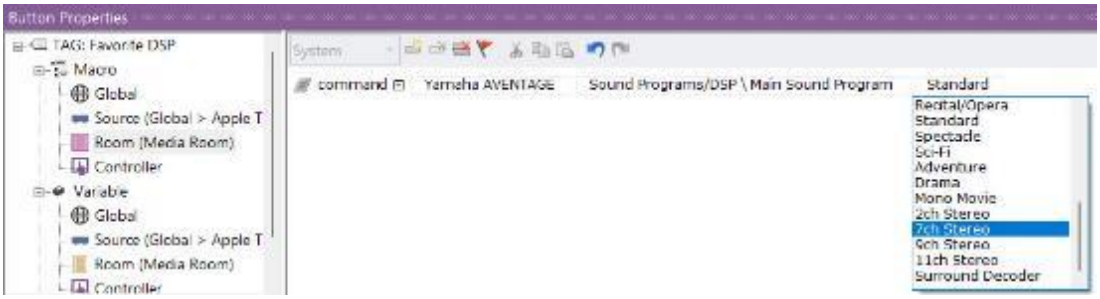
Scenario: The end-user has several controllers in his media room. We created a tag labeled "Favorite DSP" and programmed a receiver sound program on a room level so all the controllers could use it on any source page in the Media Room. The client is fine with all the controllers having the same favorite but wants his favorite to be different.

1. Create a project file with several **controllers** and **sources** in a Media Room.

2. Add a Yamaha Aventage Receiver driver to the Media Room.

3. Create a **button** on any **source** page for any **controller** and tag it "**Favorite DSP**."

4. On the button, program a **sound program** using a **room-level** macro from the Yamaha driver.



5. Next, navigate to a **different controller** in the Media Room and select any source page. Create a button on the source page and tag it "**Favorite DSP**."

The sound program previously programmed should auto-populate.

6. Select the **Controller Macro Level** in the **Button Properties** and program a different sound program on the button.

   The controller will now have two macro functions, one on a room level and the other on a controller level. When the button is pressed, the controller macro will run and ignore the room macro because it has a higher priority.

7. Navigate to another **source** page on the **same controller** and tag a button “Favorite DSP." The controller level macro level should auto-populate.

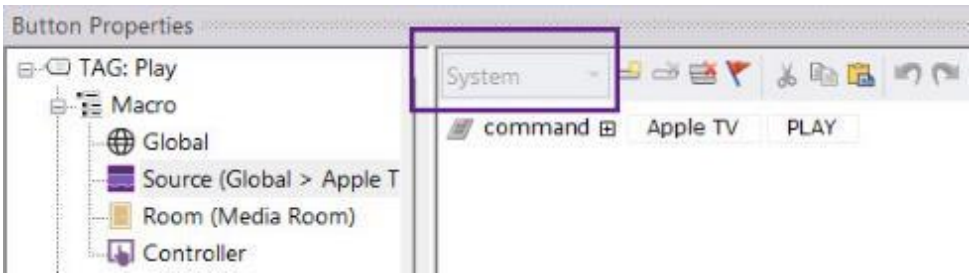
8. Now navigate to a different controller in the same room and tag a button “**Favorite DSP**." The existing room macro should auto-populate, and not the controller-level macro, since it is exclusive to the controller it was created.

   This exercise demonstrates how controller-level macros ensure a tag will do the same thing for all the sources in the room, but only on the controller it was created. In this way, the rest of the controllers maintain their room-level macros, and only the controller where the controller-level macro was created reflects the function.

   Use controller-level macros when you want a tagged button on a single controller to have a different function than other controllers in the room. Controller-level macros include commands specific to an individual controller and do not impact other controllers, even though the tag is the same.

Programming Controller Standalone Macros

Global, **Source**, and **Room** Level macros run in **System** Mode. System Mode indicates the command is run on the control processor. When creating these macro types, the option is greyed out and may not be changed.

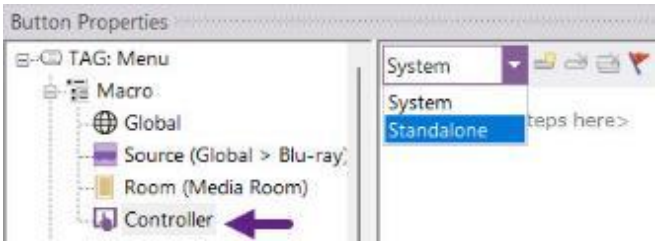


When a macro needs to be sent directly from the controller and bypasses the processor, it must be set from **System** to **Standalone**. Controller-level macros are the only macro level that can change from System Mode to Standalone.

1. Select a button on any controller. **Please note** not all controllers can send standalone commands. RTiPanel devices, KA-series panels, and other RTI control devices incapable of infrared transmission may not send commands in **Standalone mode**.

2. Select the controller level macro in the button properties.

   The communication mode can be changed from System to Standalone by selecting the option in the drop-down menu.



3. Once in **standalone mode**, only source device commands that were created in standalone in the output port settings will be accessible.

The Media Room TV was set to standalone when adding the device to the workspace and is available in standalone mode.

S**tandalone** and **system mode** macros in the same macro must be sent in certain situations. In this scenario, the controller sends standalone

commands directly to a source device, and the rest of the commands are run on the processor.

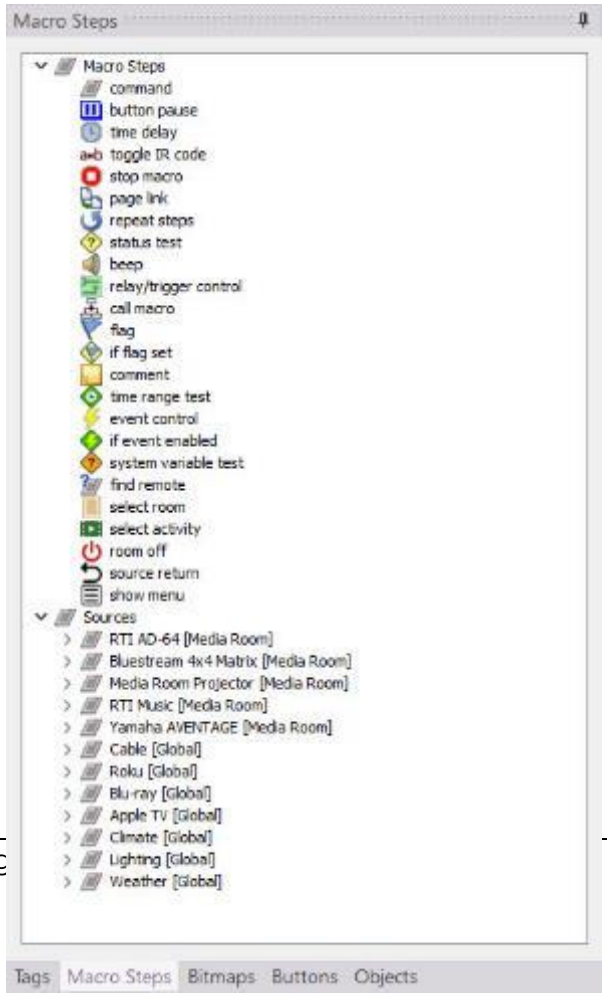
Pro Tip: It is best practice to avoid standalone commands. While reliability is compromised, the System Manager cannot be utilized since it can only track commands sent via the processor. Use standalone devices as a last resort.

Programming Macros

MACRO STEPS EDITOR

There are several ways to program macros in Integration Designer.

- The System Manager
- Buttons on the user interface
- Driver Events
- Global, Source, and Room Level System Macros



System Mode Macro Steps

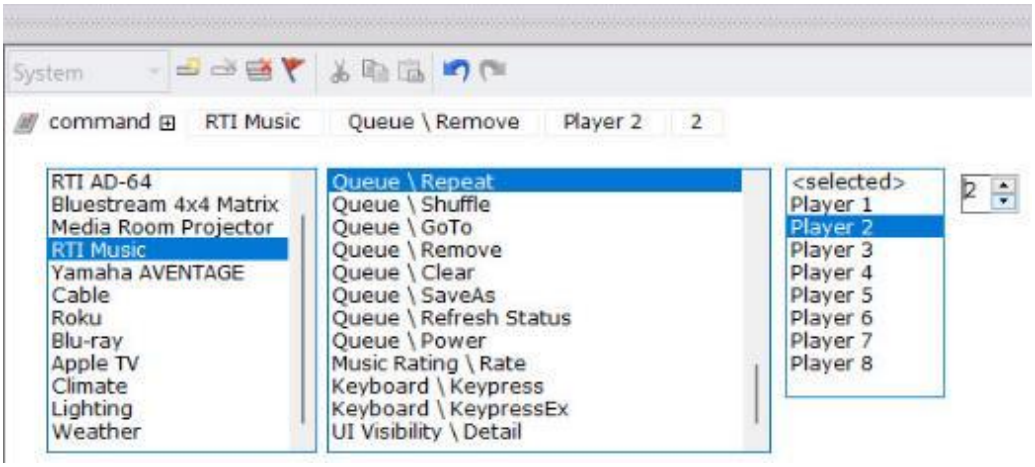
The **Macro Steps Editor** in the tabbed menu provides several methods and tools for programming macros. RTI Installers can address scenarios where a solution is needed for stubborn components, client requests, and special functionality. Please note that different macro steps are exclusive to **System** and. **Standalone** modes.

Commands for available devices may be found under **sources**. Local room and global sources may be accessed when programming a **local** controller. When

programming a **global** controller, all source devices will be available.

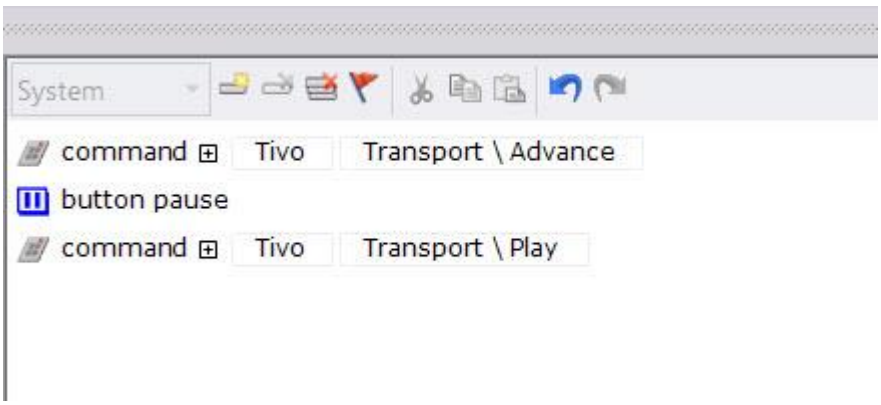
command

The **Command** macro step can be placed in the macro window, and any device, command, and additional arguments can be selected in the drop-down menu.



button pause

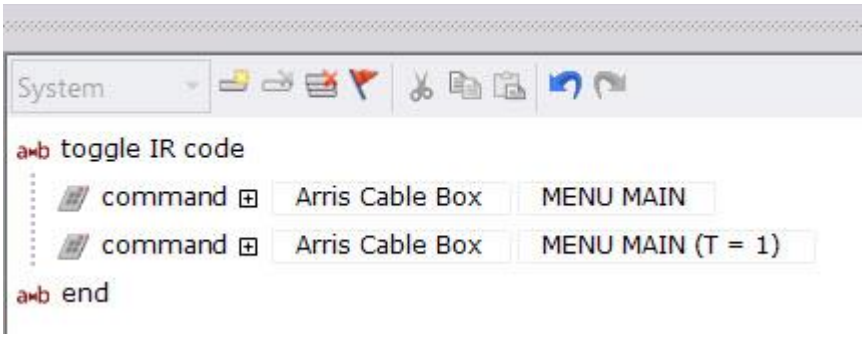
The **Button Pause** step causes the macro to stop executing until the button it is attached to is released, at which time the rest of the macro is executed. Drag the Button Test icon from the lower right to the macro display on the left.



time delay

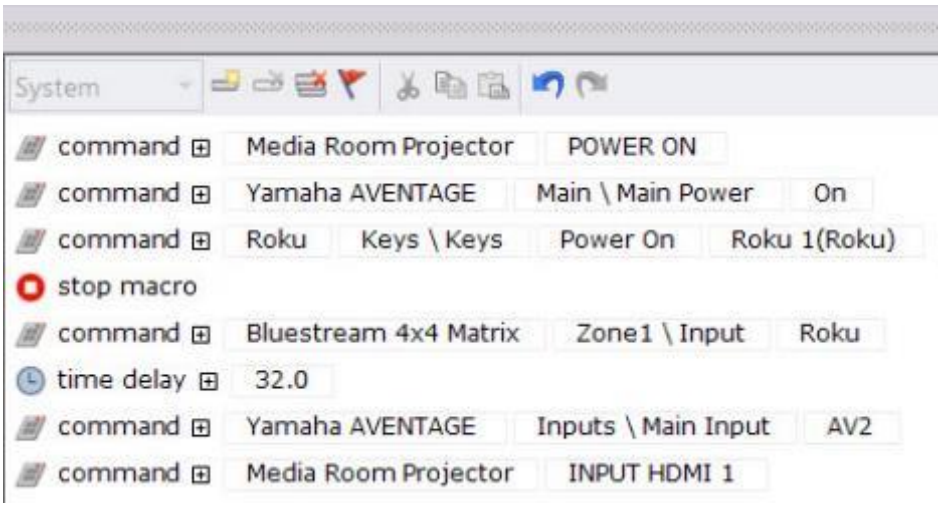
toggle IR code

Toggle IR codes are used on controllers with two full sets of infrared codes from every button on the remote. A macro is required for each control function to control devices with toggle-bit IR codes. Drag the main and the T=1 command between the (toggle bit) brackets within the macro window using the toggle bit macro step. Note this feature can be used to toggle between any two regular commands.



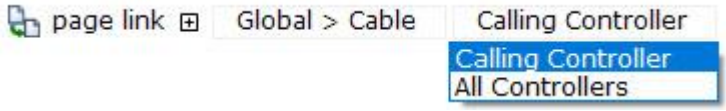
stop macro

The **Stop Macro** step causes the macro to stop executing at the point where it is inserted. This step can be used for troubleshooting or as part of an if/else statement.



page link

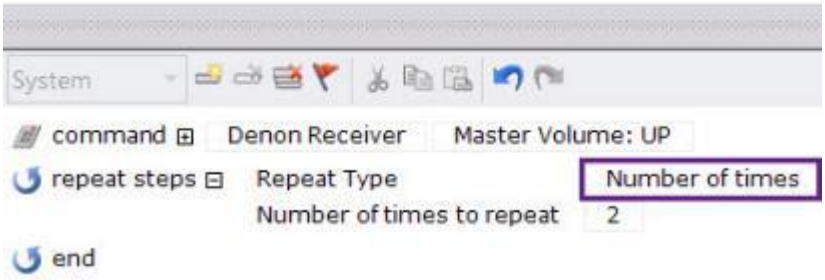
The **Page Link** step causes the device to change to the first page on any source device. The link will default to the **Calling Controller** but can be changed to **All Controllers**. The button must be set to standalone mode to jump to a specific frame. To jump to a specific **page** on a source, use the page link command in the **button properties**.

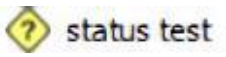


repeat steps

The **Repeat Steps** step causes the macro to output a group of steps multiple times. Set the options to continuously repeat the following macro steps for as long as the button is held down or to have the steps the indicated number of times. You can enter any repeat count from 1 to 60.

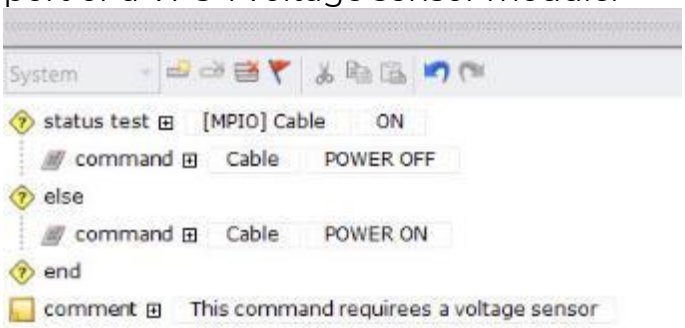
This step is used most often to sustain an RS232 command.





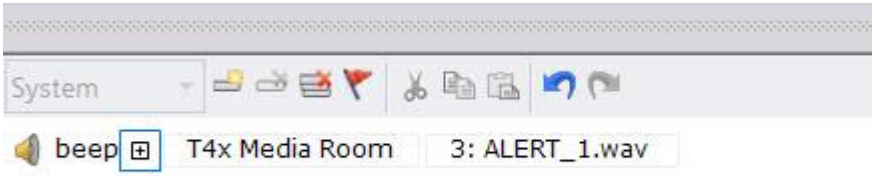
The **Status Test** step causes the macro to execute or not execute a group of steps based on the status of an input or output line. Select the sensory input or output line you would like to monitor from this list, then select the condition you would like to check for on the selected input/output line.

Note: This option requires an RTI control processor with an MPIO port or a VPS-1 voltage sensor module.



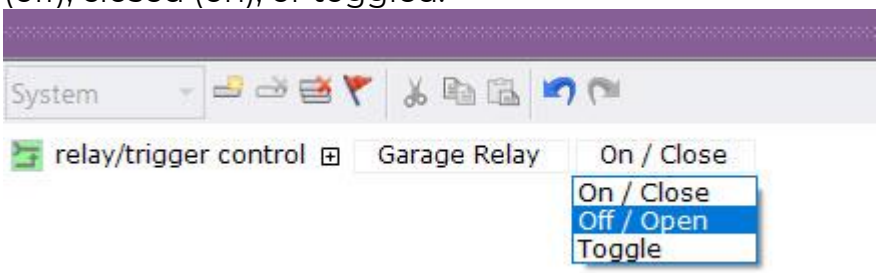
beep

The **Beep** step will play a system default or custom sound added in the controller properties. The sound will play even if the Enable Beeper setting is turned off in the controller properties.



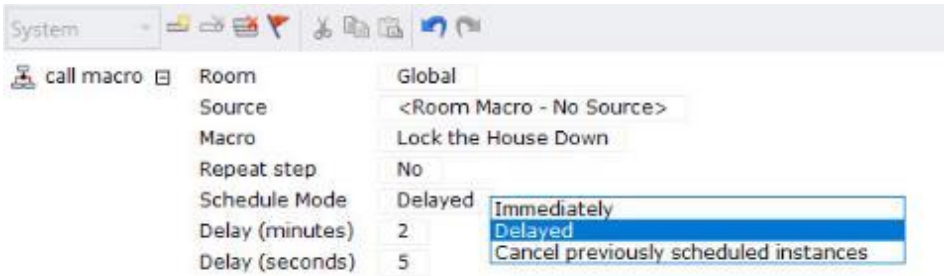
relay/trigger control

The **Relay Control** step changes the current state of one of the device's relay outputs. Select the action you want for the relay output, then select the relay port to control. Relays may be open (off), closed (on), or toggled.



call macro

The **Call Macro** step can call global, source, and room-level macros from buttons and events. Selecting the + sign in the command will display options. Repeat steps can be set from no to yes, and the macro can be delayed. Once the delayed option is set, enter the delay time in minutes and seconds. A delayed call macro can also be canceled if it has not expired.



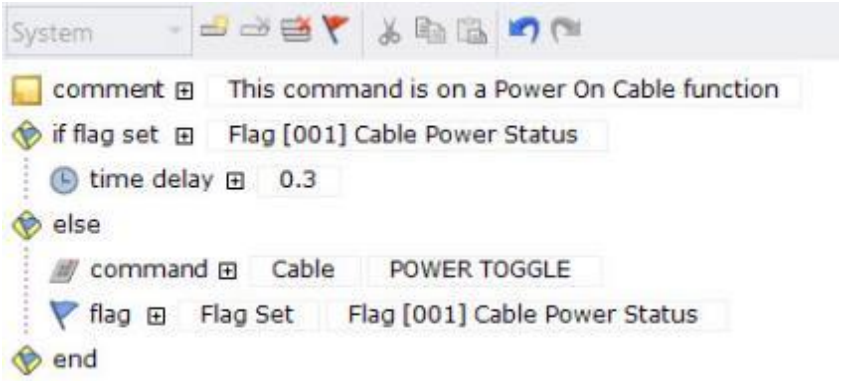
flag

The **Flag** step keeps track of a component or a remote device's status and acts based on that status. Multiple flags are available, which you set or clear under macro control. You can use the "Test Flag" option to take different actions based on whether a certain flag is set.

For example, you can use this functionality to make discrete power ON and OFF macros for components for which you have only toggle codes. Use the "**Set Flag**" option when you turn on the component, the "**Clear Flag**" when you turn it off, and the "**If Flag Set**" option to determine if a power command should be sent based on the flag status. **Note**: 256 system flags may be accessed on the processor, and 256 are available on a controller. Controller flags may be accessed in standalone mode.

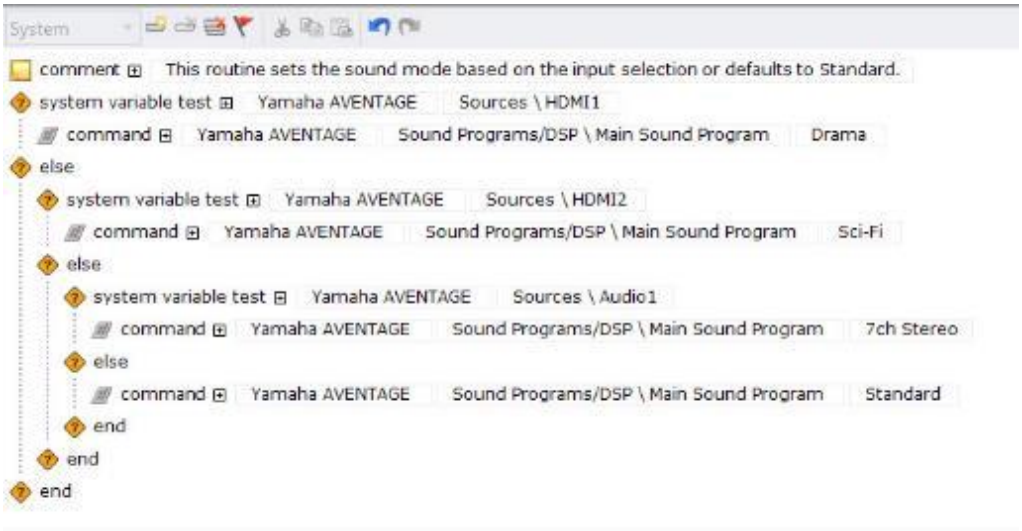
if flag set

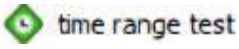
Once a flag is set or cleared in the programming, an If Flag Set step can be used for a conditional if/else statement, pressing a button could send multiple commands, page links, and other functionality based on the state of the flag.



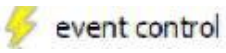
comment

A comment may be used for complex macro programming to understand the programmer's intent or to label different sections based on functionality.

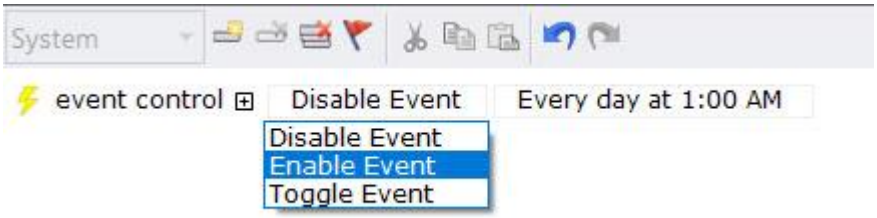


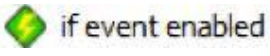


The **Time Range** Test will run a macro between a certain time, sunrise or sunset, or day of the week. Each may be mixed into the command. If the time range is not matched, the macro can be made to run other commands.

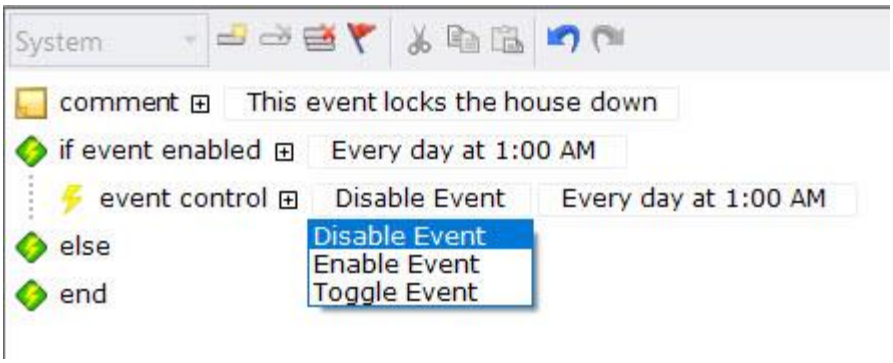


Use the **Event Control** step to turn existing processor events on or off processor events. These events include startup, periodic, time-based, and sense events.



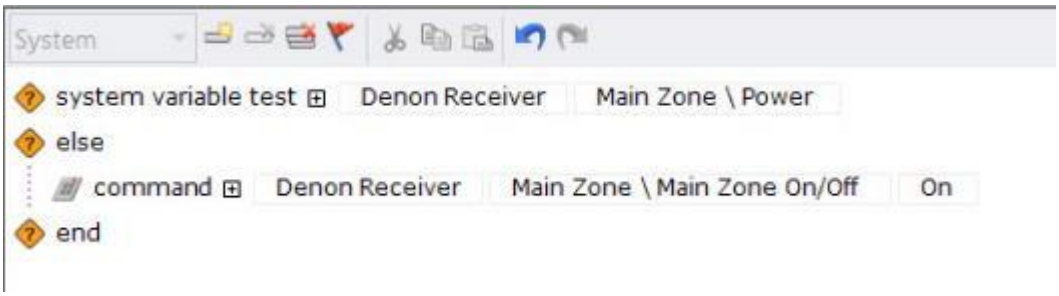


To test if Event Control is enabled or disabled, use the **If Event Enabled** step. An if/else macro routine can be programmed based on the event status.



system variable test

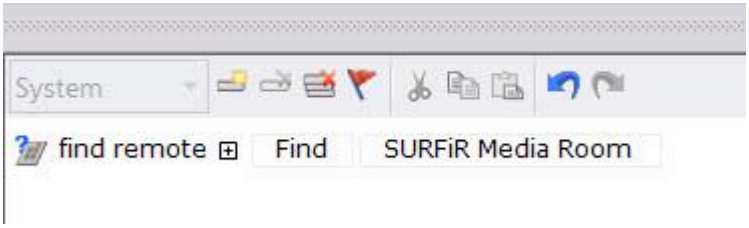
When testing variable statuses, use the **System Variable Test**. An if/else macro routine may be programmed based on feedback from a device. Boolean variables can be either true or false.



The receiver power status is evaluated. If Power = True (On), then do nothing. Else, Power = False (off) turn on the receiver.

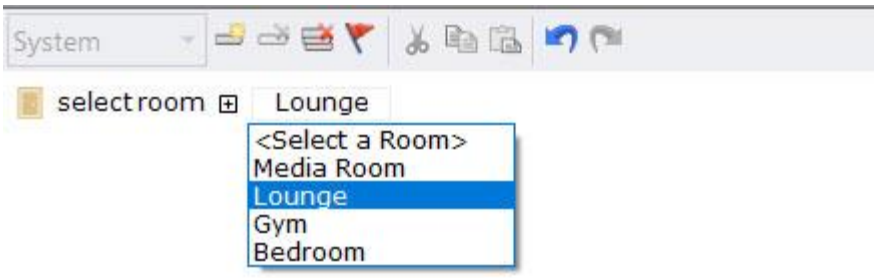
find remote

Find Remote will trigger a beeping sound on any SURFiR companion controller selected in the drop-down menu.



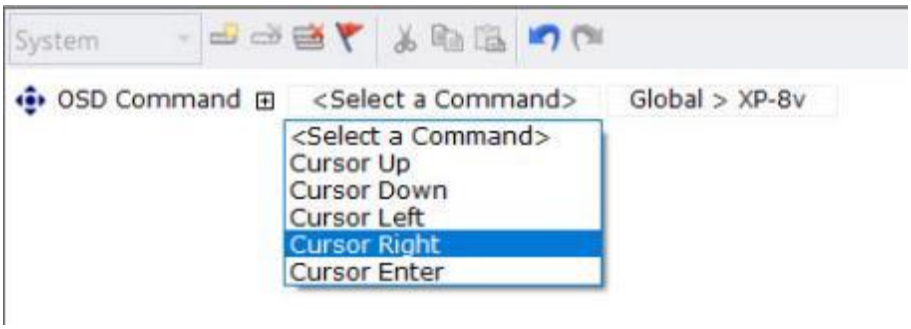
select room

Select Room is a command that tells the System Manager the selected room. Once a room is selected, room-level macros will point to the macro for that room on global controllers. Other functionality can be triggered, such as layers, menus, and other options based on the room selection.



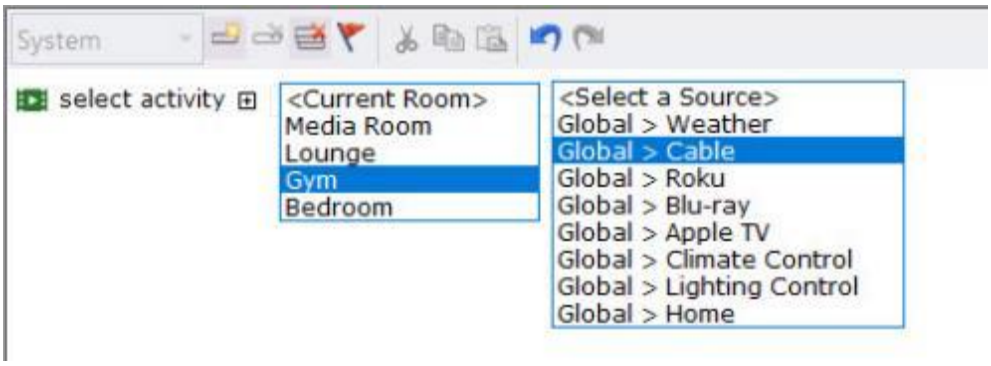
OSD Command

OSD or on-screen-display commands are used when using the HDMI output on the XP-8v to create a user interface on a video display. These commands allow navigation and button selection.



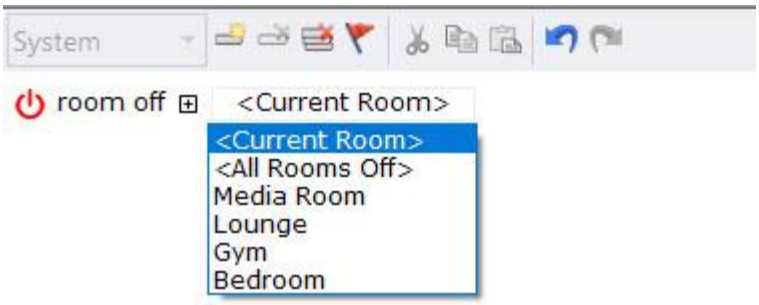
select activity

Select Activity is a command built into an activity tag and placed on buttons to tell the System Manger which room and activity to run. These are used for button menu controllers.



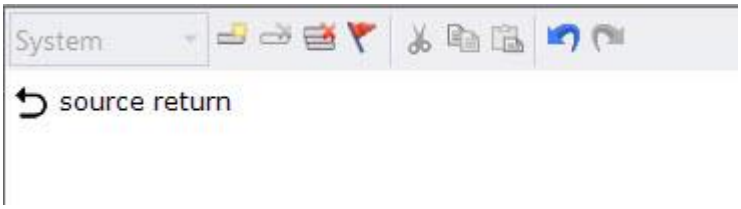
room off

The **Room Off** command tells the **System Manager** to run deselection and power off commands for the currently selected room, all rooms, or a specific room.



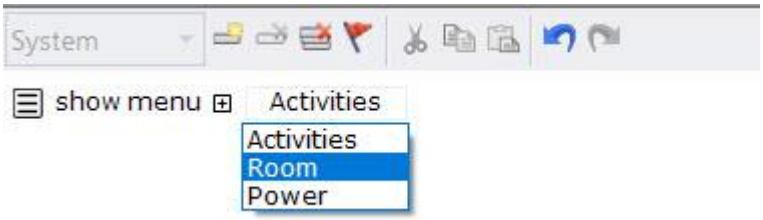
source return

The **Source Return** command is used on a button to navigate to the last selected video or audio source. It is usually on a "Back" button visible on sources set to "Other" in the system manager.



show menu

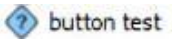
The **Show Menu** command will display any layers with **room**, **activities**, and **power list** visible state variables. It can display one menu at a time while suppressing others.



Sources

A list of available devices and commands for the current controller selected is available in the **Sources** section in Macro Steps. Commands can be dragged into the macro window when programming. Most drivers are categorized, and IR library commands are alphabetized.

Standalone Mode Macro Steps



The **Button Test** step causes the macro to execute a group of steps ONLY IF the button that the macro is attached to is pressed (or not pressed, or double-clicked, depending on the option selected). After selecting options, drag the Button Test icon from the lower right to the macro display on the left.

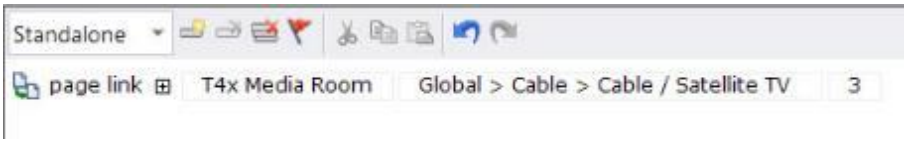
The **Button Test** steps are conditional statements or “Else” commands, meaning that if the condition is met, the first command will be sent, and if the condition is not met, the second command will be sent.

In the top example, if the button is held down for two seconds, it will send a power-on command and switch the TV's input.

In the bottom example, if the button is double-clicked within two seconds, it will send a power-on command.

page link

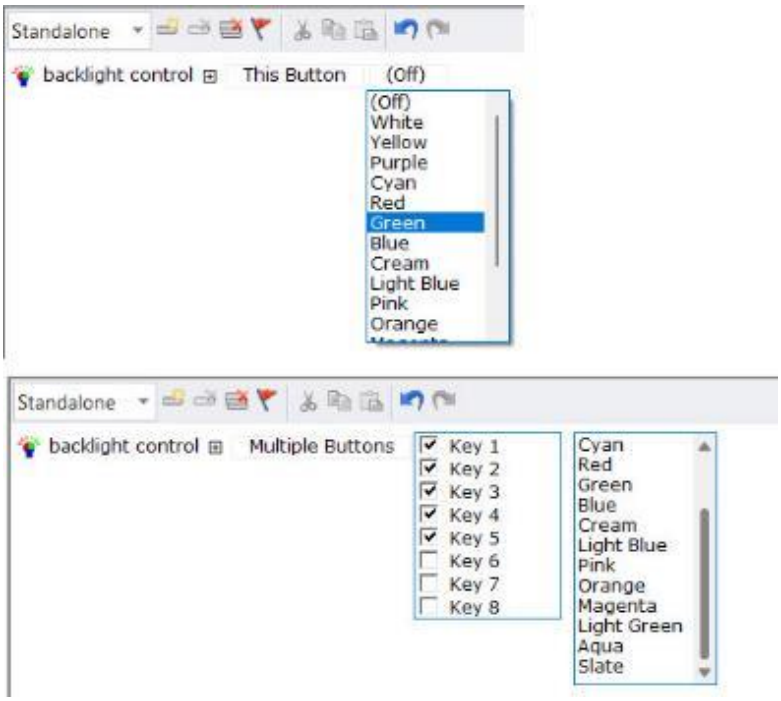
A **page link** can jump to a source page and frame on the controller in **standalone** mode. In addition, the page link can jump to a previous page or an RTI control panel. The control panel provides a preference screen for a controller.



The page link above will jump to the third **frame** on the initial page of the cable source device.

backlight control

The backlight control option is available on the RK1+ two, four, and eight-button keypads. An option exists to change a button to a certain color in the drop-down menu, or multiple buttons can also reflect a backlight color.

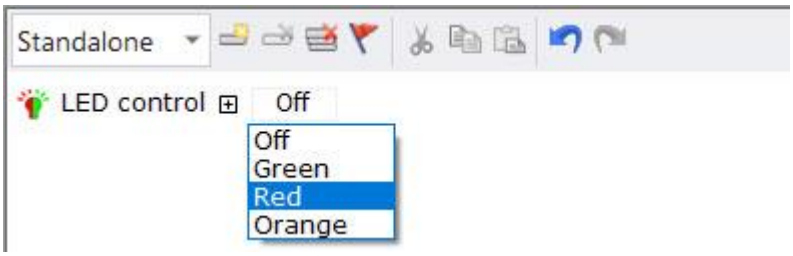


flag

In standalone mode, 256 flags can be accessed on a controller. Since they are exclusive to the controller, the processor and other controllers are unaware of the flag status.

LED control

LED control is used on the legacy KX3 in-wall keypad and control processor to turn on and change a room status LED on the front of the unit to green, red, orange, or off.



Programming Variables for Feedback

Integration Designer software offers simplistic and powerful feedback programming options. The software offers feedback options with an XP processor and a compatible two-way device. Feedback may be programmed in text, cover art, sliders, two-way scrolling lists, and more. Specialized graphics objects are available in the bitmap library, making programming easier for RTI installers.

Integration Designer features the following **feedback** types:

Button State: A feedback type that displays **text** and forces **inactive**, **reversed,** and **visible** states on a button if a Boolean variable is true. For example, the text would include an input indicator, while the power button would be forced to remain reversed if the AVR is powered on. Buttons and layers can be visible or hidden based on a variable status.

 Slider: A slider is a graphical object providing a measure and the capability to adjust a designated level indicator up or down. They are commonly used for volume, lighting, bass, and treble.

 Gauge: A gauge is a graphical object that provides a unit of measure but may not be adjusted and used for reporting volume, bass, treble, and lighting levels.

Toggle: A toggle is a graphical object with two different on and off states based on feedback, used for power and lighting.

Item Lists: Item lists are a scrollable list of feedback items that can be browsed and selected. They are commonly used for music lists, channels, activities, and room selection lists.

Dynamic Images: An object that can capture image-based feedback. Common in music album cover art, channel selections, and others.

Image Lists – An image list consists of several compiled images. Based on feedback, the applicable image will be displayed. They are used for reporting weather and frame indicators.

- **Spinners**: Spinners are a type of item list available when using an XP-8v OSD that requires cursor selection and control.

Button State Feedback

Button State Feedback can be programmed using **three** different methods:

1. Button properties window
2. Using Text Tags (button text feedback)
3. Inserting a control variable in the text editor (button text feedback)

A **tag** must be entered when programming feedback except when inserting a control variable in the text editor for text-based feedback.

There are four types of button-state feedback:

1. **Button Text** Feedback
2. **Reverse State** Buttons
3. **Inactive State** Buttons
4. **Visible State** Buttons

Button Text Feedback

INSERTING CONTROL VARIABLES

The **first** method of programming text-based feedback on a button is **inserting a control variable**.

Advantages: Easy and faster approach. This method does not require tagging a button—a good option if variable levels are unnecessary.

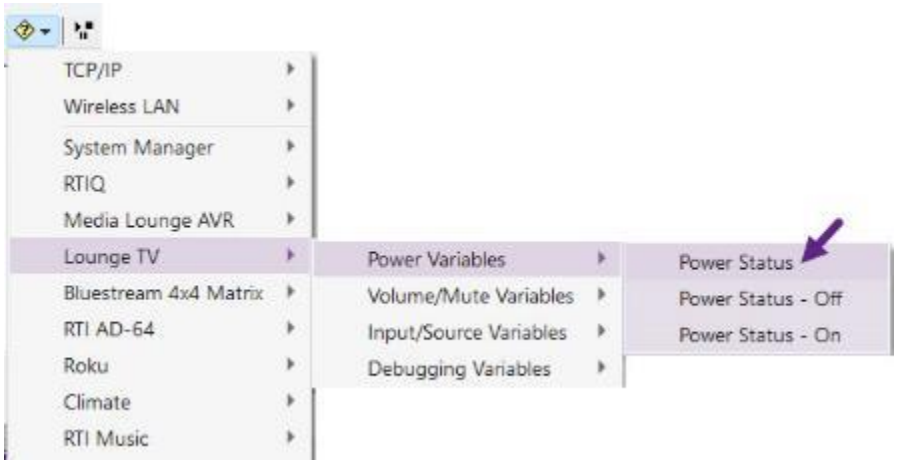
Disadvantages: Since tags are not used, variable levels may not be used. If text feedback is required in multiple locations, variables must be programmed repeatedly. If a revision is needed, it must be changed in multiple locations.

1. **Draw** a **button** on the user interface.

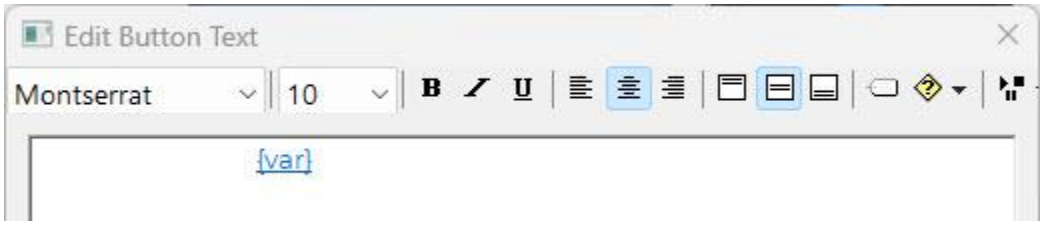
2. Right-click and select **Edit Text**.

3. Select the **insert control variable** button in the text editor toolbar.

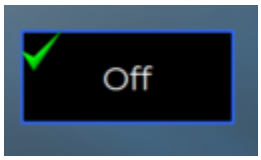
4. From the drop-down menu, select a driver, then a category, then a variable from the list of available text-based feedback. In this example, we will choose the Lounge TV "Power Status."



5. The status will be updated in the text box as a **{var}** for the variable. Exiting the text editor, in most cases, will show you a preview of the feedback from the device. Use the text editor for fine-tuning and adjustments.



6. Select OK to exit the text editor and view the programmed button.

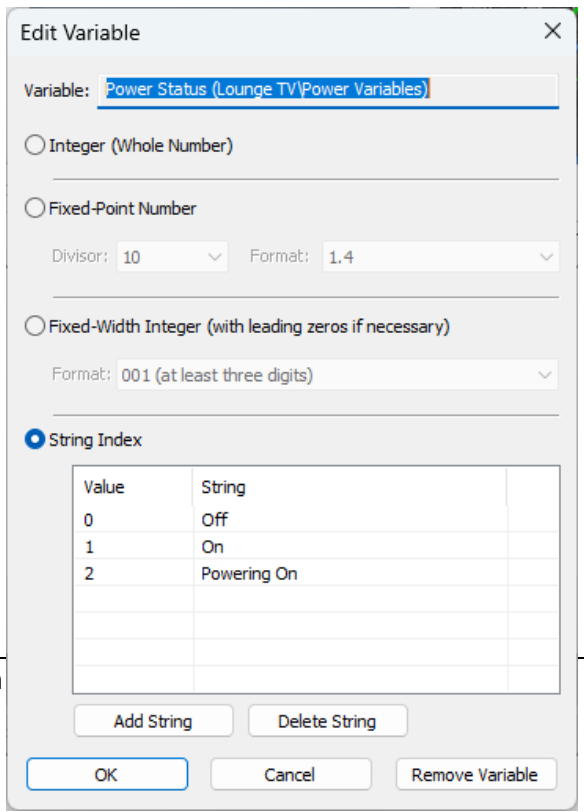


In this example, the power variable will display “Off” or “On,” depending on the state of the TV. The correct status will be returned even if someone turns on the TV manually.

CHANGING THE VARIABLE TEXT VALUE

In some cases, it is possible to change the feedback value of the variable programmed once a control variable is inserted in the text editor.

1. Select the button with the text feedback, right-click, and select **Edit Text**.



2. Select the {var} variable feedback to edit the variable. A window will display detailing different formats and values for the variable.

Variable: The variable will display the name, source, and driver category.

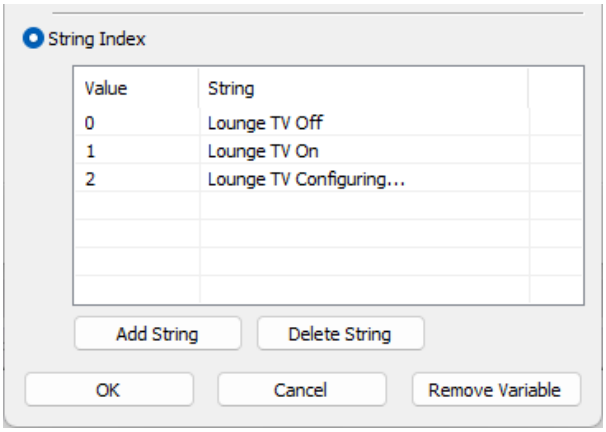
Integer: If the variable reports a decimal value, it can be converted to an integer.

Fixed-Point Number: The number can be converted to a format by selecting a divisor and format.

Fixed-Width Integer: This option will convert the integer to a certain format with leading zeros when necessary.

String Index: If the variable is an alpha value, the value on the right can be converted to another naming convention. For the example above, the driver reports a value of 1 and converts it to "On." These values may be changed in the index section, and additional strings can be added if necessary. This scenario may occur if the device manufacturer adds additional values before a driver revision.

3. Change the **string values** in the **string** index to a custom naming convention.



4. Select **OK** to apply the changes. The preview text on the button will reflect the new value if the variable changes.

INSERTING CONTROL VARIABLES

Text tags are the second method of programming text-based feedback on a button**.**

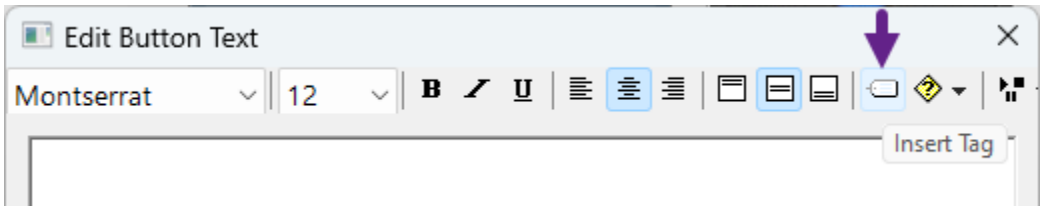
Advantages: Variable levels can be used based on the value of the **text tag**. This method can be much more efficient than inserting many control variables and programming the same value.

Disadvantages: Programming **text tags** requires more effort than **inserting a control variable**, but the initial investment in time can save time later.

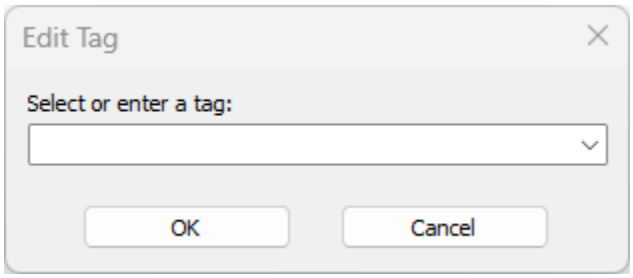
1. **Draw** a **button** on the user interface.

2. Right-click and select **Edit Text**.

3. Select the **Insert Tag** tool in the text editor toolbar.



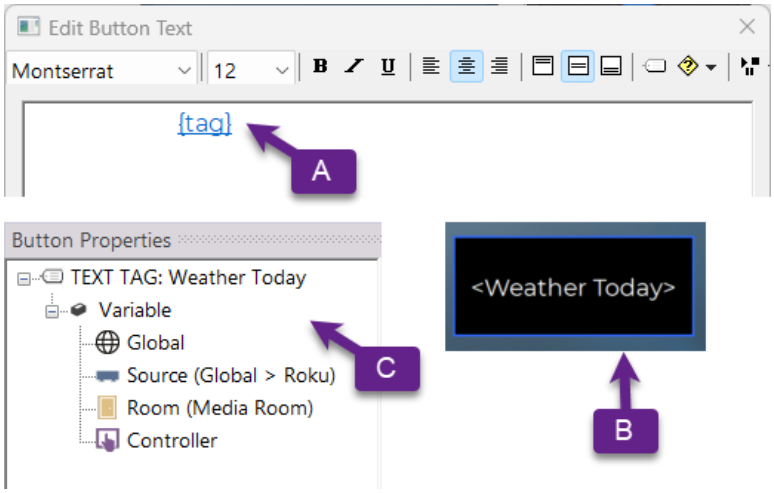
4. The **Edit Tag** window will display and prompt for a tag name. Enter a standard or personalized tag name.



Note: Creating a text tag with an applicable, standard tag name on a source device will auto-populate the variable in the button properties. If the variable does not populate, you will need to program the variable based on the level desired.

In this example, a non-standard tag name on a neutral source is entered.

5. Enter a “Weather Today” tag name and select **OK** to exit the Edit Tag window.



A. The Text Editor will have a placeholder for the tag in brackets.

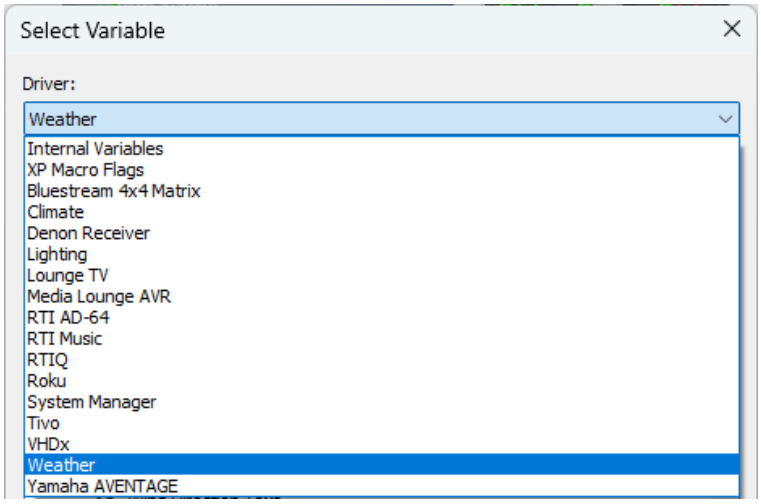
B. The **Button** will show a preview and can be adjusted if needed.

C. The **Button Properties** window reflects the tag type and a variable level may be assigned.

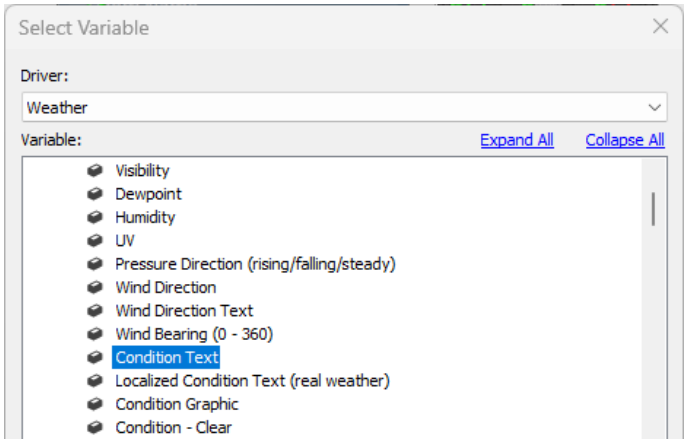
6. Select the **global variable** option in the button properties.

7. Select the ellipsis menu to the right of the text variable to access the drivers and other available variable types.

   Select a driver from the list of variables available in the project file.

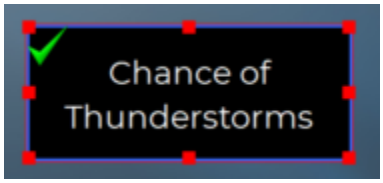


8. Once the driver is selected, select a **variable** to track the text state.

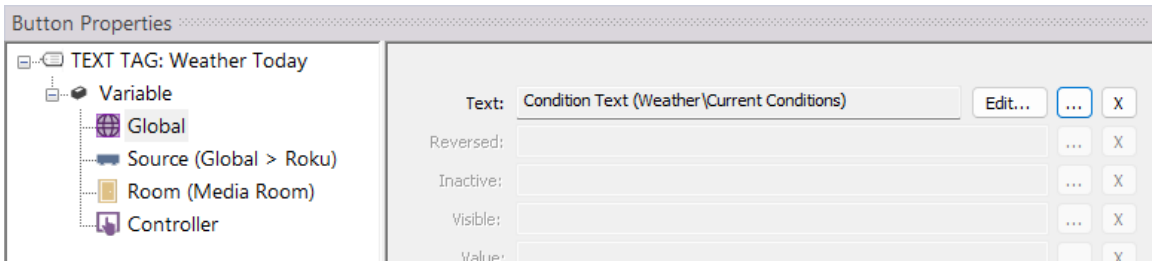


The **condition text** variable from the weather driver has been selected.

9. Select OK to apply your changes and revise the text format based on the preview text if necessary.



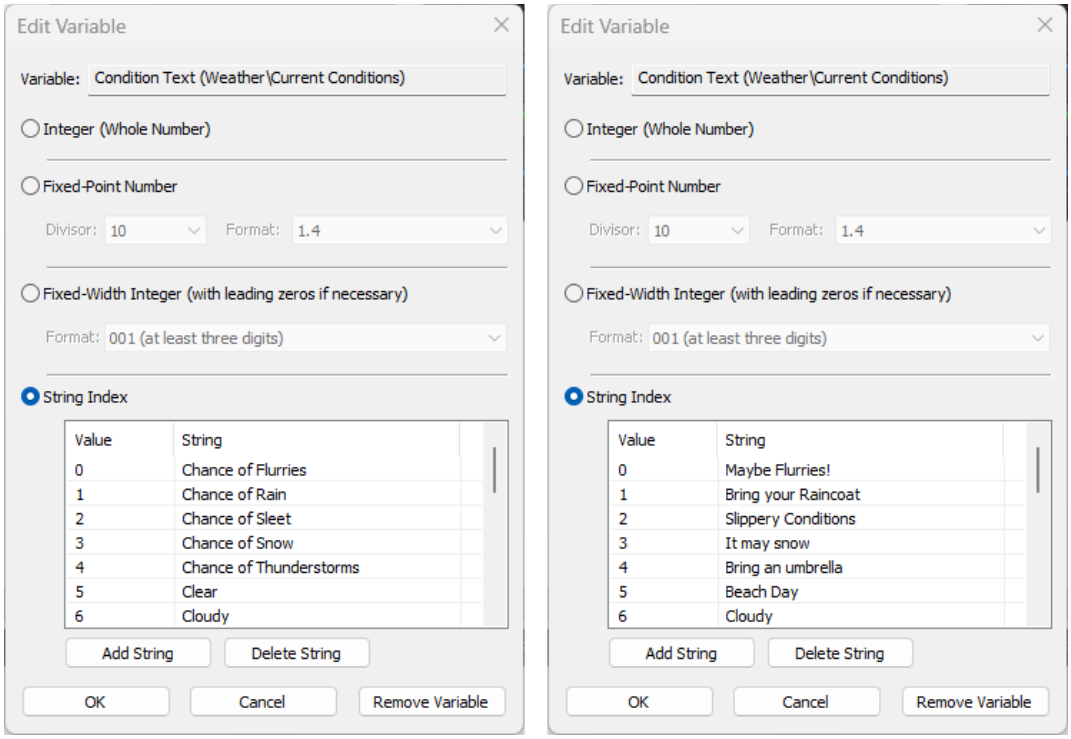
The **Button Properties** will reflect the **text tag name**, **variable level**, and the **variable value** in the variable text field.



Text may be remapped to personalized text when using text tags, like inserting control variables.

1. In the variable status window, select **Edit**.

   Since the weather condition text is in string form, you may enter personalized string values on the right side.

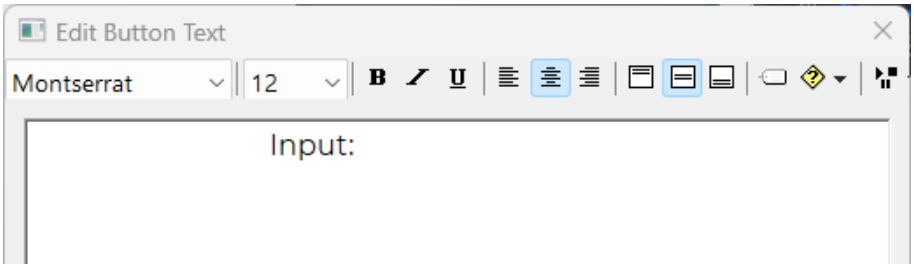


2. Select **OK** to apply any changes. The button preview will change if affected based on the new text entry.

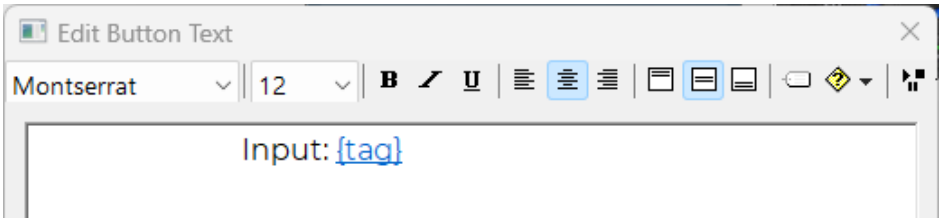
INSERTING MULTIPLE TEXT VARIABLES

Multiple text values may be inserted on the same button, whether **inserting a control variable** or using the **text tag** method. You may also use a combination of **regular text** and **variables**.

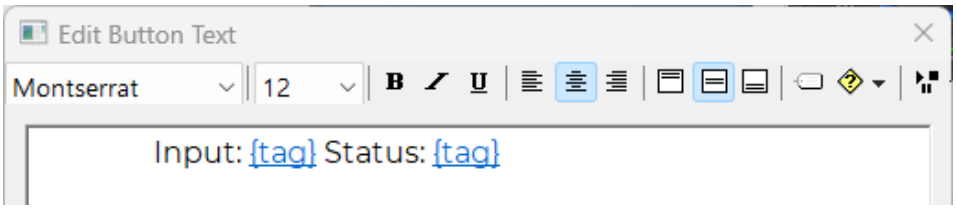
1. Make sure the button is wide so it may support multiple values.
2. Right-click and select the **Edit Text** option.
3. Before inserting a control variable or a text tag, place some **text** in the button to help identify the variable to the end user.



4. Insert a **text tag** or **control variable** for the source input on the device.

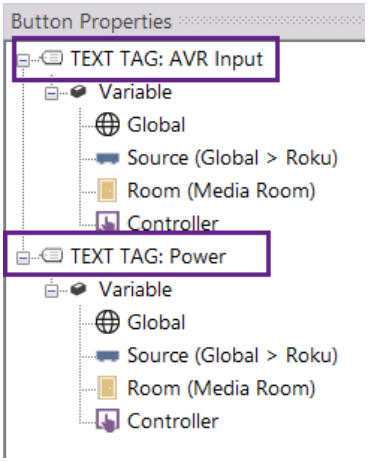


5. Insert additional **text** after the tag or control variable to indicate the next value.



Repeat this process with as many variables and text as the button size allows.

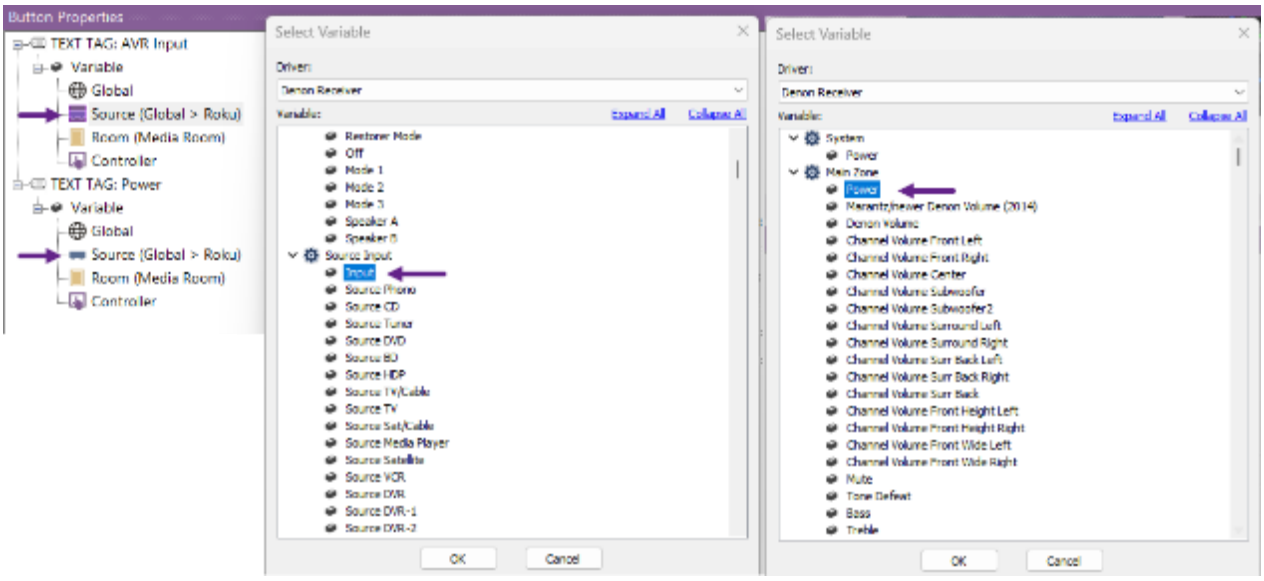
6. If using control variables, the process is complete. If text tags were used, a variable level must be set, and a variable added to each value.



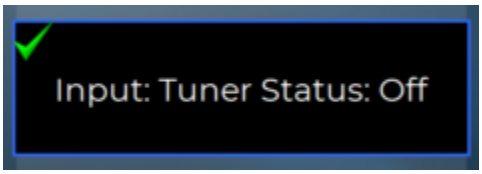
Be sure to enter the variable level for the correct text tag. Please note there could be additional values in the button properties if there is also a tag on the button.

A button can have multiple text tags and a tag for a macro and variable level. Each must be programmed separately.

7. Select a **variable level** based on preference for both text tags. Select variables for both text tags in the variable programming window.



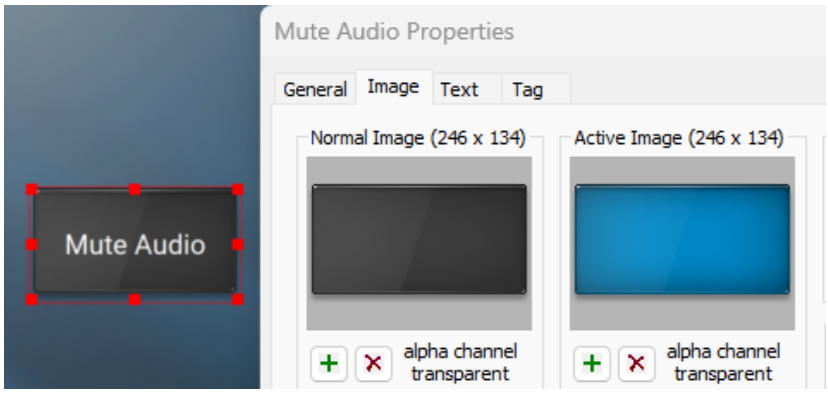
8. Adjust button text and properties as needed.



Reverse State Feedback

It is possible to force a button to display its reversed state based on a variable status. For example, if a device is on, a button can display a reversed state until the device is turned off. Reverse State feedback can make an interface more intuitive.

1. Check the button properties to ensure the normal and reverse states differ. An effective reverse state button will contrast with the normal state.



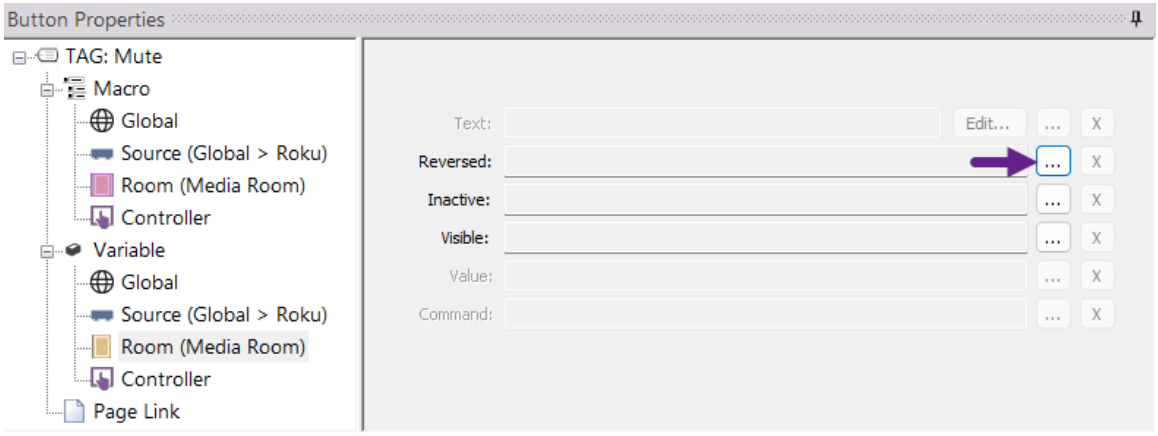
When the audio device in the room is muted, the button will go from a black to a blue tone and let the client know the system is muted.

2. Ensure the button is using the standard mute tag. (**In most cases, this type of tag will automatically program a room macro, but this is only an example**.)

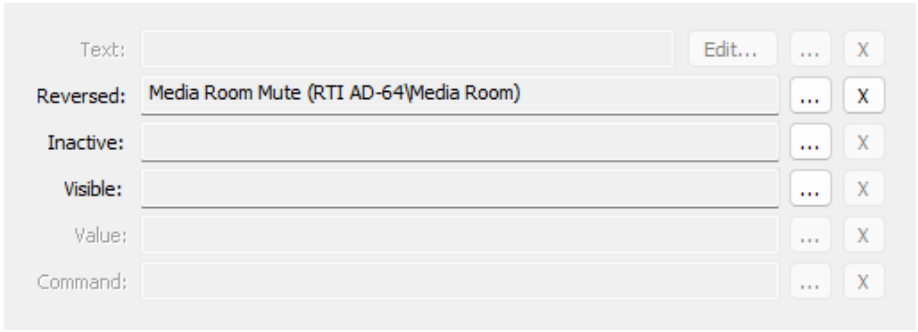
3. Program a **macro** on the button to mute the system. Typically, this would be a room macro, depending on the scenario.

4. Select the **variable level** in the button properties. While variable levels usually match the level of a macro on the same button, it is not a requirement.

5. Select the ellipsis menu to the right of the reversed variable entry.



6. Select a **variable** from the list of available drivers.



When the device reports a **mute state** to the processor, the button will invert to its reversed state. Even if the device is interacted with manually, the button will display the same status. The normal state will return when the mute status is false, indicating the device is no longer muted.

Selecting the reversed state option in the button properties will display the active state normally. When using reverse state variables, the normal state becomes the active state.

Pro Tip: Reversed state variables like the example above could avoid service calls. With an indicator the system is muted, the client understands why the audio is not playing even if the mute button is accidentally pressed.

Inactive State Buttons

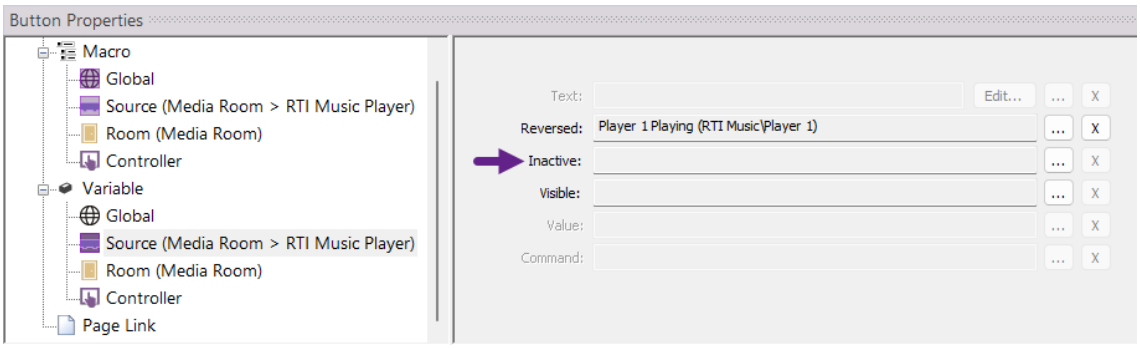
An inactive state button will ignore any button presses. A button may be rendered inactive if a variable condition is true.

In this example, a play button is inactive when playing a song selection.

1. Please select the button and ensure it is properly tagged.

2. Select the variable level in the button properties. There may be other variable types, such as reversed state and visible state, already on the button.

3. Select the ellipsis menu to the right of the inactive variable. Select a driver, then select a variable to force the inactive state.



4. Once the variable is selected, the button will display the reversed state when the music plays a song selection and ignore any button presses.

Visible State Variables

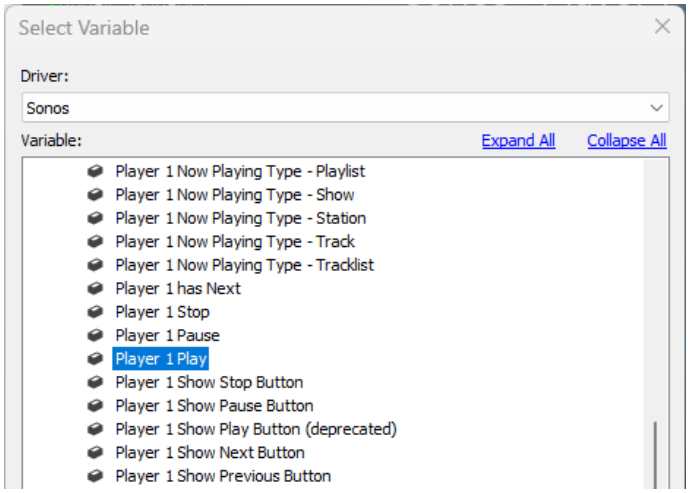
Visible state buttons may be made visible when a variable status is true. Once the variable returns to a false state, the button will hide from view on the interface.

In this example, we will display a pause button only when music is playing. The pause button will be visible only when the music is playing, which can make the interface less busy and aesthetically pleasing to the end user.

1. Ensure the pause button is tagged and properly programmed on the user interface. A reversed state variable may be auto-programmed when the button is tagged.

2. Select the variable level in the button properties. If existing variables are programmed for other types of button feedback, consider evaluating if the current feedback is correct. Realize that the variable levels have priorities, and creating a higher or lower priority could impact the button's functionality.

3. To the right of the visible variable state in the variable configuration window, select the ellipsis menu and find the driver/variable that must be true for the button to be visible on the interface.



The pause button will now perform the following tasks:

1) Pause the player (macro)
2) Display a reversed state when the player is paused (reversed state)
3) Display only when the player is playing (visible state)

Pro Tip: If multiple buttons on a user interface require a visible state variable, consider using layers as an alternative. Buttons can be created on a layer programmed with a visible state variable, saving time and effort.

Select the X button next to each variable option to **delete** a programmed variable.

Gauges and Sliders (Graphs)

Programming feedback on graph objects is straightforward and provides a way to view and control lighting, audio volume, and tone settings. Special graph objects have been built and are available in Integration Designer that can be resized and made horizontally and vertically.

GAUGES

A **gauge** is a graph object prebuilt to use variables to indicate a measurement.

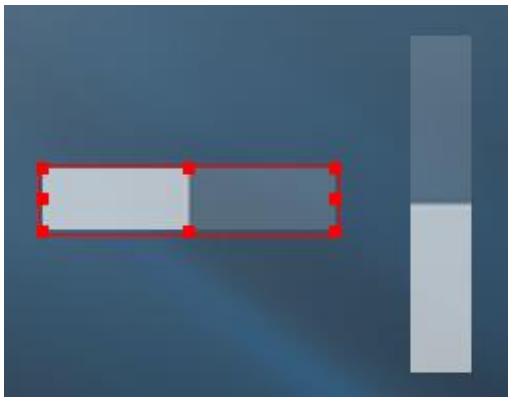
1. Search for "Gauge" in the bitmap library. Take notice of the symbols that appear next to the gauge graphic objects.

   Indicate a graph that can be resized. These can be used on most RTI color screen controllers, including RTiPanel and Virtual Panel devices. They may be brought into the interface horizontally before stretching to create a vertical gauge.

Indicate a graph that cannot be resized. These graphs may not be used on RTiPanel and KA controllers.

Some graph objects outside the Coral template assets will only work on RTI-based controllers, excluding RTiPanel and Android-based controllers.

2. Resizable graphs can be resized using the left or right frame handles or stretched into a vertical orientation using the middle handles.

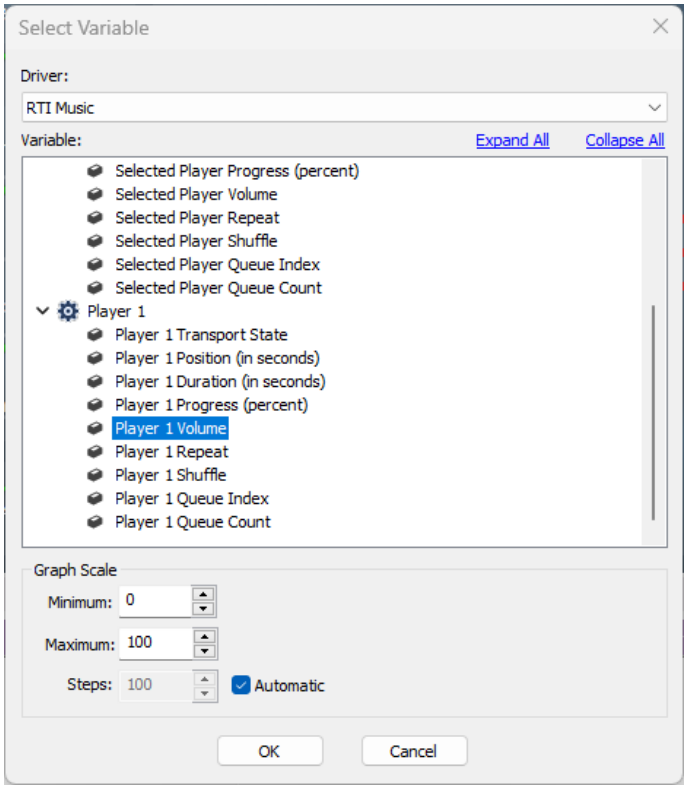


Resizable graphs come in assorted sizes to accommodate larger or smaller user interfaces.

3. Select the **graph and** provide a **tag** name in the tag window.

4. Select a global, source, room, or controller **variable level** in the button properties window.

5. The **Value** option will be available for this type of object. Select the ellipsis menu for the value variable and select the driver and variable you want to use for the graph. Options such as volume, bass, treble, and song progress are an example of some of the variables that may be selected.



The **Graph Scale** at the bottom of the variable window is based on the driver's requirements and does not need to be changed.

Once the variable is selected, the gauge will measure the current volume level in this example.

Pro Tip: Provide buttons to adjust the level of the gauge displayed.

SLIDER

A **slider** is a graph object prebuilt to use a variable to provide feedback on a measurement. The slider can be adjusted up or down (or left to right) to increase or decrease the value.

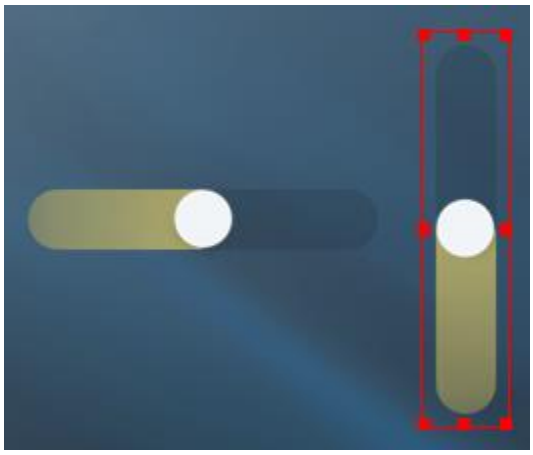
1. Search for "**Slider**" in the bitmap library. Take notice of the symbols that appear next to the gauge graphic objects.

   Indicate a graph that can be resized. These can be used on most RTI color screen controllers, including RTiPanel and Virtual Panel devices. They may be brought into the interface horizontally before stretching to create a vertical gauge.

   Indicate a graph that cannot be resized. These graphs may not be used on RTiPanel and KA controllers.

Some graph objects outside the Coral template assets will only work on RTI-based controllers, excluding RTiPanel and Android-based controllers.

2. Resizable graphs can be resized using the left or right frame handles or stretched into a vertical orientation using the middle handles.



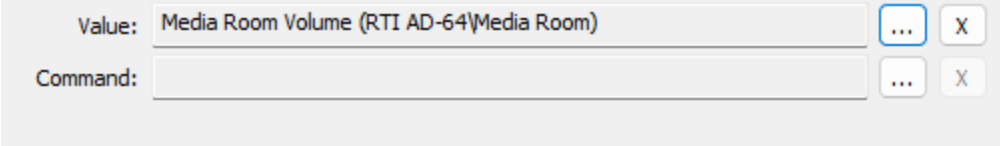
Resizable graphs come in assorted sizes to accommodate larger or smaller user interfaces.

3. Select the **graph and** provide a **tag** name in the tag window.

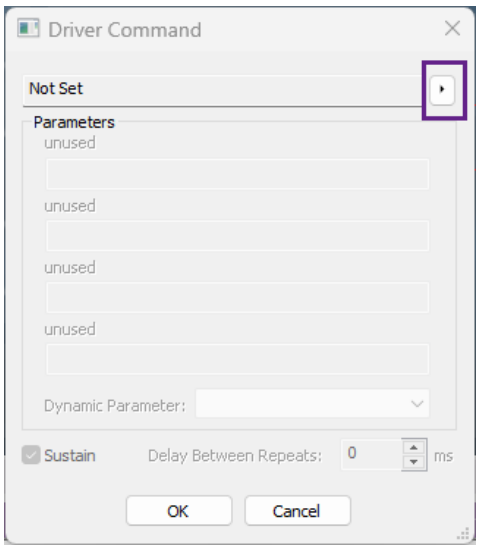
4. Select a global, source, room, or controller **variable level** in the button properties window.

5. An option will be available for both a **value** and a **command**. The value will provide a measurement like a gauge, and the command will allow it to be adjusted.

6. Set the value by selecting the ellipsis menu and choosing a driver variable.



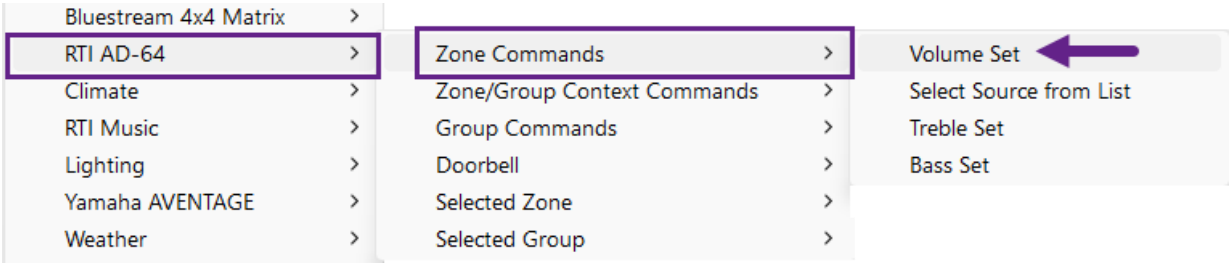
7. Set the command by selecting the ellipsis menu next to the command.

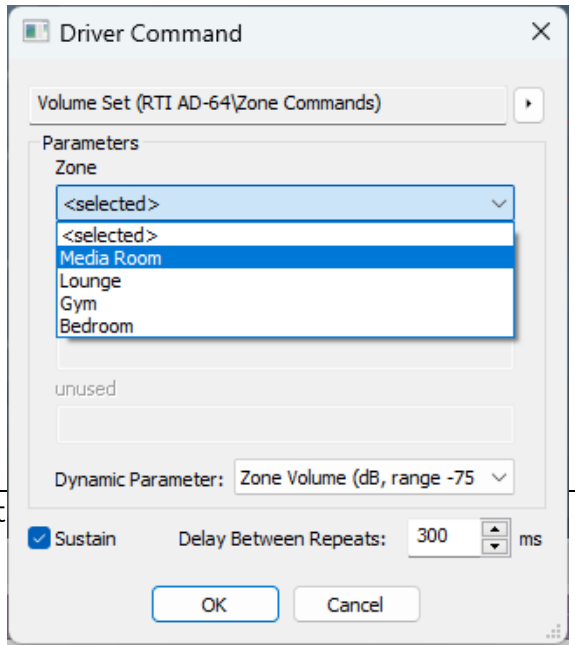


A driver command window will display, allowing the command to be selected. Select the right arrow where it indicates “Not Set.”

Since no commands have been selected, the rest of the options are greyed out.

8. Select the driver, then select the category and the command required. The “Set” command is normally used for setting commands for sliders.

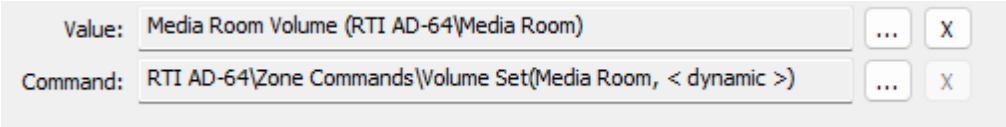




9. If programming a driver that controls multiple zones or players, the **Driver Command** window will prompt for a selection. In this example, select the zone for the volume set command in the dropdown menu.

The values on the bottom of the driver command window are set using the driver requirements and do not need to be adjusted.

Once the value and command are added, the slider will report feedback on the volume level and allow the volume to be raised and lowered.



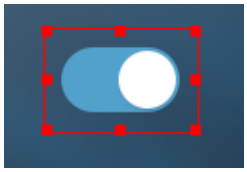
Pro Tip: When using sliders on a user interface, provide a text box that reports the volume level as a reference point. On smaller touchscreens, forego sliders and use a gauge with raise and lower buttons. End users with smaller touchscreens are prone to ramping up the volume accidentally, causing ear or speaker damage.

Toggle Buttons

Toggle buttons are graphic objects designed with images for on and off states. The button can be toggled to either position, and the variable will determine the state of the button.

1. Search for “Toggle” in the bitmap library. Toggles are identified with a symbol.

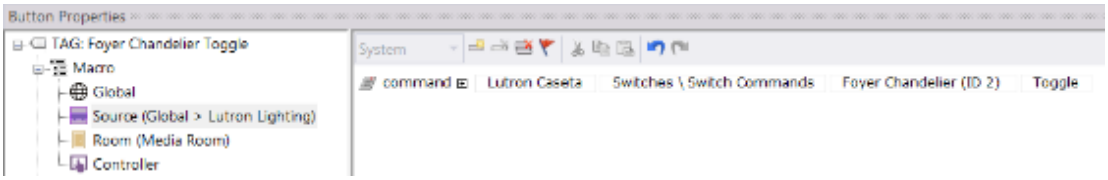
2. Place the toggle button into the controller interface and **tag** it appropriately.



Toggles may not be resized or change orientations, but the frame handles can widen the footprint.

Some toggle objects outside the Coral template assets will only work on RTI-based controllers, excluding RTiPanel and Android-based controllers.

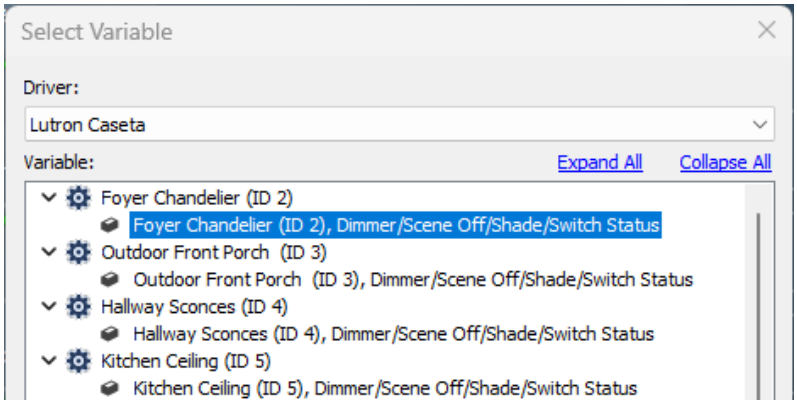
3. **Toggle** objects require a macro and a variable. Start by selecting a macro level and entering the command in the macro window.



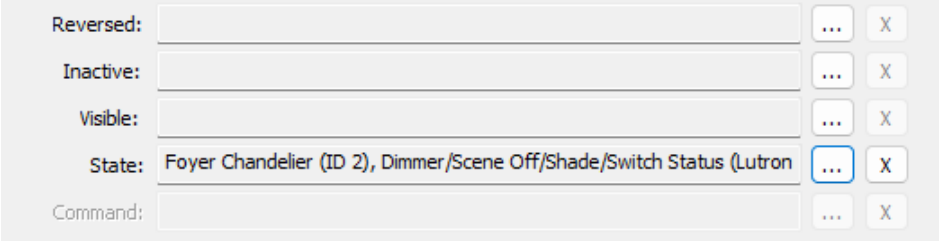
A toggle command must be used for toggle buttons. A system variable test must be utilized if a toggle command is unavailable.

4. Select a **variable level** in the button properties window. For toggle objects, a **state** option will be displayed. A variable is used to reflect the state of the device.

5. Select the ellipsis menu for the state. Select the driver, then the variable for the device.

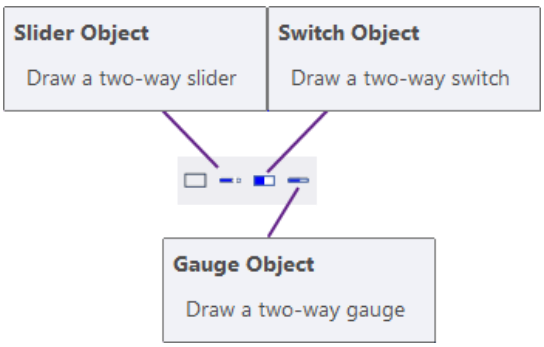


The toggle button will now toggle the lighting switch and reflect the on and off status of the device.



Drawing Graphs and Toggles

iOS and Android-based RTiPanel devices and KA-series controllers support the ability to **draw** a gauge, slider, or toggle button. These options will be available in the button styles toolbar for applicable controllers.



1. Select the **Draw Button** tool in the selection palette.

2. Select the slider, switch, or gauge object in the button styles toolbar.

3. Holding the left mouse button, draw the object on the user interface. Sliders and gauges may be drawn horizontally or vertically.

Item Lists

Item lists provide a selectable scrolling list of feedback from music libraries, rooms, activities, or favorite channels. Depending on the RTI controller, they may be programmed using different methods.

Draw List – For iOS and Android-based RTiPanel d devices, it is necessary to draw an item list using the two-way scrolling list icon in draw mode.

Item List (Gesture) – A gesture-controlled item list allowing navigation and selection without buttons.

Item List (Buttons) – An item list that requires buttons using the object tab to identify multiple lists, scroll, and make selections.

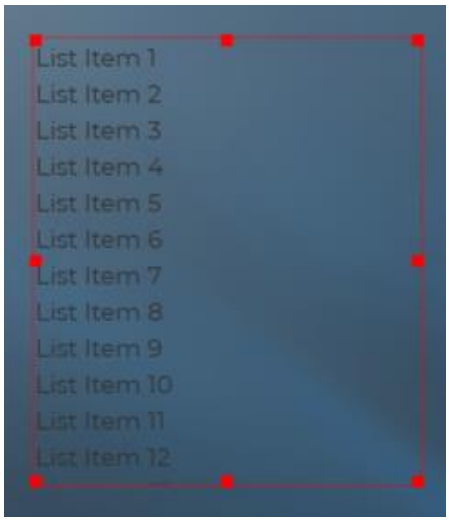
Search for item lists in the bitmap library by entering "**Item List**" in the search box. They can be identified with a symbol. In some situations, the preference may be to use a two-way item list scroll bar. They can be identified with a symbol.

Draw List

1. Select the **draw mode** icon in the selection toolbar and the **two-way scrolling list** icon in the style's toolbar.

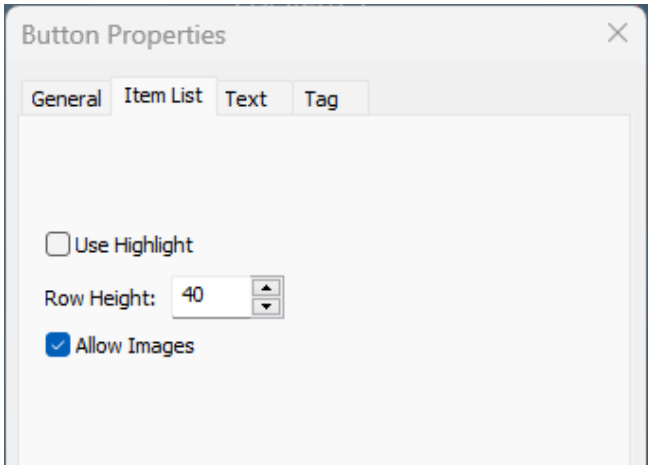
2. Holding down the mouse, **draw** the **item list** on the screen.

3. Return to **Select Button** mode and refine size and location using the item list frame handles.



In this example, the text color is too dark and close together. Use the button styles toolbar to change the item list's text or fill color.

4. Right-click on the item list, then select “Edit Properties.” Select the item list tab in the button properties window.



Use Highlight: Select this option if you want the current selection to highlight when pressed.

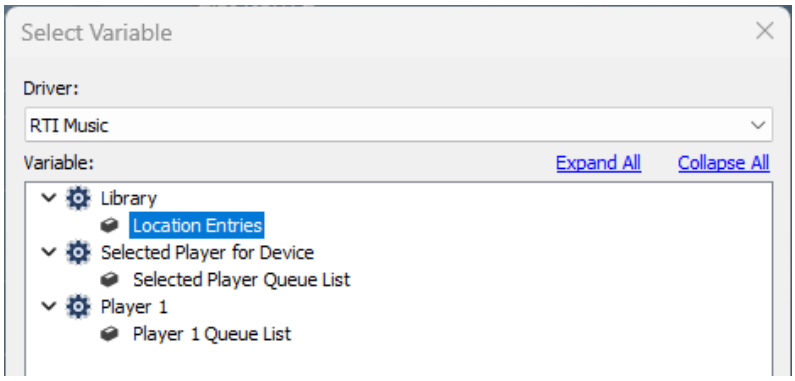
Row Height: Increase or decrease the row height if you want the text to be further apart or closer.

Allow Images: Used with RTI Music item lists only, but always selected by default.

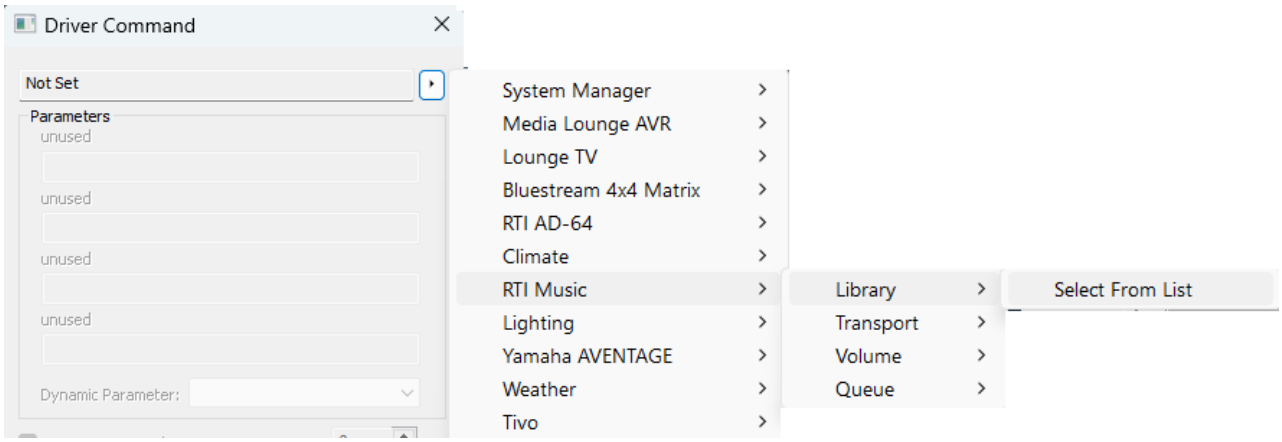
5. Once the item list is properly sized and formatted, **tag** the item list if a tag name does not exist. Select a **variable level** in the **button properties.**

A list and command will need to be programmed for the item list.

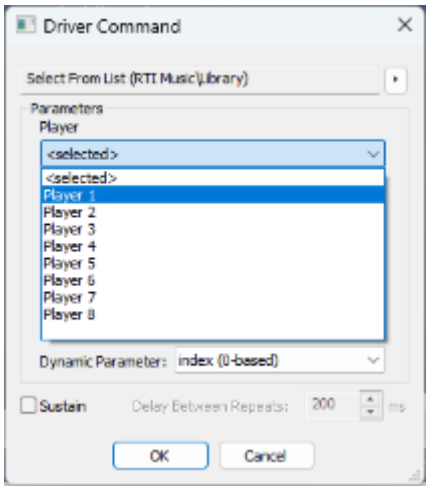
6. Select the ellipsis menu for the list variable type, then select the driver and variable for the feedback required.



7. Once the list is programmed, select the ellipsis menu next to the command.



8. Select the player or zone from the **driver command** list.



Dynamic Parameter should not be changed and is usually built into the driver command.

Sustain may be left unchecked in most cases.

9. Select **OK** to apply the settings.

Gesture Controlled Item Lists

Item lists for applicable controllers may be added using the bitmap library.

1. Search for "**Item List**" in the bitmap library **search** box. Some item lists may be represented by a white or black bar, and some might not display a graphic in the bitmap library. It may be necessary to hover over some item lists to see if the size accommodates the controller.

Hovering over each scrolling list bar will provide the resolution, making it easier to find the right size.

Item Lists may be represented with a black or white bar or without embedded graphics.

2. Drag the **item list** to the user interface and resize it vertically and horizontally to fit the requirements.

3. Once the **item list** is resized and tagged, select a **variable level** in the button properties window.

4. Program the **List** and **Command** variables.

Use gesture control to scroll through the list of items on the user interface and select an entry.

Item Lists with Object Control

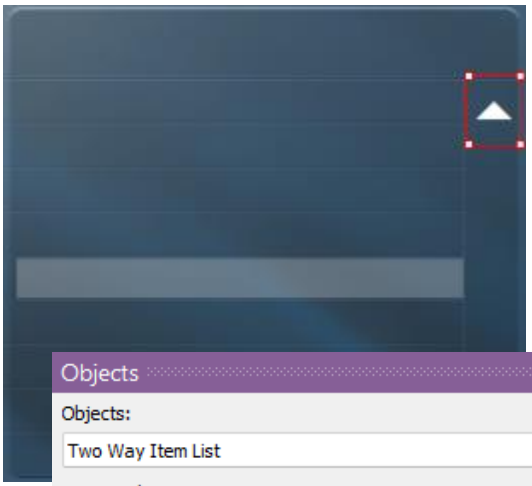
Certain controllers require objects to scroll through the item lists and make selections.

1. Search for "**Item List**" in the **bitmaps library** and bring it into the user interface. Ensure the width will accommodate the size of the list on the controller.

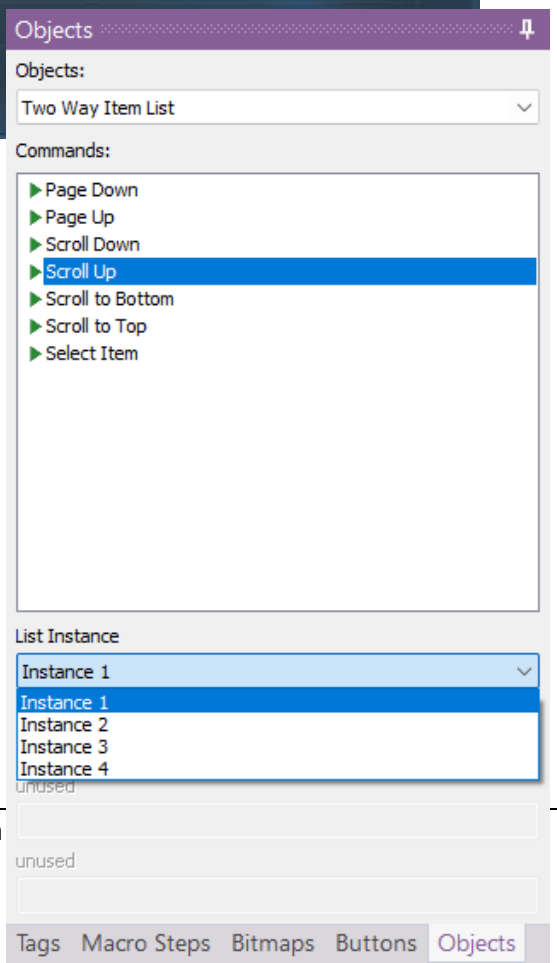
2. Optionally, an item list shadow box can be placed behind the item list to provide a better aesthetic.

3. **Overlap** the item list on the top of the shadow box and align it appropriately.

4. Draw an up and a down button on the upper left and right of the shadow box. These will be a way to scroll through the list.



Use the **transport symbols** in the text editor to create the up and down symbols in the drawn buttons.



5. Select the **object** tab in the tabbed menu. Select the **Two-Way Item List** object from the list of available objects.

Several objects exist for the **two-way item list**, including **page**, **scroll**, **select**, and navigate to the **top** and **bottom** of the list.

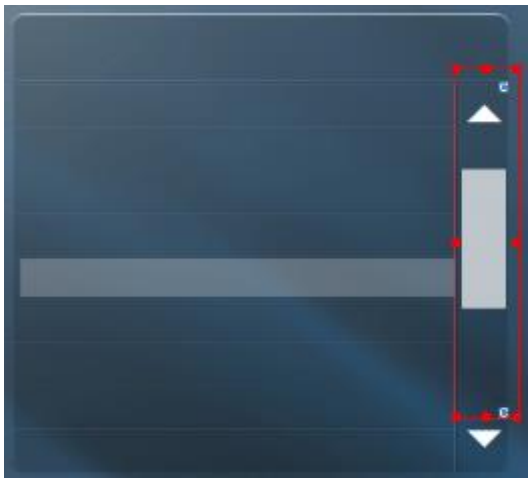
These options can be added as buttons if more functionality is required.

Drag the **scroll up** and **down** commands to the up and down arrow buttons. No tags are necessary, and command objects will be added to the buttons. A "C" will appear on the button indicating a command.

If multiple lists are utilized on the controller, select the **instance** of each list. Up to **four lists** or **instances** can be added to a controller.

6. Optionally, you may add a **scroll bar** between the scroll functions in the shadow box to provide where the end user is in the list.

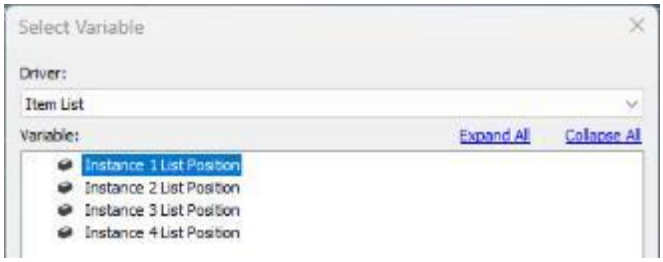
   Search for "**Item List**" in the **bitmap library** search box and find a scrolling bar that fits the shadow box dimensions. Set the scrolling bar to **inactive** in the **button properties** and align it to the right of the shadow box.



7. **Tag** the scrolling bar before selecting a variable level in the button properties.

8. Select the ellipsis menu next to the **value** type, then select **Item List** in the list of available drivers.

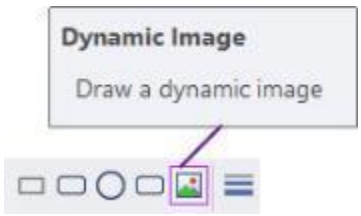
9. Populate the variable with the scrolling list's corresponding instance list position number.



Dynamic Images

Dynamic Images are used for album cover art, current stations, and other images based on feedback. If a controller supports Dynamic Images, they can be drawn and programmed with feedback.

1. Select the **Draw Button** mode in the selection toolbar, then select the **Dynamic Image** in the button styles toolbar.



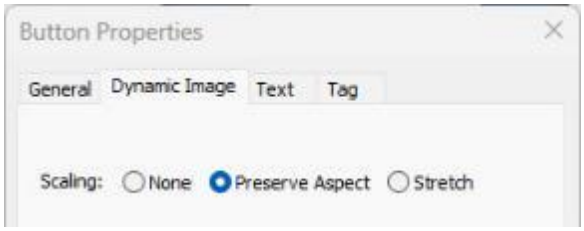
2. Hold the left mouse button while drawing the dynamic image. For best results, try to match the height and width of the button so images are not distorted.

3. Refine the image size and position by returning to selection mode and using the frame handles to adjust.

The image indicates that **no scaling** is being used.

4. Once the image is resized and positioned, right-click the button and select **edit properties**.

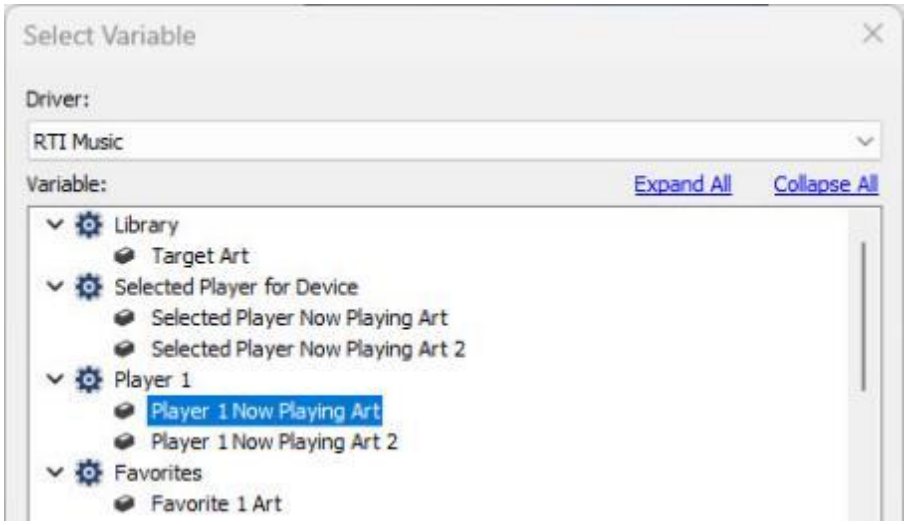
5. Select the **Dynamic Image** tab and set the **scaling** preferred.



6. **Tag** the dynamic image and select a **variable level** in the **button properties**.

7. An **Image** option will be available for dynamic images. Select the ellipsis menu for the image.

8. Select the driver and variable for the image that is required.



The **dynamic image** will display the music player now playing art. You can set the cover art to inactive or use it to send a command.

Pro Tip: Selecting the proper scaling method can be trial and error, especially if the height and width of the dynamic image differ. Experiment with different scaling methods until you are satisfied with the result.

Image Lists

An **Image List** is a specialized object consisting of multiple images. Based on feedback, the designated image will be displayed. The weather driver uses an image list to display the current weather graphic, and frame indicators report the current frame in use.

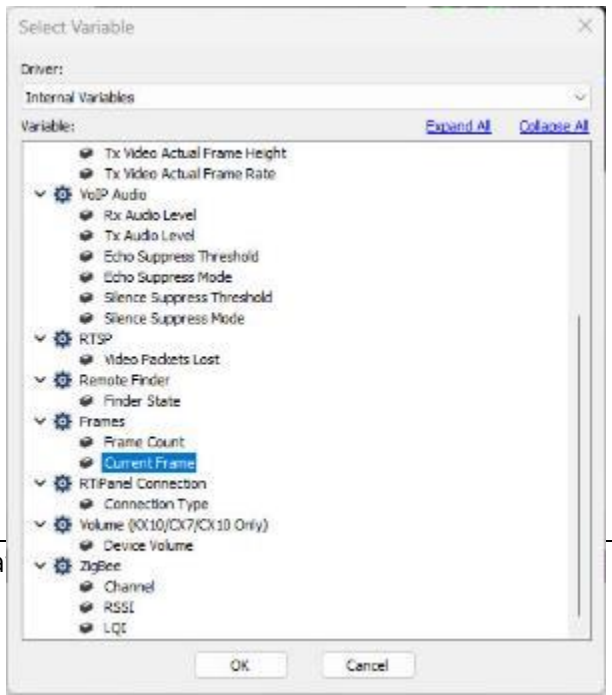
1. Search for "**frame indicator**" in the **bitmap library**. An image list is indicated with a symbol.

2. Select the corresponding **frame indicator** based on the **number of frames** in use on the controller. The controller uses three frames in this example, so a three-frame indicator is required.

Select different frame sizes until you are satisfied with the result.

3. Position the frame in an area of the screen that will not distract from the functionality.

4. **Tag** the **frame indicator** with the standard tag name "**Frame Indicator**." Since there is the potential for several controllers using the same tag name, select the **controller**-level **variable** in the **button properties**.



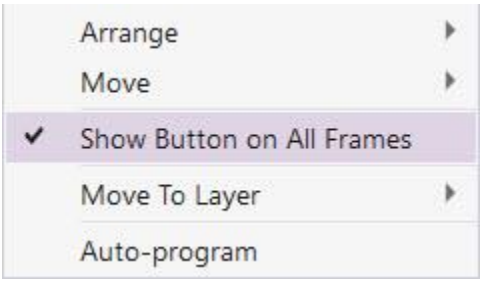
5. Next to **Index**, select the **ellipsis menu**. Since frames are not dependent on a specific driver, select the **Internal Variables** from the driver list.

The **internal variables** will display all the built-in controller functions in the

software. Features such as **VOIP**, **RTSP**, **Connection**, and **Zigbee variables** can be found in this section.

6. Select the **Current Frame** under the **Frames** category.

7. Right-click the **frame indicator** and select the "**Show Button on All Frames**" option instead of placing the frame indicator on each frame.

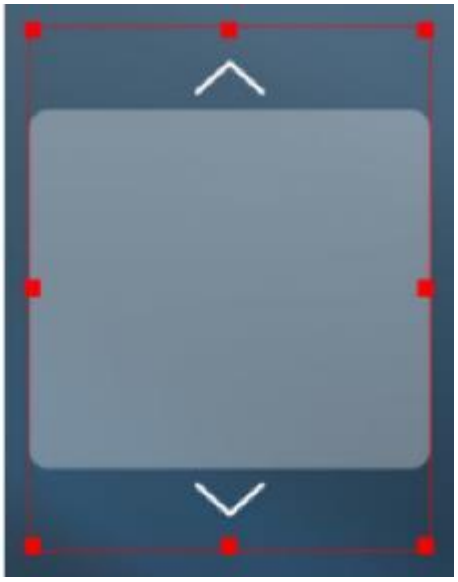


Pro Tip: When using frames, always use frame indicators so the end user can tell how many frames they can advance and the current frame position. Use frames for pages that share functionality like channel presets, favorites, and status pages. Ensure the end user has room to swipe from left to right on the interface.

Spinner (XP-8v OSD)

Spinners can be used as **item lists** when specifying an **XP-8v on-screen interface** using the available HDMI output on the processor. A spinner is programmed with variables in the same way as an item list. The difference is that the spinner must be selected before scrolling up, down, or selecting items in the list when navigating the on-screen display.

1. Search for **Spinner** in the bitmap library. Bring the spinner into the user interface after verifying the size.



2. **Tag** the spinner and select a **variable level** under **the button properties**.

3. Program with the same requirements as an item list.

Advanced Macro Programming

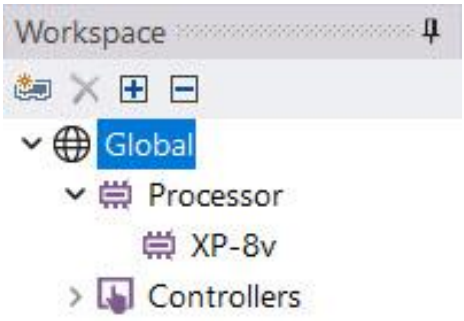
Advanced macro programming options included with Integration Designer software make RTI the ideal solution for custom integrators in any market or area of specialty, no matter the project size.

Creating System Macros

It is possible to create **system macros** in Integration Designer and call them from other **macros** and **events**. Global, room, and source macros can be created without a button.

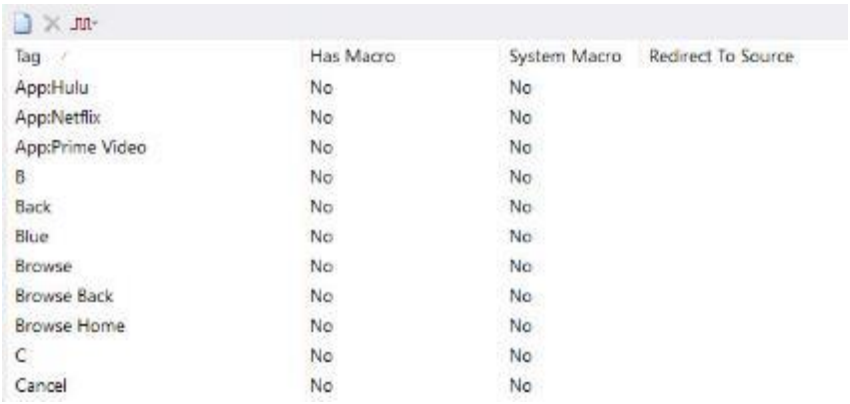
GLOBAL SYSTEM MACROS

1. Select the **global** area in the project **workspace**.



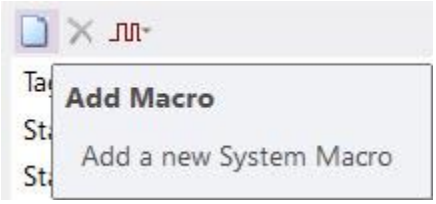
2. Select the **macro list** in the **global** configuration to the right of the **workspace**.

A list of all the **tags** that exist in the **global** area appears. A row labeled **Has Macro** indicates if a **global macro** is programmed for that **tag** name. **System Macro** indicates a **system macro** was programmed with a **global macro level** for that **tag** name. **Redirect to Source** is a way to redefine the macro for that tag by **linking** it to a source available in the global area. For example, **redirecting** a Volume Up tag to a device placed in global will populate the volume commands for that device if they exist on a global macro level.



Tag	Has Macro	System Macro	Redirect To Source
App:Hulu	No	No	
App:Netflix	No	No	
App:Prime Video	No	No	
B	No	No	
Back	No	No	
Blue	No	No	
Browse	No	No	
Browse Back	No	No	
Browse Home	No	No	
C	No	No	
Cancel	No	No	

3. Select the **Add Macro** icon on the toolbar to add a **system macro**.

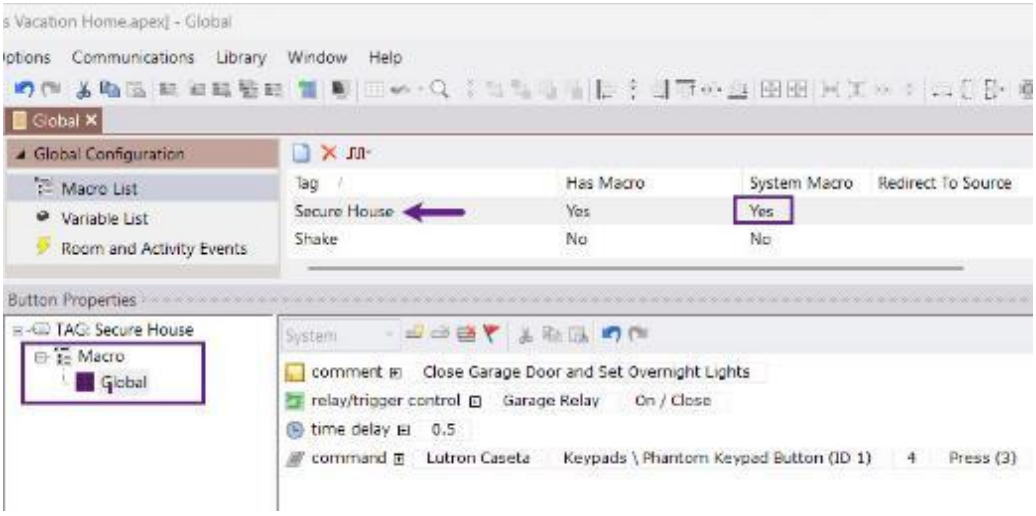


4. Create a general tag name for a function that would typically exist on a global macro level, meaning a shared function in the project and not specific to a room.

Note that the **tag** name and the **macro** name are the same. By creating a name for the macro, a tag name was created.

5. Create a **macro** to secure the home overnight using the **new tag** and **macro name**.

   The new macro will appear at the top of the list to create a macro without the need to scroll. This system macro can now be called from other macros or an event.



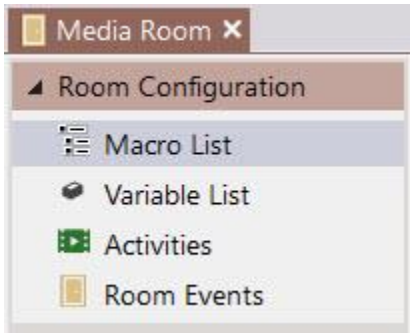
Once the macro has been added and programmed, a **Yes** will appear under the **System Macro** column. In addition, if that tag name is entered anywhere in the project file, it will auto-populate the global system macro command.

ROOM SYSTEM MACROS

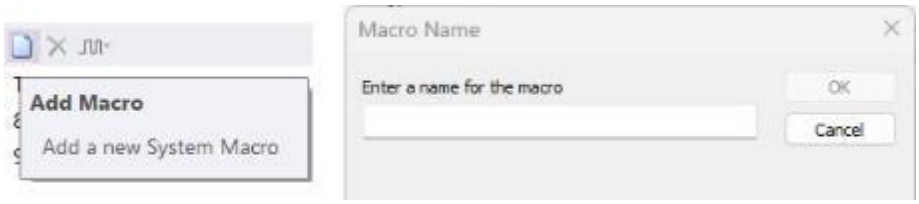
1. Select any room in the workspace.

2. Select **Macro List** in the room configuration window.

   A list of tags that are in the room will be displayed. If a room macro has been programmed with the tag, the **Has Macro** field will indicate "**Yes**." **System Macro** indicates if a system macro was programmed with a room macro, which this process describes. **Redirect Source** will allow the RTI installer to redirect any tag to a device available in the room and global area. In this way, the tag can auto-populate a room macro based on the functionality of that source device.



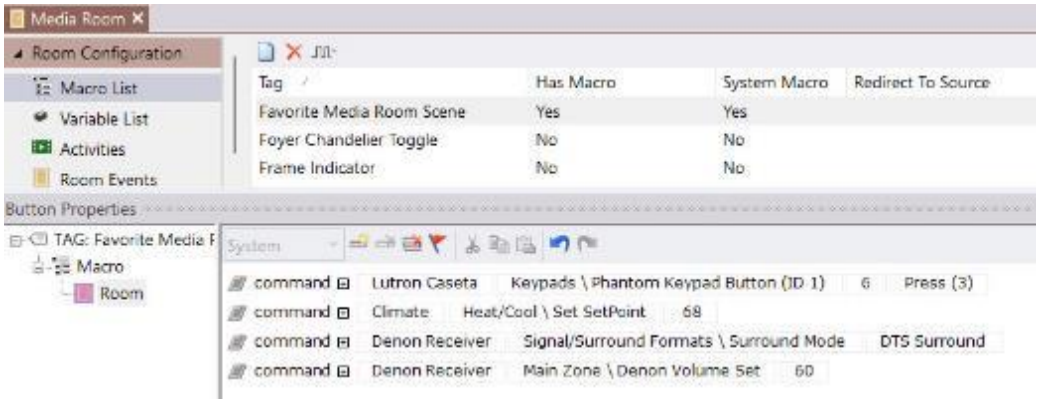
3. Select the **Add Macro** icon on the toolbar to add a system macro.



Note that the **tag** name and the **macro** name are the same. A tag name was also created by creating a name for the macro.

4. Create a tag and macro name that best applies to the room selected. If the function is broad and covers other room functions, consider making it a global system macro.

5. Program the macro with a room-level macro.



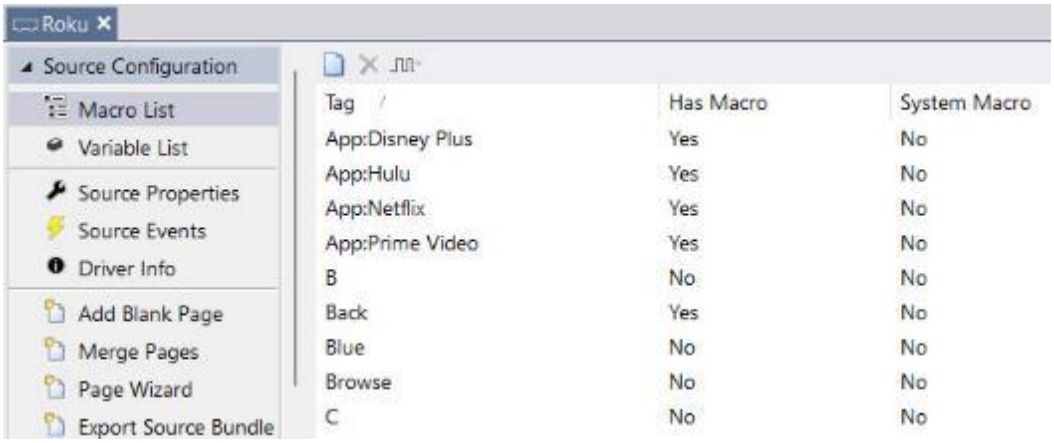
The **room system macro** can be called from other macros and events.

Only **system macros** may be **deleted** by selecting the **X** in the toolbar.

SOURCE SYSTEM MACROS

1. Select any source device in the project workspace.

2. Select Macro List in the source configuration window.

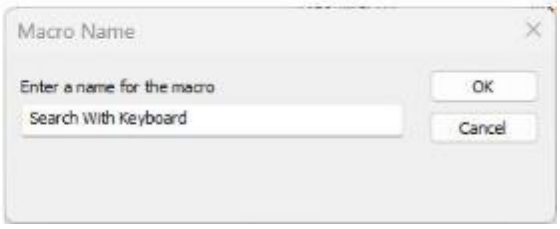


Tag: The name of the tag created on the source device

Has Macro: Indicates if a macro has been programmed for the tag name.

System Macro: This indicates a system macro has been programmed.

3. Select **Add New Macro** from the toolbar. Provide a **tag** and **command name** that best applies to the current source device.



4. **Program** a **macro** for the new tag name. The macro programmed will be a source-level macro available on the selected source device.

This source macro can be called from other macros and events.

MACRO LIST PROGRAMMING

Macro list programming is ideal for programming system macros but can also be used to program global, source, room, and controller macros. In many cases, programming via the macro list can be easier than programming a series of buttons with the same macro level.

Scenario: Programming a source macro on a series of button favorites.

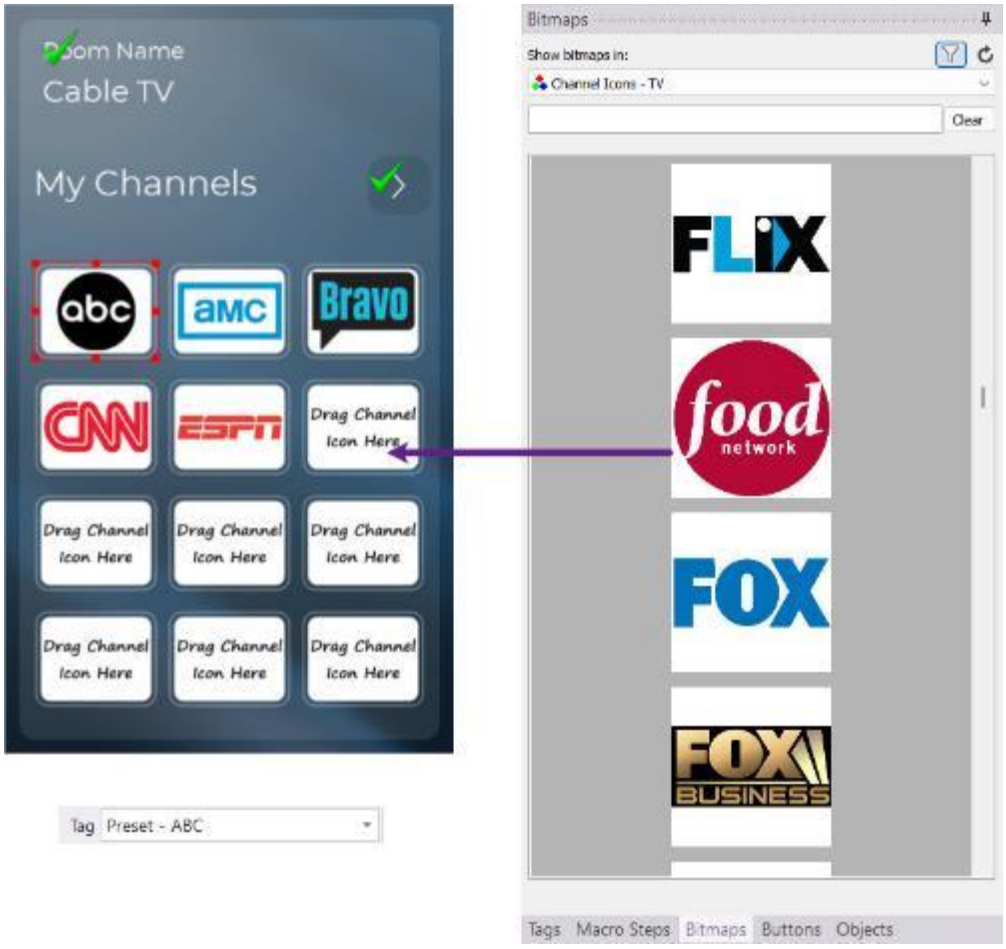
In this example, several channel favorites need to be programmed. None of these buttons have tags. Usually, the RTI installer would have to tag these buttons, select a source-level macro, and program each button.

The **macro list** makes programming multiple tags with the same macro type easier.

First, tag the buttons by dropping favorite channel icons onto the button. These icons have been created with a special naming convention that automatically tags the button.

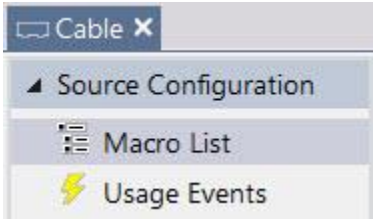
1. Select the channel icons bitmap library.

2. Drag each channel icon to the button in the user interface. Note that these icons will automatically tag the button with a tag name of “Preset – Channel Name.”



3. Once a channel icon has been added to all the buttons, ensure they all have consistent tags.

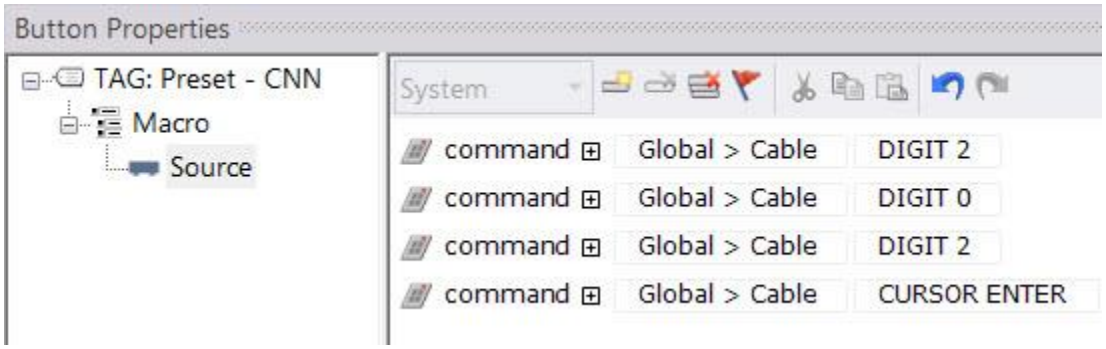
4. Select the source device for the favorite channels in the workspace, then select Macro List in the source configuration window.



5. Scroll down to the preset tag names in the alphabetical macro list.

Preset - ABC	No	No
Preset - AMC	No	No
Preset - Bravo	No	No
Preset - CNN	No	No
Preset - ESPN	No	No
Preset - Food Network	No	No
Preset - HBO	No	No
Preset - HGTV	No	No
Preset - MLB Network	No	No
Preset - MSNBC	No	No
Preset - MTV	No	No
Preset - TLC	No	No

6. Select each preset and program the source macro in the macro window.



7. Once programming is complete, return to the user interface and ensure everything is programmed. A green check mark on each button should indicate that the button is tagged and programmed.

Green check marks indicate the button is tagged and programmed in most cases.

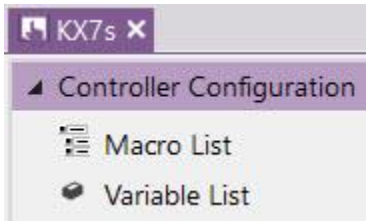
It is good practice to spot-check some buttons to ensure the tag and programming are correct.

CONTROLLER SYSTEM MACRO

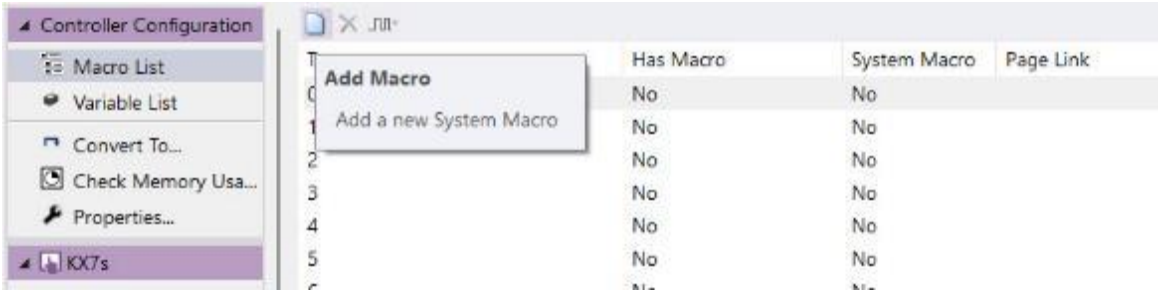
Controller system macros are the only macro type that may not be called from other macros or events. However, programming via the macro list will allow the RTI installer to program controller-level macros when programming multiple buttons is required.

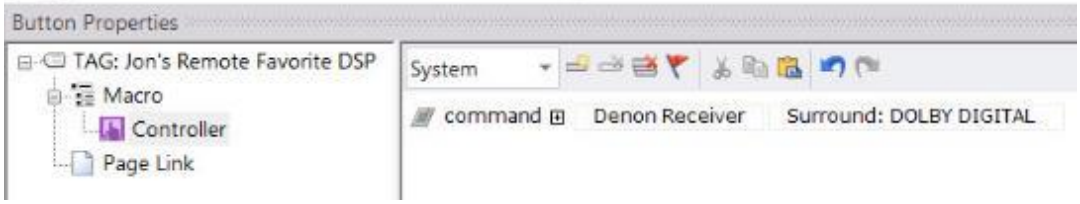
1. Select any controller in the workspace.

2. Select Macro List in the controller configuration window.



3. Create system macros by creating a tag and macro name. Programming existing tags that do not have an existing controller macro is also possible.





4. Another option available in the macro list is the **Page Link** function. **Page links** can be set on a tag basis, usually on the controller level.

Page links can be viewed, changed, or removed from existing tags.

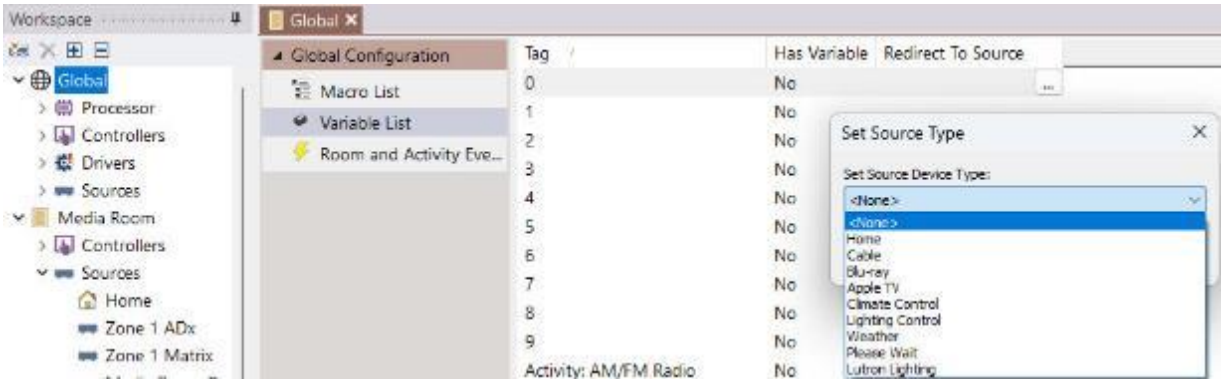
5. Once a system macro or macro is programmed, creating the tag anywhere on the controller will auto-program a controller-level macro.

VARIABLE LISTS

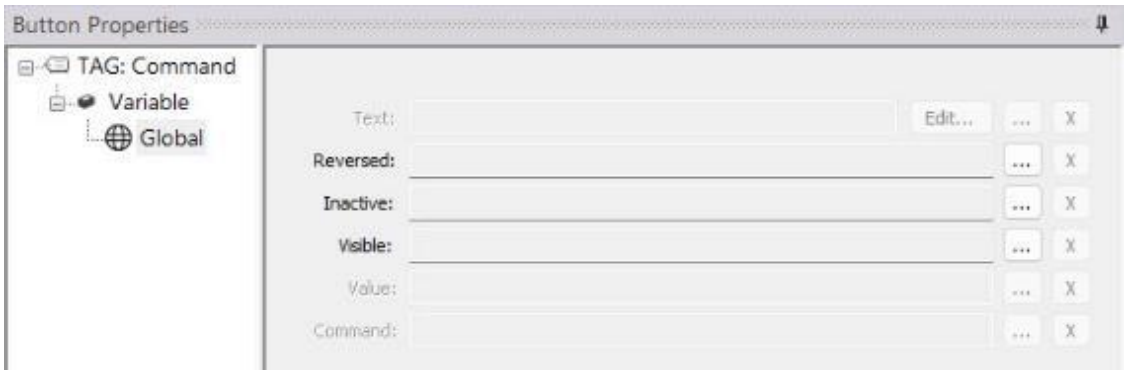
Variable lists provide fewer options than macro lists but can be used to **view**, **program** existing tags, and **redirect** functionality.

GLOBAL VARIABLE LIST

1. Select the **global** area in the workspace, then select the **variable list**.

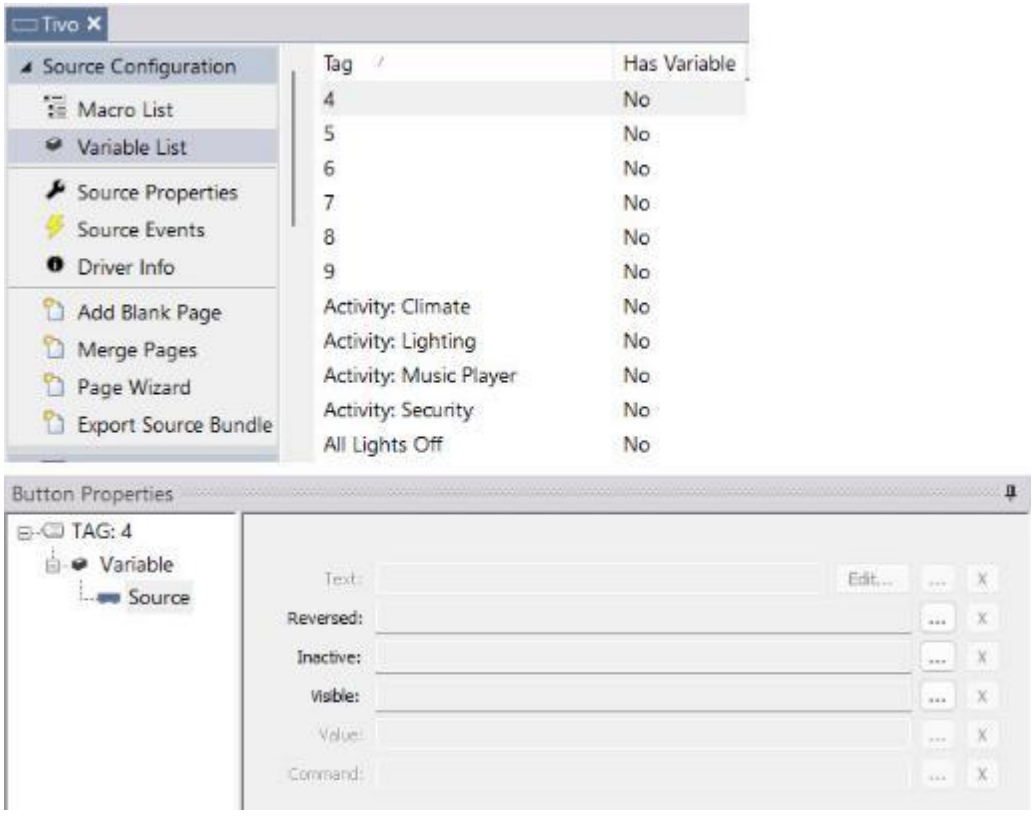


Tags in the **global** area can be **viewed** or **redirected** to any source device in the global area for variable programming on a **global** level.



SOURCE VARIABLE LIST

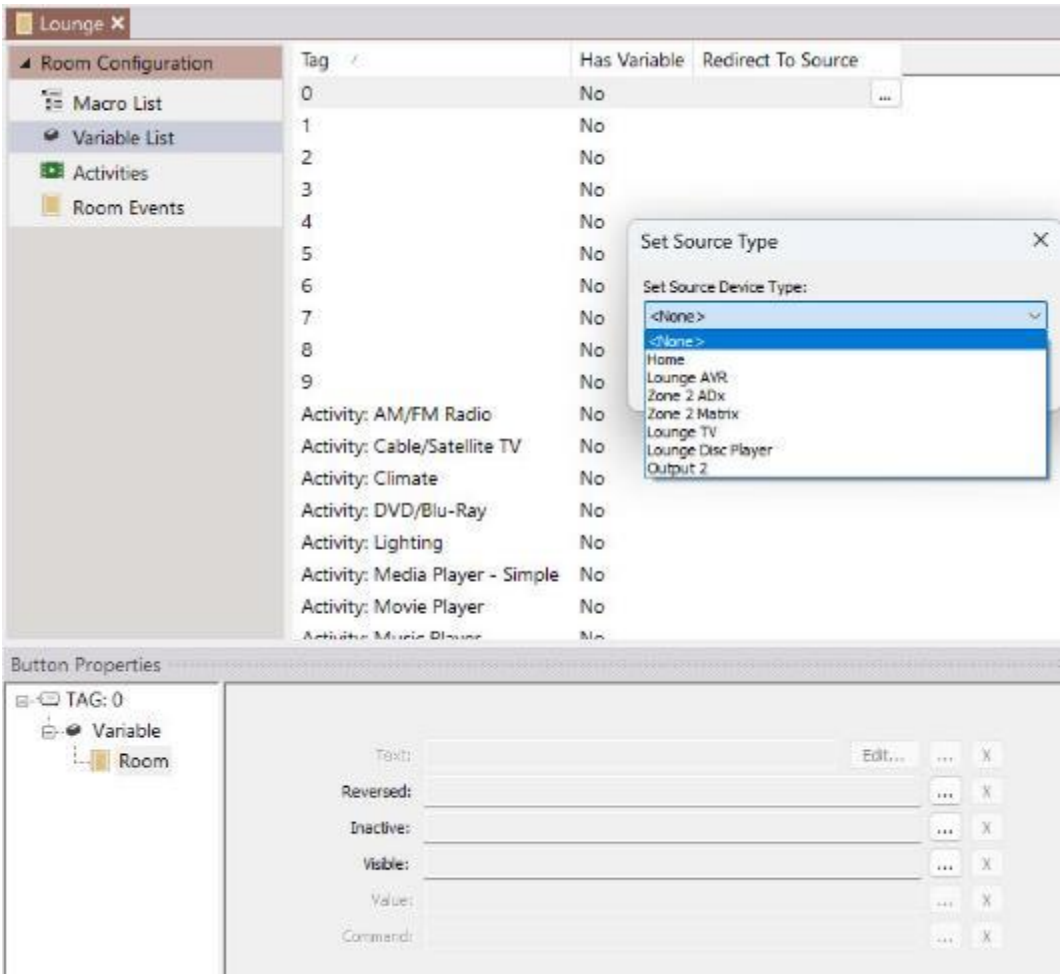
1. Select any **source** in the workspace, then select the **variable list** in the source configuration menu.



Tags for the selected source device can be viewed and programmed in the variable window with a source-level variable.

ROOM VARIABLE LIST

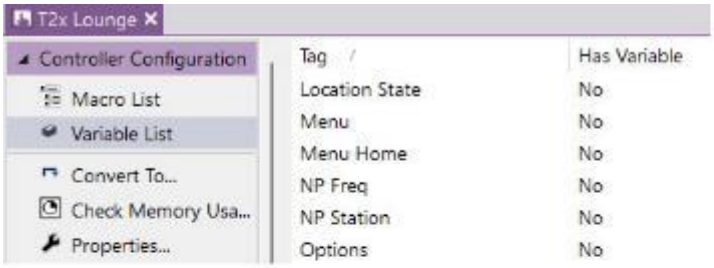
1. Select any **room** in the workspace, then select the **variable list** in the **room configuration** window.

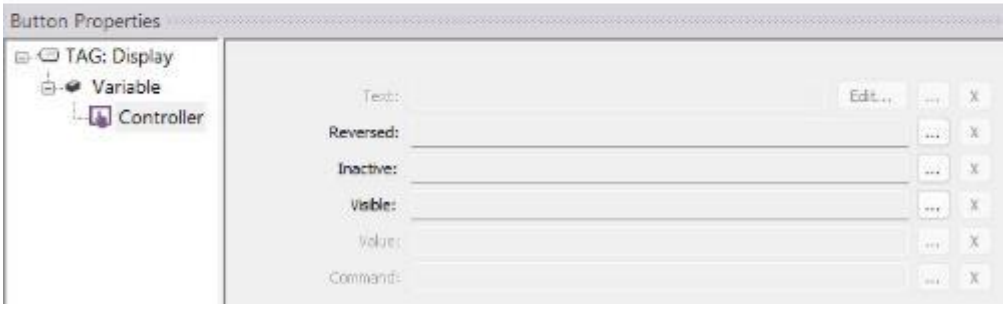


Variables available in the room may be **viewed** or **programmed** with a **room-level variable** in the variable window. The functionality for any tag can be **redirected** to any source device available in the room.

CONTROLLER VARIABLE LIST

1. Select any **controller** in the workspace, then select the **variable list** in the controller configuration window.



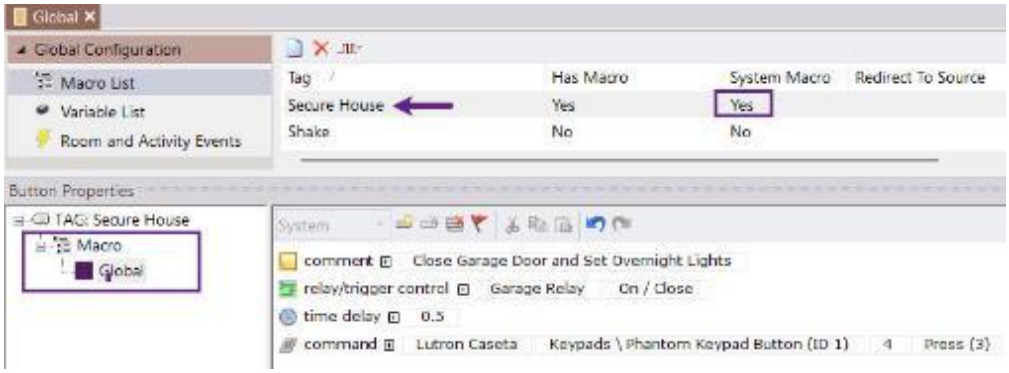


Variables available on the selected **controller** can be **viewed** or **programmed** with **controller**-**level variables**.

CALL MACRO

The **Call Macro** step can run **system macros** created on a global, room, or source level from any macro in any room. Macros may be called **immediately** or in **delay** mode. When in delay mode, macros run on other processor macro engines and avoid holding up the interface during processing.

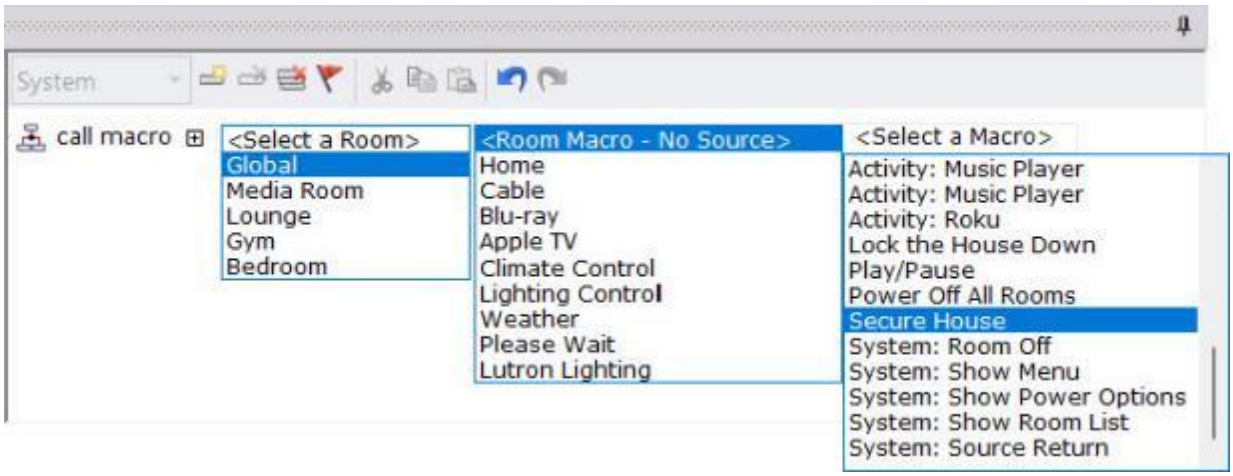
1. Create a **global system macro** by selecting the **global** area in the **workspace**. Select **Macro List**, then **Add New Macro**.



2. Create a button macro in any room, on any source, and on any controller. Name the tag "**Lock House**."

3. Since the macro is centralized, there is no need to recreate the macro steps on a button. Select any **macro level** in the button properties.

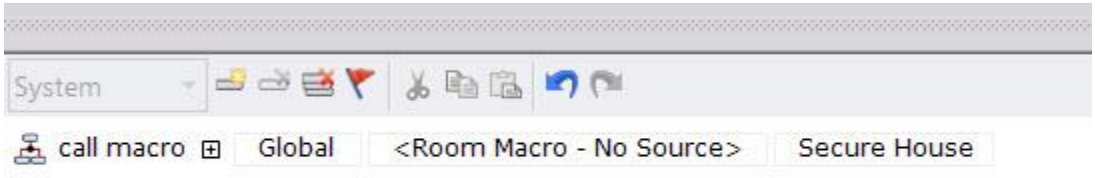
4. Drag or double-click the **Call Macro** command from the macro steps. Select the arguments from the drop-down menu to call the **global system macro** "Secure House."



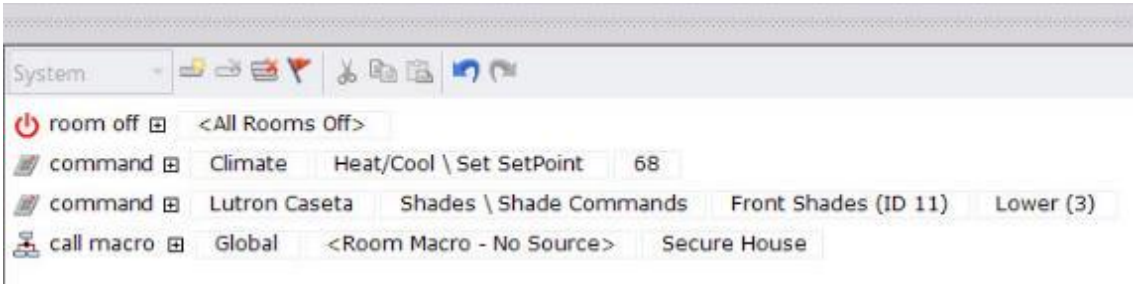
For **<Select a Room>,** select the **global** area where the system macro was created.

For the **<Room Macro – No Source>**, select this option if the system macro was created on a source device as a source macro. Since this system macro was created in a room (global), leave this option blank.

For **<Select a Macro>,** select the **global** system macro "Secure House" available in global in the system workspace.



The **Call Macro** command can also run with **other** commands within the macro.



This macro will **turn off all the rooms** in the home, **set the temperature**, lower the **shades,** and **call the macro** created in the **global** area to **secure the house** for the night.

Delaying Call Macro Commands

The **Call Macro** command may be programmed to send a command after a configured number of **minutes** and **seconds**.

This macro will occupy the screen for twenty-five seconds while the projector warms up to accept an input change. The end user is **prevented** from entering commands while the macro runs.

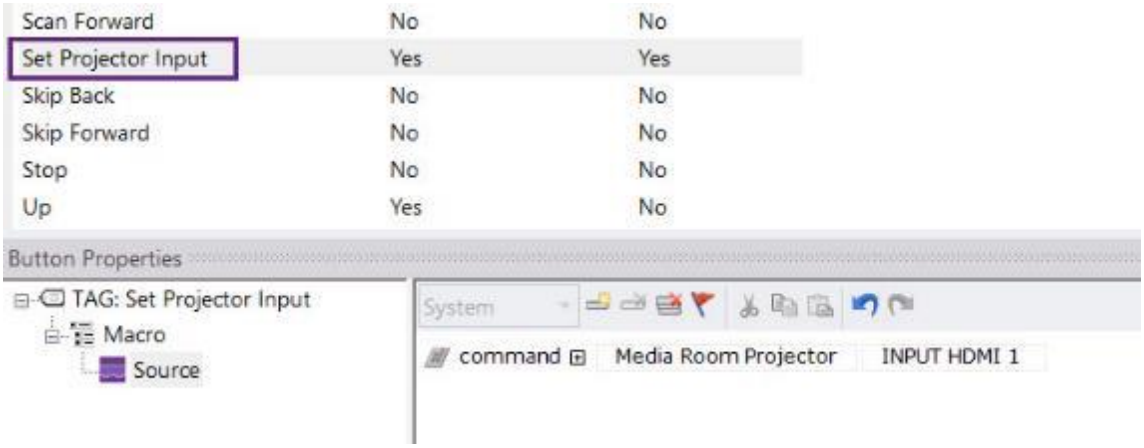
It is possible to run the macro in the **background** while utilizing other macro engines. Doing so will allow the end user to interact with the user interface while the macro commands are sent after a specified time elapses.

Use the **Call Macro** command in delayed mode.

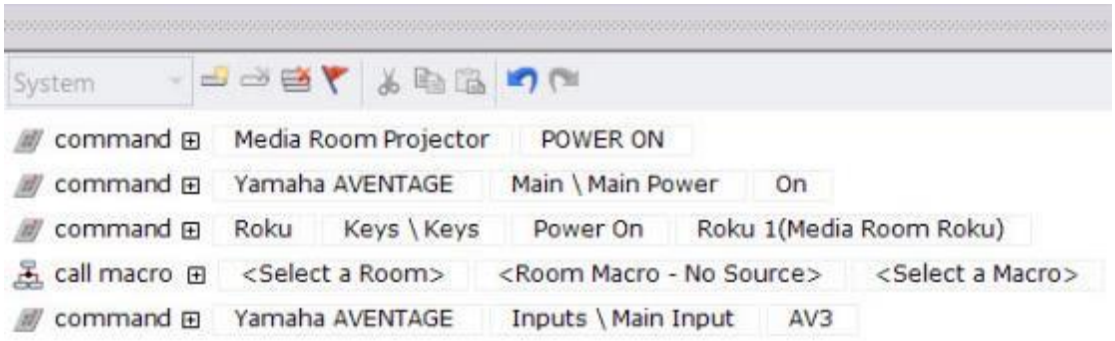
1. Select the projector **source device** in the **workspace**. Select **Macro List** in the source configuration window.

2. Select the "**Add New Macro**" button in the toolbar.

3. Name the macro "**Set Projector Input**." **Program** a **source macro** to switch the **input** on the projector to the appropriate input.



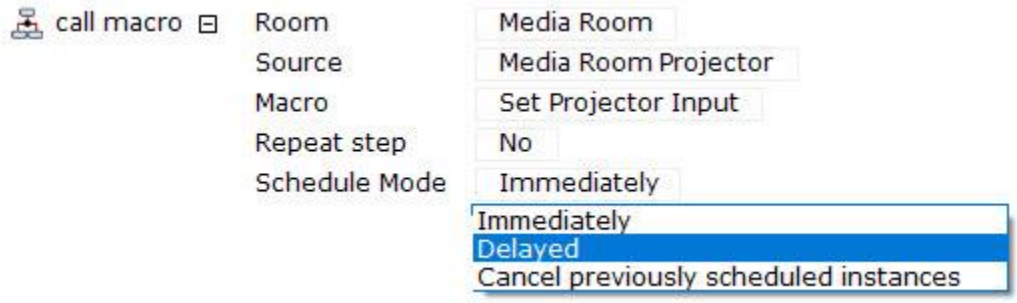
4. Return to the button macro and **delete** the projector **input** macro. **Delete** the 25-second **delay** as well. Insert a "Call Macro" command where the delay was.



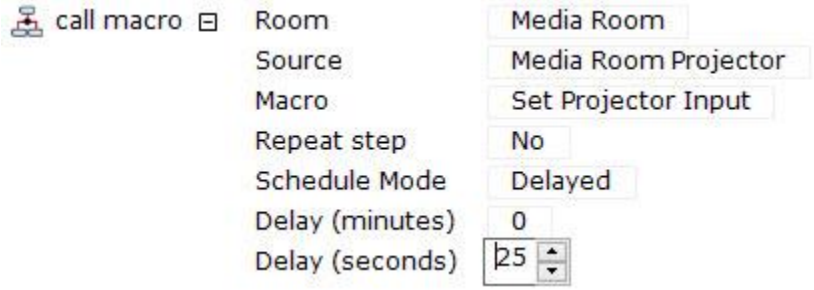
5. For **<Select a Room>**, select the **room** where the **projector** is located. Select the projector source device for **<Room Macro – No Source>**. Select the **Set Projector Input** system macro for the macro selection.

6. Select the **+** sign to the right of the **Call Macro** command.

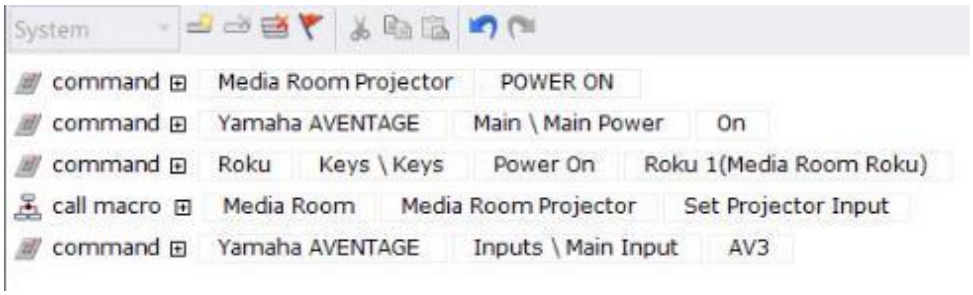
7. Select the dropdown menu for **Schedule Mode** and select **Delayed**.



8. For **Delay (minutes)**, enter the required **minutes** before a command can be sent. **For Delay (seconds)**, enter the required **seconds** before a command can be sent. You can enter the time manually or use the up and down arrows.



9. Once configured, select the **+ sign** to collapse the menu.



The macro will now run quickly without tying up the screen and locking out the end user from pressing any buttons. The projector input will be sent in twenty-five seconds using another macro engine on the processor.

Pro Tip: Consider **canceling** a **Call Macro** with a delay if the button is pressed twice. Doing so will prevent multiple commands from being sent. Under **Schedule Mode**, select the "**Cancel previously scheduled instances**" option.

MULTI-ROOM CONTROLLER PROGRAMMING

When adding a controller to the global area, specific features are activated:

- Room Selection Menu
- Multiple Room Macro Levels
- Specialized Tags with commands and variables designed for multi-room controllers are added to the user interface.

Selection Menus

A **room selection menu** is required when a controller is added to the global area, and more than one room exists in the workspace. A special menu system on applicable controllers allows the end user to select a room from the room selection menu, select a source from the source selection menu, and power the room or system down with the power menu.

Specialized tags are used to program the commands above:

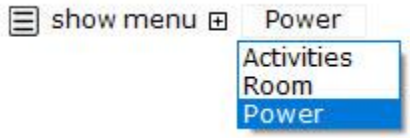
Current Room Variable: **System: Selected Room** tag will program feedback from the Selected Room Variable in the System Manager. This variable will also report the current room on local room controllers.

Select Room Menu: System: Show Room List tag will program a command to show the room selection menu and create a reversed state on the room menu button when that list is visible.

Select Activity Menu: System: **Show Menu** tag will program a command to show the activity menu and create a reversed state on the activity menu button when that list is visible.

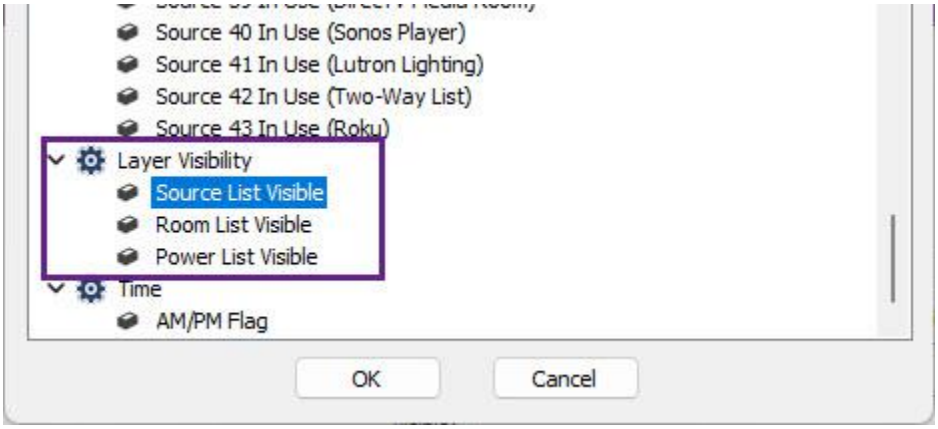
Select Power Menu: System: Show Power Options tag will program a command to show the power menu and create a reversed state on the power menu button when that list is visible.

Use the **show menu** macro step to program the three menus to display without standard tags.



When a menu is **selected**, the other menus are **hidden** from view on the interface.

Select the **system manager** in the variable drop-down menu to program **variables** for the menu using **layer visibility** variables.



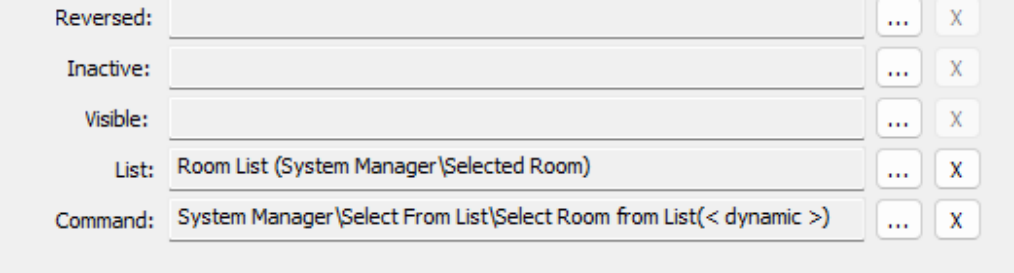
Select the variable drop-down menu to program the current room-selected variable, then select the **system manager**. Use the "**Selected Room Name**" variable to provide feedback for the current room on **global** and **local** controllers.

List Menus

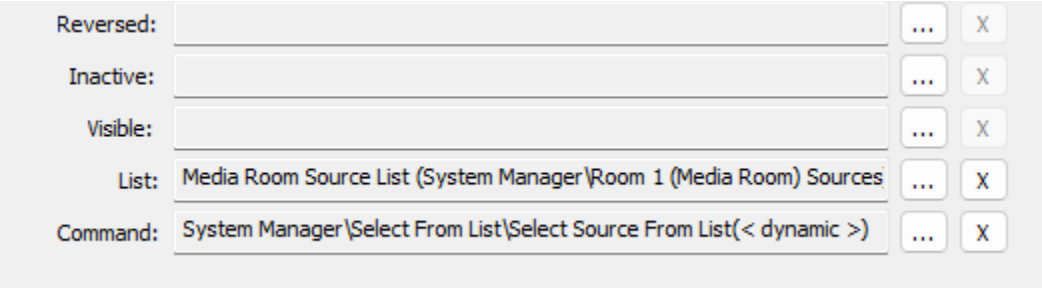
A **list menu** in Integration Designer is the easiest way to build room, source, and power selection menus. Button menus require more work but should be used when user interfaces require custom buttons.

There are **three menu selection types** when using the list menu option:

Room Menu: The **System: Room List** tag will auto-program the item list to display the selection of rooms available in the project for global controllers.



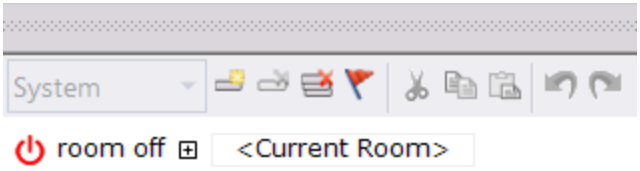
Source Menu: The **System: Menu** tag will auto-program the item list to display the activities available for the current room selection.



The Media Room list is shown above. Each room list is programmed with a room variable and displayed when that room is selected.

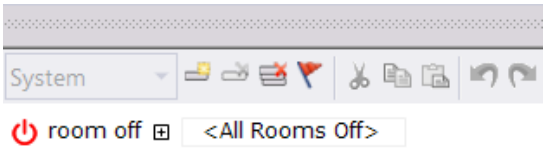
Power Menu: The **Power Menu** is not in list form. The following buttons are available on the power menu:

Turn Off AV (Room): The tag is **System: Room Off**, programmed with the **Room Off {Room Name}** macro step.



Turn Off Lights (Room): The tag is named **All Lights Off** and is programmed by the RTI Installer.

Turn Off AV (Whole House): The tag is **Power Off All Rooms**, programmed with the **Room Off (All Rooms Off}** macro step.

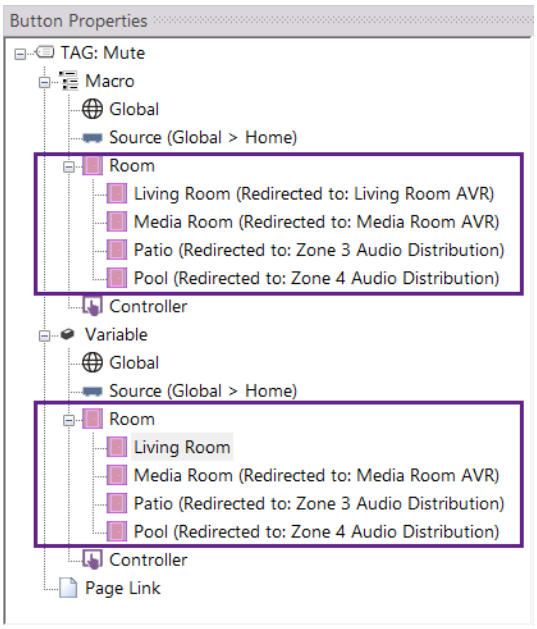


Turn Off Lights (Whole House): The tag is **All Lights Off All Rooms** and is programmed by the RTI Installer.

GLOBAL CONTROLLER ROOM MACROS

The room selection feature available in the System Manager can track the room selected and run the room-level macro that corresponds to that room without needing multiple buttons or pages.

This feature is only available on multiroom controllers located in the global area. Buttons and feedback located on a global source device will have as many room macros as there are rooms and use tags to point at any room macro with the same tag in a local room.



The mute button on the global controller will display the **macro** and **variable** since the mute tag is programmed in each of the four rooms.

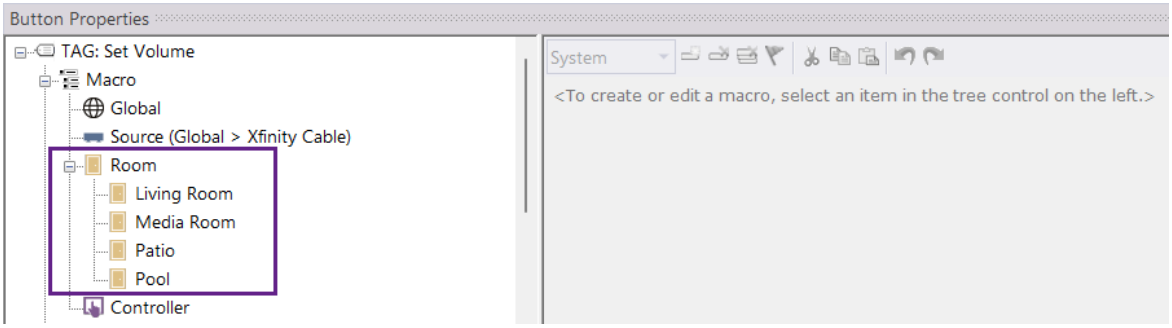
If any rooms had a missing **macro**, **variable**, or **both**, the RTI Installer could program the tag in the local room or on the global controller.

Programming a **room macro** or **variable** on the **global** controller will run the macro or display the feedback for the **current room** selected.

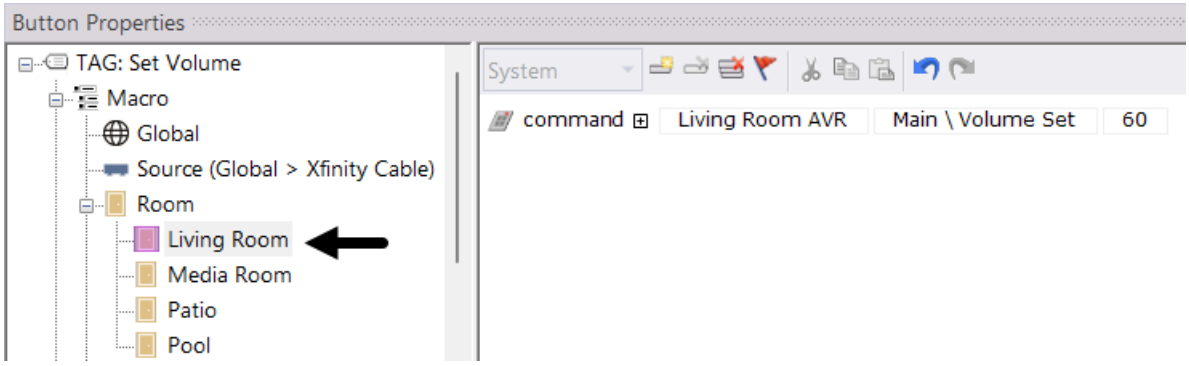
Programming Room Macros on a Global Controller

1. Create a **button** on a **global source** device page. Tag the button **Set Volume**. This button will set the ideal volume level in each room.

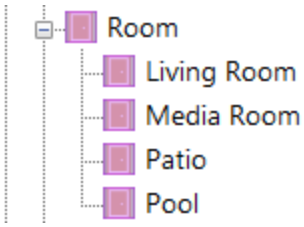
2. Since there are four rooms in this example, up to four room macros may be programmed.



3. Select the **Living Room** and program a room macro to set the ideal volume level.



4. Continue to each room and program the room macro to set the ideal volume level. If the command or driver is for a room, you may skip it.



A square purple ring appears around the room icons with a programmed macro.

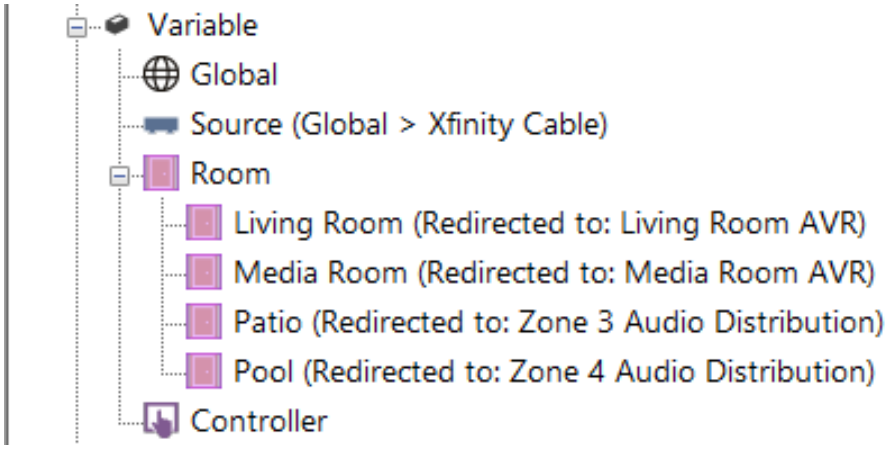
When the Living Room is selected, the room macro for the Living Room will run. When any other room is selected, the corresponding room macro will run.

Programming Room Variables on a Global Controller

Like room macros, source device pages on a global controller have an added feature to provide two-way feedback based on a room selected. Variables can be programmed for each room by programming a variable on the room macro in a local room or on the global source page.

1. **Create** a **gauge** above the set volume button created in the previous step.

2. **Tag** the gauge with the standard **Volume** tag. The variable values will auto-populate since **Volume** is a standard tag and a device exists in each room.



3. Fill in the values if any rooms are missing and a driver is available by programming the room variable. If a non-standard tag name is used, program the variables in the local room or the global source page.

Pro Tip: Take advantage of multiple room macros and room variables available on the global controller for source devices created in the global area. Buttons and feedback that change based on the room selected can save work and reduce the number of pages required to complete a project. Doing so will ensure the user interface is running in the most efficient way possible.

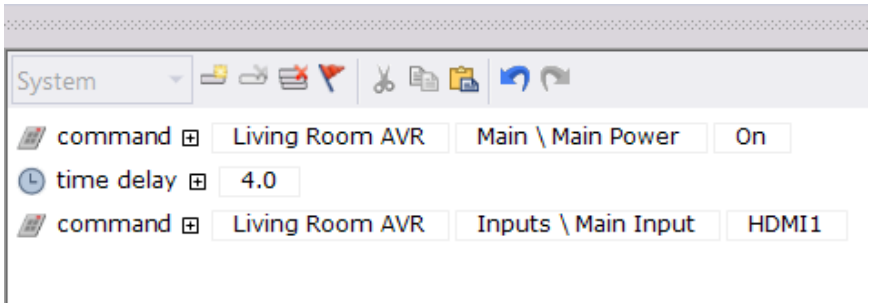
PROGRAMMING SYSTEM VARIABLE TESTS

One of the biggest advantages of **two-way drivers** is the ability to **poll** devices for their status. **Variables** such as **power**, **input**, **sound program**, and **preset** are just some examples that can be **tracked**. In addition to tracking the current statuses, **system variable tests** can be used to base **if/else** statements on the variable returned from the driver.

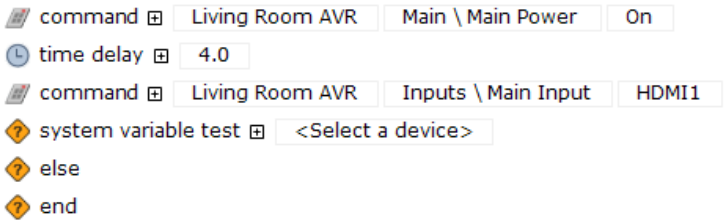
Scenario 1 – Avoid sending needless delays.

The AVR requires a 4-second delay when it turns on before input changes can be made. The delay should only be sent when the AVR is off.

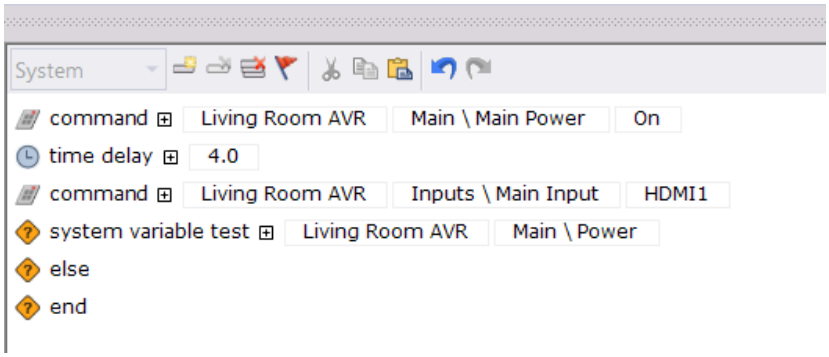
1. This macro is not conducive to the client experience since the delay is sent even when the device is on.



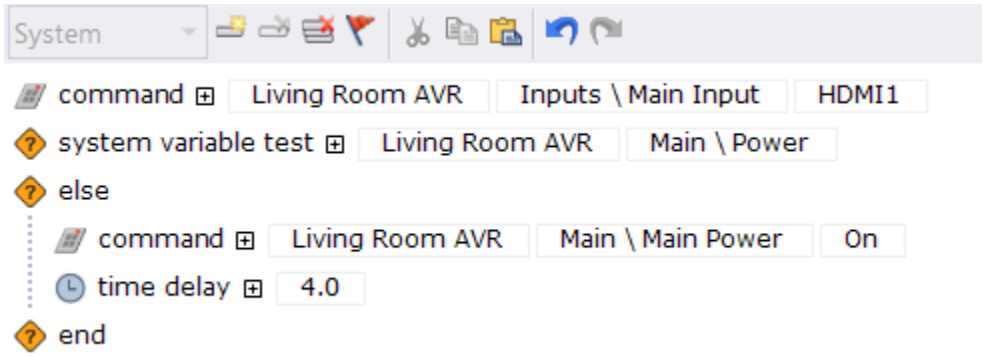
2. Insert a **system variable test** from the **macro steps**.



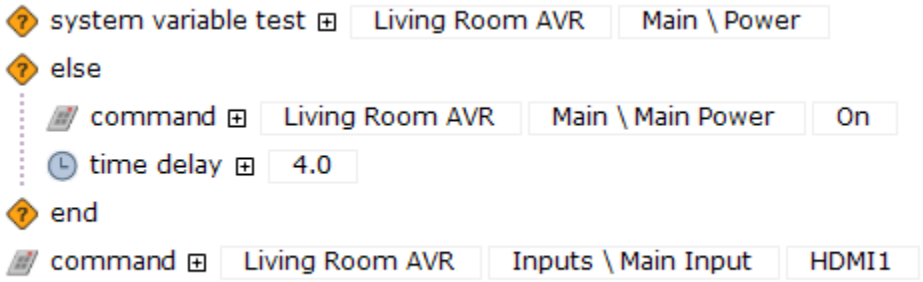
3. Select the device and variable to test the status. In this example, the receiver power is evaluated. If it is true, the receiver is on. If it is false, the receiver is off.



4. Do nothing inside the if part of the statement when the power is true (on). If the power is false (off), turn on the AVR and send the delay. Drag the on command by holding down the mouse and time delay to the “else” side of the system variable test.



5. Holding the mouse, drag the system variable test to appear first in the macro procedure.



Scenario 2: Page Link

A button exists on the global controller to jump to a selected room's lighting page. While this can be done by creating different page links on each room macro, the RTI Installer wants a single button to jump to a specific room's lighting page. The button should jump to a general lighting page if no lighting page is found for that room.

1. Use a system manager system variable test to test if a room is selected and jump to the respective lighting page.

While this macro will work, there is no way to jump to a general lighting page if a room other than the Living Room, Media Room, or Patio is selected.

Nesting system variable tests within a macro are possible with Integration Designer. By nesting these system variable tests, each test is evaluated inside the "else" part of the statement.

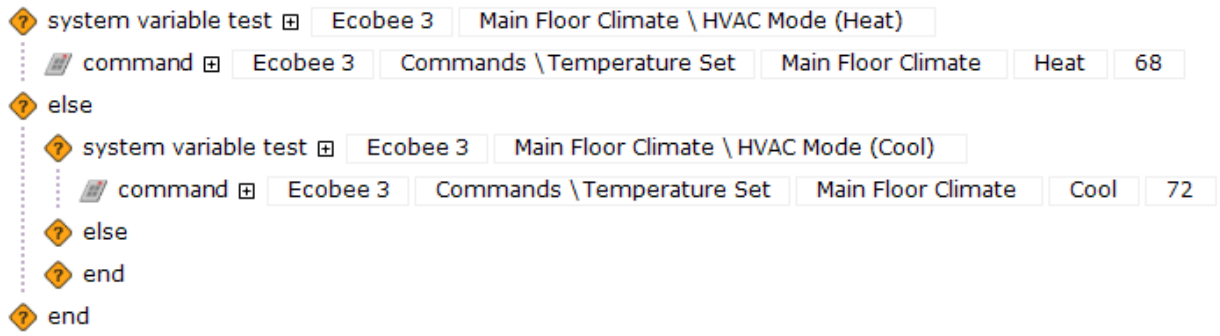
Nesting can evaluate multiple system variables and choose a different path for as many needed variables as possible.

If the **Living Room** is **selected**, it jumps to the **Living Room** lighting page. **Else**, if the Living Room is **not** selected, it checks to see if the **Media Room** is **selected**. If it is, it will **jump** to the **Media Room** lighting page. If it is **not** (**else**), it will check to see if the **Patio** is **selected**. If the Patio is **selected**, it will jump to the **Patio** lighting page. If the Patio is **not** selected, no rooms meet the criteria, and the controller jumps to the main lighting page.

Other Examples

The example below will evaluate if a TV and AVR are powered on using a nested system variable test.

This system variable test will poll the thermostat and set two different temperature setpoints based on the heat or cool mode.



In this example, when the end user turns on the garage light, the kitchen ceiling light will turn on at 75% to welcome them into the home.

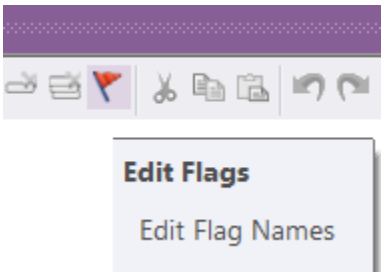
Each of these three rooms was tested in this system variable test example to see if the current source is RTI Music. It will switch the three zones to the RTI Music source activity if RTI Music is not the current source selection.

FLAG PROGRAMMING

When system variables are not available, flag programming may be used. Flags are Boolean indicators set and cleared by the RTI Installer in Integration Designer. Once set within a macro, the flag state can be tested with an if/else statement. While flags are not as reliable as system variable tests, they can be effective when used properly.

Three steps are required for **flag** programming:

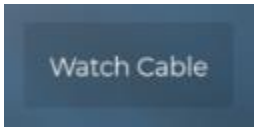
- Define the flag by selecting the **Edit Flags** icon in the macro window toolbar.

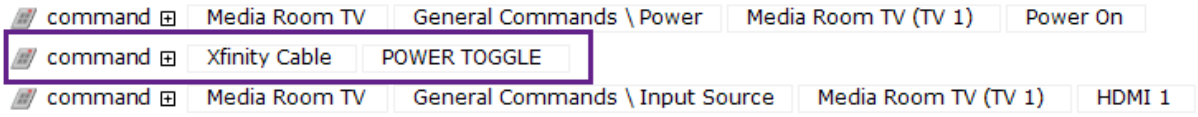


- Test the flag status using the **If Flag Set** macro step.

- Set, clear, or toggle the flag state using the **Flag** macro step.

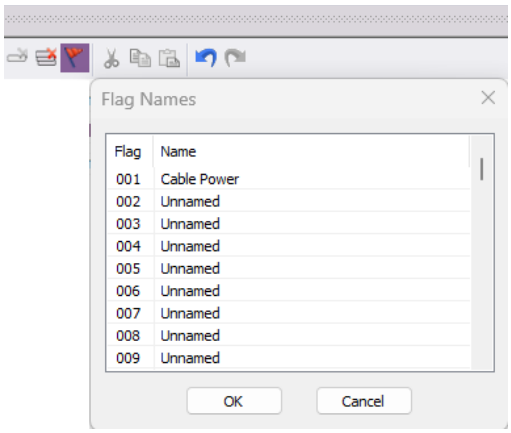
1. A Watch Cable activity button requires programming. A TV and cable box must be turned on, and the TV must switch to the correct input.





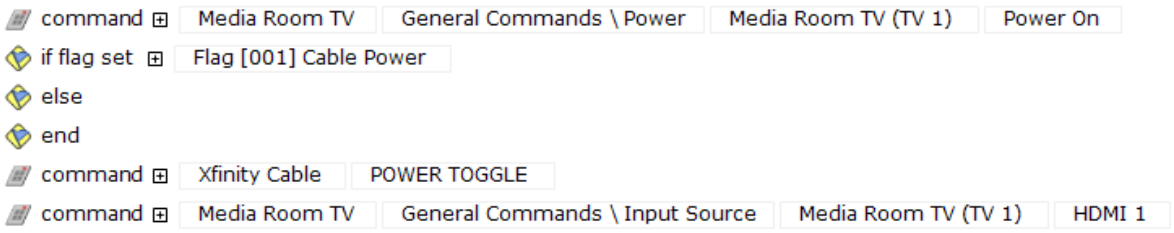
The issue with the above macro is that there are no discrete power codes for the cable, and only a toggle exists. This macro will work when the system is off, but the cable will turn off if the client presses the Watch Cable button again. Flags may be used to track the button presses and the cable power status.

2. Select the red flag icon in the macro window toolbar. Two hundred fifty-six flags are available on the control processor. Select the first flag and name it “Cable Power.” Select OK to continue.



3. Drag or double-click the If Flag Set macro step to bring it into the macro window. It will default to the list's first flag, Cable Power.

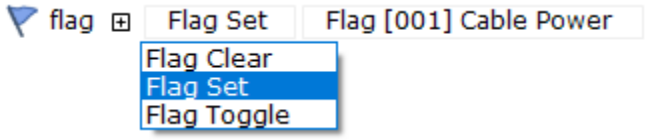
4. Drag the If Flag Set statement above the cable box power command.



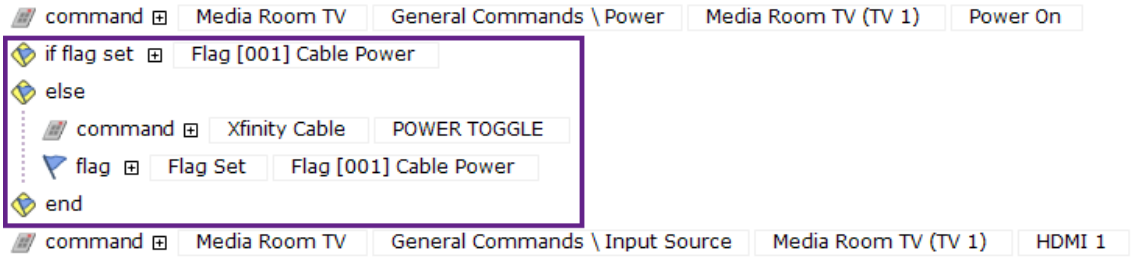
The first time this macro runs, it will check to see if the cable power flag is set (meaning it is on.) It will turn it on with the power command if it is not set (meaning it is off).

It is up to the RTI Installer to set the flag within the flag test after sending the power toggle command. If the flag is not set, the next time it runs, it will send the power toggle command again, turning off the cable box and confusing the end user.

5. Drag or double-click the **Flag** macro step and bring it into the macro after sending the cable box power command. Flags may be set, cleared, or toggled from one state to another.



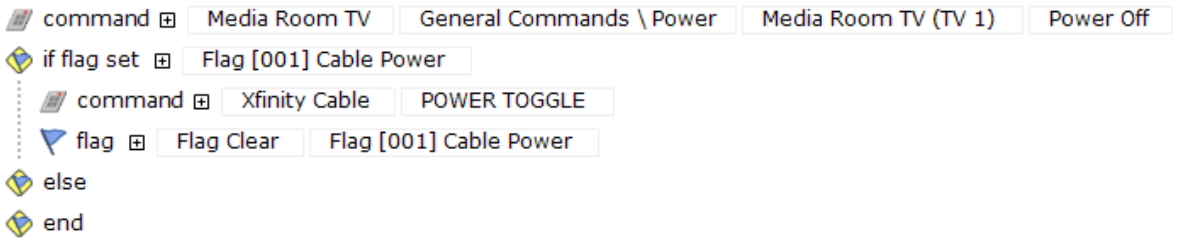
When using flags for power tracking, a **set** state implies the device is **on**. **Cleared** implies the device is **off**.



The first time this button is pressed, the macro will check to see if a flag was set, indicating the cable device is on. Since it was not set, it will send a power toggle command to turn on the cable box. The flag is then set, indicating the device is on.

If the button is pressed again, the flag is set, and the macro will skip the power toggle routine within the if/else statement. The flag must be cleared when the cable box is turned off.

1. A Power Off Media Room button needs to turn off the devices in the room, including a cable box. Since flags are designed to track button presses, the power toggle to turn off the device must only be sent if the cable box is on.



In this example, the end user presses the power off button. The flag status is checked, and the cable box is on if set. A power toggle command will turn off the box, and the flag is cleared. If the end user selects the power off command again, the power toggle will not be sent because the flag is cleared, indicating the cable box is off.

Pro Tip: Using flags to track power on devices that lack discrete power codes is not as reliable as system variable tests. Power outages, turning on or off the device without the control system, or any other interruption will cause the flag status to not match the power condition. When using flags for power, be sure to instruct the client to press the provided power button on the user interface should the box turn off when it should turn on or turn on when it should turn off.

Case Uses of Flags

- Pop-up menus using layers.
- Indicate an event has been triggered.
- Page links based on user.
- Display buttons on a user interface.
- Password User Interfaces.
- Tracking input, power, or presets.
- And many more.

ADVANCED LAYER PROGRAMMING

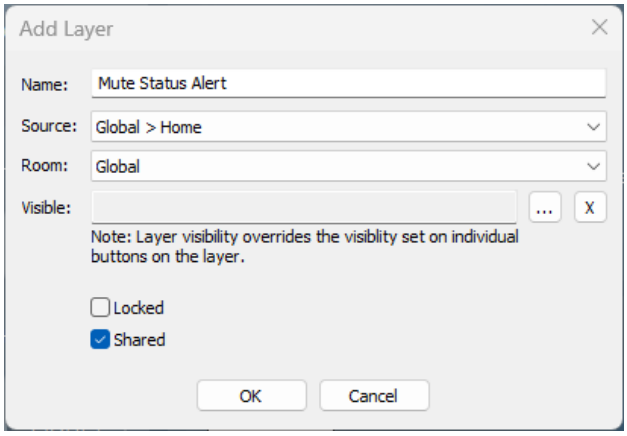
Layers can deliver powerful features in the user interface but also make it easier to manage different areas of the screen separated by functionality. Some projects require **pop-up status messages**, **alerts**, and **menu items** that can be made to deliver a series of zone volume sliders.

Programming a Mute Alert

Layers can be displayed based on **system variable statuses**, **processor flags**, and **internal variables**. In this example, a simple status alert when the system is muted can avoid client confusion, leading to service calls.

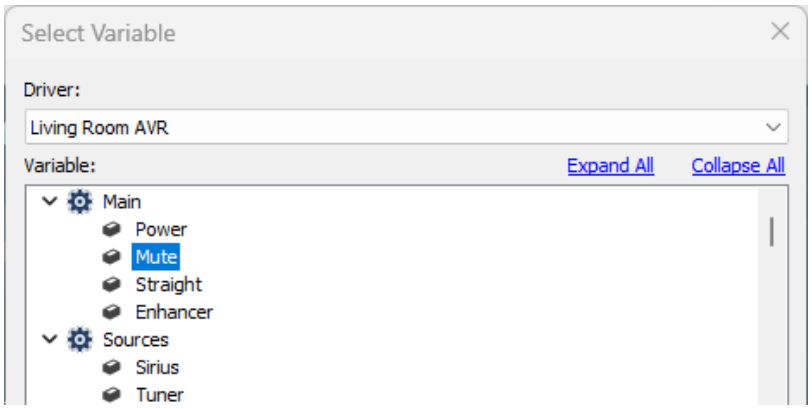
1. **Create a layer** by selecting the **Add a New Blank Layer** in the layer toolbar.

2. Provide a descriptive **name** for the layer and mark it **shared** if it needs to be used on other pages in the project.



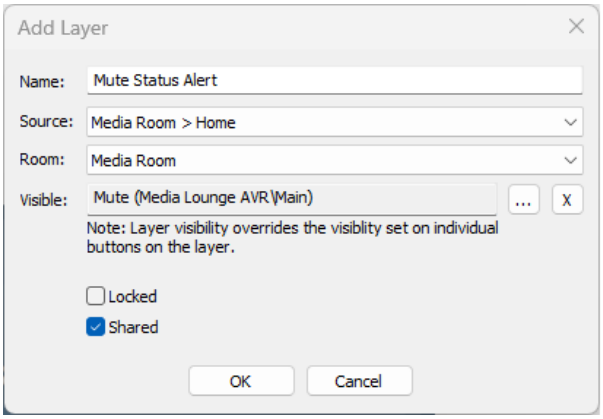
3. Select the ellipsis menu next to the **Visible** field since this layer will be **visible** when the audio device reports a true mute status.

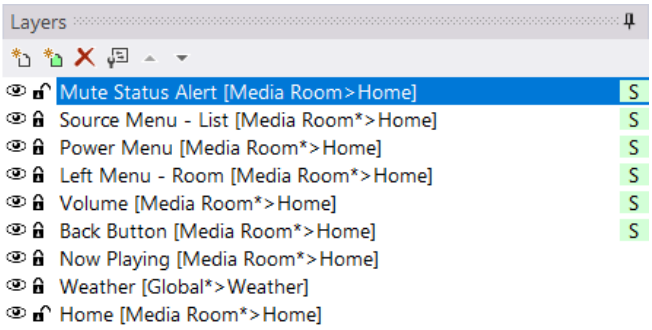
4. In the **dropdown menu**, select the **driver** and **variable** that would require a **true** status to display the **layer**. In this example, select the AVR in the room and the **Mute variable** in the list.



Selecting the **Mute** variable will make the layer visible when the variable for mute is true.

5. Select OK to apply the changes. The new layer will appear at the top of the list so no other layer contents can overlap.



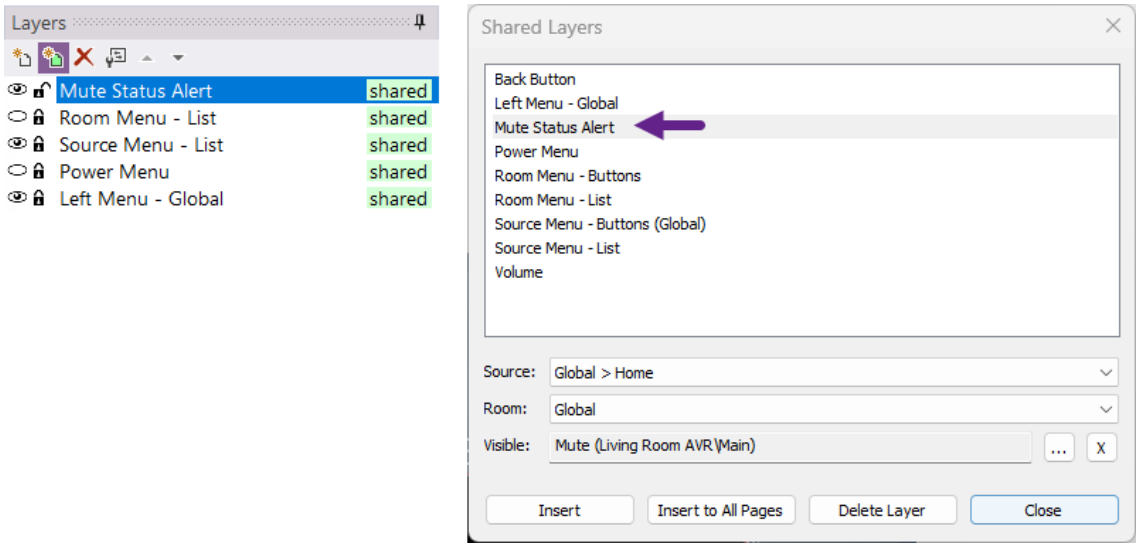


6. Select the new layer in the layer window and **create** the mute message alert. You can **hide** other layers if they are distracting or **lock** them to ensure they are not impacted.

Center the message on the screen and ensure the alert is easily visible to the end user. If the status alert does not have a command, make the button inactive in the properties.

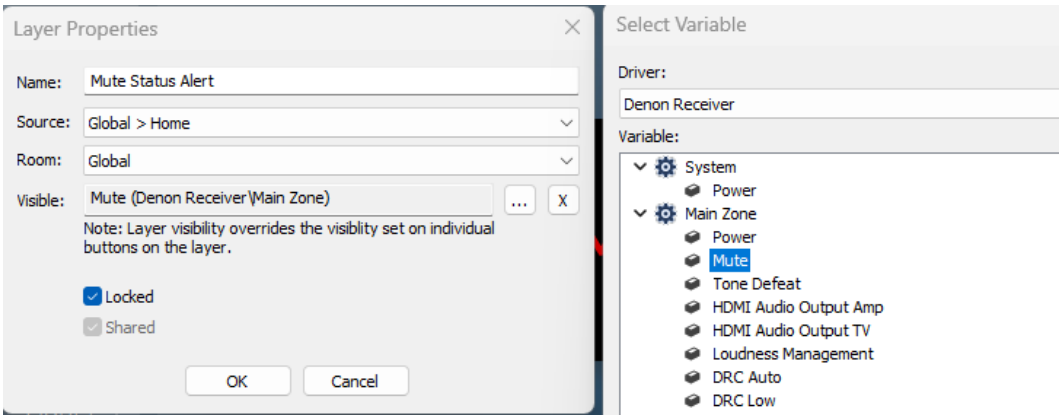
7. Go to the next page

on the controller where the mute layer is needed. Select the "**Insert a Shared Layer"** green icon in the layer window and insert the layer to the page or all pages.



Before inserting a shared layer, the variable can be changed for that layer without impacting the existing layers. If inserted into all pages, visible states may be changed without impacting other layers, making it possible to use the same layer for other rooms with separate variables.

8. Make any changes to the visible state if other devices control the volume commands.

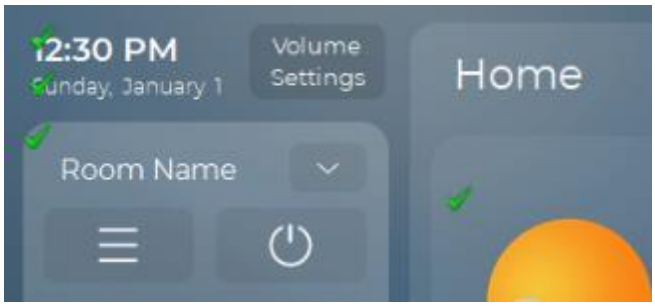


Pro Tip: If the project contains many pages, consider making separate layers based on the room's mute variable. Ensure status alerts are on the top of the layer list so other buttons on other layers won't block them from view.

Programming a User-Selectable Menu

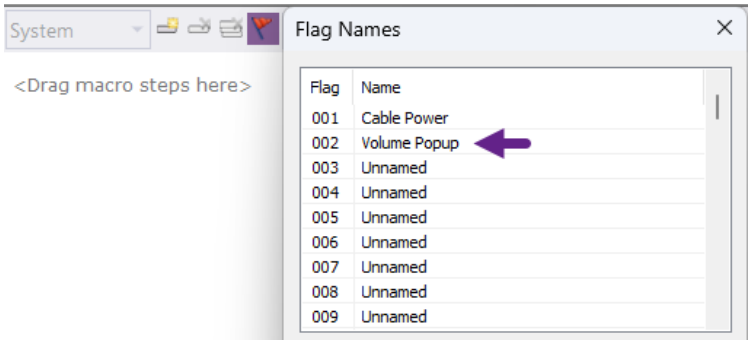
There may be cases where an end user invokes the layer visibility manually—for example, by pressing a button to display a menu of items. Instead of using a variable, flags could be used to display layers when needed.

1. Create a button on a shared existing or new layer. Label the button “Volume Settings.” Ensure the screen position of the button is available on all pages.

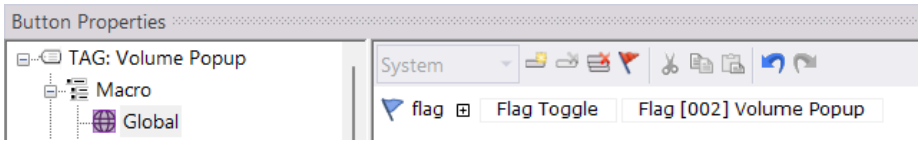


2. Tag the button “**Volume Popup**” in the **tag window**.

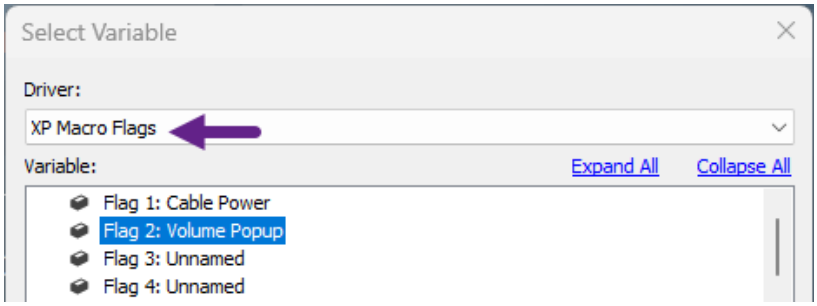
3. Select a **global** macro in the **button properties**. In the **macro window**, select the flag icon. Name an available flag “KX7s Volume Popup.”



4. Once the tag name has been defined, create a **macro command** to **toggle** the **flag** status so that when the button is pressed, it will toggle between viewing and hiding the volume menu.

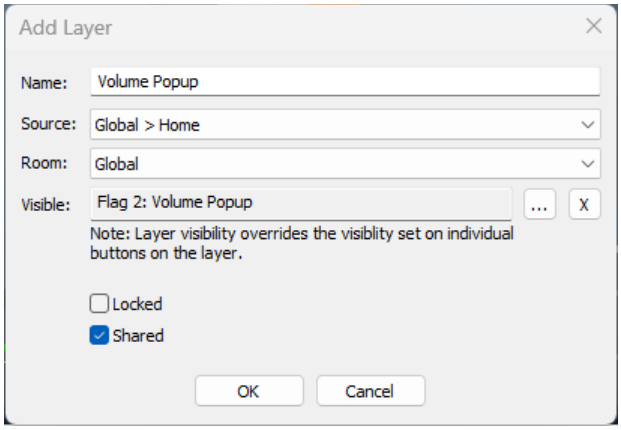


5. On the same button, program a reversed state variable by selecting the global variable and selecting the ellipsis menu for the reversed state. Select **XP Macro Flags** from the dropdown list, then select the **Volume Popup flag**.



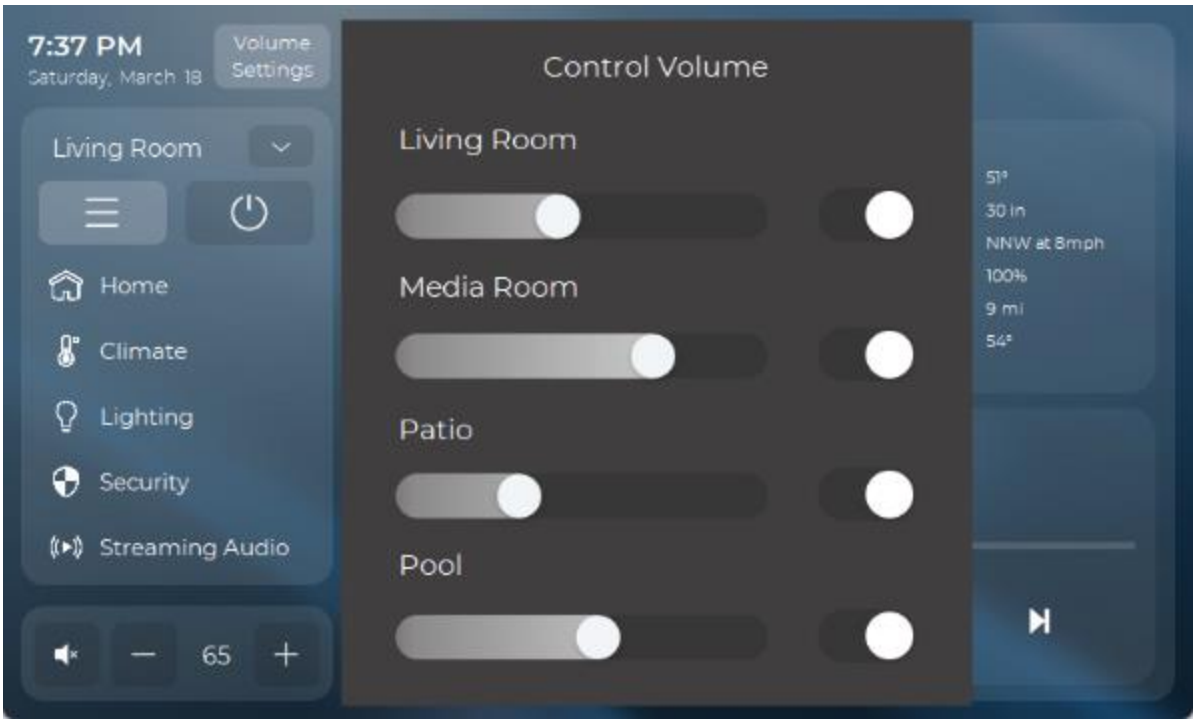
This variable will force the reverse state on the button when the flag status is set.

6. Once the button is programmed, create a shared layer for the volume commands. Select the **XP Macro Flags** from the dropdown menu for the visible state, then select the **flag** to display the layer when it is set.



7. Build the items on the Volume Popup layer to control the volume levels.

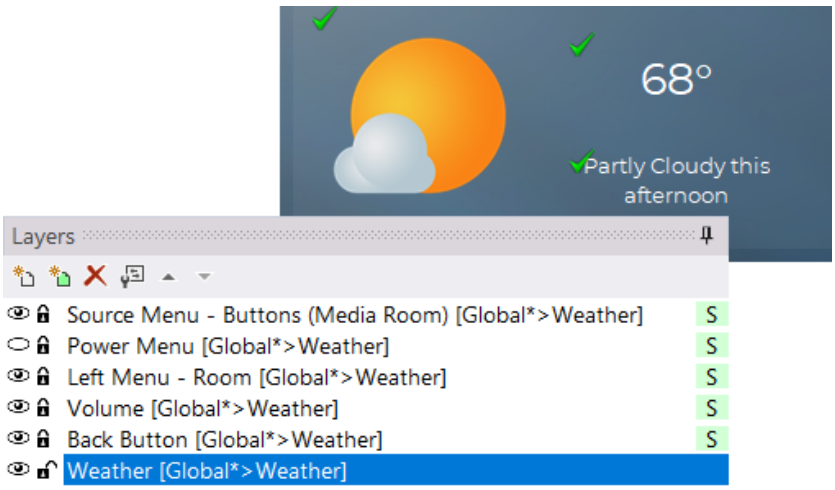
   In this example, there is a volume slider and a power toggle for each zone in the project. The end user can access the volume by pressing the Volume Settings button. A flag is set, which displays the layer for the volume. Once the button is pressed again, the flag will clear and hide the volume menu on the interface.



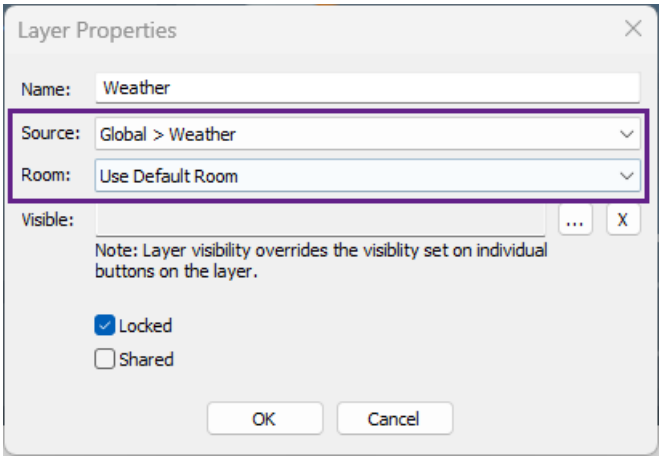
Room and Source Layer Programming

Layers can be pointed at any source device from any room in the project. When programming a layer, you can use standard tags for any source device if the layer room and source are set to that device.

The weather driver is added to every project, and a weather dashboard is included on larger controllers. Standard weather tags are used on the layer and pointed to the weather source, which will auto-populate the feedback on the layer.



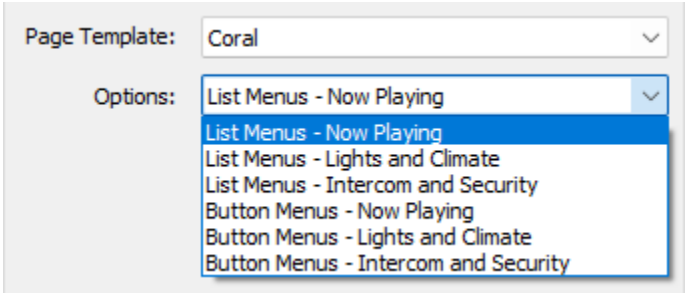
In this example, the tags will reference the weather source by setting the weather layer to the weather source in the layer properties.



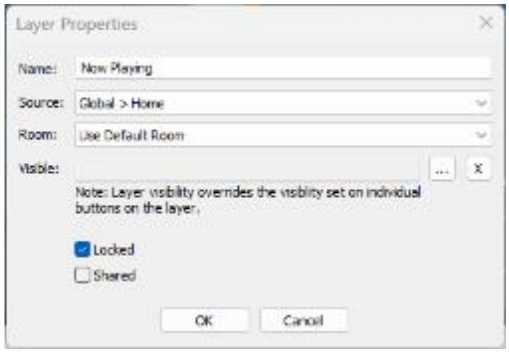
Since the source and room are linked to the weather source, all tags on the layer will reference the weather source.

The **tag lookup process** occurs on the weather source page instead of where the layer originates.

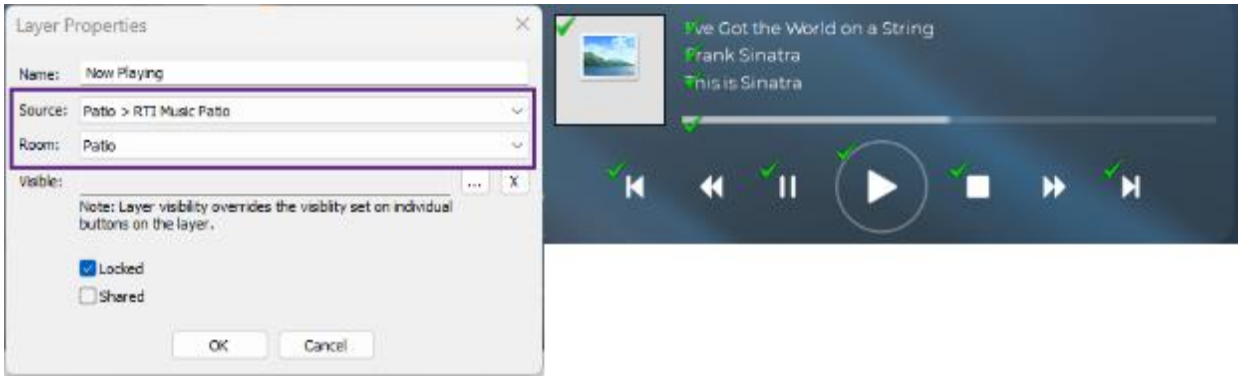
When adding a controller and selecting a **Now Playing** menu type, the layer can be pointed at any source for auto-programming the existing tags.



The **Now Playing** layer is not programmed because it uses the home source for tag lookup. The buttons and feedback will **auto-program** by selecting an appropriate **source** and **room** for the layer.



Set the **Source** to the RTI Music Player from the Patio. Select the **room** where the source is located.



Select **OK**, and the layer will auto-populate the commands and feedback using the RTI Music source from the patio.

Pro Tip: Linking layers to different source devices allows the RTI installer to create duplicated tags on the same source page. They will have different commands if the layer they reside on is linked to another source.

SYSTEM MANAGER PROGRAMMING

Integration Designer's System Manager built-in intelligence offers a time-saving, centralized, and segmented approach to programming. It also includes various tracking variables that can be used for layers, user interface reporting, and creating bullet-proof macros.

System Manager Variables

Room Name: Provide text-based feedback on the room's name as shown in the workspace. The only way to change how this variable is displayed on the user interface is to change the room's name in the workspace.

Text Variable	Reversed State	Inactive Variable	System Variable Test
YES	YES	YES	YES

Room Source Return Visible: Indicates the end user jumped from a source designated in the system manager as an audio or video source to an “other” source type. It primarily displays a back button layer to return to the audio or video source when sent to an “other” source type.

Text Variable	Reversed State	Inactive Variable	System Variable Test
YES	YES	YES	YES

Room Source List Visible- Feedback to determine if a source list is currently displaying on the user interface for a specific room using the **show menu** macro step.

Text Variable	Reversed State	Inactive Variable	System Variable Test
YES	YES	YES	YES

Room Selected Source- Display the current audio, video, or other source selected in the room.

Text Variable	Reversed State	Inactive Variable	System Variable Test
YES			

Room Selected Source Index- Text-based feedback displays the selected source index. The index starts at zero for the first source selected in a room and can be remapped in the string index.

Text Variable	Reversed State	Inactive Variable	System Variable Test
YES			

Room On- Once audio or video is initiated, the room will report an on status.

Text Variable	Reversed State	Inactive Variable	System Variable Test
YES	YES	YES	YES

Room Off- Once powered off, the room will report an off status.

Text Variable	Reversed State	Inactive Variable	System Variable Test
YES	YES	YES	YES

Room Audio On- Once the system manager "**audio on"** selection event runs, audio is on in that room.

Text Variable	Reversed State	Inactive Variable	System Variable Test
YES	**YES**	**YES**	**YES**

Room Audio Off- Once the system manager "**audio off"** deselection event runs, audio is off in that room.

Text Variable	Reversed State	Inactive Variable	System Variable Test
YES	**YES**	**YES**	**YES**

Room Video On- Once the system manager's "**video on**" selection event runs, the video is on in that room.

Text Variable	Reversed State	Inactive Variable	System Variable Test
YES	**YES**	**YES**	**YES**

Room Video Off- Once the system manager's "**video off**" deselection event runs, the video is off in that room.

Text Variable	Reversed State	Inactive Variable	System Variable Test
YES	**YES**	**YES**	**YES**

Room State- Feedback that reports if a room is currently off or utilizing an audio or video source. Note that this variable does not report when an "other" source type is active.

Text Variable	Reversed State	Inactive Variable	System Variable Test
YES			

Room Source Name- Feedback reporting the source's name in a room as named in the workspace.

Text Variable	Reversed State	Inactive Variable	System Variable Test

YES			

Room Source Selected- Feedback for when audio, video, or other source type is selected in a room.

Text Variable	Reversed State	Inactive Variable	System Variable Test
YES	**YES**	**YES**	**YES**

Room Source Not Selected- Feedback when an audio, video, or other source type is not selected in a room.

Text Variable	Reversed State	Inactive Variable	System Variable Test
YES	**YES**	**YES**	**YES**

Selected Room Name- Feedback indicates the name of the current room selected in a room menu.

Text Variable	Reversed State	Inactive Variable	System Variable Test
YES			

Selected Room Number- Feedback indicates the index of the current room selected in a room menu. The index can be remapped without changing the room's name in the workspace. The global area starts at zero, and the rest of the rooms follow in numerical order.

Text Variable	Reversed State	Inactive Variable	System Variable Test
YES			

Selected Room is Room Name- Feedback that indicates a specific room is selected.

Text Variable	Reversed State	Inactive Variable	System Variable Test
YES	**YES**	**YES**	**YES**

Selected Room is Not Room Name- Feedback that indicates a specific room is not selected.

Text Variable	Reversed State	Inactive Variable	System Variable Test
YES	YES	YES	YES

Source # Name- Feedback for the source name on a project level as named in the workspace.

Text Variable	Reversed State	Inactive Variable	System Variable Test
YES			

Source # Name In Use- Indicates an audio or video source name currently used on a project level. In-use variables do not apply to source types of “other.”

Text Variable	Reversed State	Inactive Variable	System Variable Test
YES	YES	YES	YES

Source List Visible- Feedback indicating a source list is visible on the current controller using the show menu macro step. A single list variable may be shown at one time.

Text Variable	Reversed State	Inactive Variable	System Variable Test
YES	YES	YES	YES

Room List Visible- Feedback indicating a room list is visible on the current controller using the show menu macro step. A single list variable may be shown at one time.

Text Variable	Reversed State	Inactive Variable	System Variable Test
YES	YES	YES	YES

Power List Visible- Feedback indicating a power list is visible on the current controller using the show menu macro step. A single list variable may be shown at one time.

Text Variable	Reversed State	Inactive Variable	System Variable Test
YES	YES	YES	YES

Current Time (H: MM AM/PM)- The current time is displayed in (H: MM AM/PM) format.

Text Variable	Reversed State	Inactive Variable	System Variable Test
YES			

Current Time (HH: MM 24 hour)- The current time is displayed in (H: MM 24 hour) format.

Text Variable	Reversed State	Inactive Variable	System Variable Test
YES			

Current Hour (12-hour time)- The current hour is displayed using a 12-hour time convention.

Text Variable	Reversed State	Inactive Variable	System Variable Test
YES			

Current Hour (24-hour time)- the current hour is displayed using a 24-hour time convention.

Text Variable	Reversed State	Inactive Variable	System Variable Test
YES			

Current Minute- the current minutes are displayed.

Text Variable	Reversed State	Inactive Variable	System Variable Test
YES			

Current Hour Tens Digit- The current hour tens digit is displayed.

Text Variable	Reversed State	Inactive Variable	System Variable Test
YES			

Current Hour One's Digit- The current hour one’s digit is displayed.

Text Variable	Reversed State	Inactive Variable	System Variable Test
YES			

Current Minute Tens Digit- The current minute's tens digit is displayed.

Text Variable	Reversed State	Inactive Variable	System Variable Test
YES			

Current Minute One's Digit- The current minute's one digit is displayed.

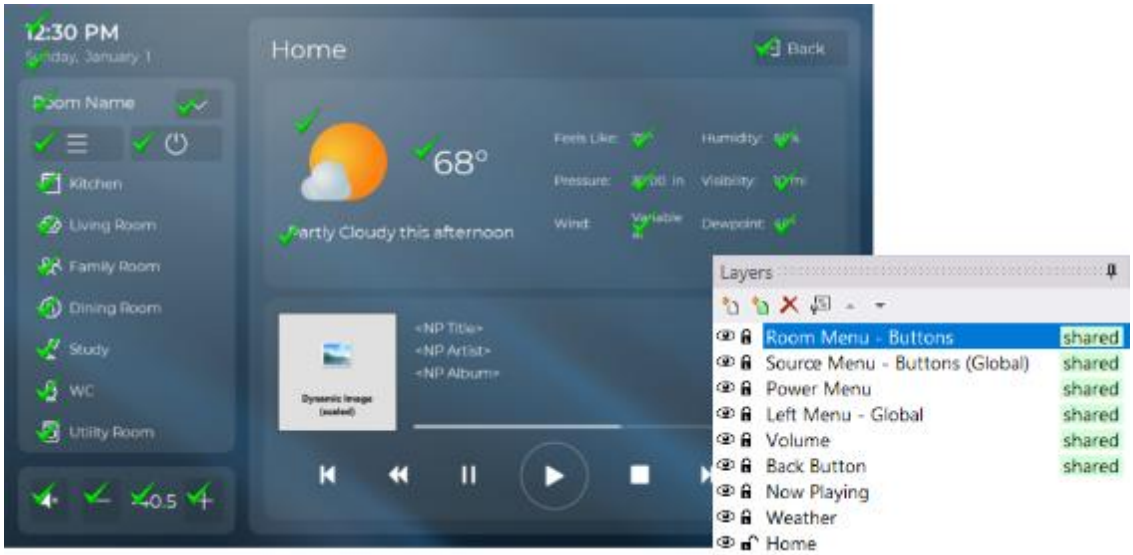
Text Variable	Reversed State	Inactive Variable	System Variable Test
YES			

AM/PM Flag- Feedback indicating if the current time is AM or PM.

Text Variable	Reversed State	Inactive Variable	System Variable Test
YES	**YES**	**YES**	**YES**

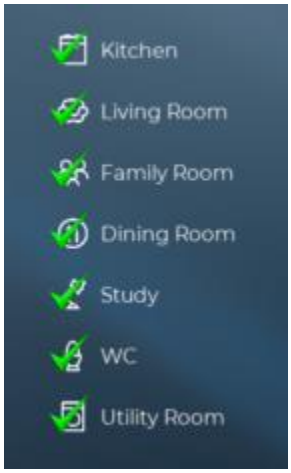
PROGRAMMING MULTI-ROOM BUTTON MENUS

1. Add a KA8 controller to the global area in the workspace.

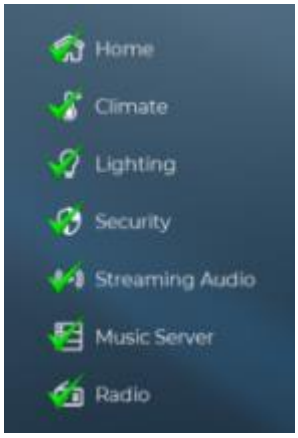


Several layers are created when adding the controller:

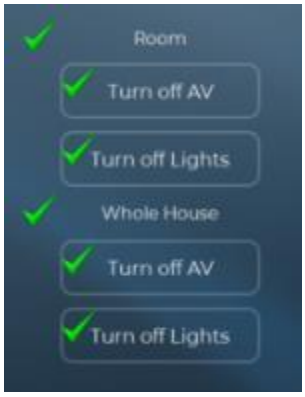
Room Menu – Buttons- A collection of room-based buttons that require revising the icon and text and adding a room selection tag.



Source Menu – Buttons (Global)- A collection of source-based buttons that require revising the icon and text and adding an activity tag.



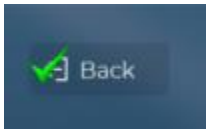
Power Menu- A menu that provides the capability of powering off a local room and lighting as well as all rooms and all lighting.



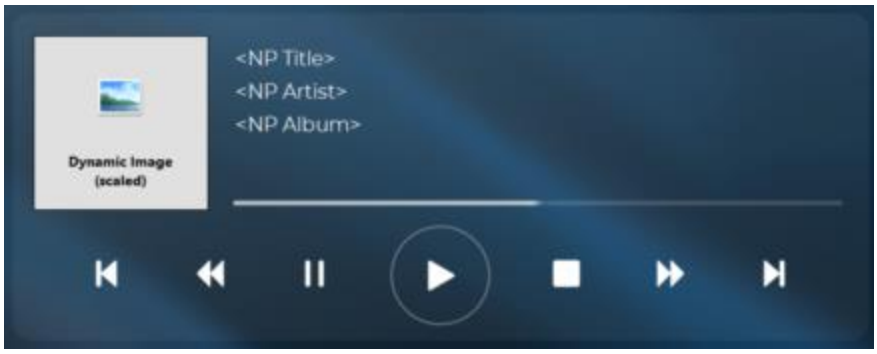
Left Menu – Global- A layer that features support graphics for the button menu lists and the buttons required to display the button menus. It also includes time variables and the current room name variable.

Volume- Volume buttons are included on this layer, as well as volume feedback.

Back Button- A back button on the back layer includes a source return command and visible state feedback on a room level for the source return visible variable. The back button is displayed on sources with a source type of “other.”

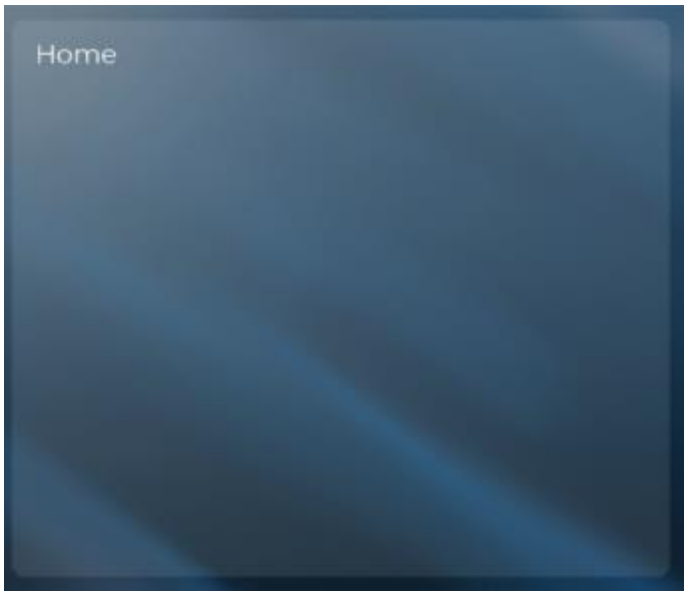


Now Playing- One of the possible dashboards that can link to a music or video source device if feedback is available. Other button menu dashboards are available for lighting, climate, intercom, and security. These can be linked to sources or programmed manually.



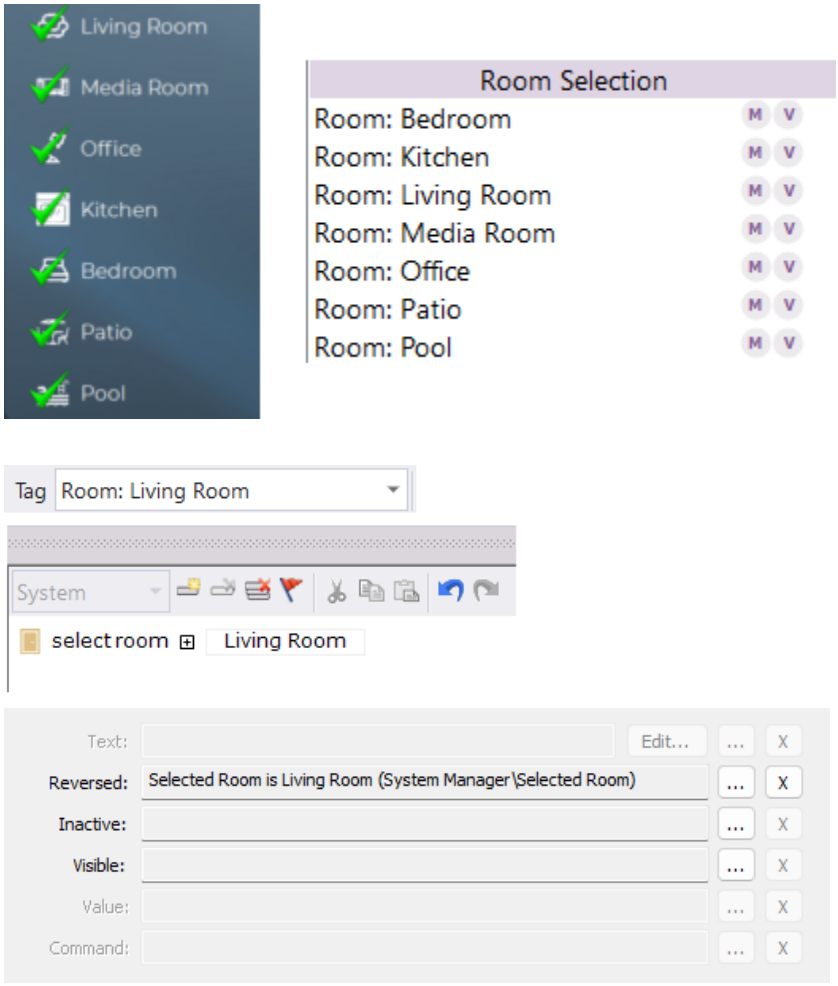
Weather- Included with larger controllers; provides a weather preview on a home page linked to the weather source page and auto-programmed.

Home- The baseline layer for the home page. This page features background graphics and a title for the home page source.



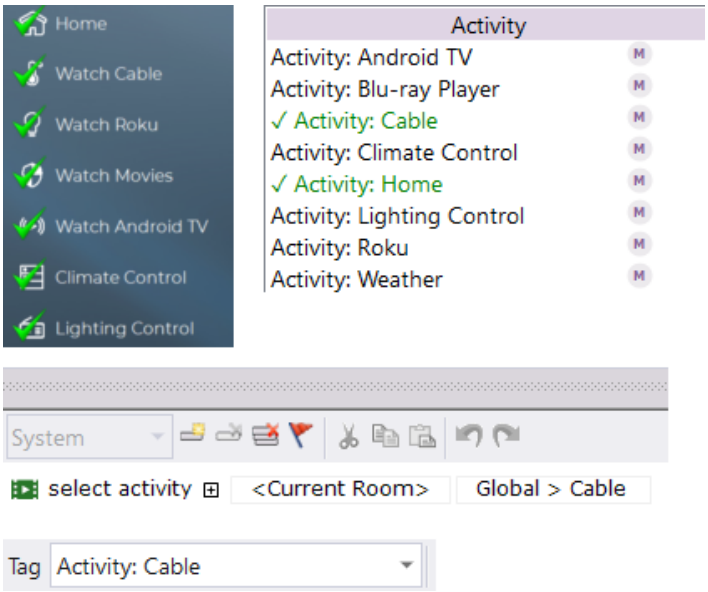
2. Build the **Room Menu – Buttons** menu by dropping graphics onto the buttons from the bitmap library. Once the icons reflect the rooms in the project, change the room names on the button via the text editor.

3. Drag room selection tags from the tags menu found in the tabbed menu. Each tag will contact a macro that will select the room and reflect a reversed state on the button when that room is selected.



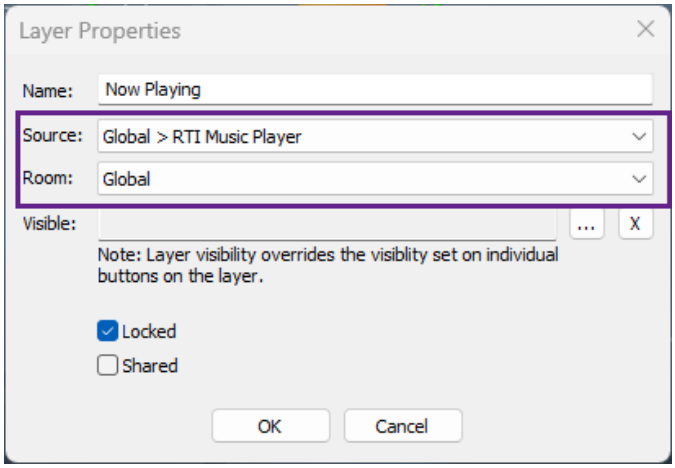
4. For the **Source Menu – Buttons (Global**) menu, revise the source icons and button text to reflect the sources available for each room. If there are different activities in each room, **create a source button menu layer for each room** and set the visible state to the room selected variable.

5. Drag activity tags from the activity tag section of the tags menu. These tags will program macros that tell the System Manager to run the selection macros for the activity selected. If bypassing the System Manager, you can still use these activity tags for manual programming.



Once tags are dropped on the activity icons, the text will turn green in the tag window.

6. Once the button menus have been configured, link the Now Playing, Lighting, Climate, Intercom, and Security dashboard layers to the appropriate source. These items may be manually programmed or customized with other functions if needed.



7. Make custom changes that meet the project's needs, such as text, background, or additional layers.

Pro Tip: Text-based list menus will automatically populate the room and source menu items without the need to customize the interface or designate tags to the buttons. This method is a quicker approach to programming and should be used to save time and effort.

RTI BEST PRACTICES

LEARNING TIPS

Welcome to RTI! Here are some important tips on getting started with RTI company-based resources, communication, and training opportunities.

- All employees programming the software should engage RTIXCEL Learning Portal and complete the certification training. An RTI training kit dedicated to testing and training is recommended for optimal success.

- Start with a basic project when deploying an initial project. Use the quickest path to programming Integration Designer using list menus and template building.

- Stay in touch with RTI by regularly checking the training calendar on the RTI dealer site. Participate in local and webinar training when possible and receive RTI marketing communications for important information.

- Know when to contact technical support, customer service, sales, training, and sales engineering. Customer service coordinates returns, important account information, and administrative items. Technical support should be contacted for on-field issues with hardware or software. Sales should be contacted to discuss how your business can benefit from RTI. Training is contacted with questions on training opportunities and understanding software or product questions. Sales Engineering assists with project design, advanced programming inquiries, and product information.

- For the highest level of support and taking advantage of RTI programs and resources, ensure all company employees have been registered with RTI under the company account and have their dealer website access.

- Take part in RTI dealer forums for additional troubleshooting assistance, product recommendations, and sharing valuable experiences with others. In addition, subscribe to RTI social media to keep up with the latest company announcements and news.

USER INTERFACE DESIGN

RTI Installers can create custom user interfaces and user experiences for their clientele with award-winning Integration Designer software. Here are some basic principles to adhere to when designing an interface for your clients.

- Design the interface in the best interests of the end-users, including all family members who interact with the system.

- Avoid “crowding,” which means minimizing the functionality on the screen. Group functions by type and avoid repeating functions in multiple areas. If the interface is too busy, it loses its appeal.

- When labeling the controller, use verbs to explain the function. Using words like “Yes,” “No,” “OK,” and “Cancel” will confuse the client. Instead, verbs like “Power Off” or “Shut Down.”

- Pick a color scheme that compliments your client’s tastes and décor. In the interview process, steer the conversation towards color selection and note the color themes in the home.

- Small buttons are difficult to press and can frustrate the user experience. As a rule of thumb, buttons should be greater than 64 pixels and have a 1.25 to .75” ratio in size. On smaller touchscreens, make the buttons as large as possible.

- Observe similar rules to building web pages. Stay consistent with color selections and keep the interface as simple as possible. Too many hidden or embedded menus can lead to excessive drilling down to a menu, making the user interface difficult to manage.

- Use buttons with contrasting reverse states and ensure the background images are consistent and blend with the buttons so they may be seen clearly. Some color combinations are aesthetically pleasing, while others are distracting and complicate the user experience.

- Test the interface thoroughly before placing it in the client’s hands while you train them to use the new control system.

PROGRAMMING BEST PRACTICE

When programming macros and user interfaces, observe some basic rules to help enhance the user experience while minimizing issues on the field. Here are some basic principles to adhere to when programming your first project.

- Build a welcome page as a starting point each time the system is used.
- Provide “Please Wait” pages if the macro processing requires time to calibrate the system properly before the client can interact.
- Keep macro programming simple. Eliminate any excess flags, system variable tests, and layers that could complicate the system and make it difficult to troubleshoot. Too many layers and options often mean too many decisions the client needs to make to enjoy the experience.
- Make a list of features that meet the client’s needs and demands. Make sure each item is addressed carefully and only provide what is needed.
- Specify the proper processor for the project. Here are some considerations:
    - Size of the project (Rooms & Sources)
    - Number and Type of Drivers
    - Number of Expansion Devices
    - Number of Controllers and Type
    - Automation such as Lighting, Climate, & Security
    - Future Growth of Project
- Specify controllers that best fit the client profile. Deduce whether the client and users prefer more hard buttons or touchscreen real estate.
- Build macros so they may be rerunnable without causing problems. Keep a simple structure.
- Provide feedback when needed. Consider that certain feedback may be obvious to the client without viewing the controller and only complicates the user interface.

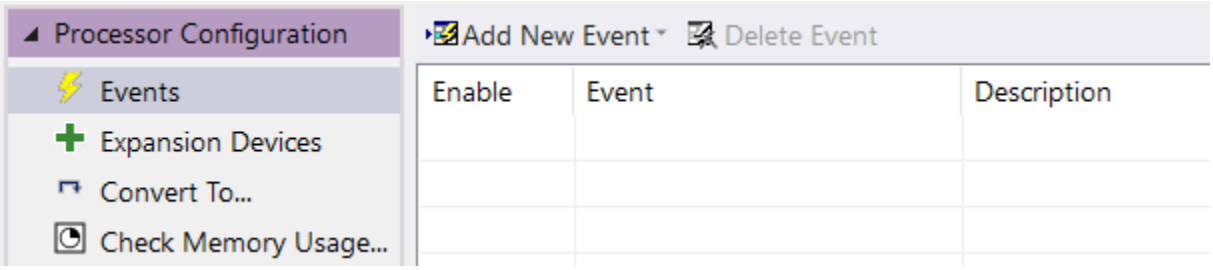
Event Programming

One of the main benefits of an RTI control system is the automation that occurs without interacting with a controller by pressing a button. Commands can be run passively, providing the end user peace of mind, safety, and convenience. Integration Designer offers two different types of event programming based on the type of functionality required.

PROCESSOR EVENTS

RTI control processors capable of event programming have built-in intelligence and features such as an astronomical clock and sense input ports.

Processor events can be accessed and created by selecting the **processor** in the workspace and selecting **Events** in the **processor configuration.**



In the event window, new events may be added by selecting **Add New Event**. Existing events may be deleted by highlighting an event and selecting **Delete Event.** In the grid, events are detailed and may be set to **enabled** or **disabled**. An event is named based on the options selected in the **Event** field. A **description** for each event describing the case use or specific event details may be entered.

There are **four** types of **processor events**: **Sense**, **Periodic**, **Daily**, and **Startup**.

SENSE EVENT

A sensory event can be created with these **prerequisites**:

- Wiring the XP processor sensory input leads to a device capable of outputting between 2-24VDC or using an RTI voltage sensor connected tan an MPIO port of the processor.

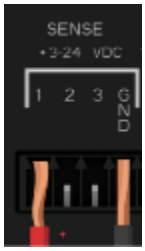
- Identify and name the sensory input port in the processor configuration port settings.

- Create a global, room, or source macro to run when the event is triggered.

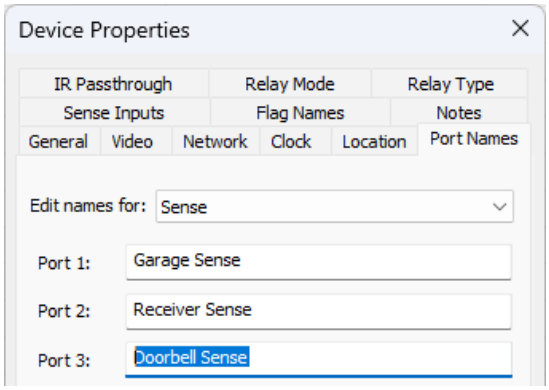
  Sense events are triggered when the sensory input or MPIO sensor detects any voltage in the 2-24VDC range. Two states are possible: a high and a low state. A high state is achieved when power is detected, and a low default state is when no power is detected.

  Scenario: When the doorbell is pressed, the client wants a pop-up warning on all his controllers. He also wants to turn on the Kitchen television and show the front door camera on the television.

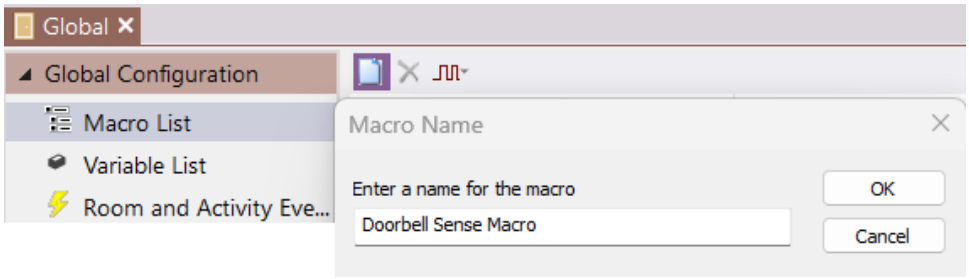
  1. Wire the sense input to the doorbell mechanism.

  2. **Name** the sensory input in the **processor port settings** in the processor configuration window.

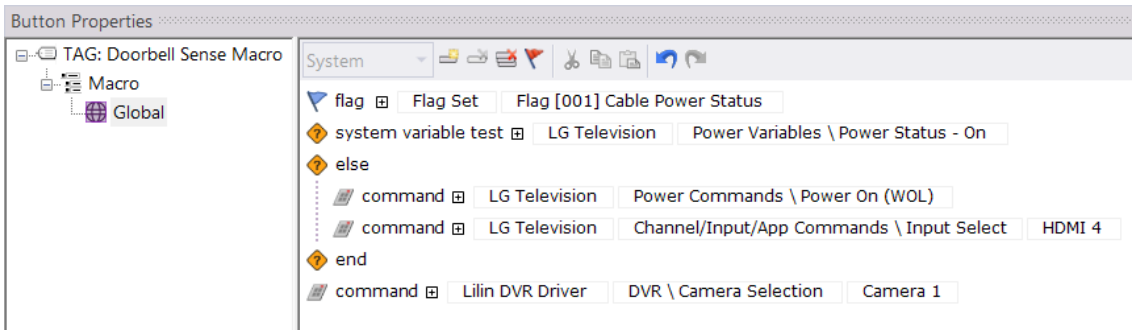
     

3. Greate a **global** macro by selecting the **global area** in the workspace. Select the **Macro List** in the global configuration window and **add a new macro** for the Doorbell.

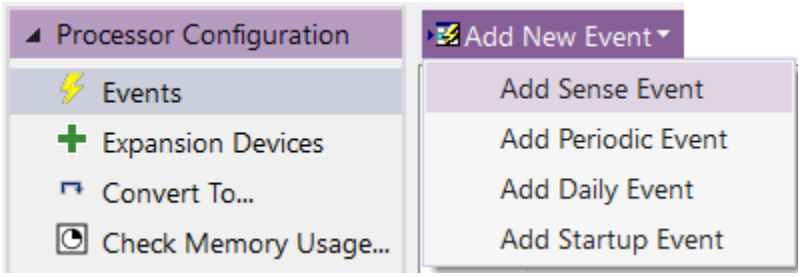


4. Program the macro to set a processor flag to pop up a doorbell alert on the required controllers. Complete the macro by turning on the TV in the kitchen, setting it to the camera input, and selecting the correct camera.

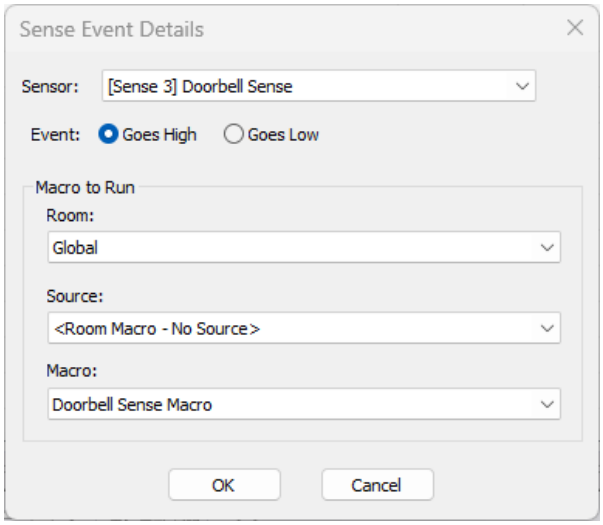
   Set up the layer to alert the end user that the doorbell was pressed. Use the flag set in the macro for the visible state.



5. Go to the Event section of the processor configuration and select Add a New Event – Add Sense Event.



6. Select the settings and macro to run in the **Sense Event Details** window.



Sensor: Select the sensory input or MPIO port from the drop-down menu.

Event: Tick the box for the event to trigger based on a **high** (2-24VDC power) or **low** (no power).

Macro to Run

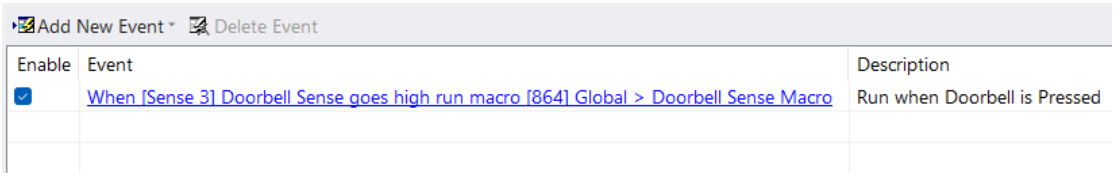
Room: Select the room where the macro was created. Select global if a global system macro was created.

Source: If a source macro was created, select the source it was used to create. Leave this field alone if using a room macro.

Macro: Select the macro (global, room, or source) to run.

In the above example, The sensed event will run when power is detected. A global macro named Doorbell Sense Macro will run.

The event will be detailed in the event window and can be edited, disabled, or deleted. A description can be added to help organize multiple events of different types.



PERIODIC EVENT

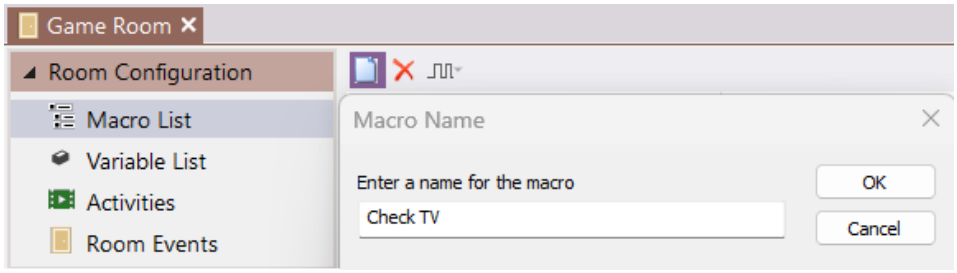
A periodic event will run a macro in second or minute intervals determined by the RTI installer. While case uses for a periodic macro are limited in this form, a common situation would be turning off the event and letting the end user control when the event is enabled or disabled.

Prerequisites for creating a periodic event:

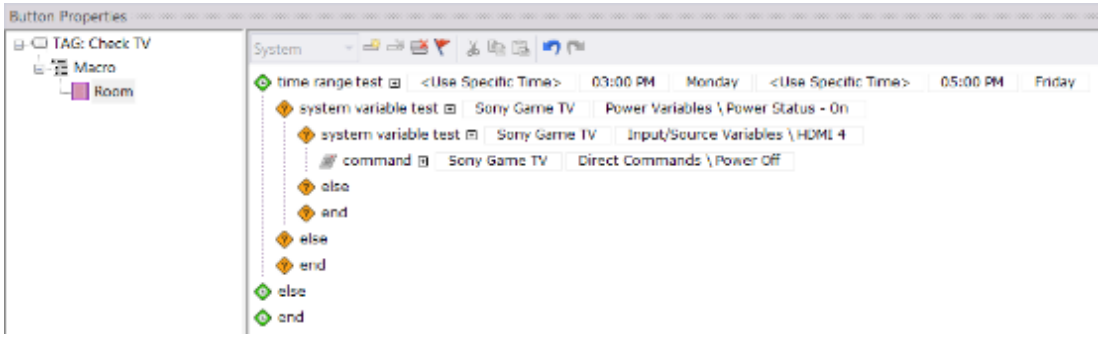
- Create a global, room, or source macro to run when the event is triggered.

Scenario: The end user requests that between 3:00 PM and 5:00 PM, his child may not use the television in the game room during homework hours. A periodic macro can be used to check if the Television is on and switched to the game input.

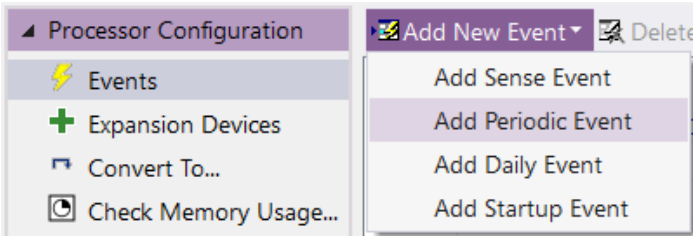
1. Create a room macro in the game room. Select the game room in the workspace, then select Macro List in the room configuration window.



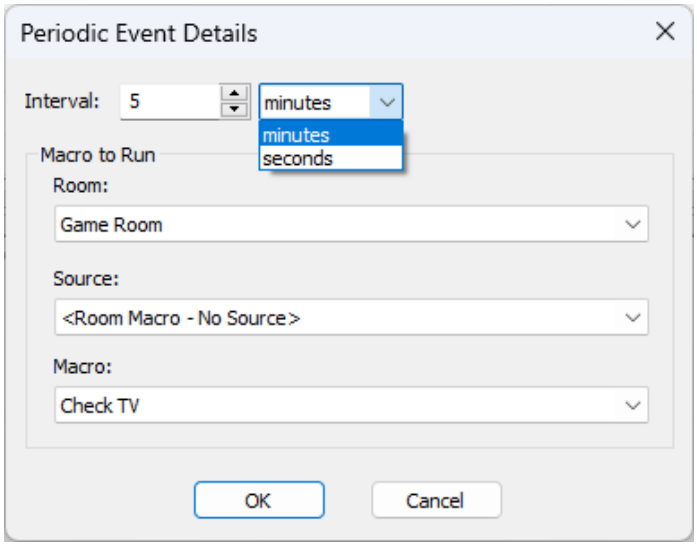
2. Create a macro name that the periodic event can use. The macro uses a time range test to check if the TV is on and set to the game input between 3 PM and 5 PM, Monday through Friday. If it is, it will turn it off.



3. Create the periodic event trigger by selecting the processor in the workspace and selecting **Events** in the **processor configuration** window. Select Add New Event, and select **Periodic Event**.



4. Configure the periodic event details window options.



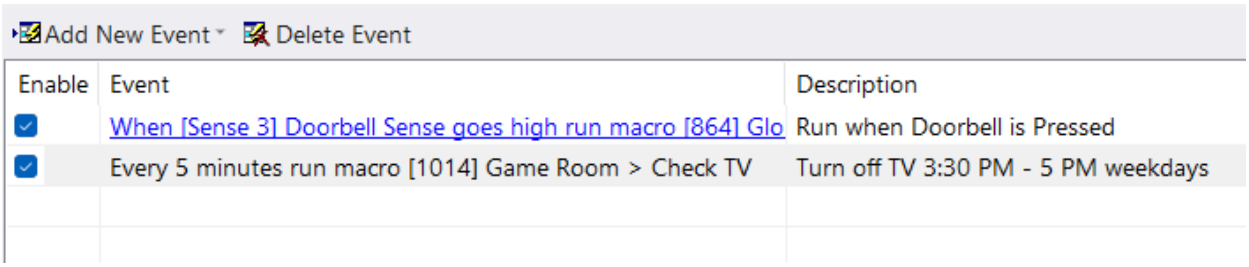
Interval: Select the number of minutes or seconds the event should run.

Macro to Run
Room: Select the Room where the macro is located.

Source: Select the source macro in the drop-down menu if using a source macro. Leave this field as is if using a room macro.

Macro: Select the macro to run.

5. Ensure the macro is enabled in the macro window. Optionally, a description can be added to provide details on the event.



Daily Event

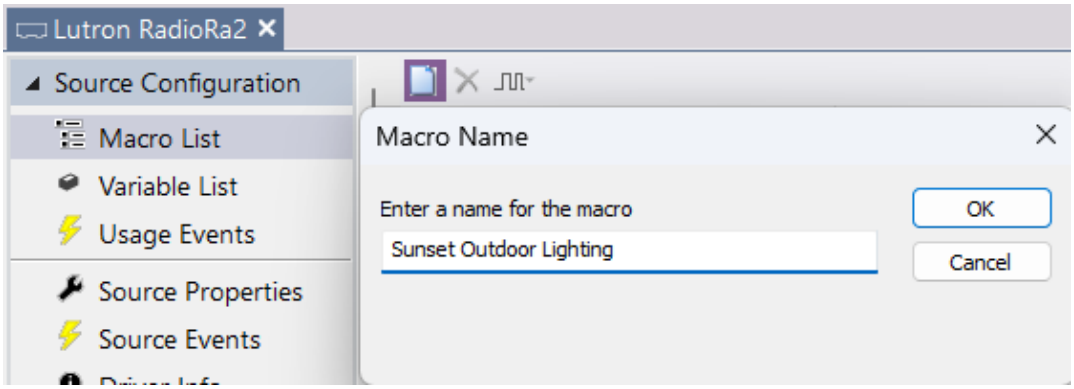
A daily event uses the XP processor's built-in astronomical clock to run events at designated times and days. Events can be run before or after sunrise based on designated minutes.

Prerequisites for creating a daily event:

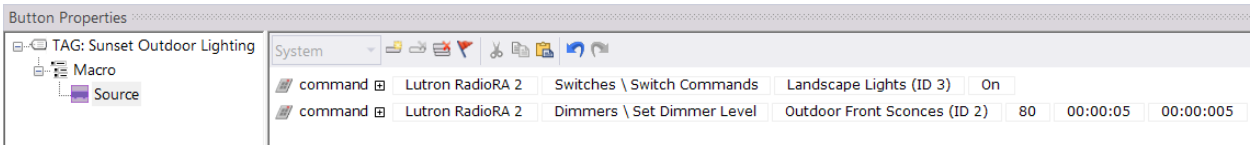
- Create a global, room, or source macro to run when the event is triggered.

Scenario: The end user requests that his outdoor and landscape lights turn on 10 minutes after sundown each night of the week.

1. Create a source-level macro on the lighting control source device. Please note this can be done on the global or room level. Select the lighting source in the workspace, then select Macro List in the source configuration window. Create a macro name that best describes the event functionality.

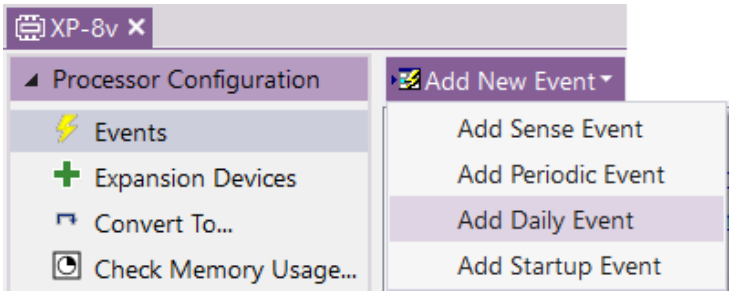


2. Create a macro to turn on the outdoor lights and patio as required.

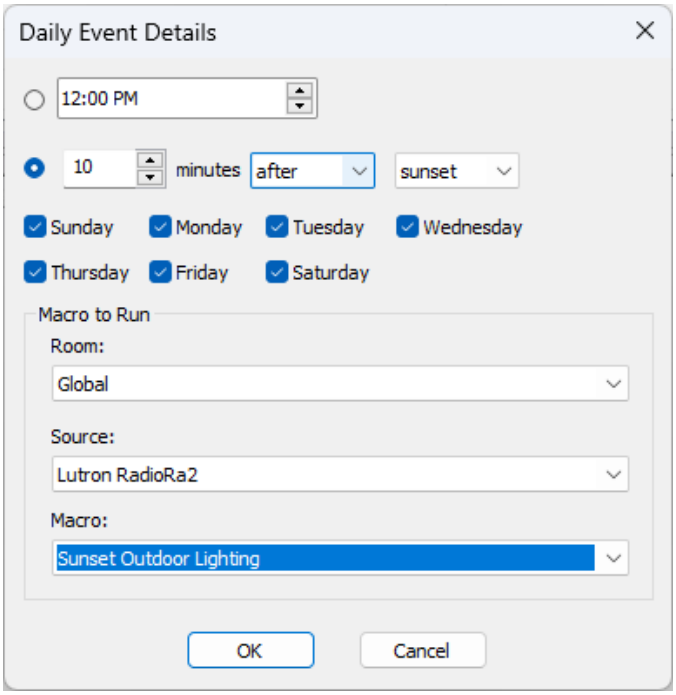


3. Once the source macro is created, create a **daily event**. Select the XP Processor in the workspace, then select **Events** in the **processor configuration**.

4. Select **Add New Event** and select **Daily Event**.



5. In the **Daily Event Details** window, enter the available parameters.



Time: Tick this option if the event runs at a specific time.

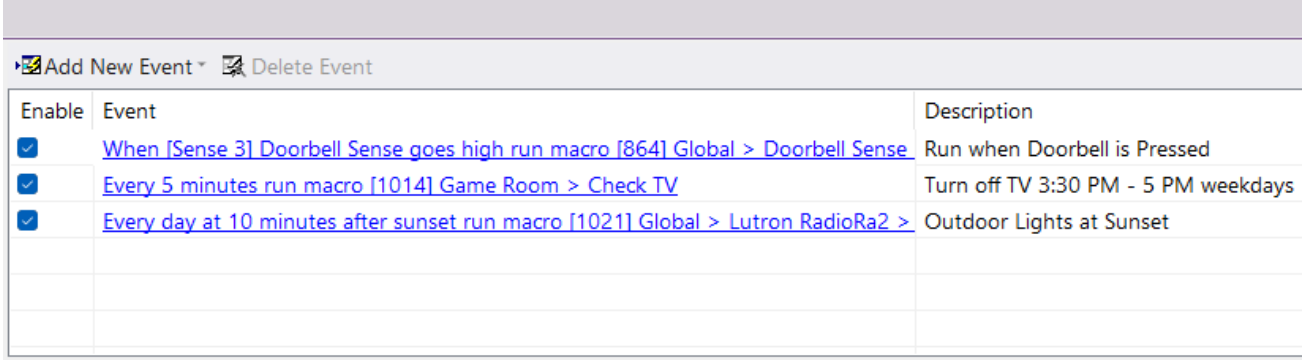
Sunrise/Sunset: Tick this option if the event occurs before or after sunrise or sunset. Select the minutes, select before or after, and sunrise or sunset.

Days of Week: Select the days of the week the event should run.

Room: Select the room where the macro or source is located.

Source: If running a source macro, select the source device.

Macro: Select the macro based on the room and source.



6. Select OK to save your changes and evaluate the event summary to ensure the event is enabled. Optionally, add a description to the event.

Startup Event

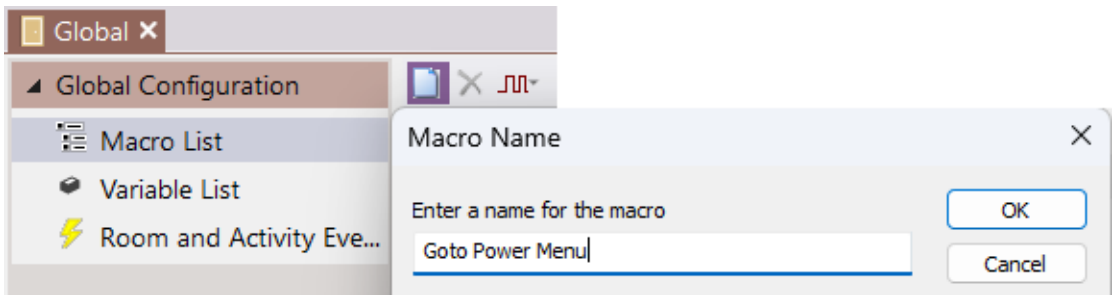
A **startup event** will automatically trigger when the processor is synchronized or rebooted.

Prerequisites for creating a startup event:

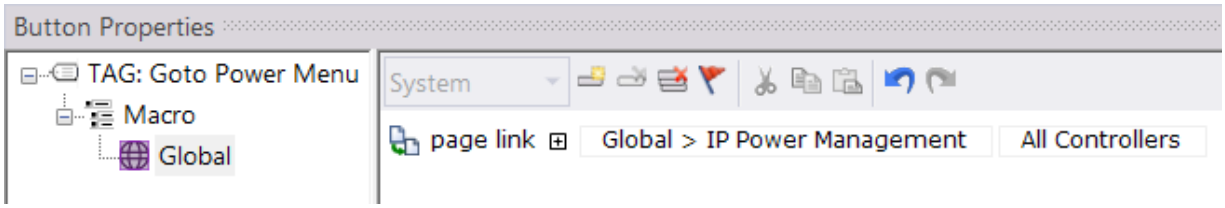
- Create a global, room, or source macro to run when the event is triggered.

Scenario: The project location is subject to severe power outages. When there is a power outage, it is common to lose power several times before the issue is resolved. If the processor recovers from a power outage, the client wants to be taken to a power management screen to turn off his power outlets to avoid repeated surges.

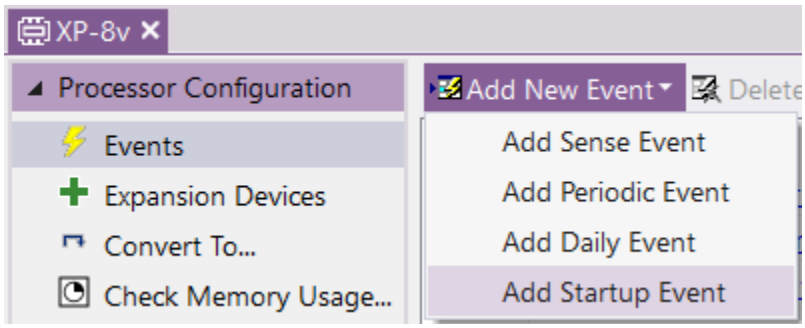
1. Create a **global** macro by selecting the processor and selecting **Macro List**. Select **Add New Macro** and create a **macro name** that suits the function.



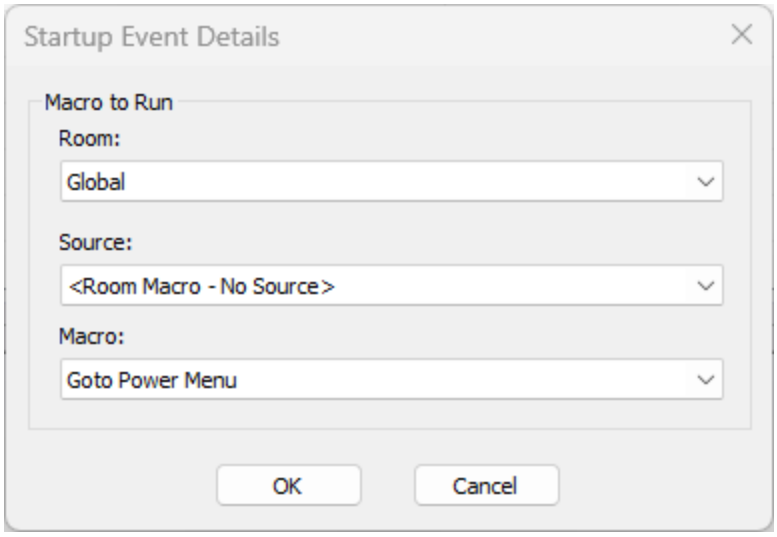
2. Program a macro that will page link to the IP Power page for selected or all controllers in the project.



3. Create a **startup event** by selecting the **processor** in the workspace and selecting **Events** in the **processor configuration** window. Select **Add New Event** and select **startup event**.



4. Configure the **startup event details** window.

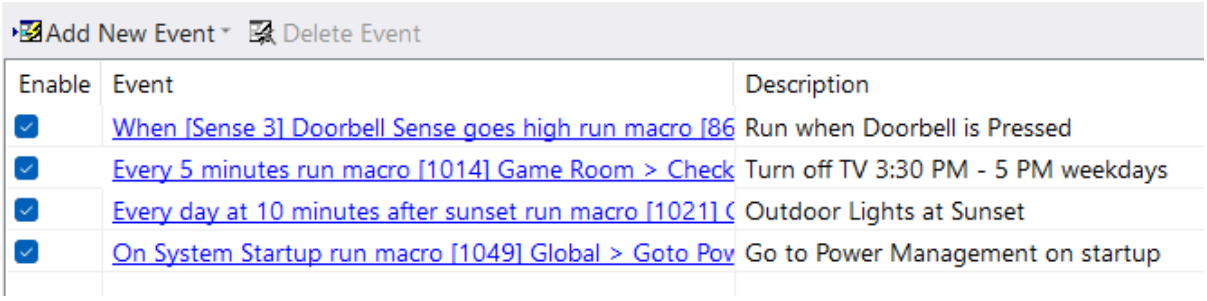


Room: The location of the macro.
Source: If created on a source, select the source. If created in a room or the global area, leave this option as is.

Macro: Select the macro to run when the event is triggered.

5. Select OK to apply the startup event details.

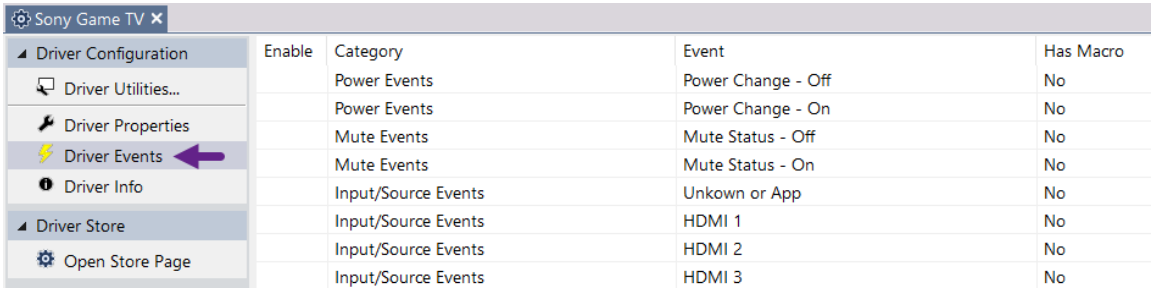
6. Review the event grid to ensure the event is enabled. Optionally, add a description to the event for reference.



Driver Events

In addition to processor events, most two-way drivers support **driver events**. Some drivers do not support events, and event types vary by device.

Driver events can be found by selecting the driver under the driver tab in the workspace and selecting **Driver Events** in the **driver configuration** window.



Scenario: The client wants his lighting keypad top button to turn on her Kitchen fixture. During the week, Monday through Friday, from 6:30 AM to 8:00 AM, she wants the button to turn on her favorite music while she is getting ready.

1. Select the lighting control driver in the driver tab, then select **Driver Events** in the **driver configuration** window.

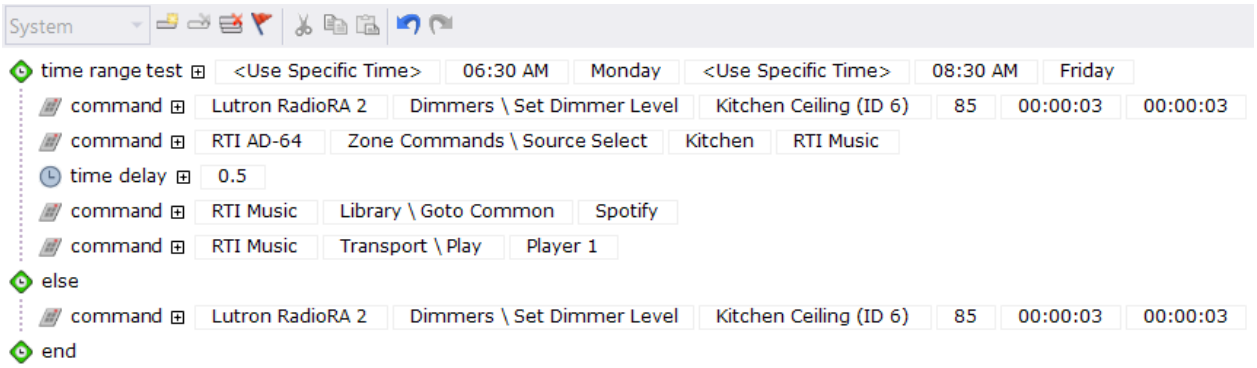
2. Find and highlight the event trigger, then program an event macro in the macro window. In this example, the event trigger is the Lutron Driver Kitchen Keypad, Scene 1.

3.

Enable	Category	Event	Has Macro
	KITCHEN KEYPAD (ID 14)	KITCHEN KEYPAD (ID 14), Dimmer/Scene Off/Shade/Switch Status	No
	KITCHEN KEYPAD (ID 14)	KITCHEN KEYPAD (ID 14), LED 1/Scene 1 Status	No
	KITCHEN KEYPAD (ID 14)	KITCHEN KEYPAD (ID 14), LED 2/Scene 2 Status	No
	KITCHEN KEYPAD (ID 14)	KITCHEN KEYPAD (ID 14), LED 3/Scene 3 Status	No
	KITCHEN KEYPAD (ID 14)	KITCHEN KEYPAD (ID 14), LED 4/Scene 4 Status	No
	KITCHEN KEYPAD (ID 14)	KITCHEN KEYPAD (ID 14), LED 5/Scene 5 Status	No
	KITCHEN KEYPAD (ID 14)	KITCHEN KEYPAD (ID 14), LED 6/Scene 6 Status	No

We selected the **Scene 1 Status** on the **Kitchen Keypad**, the top button on the scene controller.

4. The keypad button press will trigger an event to turn on the kitchen ceiling light and play the client's favorite streaming station Monday through Friday, 6:30 AM to 8:30 AM. At all other times, the light will turn on.



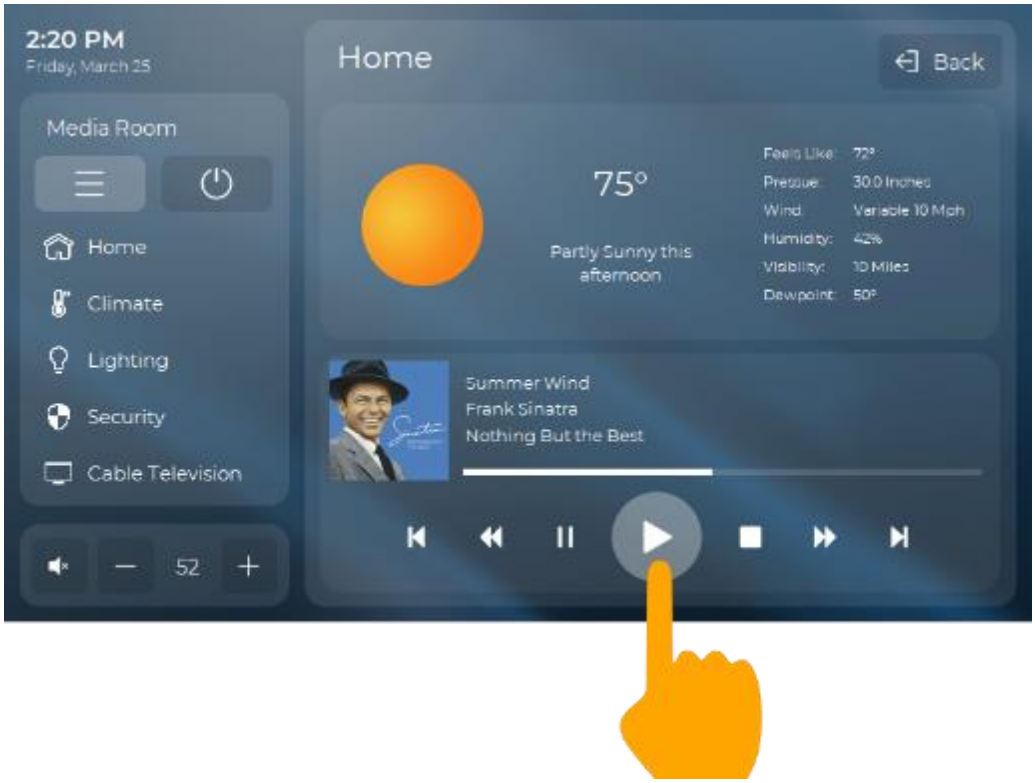
5. Ensure the event is **enabled** in the event grid.

Enable	Category	Event	Has Macro
	Master Bath Fan (ID 13)	Master Bath Fan (ID 13), Button 19/Bottom Raise Released	No
	KITCHEN KEYPAD (ID 14)	KITCHEN KEYPAD (ID 14), Dimmer/Scene Off/Shade/Switch Status	No
☑	KITCHEN KEYPAD (ID 14)	KITCHEN KEYPAD (ID 14), LED 1/Scene 1 Status	Yes
	KITCHEN KEYPAD (ID 14)	KITCHEN KEYPAD (ID 14), LED 2/Scene 2 Status	No

The event is **enabled**, and a "**Yes**" is indicated in the "**Has Macro**" column.

Pro Tip: The intrigue in using events is based on not having to interact with the system and the devices' interoperability. Events can be a major selling point because they provide safety, comfort, convenience, and enjoyment without requiring training or interaction with an interface.

Scenario: The client wants to press the play button on his RTI controller dashboard and activate the music in the room where the keypad is available.

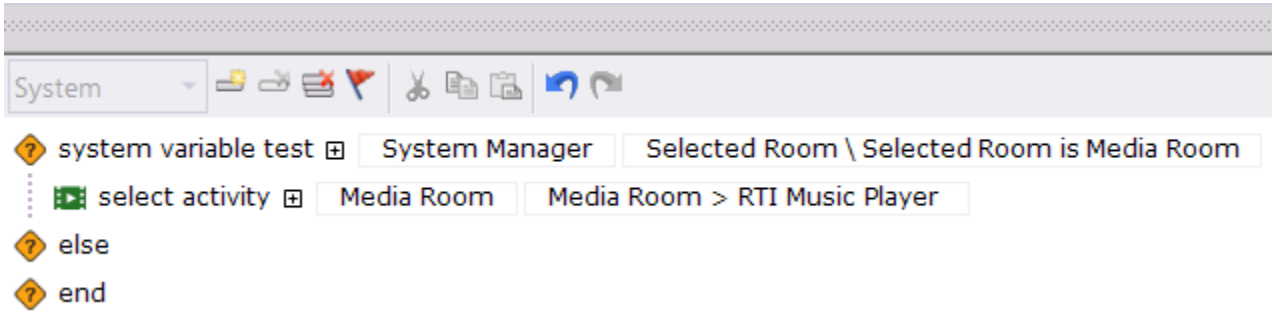


1. Select the RTI Music control driver in the driver tab, then select **Driver Events** in the **driver configuration** window.

2. Find and **highlight** the **event trigger**, then program an event macro in the macro window. In this example, the event trigger is RTI Music/Player1/Play.

Enable	Category	Event	Has Macro
	Player Player 1 (98:fe:95:00:38:99) Events	Play	No
	Player Player 1 (98:fe:95:00:38:99) Events	Pause	No
	Player Player 1 (98:fe:95:00:38:99) Events	Stop	No
	Player Player 1 (98:fe:95:00:38:99) Events	Shuffle None	No
	Player Player 1 (98:fe:95:00:38:99) Events	Shuffle Song	No
	Player Player 1 (98:fe:95:00:38:99) Events	Shuffle Album	No
	Player Player 1 (98:fe:95:00:38:99) Events	Repeat None	No

If the selected room in this example is the Media Room where the keypad is located, it will run the RTI Music activity via the System Manager when the play button is pressed.



3. Verify the macro is **enabled** in the macro grid and the "Has Macro" column is set to "Yes."

Enable	Category	Event	Has Macro
☑	Player Player 1 (98:fe:95:00:38:99) Events	Play	Yes
	Player Player 1 (98:fe:95:00:38:99) Events	Pause	No
	Player Player 1 (98:fe:95:00:38:99) Events	Stop	No

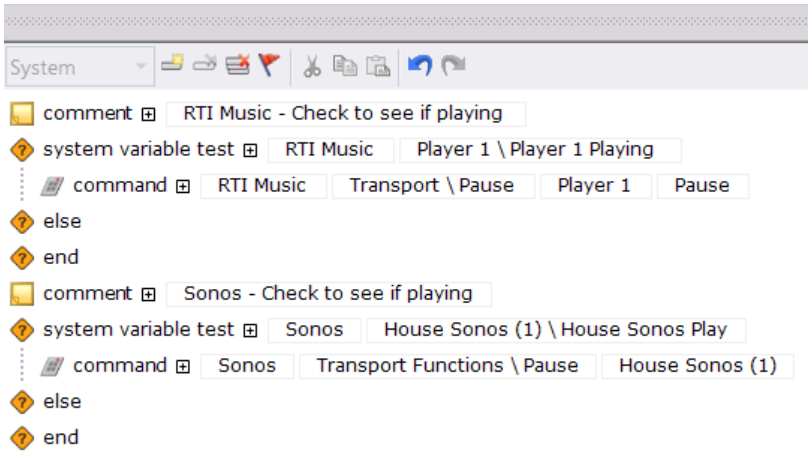
Scenario: When the client and his family shut down the audio distribution system in the large residence, they want to ensure that all music sources stop streaming while the system is off.

1. Select the AD-64 audio control driver in the driver tab, then select **Driver Events** in the **driver configuration** window.

2. Find and **highlight** the **event trigger**, then program an event macro in the macro window. In this example, the event trigger is the AD-64/Last Zone Off.

Enable	Category	Event	Has Macro
	Power	First Zone On	No
	Power	Last Zone Off	No
	Power 1-8	Media Room On	No
	Power 1-8	Media Room Off	No

3. The macro will pause the RTI Music and Sonos player if they are playing.

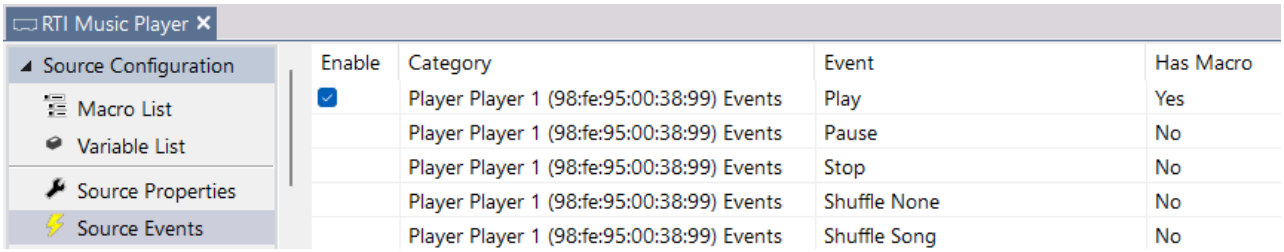


4. Ensure the event is enabled in the event grid and the “Has Macro” field contains a “Yes.”

Enable	Category	Event	Has Macro
	Power	First Zone On	No
☑	Power	Last Zone Off	Yes
	Power 1-8	Media Room On	No
	Power 1-8	Media Room Off	No

Source Events

After creating a driver event, you can view, edit, and create events by selecting the **source** in the **workspace** and then **Source Events**.

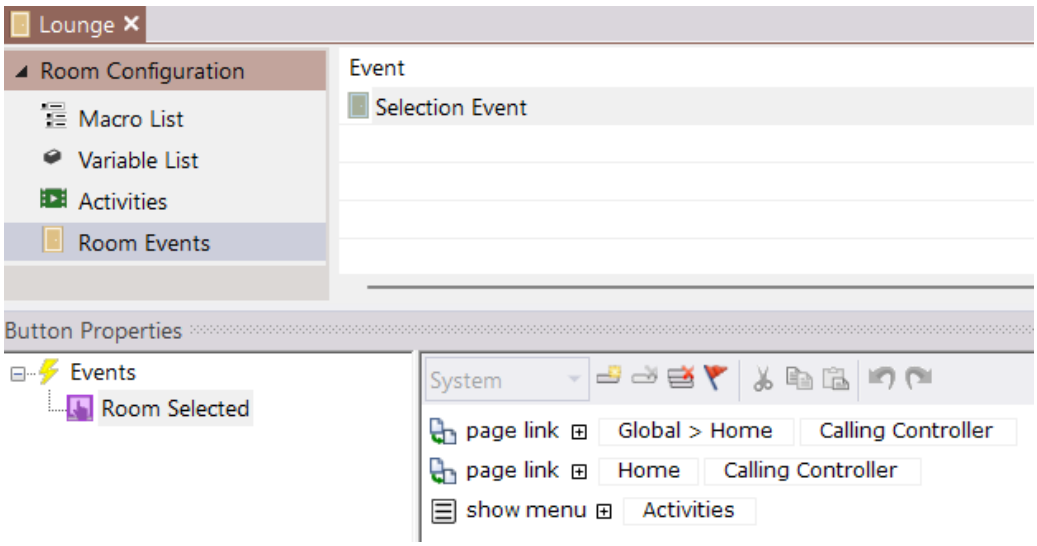


Pro Tip: It is best practice to manage **source events** under **driver events**. Use the source events to browse or view existing events when performing source maintenance.

Room Selection Events

Events are available when **selecting a room** and are pre-programmed with basic functions when a room is added to the workspace. These **events** are known as **room selection events**.

1. To view **room selection events**, select the **room** in the **workspace**, then select **Room Events** in the room configuration window.

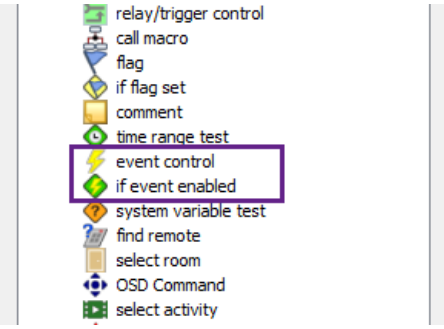


2. Select the **Selection Event**, then **Room Selected** in the button properties, and you can view or change the existing event for the current room.

In this example, the room selection event will run when the lounge is selected using the “selected room” macro step and is available in the list and button menus.

Event Control

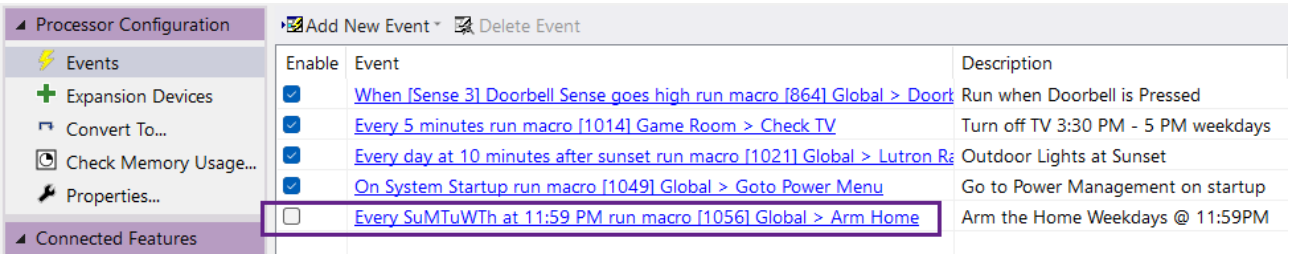
Sometimes, a client wants to cancel an event due to an unforeseen circumstance. For example, if the house is set to arm the security, close the garage, and turn off lights at a certain hour, a late-night party would warrant turning off the event and enabling it at a convenient time. For this reason, there are event control macro steps to help manage events if the situation calls for it.



Event Control- A macro step that turns processor or driver events on or off.

If Event Enabled- A macro step that provides an if/else statement based on events with an enabled or disabled status.

1. If event control is needed, the processor or driver event must be turned off in the event grid, as the end user will manage the event. Ensure the event is disabled by removing the checkmark in the enabled column.

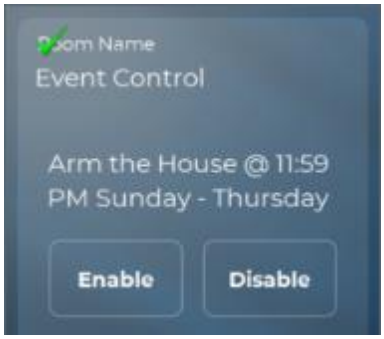


This event will arm the home on weekdays at 11:59 PM. In addition to arming the home, it will lock the doors, close the garages, and turn off lights around the premises. While this works well much of the time, there are occasions when the event needs to be disabled when guests are visiting.

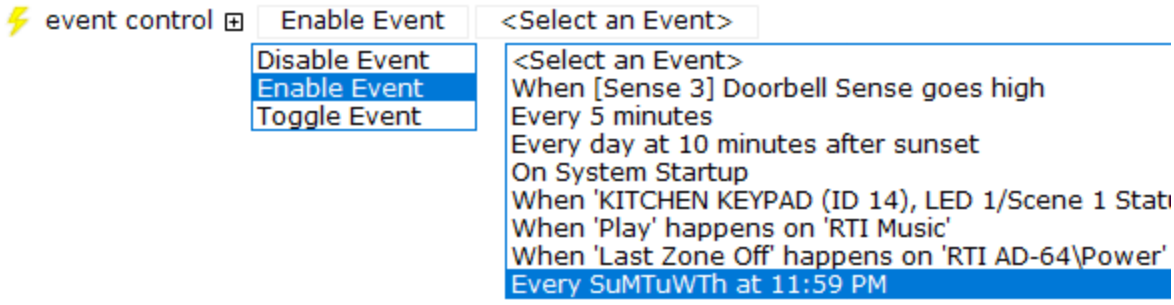
2. Create a source activity named “**Event Control**” for the end user by adding a **blank pageset** to the workspace.

3. Ensure the activity is configured and programmed in the System Manager.

4. Create an interface with a descriptive event name and an "**Enable**" and "**Disable**" tagged buttons. Multiple events may be created if needed.

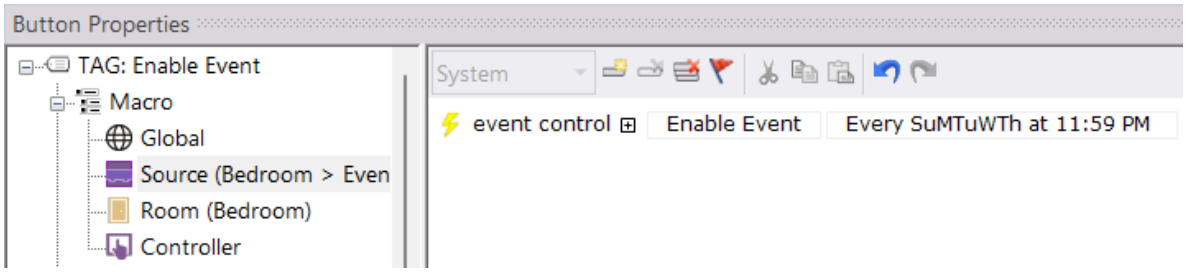


5. Double-click or drag an **Event Control** macro step into the macro window for the **enable** button.



6. Events may be set to **disabled**, **enabled**, or **toggled**. Select **Enable Event**.

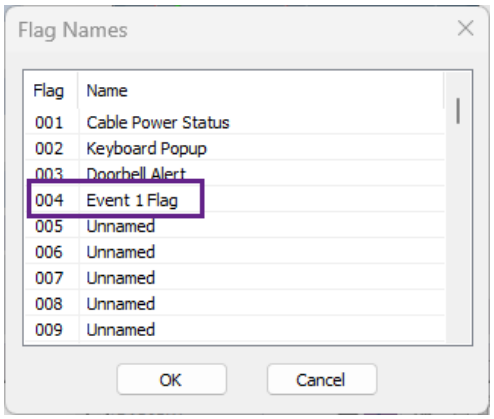
7. **Select** an **event** from the drop-down menu. All **processor** and **driver** events will be displayed.



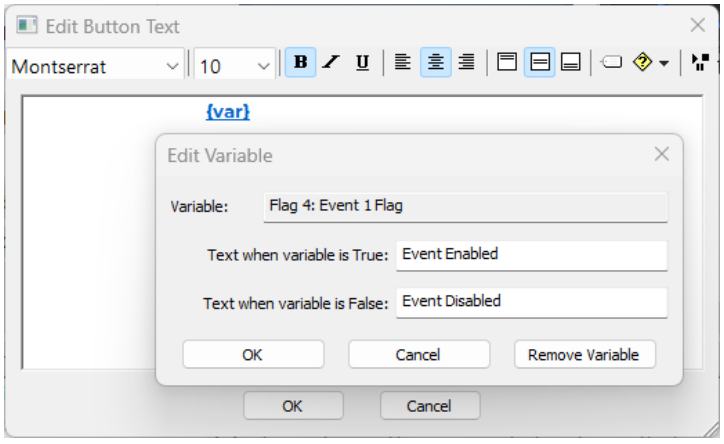
8. **Repeat** the same process on the "**disable** event" button. Copy and paste the enable event control to the disable button and change the parameters for quick action.

RTI installers may want to create a more detailed event management screen in certain situations. For example, the client wants a dashboard showing if an event is active or disabled and a single button to toggle the event.

1. Create an event flag status for the specific event you want to check by selecting the red flag icon in the macro window toolbar. Name the flag. "Event 1 Flag." This flag will be used to track the status of the event.

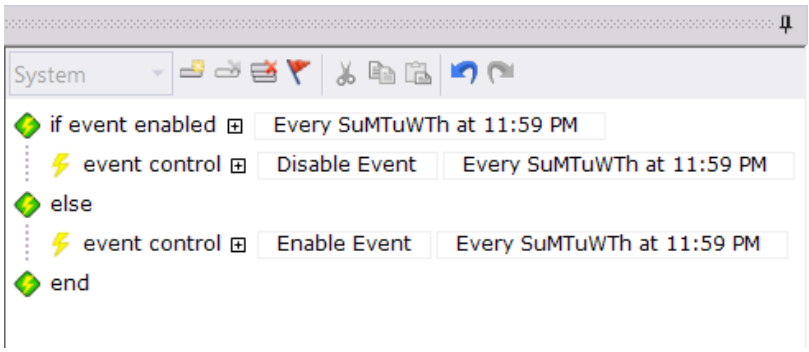


2. Create a button, and insert a control variable using the XP Macro flag "Event 1 Flag." Edit the variable and change the true and false feedback to read "Event Enabled" for true and "Event Disabled" for false.



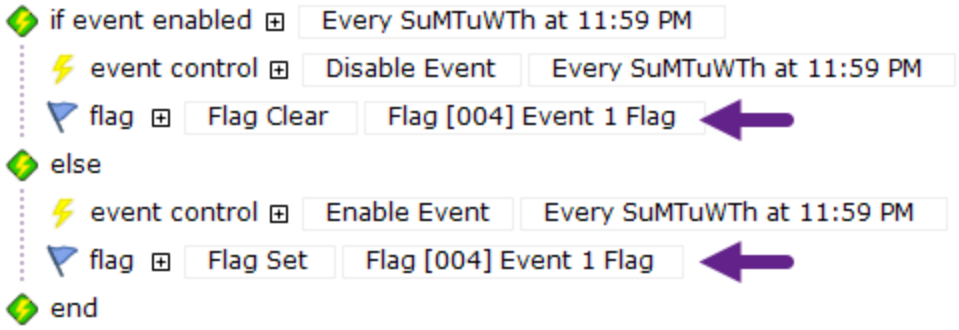
3. Double-click or drag an **If Event Enabled** macro step on the tagged button to the macro window after selecting a macro level.

4. Select the event in the if event enabled statement. Double-click or drag an event control statement. Set the event control to “disable event” if enabled and another event control command to enable the event if else.

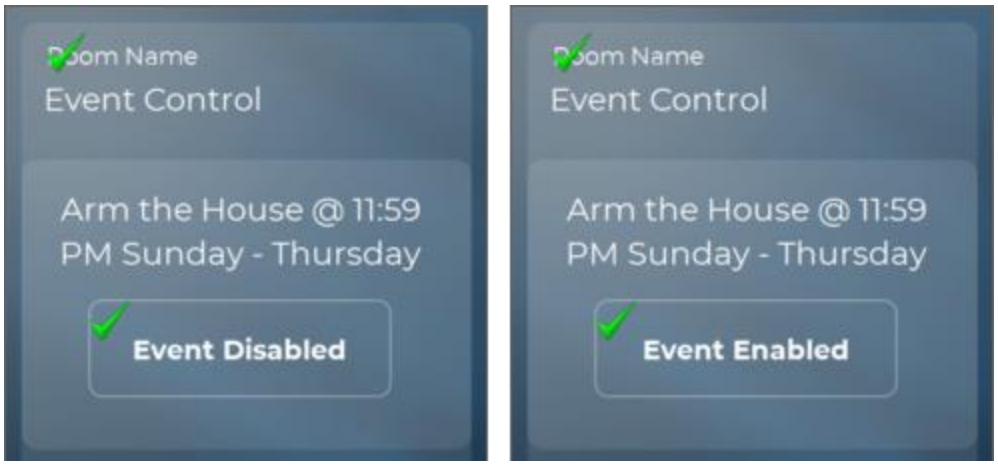


This macro will turn off the event if it is set to enabled. Otherwise, it will set it to “enabled.”

5. Drag or double-click a **Flag** macro step into the macro window for the **disabled** side of the **if event enabled** statement. Add another for the **enabled** side. **Set** the **flag** status when the event is enabled and **clear** it when it is **disabled**.



6. Now, the user interface will reflect the correct status of the event and allow the client to turn the event on or off as needed. Optionally, you may add the flag to the reversed state of the button so that the reversed state will display when the event is enabled.



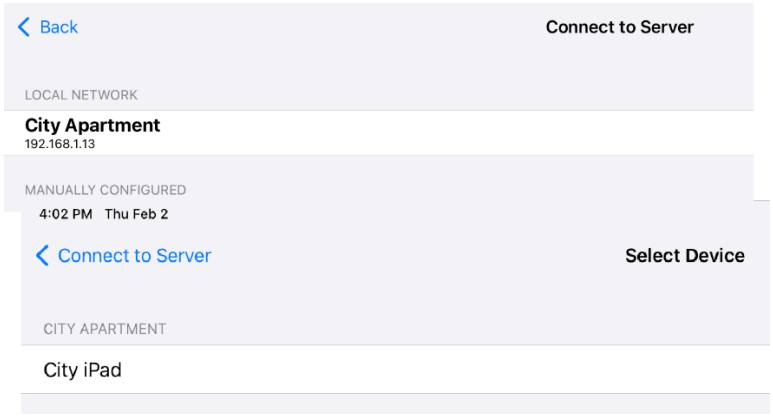
RTiPanel Multi Location

Integration Designer features a "**multiple location"** management screen for clients using RTiPanel controllers with multiple homes or businesses. End users can easily switch between client locations using preconfigured object commands available in Integration Designer software. The interface allows installers to add, remove, and edit the names of multiple locations.

1. To upload a site configuration for iOS products:
   - **Apple® iOS devices**: Go to the iOS device settings and select RTiPanel. Set the "Reconfigure" button to "ON" (screenshot on the left). This Reconfigure setting resets itself to "Off."
   - **Android™ devices**: Go to Setting>Apps>Clear Data. Click "OK" to clear app data.

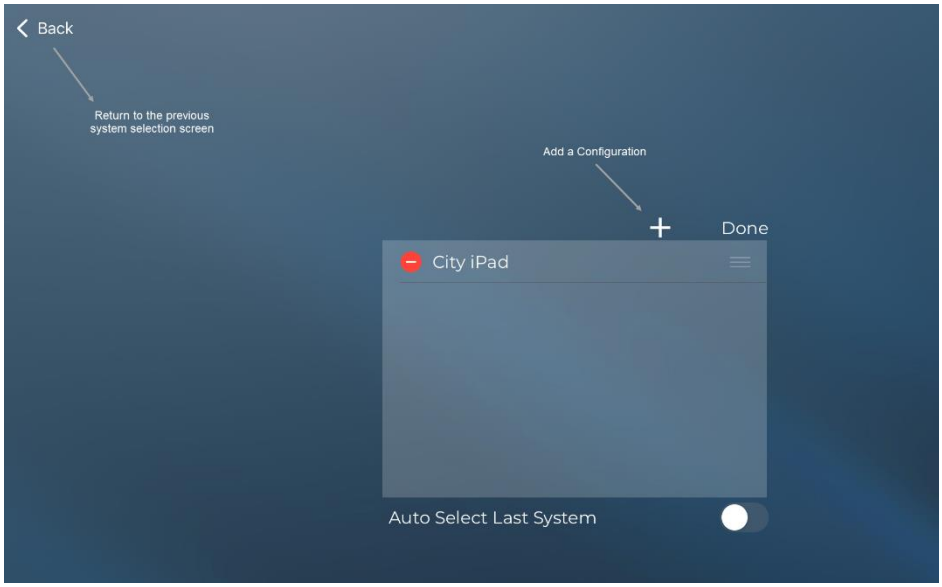
2. You will be taken to the system selection screen. You can add and select existing projects from this screen. Select **Click here to add a new system** option.

3. Select the project and the controller(s), and they will be added to the system selection screen.

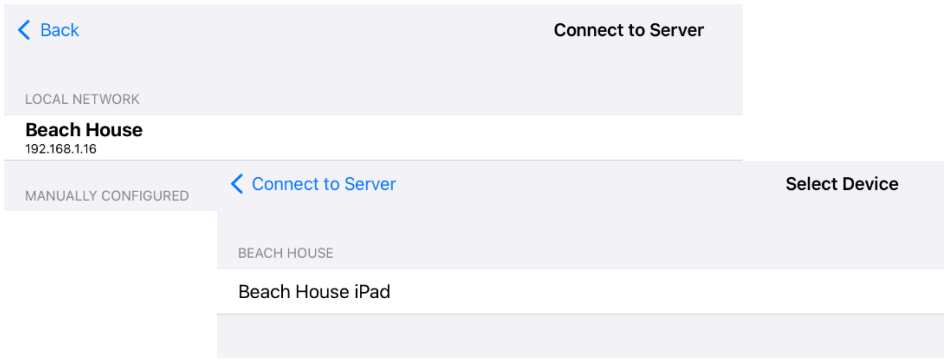


In this example, the City iPad has been added.

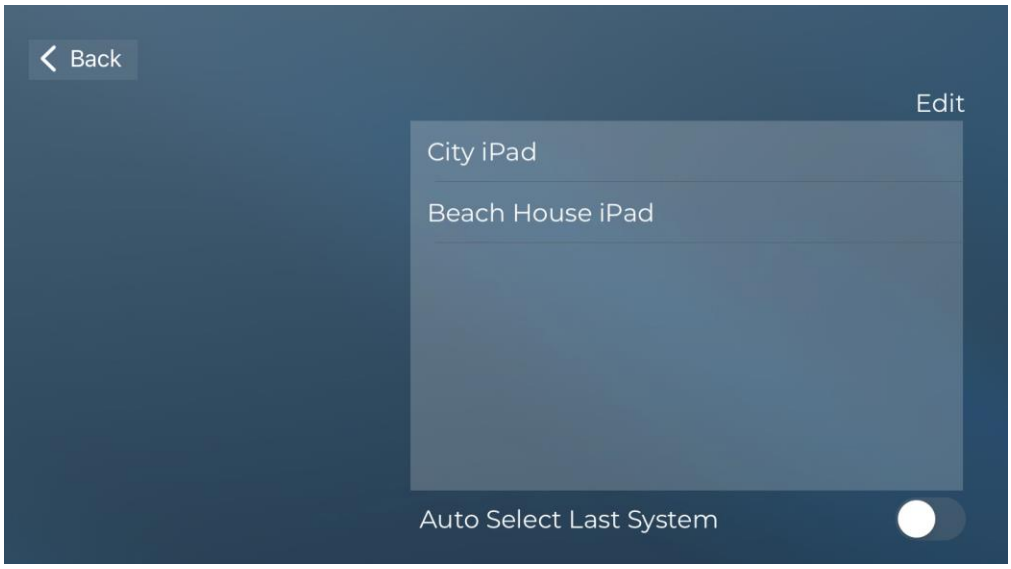
4. To add other project controllers to the system selection screen, select the **Settings** icon in the upper right corner of the system selection screen.



5. Select the **+** icon and repeat the project download to add devices to the system location screen.



Once added projects are downloaded, they appear on the system location screen.



Several options exist in the system maintenance screen.

Back: Go back to the earlier system configuration menu.

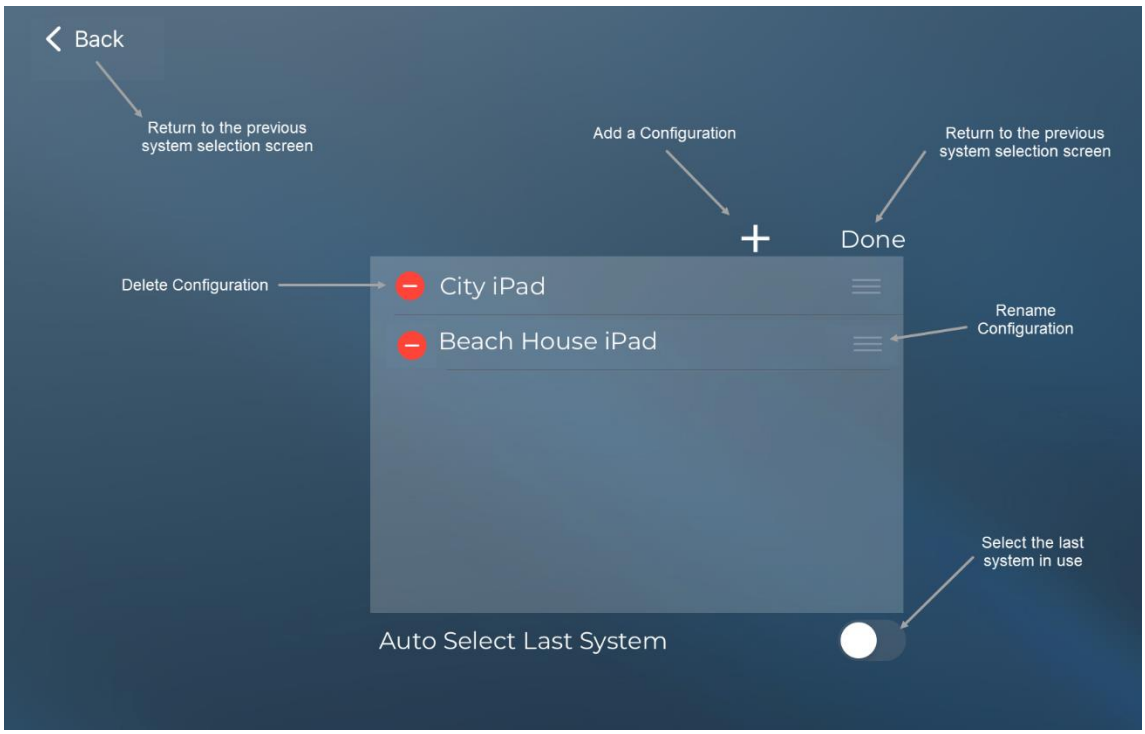
Delete Configuration: Remove the configuration from the selection menu.

Add a Configuration: Add another project to the system selection menu.

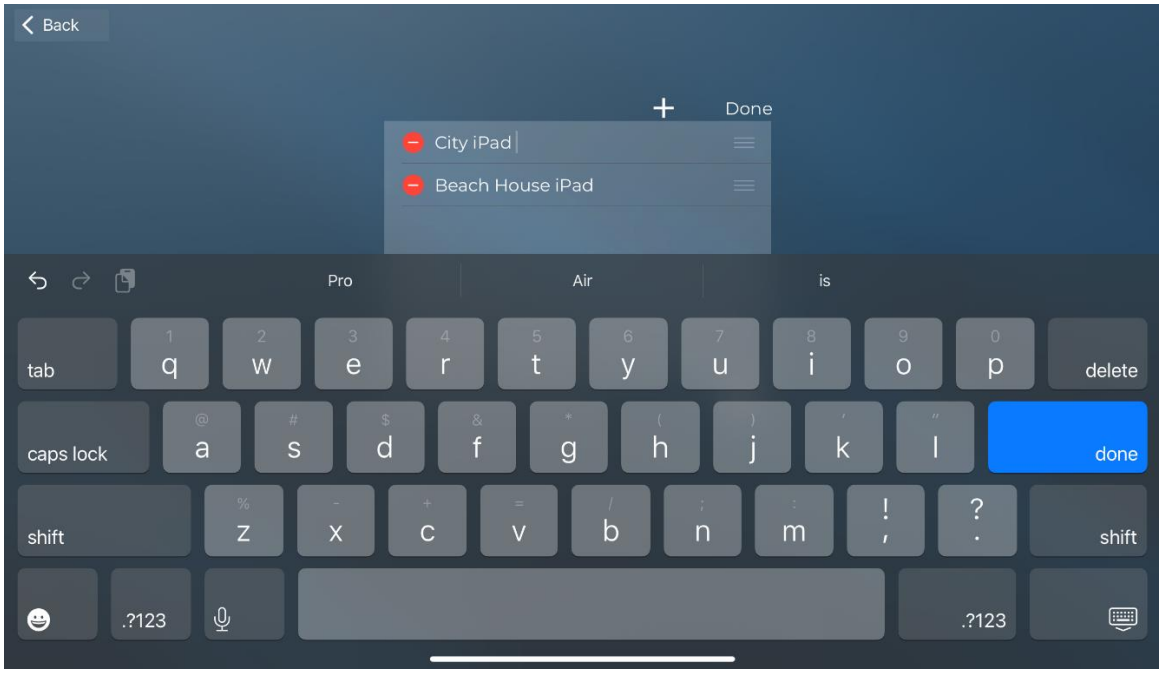
Done: Go back to the earlier system configuration menu.

Rename Configuration: Use a keyboard to change the name of the project.

Auto Select Last System: If enabled, automatically go to the last system used when launching the RTiPanel app. If disabled, go to the welcome screen when the RTiPanel app is launched.

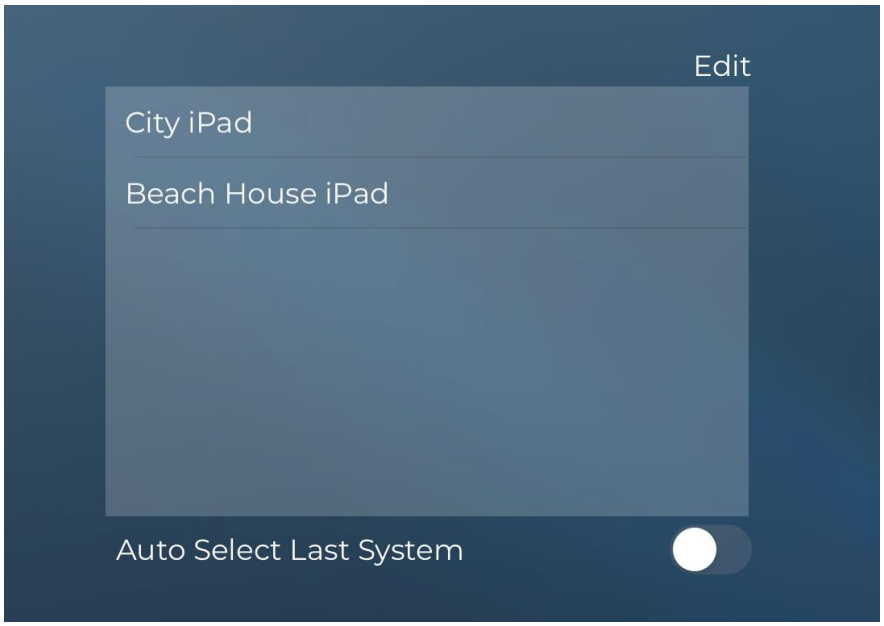


6. Selecting the **rename configuration** menu item will display a keyboard. Rename a system to a name that the client can easily identify.



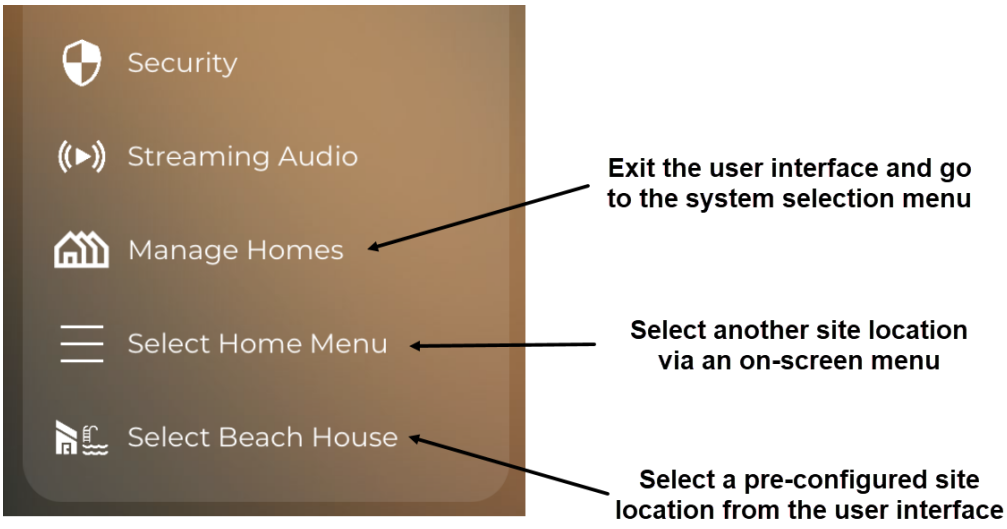
7. Once finished, select **Done** to save your changes and return to the system selection menu.

8. To access a project, select the project from the system selection menu.

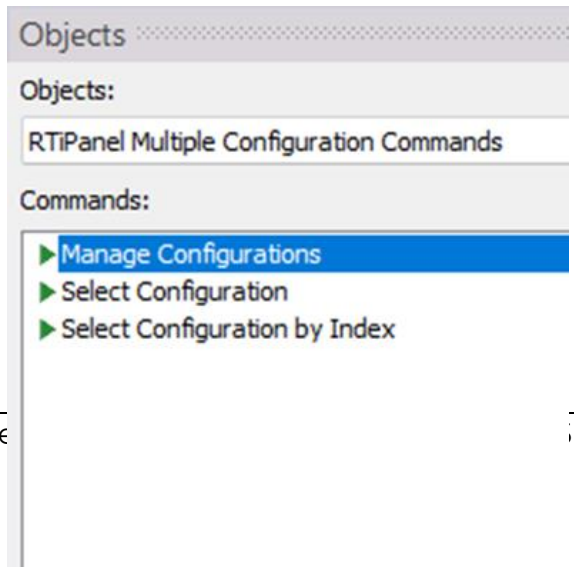


Integration Designer software has added more objects to allow end-users to access other site configurations directly from a user interface via the system selection menu. In addition, they may also select another site directly from a user interface.

You can choose which object commands to supply the client based on their needs.



1. Select the Coral Icons template in the bitmap library. Search for “Switch” in the search box field.



2. Drag the applicable icons to the user interface and select the Objects tab.

Manage Configurations: This command sends the user back to the RTiPanel home page to select a different location. Best used when a user seldom switches between systems.

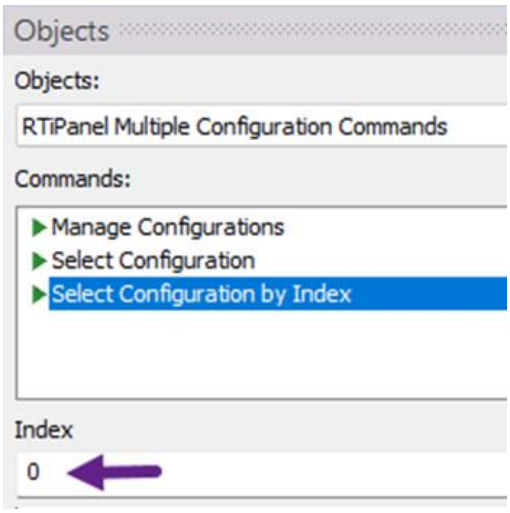
Select Configuration: This command pops up a list, allowing the user to select a different location from the current page. Best used when the user often switches between systems.

Select Configuration by Index: This command allows you to create individual buttons for each system. Best used when a user needs to quickly switch to a specific system with a single button press.

In this example, an index is entered for each room. The order of the list of locations decides the index number to enter.

In the example below, the City iPad has an index of 0 (zero), and the Beach House iPad has an index of 1.

Enter the index number in the index field, then drag the command to the button.



Pro Tip: Name your projects based on the location to make it easier for the end user to find the location. Avoid taking the user back to the configuration screen when possible. Supply fields on the home page to help find the location so that when switched, the client can easily find the other location. Toggle the Auto Select Last System to on if the client frequents a particular location.

Viewports

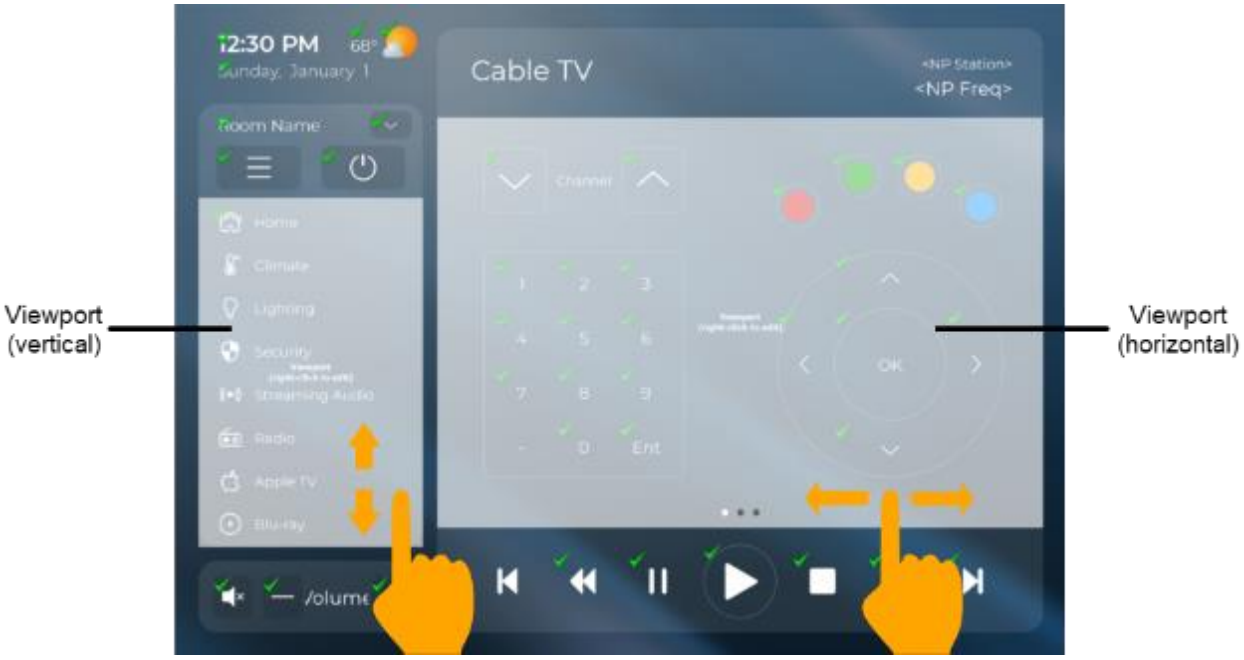
Viewports are available on iOS and Android-based RTiPanel devices and KA-series panels. Viewports can maximize screen real estate and make navigation easier by giving the end user a scrolling window to advance through multiple frames while the rest of the interface remains.

There are two types of **viewports**:

Horizontal Viewports- These may be drawn and scrolled through horizontally. (Available for iOS RTiPanel devices.)

Vertical Viewports- May be drawn and scrolled through vertically (Available on iOS and Android-based RTiPanel devices and KA-series controllers.)

Horizontal and **Vertical** viewports can be made to scroll using **page mode** (left to right) or **vertical mode** (up and down.)



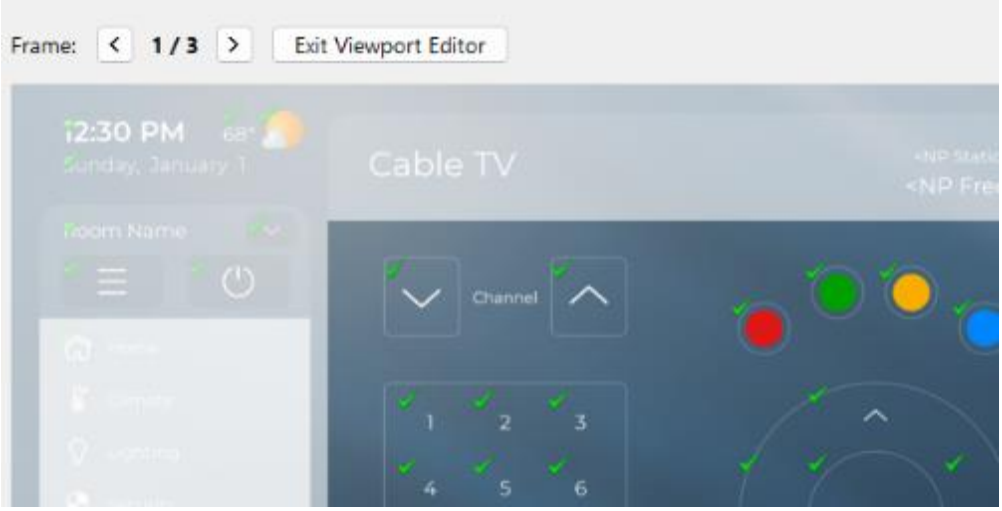
The horizontal viewport will have frame indicators showing the number of frames and current frame position. The vertical viewport has a scroll bar, which will help the end user navigate but will not be seen unless viewing the controller interface.

The viewport on the left side will scroll vertically, allowing the end user to navigate through several buttons. The viewport in the middle separates each function type in the user interface. The non-greyed-out part of the screen will be displayed as the user navigates through the different frames of each viewport.

Editing and Viewing an Existing Viewport

1. Right-click the **viewport** and select **Edit Viewport**.

2. The viewport contents will display, and the rest of the screen will be grayed. On the top of the screen, a frame indicator with left and right arrows allows navigation from one frame to another, horizontally or vertically. Exit the viewport by selecting the Exit Viewport Editor function or any gray area outside the viewport.



3. The alignment and centering tools use the viewport as the new screen when inside a viewport. The rest of the user interface outside the viewport is disregarded. For example, centering or distributing buttons will use the viewport, not the entire screen.

4. The viewport has the same capabilities as the screen. You can drag buttons, tags, objects, two-way feedback, and create layers. However, jumping to specific viewport frames via page links is not possible.

Pro Tip: Viewports are an excellent way to use a small screen section to supply functionality. For example, a small viewport on a cable page can supply multiple channel favorites without the end user navigating to multiple pages.

Creating a Viewport

A viewport is needed to show various channel favorites so the client will not have to navigate off his cable page. They would rather select favorites than enter channel numbers. Favorites can be browsed without losing access to the main functions available on the cable page.

1. Pick a location for the vertical viewport on the RTiPanel user interface.

2. Select the **draw button** icon in the tool palette toolbar, then select the **Viewport** icon. Holding down the **left** mouse button, **draw** your viewport on the screen. Return to selection mode and make any changes to the size and location.

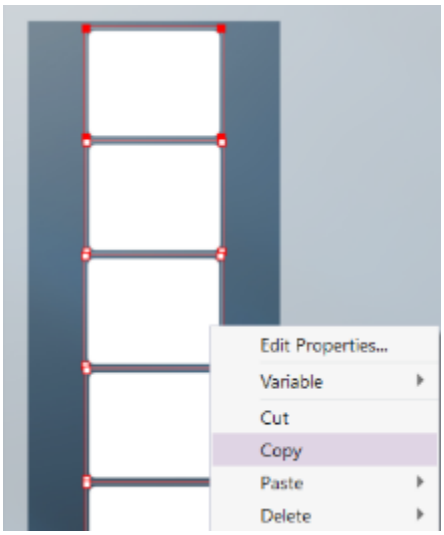
When drawing the viewport, there is no need to get the perfect size and location. These changes can be made in selection mode.

3. .Select the Coral – Elements bitmap library folder and search for “button favorite.”

4. **Right-click** the viewport, and select **Edit Viewport**. The rest of the screen will be greyed while working inside the viewport.

5. Drag the **button favorites** into the viewport. **Resize** to fit the viewport and **align** them symmetrically.

6. **Select** the buttons on the viewport by selecting CTRL-A. Select the **copy** function.



7. **Advance** to the next viewport by selecting the right arrow button above the user interface.

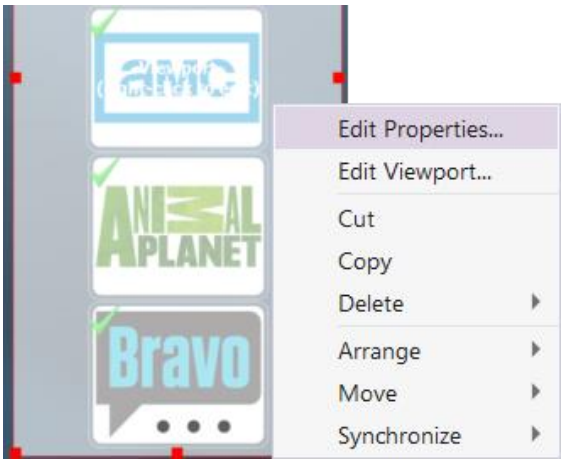
8. **Paste** the buttons to the second viewport frame. Continue advancing and pasting the buttons on the next frame until you have added all the frames.

9. Once you have created as many viewport frames as needed, select the left frame arrow until you are on the first frame of the viewport. While continuing in Viewport mode, select the channel favorites bitmap library, drag and drop channels to each button, and advance to the next Viewport frame.

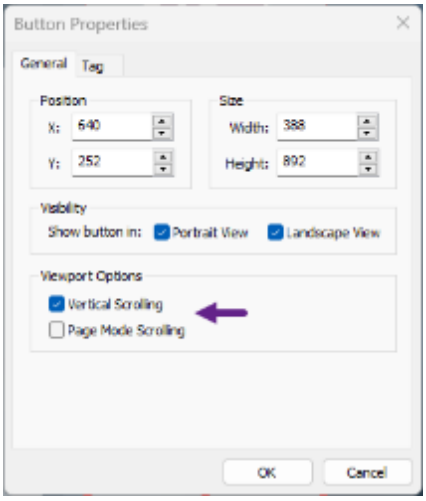
Once the channels have been distributed to the buttons, **programming** them is necessary. If you use the channel icon set included in the Integration Designer bitmap library, the buttons should already be tagged properly.

10. When the channels have been programmed, select any area outside the viewport to exit the Viewport editor.

11. The last step is to **set** the **Viewport** to **page mode** (horizontal) or **vertical scrolling**. Right-click Viewport and select **Properties**.



12. Set the **Viewport Options** to **Vertical Scrolling** or **Page Mode** if you want to use your Viewport with horizontal gesture control.



Select **OK** to apply your changes; your Viewport is programmed and ready for use. Upload your project to test the viewport and make any changes necessary.

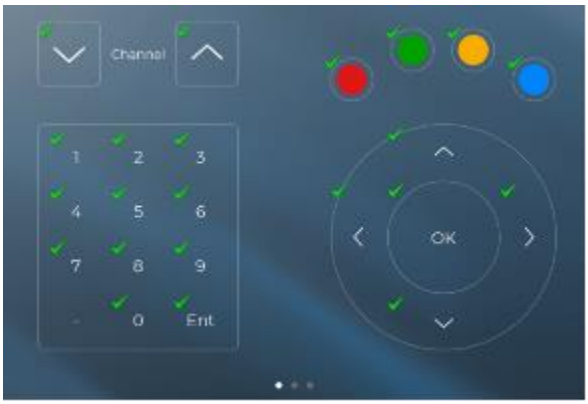
Pro Tip: Avoid using too many frames on a Viewport. Consider adding pages if you need to add a high number of Viewports. Leave room on the sides if you use vertical scrolling so the client can navigate the viewport without selecting buttons. If you use horizontal scrolling, leave plenty of room on the bottom for the frame indicators and supply an area to gesture without selecting a function.

Other Viewport Uses

Lighting Control

Activities and Rooms

Separating Functionality



Programming Hybrid Macros

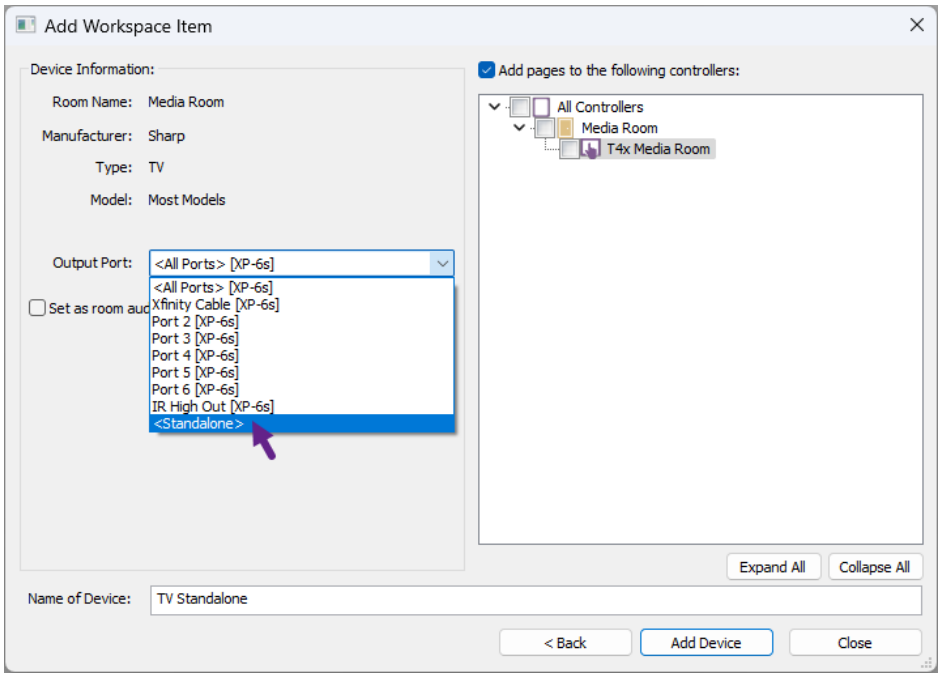
A **hybrid macro** is a macro that runs a combination of commands from the controller (**standalone mode**) and the processor (**system mode**.) In certain situations, wiring and control options are exhausted, and a device requires control via infrared line-of-sight from the controller. The System Manager may not be used, while these situations can be controlled through other programming methods.

Scenario: A Media Room is using the following equipment:

- Television
- Receiver
- Blu-ray Player
- Roku Player

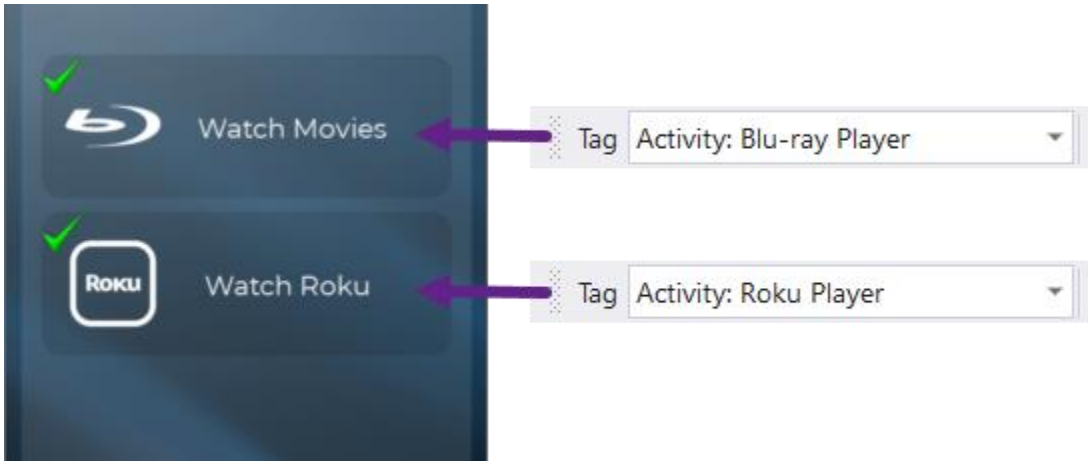
The older television has no driver support, and wiring to the TV location is not workable. While all other devices can be controlled through the processor, commands must be sent to the television via infrared from the controller.

1. Add the television to the room. For the **output port**, select **Standalone**.

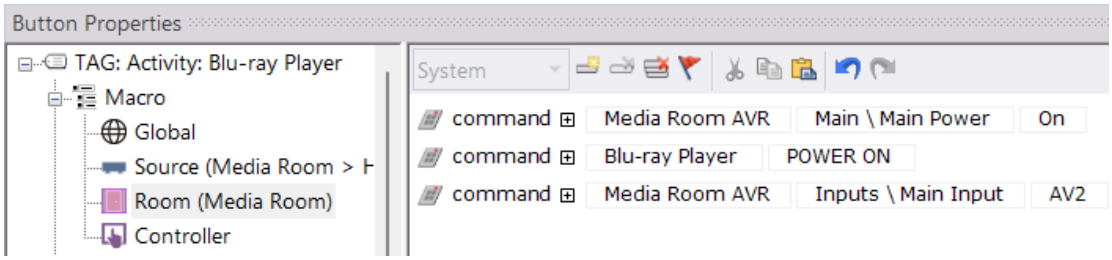


2. Add the rest of the devices in the room which does not require standalone control.

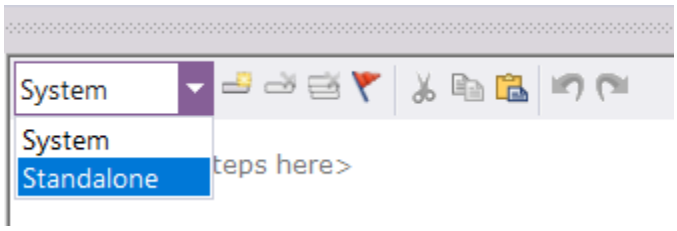
3. Since the system manager cannot be used, select the button menu option for the controller. Create the button menu on the controller and drag the activity tags to the activity buttons.



4. Create a **room macro** for the devices in **system mode** (**processor**.) Note that any activity commands auto-programmed from the tag referencing the system manager must be **removed**.

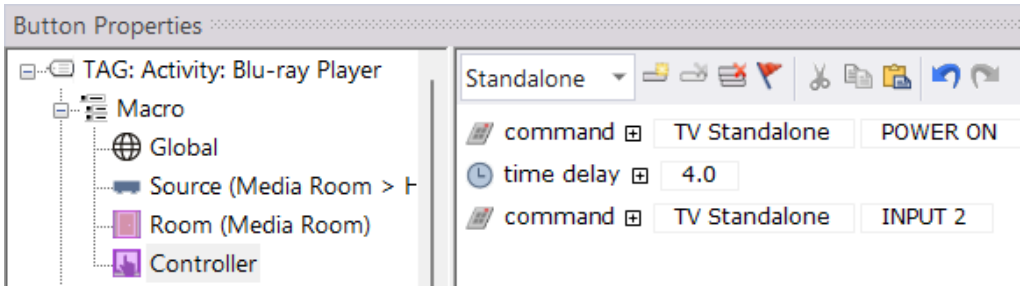


5. On the same button, select the **controller-level macro type** in the button properties. Where it shows the system mode, select **standalone** from the drop-down menu.



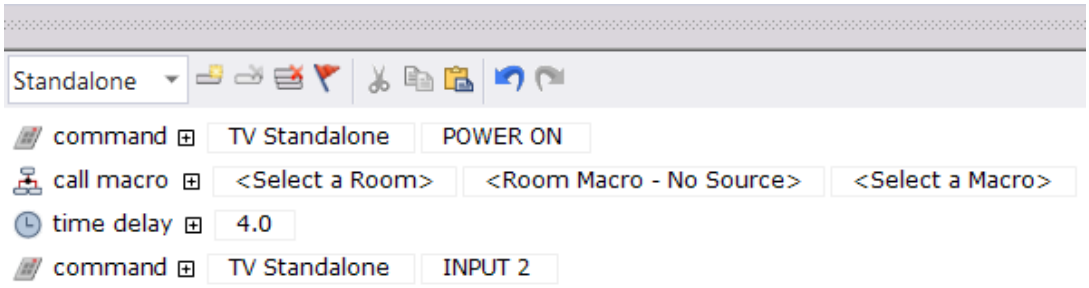
Only **standalone devices** will be available when selecting **Standalone** in the macro steps window.

6. **Program** the commands that need to run in **standalone mode** (from the controller.)

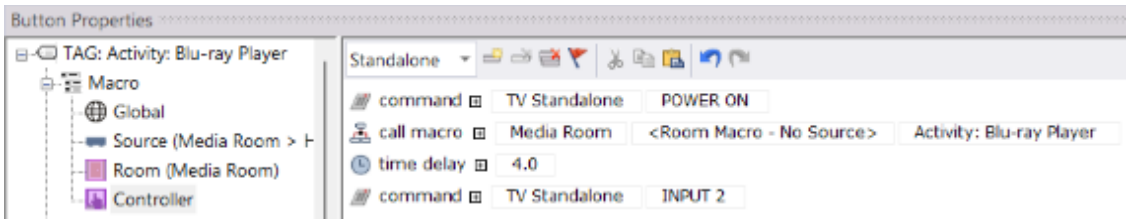


7. Two macro levels are programmed on the button, one in system mode and the other in standalone mode. While the controller macro has a higher priority than the room macro, the **room macro** can be **called** by the **controller macro**.

   Drag a "**Call Macro**" macro step between the TV power command and the delay.



8. For **<Select a Room>**, select the Media Room. Leave the **<Room Macro – No Source>** field as a source macro is not being programmed. Select the activity tag on the programmed button for **<Select a Macro>**.



   In this example, a controller macro sends standalone commands and calls a room macro, which runs in system mode. The room and controller macro share the same button and activity tag.

Programming Relays

XP Processors feature built-in low-voltage dry contacts that supply contact closure or switching control for loads up to 3A-5A/30VDC (external power supply needed). Relays can control curtains, drapes, shades, motorized screens, TV lifts, and other devices on the market today. All three relays are Normally Open when not energized, but they can be programmed to behave as Normally Closed if power is applied to the XP processor.

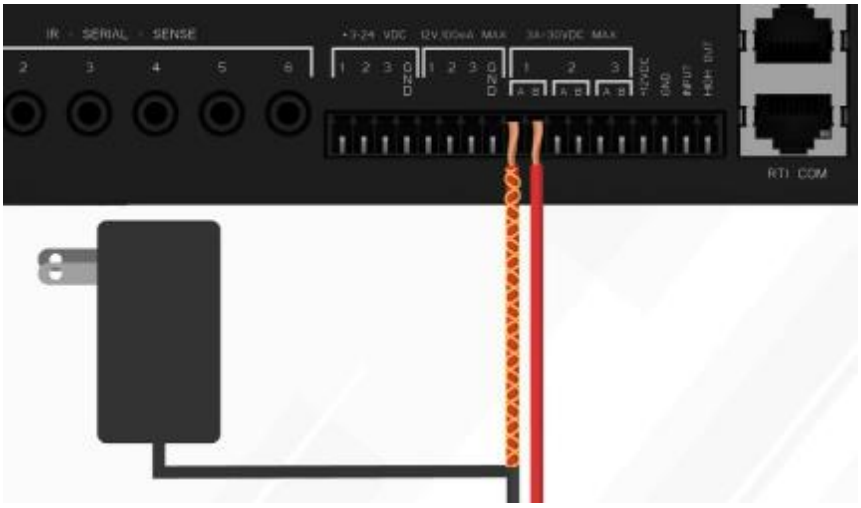
RELAY WIRING TYPES

CONTACT CLOSURE

Connect contact A and B contact closures to the desired device.

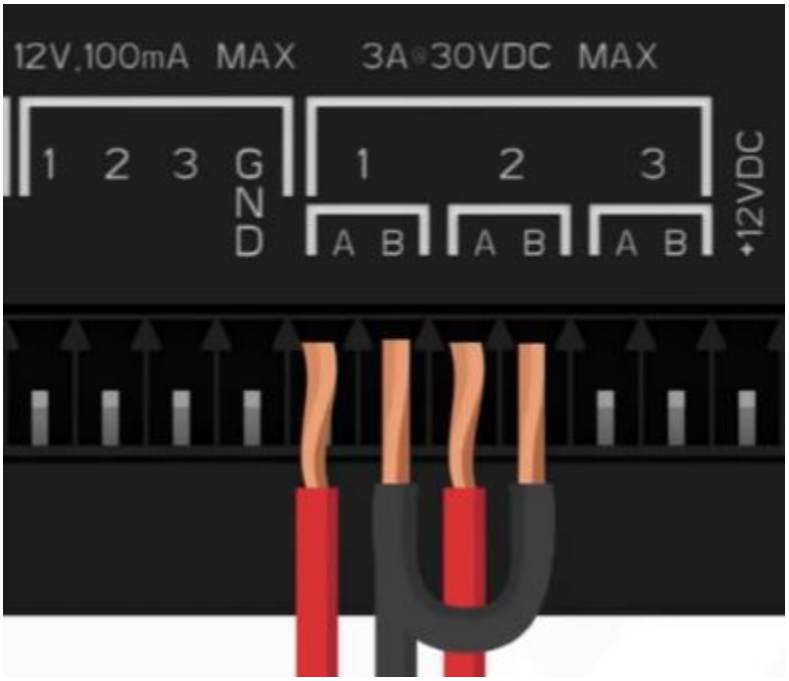
12V TRIGGER

A relay can be used as a voltage trigger by connecting the positive side of the power supply to contact A. Connect contact B and the ground side of the power supply to the desired device.



THREE-WIRE

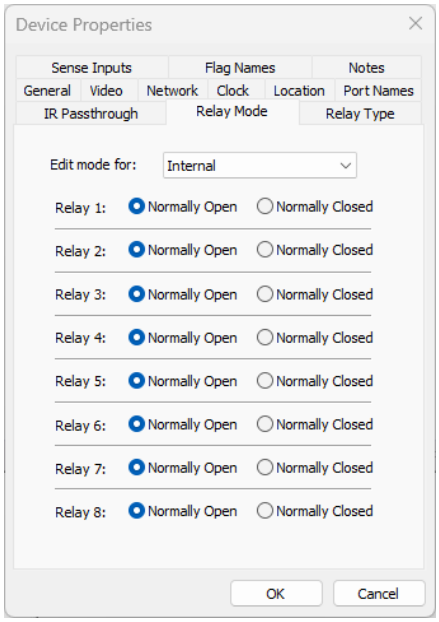
Some devices require multiple contact closures to trigger different functions. Therefore, multiple relays must be used. For example, a projector screen control requires a contact closure between wire A and ground to lower the screen and a separate contact closure between wire B and ground to raise the screen – in this example, you will need to create a loop for ground.



CONFIGURING A RELAY

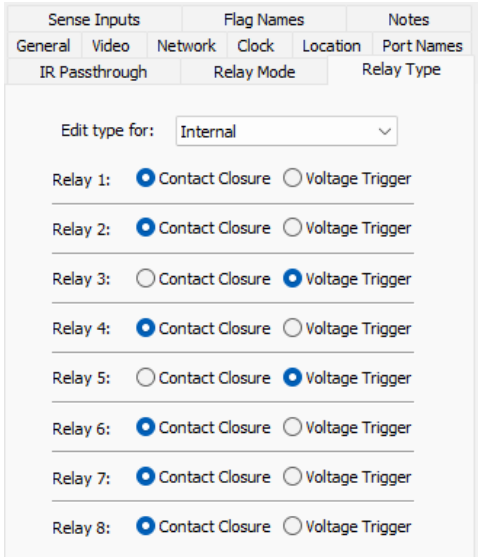
1. Select the **XP-8v processor** in the workspace, then select the **properties** tab.

2. Select the **Relay Mode** tab and configure the default state of the relays in use.

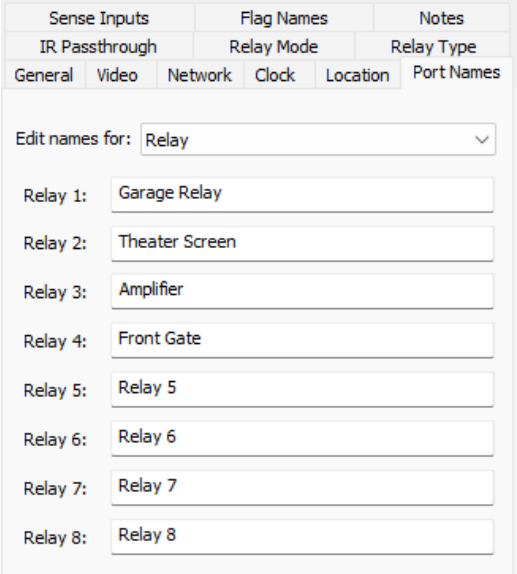


By default, relays are normally open. Please note that the relay states will revert to the open state if power is lost. Please plan accordingly.

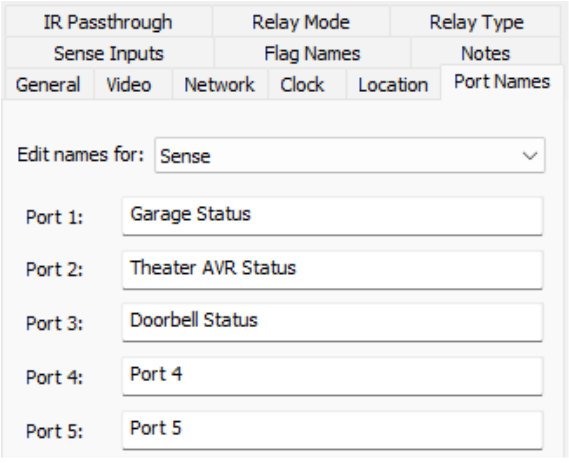
3. Select the **Relay Type** tab and configure the relays used as a contact closure or voltage trigger.

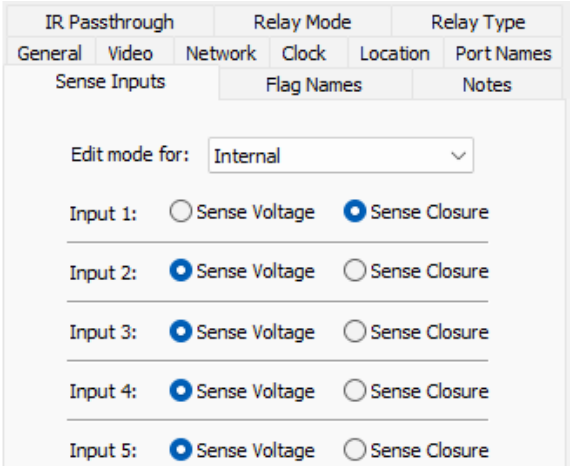


4. Select the **Port Names** tab and select **Relay** from the drop-down menu. Name the relays accordingly.



5. If using **sensory inputs** to detect if a relay should be open or closed, select the **Sense** option in the drop-down menu in the port names tab. Name each sensory port accordingly. Next, select the **Sense Inputs** tab and configure the sensory inputs to sense **voltage** or **closure**.



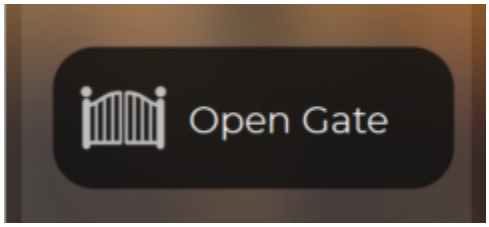


Pro Tip: Use sense inputs to track relay statuses and receive expected results. Sensing voltage and contact closure are useful when controlling garage doors, doors, drapery, and other devices.

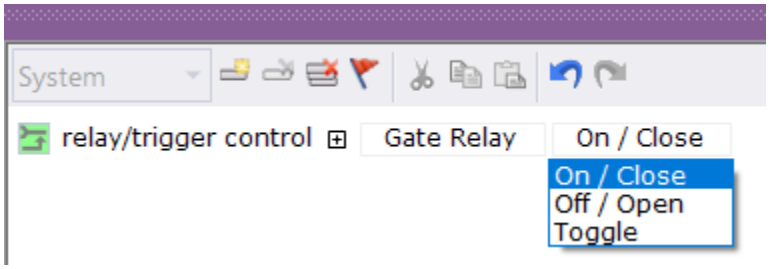
PROGRAMMING

Scenario: When visitors arrive, the client requests a button to open his front property gate. A **normally open contact closure** has been wired to the gate motor.

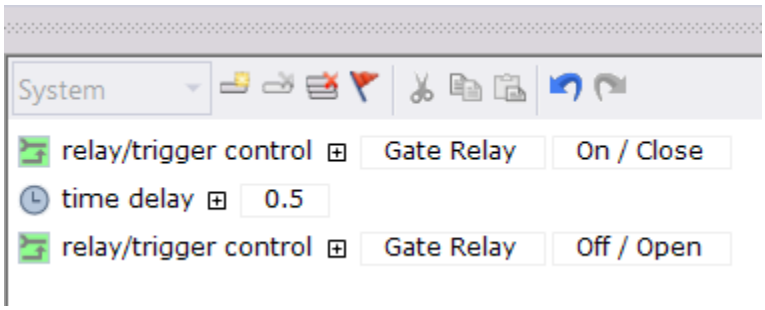
1. Create a button on the controller and properly tag it. Name the button “Open Gate.”



2. Create a macro on the button by selecting a macro level and dragging a relay command to the macro window.



3. When the relay is closed, the gate opens but immediately closes. The relay must be closed and quickly opened to keep the gate from closing immediately.

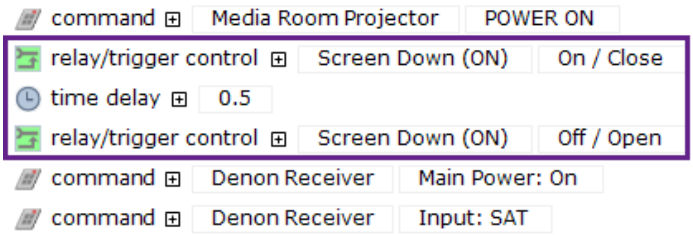


Using the right delay is critical to making the macro work properly. Experiment with different delays until the required result is achieved. The gate opens and stays open for a few minutes until the internal timer closes the gate.

Scenario: A conference room projector screen uses **a three-wire contact closure** wiring method. Relay 1 sends the screen down when the projector is turned on, and Relay 2 brings the screen up when the projector is turned off.

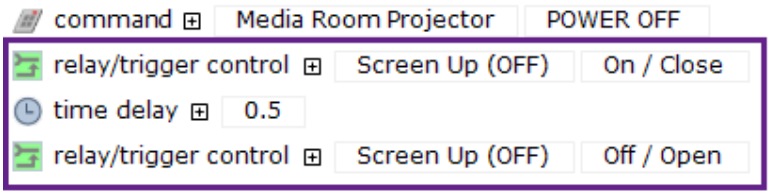
1. Create a Power On and Power Off button, tagging both buttons properly.

2. Program the **Power On** commands, then double-click or drag a relay command to the macro window. Place the relay commands in the early section of the macro for best practice.



The **close** command closes the relay and starts the flow of current required to drop the screen down. The **open** command stops the voltage flow as the screen descends to the lower limit. The delay between relay commands can be changed to get the right results.

3. Program the **Power Off** commands, then double-click or drag a relay command to the macro window. Place the relay commands in the early section of the macro for best practice.

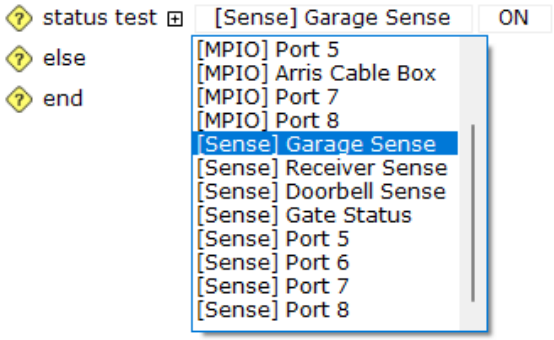


The second relay closes and starts the flow of current required to send the screen up. The open command stops the voltage flow while the screen ascends to its upper limit.

Scenario: Utilize a sensory input wired to a contact sensor on a garage door to determine if the door should be closed.

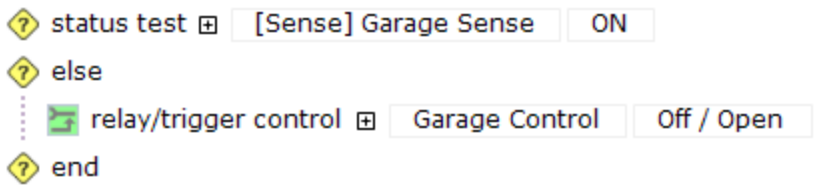
1. Create, tag, and program a Garage Open and Close button on the user interface.

2. Double-click or drag a Status Test macro step to the macro window after selecting a macro level from the button properties menu. Select the garage sensory port from the drop-down menu.



3. Add a **relay** command to open the garage if it is not open.

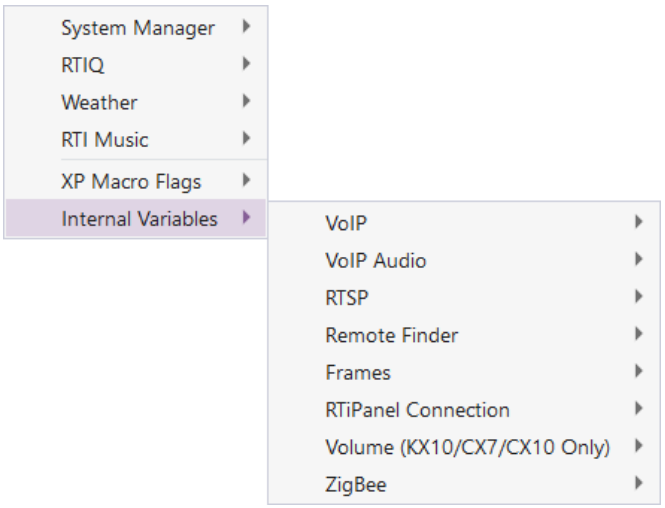
4. On the **close** button, create the opposite status test.



Internal Variables

In addition to driver variables, internal variables based on specific controller features are available for feedback.

Internal Variables may be found in the same location as drivers when programming variable feedback. Programming variables may be used for button states, dynamic images, gauges, toggles, and image lists.



VOIP

VOIP, or Voice over IP, is used for controllers compatible with an intercom feature. VOIP settings can be set with the VOIP object, and feedback for several VOIP data points is available for compatible controllers.

- **Auto Answer Enabled**: indicates whether auto-answer is enabled on the controller.
- **Auto Answer Ring Once Enabled**: indicates whether auto-answer ring once is enabled.
- **Caller**: a string containing the identification of the other party.
- **Call Duration**: the duration of the current call-in seconds format
- **Call Duration (MM: SS)**: the duration of the call in MM: SS format.
- **Call In Progress** indicates whether a call is in Progress.
- **DND Enabled**: indicates whether "**do not disturb**" is enabled.
- **Incoming call**: indicates whether a call is incoming.
- **Mic Level**: Feedback on the current microphone level of the controller.
- **Call Volume**: the current call volume level of the controller.
- **Ring Volume**: The volume of the call ring on the device.
- **Mute Enabled**: indicates whether the current call is muted.
- **On Hook**: indicates whether the device is on hook (opposite of Call In Progress.)
- **Rx Video Enabled**: indicates if the Rx video is being received and displayed.
- **Rx Video Auto Receive Enabled**: indicates if RX Video Audio Receive is enabled.
- **Rx Video Actual Frame Rate**: the frame rate of frames received from the other party.
- **Rx Video Actual Frame Width**: the actual width of frames received from the other party during a call, as negotiated with the other party on call setup.
- **Rx Video Actual Frame Height**: The actual height of frames received from the other party during a call, as negotiated with the other party on call setup.
- **Tx Video Enabled**: true indicates that Tx video is being sent.
- **Tx Video Auto Send Enabled**: If true, the video will be sent automatically when a call is instantiated.
- **TX Video Preferred Frame Width**: The controller's preferred video object frame width.
- **TX Video Preferred Frame Height**: The controller's preferred video object frame height.
- **TX Video Preferred Frame Rate**: The controller's preferred frame rate.
- **Tx Video Max Frame Width**: The current set outgoing video frame width.
- **Tx Video Max Frame Height**: The current set outgoing video frame height.
- **Tx Video Max Frame Rate**: The current set outgoing video frame rate.
- **Tx Video Actual Frame Width**: The actual width of frames sent to the other party during a call, as negotiated with the other party on call setup.

- **Tx Video Actual Frame Height**: The actual height of frames sent to the other party during a call, as negotiated with the other party on call setup.
- **Tx Video Actual Frame Rate**: the actual frame rate of frames sent to the other party.
- **SIP Registration - In Progress**: This indicates a SIP registration session is in progress.
- **SIP Registration – Succeeded**: This indicates a SIP registration session succeeded.
- **SIP Registration - Not Registered**: The SIP registrant is not registered.
- **SIP Registration – Failed**: The SIP registration session failed.

VOIP AUDIO

- **Rx Audio Level**: Feedback on the current incoming audio level.
- **Tx Audio Level**: Feedback on the current outgoing audio level.
- **Mic Suppressed**: Indicates if the microphone is suppressed because of echo suppression.
- **Half-Duplex Enabled**: A conversation may occur one way only if half-duplex is enabled.
- **Push To Talk Enabled**: indicates if push-to-talk is enabled.
- **AGC-Speaker enabled**: indicates if AGC is enabled for the speaker.
- **AGC-Mic enabled**: indicates if AGC is enabled for the microphone.
- **Echo Suppress Threshold Hit**: If true, it indicates that the incoming audio level is above the echo threshold, resulting in the microphone being disabled.
- **Echo Suppress Threshold**: Reports the echo suppression threshold.
- **Echo Suppress Mode**: Reports the current echo suppression mode (None, Automatic, or Fixed.)
- **Echo Suppress Enabled**: indicates if echo suppression is enabled (Automatic or Fixed.)
- **Silence Suppress Threshold Hit**: If true, it indicates that the outgoing audio level is below the silence threshold, resulting in the microphone being disabled.
- **Silence Suppress Threshold**: Reports the silence suppression threshold.
- **Silence Suppress Mode**: Reports the current silence suppression mode (None, Automatic, or Fixed.
- **Silence Suppress Enabled**: indicates if silence suppression is enabled (Automatic or Fixed.)

RTSP

The Real Time Streaming Protocol (RTSP) is a network control protocol for entertainment and communications systems to control streaming media servers. The protocol is used for establishing and controlling media sessions between endpoints. RTI Installers typically create user interfaces that capture this streaming media through surveillance camera feeds.

Connecting: The controller is connecting to an RTSP server.

Connected: The controller has successfully established a connection to the RTSP server.

Connection Failed: The controller failed to connect to the RTSP server.

Authentication Failed: The controller can not connect to the RTSP server due to an authentication issue.

Video Packets Lost: Packet loss occurs when network congestion, hardware issues, bugs, and several other factors cause dropped packets during data transmission.

REMOTE FINDER

Remote finder is a feature available for RTiPanel devices to locate SURFiR controllers by making them emit a beeping sound.

Find in Progress: A session has been initiated to find the controller.

Finder State: The status of the search for the controller.

FRAMES

Frame Count: The total number of frames on the controller. Up to ten frames are supported on compatible controllers.

Current Frame: The current frame indicates the current frame position on the controller.

Frame # (1-10): A variable that indicates if the frame number is currently selected on the controller. Up to ten frames are supported on compatible controllers.

RTIPANEL CONNECTION

Connected System Name: Feedback providing the name of the current RTiPanel project.

Connected Address: Feedback providing the IP address of the RTiPanel device.

Connection Type: Feedback that provides the connection type of the RTiPanel. The connection type may be LAN (Local Area Network,) WAN-Wi-Fi (Wide Area Network connected through Wi-Fi,) or WAN-Cell (Wide Area Network connected through a cellular signal.)

Connected via LAN: This shows the RTiPanel is connected through the LAN or local area network.

Connected via WAN-Wi-Fi: This shows the RTiPanel is connected through the WAN (Wide Area Network) using a wireless network.

Connected via WAN-Cell: Indicates the RTiPanel is connected through the WAN (Wide Area Network) using a cellular signal.

Not Connected: This status variable supplies feedback on the RTiPanel connection state.

VOLUME

Device Volume: A volume level indicator for the current device that supports volume commands through objects.

ZIGBEE

Channel: Indicates the current selected Zigbee channel, which is usually auto-selected when establishing a Zigbee network.

RSSI is the received signal strength indicator between the Zigbee transceiver and the controller. It is used to troubleshoot or determine whether the Zigbee signal is strong enough to communicate properly.

-40 to -69: Excellent signal.

-70 to -79: Good signal.

-80 to -99: Poor single.

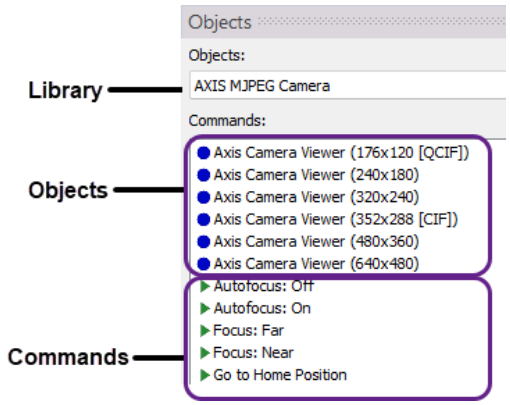
-100: No signal is Present.

LQI: The link quality index measures the quality of a Zigbee mesh network in terms of packet loss.

Objects

The Object tab of the Library Browser in the tabbed menu allows you to assign objects (IP camera, VoIP, video) and the commands to control these objects to the buttons in your device by dragging them in the user interface. The name of the current object library is listed at the top of the Object window and can be changed by pulling down the Objects list. The Objects and Commands in the library are listed in the command list.

Objects should not be confused with commands. An object will be represented with a blue circle, while commands with a green arrow.

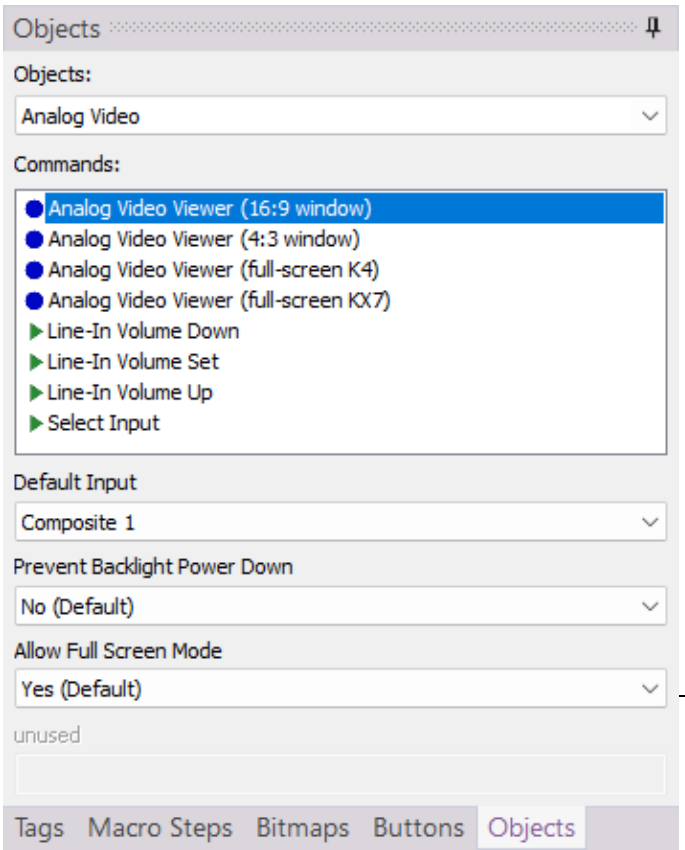


ADDING AN OBJECT

To add an Object, click the object name (those denoted with a blue circle) and drag it into the user interface. Some objects require more information, which can be entered in the fields at the bottom of the list. Objects create a new button that cannot be embedded in an existing one. For decorative purposes, a bitmap may be placed behind the object.

ADDING A COMMAND

To add a command to a button, click the command name in the list and drag it onto it. You can enter the command in the fields below the Commands list if it requires additional information. Certain commands will create a button when you drag them into an interface.



ANALOG VIDEO

Analog Video Viewer (16:9 window)

An object viewer is used for a 16:9 video display on supported legacy controllers.

Analog Video Viewer (4:3 window)

An object viewer is used for a 4:3 video display on supported legacy controllers.

Analog Video Viewer (full-screen)

An object viewer is used for a full-screen video display on supported legacy controllers.

Line-In Volume Down/Set/Up

A command adjusts the line-level volume control on supported legacy controllers.

Select Input

A command is used to select the video input on supported legacy controllers.

Default Input

Select the input of the supported legacy device. Composite 1, 2,3, and 4, Component, S-Video 1, S-Video 2, and Camera on compatible controllers are available.

Prevent Backlight Power Down

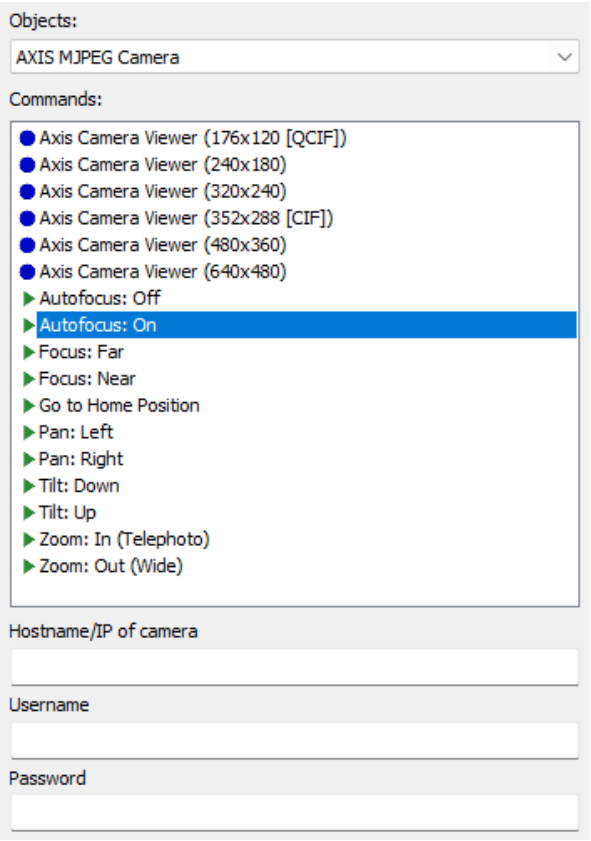
An option that will not power down the backlight after a set time when viewing the video. Available options are yes and no(default).

Allow Full-Screen Mode

Select yes(default) or no to allow full-screen mode on compatible controllers.

AXIS CAMERA VIEWER

An object was created to view available Axis camera resolutions with supported commands.



Axis Camera Viewer (Resolution)
These are specialized Axis objects with different resolutions depending on the controller used. Fill out the information at the bottom to configure.

Autofocus: (On and Off)
Turn autofocus on or off through a button command.

Focus: (Far and Near)
Commands are available to focus far or near.

Go to Home Position
Return to the camera's home position after panning.

Pan (Left, Right, Down, Up)
Use these commands to pan the camera view in any direction.

Tilt (Down and Up)
Use these commands to tilt the camera view down or up.

Zoom (In and Out)
Use these commands to zoom In (telephoto) or out (Wide.)

Hostname/IP of the camera
Enter the camera DNS or IP address in this field.

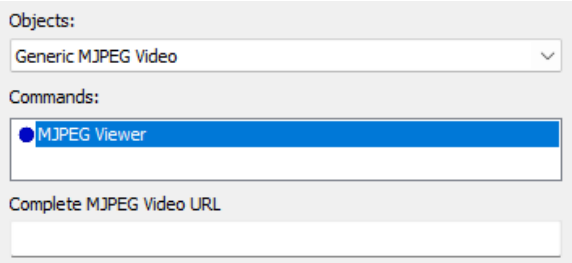
Username

Enter the username of the axis account.

Password
Enter the password of the axis account.

GENERIC MJPEG VIDEO

Use this object to view the MJPEG video format for IP and analog cameras.



MJPEG Viewer

Drag this object to the user interface and resize it properly after entering the full video URL. The URL usually includes the IP address or hostname, login and password, camera port, and frame rate.

Complete MJPEG Video URL

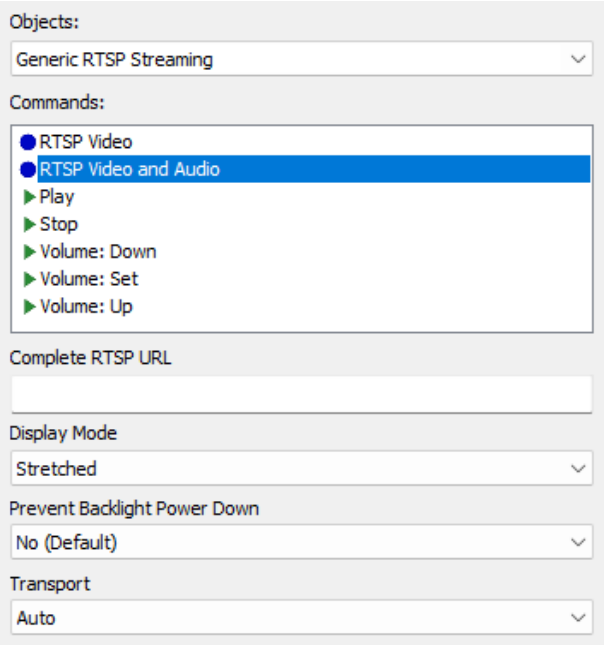
Enter the complete URL of the MJPEG video feed. Please refer to the manufacturer's documentation to find the proper URL.

GENERIC RTSP STREAMING

The RTSP generic streaming object can capture RTSP video transmissions from most camera manufacturers. Drag this object to the user interface after entering critical information and resize the video as needed.

Please note that RTI controllers support different H.264 profiles. When implementing a camera on multiple controllers, check RTI tech notes and product information on supported profiles.

Pro Tip: Certain camera systems require configuration settings on the camera or NVR to accommodate RTSP feeds on the RTI controller. Settings may include H.264, RTSP, substreams, and bit rates. Please read the documentation included with the third-party camera system carefully to review the default settings.



RTSP Video

Drag this object to a user interface to view an RTSP video feed.

RTSP Audio and Video

Drag this object to a user interface to view an RTSP video feed with audio.

Play and Stop

Play or Stop the RTSP video feed.

Volume: Down, Set, and Up

Control the audio volume of the RTSP feed.

Complete RTSP URL

Enter the RTSP URL from the manufacturer documentation. The RTSP URL could include login, password, camera, and profile information. If the URL is not in the manufacturer documentation, refer to applications such as VLC Media Player and ONVIF Device Manager for assistance.

Display Mode

Select the video display of the RTSP feed. Available options are stretched and centered.

Prevent Backlight Power Down

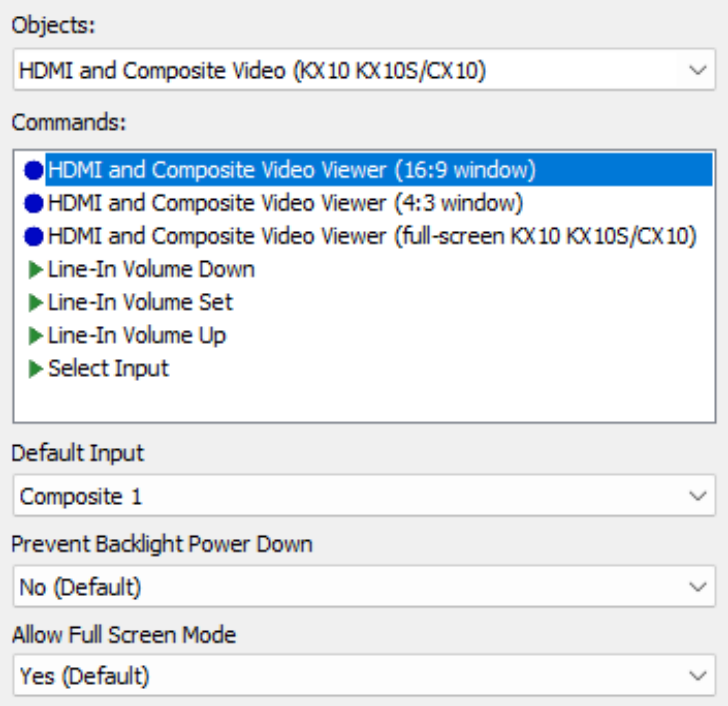
Select Yes or No(Default) to prevent the controller backlight from going down during an RTSP video transmission.

Transport

RTSP uses a Transmission Control Protocol (TCP) connection to control the streaming media session, although it is also possible to use UDP. Select TCP or UDP from the transmission transport options.

HDMI AND COMPOSITE VIDEO

Supported RTI controllers can output video through various connections for viewing on the controller screen.



HDMI and Composite Video Viewer (16:9 window)

Drag this object to the interface on compatible controllers to create a 16:9 window for viewing video connections.

HDMI and Composite Video Viewer (4:3 window)

Drag this object to the interface of compatible controllers to create a 4:3 window for viewing video connections.

HDMI and Composite Video Viewer (full-screen window)

Drag this object to the interface of compatible controllers to create a full-screen window for viewing video connections.

Line-In Volume Up, Down, and Set

Control the line-level volume of compatible controllers by dragging these commands to buttons.

Select Input

Use this command to select the input for compatible controllers with multiple inputs. Options include Composite 1, HDMI, and the camera.

HTTP REQUEST

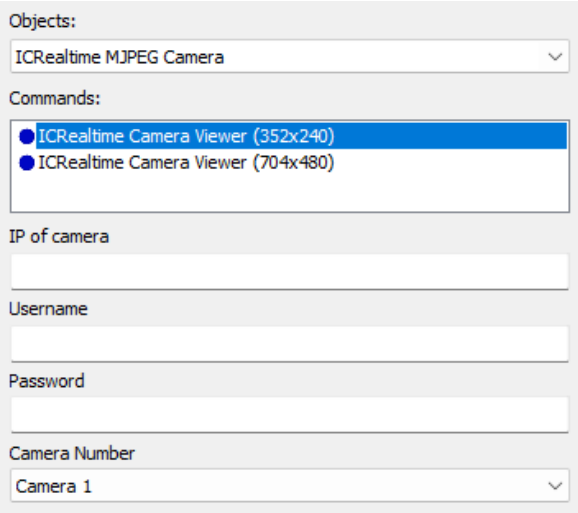
The HTTP Request object is used to issue arbitrary HTTP requests. Many IP-controllable devices can be controlled by requesting specially formatted URLs from them. For example, if you use the "Generic MJPEG Viewer" object to display video, you can use the HTTP Request objects to send pan, tilt, and zoom commands to the IP camera. Any data returned by the device in response to the request is discarded.

Request URL

For the Request URL command, enter the command URL and drag to a button. Most commands start with an IP address and include a port and protocol. If unsure what the URL command is, check with the device manufacturer.

ICREALTIME MJPEG CAMERA

The IC Realtime camera object is customized to work with IC Realtime surveillance cameras using MJPEG compression. Commands that control the cameras can also be assigned to buttons on the touchscreen interface.



Camera Viewer (352x240)

Use this object for smaller controllers with a maximum MJPEG video viewer of 352x240.

Camera Viewer (704x480)

Use this object for larger controllers with a maximum MJPEG video viewer of 704x480.

IP of camera

Enter the IP address of the camera on the network.

Username and Password

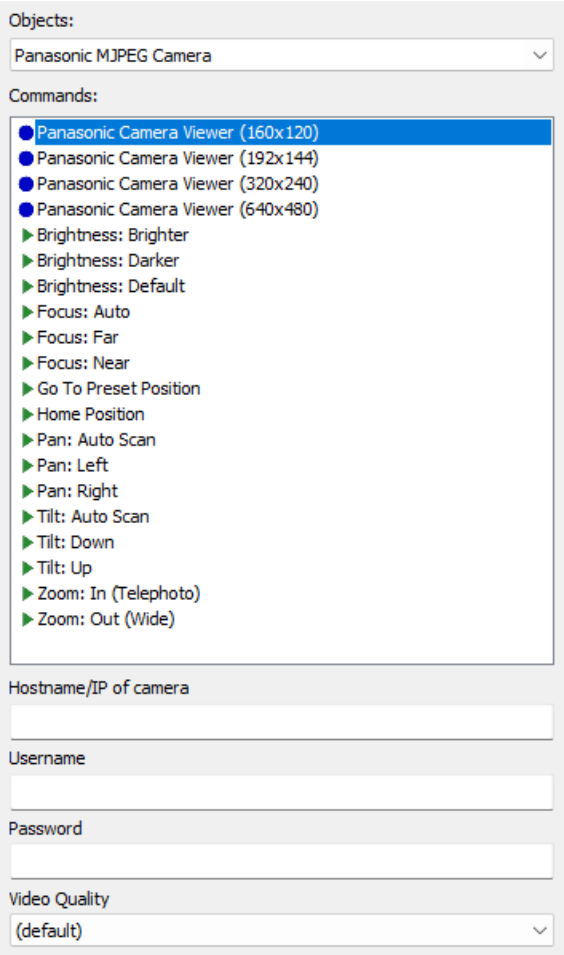
Enter the username and password of the camera account.

Camera Number

Select the camera number for the video feed.

PANASONIC MJPEG CAMERA

The Panasonic MJPEG camera object is customized to work with IC Realtime surveillance cameras using MJPEG compression. Commands that control the cameras can also be assigned to buttons on the touchscreen interface.



Panasonic Camera Viewer (160x120)

Use this object on smaller controllers with a maximum video viewer size of 160x120.

Panasonic Camera Viewer (192x144)

Use this object on smaller controllers with a maximum video viewer size of 192x144.

Panasonic Camera Viewer (320x240)

Use this object on controllers with a maximum video viewer size of 320x240.

Panasonic Camera Viewer (640x480)

Use this object on larger controllers with a maximum video viewer size of 640x480.

Brightness (Brighter, Darker, Default)

Control the brightness level on the camera or use the default settings.

Focus (Auto, Far, Near)

Control the focus of the camera by entering the focus options.

Go to Preset Position

Use presets to go to a certain position. This command requires entering the preset number in the Preset Number field.

Pan (Auto Scan, Left, Right)

Pan the camera using the options available.

Tilt (Auto Scan, Down, Up)

Tilt the camera viewing angle using the tilt options.

Zoom In (Telephoto)

Zoom in using the zoom feature of the camera.

Zoom Out (Wide)

Zoom out using the zoom feature of the camera.

Hostname/IP of Camera

Enter the DNS or IP address of the camera.

Username and Password

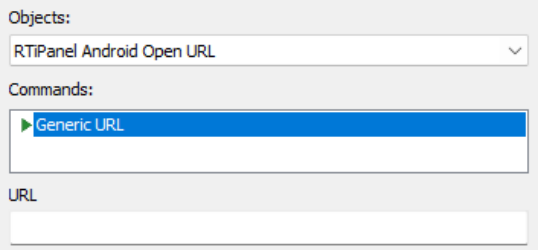
Enter the username and password to access the camera video feeds.

Video Quality

Select the video quality of the camera view. Options include default, standard, favor motion, and favor clarity.

RTIPANEL ANDROID OPEN URL

The RTiPanel Android “Open URL command”, will jump to an Android app based on the entered URL. URL can be found in the Google Play store.

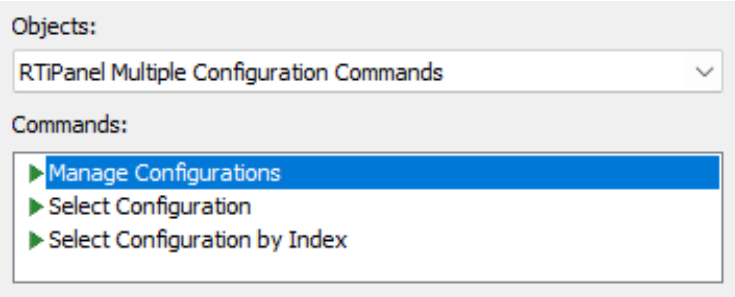


Generic URL

Enter the URL field of the Android app and drag it to a button on Android-based RTiPanel devices.

RTIPANEL MULTIPLE CONFIGURATION COMMANDS

RTiPanel commands provide buttons on an RTiPanel to access other locations without resynchronizing the device repeatedly to access other homes and businesses.



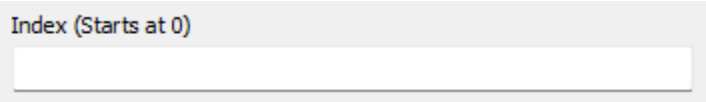
Return to Welcome Screen

This command will return the RTiPanel to the welcome screen, where other locations can be selected, edited, added, or deleted.

Switch System

When placed on a button, this command will display a popup menu of available locations. Locations can be switched easily without leaving the interface.

Switch to a Specific System

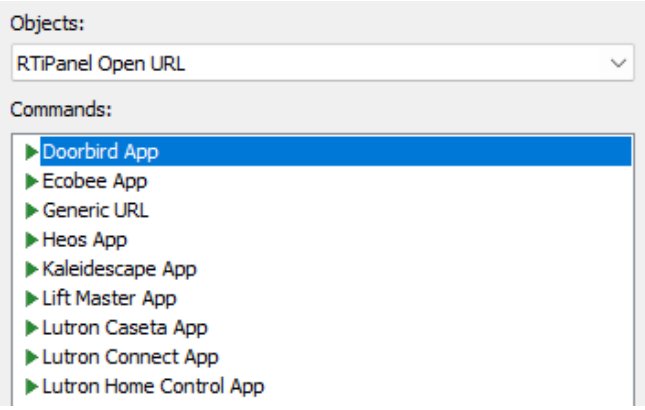


Enter the index number of the location for quick transitioning to a specific location. The index is based on the location order in the location list.

Pro Tip: Icons for these features can be found by searching “Switch” in the coral icons bitmap library. Avoid issues and give end users a “Switch System” command to access other venues quickly without exiting the app.

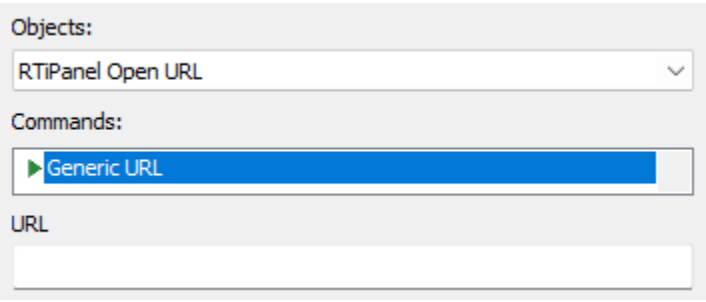
RTIPANEL OPEN URL

When placed on a button, these commands will jump to another iOS-based app when using the RTiPanel for control. A list of commonly used apps has been compiled and made available without having to locate the URL.



Dozens of canned commands for popular apps are provided and can be placed on any button on RTiPanel iOS-based devices. A return URL is often built into the URL to take the end user back to the RTiPanel control session.

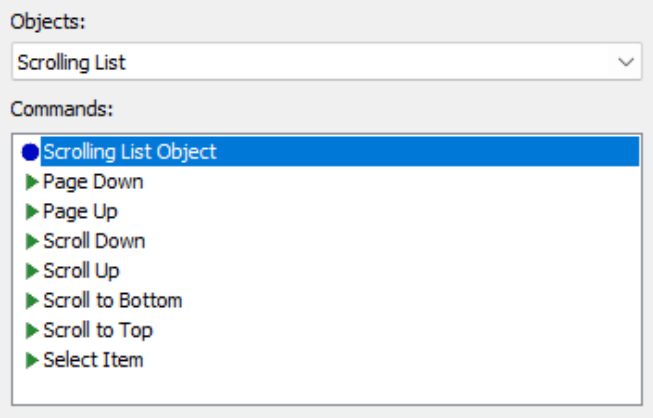
Generic commands can also be entered by selecting the **Generic URL** option.



In the best-case scenario, the URL name has a specified return URL, which can return the user to the RTiPanel control session without selecting the app again. Not all apps have published return URL information. URL information can be found in various online resources and technical support documents.

Scrolling List (One-Way)

A scrolling list object is typically used for presets, channel favorites, and other selectable items. These objects can be sent to the user interface and configured with various options to build and customize the list. Many aspects of the list may be controlled, including list size, properties, icon appearance, macros, and other features. When using the scrolling list object, be sure to accompany the list with the commands that allow the navigation and selection of items in the list.

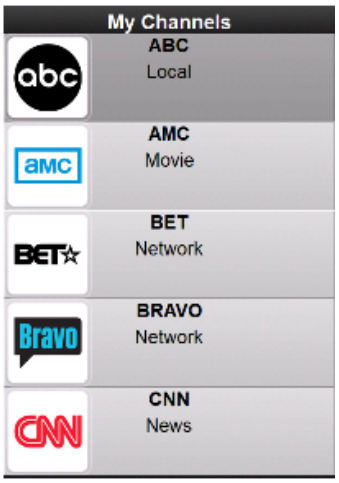


Scrolling List Object

Drag the scrolling list object to the screen and right-click to add buttons, icons, identifiers, and macros. Configure the list properties and appearance and icon appearance.

List Properties | List Appearance | Icon Appearance | ✓ Close Editor

Item	Tag	Button	Icon	Macro	Line 1	Line 2
1	Preset - ABC			Yes	ABC	Local
2	Preset - AMC			Yes	AMC	Movie
3	Preset - BET			Yes	BET	Network
4	Preset - Bravo			Yes	BRAVO	Network
5	Preset - CNN			Yes	CNN	News
6	Preset - ESPN			Yes	ESPN	Sports
7	Preset - ESPN 2			Yes	ESPN2	Sports
8	Preset - Food Network			Yes	FOOD	Network
9	Preset - HBO			Yes	HBO	Movies
10	Preset - HGTV			Yes	HGTV	Education



Page (Up, Down)

Page up and down in the list to quickly browse the list.

Scroll (Up, Down)

Scroll up and down in the list one item at a time.

Scroll to (Bottom, Top)

A quick jump to the bottom or top of the list.

Select Item

Select the currently highlighted button.

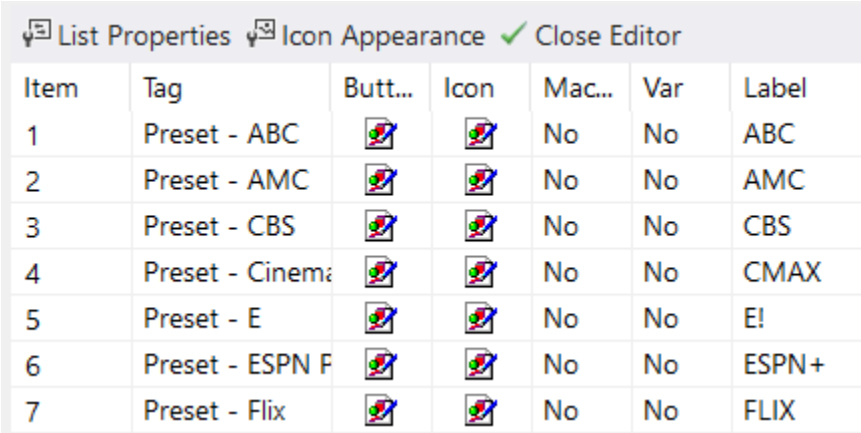
Note: One-way scrolling lists are not compatible with all controllers. While the object may not be used on RTiPanel and KA-series controllers, a one-way list can be configured to reflect limited feedback.

To draw a one-way list on RTiPanel or KA-series controllers:

1. Select the draw button icon in the tool palette
2. Select the one-way list icon in the button styles toolbar.

3. Hold down the mouse and draw the list on the user interface. It can be resized and positioned in selection mode.

4. Configure the list properties and icon appearance options in the scrolling list.



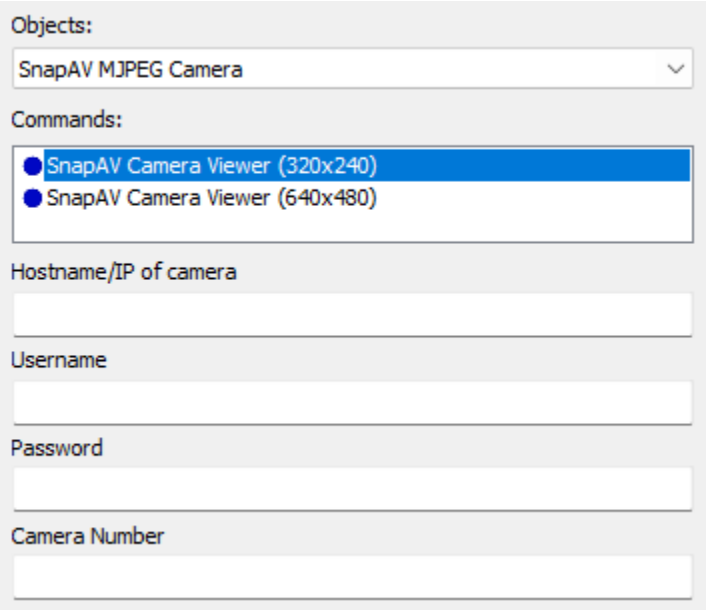
Item	Tag	Butt...	Icon	Mac...	Var	Label
1	Preset - ABC			No	No	ABC
2	Preset - AMC			No	No	AMC
3	Preset - CBS			No	No	CBS
4	Preset - Cinema			No	No	CMAX
5	Preset - E			No	No	E!
6	Preset - ESPN F			No	No	ESPN+
7	Preset - Flix			No	No	FLIX

5. Make any color changes to the list using the button styles toolbar options.

Note: Navigation and selection can be done using gesture control without requiring commands.

SNAPAV MJPEG CAMERA

The SnapAV MJPEG camera object captures MJPEG video for Snap One surveillance products.



SnapAV Camera Viewer (320x240)

This object provides a camera viewer for controllers with smaller screen resolutions.

SnapAV Camera Viewer (640x480)

This object provides a camera viewer for controllers with larger screen resolutions.

Hostname/IP of the camera

Enter the DNS or IP address of the camera (port may be required.)

Username and Password

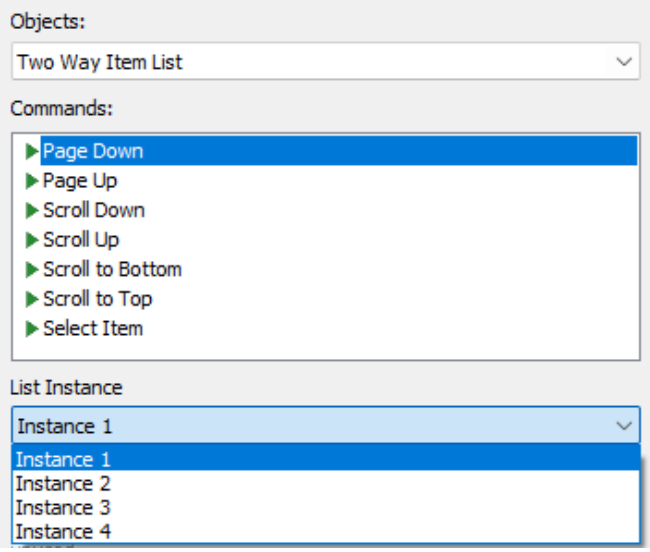
Enter the username and password of the camera account.

Camera Number

Enter the camera number for viewing.

TWO-WAY ITEM LISTS

Two-way item lists are used to navigate through a two-way item list and select items in the list. Not all controllers require two-way item list commands.



Page (Down, Up)

Page down and up the item list, a page at a time.

Scroll (Down, Up)

Scroll down and up the item list, one item at a time.

Scroll to (Bottom, Top)

Scroll to the bottom or top of the item list.

Select Item

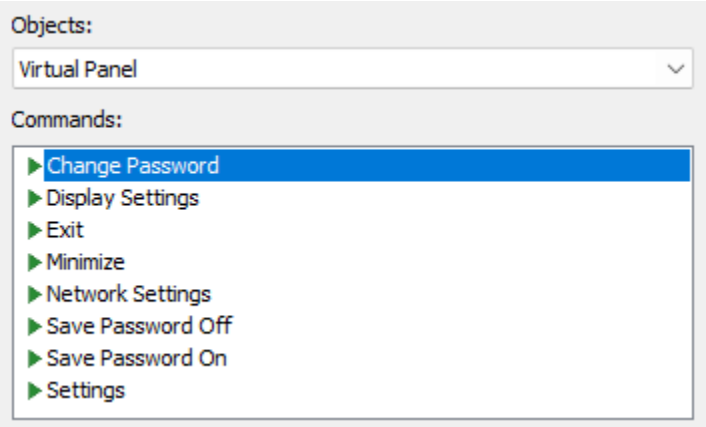
Select the highlighted item in the item list.

List Instance

When using additional item lists on a controller or page, use additional instances for the list and commands. Up to four instances can be used per controller.

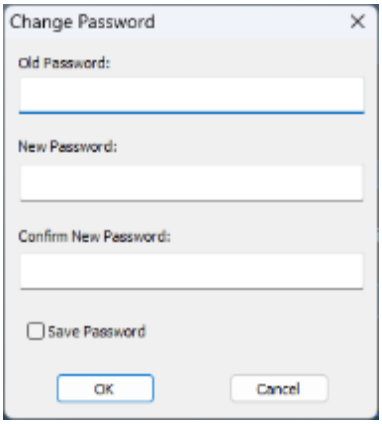
VIRTUAL PANEL

Special commands are designed for Virtual Panel controllers. These commands should be placed on the virtual panel to control settings and options.



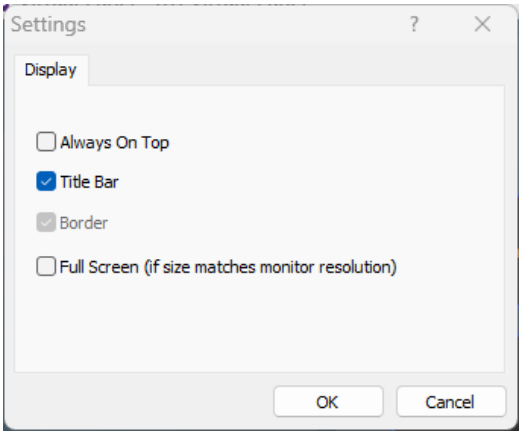
Change Password

If the Virtual Panel uses a password required for starting a session, it may be changed using the change password command.



Display Settings

This command accesses the display settings of the Virtual Panel.



Exit

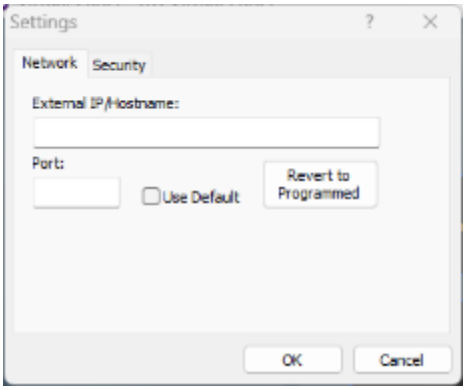
Close the Virtual Panel and exit the control session.

Minimize

Minimize the Virtual Panel in the tray.

Network Settings

Access the network settings and security configuration options of the Virtual Panel.

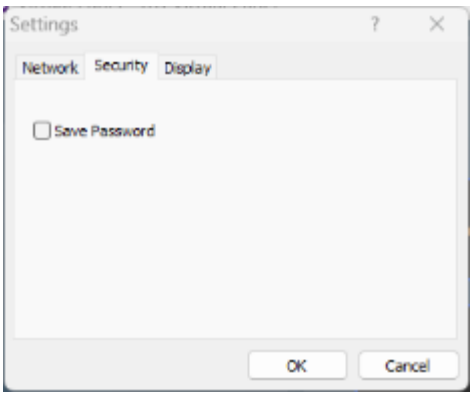


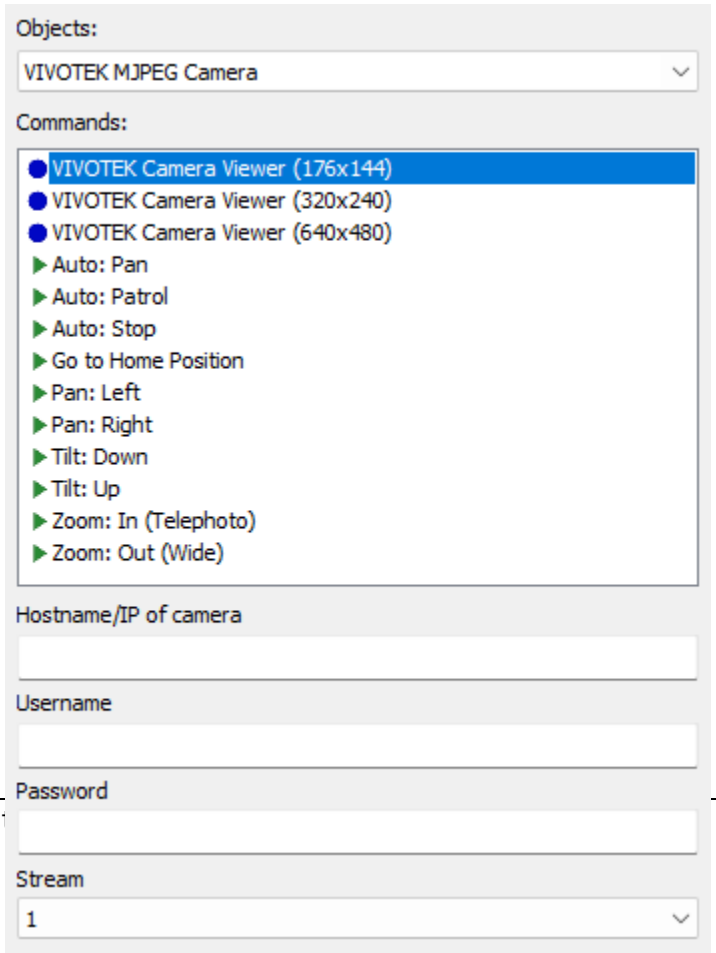
Save Password (Off, On)

This command will turn on and off the save password options of the Virtual Panel.

Settings

Access the settings tab of the Virtual Panel.





VIVOTEK MJPEG CAMERA

These objects and commands enable viewing the MJPEG video streams on the VIVOTEK camera and surveillance products.

VIVOTEK Camera Viewer (176x144)

This object provides a camera viewer for controllers with smaller screen resolutions.

VIVOTEK Camera Viewer (320x240)

This object provides a camera viewer for controllers with smaller screen resolutions.

VIVOTEK Camera Viewer (640x480)

This object provides a camera viewer for controllers with smaller screen resolutions.

Please enter the required information for each object before adding it to the controller.

Auto (Pan, Patrol, Stop)

These commands are auto features available for VIVOTEK cameras.

Go to Home Position

Return to the home position camera angle.

Pan (Left, Right)

A camera pan is a horizontal movement in which the camera moves right to left (or vice versa) from a central axis.

Tilt (Down, Up)

Tilt the camera up or down based on the view.

Zoom In (Telephoto)

Zoom in on the current view.

Zoom Out (Wide)

Zoom out on the current view.

Hostname/IP of the Camera

Enter the DNS or IP address of the camera. Port information may be required.

Username and Password

Enter the username and password of the designated user on the camera account.

Stream

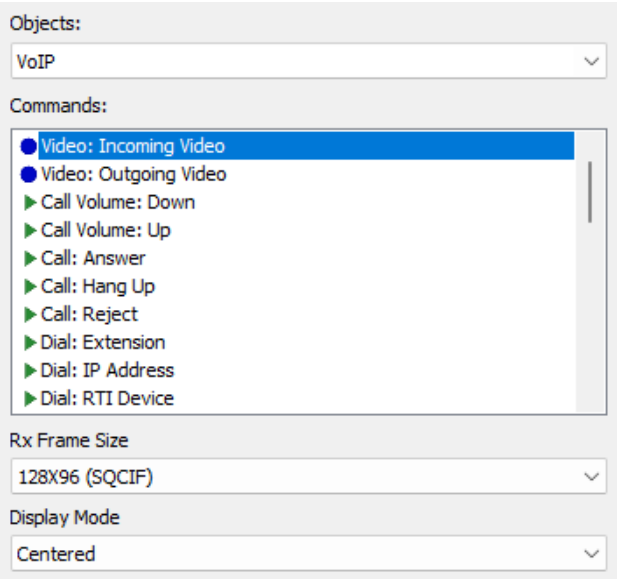
Enter the stream number for the camera view.

VOIP

VoIP (Voice over IP) objects and commands facilitate intercom features, allowing RTI installers to communicate with other RTI controllers and third-party door stations. Please check the specifications of the RTI controller for compatible features.

RTI controllers have different options and features that can differ from model to model. Please consult the documentation, as some objects and commands may not be available on a controller even if VoIP is supported.

RTI recommends using a SIP server if communicating between more than two controllers. A SIP server will facilitate communication and provide a much better experience for the end user.



Video: Incoming Video

This object defines the area to place incoming video. An Rx Frame Size and Display Mode must be provided.

Video: Outgoing Video

This object defines the area to place outgoing video.

Call Volume: Down

This command will lower the volume of a call.

.

Call Volume: Up

This command will raise the volume of a call.

Call: Answer

This command will answer an incoming call.

Call: Hang Up

This command will terminate an existing call.

Call: Reject

This command will reject an incoming call.

Dial: Extension

This command will dial a device via the SIP server. A SIP server is required to utilize this feature and must be registered.

Dial: IP Address

This command will dial the device with a provided IP address.

Dial: RTI Device

This command will dial a VoIP-capable RTI device.

Mic: Toggle Mute

This command will toggle the mute status of the microphone.

Ring Volume: Down

This command will lower the ring volume.

Ring Volume: Up

This command will raise the ring volume.

Send DTMF

This command will send DTMF (Dual Tone Multi-Frequency) characters to the other party. A DTMF string, Duration (milliseconds), and Mode must be provided.

Settings: Toggle Auto Answer

This command will toggle the auto-answer state.

Settings: Toggle Auto Answer Ring Once

This command will toggle the auto answer ring one state.

Settings: Toggle Do Not Disturb

This command will toggle the do not disturb state.

Video: Disable

This command will turn off video transmission.

Video Enable

This command will enable video transmission.

Video RX Video: Auto Receive Disable

This command will turn off the auto-receive for the received video.

Video RX Video: Auto Receive Enable

This command will enable the auto-receive for the received video.

Video RX Video: Auto Receive Toggle

This command will toggle the auto-receive state.

Video: Rx Video: Disable

This command will turn off the received video.

Video: Rx Video: Enable

This command will enable the received video.

Video: Rx Video: Toggle

This command will toggle the received video state.

Video: Set Preferred TX Frame Rate

This command will send the preferred transmit frame rate. A Tx Frame Rate must be provided.

Video: Set Preferred TX Frame Size

This command will send the preferred transmit frame size. A size resolution must be provided.

Video: Switch Camera

This command will switch the camera view.

Video: Toggle

This command will toggle the video state.

Video: Tx Video: Auto Send Disable

This command will turn off the auto-sending of the transmitted video.

Video Tx Video: Auto Send Enable

This command will enable the auto-sending of the transmitted video.

Video Tx Video: Auto Send Toggle

This command will toggle the auto-sending of the transmitted video state.

Video: Tx Video: Disable

This command will turn off the outgoing or transmitted video.

Video: Tx Video: Enable

This command will enable the outgoing or transmitted video.

Video: Tx Video: Toggle

This command will toggle the outgoing or transmitted video state.

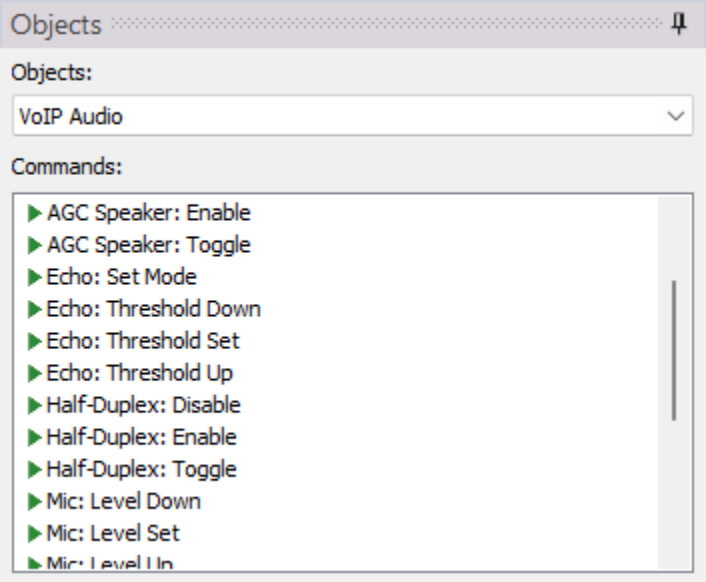
Volume: Set

This command will adjust the call volume to a value that must be provided.

VOIP AUDIO

Compatible RTI controllers have VoIP Audio commands that can improve the quality of communication while addressing environmental issues in the field. Please note not all RTI VoIP controllers have VoIP Audio settings. Please plan

projects accordingly after consulting the product documentation and programming features.



AGC Mic: Disable

This command will disable Automatic Gain Control for outgoing audio.

AGC Mic: Enable

This command will enable Automatic Gain Control for outgoing audio.

AGC Mic: Toggle

This command will toggle the state of Automatic Gain Control for outgoing audio.

AGC Speaker: Disable

This command will disable Automatic Gain Control for incoming audio.

AGC Speaker: Enable

This command will enable Automatic Gain Control for incoming audio.

AGC Speaker: Toggle

This command will toggle the state of Automatic Gain Control for incoming audio.

Echo: Set Mode

This command will set the echo suppression mode (None, Automatic, or Fixed.)

Echo: Threshold Down

This command will decrease the echo suppression threshold (Fixed mode only.)

Echo: Threshold Set

This command will adjust the echo suppression threshold to the given value (Fixed mode only.)

Echo: Threshold Up

This command will increase the fixed echo suppression threshold (Fixed mode only.)

Half-Duplex: Disable

In half-duplex mode, audio is sent in both directions, but only one party at a time can send it. This command will turn off half-duplex mode.

Half-Duplex: Enable

In half-duplex mode, audio is sent in both directions, but only one party at a time can send it. This command will enable half-duplex mode.

Half-Duplex: Toggle

In half-duplex mode, audio is sent in both directions, but only one party at a time can send it. This command will toggle the half-duplex mode state.

Mic: Level Down

This command will decrease the microphone level.

Mic: Level Set

This command will set the microphone level to a given value.

Mic: Level Up

This command will increase the microphone level.

Push to Talk: Disable

This command will turn off the push-to-talk feature.

Push to Talk: Enable

This command will enable the push-to-talk feature.

Push to Talk: Talk

This command requires a push and hold to talk. The mic is enabled, and the speakers are disabled.

Push to Talk: Toggle

This command toggles the push-to-talk feature state.

Silence: Set Mode

This command sets the silence suppression mode (None, Automatic, or Fixed.)

Silence: Threshold Down

This command decreases the silence suppression threshold (Fixed mode only.)

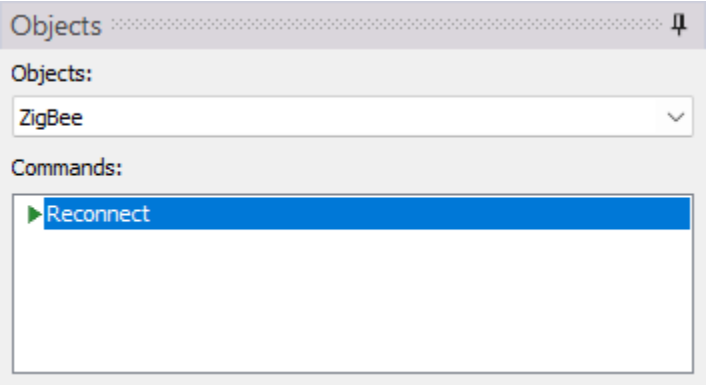
Silence: Threshold Set

This command adjusts the silence suppression threshold to the given value (Fixed mode only.)

Silence: Threshold Up

This command increases the silence suppression threshold (Fixed mode only.)

ZIGBEE



Reconnect

The Reconnect command forces Zigbee controllers to reconnect to another parent in range. Since Zigbee-based controllers do not have roaming capabilities, this command can use a controller in another location in the Zigbee repeater-based network.

Zigbee-based controllers will automatically try to connect to a new parent when they are power-cycled.

Creating a Separate Zigbee Network

Zigbee provides many advantages for RTI controllers. The fast connection, efficient battery usage, and world standard make it a good option for hand-held devices.

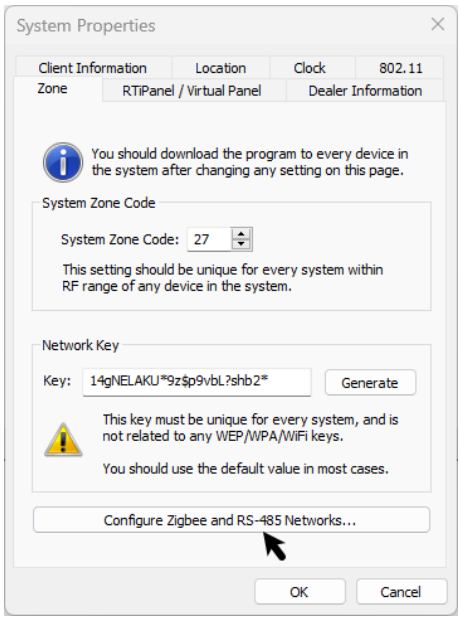
Scenario: A Zigbee network exists in the home theater room in the basement. The client's bedroom is on the other side of the home, where they use a Zigbee controller to control some local equipment in the closet. The ZM-24 in the basement does not provide the bedroom area with Zigbee coverage, and a separate Zigbee network must be established.

1. An expansion processor is placed in the client's closet and will communicate with the controller via Zigbee and control local equipment in the closet. A ZM-24 is wired to the processor and placed in the closet with the equipment.

2. Add the expansion processor to Integration Designer. Name the processor with the bedroom location and configure it on the network.

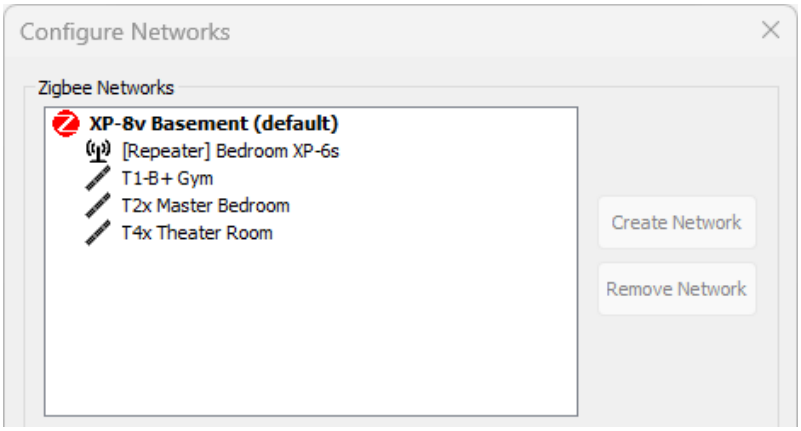
3. Add a room and any controllers for the bedroom to the workspace.

4. Select **File** in the toolbar, then **System Properties**. Select the **Zone** tab, then **Configure Zigbee and RS-485 Networks**.

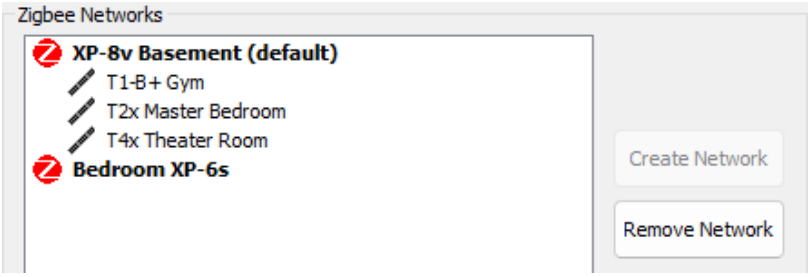


5. The Configure Networks window displays the existing system hierarchy along with any Zigbee networks, repeaters (processors not established as a separate network,) and Zigbee-based controllers.

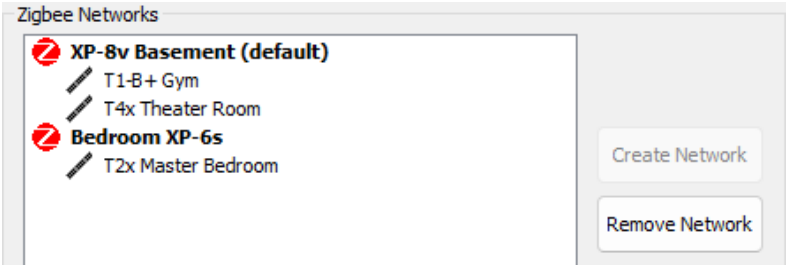
   A single Zigbee network created by the XP-8v processor in the basement is responsible for communication with all the controllers.



6. Create a new Zigbee network for the Bedroom XP-6s by selecting the Bedroom processor and **Create Network**.



7. Two Zigbee networks now exist, but all the controllers are set to communicate with the Basement processor. Drag the controllers that need to communicate with the Bedroom processor by dragging them to the current network.



In this example, the T1B+ and T4x communicate with the XP-8v in the basemen. The T2x is communicating with the Zigbee network established by the Bedroom XP-6s.

8. Power cycle the Bedroom controller close to the bedroom processor's ZM-24 to ensure proper connection.

Technical white papers and training videos for properly setting up a Zigbee network and troubleshooting are available on the RTI dealer website. The following topics are explained in detail:

- Important notes for review
- How to Download Zigbee® files for testing.
- Location for ZM-24 and "best practice" notes
- Installing the Hard-wired ZM-24
- Testing Zigbee coverage
- Installing ZM-24 repeaters
- Troubleshooting Zigbee® Network
- Resetting Zigbee Network
- Zigbee® network using multiple control processors

Creating a Separate RS-485 Network

RS-485 may be used on supported controllers to address wiring distances, commercial requirements, and other limitations. Certain controllers require RS-485 support. In cases where several keypads are specified in a project using RS-485, a CB8 connecting block must be used. The CB-8 installation accessory provides convenient and reliable power connections and communication for up to eight (8) RTI in-wall keypads, touch panels, and other devices. Please note that the number of connections varies depending on the power draw of the controller. In addition to providing power, the CB-8 combines and routes the infrared outputs from the touch panels to a single output for connection to an RTI control processor. Up to four CB8 units can be daisy-chained for connecting more than eight devices. A 4.3 Amp, 16VDC power supply is included.

The CB-8 uses category wiring from the XP processor to the first CB-8. Additional CB-8 units are daisy-chained together using category cabling as well. The first CB-8 connects from the XP processor expansion port to the output on the first CB8. To wire a second CB-8, connect from the Loop Out on the first CB8 to Loop In on the second CB-8. You can then Loop Out to the Loop In on the subsequent CB8 devices. Be sure to power each CB8 with the included 4.3 Amp, 16VDC power supply.

The maximum wiring budget for the CB8 is one thousand feet. Due to power consumption, this limit does not pertain to all controllers. This wiring limit begins from the XP processor to the last CB8 unit in the daisy chain. For example, if you

install (32) RK1+ devices, the total wiring limit from the XP processor to the first CB8, including wiring to subsequent CB8s, may not exceed the 1000' limit. If wiring is exceeded, it will result in communication loss. RTI controllers with higher power draws may limit the wiring distance limit.

All devices requiring ½ Amp or less can be supported on a CB8 utilizing all eight inputs.

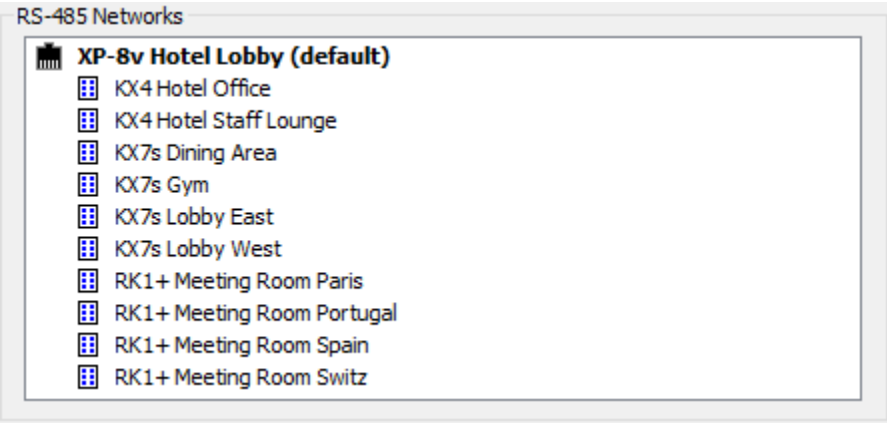
The other devices will need to consider the remaining amperage to determine the amount of which can be connected and provide sufficient power for normal operation. You may also use a 16v PSU (Part# 40-210285-22) when using the maximum amount of amperage to avoid voltage issues, if necessary.

Here are the limits that should be observed per RTI controller:

- KX7s, K4: 1A - 4 maximum - Up to 150 feet away.
- KX4: .5 A 8 maximum - Up to 500 feet away. (PS16-1 16VDC Power Supply recommended)
- KX1, KX2,RK-1+, RKM-1+: 350mAh/ .3 A - 8 maximum - Up to 1000 feet away.
- KX10s/CX10 – 2A max or PoE+ (Using Class 3 PoE+ Ethernet switch)

A maximum of **32** RS-485 devices can be connected to the XP processor expansion port using multiple CB8 connection blocks. However, it's recommended to monitor the speed of the link because adding too many devices with a low baud rate may affect the speed of data transfer. The 32-device rule will only be valid for RTI controllers occupying all eight ports on the CB8. RTI has tested multiple scenarios, and communication loss could be unavoidable when approaching the maximum device limit. In this case, you may consider **separating the RS-485 networks** in Integration Designer to help balance the load.

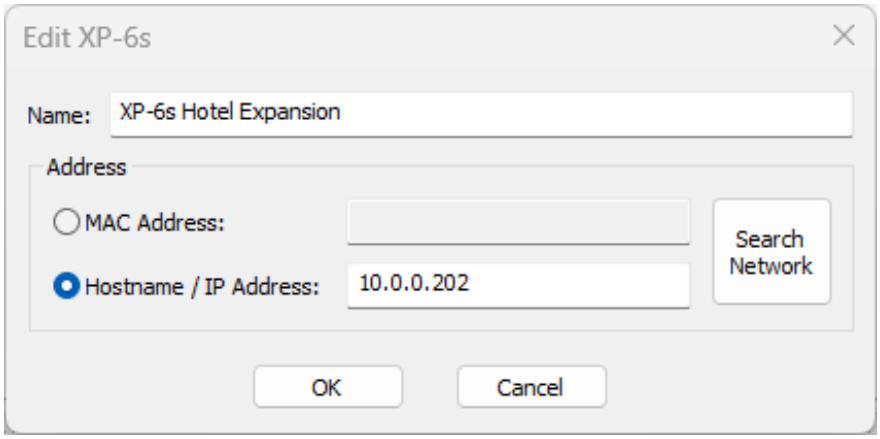
Scenario: A commercial hotel installation calls for a mix of RS-485-controlled devices that exceed the maximum number of CB8 devices. In this situation, separated RS-485 networks are required to support the number of keypads and distribute the load evenly.



In this example, the number of keypads exceeded the RS-485 capability of the initial RS-

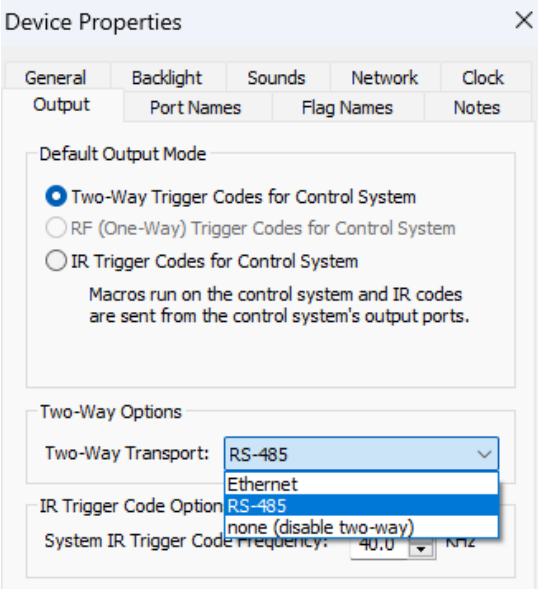
485 network.

1. Add as many expansion processors as additional RS-485 networks are needed. Balancing the load will ensure no latency or missing commands, if possible. For example, try not to hit the limits for any RS-485 network for best practice.



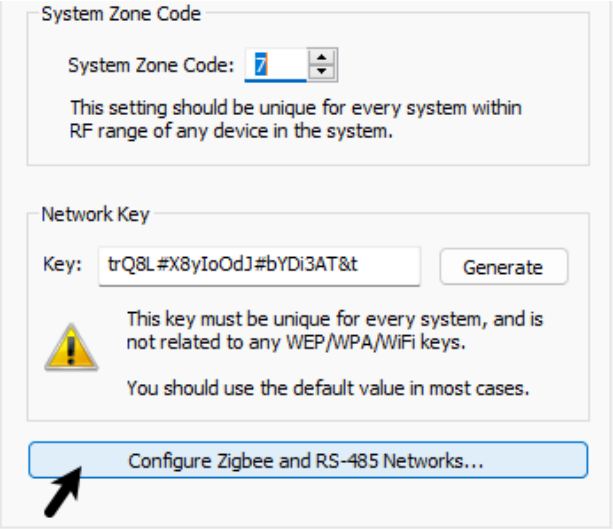
   In this example, another processor was added to expand the RS-485 network. Since there are ten controllers on the system, five will be allocated to each RS-485 to balance the load properly. (Please note the number of keypads depends on the voltage.)

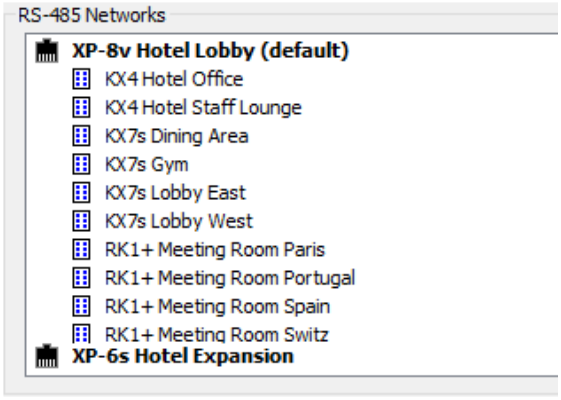
2. For controllers that support additional two-way transport protocol, ensure RS-485 is selected in the Two-Way transport dropdown menu in the controller properties.



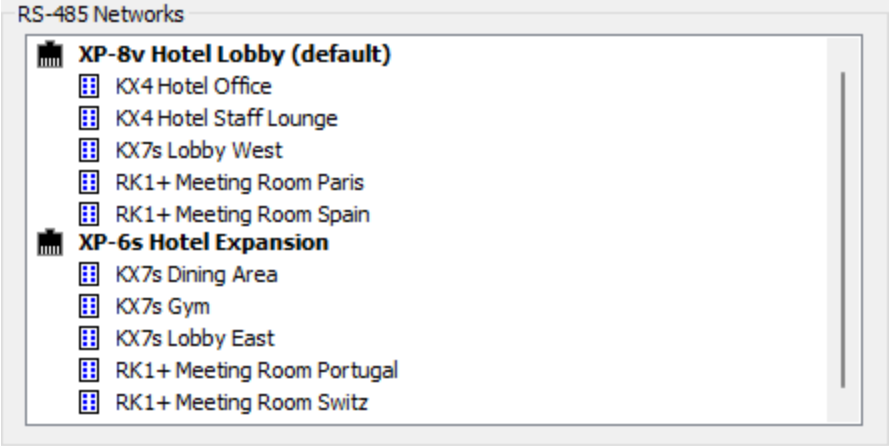
3. Select the **File** option in the main toolbar, then select **System Properties**.

4. Select **Configure Zigbee and RS-485 Networks** located in the **Zone** tab.





5. **Rearrange** the RS-485 devices by placing them under the underlying processor. Doing so will separate the RS-485 network so that proper communication occurs.



The RS-485 network is now evenly distributed without hitting limits on either network.

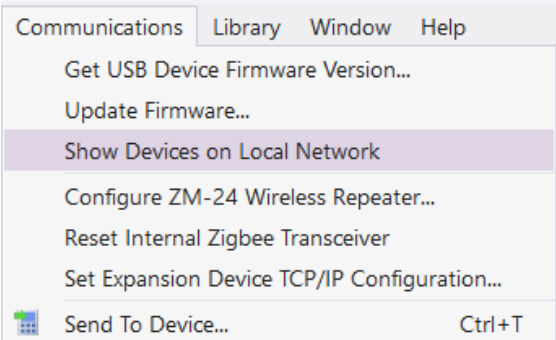
Updating Firmware

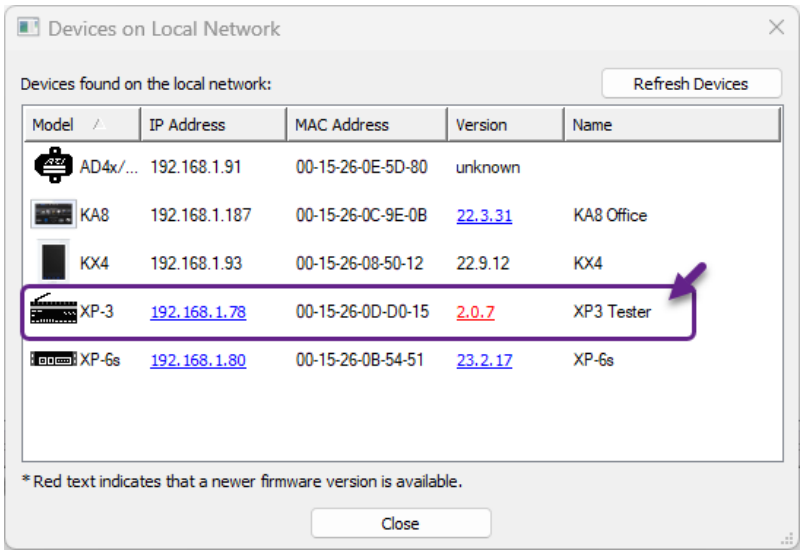
Updating firmware on RTI devices is a critical step and should be performed before testing or evaluating project performance. Before uploading a project or contacting technical support, ensure that firmware is updated to the current version.

VIEWING FIRMWARE VERSIONS

If you are uncertain about the firmware version on an RTI device, there are generally three different methods:

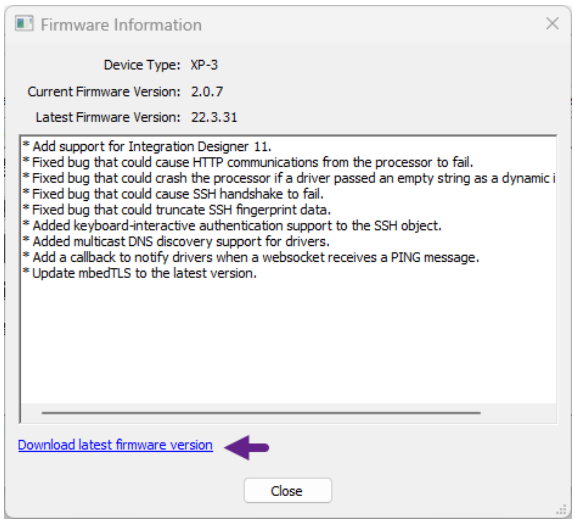
1. If the device is on the network and supports firmware reporting, select the **communication** tab and then select **Show Devices on Network**. RTI network devices, including controllers and processors, configured for network use will display. The firmware version will appear next to supported devices. If the firmware is out-of-date, it will have red text.





Red text in the version section indicates the firmware is out-of-date and a new version is available.

2. Select the firmware version link in red text to download the latest version. A firmware information window will display and provide information on the latest version available. Select the **link** to **download** the **latest firmware version**.

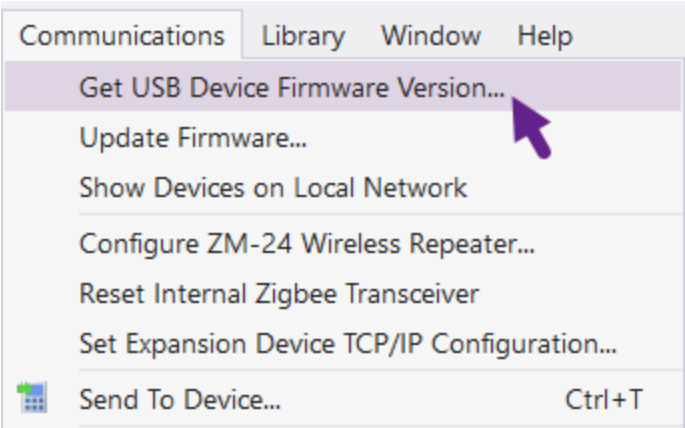


3. Unzip the contents once the file is downloaded and save it in a dedicated folder.

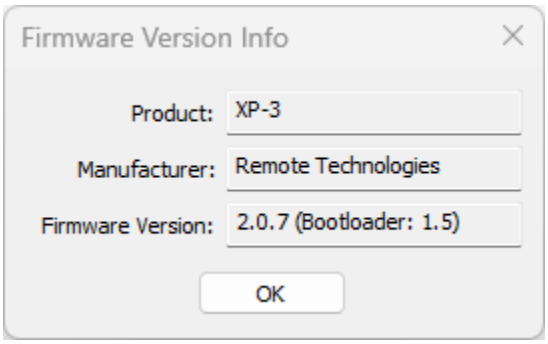
A **USB connection** is another option for devices that do not support network communication or are not on the network.

1. Plug the USB cable into the device and wait for it to connect properly.

2. Select the **Communications** option from the top toolbar in Integration Designer, then select **Get USB Device Firmware Version.**

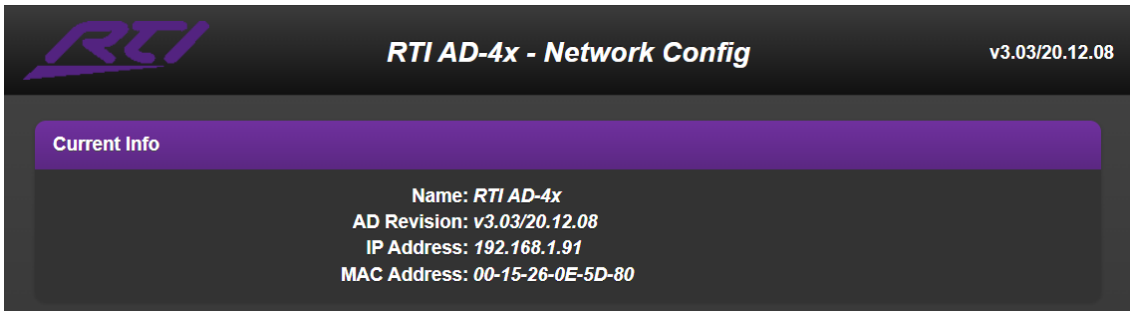


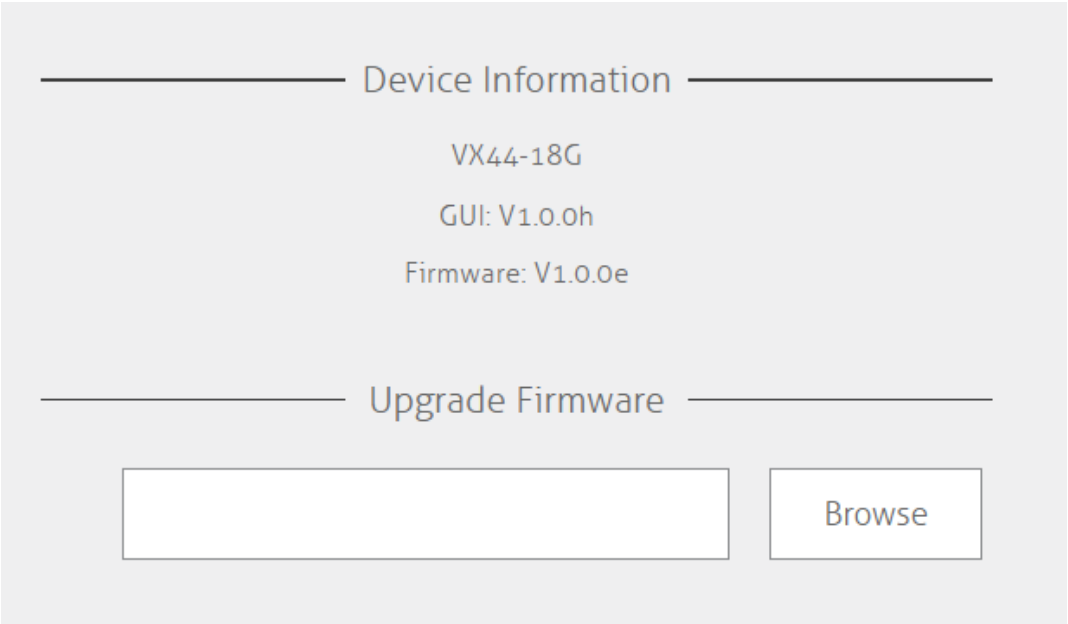
3. A pop-up will display the firmware information for the device. While this will not indicate if a firmware update is required, the dealer website will have the latest firmware for applicable RTI devices.



Certain devices require viewing the device interface on a web-based interface. Generally, this can be used to view and update the firmware.

1. Go to the device page by entering the IP address in a supported web browser.





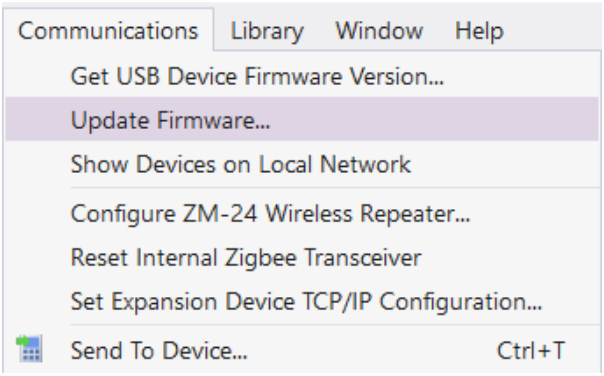
2. Important information may be viewed. If a firmware update option is unavailable through the web interface, go to the RTI firmware page on the dealer website, find the device, and follow the firmware update instructions.

UPDATING FIRMWARE VIA ETHERNET

Compatible RTI devices configured for network communication may be updated via the network through a wired or wireless connection.

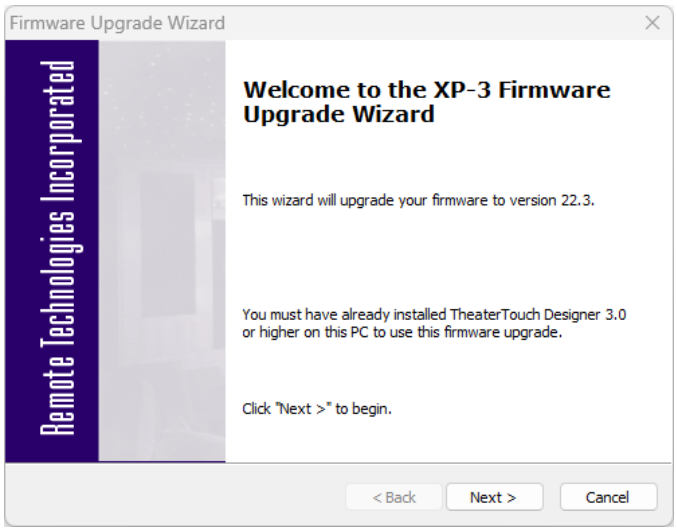
1. Download the latest firmware update on the RTI dealer site manually, or select the Communication option in the toolbar, then select Show Devices on the Network." Click the link for the out-of-date software and save it to a local PC.

2. If the device supports firmware updates via a local file, select the **Communications** option in the toolbar in Integration Designer and select the **Update Firmware** option.



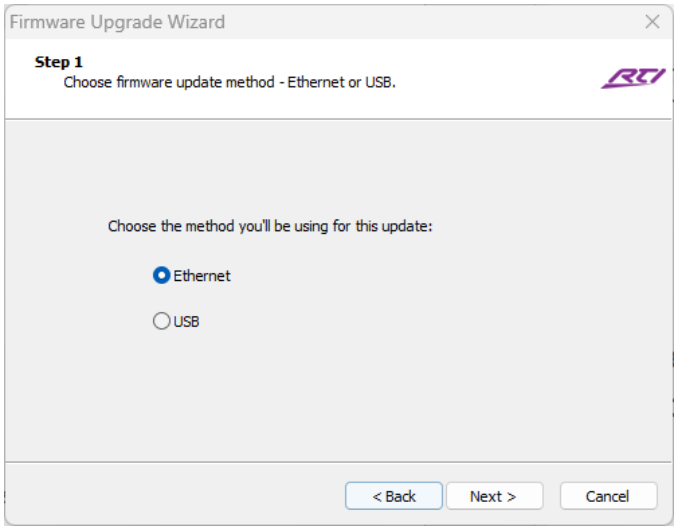
3. Navigate to the directory where the firmware is stored. Select the firmware file and follow the instructions on the screen until the firmware is updated.

4. If the firmware update file is a .exe file, navigate to the directory where it was downloaded and run the executable. A window will display and guide you through the firmware process.

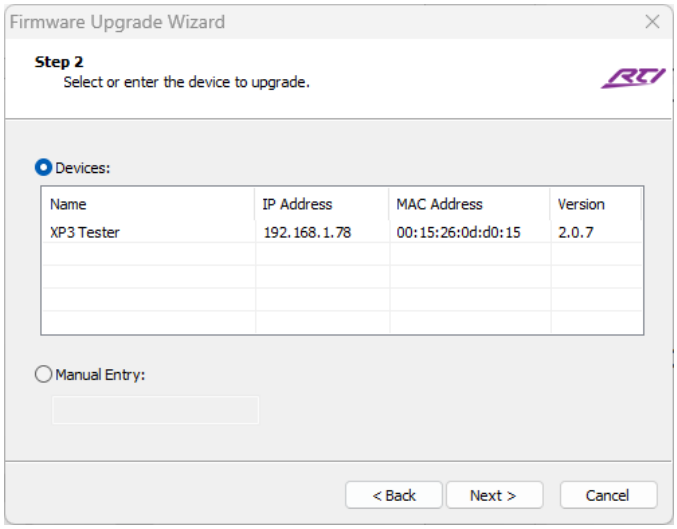


5. Select Next to begin.

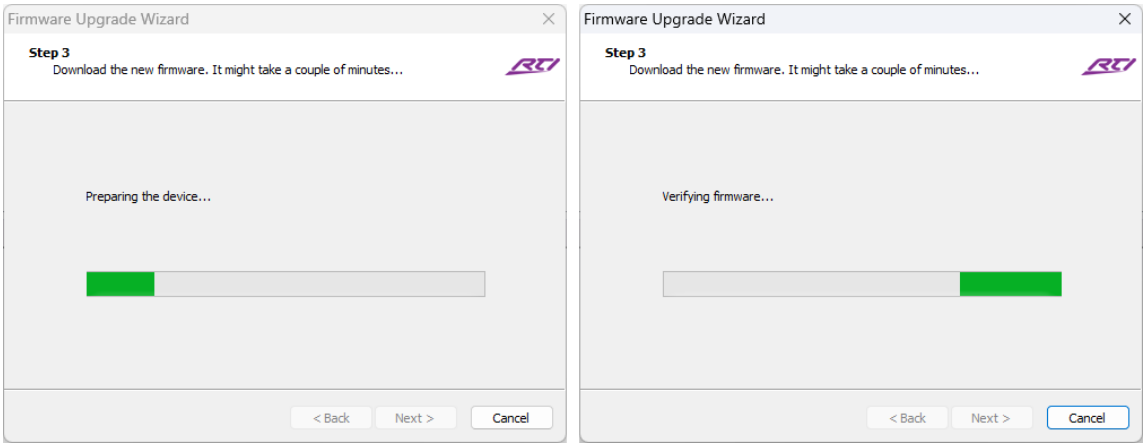
6. If using a USB transmission, plug in a USB cable to the device and select the USB option. If using Ethernet, ensure the device is discoverable on the network.



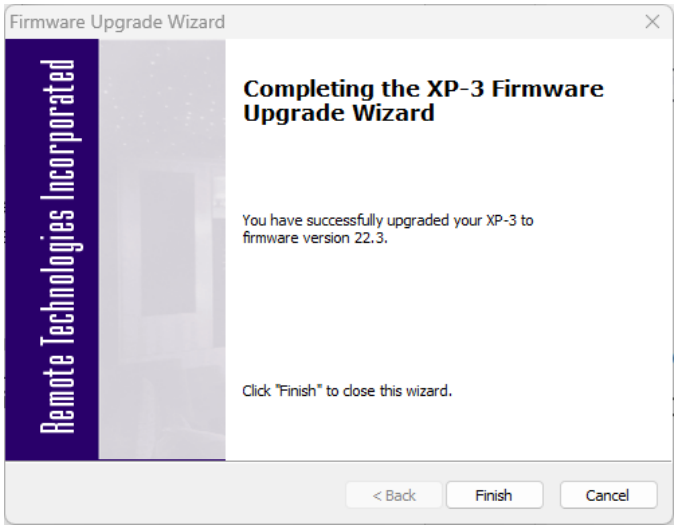
7. A list of devices compatible with the firmware version will be displayed. Select the product in the list of devices. One device may be updated at a time. An IP address may be entered manually by selecting the manual entry option and entering the device's IP address. Select Next to begin the update.



8. The device will go through the firmware process by downloading, verifying, and performing some checks. Do not interrupt the firmware update process.



9. Wait patiently while the firmware update is completed. Observe the device, which could display a progress screen or blinking lights. If you are unsure about the light status on the processors, consult the operation manual for details.



10. Once the firmware is completed, exit the update window by selecting the Finish button. Verify the firmware version via USB or by displaying the devices on the network.

Pro Tip: Firmware updates often include a **README** file for important information. If performing firmware updates for the first time, proceed carefully, reading all the procedure notes and following the on-screen instructions carefully. Firmware is crucial to ensuring the device is properly communicating without issue. In addition, be sure to use the latest version of Integration Designer software.

Updating the Project File

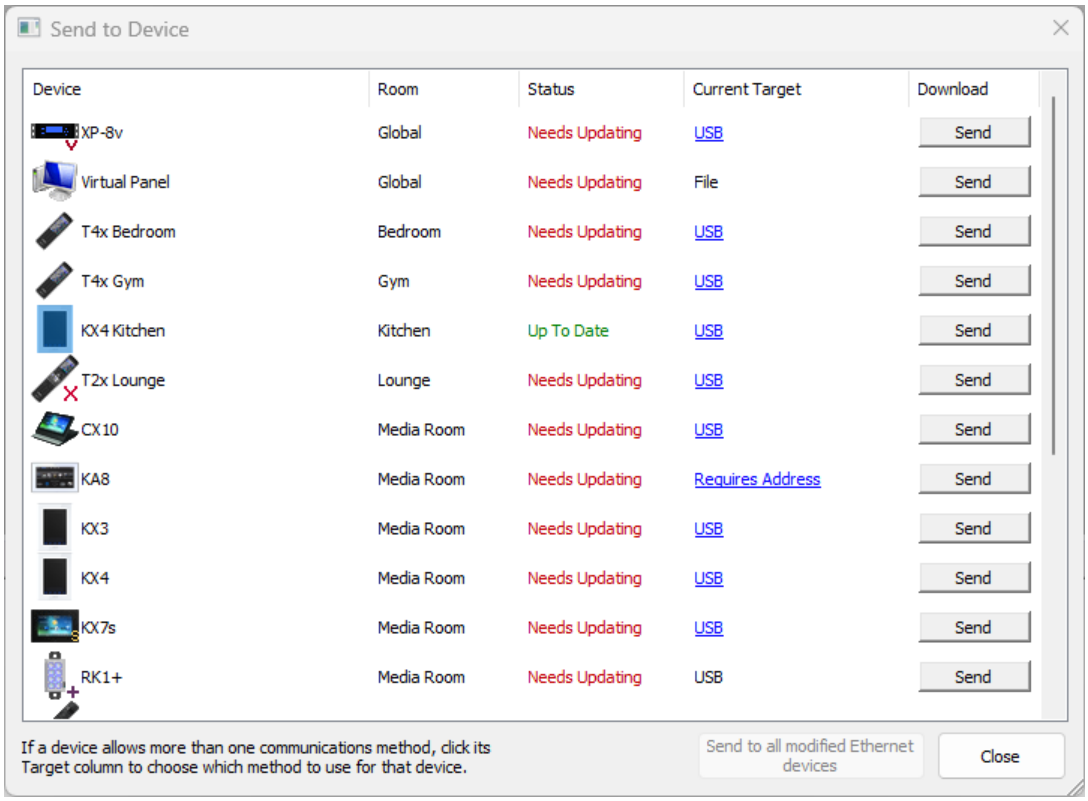
Congratulations, you completed the programming and are now ready to synchronize the software configuration with the hardware. Here is a checklist of items that are critical before updating and testing the project file:

1. Use the latest version of Integration Designer software.
2. Ensure that all devices have been updated with the current version of firmware and are properly configured.
3. The latest driver versions are being used, and the driver notes have been reviewed and verified.
4. All devices are properly charged and situated in the proper location.
5. The high-grade network has been tested and is operating properly. Network provisions have been made to facilitate the RTI ecosystem.
6. All Zigbee and RS-485 networks have been properly configured and evaluated.
7. Status lights on antennas, transceivers, and processors are operating normally.
8. Third-party devices have been configured per driver notes to communicate effectively with the RTI control system. The firmware is up to date.
9. A thorough site survey has been done to anticipate any devices that could impede signal communication.
10. You have scheduled time with the end users to train them on the options and features of the control system.

Devices with ethernet capabilities and the "**Allow the device to be programmed over the network**" option enabled may be programmed via the ethernet interface.

1. Select the **Communications** option in the Integration Designer toolbar, then select **Send to Device**.

2. A list of RTI devices will be displayed. The list is sorted with the master processor at the top, followed by devices in the global area. The rest of the rooms and devices are included next. The last order of priority are devices such as RTiPanel devices that are stored on the processor.



Device: The name of the RTI device in the workspace.

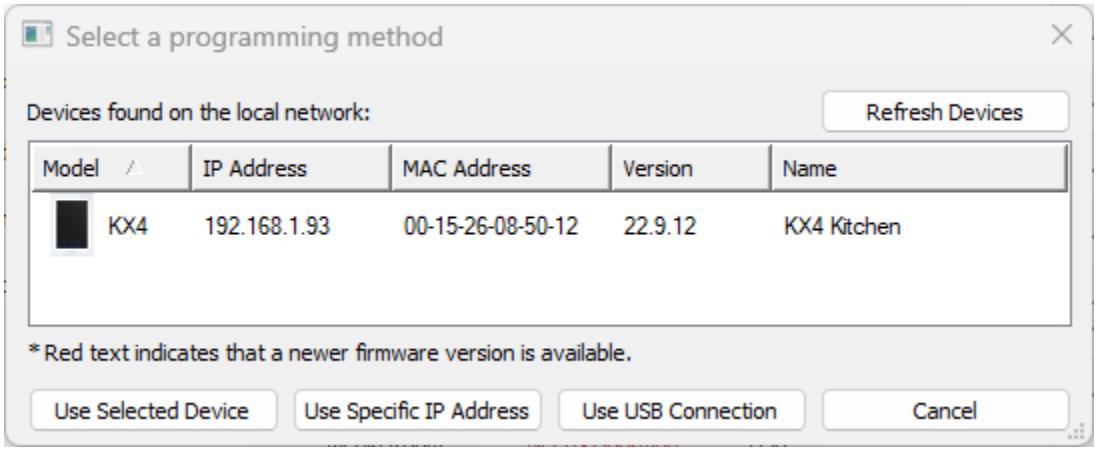
Room: The location of the RTI device in the workspace.

Status: The current update status of the RTI device. Available options are:

- **Needs Updating**
- **Up to Date**
- **Requires Address** – In this case, the device must be discovered on the network, and an initial download must occur.

Current Target: The update method used to send the project file to the device. Available options are USB or via network method.

3. Click on the device's Target column to access the network programming method.



4. Click on the device in the device list.

5. Choose one of the following programming methods.

USE SELECTED DEVICE

- Click "Use Selected Device" to identify the device by its MAC address.
- Click on the Send button.

Note: This programming method only works if both the device and the programming PC are on the same local network, but it allows the PC to always find the device, even if the DHCP Server assigns it a different IP address. The recommended network programming method is preferred.

USE A SPECIFIC IP ADDRESS METHOD

- Click "Use Specific IP Address" and enter the device's IP address.
- Click on the Send button.

Note: This option can be used to update the programming over the internet if the device has a public IP address or if a VPN connection is configured to access the network that the device is on.

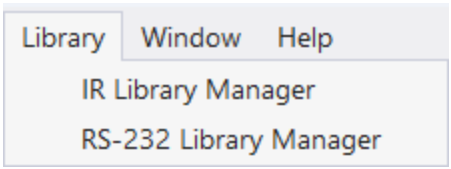
6. You may use the USB connection if preferred or if a device does not support the network method.

7. Send to “All Ethernet Devices” will send the configuration to each device consecutively and is ideal for projects with many devices.

Note: Devices such as RTiPanel devices stored on the processor do not require manual downloading and will be automatically updated with the processor.

IR Library Manager

The Infrared Library Manager is a software program for creating or modifying infrared Command Libraries. An “Infrared Command Library” is a collection of Infrared codes grouped into " Remotes " units. Infrared Command Libraries store Infrared codes from remote controls that can be placed on buttons or in macros. Typically, you will have a single Infrared Command Library containing all your Infrared codes.

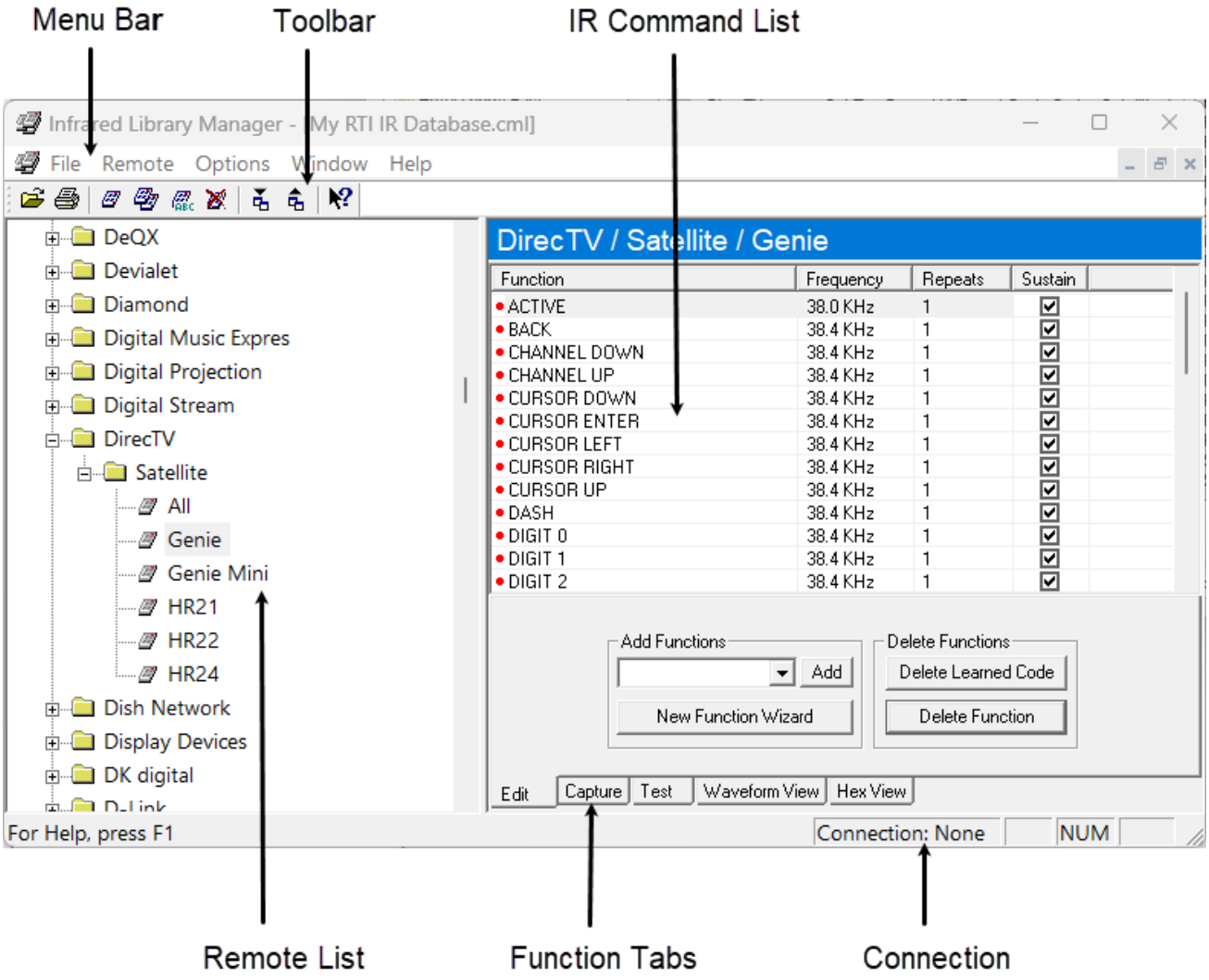


The IR Library Manager software program is made available from the library menu of Integration Designer.

To use the IR Library Manager:

1. Choose IR Library Manager from the Library menu from the Integration Designer toolbar.

2. From here, you can decide to (1) create a new library of commands or (2) add to or edit an existing library by choosing from the File menu either New or Open.



Menu Bar: Provides drop-down menus of all programming features.

Toolbar: Icons for easy access to frequently used programming features.

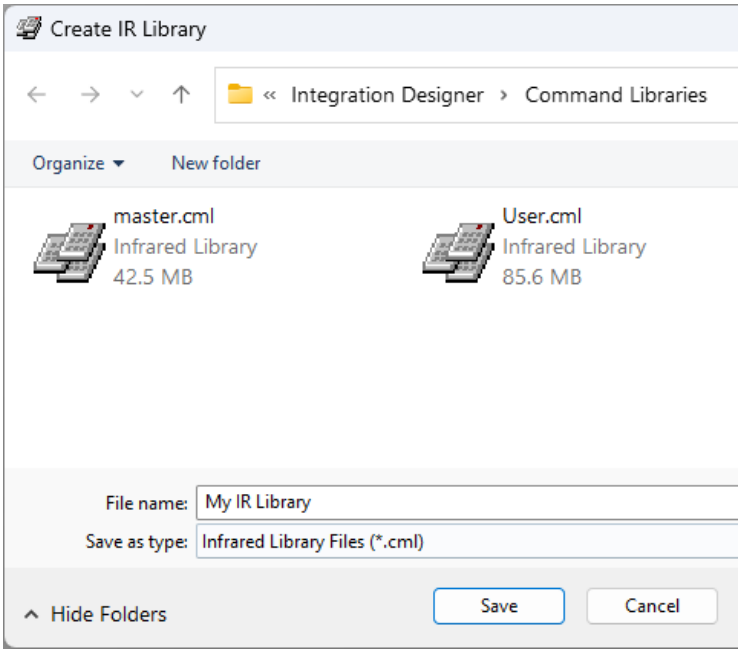
Remote List: Displays the brand, type, and model of the device of the remotes included in the command library.

IR Command List: Displays an editable list of the IR codes and their function for the selected remote.

Function Tabs – Programming option tabs for adding and editing individual functions and IR codes.

Connection – Displays the current connection status of the IR capture device.

CREATING A NEW IR COMMAND LIBRARY



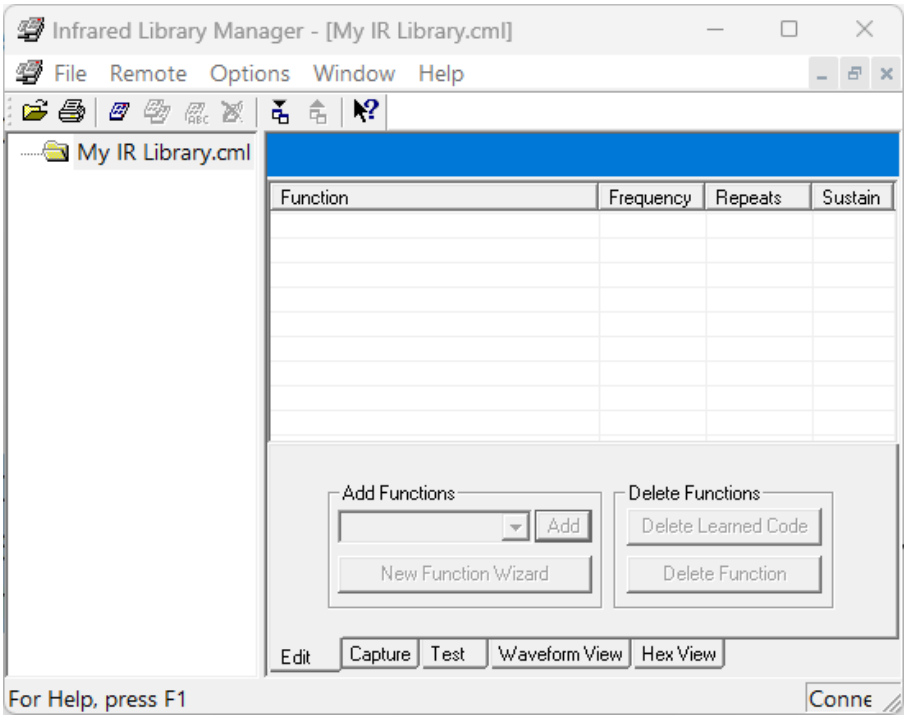
1. Choose New from the File menu, and the Create IR Library dialog box will appear.

2. Enter the name of the new library in the File name box.

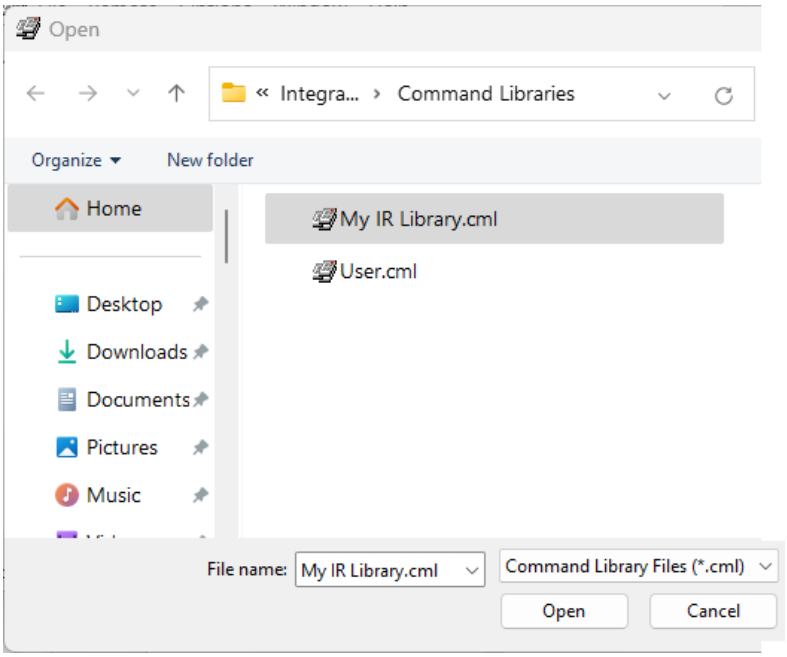
3. Be sure to save your new file in the Command Libraries folder of the Integration Designer software folder.

4. Choose the Save button to create the library.

Once you choose the Save button, the IR Library Manager window will show no remotes in that library, and the fields will be empty.



OPENING AN EXISTING IR COMMAND LIBRARY

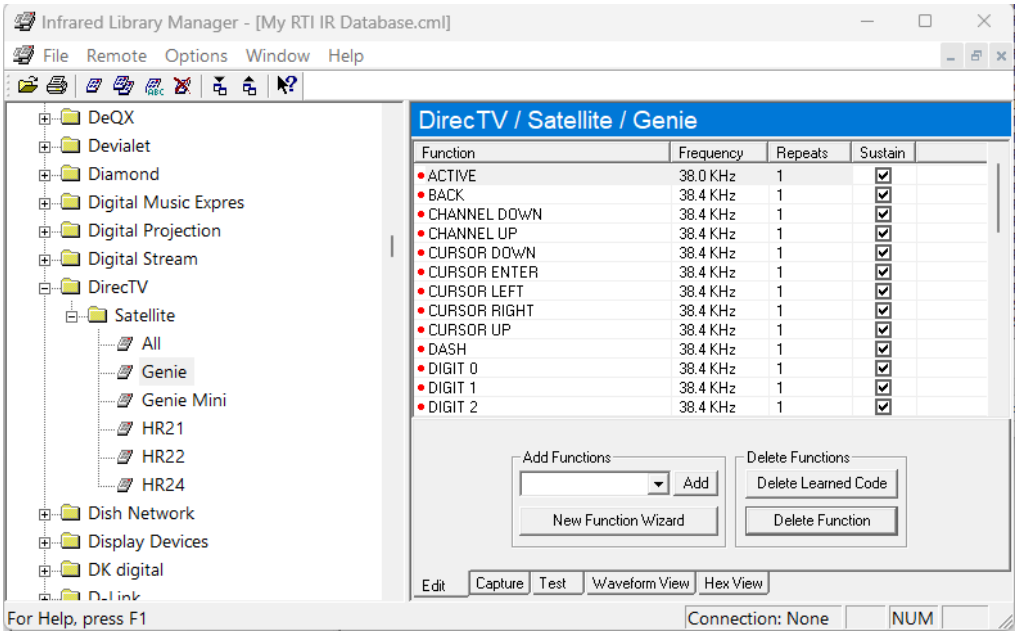


1. Choose Open from the File menu or select the Open icon from the Toolbar.

2. When the Open IR Library dialog box appears, select the library command file you wish to open.

3. Select the Open button.

Once you select the Open button, the IR Library Manager window will open, showing the remotes in that library.

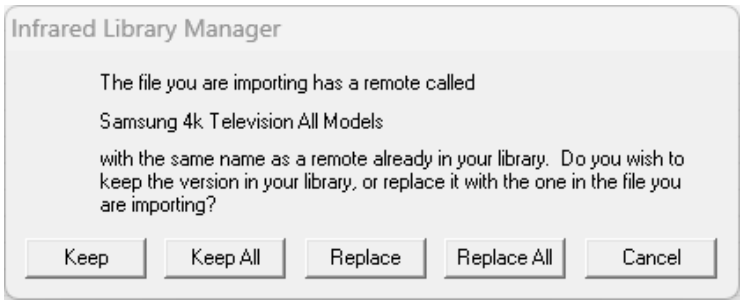


IMPORTING AN IR COMMAND LIBRARY

This programming option allows importing all remotes from one library file into the current library.

1. Choose Import from the Remote menu or the Import Remote icon.
2. The Import IR Library dialog box will appear.
3. Choose the library you wish to import from the file directories.
4. Select Open to import the library.

   Note: If any remotes in the imported file have the same names as remotes in the current library, you will be warned and asked which version to keep.



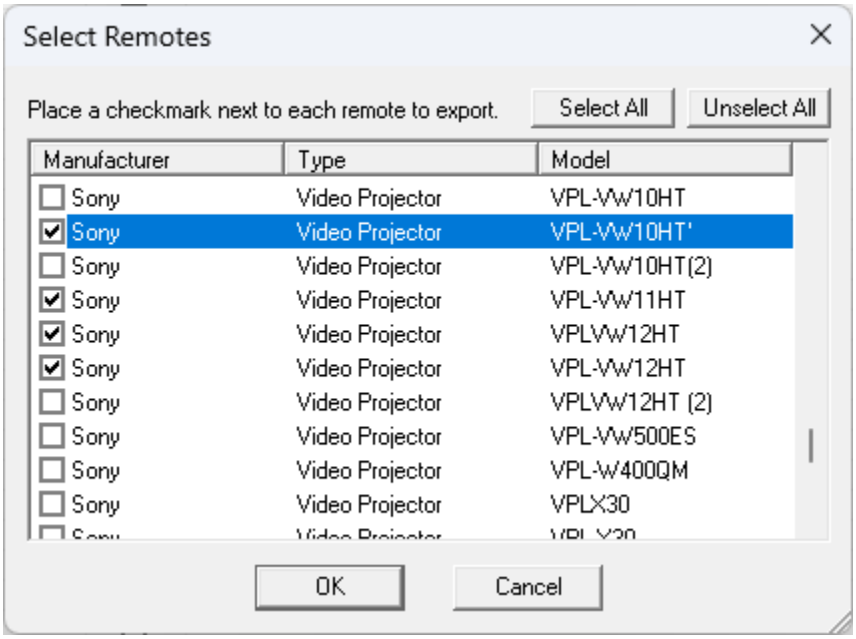
EXPORTING AN IR COMMAND LIBRARY

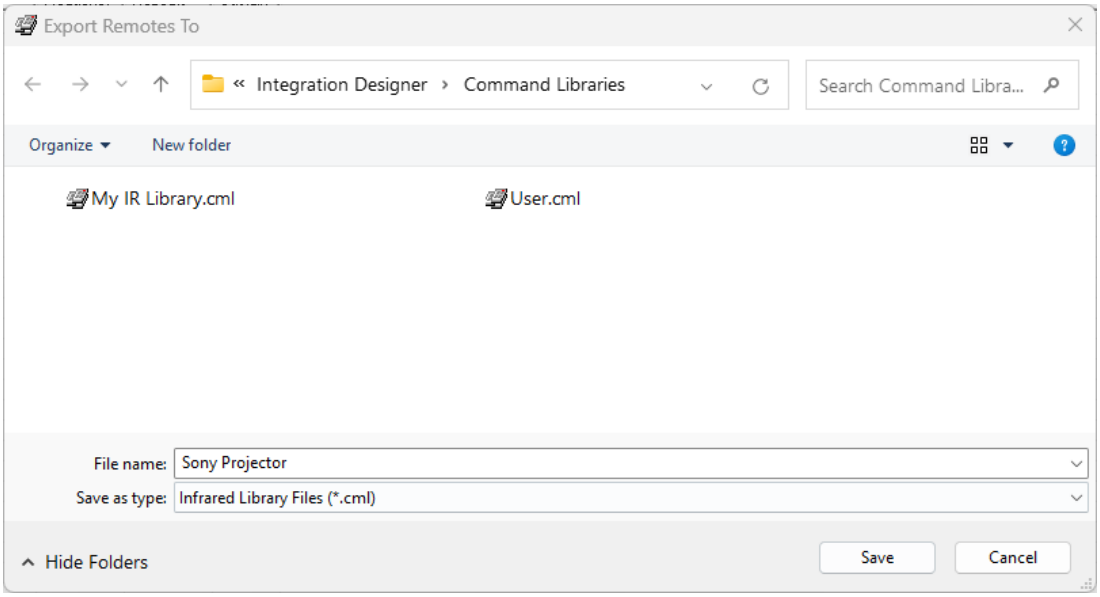
This programming option allows you to export selected remotes from one library file into a new library file.

1. Choose Export from the Remote menu or select the Export Remote icon.
2. The Select Remote to Export dialog box will appear.
3. Choose the remotes you wish to export to the remote list.

4. Choose OK, and the **Export Remotes** dialog box appears.

5. Enter a name and directory for the new library file and choose OK to save the file.





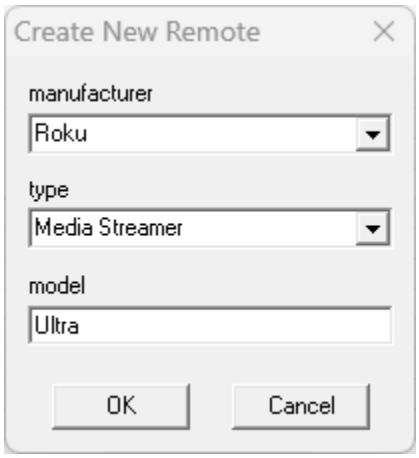
ADDING A NEW REMOTE

1. Choose New from the Remote menu or the New Remote icon.

2. The Create New Remote dialog box will appear.

3. Enter the manufacturer's name and the equipment type for the added remote control (such as a TV, Media Streamer, Cable Box, and so on) in the appropriate fields, or choose the name and type from the drop-down lists.

4. Enter the model number of the component separated by the remote.



5. Choose OK to add the remote to the IR Command Library.

DUPLICATING A REMOTE

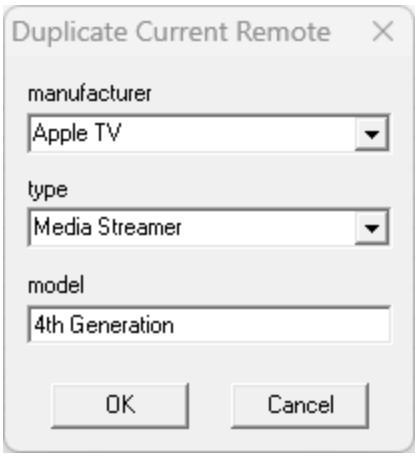
Use this programming option to create a new remote that is an exact copy of the selected remote in the current IR Command Library.

1. Choose Duplicate from the remote menu or select the Duplicate Remote icon.

2. The Duplicate Current Remote dialog box will appear.

3. Edit the new manufacturer's name and equipment type for the duplicated remote control (such as a Television, Media Streamer, Cable Box, and so on) in the appropriate fields or choose the name and type from the drop-down lists.

4. Edit the model number of the component operated by the remote.



5. Choose OK to duplicate the remote into the IR Command Library.

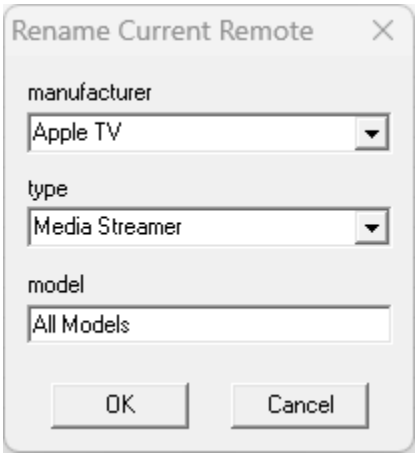
RENAMING A REMOTE

1. Choose Rename from the Remote menu or the Rename Remote icon.

2. The Rename Current Remote dialog box will appear.

3. Edit the new manufacturer's name and equipment type for the renamed remote control (such as Television, Media Streamer, Cable Box, and so on) in the appropriate fields or choose the name and type from the drop-down lists.

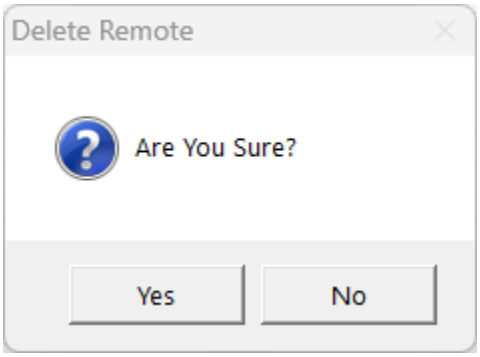
4. Edit the model number of the component operated by the remote.



5. Choose OK to rename the remote in the IR Command Library.

DELETING A REMOTE

1. Choose Delete from the Remote menu or select the Delete Remote Icon.
2. The Delete Remote dialog box will appear.



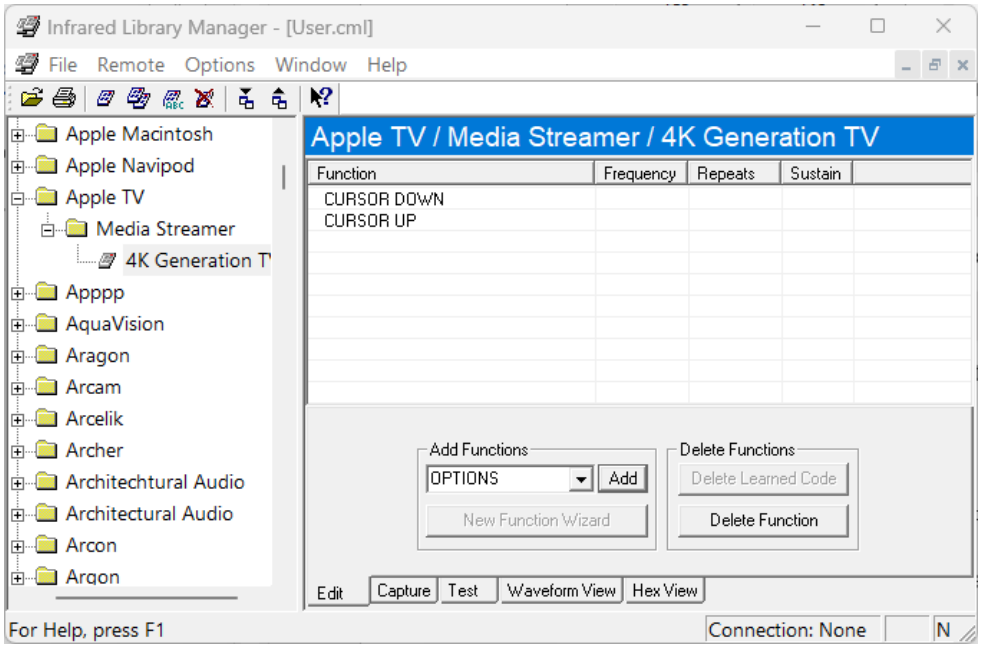
3. Choose OK to delete the remote from the IR command library.

ADD FUNCTIONS TO A REMOTE

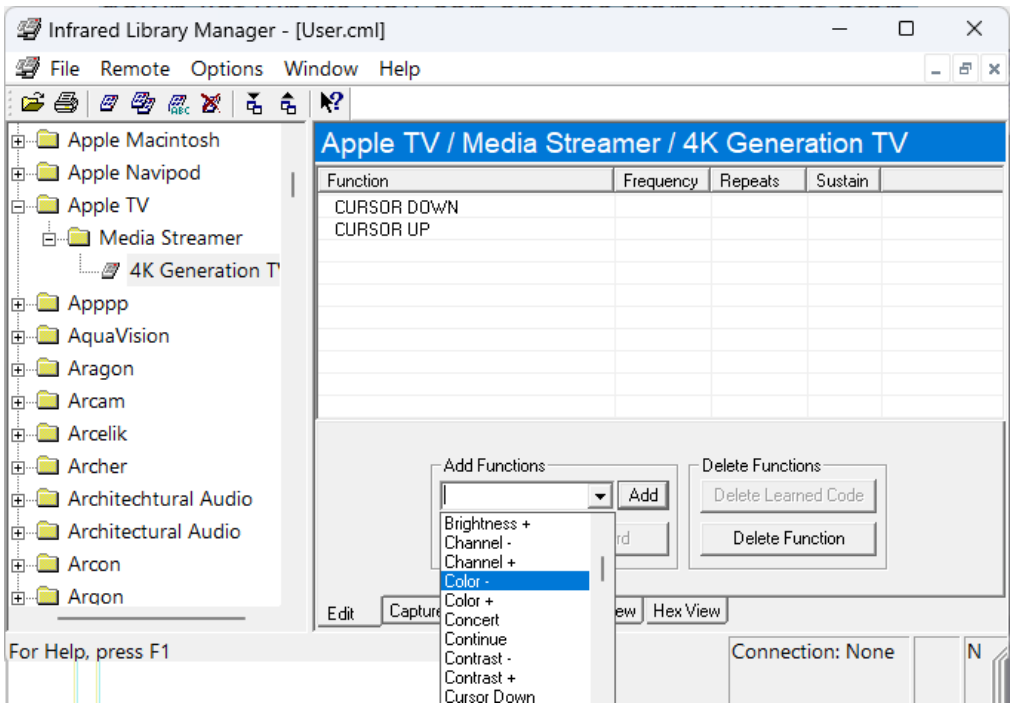
The Edit tab is where individual functions for a remote file are edited and created.

To add functions to the currently selected remote:

1. Type the function name in the "Add Functions" box and click the Add button.
2. The added function appears in the Function list.
3. Repeat step 1 for each function to be added.



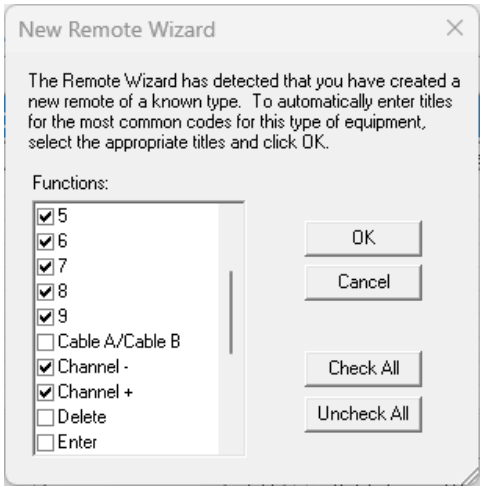
The “Add Functions” box also includes a pulldown list where you can choose from a list of standard functions. If the remote type is recognized, the list will be filtered to functions only appropriate for that remote type.



USING THE NEW FUNCTION WIZARD

If you are entering a commonly used component into the IR Library, the New Function Wizard option enables quick access to commonly used functions for that remote type.

1. Choose the New Function Wizard button from the New Remote dialog box.
2. The New Function Wizard dialog box appears.
3. Check the functions you wish to add to the Function list.
4. Use Check All to select the entire list and Uncheck All to deselect the entire list.



5. Choose OK to confirm the selected functions.

DELETING FUNCTIONS FROM A REMOTE

You can delete functions by selecting them in the function list and then clicking the "Delete Function" button. If you wish to remove the IR code but leave the function title (so you can re-learn the codes), select the function and click the "Delete Learned Code" button.

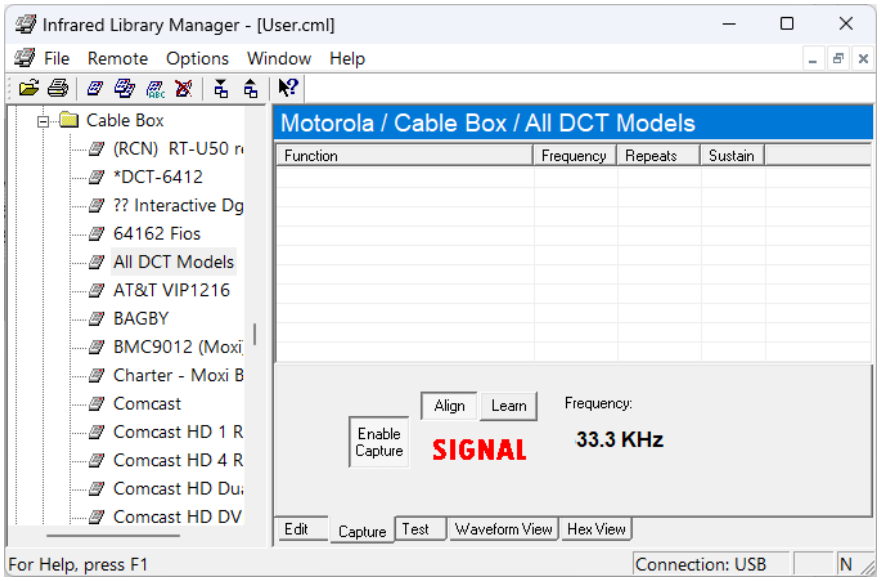
EDITING FUNCTIONS IN A REMOTE

Select the property you wish to edit in the function list and make the changes directly in the list. For example, if you want to change the name of a code, select the function name in the function list and type in the new name. Similarly, you can edit the frequency and number of repeats and sustain options. Note that you must have the Edit tab active to edit the items in the list.

CAPTURING IR CODES WITH AN IR-PRO LEARNING DEVICE

The first step in capturing IR codes with an IR-PRO learning device is correctly aligning the donor remote. This step is very important, as it allows the Infrared Library Manager to determine the carrier frequency of the donor remote.

1. Connect the IR-PRO to a free USB port on your PC.
2. Select the Capture tab.
3. Choose the Enable Capture button.
4. Choose the Align button.
5. When the device is in Align mode, you will see the following on the Capture tab:



6. Hold down any key on the donor remote and move it around the face of the capture unit until the signal strength meter turns red and the frequency reading is stable.

Once the remote is properly aligned:

1. Select the "Learn" button. The Infrared Library Manager will automatically highlight the first function in the current library without an associated IR code.

2. If you want to learn another function, select it in the function list. Your Capture tab should now look like this:

3. Press and hold down the requested function (The Play command in this example) until the Infrared Library Manager learns the code and moves on to the next code.

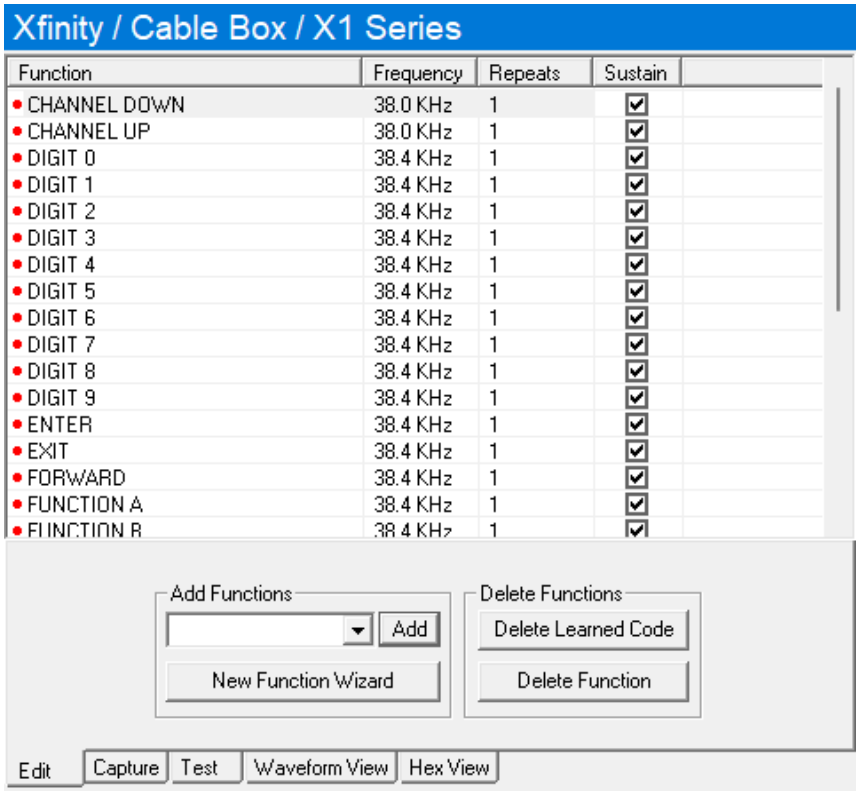
Important: Avoid moving the donor remote while capturing codes once you have the donor remote aligned. If the remote does move out of position, select the Align button and go through the alignment procedure again.

EDITING IR CODES

Most infrared codes will transmit continuously if the remote-control key is held down. This action means that the entire code or some part of the code is repeated consecutively.

The Infrared Library Manager software allows you to adjust the number of commands repeat times and turn continuous transmission on or off.

Select the property you wish to edit in the function list and make the changes directly in the list.



Repeats - Enter the number of times the remote sends the repeating part of the IR code. Some infrared receivers need a code to be repeated a certain number of times to work properly, and some may not work properly if the code is repeated at all. Change this setting only if you are having trouble with a code.

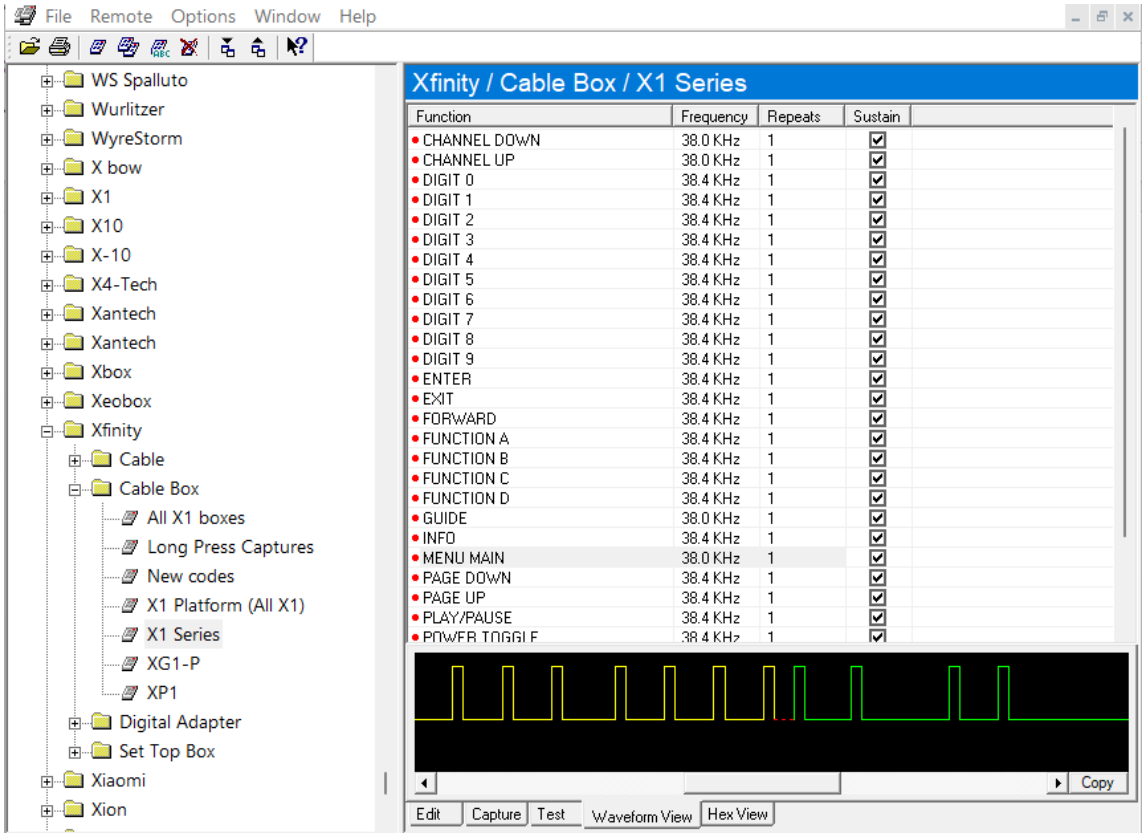
Frequency - This box shows the carrier frequency of the IR code. This value is set automatically. You should not need to change this value unless directed to do so by Technical Support.

Sustain - Check this box to control if the IR code repeats if the button is held down. Most infrared codes will transmit continuously if the button is held down, but some receivers may not work properly if the command is repeated. If the Sustain button is checked, the code will be transmitted for as long as the button is held down or the **Minimum Repeat** times, whichever is greater. If the Sustain button is not checked, the code will be transmitted only the **Minimum Repeat** number of times regardless of how long the button is held down.

VIEWING IR CODES

WAVEFORM VIEW

The Waveform View tab displays a picture of the currently selected Infrared code. This information may be useful to Technical Support if you experience problems learning or using Infrared code.



Yellow Waveform - This portion of an infrared code is sent once.

Green Waveform - This portion of an infrared code is repeated.

Red Waveform - This represents a portion of the infrared code that is very long and is shortened for display purposes.

HEX VIEW

The Hex View tab allows you to import, export, and edit codes in the fully learned Hex format (those that start with 0000.) All other codes must be converted to this format first.

To use the Hex View:

1. Select a function in the function list. If an Infrared code is already associated with that function, it will be converted to Hex format and displayed in the "HEX Data" field. If you select a function that does not have an Infrared code, the "HEX Data" box will be empty.

2. Type in the Hex data for the code or copy it from an external program (such as a web browser), and then right-click in the "HEX Data" box and choose "Paste" from the menu.

3. Select Apply to update the Infrared code in the library.

You can use the "Auto-Paste" feature to save time if you convert many codes.

To use the Auto-Paste function:

1. Click on the Edit tab and add the names of functions you have hex codes for.

2. Select the Hex View tab and check the "Auto-Paste" box.

3. Switch over to the application containing the hex data for the currently selected code. In that application, select and copy the hex data to the clipboard.

4. The Infrared Library Manager will automatically see and paste the copied data to the currently selected function. It will automatically move to the next function, ready for you to copy the next function's data.

5. Repeat step 3 until all codes are added.

Note: Not all codes in the library can be represented in Hex format. If you try to view a code that cannot be represented in Hex, you will see **"<<this code is not representable in HEX format>>**" in the HEX Data field.

UPDATING THE RTI MASTER IR COMMAND LIBRARY

Please follow the steps listed below to update the RTI Master IR Command Library:

1. Before downloading the updated IR Command Library, rename the library file you currently use to store codes you learn. This command library is in the Documents/Integration Designer/Command Libraries folder. We recommend renaming this current file **"User.cml."**

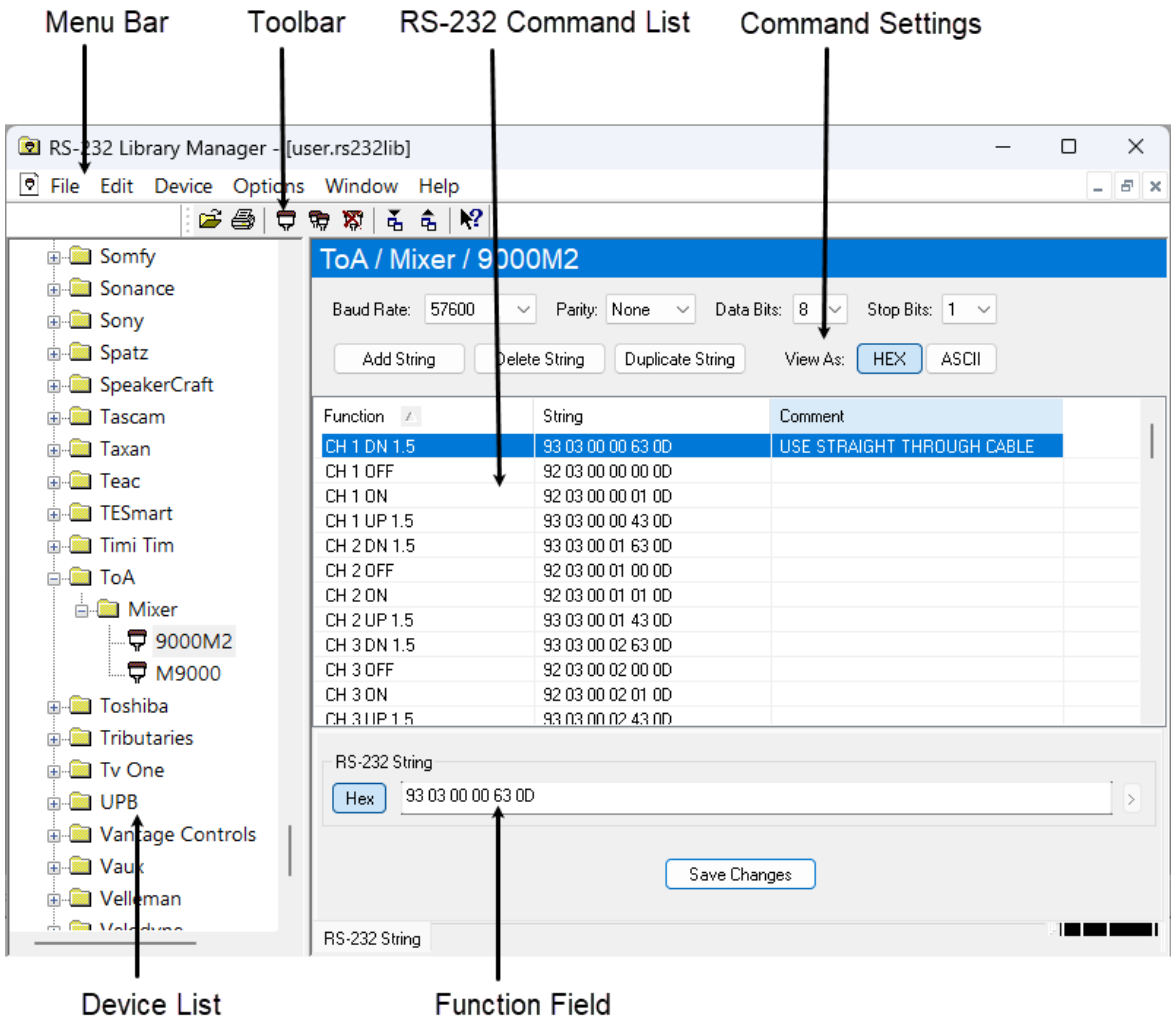
2. Download the new master library from the RTI dealer website.

3. The Master Command Library File is available for download in the Infrared command codes section. The entire new library will be downloaded as a file labeled "**master.cml**." Please save the file to a location where you can easily find it again. We recommend saving it to the command libraries section under the Documents/Integration Designer directory.

Please note: The new master library is a read-only file and may not be opened within the IR Library Manager. It can only be opened within the Integration Designer program when programming an RTI Control System.

RS-232 Library Manager

The RS-232 Library Manager is a software program for creating or modifying RS-232 Command Libraries. An “RS-232 Library” is a collection of one-way serial codes grouped by the device. RS-232 Libraries store one-way serial codes from a protocol that can be placed on buttons or in macros. Typically, you will maintain a single RS-232 library containing all your one-way serial device codes.



Menu Bar: A list of options available in the RS-232 library.

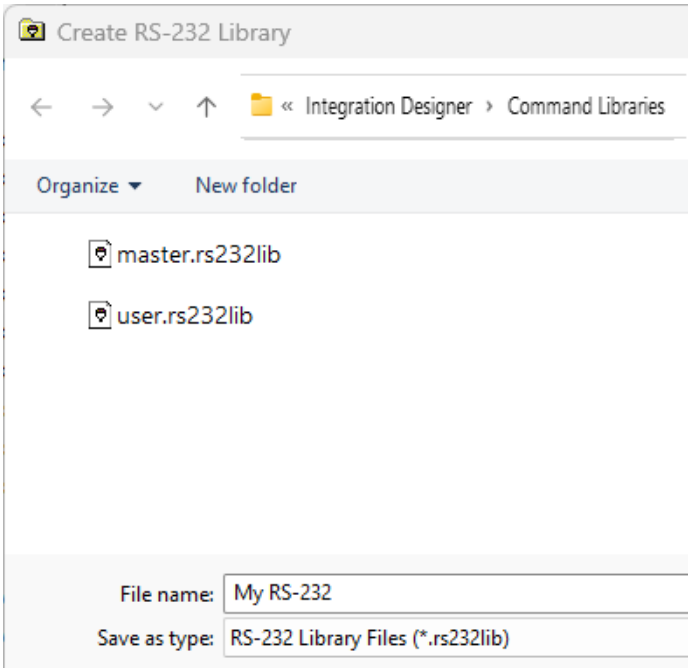
Toolbar: Icons for easy access to frequently used programming features.

Command Settings: Parameters are entered here based on the serial protocol provided by the device manufacturer.

Device List: Displays an alphabetical list sorted by the manufacturer and separated by device type and model numbers. New devices may be added, and existing devices may be edited or deleted.

Function Field: Enter the RS-232 command string in ASCII or HEX format.

CREATING A NEW RS-232 COMMAND LIBRARY



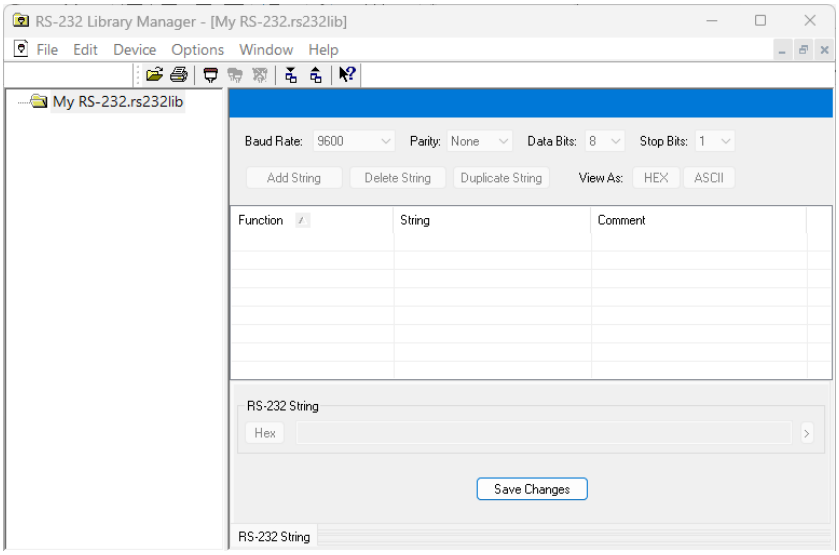
1. Choose New from the File menu, and the Create RS-232 Library dialog box will appear.

2. Enter the name of the new library in the File name box.

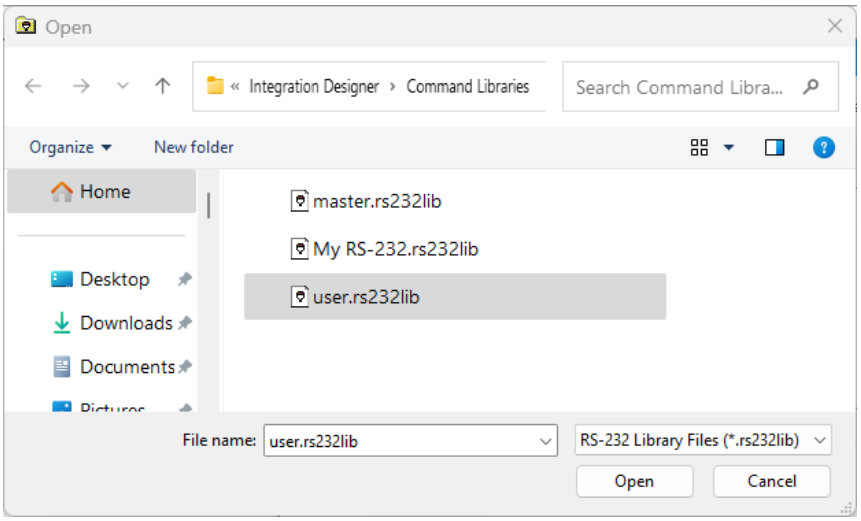
3. Be sure to save your new file in the Command Libraries folder of the Integration Designer software folder.

4. Choose the Save button to create the library.

Once you choose the Save button, the new RS-232 library window will show no remotes in that library, and the fields will be empty.



OPENING AN EXISTING RS-232 COMMAND LIBRARY

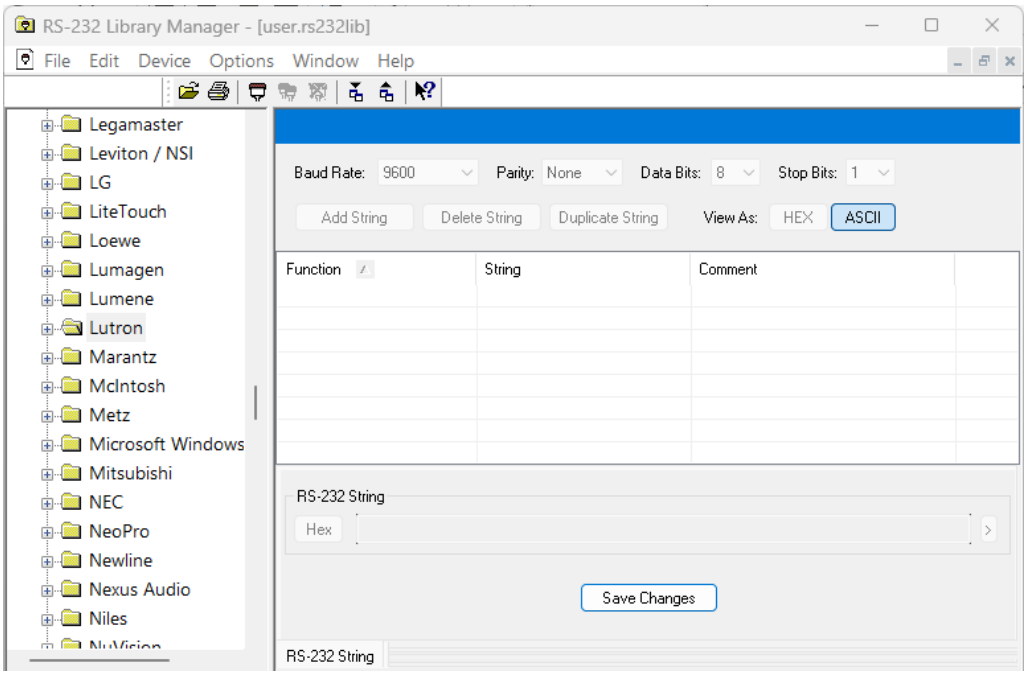


1. Choose Open from the File menu or select the Open icon from the Toolbar.

2. When the Open RS-232 Library dialog box appears, select the library command file you wish to open.

3. Select the Open button.

Once you select the Open button, the RS-232 Library Manager window will open, showing you the devices in that library.



IMPORTING AN RS-232 COMMAND LIBRARY

This programming option allows importing all remotes from one library file into the current library.

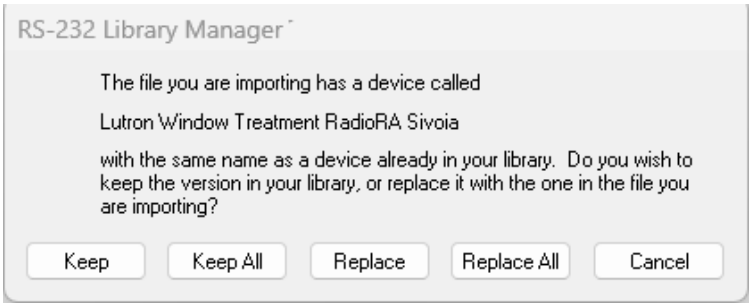
1. Choose Import from the Remote menu or the Import Remote icon.

2. The Import RS-232 Library dialog box will appear.

3. Choose the library you wish to import from the file directories.

4. Select Open to import the library.

Note: If any remotes in the imported file have the same names as remotes in the current library, you will be warned and asked which version to keep.



EXPORTING AN RS-232 COMMAND LIBRARY

This programming option allows you to export selected devices from one library file into a new library file.

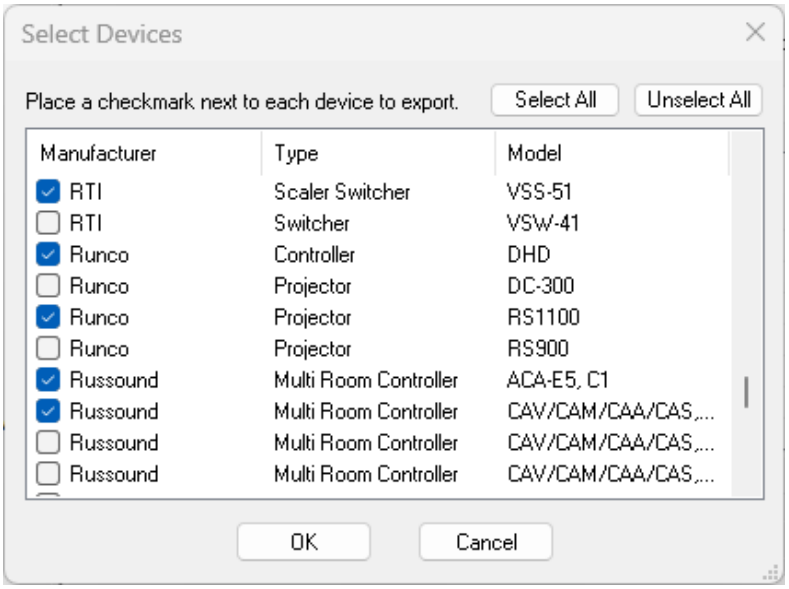
1. Choose Export from the device menu or select the Export Device icon.

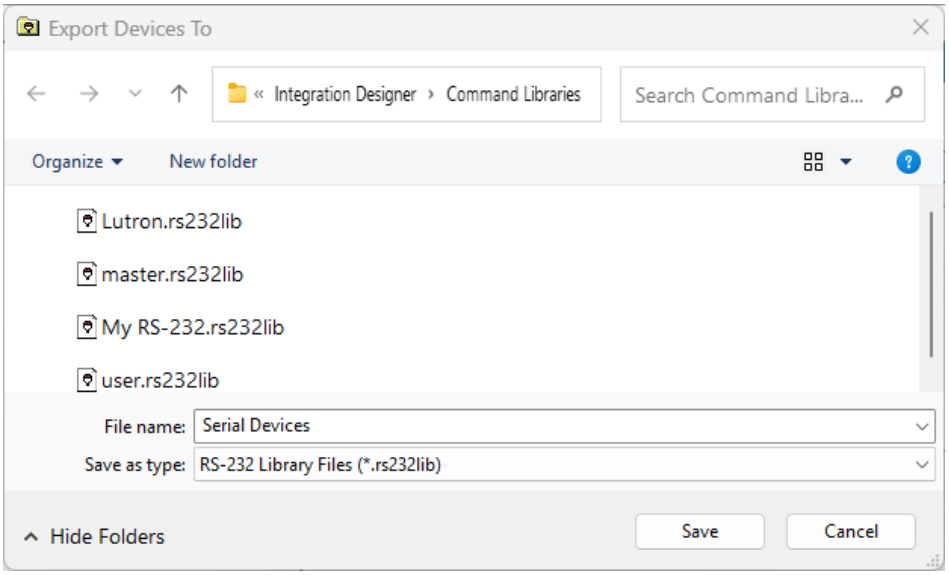
2. The Select Device to Export dialog box will appear.

3. Choose the devices you wish to export to the remote list.

4. Choose OK, and the Export Devices dialog box appears.

5. Enter a name and directory for the new library file and choose OK to save the file.





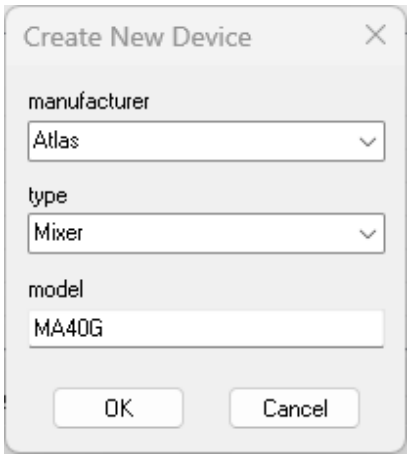
ADDING A NEW RS-232 DEVICE

1. Choose New from the Device menu or the New Device icon.

2. The Create New Device dialog box will appear.

3. Enter the manufacturer's name and the equipment type for the device control being added (such as a matrix switcher, mixer, amplifier, and so on) in the appropriate fields, or choose the name and type from the drop-down lists.

4. Enter the model number of the device.



5. Choose OK to add the device to the RS-232 Command Library.

DUPLICATING A DEVICE LIBRARY

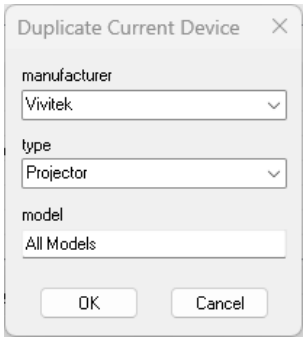
Use this programming option to create a new device that is an exact copy of the selected device in the current RS-232 Command Library.

1. Choose Duplicate from the device menu or select the Duplicate device icon.

2. The Duplicate Current Device dialog box will appear.

3. Edit the new manufacturer's name and equipment type for the duplicated remote control (such as a Matrix Switcher, Mixer, Amplifier, and so on) in the appropriate fields or choose the name and type from the drop-down lists.

4. Edit the model number of the device operated by the remote.

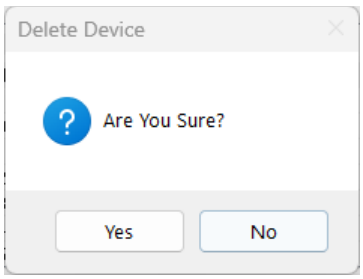


5. Choose OK to duplicate the device into the RS-232 Command Library.

DELETING A REMOTE

1. Choose Delete from the Device menu or select the Delete Device Icon.

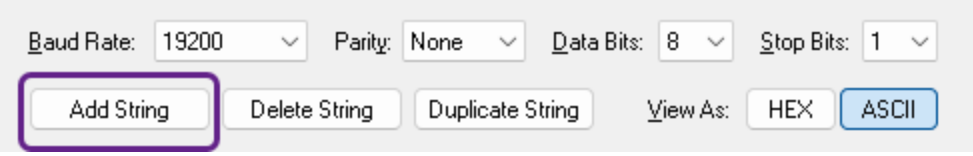
2. The Delete Remote dialog box will appear.



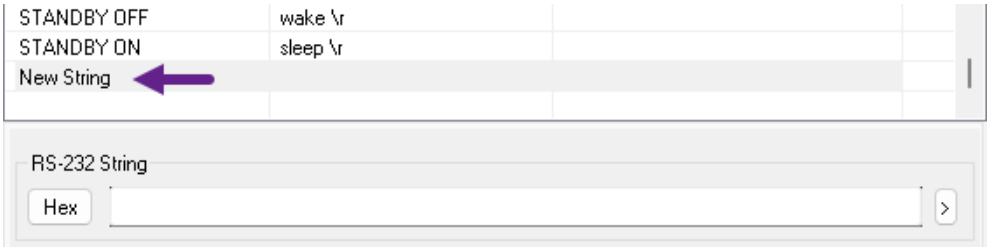
3. Choose OK to delete the device from the RS-232 command library.

ADD FUNCTIONS FOR A DEVICE

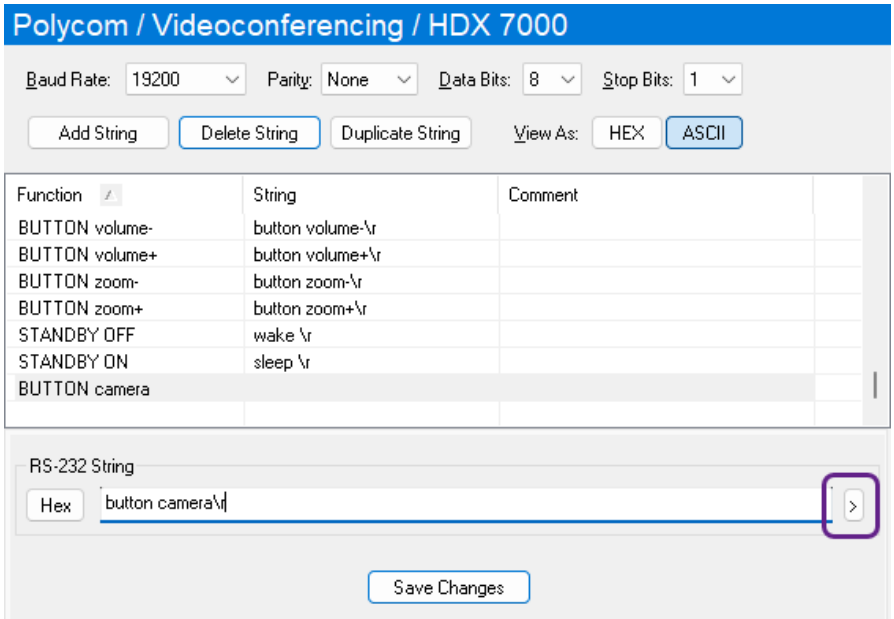
Use the **Add String** command to add commands to an existing RS-232 library.



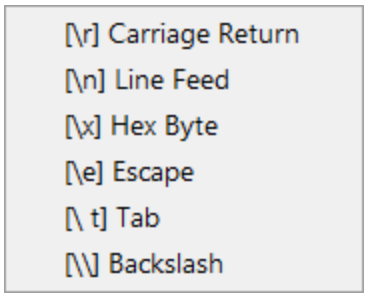
1. A new command string name will be added at the bottom of the list named "New String." Click the name and enter the name of the command.



2. In the RS-232 String field, enter the command. If the command is in hexadecimal format, select the Hex option to the left of the command field. If the Hex field is not selected, the code is assumed to be in ASCII format.



3. If the protocol indicates an "Escape Sequence" is required after a command is sent, select the right arrow to the right of the field. A menu will display various escape sequences.



4. If an escape sequence is not included in the list, you must enter it manually.

5. Select the Save Changes button and start the process again for other commands.

DELETING FUNCTIONS FROM A DEVICE

You can delete functions by selecting them in the function list and then clicking the "Delete String" button.

UPDATING THE RTI MASTER RS-232 COMMAND LIBRARY

Please follow the steps below to update the RTI Master RS-232 Command Library:

1. Before downloading the updated RS-232 Command Library, rename the library file you currently use to store codes you learn. This command library is in the Documents/Integration Designer/Command Libraries folder. We recommend renaming this current file **"User.rs232lib."**

2. Download the new master library from the RTI dealer website.

3. The Master Command Library File is available for download in the command codes section. The entire new library will be downloaded as a file labeled "**master.rs232lib**." Please save the file to a location where you can easily find it again. We recommend saving it to the command libraries section under the Documents/Integration Designer directory.

Please note: The new master library is a read-only file and may not be opened within the RS-232 Library Manager. It can only be opened within the Integration Designer program when programming an RTI Control System.