Version 1.8

PathfinderCore PRO

Pathfinder Core PRO—either Appliance or VM—is a toolbox with powerful features that create efficient workflows and facility management.

Quick Start Guide

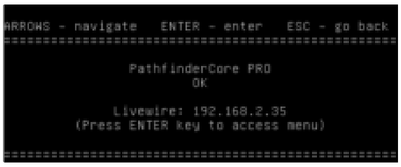
Congratulations on your purchase of the Pathfinder Core PRO. We know that you're anxious to get started, so this section will take you from “out-of-the-box” to “up-and-running” as quickly as possible.

We've made a few basic assumptions:

- That you have some knowledge of network basics and network terminology
- That you are familiar with other Axia Livewire products
- That you have a correctly configured network switch

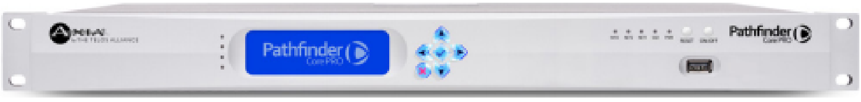
Once you’re all set up, the manual will provide in-depth knowledge of Pathfinder Core PRO's more advanced options. Please refer to the Quick Start Guide below that matches the platform you purchased; R/2, AE-2000 (2021), or Virtual Machine. The R/2 and AE-2000 (2021) look identical from the back and use the same software update packages, but the AE-2000 (2021) platform does not have an LCD display on the front. Once you're completed the product specific section, finish with the "Discovery" section of this Quick Start Guide.

AE-2000 (2021) Platform



```
ARROWS - navigate   ENTER - enter   ESC - go back
==================================================

              PathfinderCore PRO
                      OK

           Livewire: 192.168.2.35
       (Press ENTER key to access menu)

==================================================
```

Virtual Machine

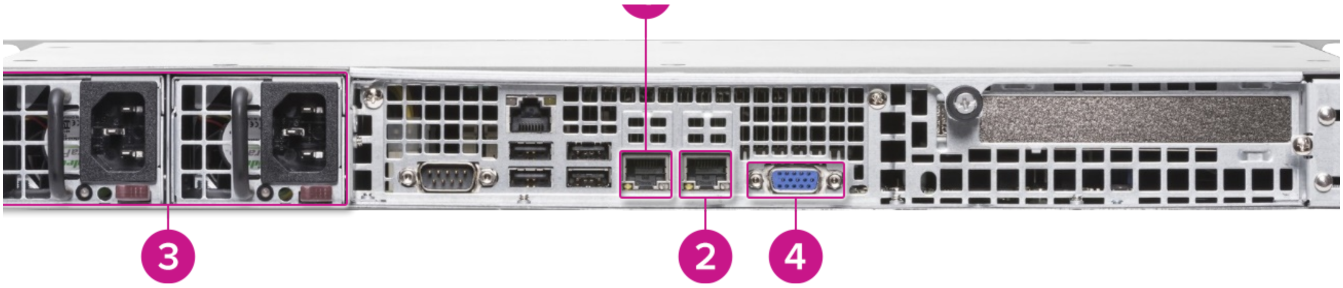


R/2 Platform

AE-2000 (2021) Platform

Connections

1



Reference	Notes
1	Connect the **left** NIC (when viewed from the back of the Pathfinder Core PRO) to your office networl to use panels and other capabilities provided by Pathfinder Core PRO from office computers.
2	Connect the **right** NIC to your AoIP network usinc a Cat-6 Ethernet cable
3	Using the supplied IEC cables, connect the Pathfinder Core PRO's power supply input to AC mains.
4	The VGA connector on the back of the Pathfinder Core PRO studio engine is only needed for configuring the initial IP address settings. Alternatively, you can configure a network in the same range as the default IP addresses and configure via the web interface.

> Warning: Starting with PathfinderCore PRO AE-2000 (2021) platforms shipping with version 1.8 of the software, the order of the nics has changed such that NIC1 (left) is Office and NIC2 (right) is AoIP/Livewire. This is in order to bring consistency across the products using this platform. Software updates to 1.8 of existing systems in the field will not be affected as this is accomplished by adding a config flag during the production build process.

IP Address Configuration

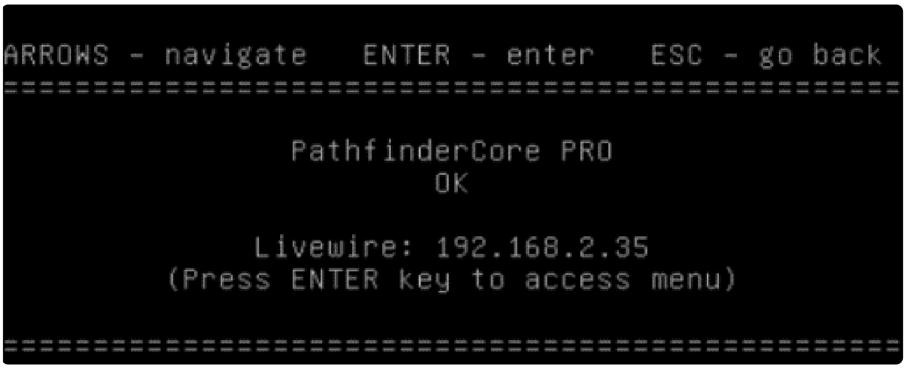
Your system is configured from the factory with default Livewire and office IP address as follows:

Left NIC (Office): 10.10.2.100 with a netmask of 255.255.255.0
Right NIC (AoIP/Livewire): 192.168.2.100 with a netmask of 255.255.255.0

These IP addresses can be changed either by attaching a monitor and USB keyboard and using the intuitive menu or by browsing to the default IP address as described below.

Screen and Keyboard IP Address Assignment

Attach a monitor to the VGA port on the back of PathfinderCore PRO and a USB keyboard to the USB port. You will see a menu dialog similar to this:



```
ARROWS - navigate   ENTER - enter   ESC - go back
==================================================

              PathfinderCore PRO
                      OK

            Livewire: 192.168.2.35
         (Press ENTER key to access menu)

==================================================
```

Menu navigation uses the arrow keys and the ENTER and ESC keys.

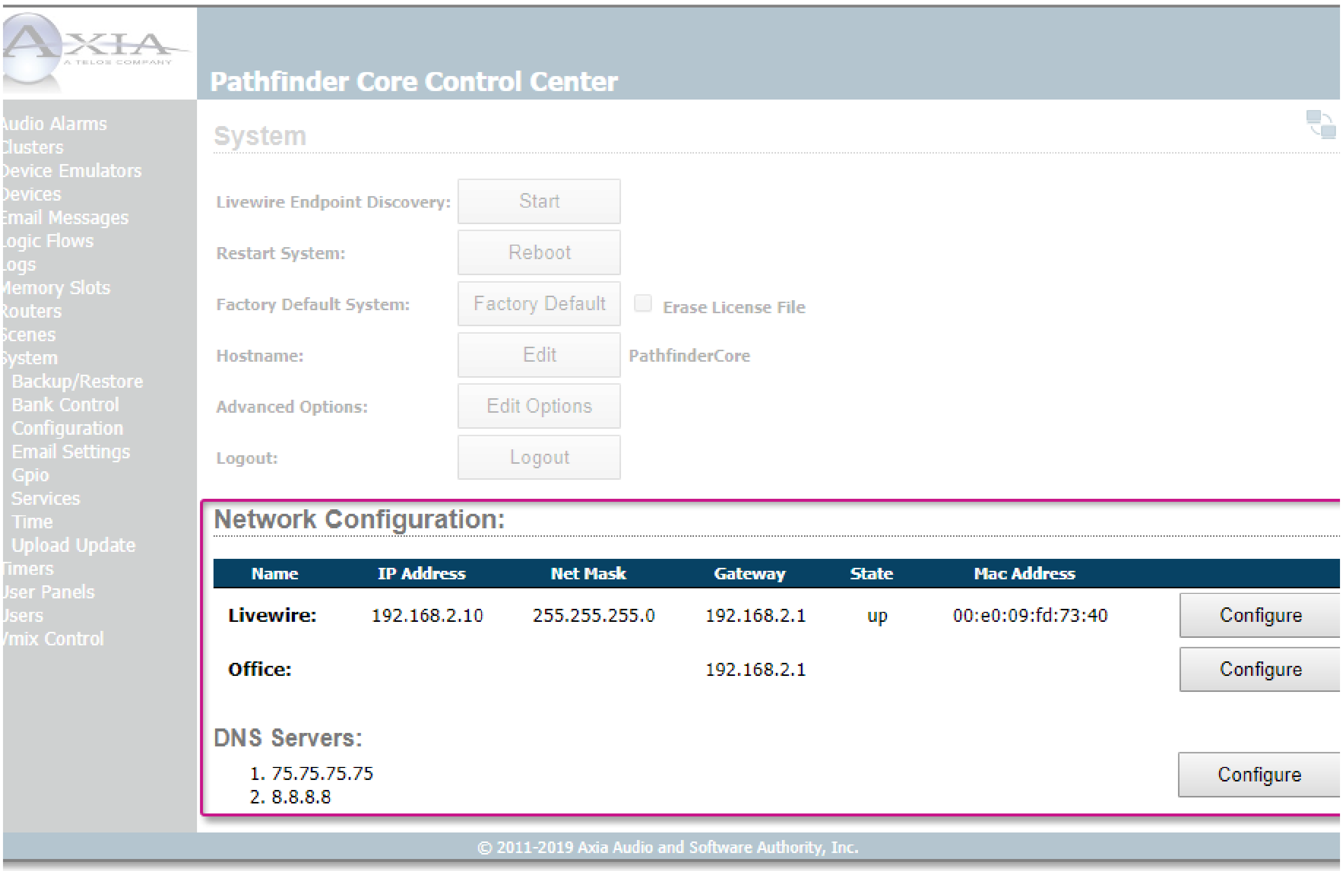
- Press **ENTER** to access the configuration menu
- Use the UP/DOWN arrow keys to highlight the NETWORK menu option, then press **ENTER**
- Use the UP/DOWN arrow keys to highlight the LIVEWIRE network interface, then press **ENTER**
- Configure the IP settings using the UP/DOWN arrow keys to increment or decrement the values and the ENTER or ESC keys to edit or exit each option; you can also type numerals instead of using the arrow keys
- After configuring the IP, NETMASK, and GATEWAY settings, select **DONE** then press **ENTER**
- Repeat this process for the **Office** interface and the **DNS configuration** (DNS may be required for email configuration)
- Once all configurations are complete, navigate to **Reboot and Apply Settings** and press **ENTER;** confirm any request; the system will reboot to the new network configuration.

You may now move on to the Discovery section of this guide.

Discovery

Alternate Network IP Address Configuration

- Attach the right NIC of Pathfinder Core PRO to a network switch
- Attach a computer to the same switch and configure the computer’s IP address to be within the 192.168.2.xx range, but avoid using 192.168.2.100
- Using a web browser, browse to 192.168.2.100
- Log in using the default Username and Password:
  Username = Admin
  Password = Admin
- Click on the Configuration link



Pathfinder Core Control Center

System

Livewire Endpoint Discovery:

Restart System:

Factory Default System: □ Erase License File

Hostname: PathfinderCore

Advanced Options:

Logout:

Network Configuration:

Name	IP Address	Net Mask	Gateway	State	Mac Address
Livewire:	192.168.2.10	255.255.255.0	192.168.2.1	up	00:e0:09:fd:73:40
Office:			192.168.2.1		

DNS Servers:

1. 75.75.75.75
2. 8.8.8.8

© 2011-2019 Axia Audio and Software Authority, Inc.

- Use the Configure buttons to change the IP address, Netmask, and Gateway for both the Livewire and Office network interfaces
- Click the Reboot button to restart with the new IP addresses

You may now move on to the Discovery section of this guide.

Discovery

Virtual Machine

Approved Virtual Machine Hypervisors

The Virtual Machine (VM) instance of Pathfinder Core Pro has been verified on the following VM hypervisor engines:

- VMWare 6.7.0
- Hyper-V Server 2016
- ProxmoxVE 5.1-41
- VirtualBox 5.2
- Stratus everRun 7.5.05

This document does not cover the setup of a VM hypervisor engine, and it is the responsibility of the customer to set up and be knowledgeable of the VM engine in use.

Minimum VM Instance Requirements

The virtual instance created should have the following minimum settings applied:

- 2 GB RAM memory
- 4 GB hard drive storage (fixed allocation)
- 2 processor allocation
- Bridged network adapter (minimum of 1 bridged for the Livewire network)
    - It is recommended to use Paravirtualized VirtIO, VMWare Vmxnet3, or Intel Pro options for network interface emulation.

> Important Note: Larger amounts of RAM, faster processors, more processors for better parallelization, and additional hard disk space can be used for large systems to improve performance and capacity or to make more space available for logging.

Installation

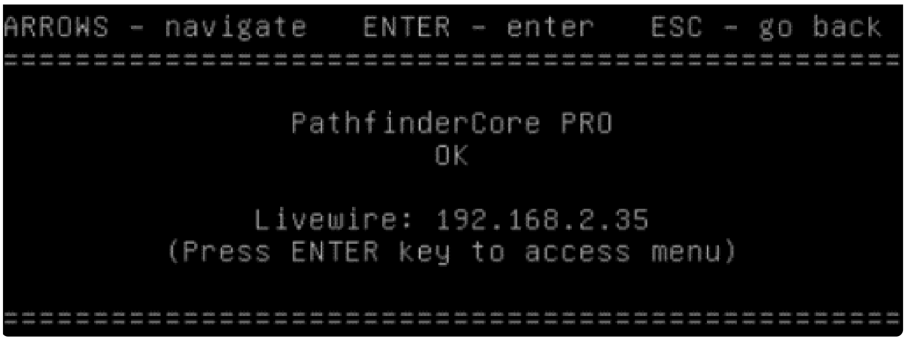
Once the VM instance is defined, point the VM to load the ISO file provided with the purchase. Start up the VM instance and the system will begin to install.

The system will ask you to type 20 random characters. Just tap random keys on the keyboard at random rates of speed until the system tells you to stop. This will initialize the random number generator. When complete, the system will instruct you to press the ENTER key.

The system will warn you about overwriting the hard drive (a virtualized instance). The system expects to find a virtualized SATA drive, and if it cannot be found, the installation will not continue. If found, it requests a second confirmation. Press "y" to proceed. At completion, the system will ask for removal of the ISO image. Make sure you remove the ISO pointer before proceeding (eject the virtual CDROM). Press any key to reboot the new installation. After bootup, the first step is to assign an IP address to the system.

IP Address Configuration

You must assign an IP Address to your Pathfinder Core PRO VM instance before you can use it. The screen at the VM terminal can be navigated using the arrow keys and the ENTER and ESC keys.

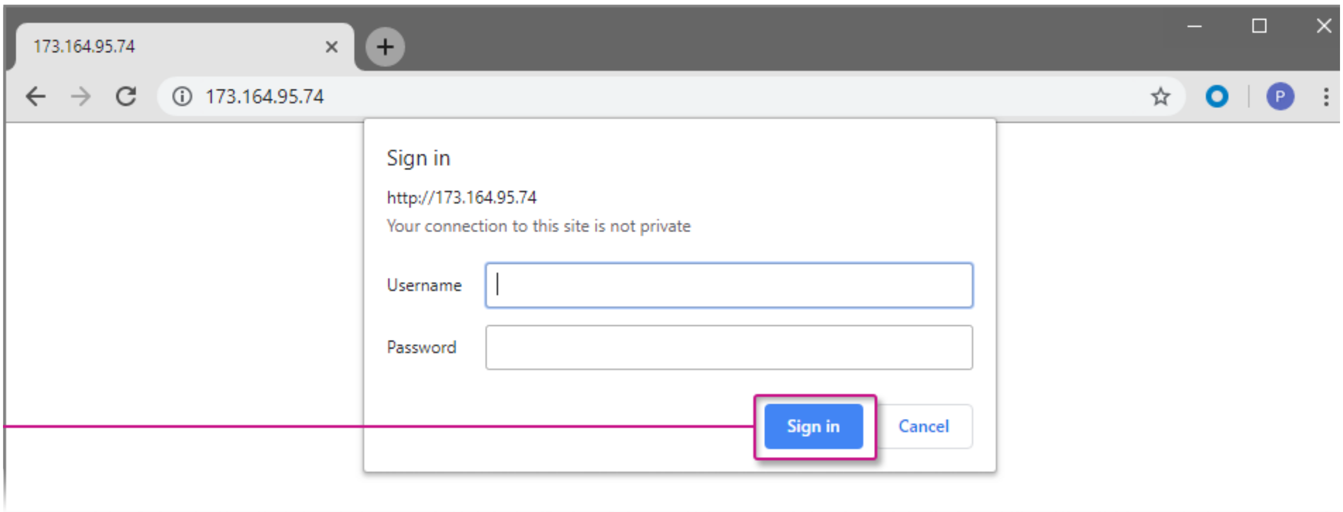


```
ARROWS - navigate    ENTER - enter    ESC - go back
====================================================

               PathfinderCore PRO
                       OK

             Livewire: 192.168.2.35
         (Press ENTER key to access menu)

====================================================
```

- Press **ENTER** to access the configuration menu
- Use the UP/DOWN arrow keys to highlight the NETWORK menu option, then press **ENTER**
- Use the UP/DOWN arrow keys to highlight the LIVEWIRE network interface, then press **ENTER**
- Configure the IP settings, using the UP/DOWN arrow keys to increment or decrement the values and the ENTER or ESC keys to edit or exit each option; you can also type numerals instead of using the arrow keys
- After configuring the IP, NETMASK, and GATEWAY settings, select **DONE** and press **ENTER**
- Repeat this process for the **Office** interface and the **DNS configuration** (DNS may be required for email configuration)
- Once all configurations are complete, navigate to **Reboot and Apply Settings** and press **ENTER;** confirm any request; the system will reboot using the new network configuration

First Time Licensing Access and Setup

- Access the web interface by typing in the IP address configured earlier in a web browser; an authentication window will appear; type the following values:
  Username = Admin
  Password = Admin
- Select **Sign In** to load the Pathfinder Core PRO control panel



173.164.95.74

Sign in

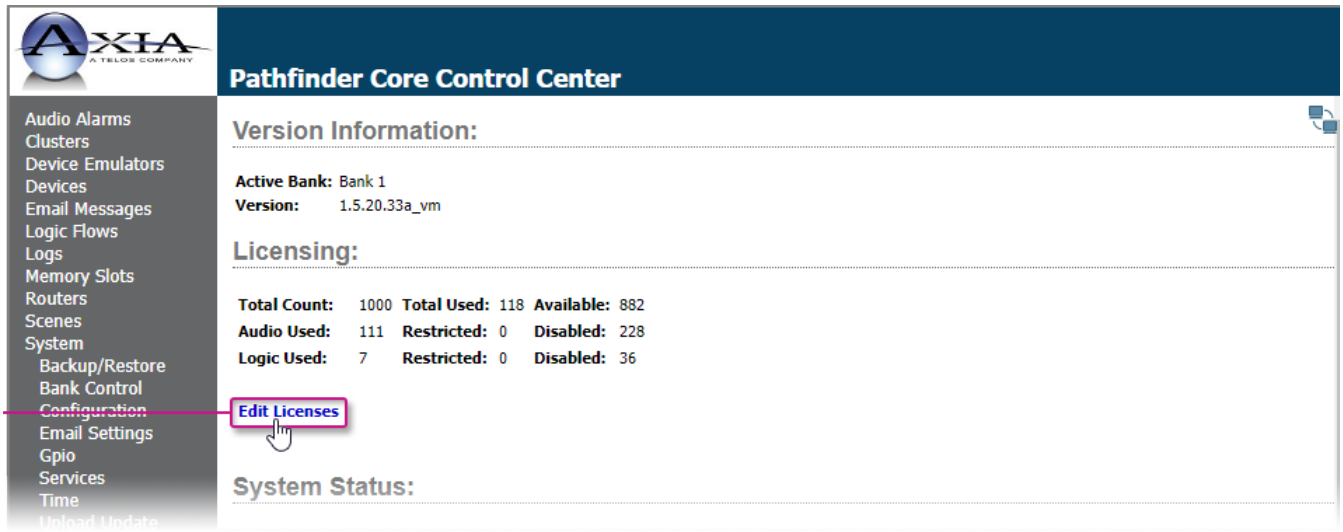
http://173.164.95.74

Your connection to this site is not private

Username

Password

- After logging in, you must enter a License; click the **Edit Licenses** link on the System page



Pathfinder Core Control Center

Audio Alarms
Clusters
Device Emulators
Devices
Email Messages
Logic Flows
Logs
Memory Slots
Routers
Scenes
System
Backup/Restore
Bank Control
Configuration
Email Settings
Gpio
Services
Time
Upload Update

Version Information:

Active Bank: Bank 1
Version: 1.5.20.33a_vm

Licensing:

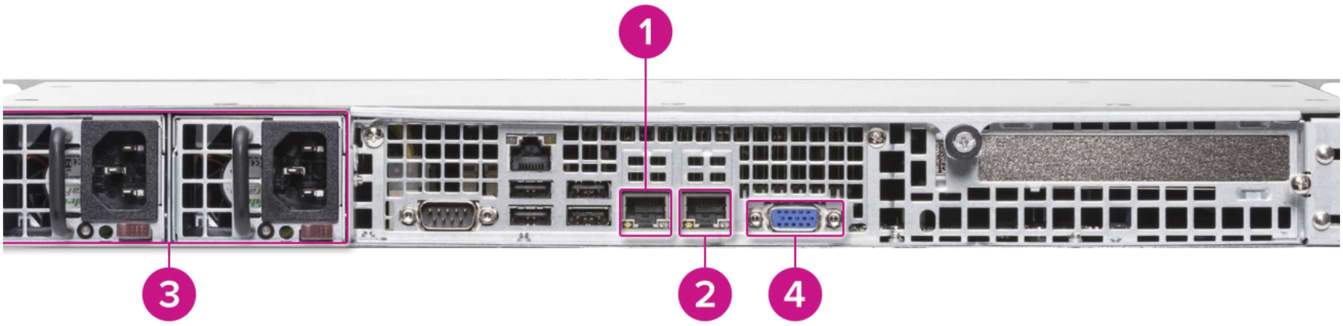
Total Count:	1000	**Total Used:**	118	**Available:**	882
Audio Used:	111	**Restricted:**	0	**Disabled:**	228
Logic Used:	7	**Restricted:**	0	**Disabled:**	36

System Status:

- Click the **New License** item in the list and type the **Access Code** you received after purchasing Pathfinder Core PRO into the *Access Code* field
- Click **Generate Request;** a *Request Code* will appear that matches your access code
- Copy the **Request Code** and **Access Code,** then click the **Generate License** link to open the Pathfinder licensing web site in a new window
- Create a new account; if you have an existing account, log in using your username and password
- The *Activation Type* drop-down, *Access Code*, and *Request Code* fields should be pre-populated with the information you entered in Pathfinder CORE Pro; if they are not:
  - Verify *Activation Type* is set to **Pathfinder Core PRO VM or VML** (depending on your purchase)
  - Paste or type your **Access Code** in the *AccessCode* field
  - Paste or type your **Request Code** in the *RequestCode* field
- The remaining fields are customer information needed required registration of the product; these fields should be completed using end customer details, not those of the contracting integrator or installer
- Once submitted, the account will be generated and license information will be available; copy the license information to the **License Key** field on the *Licenses* page
- Once a valid license is in place, reboot to make sure the license is active; once the licensing is complete and the networked devices are on line, you may move to the *Discovery* section.

Discovery

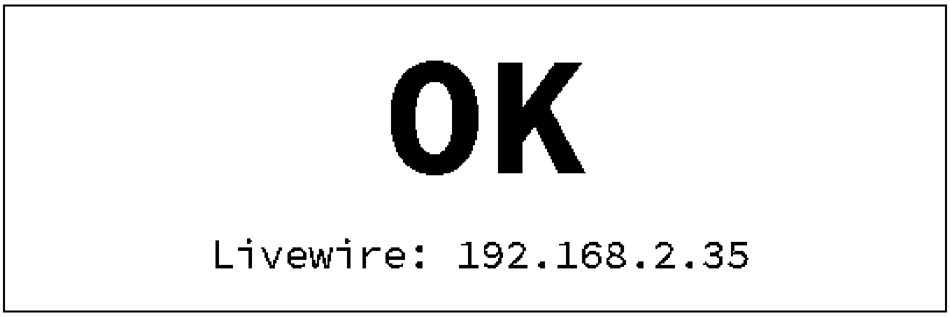
R/2 Platform



Reference	Notes
1	Connect the **left** NIC (as viewed from the back of the Pathfinder Core PRO) to your AoIP network using a Cat-6 Ethernet cable
2	Connect the **right** NIC to your office network to us panels and other capabilities provided by Pathfinder Core PRO from office computers
3	Using the supplied IEC cables, connect the Pathfinder Core PRO's power supply input to AC mains
4	The VGA connector on the back of the Pathfinder Core PRO studio engine is generally not used for normal operations except under the direction of Support; all configuration can be completed using the front panel and online web GUI

IP Address Configuration

Once Pathfinder Core PRO has booted, its display will show the OK logo along with the factory default IP address.



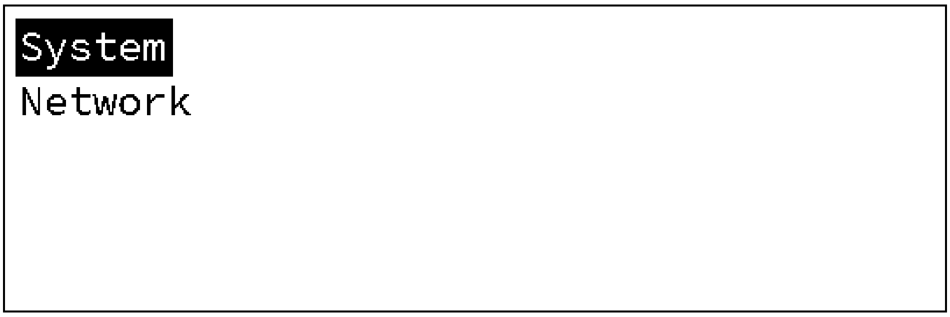
```
OK

Livewire: 192.168.2.35
```

Use the following legend when manipulating the keypad in the sections below:

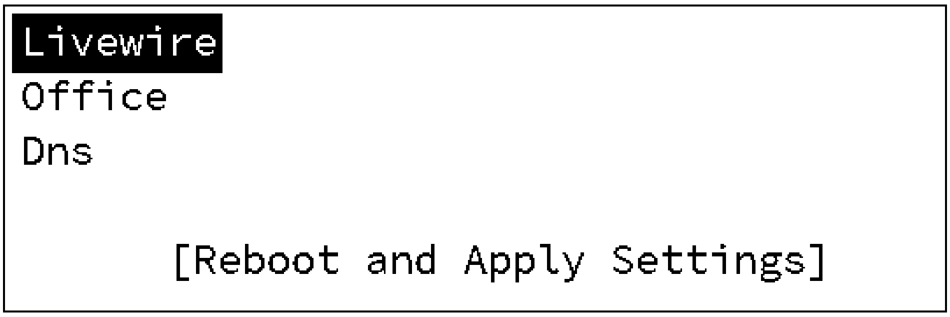
▲	=	Up
▼	=	Down
◀	=	Left
▶	=	Right
✓	=	Enter
✕	=	Escape

Enter the IP address, netmask, and gateway for the system:

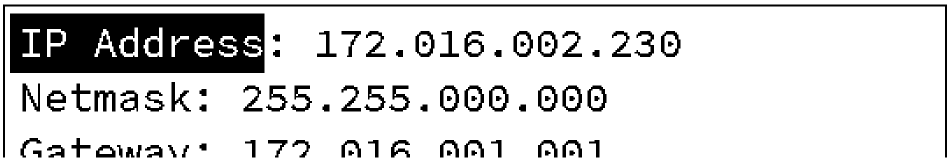
- Tap the **Enter** key in the center of the keypad once; a menu will appear



- Tap the **Down** arrow to highlight the network menu, then tap the **Enter** key to enter the network menu
- Tap the **Enter** key again to enter the Livewire network settings



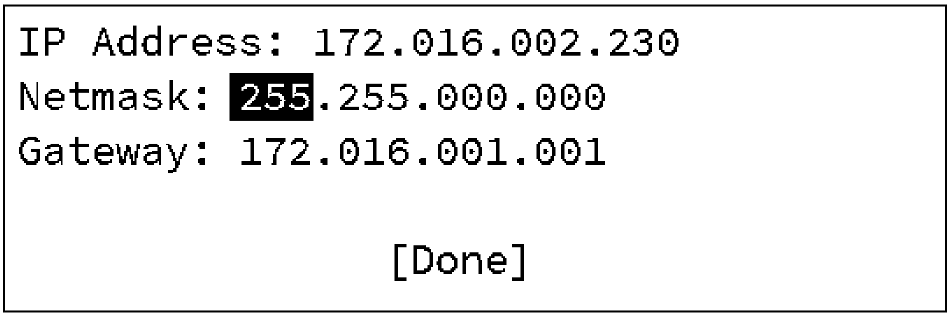
- Tap the **Right** arrow to enter the IP Address, then use the **Up** and **Down arrows** to increment or decrement each number; tapping the **Right** arrow advances the cursor to the next octet



```
IP Address: 172.016.002.230
Netmask: 255.255.000.000
Gateway: 172.016.001.001
```

```
Gateway: 172.016.001.001

                 [Done]
```

- When you reach the end of the IP Address field, tap the **Right** arrow again to return to the main item selection
- Tap the **Down** arrow to move to the Netmask
- Tap the **Right** arrow to enter the netmask assignment section
- Use the **Up** and **Down** arrows to select valid netmask entries, then tap the **Right** arrow to advance to the next octet; the system will only allow you to enter valid netmask addresses

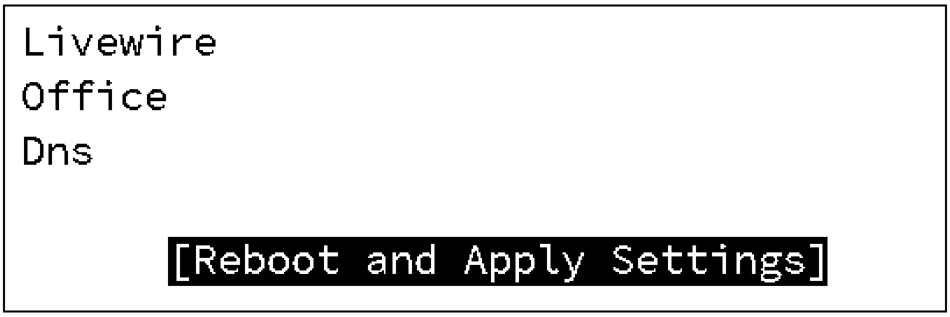


```
IP Address: 172.016.002.230
Netmask: 255.255.000.000
Gateway: 172.016.001.001

                 [Done]
```

- Once the netmask is entered properly, tap the **Right** arrow to return to the main section, then arrow down to the Gateway field
- Repeat the above procedure to set the gateway

> Important Note: Only one gateway is allowed. Whether you set it in the Livewire or Office IP Address section, only the last gateway entered will be used.

- Arrow down to click **Done;** this will return you to the network menu
- Repeat the procedure for the office network and DNS settings
- Once all addresses have been assigned, tap the **Down** arrow to select **Reboot and Apply Settings**



```
Livewire
Office
Dns

      [Reboot and Apply Settings]
```

- Tap **Enter** to reboot the system

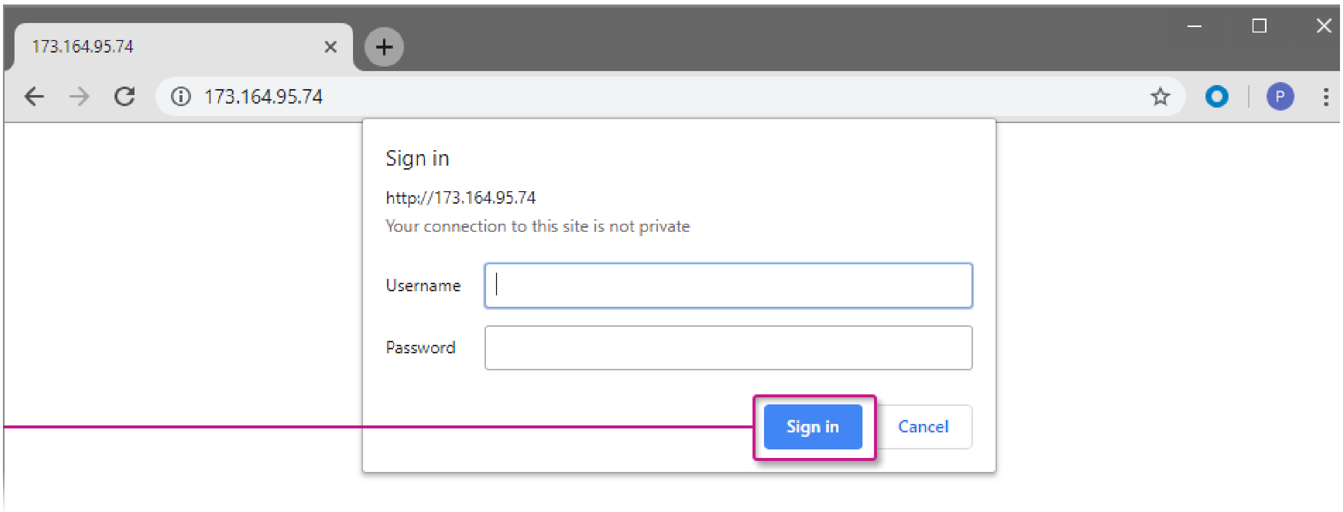
You may now move to the Discovery section

Discovery

Discovery

The next step requires a web browser on a PC connected to either the office or Livewire network.

- If your computer is connected to the office network, open your browser and enter the IP address you assigned to the WAN port into the URL bar
- If your PC is connected to the Livewire network, open your browser and enter the IP you assigned to the Livewire port into the URL bar.
- An authentication window will appear. Type the following values:
  Username = Admin
  Password = Admin
- Select **Sign In** to load the Pathfinder Core PRO control panel



173.164.95.74

Sign in

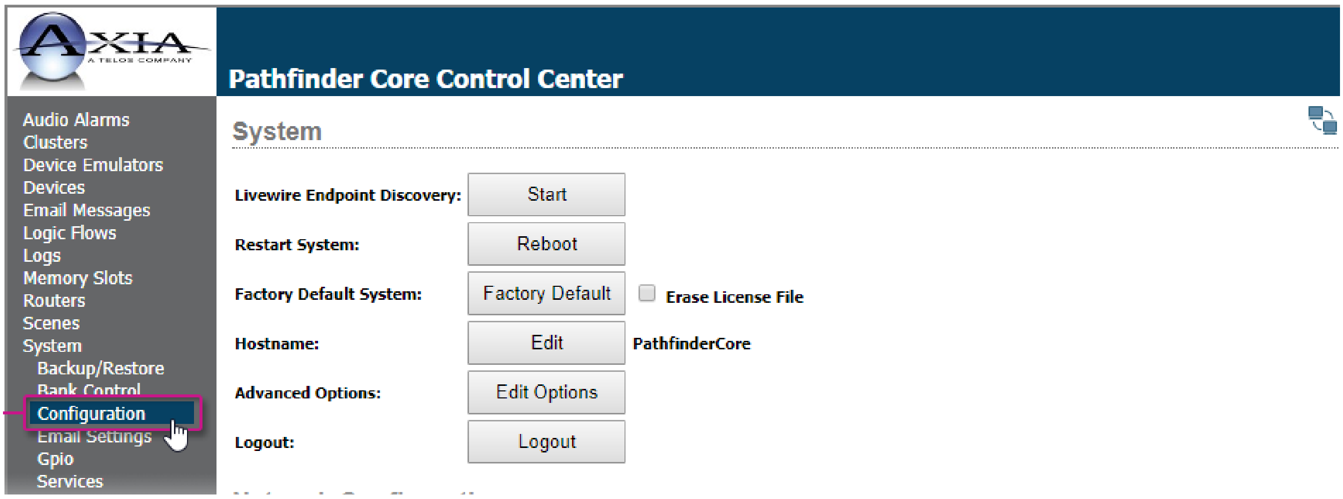
http://173.164.95.74

Your connection to this site is not private

Username

Password

- From the links in the left-hand navigation bar, click **Configuration** under the System heading



Pathfinder Core Control Center

System

Livewire Endpoint Discovery:

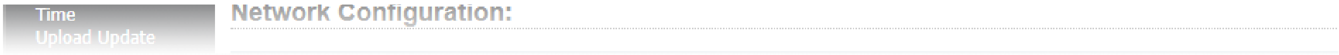
Restart System:

Factory Default System:

Hostname: PathfinderCore

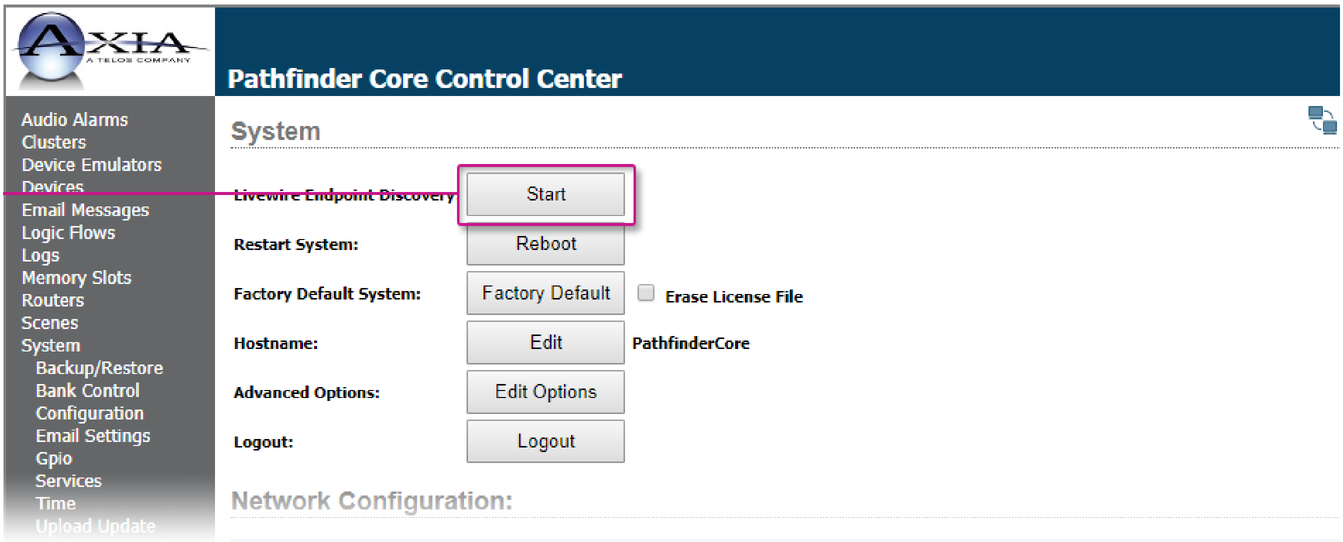
Advanced Options:

Logout:



Network Configuration:

- Next to *Livewire Endpoint Discovery*, click **Start**



Pathfinder Core Control Center

System

Livewire Endpoint Discovery: Start

Restart System: Reboot

Factory Default System: Factory Default ☐ Erase License File

Hostname: Edit PathfinderCore

Advanced Options: Edit Options

Logout: Logout

Network Configuration:

Now would be a good time to grab a cup of coffee while the system evaluates the network, discovers devices, and builds your primary audio and GPIO routers.

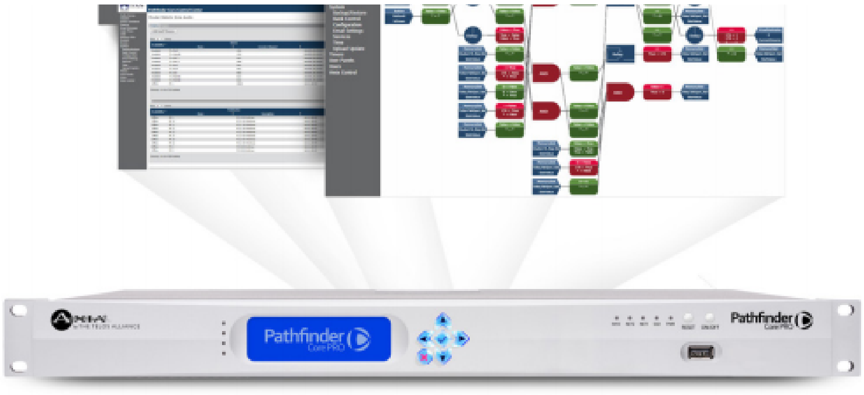
While the Discovery is running, the system will continue to look for new devices. There is no message indicating completion, but if you click the **Devices** link in the navigation bar, you should see Devices populating the Devices list.

Congratulations! Your Pathfinder Core PRO is now configured and ready to start controlling your broadcasting world.

Manual

Pathfinder Core PRO

Installation & User's Guide



User Warnings and Cautions

The installation and service instructions in this manual are for use by qualified personnel only. This equipment is not suitable for use in locations where children are likely to be present. To avoid electric shock, do not perform any servicing other than that contained in the operating instructions unless you are qualified to do so. Refer all servicing to qualified personnel. This instrument has an auto-ranging line voltage input. Ensure the power voltage is within the specified range of 100-240VAC. The "~" symbol, if used, indicates an alternating current supply. Equipment must be connected to protective earthing by means of a power cord connected to a socket outlet. Equipment with an earthing terminal should be connected to a proper ground system with 20 - 14 AWG (0.5 - 1.5mm) wire.

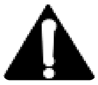
User Warnings and Cautions

Important Safety Symbols

This symbol, whenever it appears, alerts you to the presence of uninsulated, dangerous voltage inside the enclosure - voltage which may be sufficient to constitute a risk of shock. Disconnect equipment from all power sources before any servicing of parts.

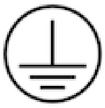
Voltage warning symbol.

This symbol, whenever it appears, alerts you to important operating and maintenance instructions. Read the manual.



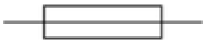
Important instructions symbol.

This symbol, whenever it appears, identifies the associated terminal as a protective earth ground for protection against electric shock in case of a fault. The terminal should be connected to a proper ground system with 20 - 14 AWG (0.5 - 1.5mm) wire.



Earth ground symbol.

This symbol, whenever it appears, indicates a fuse to protect against excessive current. When replacing fuses, use only the type and rating of the original fuse as indicated by the manufacturer. Never install a fuse with a higher current value.



Fuse Symbol.

Hazardous Voltages and Fuses

The instrument power supply incorporates an internal fuse. Hazardous voltages may still be present on some of the primary parts even when the fuse has blown. If fuse replacement is required, replace the fuse only with the same type and value for continued protection against fire.

Electrical Warnings

The product's power cord is the primary disconnect device. The socket outlet should be located near the device and easily accessible. The unit should not be located such that access to the power cord is impaired. If the unit is incorporated into an equipement rack, an easily accessible safety disconnect should be included in the rack design.

To reduce the risk of electrical shock, do not expose this product to rain or moisture. This unit is for indoor use only.

This equipment requires the free flow of air for adequate cooling. Do not block the ventilation openings on the rear and sides of the unit. Failure to allow proper ventilation could damage the unit or create a fire hazard. Do not place the units on a carpet, bedding, or other materials that could interfere with any panel ventilation openings.

If the equipment is used in a manner not specified by the manufacturer, the protection provided by the equipment may be impaired.

USA Class A Computing Device User Information

This equipment generates, uses, and can radiate radio-frequency energy. If it is not installed and used as directed by this manual, it may cause interference to radio communication. This equipment complies with the limits for a Class A computing device as specified by FCC rules, part 15, subpart j, which are designed to provide reasonable protection against such interferences when this type of equipment is operated in a commercial environment. Operation of this equipment in a residential area is likely to cause interference. If it does, the user will be required to eliminate the interference at the user's expense. Note: Objectionable interference to TV or radio reception can occur if other devices are connected to this device without the use of shielded interconnect cables. FCC rules require the use of shielded cables.

Canadian Class A Computing Device Warning

This digital apparatus does not exceed the Class A limits for radio noise emissions set out in the radio interference regulations of the Canadian department of communications.

Le présent appareil numérique n’émet pas de bruits radioélectriques dépassant les limites applicables aux appareils numériques (de Class A) prescrites dans le règlement sur le brouillage radioélectrique édicté par le ministère des communications du Canada.

CE Conformance Information

This device complies with the requirements of the EEC council directives:

- 93/68/EEC (CE marking)
- 73/23/EEC (Safety – Low voltage directive)
- 89/336/EEC (Electromagnetic compatibility

Conformity is declared to those standards: EN50081-1, EN50082-1.

Trademarks, Patents, and Licenses

Telos is a trademark of TLS Corp. All other trademarks are the property of their respective holders.

All versions, claims of compatibility, trademarks, etc. of hardware and software products not made by The Telos Alliance which are mentioned in this manual or accompanying material are informational only. The Telos Alliance makes no endorsement of any particular product for any purpose, nor claims any responsibility for operation or accuracy. We reserve the right to make improvements or changes in the products described in this manual which may affect the product specifications or to revise the manual without notice.

Patent information can be found at www.TelosAlliance.com/legal.

Software Updates

The features and operations of most products and equipment are determined largely by software. Telos Alliance strives to provide the most stable and feature-rich software available, and we encourage you to check for software updates from time to time by visiting our website or by contacting us directly.

Feedback

We welcome feedback on any aspect of our products or this manual. In the past, many good ideas from users have made their way into software revisions or new products. Please contact us with your comments or suggestions.

We support you

By phone or fax: You may reach our Telos Support Team in emergencies by calling +1 216-622-0247. For billing questions or other non-emergency technical questions, call +1 216-241-7225 between 9:00 AM to 5:00 PM USA Eastern Time, Monday through Friday.

By e-mail: Non-emergency technical support is available at Support@TelosAlliance.com.

By web: A direct support request may be initiated via our website at https://www.telosalliance.com/support-request.

Our main product support page also has a variety of helpful information at www.telosalliance.com/support.

Service

You must contact Telos Alliance before returning any equipment for factory service. We will need your unit's serial number, located on the back of the unit. We will issue a return authorization number, which must be written on the exterior of your shipping container. Please do not include cables or accessories unless specifically requested by the Technical Support Engineer. Be sure to adequately insure your shipment for its replacement value. Packages without proper authorization may be refused. US customers, please contact Telos Alliance Technical Support at +1-216-622-0247. All other customers should contact their local representative to make arrangements for service.

Warranty

For the latest Telos Alliance warrant information, please visit www.telosalliance.com/warranty.

Product Registration

Register your product today to get the full benefits of our warranty, support, and product updates at www.telosalliance.com/product-registration.

Company Headquarters

Telos Alliance, 1241 Superior Avenue, Cleveland, OH 44114 USA. Telephone +1-216-241-7225.

Overview

Pathfinder Core PRO offers you ultimate control of your Axia network. It is a toolbox that allows you to fashion your own custom workflows for your users. At its foundation it provides router control. By communicating with Axia devices on your network, it brings the sources and destinations together into a common user interface that can then be used to make route changes anywhere in the system.

It also offers control over console parameters, metering, audio levels, and many more parameters that an Axia system provides. Plus, it allows you to design your own user interfaces to display this information and provide your users with whatever degree of control and information over the system you deem appropriate. Building on the highly successful PathfinderPro product line, Pathfinder Core PRO strives to make the information about your Axia system more dynamic, visible, and accessible than ever before.

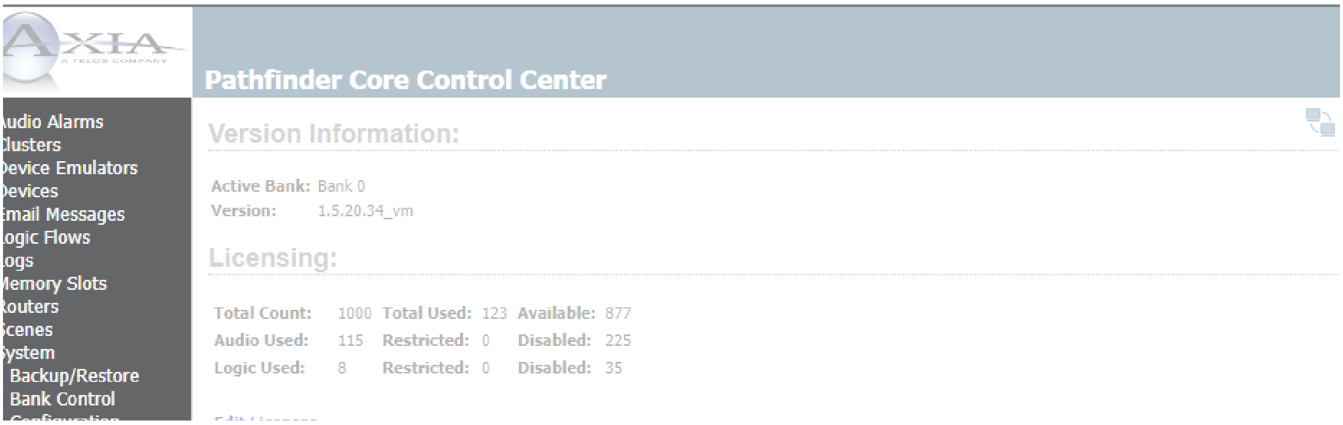
Web Browsers

Pathfinder Core PRO uses modern web browser tools to bring a rich and dynamic user experience to the product, including web sockets and HTML5. This comes at the expense of being unable to support older web browsers. For example, web sockets are only supported in Internet Explorer 10 and later. Most of our development work has been done using Google Chrome which should provide the most consistent and best user experience. If you find things that do not work or behave properly in a specific browser or do not update the data properly, please feel free to file a report with Axia support so that we can investigate the issue. We are always striving to improve the user experience across all browsers.

If data does not seem to display properly or if data is missing, refreshing the webpage is sometimes helpful. Many browsers also have a key combination that will force a refresh from the serving device rather than relying on cached data; for instance, holding the CTRL and SHIFT keys while clicking the refresh button in Google Chrome.

Navigation Bar

All links necessary to configure, control, and monitor your Pathfinder Core PRO system are on the left-hand side of the Pathfinder Core PRO web pages. This will be referred to throughout the manual as the **Navigation Bar**.

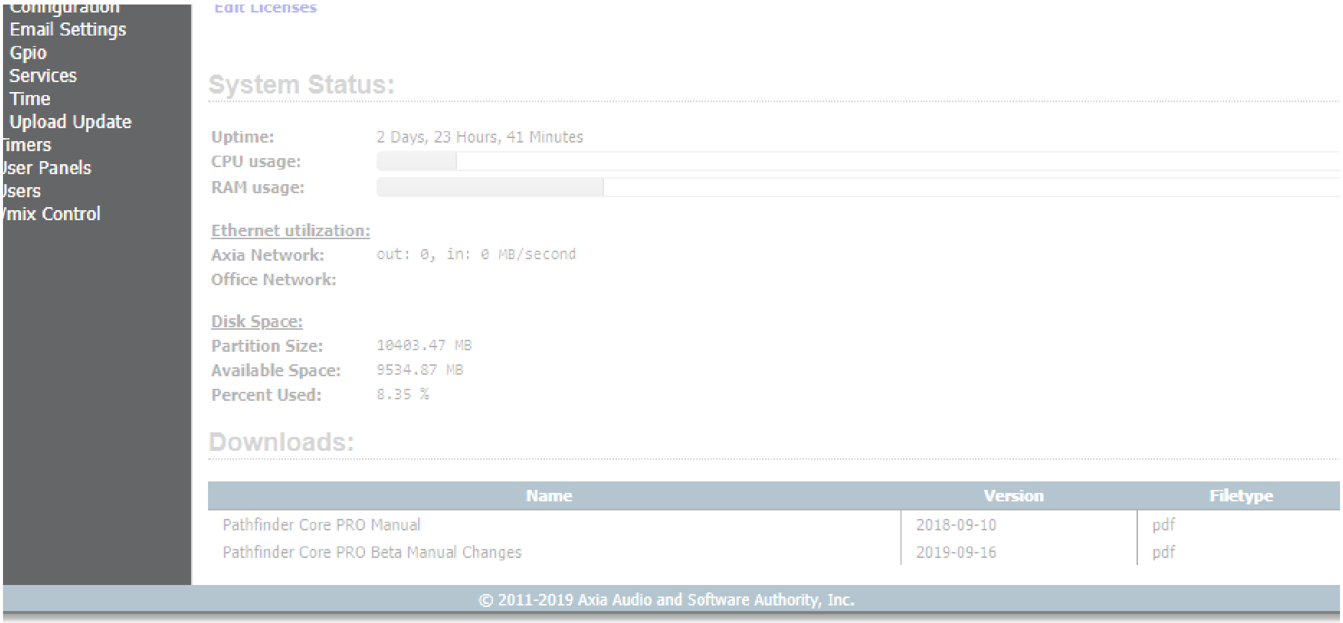


Version Information:

Active Bank: Bank 0
Version: 1.5.20.34_vm

Licensing:

Total Count:	1000	Total Used:	123	Available:	877
Audio Used:	115	Restricted:	0	Disabled:	225
Logic Used:	8	Restricted:	0	Disabled:	35



System Status:

Uptime: 2 Days, 23 Hours, 41 Minutes
CPU usage:
RAM usage:

Ethernet utilization:
Axia Network: out: 0, in: 0 MB/second
Office Network:

Disk Space:
Partition Size: 10403.47 MB
Available Space: 9534.87 MB
Percent Used: 8.35 %

Downloads:

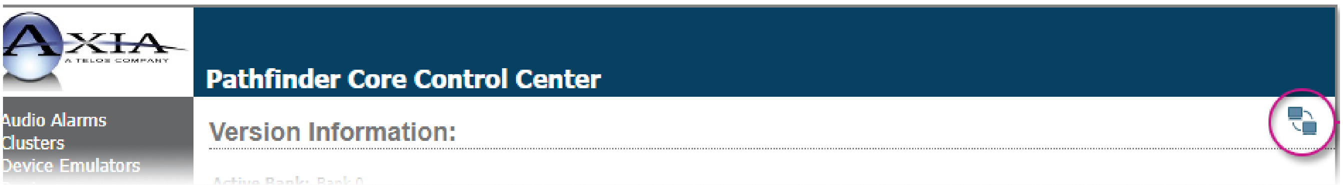
Name	Version	Filetype
Pathfinder Core PRO Manual	2018-09-10	pdf
Pathfinder Core PRO Beta Manual Changes	2019-09-16	pdf

© 2011-2019 Axia Audio and Software Authority, Inc.

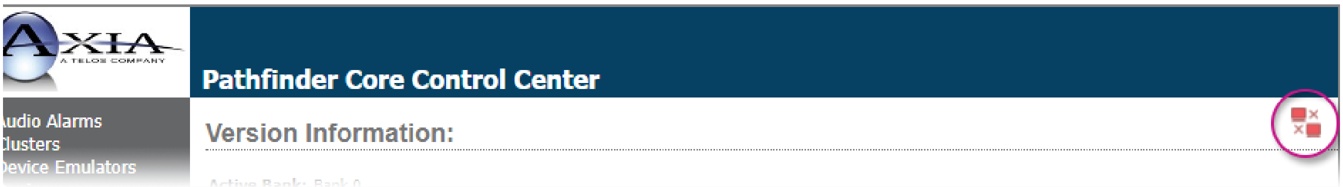
While the links on this bar are presented alphabetically for ease of use, this manual will skip from link to link as we go through a logical progression of using the system.

Network Connection Icon

The right corner of each web page displays a network connection icon. In normal situations it should look like:



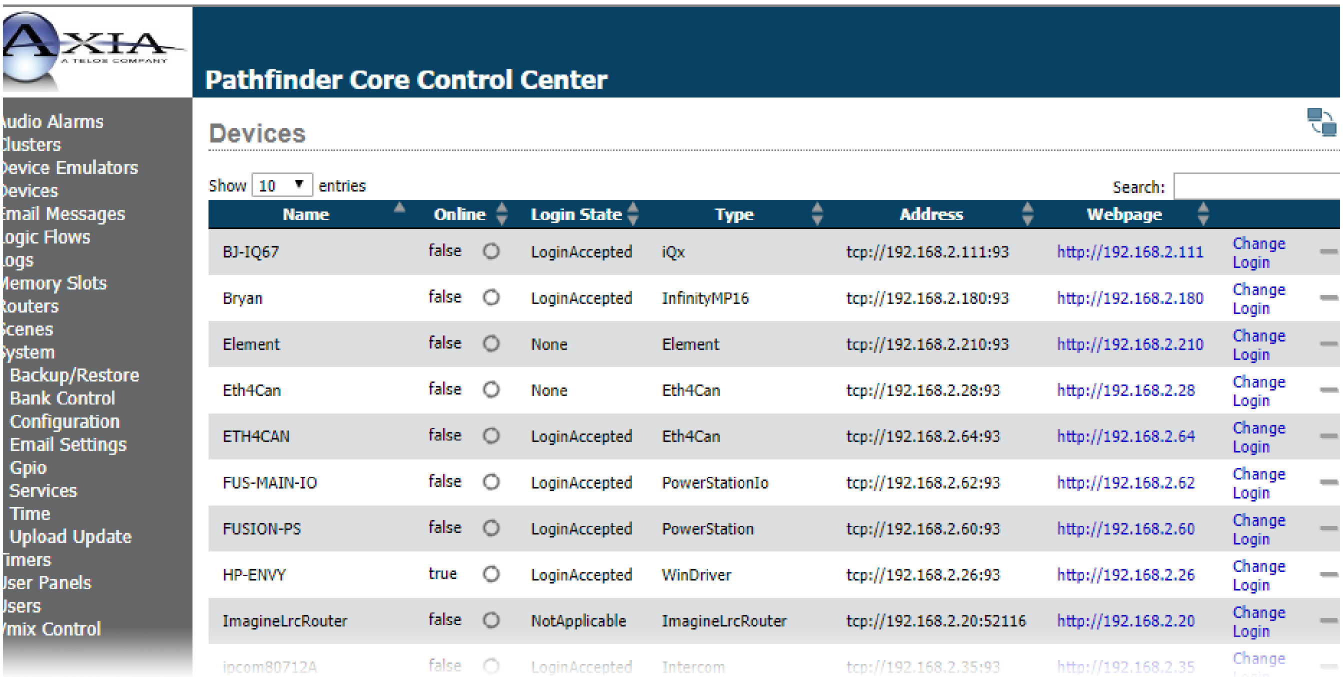
This icon tells you whether the underlying web socket used to obtain and set information from the web page to and from Pathfinder Core PRO is open and communicating properly. If it is not, the icon will look like this, indicating a communication problem between your browser and Pathfinder Core PRO:



Check network connectivity between your browser and Pathfinder Core PRO and verify Pathfinder Core PRO is booted and functioning properly. An additional symptom of this problem may be that configurations look as if they were blank. Before panicking and assuming that your Pathfinder Core PRO has somehow been wiped of its settings, check to see if the network icon is in an unconnected state.

Lists: Searching and Sorting

Many of the configuration pages in Pathfinder Core PRO need to present lists of data. For example:

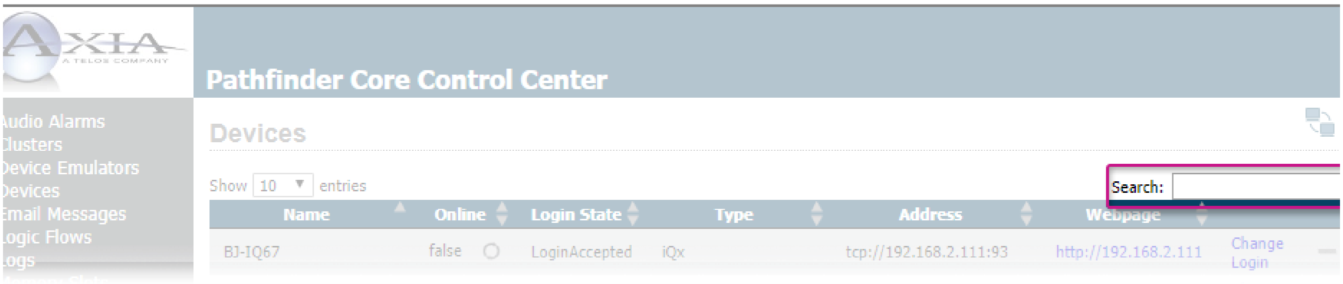


Devices

Show 10 entries Search:

Name	Online		Login State	Type	Address	Webpage		
BJ-IQ67	false	○	LoginAccepted	iQx	tcp://192.168.2.111:93	http://192.168.2.111	Change Login	−
Bryan	false	○	LoginAccepted	InfinityMP16	tcp://192.168.2.180:93	http://192.168.2.180	Change Login	−
Element	false	○	None	Element	tcp://192.168.2.210:93	http://192.168.2.210	Change Login	−
Eth4Can	false	○	None	Eth4Can	tcp://192.168.2.28:93	http://192.168.2.28	Change Login	−
ETH4CAN	false	○	LoginAccepted	Eth4Can	tcp://192.168.2.64:93	http://192.168.2.64	Change Login	−
FUS-MAIN-IO	false	○	LoginAccepted	PowerStationIo	tcp://192.168.2.62:93	http://192.168.2.62	Change Login	−
FUSION-PS	false	○	LoginAccepted	PowerStation	tcp://192.168.2.60:93	http://192.168.2.60	Change Login	−
HP-ENVY	true	○	LoginAccepted	WinDriver	tcp://192.168.2.26:93	http://192.168.2.26	Change Login	−
ImagineLrcRouter	false	○	NotApplicable	ImagineLrcRouter	tcp://192.168.2.20:52116	http://192.168.2.20	Change Login	−
ipcom80712A	false	○	LoginAccepted	Intercom	tcp://192.168.2.35:93	http://192.168.2.35	Change Login	−

There are a few conventions that apply whenever you see a list like this in Pathfinder Core PRO. The first is the Search box in the right corner.



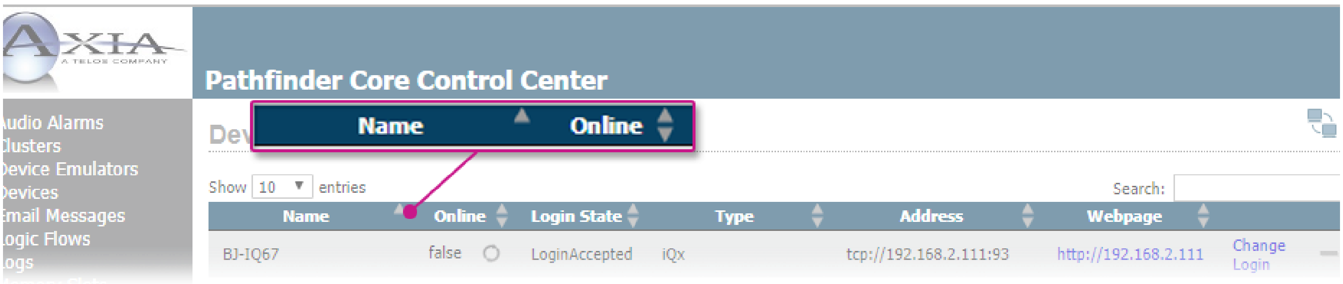
Devices

Show 10 entries Search:

Name	Online		Login State	Type	Address	Webpage		
BJ-IQ67	false	○	LoginAccepted	iQx	tcp://192.168.2.111:93	http://192.168.2.111	Change Login	−

Typing in the search box will dynamically filter the list. These filters are also often stored by the browser, so if you return to a page and do not see the expected data in the list, check the search field to see if you have a filter on the list.

Next, notice that each column header has arrows next to the header names. Clicking on the header will sort the list according to the data in that column. The column that only has the single arrow represents the current sort column. Clicking on that column a second time will flip the arrow to the other direction and flip the sort from incrementing to decrementing and back again.

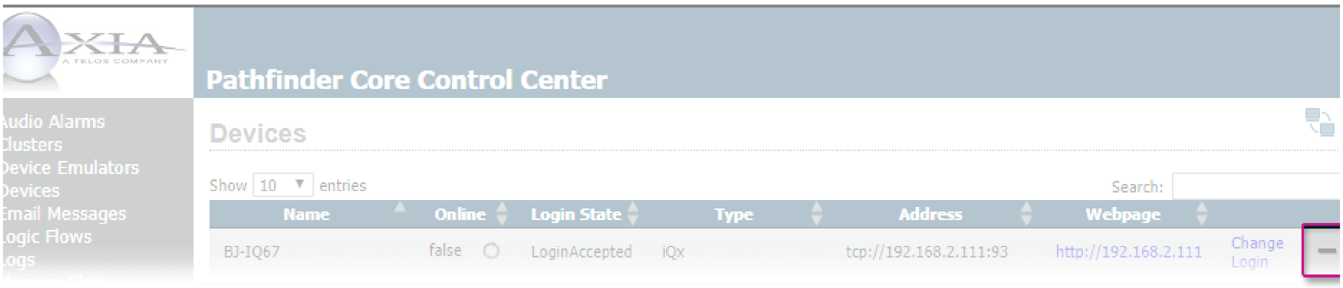


Devices

Show 10 entries Search:

Name	Online		Login State	Type	Address	Webpage		
BJ-IQ67	false	○	LoginAccepted	iQx	tcp://192.168.2.111:93	http://192.168.2.111	Change Login	−

Many of the lists will have the hyphen sign in the last column of the list. Clicking this **Delete icon** removes the selected list item from the system.



Pathfinder Core Control Center

udio Alarms
lusters
evice Emulators
evices
mail Messages
ogic Flows
ogs

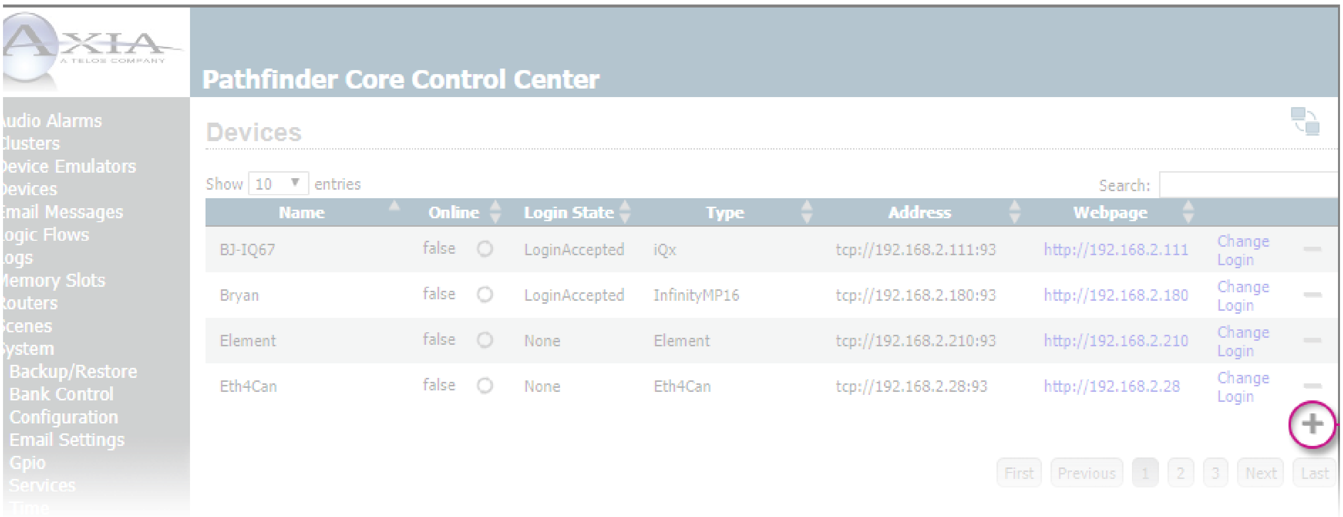
Devices

Show 10 entries

Search:

Name	Online		Login State	Type	Address	Webpage		
BJ-IQ67	false	○	LoginAccepted	iQx	tcp://192.168.2.111:93	http://192.168.2.111	Change Login	—

Many of the lists will also have a plus sign at the bottom of the list. This sign indicates that items may be added to the list. Clicking the **Add icon** will usually open another dialog box with specifics for adding the new item.



Pathfinder Core Control Center

udio Alarms
lusters
evice Emulators
evices
mail Messages
ogic Flows
ogs
lemory Slots
outers
cenes
ystem
Backup/Restore
Bank Control
Configuration
Email Settings
Gpio
Services
Time

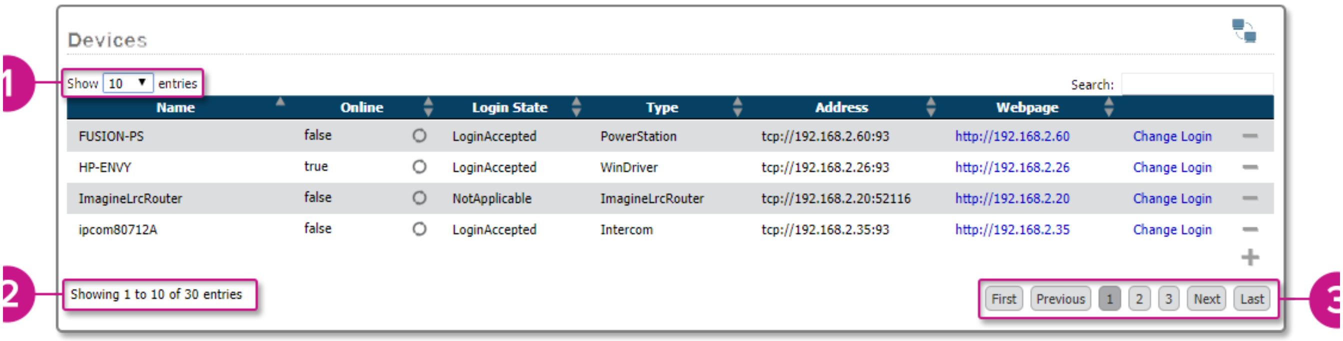
Devices

Show 10 entries

Search:

Name	Online		Login State	Type	Address	Webpage		
BJ-IQ67	false	○	LoginAccepted	iQx	tcp://192.168.2.111:93	http://192.168.2.111	Change Login	—
Bryan	false	○	LoginAccepted	InfinityMP16	tcp://192.168.2.180:93	http://192.168.2.180	Change Login	—
Element	false	○	None	Element	tcp://192.168.2.210:93	http://192.168.2.210	Change Login	—
Eth4Can	false	○	None	Eth4Can	tcp://192.168.2.28:93	http://192.168.2.28	Change Login	—

When lists expand beyond what can be viewed on a single page, page navigation buttons will appear:



Devices

Show 10 entries

Search:

Name	Online		Login State	Type	Address	Webpage		
FUSION-PS	false	○	LoginAccepted	PowerStation	tcp://192.168.2.60:93	http://192.168.2.60	Change Login	—
HP-ENVY	true	○	LoginAccepted	WinDriver	tcp://192.168.2.26:93	http://192.168.2.26	Change Login	—
ImagineLrcRouter	false	○	NotApplicable	ImagineLrcRouter	tcp://192.168.2.20:52116	http://192.168.2.20	Change Login	—
ipcom80712A	false	○	LoginAccepted	Intercom	tcp://192.168.2.35:93	http://192.168.2.35	Change Login	—

Showing 1 to 10 of 30 entries

Reference	Notes
1	Select an **entries value** from the dropdown list to define how many entries are shown on each page Options include 10, 25, 50, or 100 entries.
2	This field displays the total **number of items** shown on this page and the total **number of returned entries**.

3	If your search returns more entries than can be displayed on a single page, click the **navigation buttons** to move through the pages of returned results.

Important Note: These navigation tools are subject to the search field. In the case above for instance, if we were to type **FUSION** into the search field, we may get only a single returned entry and the page buttons would only show a single page.

Event System Catch 22

Pathfinder Core PRO includes an event system called Logic Flows. Because Logic Flows can reach into almost every part of the system, they present a bit of a Catch 22 when it comes to introducing certain subjects in a manual.

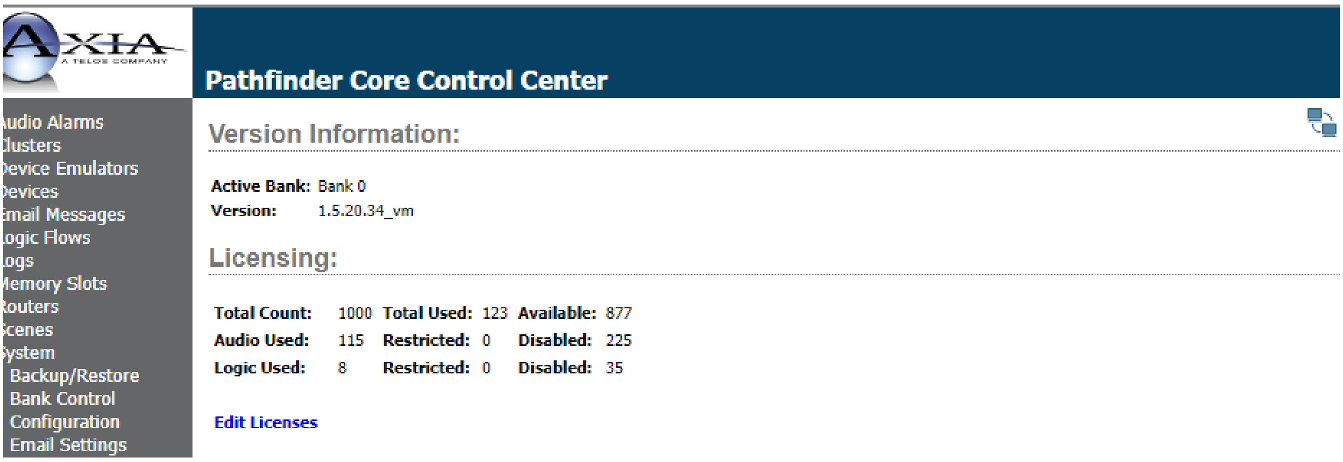
There are certain subjects - such as Memory Slots - that are primarily useful in the context of Logic Flows, so you have to understand Logic Flows in order to make the best use of Memory Slots. However, the examples and discussions within the Logic Flow section of this manual also will refer to Memory Slots in order to present some of its examples.

Wherever possible, we have attempt to note this and refer to the relevant sections of this manual. However, as a reader, be prepared to skip around if necessary. If the manual is read from front to back, there may be points where we refer to subjects covered in a later section.

System

Status

When you first log in to the Pathfinder Core PRO web page, you will be presented with the System Status page. This is the same web page displayed when clicking the **System** link in the navigation bar.

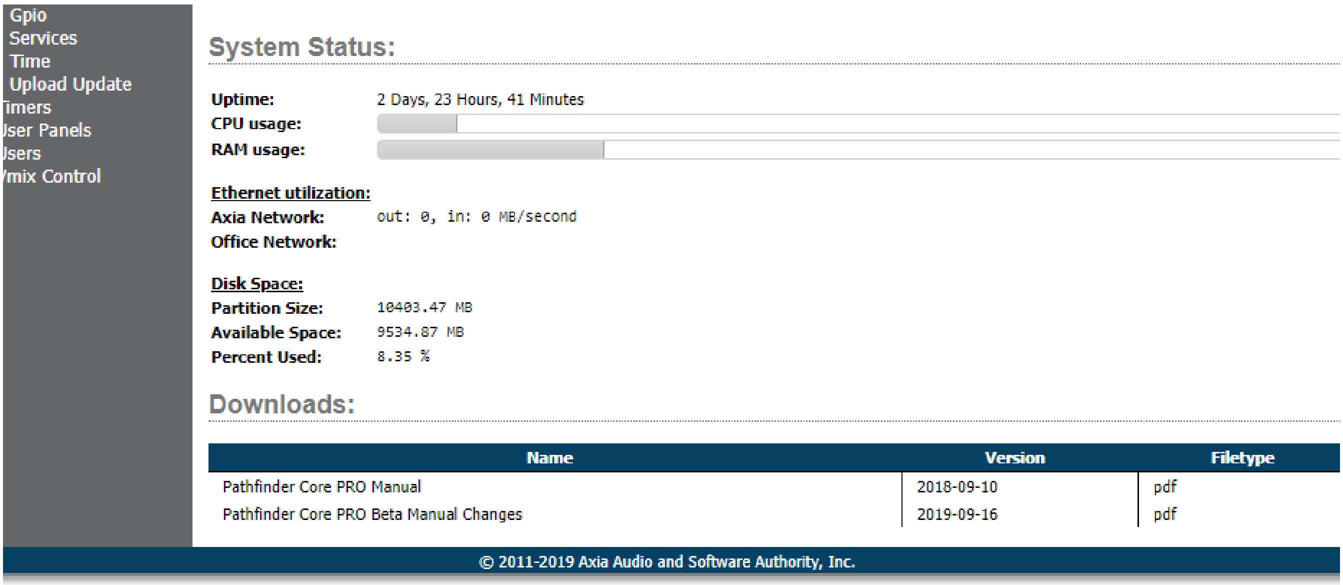


Version Information:

Active Bank: Bank 0
Version: 1.5.20.34_vm

Licensing:

Total Count:	1000	**Total Used:**	123	**Available:**	877
Audio Used:	115	**Restricted:**	0	**Disabled:**	225
Logic Used:	8	**Restricted:**	0	**Disabled:**	35



System Status:

Uptime: 2 Days, 23 Hours, 41 Minutes
CPU usage:
RAM usage:

Ethernet utilization:
Axia Network: out: 0, in: 0 MB/second
Office Network:

Disk Space:
Partition Size: 10403.47 MB
Available Space: 9534.87 MB
Percent Used: 8.35 %

Downloads:

Name	Version	Filetype
Pathfinder Core PRO Manual	2018-09-10	pdf
Pathfinder Core PRO Beta Manual Changes	2019-09-16	pdf

© 2011-2019 Axia Audio and Software Authority, Inc.

This screen is primarily informational outside of the licensing configuration fields. There are four sections:

Version Information

This shows which of the two Firmware Banks is currently active and the version of software running in that Bank.

Licensing

This shows the current license in the system and its capabilities.

System Status

This shows graphs representing the current CPU, memory, Ethernet utilization, and available disk space. The graphs are updated automatically every two seconds.

Downloads

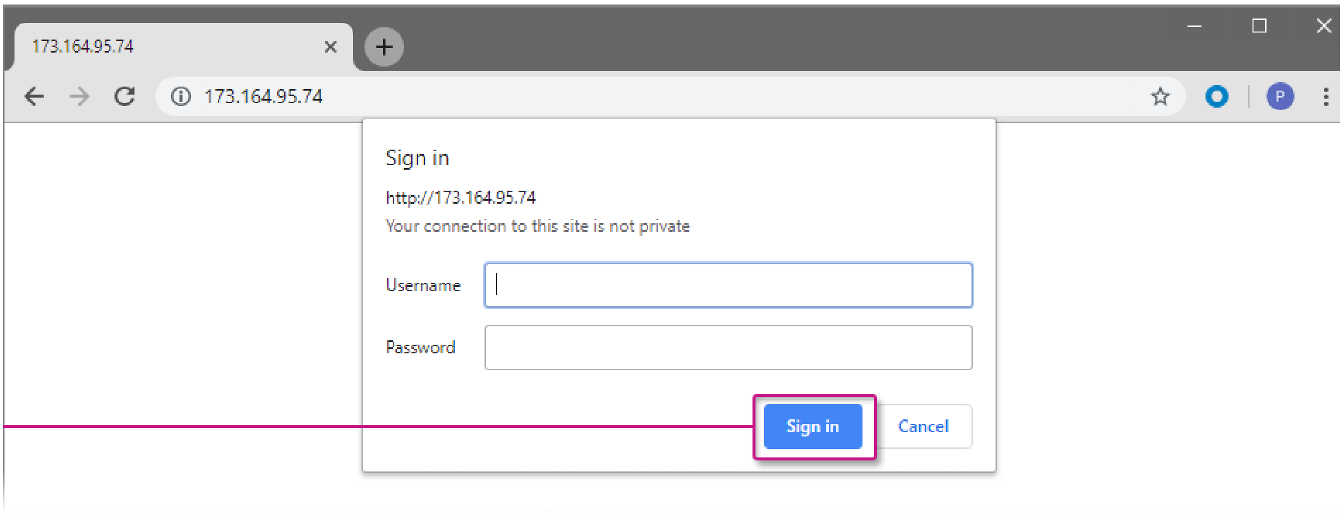
This section provides download links to the online documentation.

System Configuration

Under the System heading in the navigation bar is a **Configuration** link. If you've already referenced the Quick Start Guide, you will have already been introduced to this page.

- Access the web interface by typing in the IP address configured earlier in a web browser; an authentication window will appear; type the following values:
  Username = Admin
  Password = Admin

- Select **Sign In** to load the Pathfinder Core PRO control panel



173.164.95.74

Sign in

http://173.164.95.74

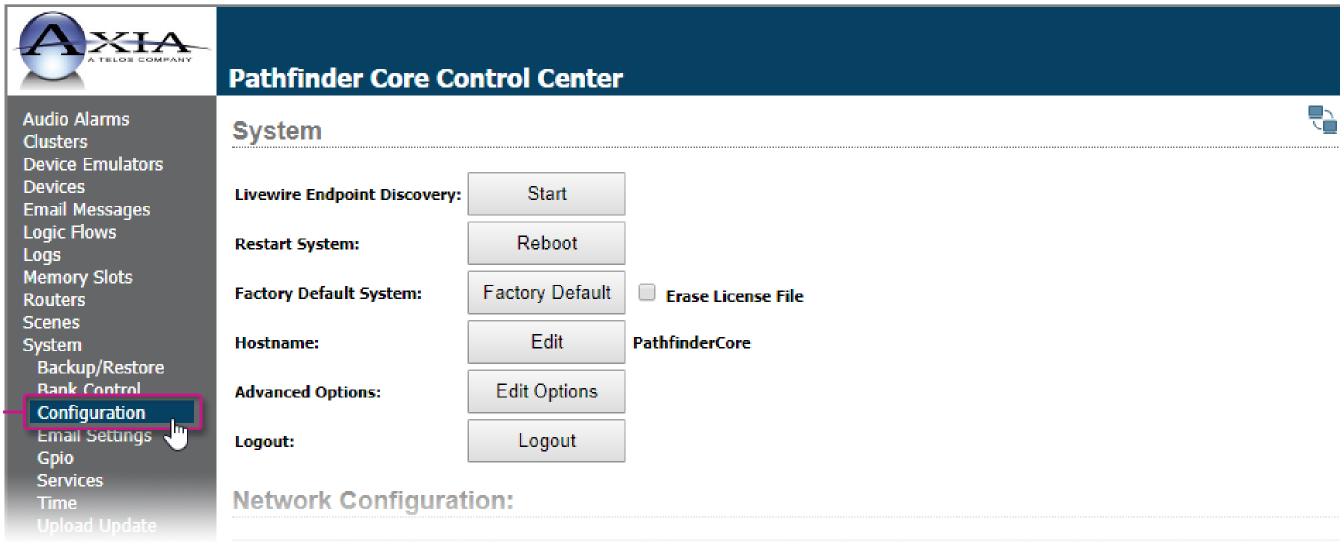
Your connection to this site is not private

Username

Password

Sign in Cancel

- From the links in the left-hand navigation bar, click **Configuration** under the System heading



Pathfinder Core Control Center

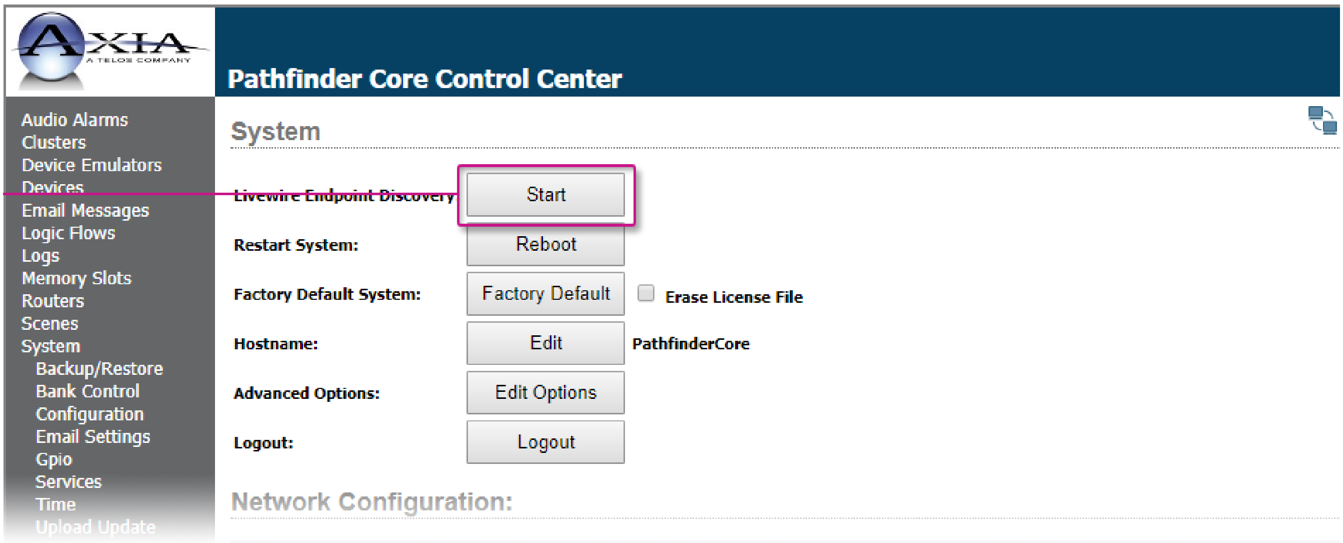
Audio Alarms
Clusters
Device Emulators
Devices
Email Messages
Logic Flows
Logs
Memory Slots
Routers
Scenes
System
Backup/Restore
Bank Control
Configuration
Email Settings
Gpio
Services
Time
Upload Update

System

Livewire Endpoint Discovery:	Start	
Restart System:	Reboot	
Factory Default System:	Factory Default	□ Erase License File
Hostname:	Edit	PathfinderCore
Advanced Options:	Edit Options	
Logout:	Logout	

Network Configuration:

- Next to *Livewire Endpoint Discovery*, click **Start**



Pathfinder Core Control Center

Audio Alarms
Clusters
Device Emulators
Devices
Email Messages
Logic Flows
Logs
Memory Slots
Routers
Scenes
System
Backup/Restore
Bank Control
Configuration
Email Settings
Gpio
Services
Time
Upload Update

System

Livewire Endpoint Discovery	Start	
Restart System:	Reboot	
Factory Default System:	Factory Default	□ Erase License File
Hostname:	Edit	PathfinderCore
Advanced Options:	Edit Options	
Logout:	Logout	

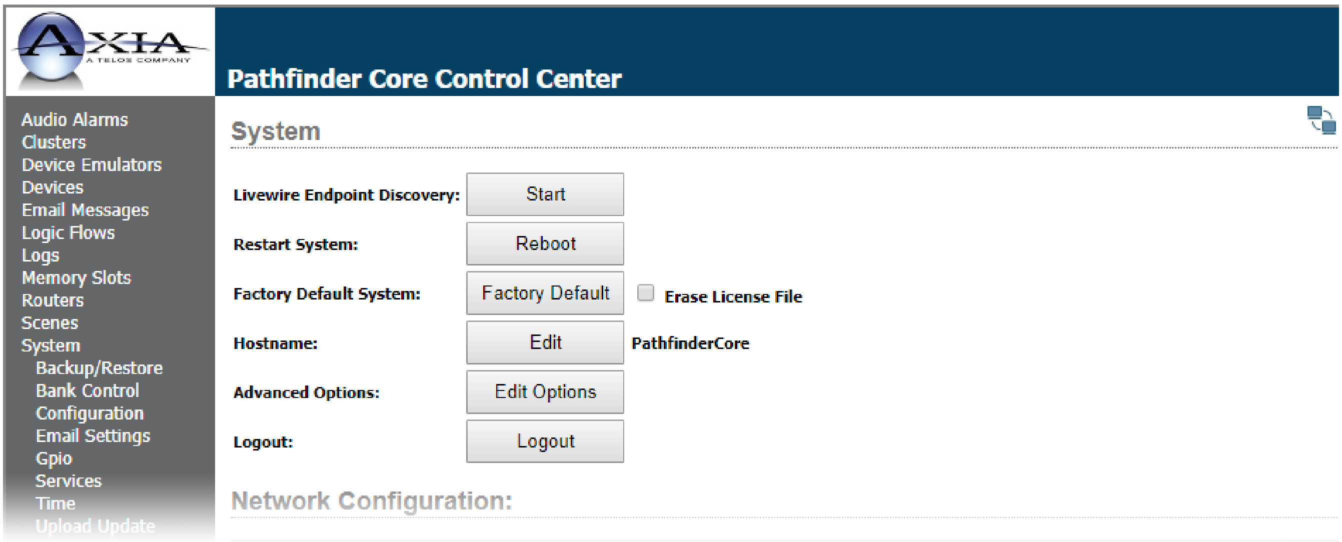
Network Configuration:

When a new device is discovered, the system will add it to its device list and add any route points provided by the device into the audio and GPIO router. This is one of the first options you will use when configuring a new system as it is how you will discover the equipment that Pathfinder Core PRO can control.

When discovery is enabled, the button will say *Stop* instead of *Start;* clicking the button will stop any additional automatic discovery.

> ⓘ Important Note: Pathfinder Core PRO uses the IP address of a device as the unique identifier of that device within the system. If you change the IP address of a device, Pathfinder Core PRO will treat it as a new device. This is why we recommend setting up your basic Axia infrastructure first before introducing Pathfinder Core PRO to the mix.

Additional System Buttons



System

Livewire Endpoint Discovery: Start

Restart System: Reboot

Factory Default System: Factory Default □ Erase License File

Hostname: Edit PathfinderCore

Advanced Options: Edit Options

Logout: Logout

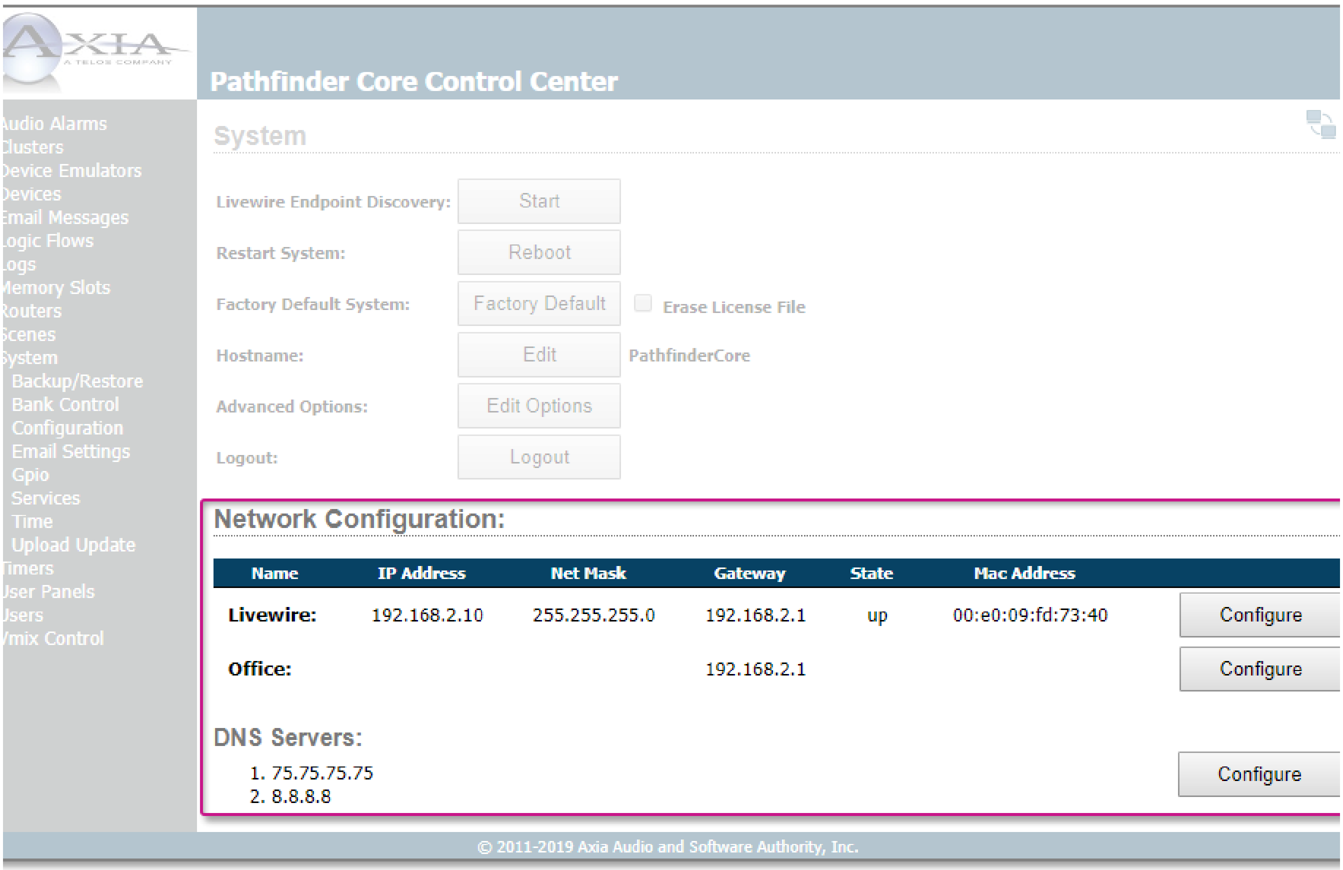
Network Configuration:

Button	Description
Restart System	If you click the **Reboot** button, the system will ask for confirmation to make sure the button was not accidentally clicked, and will then reboot the system.
Factory Default System	Clicking the **Factory Default** button will erase all existing configuration data and return the system t the way it was shipped from the factory. Multiple confirmations are required as **there is no undo option for this action**. It is highly recommended that you make a backup of your system and download it to your local computer before using th option. See the section of this manual on Backup and Restore for details. Select the **Erase License File** checkbox if you also want to erase the licensing. It is important to note that erasing the license file will also erase the base license, and a call to Axia support will be necessary to re-license

	the system. **Licenses are not included in the backup file.**
Hostname	Displays the current hostname of the Pathfinder Core PRO. Click **Edit** to assign a new hostname t the system. Hostname changes require a restart tc take effect.
Advanced Options	This section is for use in consultation with Axia support only as it provides methods for changing certain system variable and functionality.
Logout	Logs the current user out of the system and requests new login credentials.

Network Configuration Options

Current IP address information is visible in the **Network Configuration** section of the *System > Configuration* page.



Livewire Endpoint Discovery: Start

Restart System: Reboot

Factory Default System: Factory Default □ Erase License File

Hostname: Edit PathfinderCore

Advanced Options: Edit Options

Logout: Logout

Network Configuration:

Name	IP Address	Net Mask	Gateway	State	Mac Address	
Livewire:	192.168.2.10	255.255.255.0	192.168.2.1	up	00:e0:09:fd:73:40	Configure
Office:			192.168.2.1			Configure

DNS Servers:

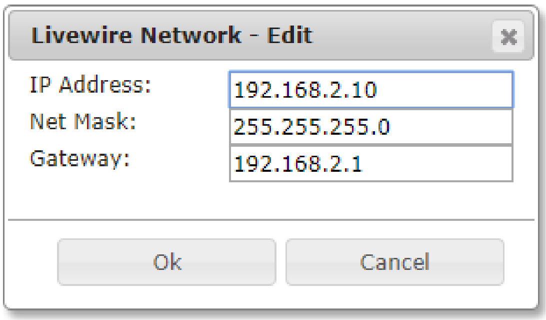
1. 75.75.75.75
2. 8.8.8.8

© 2011-2019 Axia Audio and Software Authority, Inc.

Livewire and Office Configuration

System IP options can be edited using the front panel display or through the web GUI.

- Click the **Configure** button to open the network interface's *Edit* screen



Livewire Network - Edit

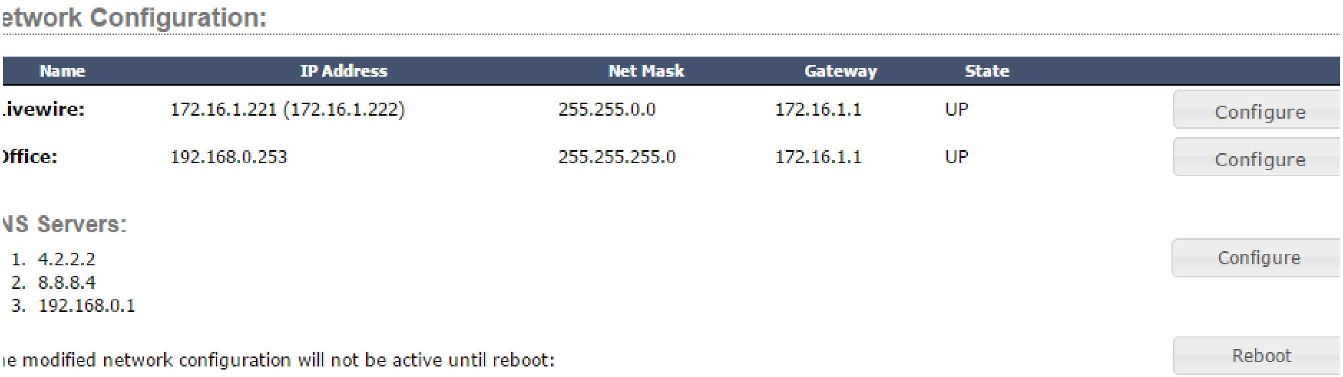
IP Address: 192.168.2.10

Net Mask: 255.255.255.0

Gateway: 192.168.2.1

- Type the **IP Address**, **Netmask**, and **Gateway** values and click **OK,** or click **Cancel** to leave this dialog without making any changes

> ⓘ Important Note: Network interface changes will not take effect until you reboot the system. Therefore, after changing the IP options, you will notice that *Network Configuration* screen will show the changes in parenthesis and will sport a shiny new *Reboot* button.



etwork Configuration:

Name	IP Address	Net Mask	Gateway	State	
.ivewire:	172.16.1.221 (172.16.1.222)	255.255.0.0	172.16.1.1	UP	Configure
)ffice:	192.168.0.253	255.255.255.0	172.16.1.1	UP	Configure

\NS Servers:

1. 4.2.2.2
2. 8.8.8.4
3. 192.168.0.1

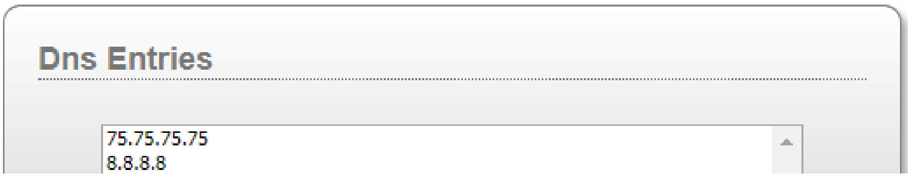
ıe modified network configuration will not be active until reboot:

DNS Servers Configuration

The DNS section lists the currently defined DNS servers.

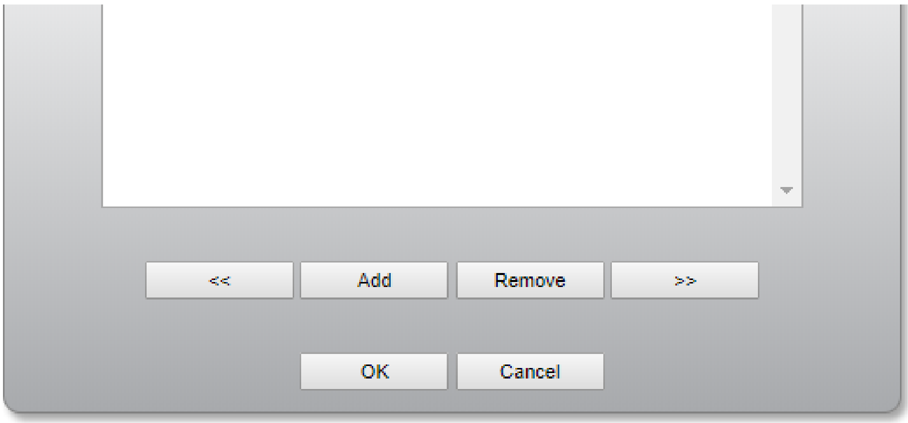
With correctly configured DNS settings, the system can send emails to administrators and employees alerting them to network changes requiring attention. For example, you can send emails when critical sources or destinations such as air chains become silent. To do this, the system needs to be able to resolve email server names with the correct IP addresses to send the email through. The DNS entries can also be used when entering NTP server information. Talk to your network administrator if you are unsure of which DNS servers to use.

- Click the DNS Server **Configure** button to open the **Dns Entries** dialog, allowing you to add, remove, and reorder DNS server entries



Dns Entries

75.75.75.75
8.8.8.8



- Click **Add** to add a new DNS server to the list and **Remove** to remove a DNS server entry
- To move a DNS Server up or down in the priority list, select a DNS server entry and click the **<<** or **>>** buttons
- Click **OK** to save your changes or **Cancel** to leave the dialog without making any changes

> ⓘ Important Note: DNS changes do not require a reboot to become active.

Licenses

Each Pathfinder Core PRO system includes a base license allowing a 1000 license point pool to be allocated between both Axia Audio Sources and Logic Flow endpoints.

> ⓘ Important Note: The licensing model changed with version 1.6 of Pathfinder Core PRO. In previous versions, the base license provided only 500 Axia Audio Sources and 500 Logic Flow endpoints, and add-on licenses would have to be applied to one or the other. Many customers found they needed more Axia Audio Source points but did not need as many Logic Flow points, resulting in the new licensing model defining the shared license point pool. We also added the new license types described below.

Virtual License Types

If you purchased a hardware-based Pathfinder Core PRO, the base license is pre-installed. If you purchased a virtual machine license, there are three possible license types:

	License Type	Points	Notes
1	Pathfinder Core PRO VM	1000	The equivalent functionality to the hardware license

2	Pathfinder Core PRO VML	300	A lower license count and a correspondingl lower cost of entry
3	Pathfinder Core PRO VM Backup	0	Ideal for situations when adding a VM-based cluster node to Pathfinder Core PRO cluster. This license will only work if clustered with anothe hardware or VM instance of Pathfinde Core PRO. In the cas of a failure of the primary licensed Pathfinder Core PRO the backup will continue to function a if it was the primary fo up to 90 days while you repair the primary instance. If this 90-da limitation is not acceptable, normal fu licenses may cluster a well without that restriction.

Additional Licenses

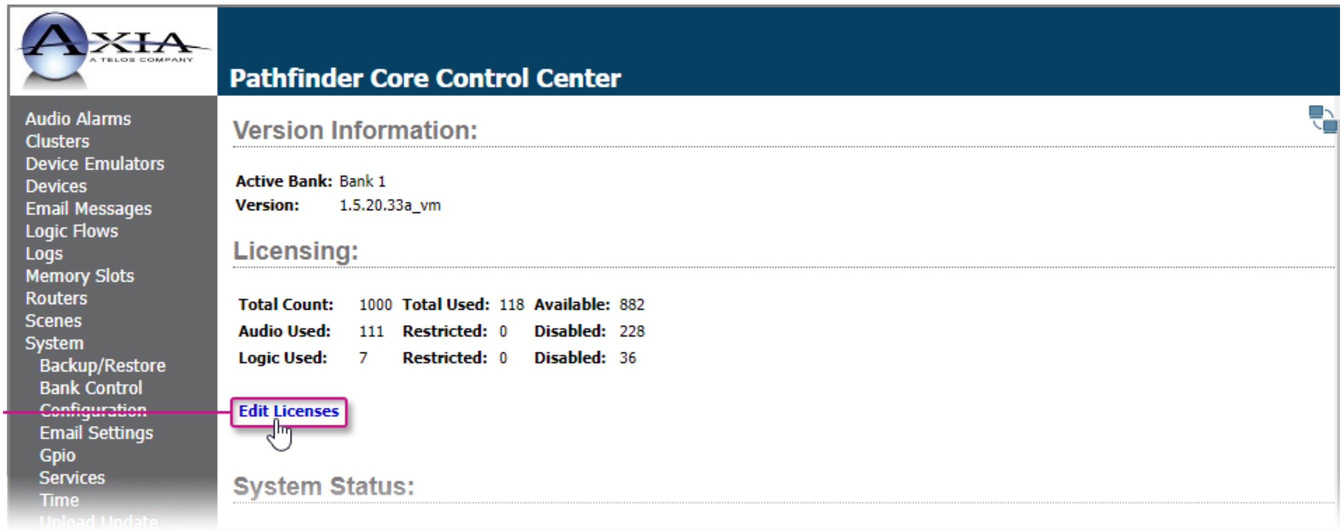
Additional licenses to add functionality may be purchased through your Axia distributer, with each additional licenses providing either an additional 500 (the *Pathfinder Core PRO Add-On 500*) or 100 (the *Pathfinder Core PRO Add-On 100*) license points.

Add-on licenses are shared between the Pathfinder Core PRO units that are participating in a clustered system. This means that if you need a redundant Pathfinder Core PRO system that supports 1400 license points, you will need to purchase *two* Pathfinder Core PRO systems (each of which comes with a base license or backup license) *and* a single add-on license to add the additional 500 sources. The add-on license allows both systems in the cluster to mirror and synchronize the full 1400 license point functionality.

> ⓘ Important Note: Virtual sources and GPIO sources do not count against the license. Only Axia Audio Sources with enabled streams and Logic Flow endpoints are counted in the licensing. Additionally, Logic Flows which are dynamically created by the system during HTML5 panel binding and/or hardware mapping do not count against your licensing. If you have questions regarding the licensing model, please contact Axia support or your Axia distributor.

-

- To view or add additional licenses, click the **System** heading on the navigation bar and then click the **Edit Licenses** link



Version Information:

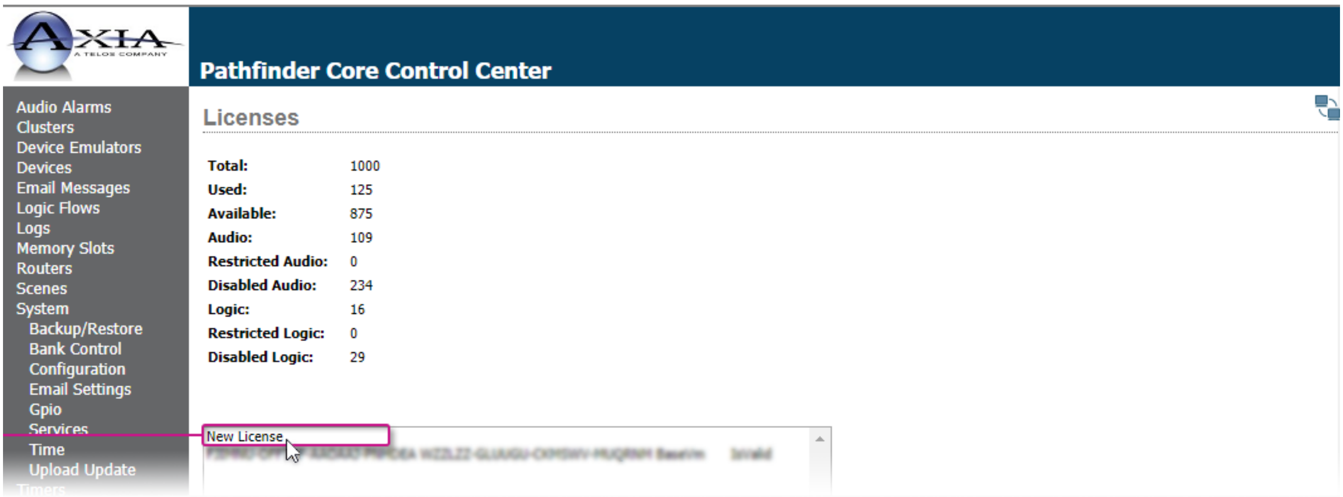
Active Bank: Bank 1
Version: 1.5.20.33a_vm

Licensing:

Total Count:	1000	**Total Used:**	118	**Available:**	882
Audio Used:	111	**Restricted:**	0	**Disabled:**	228
Logic Used:	7	**Restricted:**	0	**Disabled:**	36

System Status:

- The licensing page will display your base license information; it will also provide a list of your additional licenses
- Click the **New License** line then type the new **Request Code** and **License Key** to add a new license



Licenses

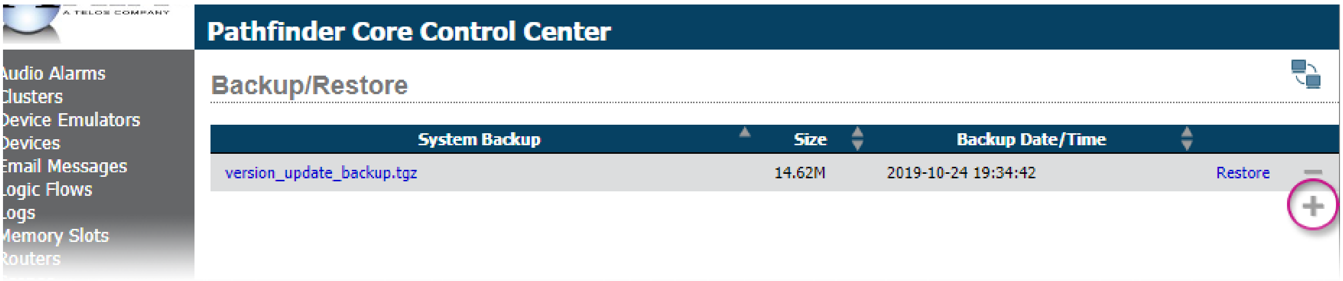
Total:	1000
Used:	125
Available:	875
Audio:	109
Restricted Audio:	0
Disabled Audio:	234
Logic:	16
Restricted Logic:	0
Disabled Logic:	29

- Finally, click **Submit Changes** to add the license to the system; license changes may require a reboot to become effective; click the **Configuration** link under System to find the reboot button

Backup/Restore

The **Backup/Restore** page presents a list of all backups in the system. If this is the first time using this page, there will probably not be any backups present in the system.

To take a new backup, or to upload a backup from your local computer back into the system, click the **plus icon**.



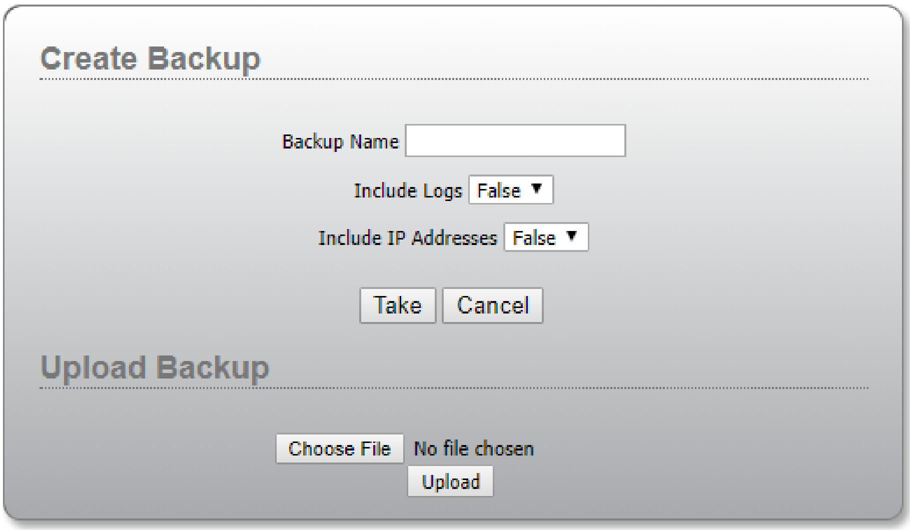
Pathfinder Core Control Center

Backup/Restore

System Backup	Size	Backup Date/Time	
version_update_backup.tgz	14.62M	2019-10-24 19:34:42	Restore

Taking Backups

To create a new backup, type a **name** for the backup, select whether you want logs and/or the IP address to be included in the backup, and then click **Take**. Click **Cancel** to exit the dialog without making any changes.



Create Backup

Backup Name

Include Logs False

Include IP Addresses False

Upload Backup

The backup will include any configuration files needed to return your system to its current state.

If logs are included, the backup will also include all system and application logs currently on the system. The log information can be very useful to Axia support if they are trying to assist in troubleshooting an issue, and it is likely they will ask you to take a backup including log files and send it to them.

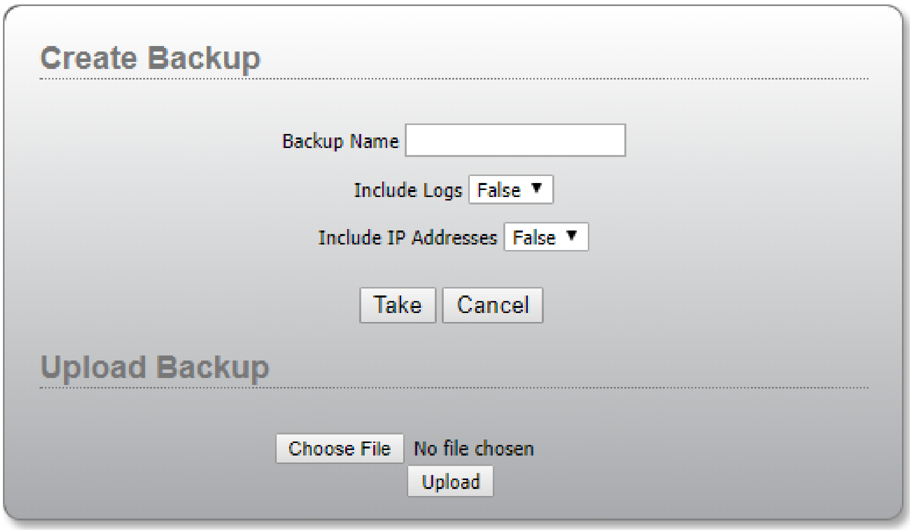
If you intend to restore the backup on a different Pathfinder Core PRO, you may not wish to include IP addresses so the restore process does not overwrite the IP addresses of the other unit.

> ⓘ Important Note: After taking a new backup it is highly recommended that you click on the link to the backup file and download it to your local computer to store it in a safe location. Backups that live on the Pathfinder Core PRO system are not true backups because if the system or storage medium were to fail, both the system and the backups could be lost. Downloading a backup to your local system will also allow you to send the backup to support if necessary.

Backups are stored in the software Bank. When you move to a new software Bank, you will not be able to directly access the backups from the previous Bank. However, if the backup is saved to your local computer, you can easily upload it back into a new Bank if desired.

Uploading Backups

To restore your configuration to a new system, a factory default system, or a system that has had the desired backup removed from the system, click **Choose File**.



Create Backup

Backup Name

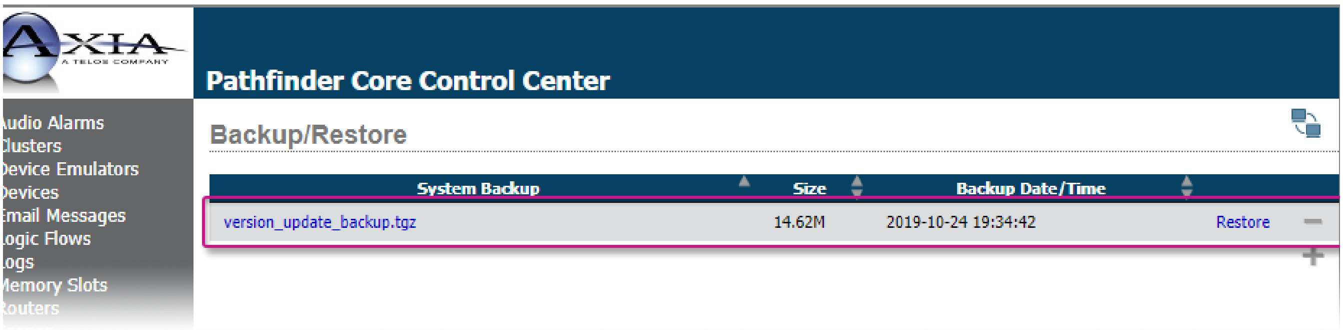
Include Logs: False

Include IP Addresses: False

Upload Backup

No file chosen

Select a **System Backup** file on your local system, then click **Upload**. After the upload is complete, the backup will be available in the Backup/Restore list.



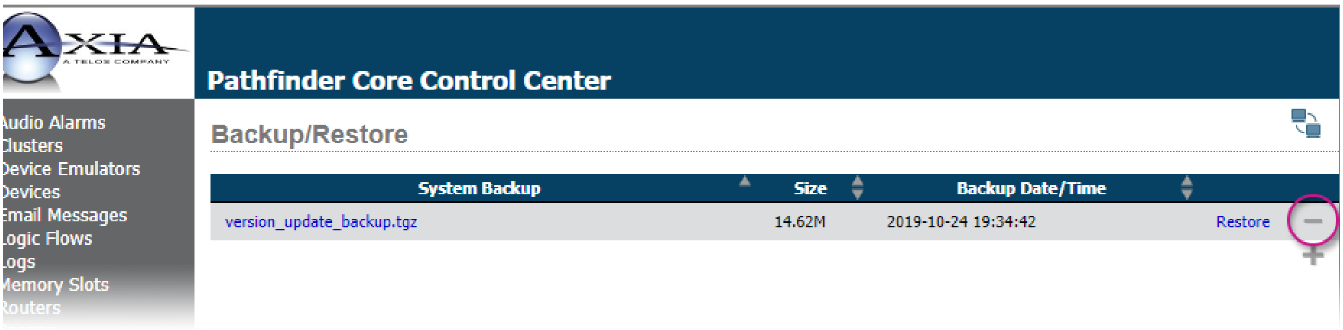
Pathfinder Core Control Center

Backup/Restore

System Backup	Size	Backup Date/Time	
version_update_backup.tgz	14.62M	2019-10-24 19:34:42	Restore

Removing Backups

The **minus** icon will delete a backup from the system.



Pathfinder Core Control Center

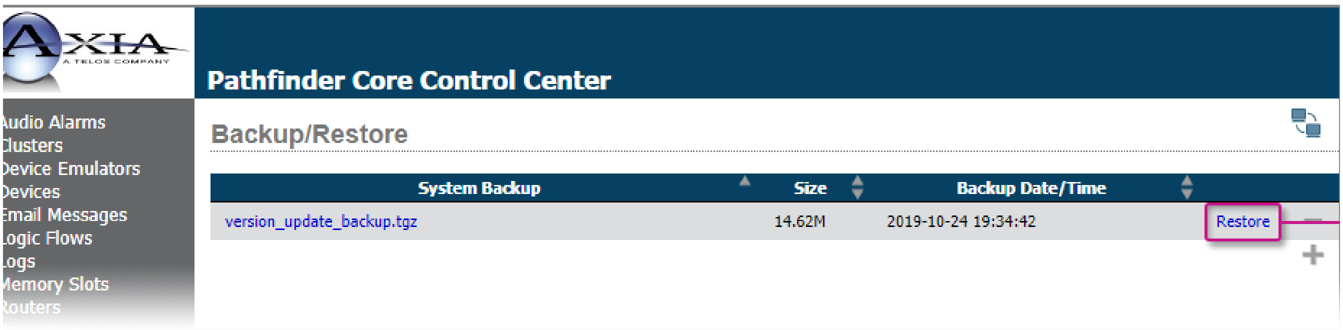
Backup/Restore

System Backup	Size	Backup Date/Time	
version_update_backup.tgz	14.62M	2019-10-24 19:34:42	Restore

Since there is limited space on the storage medium it is recommended to keep only a few backups on the system at a time.

Restoring Backups

Click the **Restore** link next to a stored backup. The system will request confirmation that you really wish to restore the system to that state



Pathfinder Core Control Center

Audio Alarms
Clusters
Device Emulators
Devices
Email Messages
Logic Flows
Logs
Memory Slots
Routers

Backup/Restore

System Backup	Size	Backup Date/Time	
version_update_backup.tgz	14.62M	2019-10-24 19:34:42	Restore

After confirming the operation, the system will shut down active services, restore the configuration files to the state of that backup, and reboot.

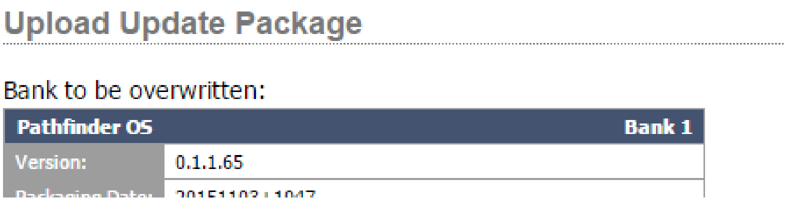
> ⓘ Important Note: The restore function typically happens quickly, but keep in mind that the system will not be operational during the restore and reboot process.

Upload Update

One of the Pathfinder Core PRO Administrator's most important tasks is periodic firmware updates. This project is undergoing a very active development lifecycle with many new features planned for future versions. That, in addition to fixes for any bugs reported to us by our users, will make occasional updates of the firmware in your Pathfinder Core PRO system necessary.

The system has two software Banks so that if there is a need to return to an earlier version, it will live in the preceding Bank. To upgrade your system:

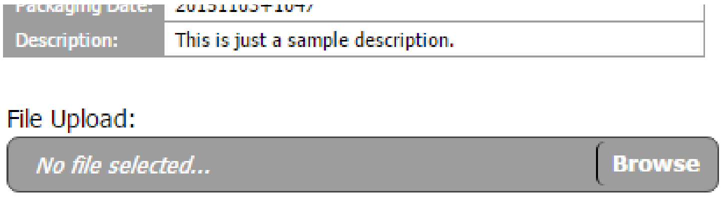
- Take a backup of your system as explained in the **Taking Backups** section above; we always recommend backing up your system before making any major change just to be safe; the backup process generates its own backup as well, but it never hurts to have another copy stored on your local computer
- Download the new firmware version from the Axia Audio web site to your local computer; the file name extension will be **.pfc_upd** which stands for Pathfinder Core PRO update package; it is also a good idea to take a moment to scan the release notes to understand the changes between the version you are currently running and the version to which you will be upgrading
- Select the **Upload Update** link under the System heading in the navigation bar; this page will display the Bank on which you are not currently running, as the update will always load to the inactive bank
- Click **Browse** and select the **file** you downloaded from the Axia web site



Upload Update Package

Bank to be overwritten:

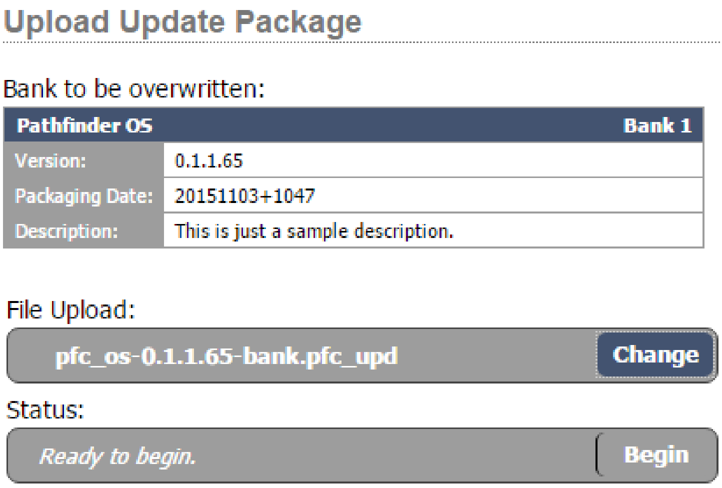
Pathfinder OS	Bank 1
Version:	0.1.1.65
Packaging Date:	20151103	1047



Packaging Date:	20151103+1047
Description:	This is just a sample description.

File Upload:

- The File Upload bar will change to the name of the file which you have selected to upload and a status bar will appear with the **Begin** button; if the file is not correct for the type of system you have (for example, you attempt to install VM software to a Fanless Engine platform) a warning will appear
- Click **Begin** to begin the updating process



Upload Update Package

Bank to be overwritten:

Pathfinder OS	Bank 1
Version:	0.1.1.65
Packaging Date:	20151103+1047
Description:	This is just a sample description.

File Upload: pfc_os-0.1.1.65-bank.pfc_upd

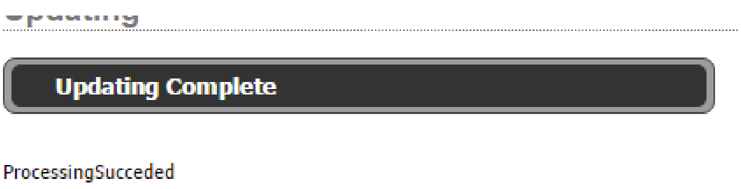
Status: Ready to begin.

- A new page will appear with a progress bar that presents stages along with some additional information regarding what is taking place at that point of the update process; the update may take several minutes to complete

> ⓘ Important Note: It is important to understand the status of the configuration files during an upgrade process. The configuration files for the system reside within the currently executing Bank. During the upgrade process, a backup of the configuration from the currently running Bank is automatically made. After the new software is written into the new Bank, the configuration is then restored into the new Bank so that when you boot into the new Bank it has the same configuration that is in the currently executing Bank. However, if you then boot into the new Bank and make changes to the configuration in the new Bank, those changes will not reside in the old Bank. If you wish to return to the old Bank and you have made changes to the configuration, you should back up the configuration on the new Bank, boot into the old Bank, and then restore the backup. This process is outlined in the section on Backup and Restore.

- Once the update is complete, you should receive a *Processing Succeeded* message; because you are updating the Bank which is currently active, the system should continue to run even if there was an error with the update

Updating



Updating Complete

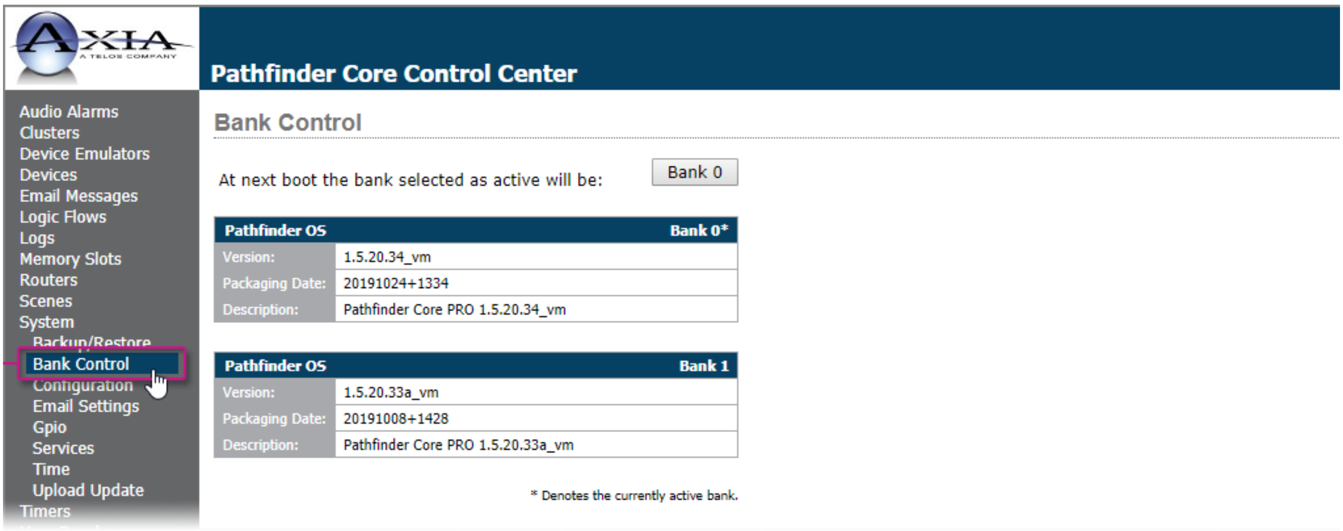
ProcessingSucceded

- If you receive an error message instead of Processing Succeeded, please contact Axia support
- To boot into the updated Bank and start using the new software, click the **Bank Control** link in the navigation bar

---

Bank Control

Bank Control displays the software versions in both Banks of the system and offers control over the currently executing software revision Bank. Click the **Bank Control** link under the System heading in the navigation bar to view this page.



Pathfinder Core Control Center

Bank Control

At next boot the bank selected as active will be: Bank 0

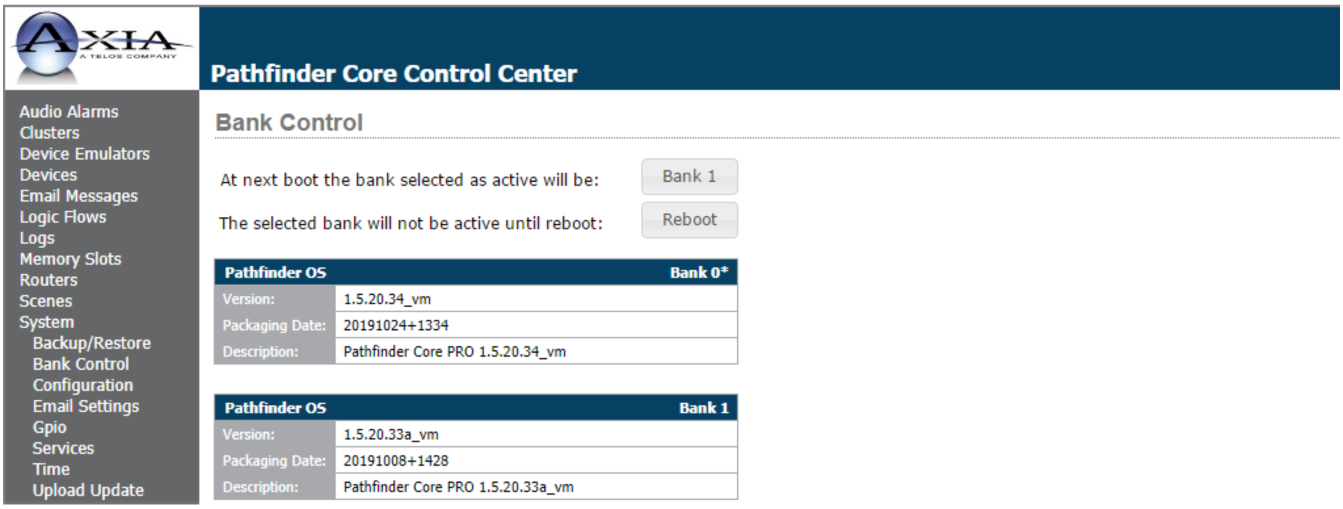
Pathfinder OS	Bank 0*
Version:	1.5.20.34_vm
Packaging Date:	20191024+1334
Description:	Pathfinder Core PRO 1.5.20.34_vm

Pathfinder OS	Bank 1
Version:	1.5.20.33a_vm
Packaging Date:	20191008+1428
Description:	Pathfinder Core PRO 1.5.20.33a_vm

* Denotes the currently active bank.

The **Bank** button can be used to select which Bank will be used on the next reboot. It also displays the currently selected Bank as the next Bank that will be active at reboot.

After changing the Bank that will be used on the next reboot, a **Reboot** button will appear on the page.



Pathfinder Core Control Center

Bank Control

At next boot the bank selected as active will be: Bank 1

The selected bank will not be active until reboot: Reboot

Pathfinder OS	Bank 0*
Version:	1.5.20.34_vm
Packaging Date:	20191024+1334
Description:	Pathfinder Core PRO 1.5.20.34_vm

Pathfinder OS	Bank 1
Version:	1.5.20.33a_vm
Packaging Date:	20191008+1428
Description:	Pathfinder Core PRO 1.5.20.33a_vm

Clicking **Reboot** will reboot into the newly selected Bank.

The information tables display the software version in each Bank. An asterisk will be present next to the Bank that is currently executing.

> ⓘ Important Note: When you select a new Bank, the system creates a temporary mark in the system to boot into that Bank. After the system boots into the new Bank, that temporary mark is turned into a normal reboot setting so the newly selected Bank will be used on subsequent reboots. This is called "cementing the Bank". If something goes wrong with the upgrade and the Bank is unusable, the Bank does not get cemented and rebooting the system either automatically or by disconnecting power will cause the system to boot into the previously working Bank. The new Bank will only get cemented for future boots if it boots successfully.

This also means that after rebooting into a new Bank you should return to the Bank control web page to make sure the boot was successful and you are executing on the desired Bank. If the boot fails, it could fail back to the previous Bank and you will only know that by double-checking this page.

> ⓘ Important Note: We have occasionally seen some browsers try to cache the state of this page even though we have requested that it does not do so in the web page code. If you reboot and the executing asterisk does not change, try refreshing the web page. Some browsers have an extra button you can hold down while refreshing to force the web page to refresh rather than rely on the cache; for example, with Google Chrome, hold Shift+CTRL while clicking the refresh icon.

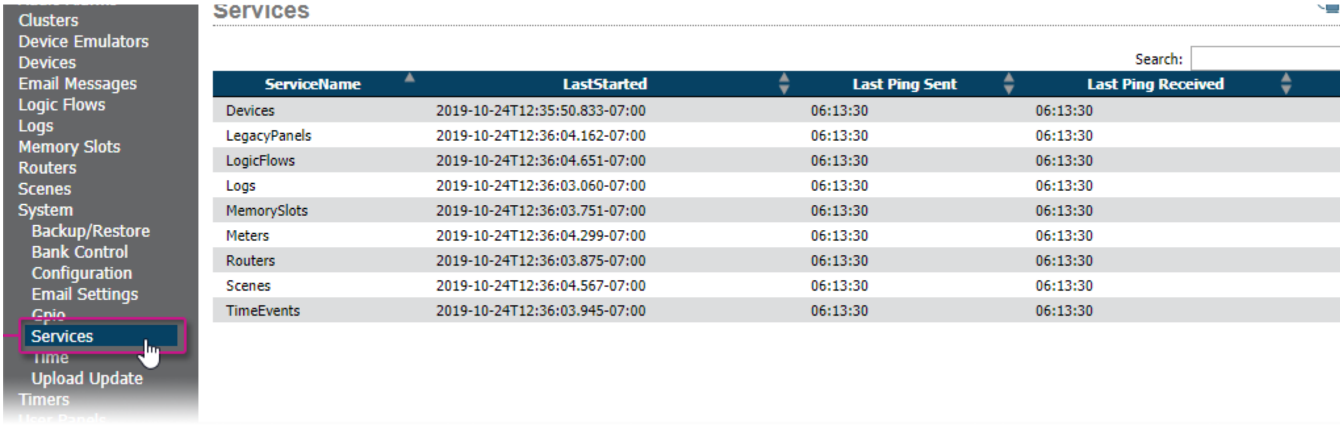
Configuration File Between Bank Changes

Since configuration files reside within the executing Bank, if you are switching to a different Bank the configuration may be different. It is always recommended to take a backup of your configuration before switching Banks, and then restore that configuration on the new Bank.

If you are switching Banks immediately after upgrading the firmware, then this backup and restore have been already completed as part of the upgrade process. Otherwise, it is recommended to review the backup and restore procedures earlier in this manual. Conversely, this also means if you make a mistake in the configuration in a new Bank, you can boot back into the old Bank to get to an older and successfully working configuration, though using backups and restores is a more efficient way of doing this.

Services

Click the **Services** link on the navigation bar to view the Services page.



ServiceName	LastStarted	Last Ping Sent	Last Ping Received
Devices	2019-10-24T12:35:50.833-07:00	06:13:30	06:13:30
LegacyPanels	2019-10-24T12:36:04.162-07:00	06:13:30	06:13:30
LogicFlows	2019-10-24T12:36:04.651-07:00	06:13:30	06:13:30
Logs	2019-10-24T12:36:03.060-07:00	06:13:30	06:13:30
MemorySlots	2019-10-24T12:36:03.751-07:00	06:13:30	06:13:30
Meters	2019-10-24T12:36:04.299-07:00	06:13:30	06:13:30
Routers	2019-10-24T12:36:03.875-07:00	06:13:30	06:13:30
Scenes	2019-10-24T12:36:04.567-07:00	06:13:30	06:13:30
TimeEvents	2019-10-24T12:36:03.945-07:00	06:13:30	06:13:30

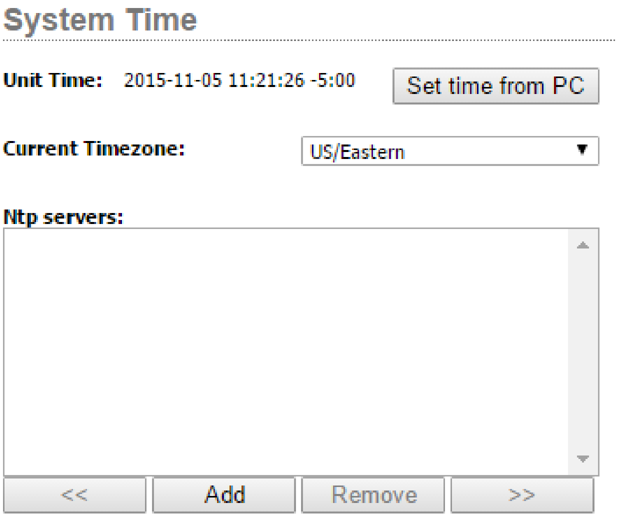
This page will only be used in conjunction with Axia tech support, but it is worth discussing what this page represents. To make Pathfinder Core PRO as robust as possible, the system has a sophisticated watchdog process. This process monitors the state of each of the services in the system and is responsible for sending ping messages to each service to make sure they are still responsive at the application layer. If a service fails to respond within a certain period of time, the watchdog will restart the service. In a rare and catastrophic situation where the watchdog is not able to achieve proper responsiveness of a service, it might also restart all services or even reboot the system after enough subsequent failures.

This screen shows each service, when it was last started, and the most recent ping and response time. If you watch this screen you should notice the pings changing at approximately 5-second intervals.

In many ways, this page is equivalent to the *Services* control panel in a Windows machine.

Time

Click on the **Time** link under the System heading of the navigation bar to view the Time settings. The System Time configuration page allows you to define NTP servers, set the system's time zone, and sync the current time to that of your local PC.



Unit Time: 2015-11-05 11:21:26 -5:00

Current Timezone: US/Eastern

Use the **Current Timezone** drop-down to select the correct time zone for the system.

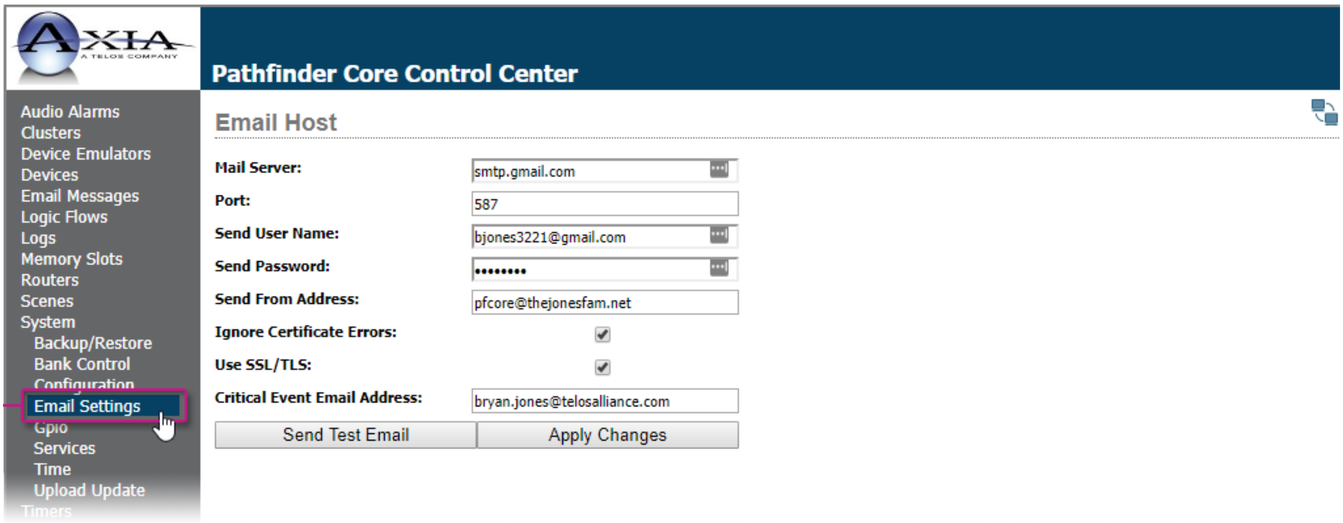
Defining NTP servers in the **Ntp servers** list is recommended so that Pathfinder Core PRO can update its time automatically and always be as accurate as possible.

There are a number of reasons why it is important to keep Pathfinder Core PRO's time accurate. The first is that the system has the ability to define events that happen at specific dates and times. Those events will, of course, fire at incorrect times if the system's time is not up to date. Additionally, log files use the system's time settings to keep track of when changes happen.

Finally, proper cluster synchronization relies on date and time settings to determine whether a specific piece of information is more up-to-date on one system or another and therefore whether synchronization of that piece of information needs to take place. Talk to your network administrator if you have questions about which NTP servers to use.

Email Settings

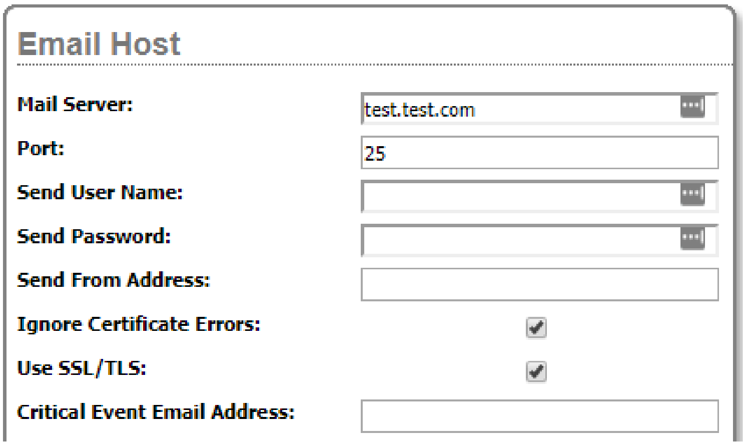
In order to send emails from Logic Flows, the server parameters must first be defined in the system in order to let Pathfinder Core PRO know which email server to use to send email messages. Under the System section of the navigation bar, click the **Email Settings** link.



Email Host

Mail Server:	smtp.gmail.com
Port:	587
Send User Name:	bjones3221@gmail.com
Send Password:	••••••••
Send From Address:	pfcore@thejonesfam.net
Ignore Certificate Errors:	☑
Use SSL/TLS:	☑
Critical Event Email Address:	bryan.jones@telosalliance.com

These settings are like those used by any other email client application.



Email Host

Mail Server:	test.test.com
Port:	25
Send User Name:	
Send Password:	
Send From Address:	
Ignore Certificate Errors:	☑
Use SSL/TLS:	☑
Critical Event Email Address:	

Field	Notes
Mail Server	Type the name of your **mail server**.
Port	Type the port number required by your mail serve Ports 25 and 587 are commonly used by SMTP mail servers. Check with your administrator or you hosting service provider to determine the correct Port setting.
Send User Name	Type the **user name** associated with your accoun on your mail server. *See note below.*
Send Password	Type the **password** associated with your account on your mail server. *See Important Note below.*
Send From Address	Type the **email address** that will be included in th outgoing message’s *Send From* field.
Ignore Certificate Errors	If you are using SSL encryption and the SSL/TLS certificate chain provided by the host is not in Pathfinder Core PRO’s trusted list, the message will not be sent and an error will be generated. Checking this **option** allows the system to use SS encryption but ignore the SSL certificate error.
Use SSL/TSL	Check this **option** to use SSL encryption.
Critical Event Email Address	Type the **email address** of the administrator who should receive all emails generated by critical events such as a system restart or a service failur

> ⓘ Important Note: While most servers require authentication, some provide security based on the source IP address rather than user credentials. Using a blank username and password will cause Pathfinder Core PRO to skip including any credentials in the email sends. After applying a blank password, the field will most likely show a series of stars, but this does not indicate the presence of a password. In addition to masking the characters of a password, this field also masks the length of the password for security reasons. Therefore, even a blank password will display a constant number of stars in the field after the password has been applied.

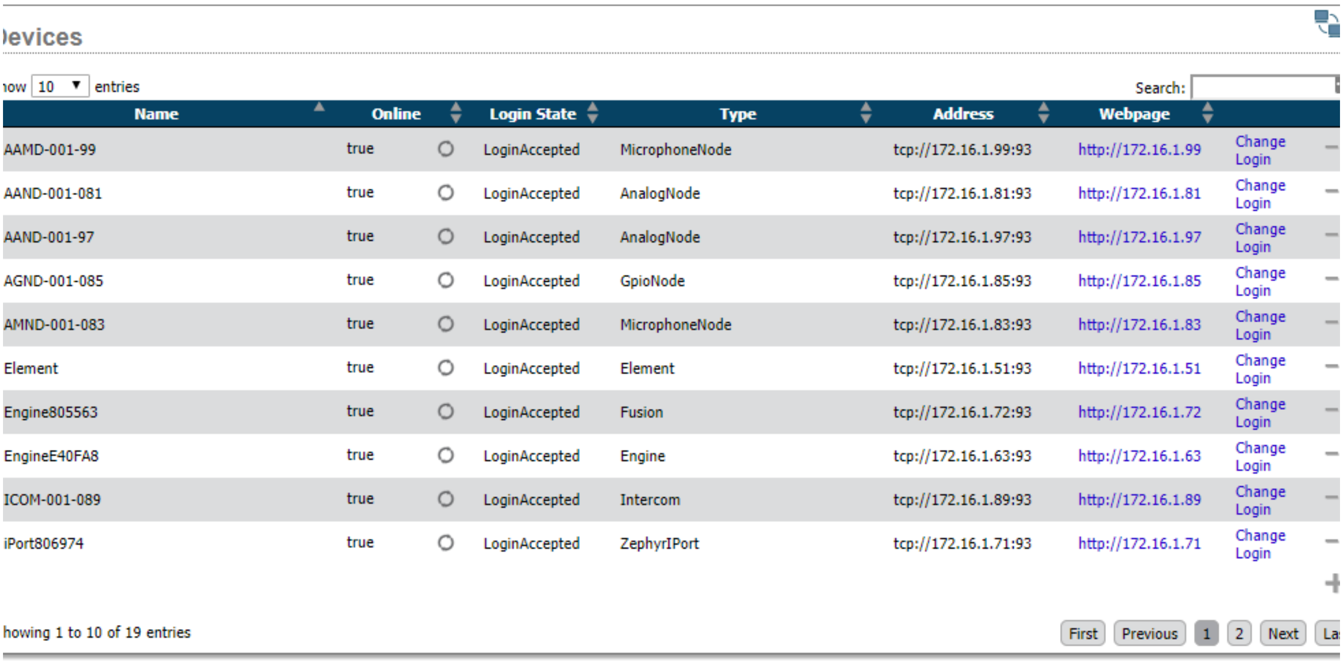
Once you have the parameters configured correctly, click **Apply Changes** to store them to the system. This button must be clicked before *Send Test Email* will use the new settings.

Use the **Send Test Email** button to generate a test email. The page will ask for a destination email address and will then attempt to send the message. Any errors will be reported. Once you can successfully send test emails, you are ready to create messages that may be used by Pathfinder Core PRO’s event system, Logic Flows

Go to the Email Messages section of this manual for more details about creating email messages determine when those messages are sent.

Devices

The Devices link in the navigation bar displays a list of all devices discovered by Pathfinder Core PRO.



evices

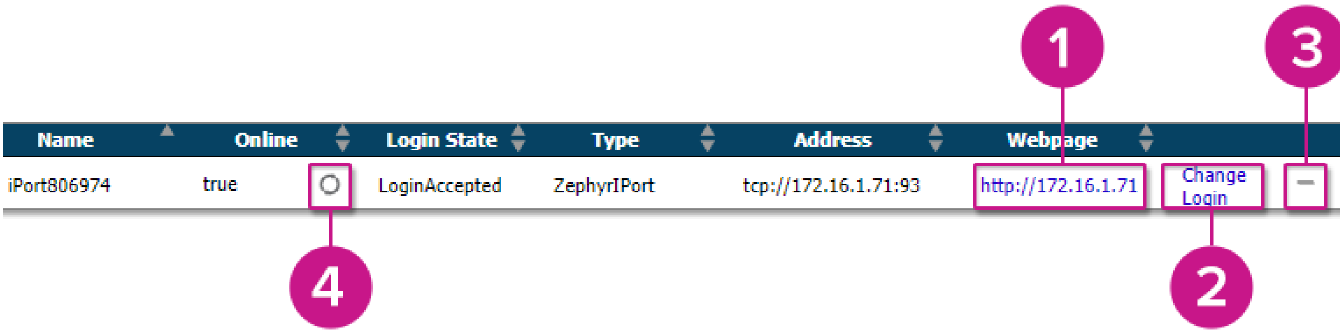
how 10 ▾ entries Search:

Name	Online	Login State	Type	Address	Webpage		
AAMD-001-99	true ○	LoginAccepted	MicrophoneNode	tcp://172.16.1.99:93	http://172.16.1.99	Change Login	−
AAND-001-081	true ○	LoginAccepted	AnalogNode	tcp://172.16.1.81:93	http://172.16.1.81	Change Login	−
AAND-001-97	true ○	LoginAccepted	AnalogNode	tcp://172.16.1.97:93	http://172.16.1.97	Change Login	−
AGND-001-085	true ○	LoginAccepted	GpioNode	tcp://172.16.1.85:93	http://172.16.1.85	Change Login	−
AMND-001-083	true ○	LoginAccepted	MicrophoneNode	tcp://172.16.1.83:93	http://172.16.1.83	Change Login	−
Element	true ○	LoginAccepted	Element	tcp://172.16.1.51:93	http://172.16.1.51	Change Login	−
Engine805563	true ○	LoginAccepted	Fusion	tcp://172.16.1.72:93	http://172.16.1.72	Change Login	−
EngineE40FA8	true ○	LoginAccepted	Engine	tcp://172.16.1.63:93	http://172.16.1.63	Change Login	−
ICOM-001-089	true ○	LoginAccepted	Intercom	tcp://172.16.1.89:93	http://172.16.1.89	Change Login	−
iPort806974	true ○	LoginAccepted	ZephyrIPort	tcp://172.16.1.71:93	http://172.16.1.71	Change Login	−

+

howing 1 to 10 of 19 entries

Each device entry in the list includes the following options:



Name	Online	Login State	Type	Address	Webpage		
iPort806974	true ○	LoginAccepted	ZephyrIPort	tcp://172.16.1.71:93	http://172.16.1.71	Change Login	−

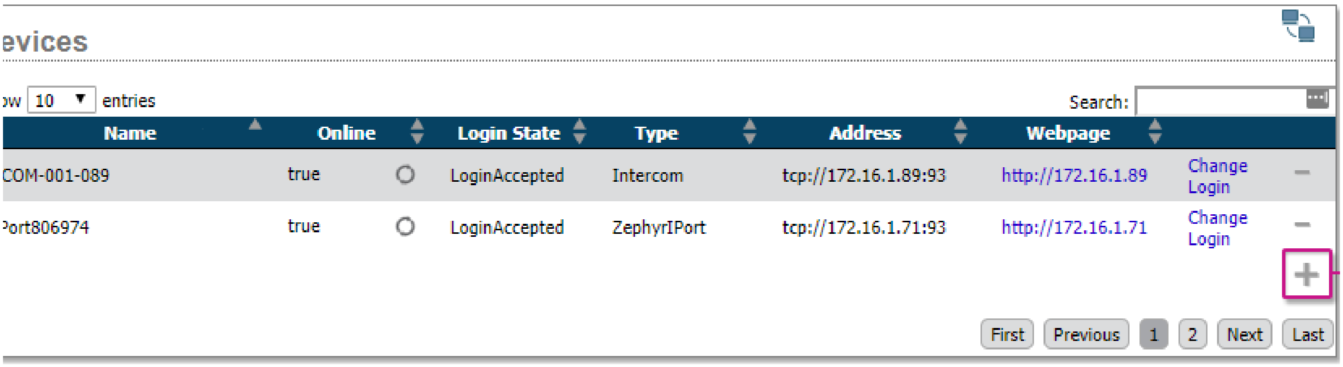
Reference	Description
1	Clicking this **link** takes you directly to the device’s configuration web page.
2	If a different login is required by the device click th **link** to set the password.
3	Clicking this icon will remove this device from the system. *See notes below for details*.

4	The connection recycle icon should rarely, if ever, be used. Clicking this **icon** will cause Pathfinder t drop and reconnect its connection to the device.

ⓘ Important Note: When a device is removed from the system, it will also remove all entries in the Audio and GPIO router for that device. It is important to note that if Livewire Discovery is enabled on the system page, the device may get re-discovered and added back into the list if it is still on the network.

Manually Adding Devices

Some devices cannot be discovered automatically. These include Axia OLED and LCD rackmount button panels. Additionally, you may want to add a device into the system without enabling the full network discovery engine. To add these into the system, click the **Add** button on the **Devices** dialog.



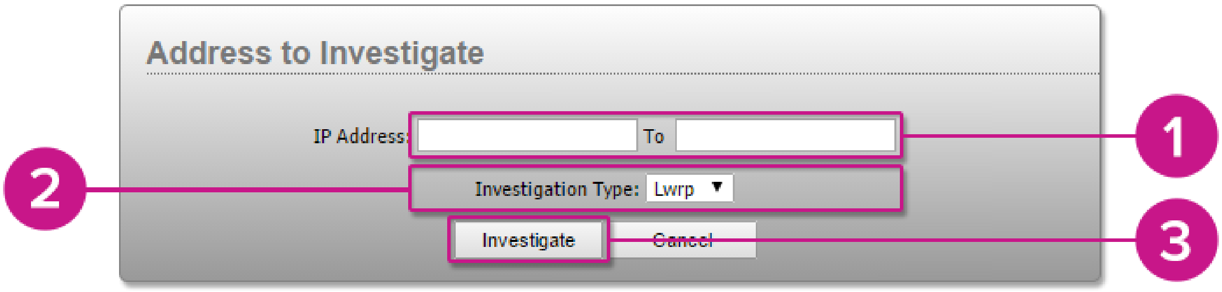
evices

ow 10 entries

Search:

Name	Online		Login State	Type	Address	Webpage		
COM-001-089	true	○	LoginAccepted	Intercom	tcp://172.16.1.89:93	http://172.16.1.89	Change Login	−
Port806974	true	○	LoginAccepted	ZephyrIPort	tcp://172.16.1.71:93	http://172.16.1.71	Change Login	−

A dialog box will appear that allows you to define a range of IP addresses to scan and the investigation method to use.



Address to Investigate

IP Address: To

Investigation Type: Lwrp

Reference	Description
1	If you are adding a single device, type the **same I address** in both boxes. The *To* box should auto-populate when you type an **IP address** into the *IF Address* box.

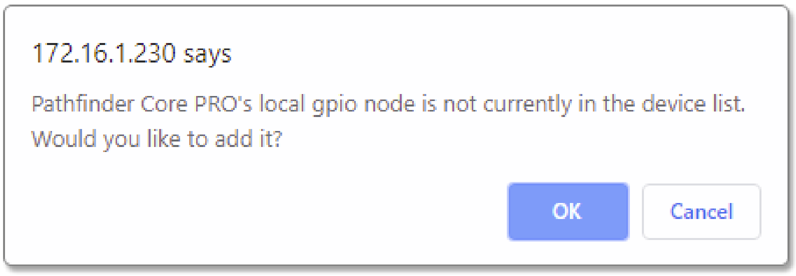
2	If you are adding an Axia audio or GPIO device, select **Lwrp** as the investigation type. *See note below.* If you are adding an Axia rackmount OLED or LCD panel, use the **Lwcp** investigation type.
3	Clicking **Investigate** will cause the system to try and contact a device on the requested IP address using the defined investigation type. If a device ca be contacted and data can be discovered from the device, it will be added to the *Devices* list. Its routing resources will also be added to the requisite routers.

Virtual GPIO Node

Pathfinder Core PRO includes a virtual GPIO node which may be accessed by browsing to the **GPIO** navigation bar link under the **System** items.

This virtual node behaves very much like the GPIO portion of an Axia Livewire driver except that the number of ports is dynamically adjustable and includes a few additional features as described below.

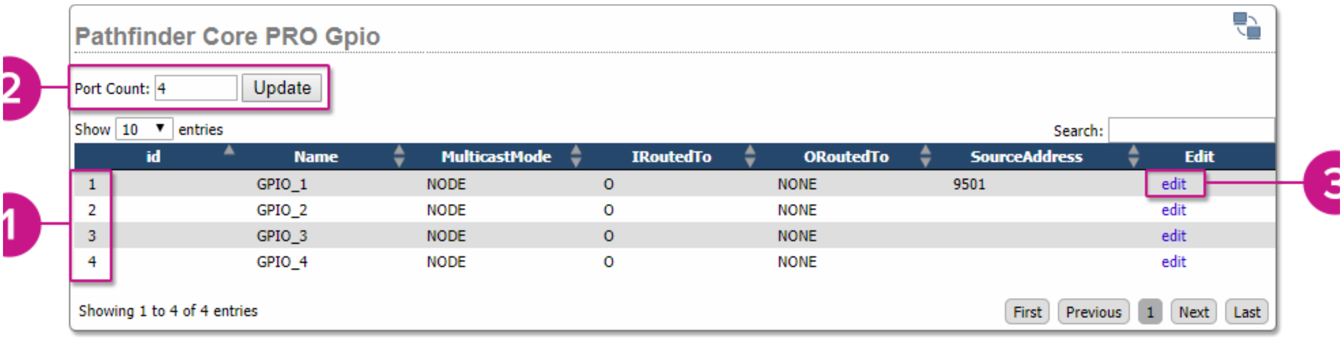
When you first open the GPIO link it will recognize that the internal GPIO node has not been added to the devices table and will ask if you would like to add it. The node may only be used and/or configured after it has been added to the device list.



172.16.1.230 says

Pathfinder Core PRO's local gpio node is not currently in the device list. Would you like to add it?

Click **OK** to discover and add the internal GPIO node into the device list. This will also discover the GPIO points into the GPIO router as well.



Pathfinder Core PRO Gpio

Port Count: 4

Show 10 entries

Search:

id	Name	MulticastMode	IRoutedTo	ORoutedTo	SourceAddress	Edit
1	GPIO_1	NODE	0	NONE	9501	edit
2	GPIO_2	NODE	0	NONE		edit
3	GPIO_3	NODE	0	NONE		edit
4	GPIO_4	NODE	0	NONE		edit

Showing 1 to 4 of 4 entries

Reference	Description
1	Once the device has been added you will see the list of GPIO ports. By default, there are 4 ports in the system.
2	You can add additional GPIO ports by typing a ne **port count** at the top of the screen and then clicking **Update**. This allows you to add any number of GPIO ports into the local virtual GPIO node.
3	Click the **edit** link to change the property values fo any GPIO port. *See the **Pathfinder Core PRO GPIO Properties** section below for details.*

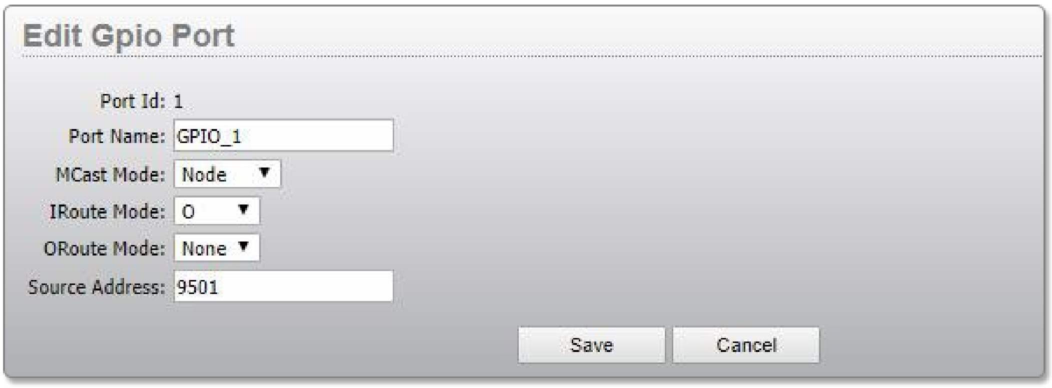
GPIO Refresher

It is important to understand the various ways GPIOs may be used in an Axia environment.

- Software GPIOs with no source address assignment will simply allow closures to be directly tripped on either GPI or GPO by Pathfinder Core PRO
  - Example: SRCA:””
- If a source address field for a GPIO port uses a Livewire channel number in its address field, it will listen to and generate closures to and from an Axia Console over multicast
  - Example: SRCA:9501
- Finally, if the source address uses an IPAddress/port format it means that the GPIO port should use TCP to connect to the device at that IP Address, monitor the GPIs from the selected port on that device, and mirror those closures on this port’s GPOs; this is the method used by the PathfinderCore PRO GPIO router to route GPIO data across a network
  - Example: SRCA:”172.16.1.23/8”

Pathfinder Core PRO GPIO Properties

By default, Pathfinder Core PRO’s GPIO node works in the same way as described above; clicking **Edit** in the GPIO table allows advanced users to manipulate advanced properties



Edit Gpio Port

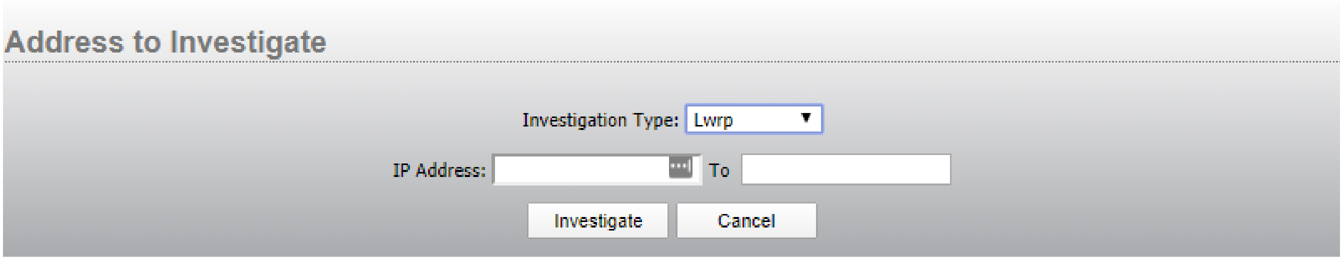
Port Id: 1
Port Name: GPIO_1
MCast Mode: Node
IRoute Mode: O
ORoute Mode: None
Source Address: 9501

Field	Description
Port Name	This allows you to type a unique **name** for this GPIO port.
MCast Mode	Selecting **Node** from the drop-down list allows the port to behave like a Livewire driver GPIO port when a Livewire channel number is typed in the source address. Specifically, the port will listen to GPIO messages from the console fader on which the Livewire channel number is loaded and chang its GPO pins accordingly. Tripping the GPI pins of the port will send closures to the Axia Console. Selecting **console** allows the port to behave as if were an Axia console. Therefore, GPIs that are tripped on this node will be sent to the GPOs of other GPIO ports on the network that have the same channel number assigned. Tripping a GPI o one of those other devices will cause the GPO on this port to change.
IRoute Mode	By default, this will be set to **O**. This property only applies to situations where an IP/Port or Livewire channel is used in the source address. When a G comes in from either of these sources, it is appliec to a GPO on the port. This property allows you to change this behavior and apply the incoming GPI to either the **GPI** or to **None** which means it is ignored.
ORoute Mode	This property defines where inbound GPOs will b sent. By default, they are sent nowhere as this setting replicates other GPIO node functionality. When using IP/Port source addressing the Pathfinder Core PRO GPIO node is unique in that will also subscribe to GPO changes for the port ar these may be routed to **I**, **O**, or **None**. *See note below.*
Source Address	This field may be left blank if the closures will be used by Pathfinder and do not need to follow another port or console functionality. Alternatively, you may type a **Livewire channel** number or **IP/Port value** to this field. If you type an IP/Port value, you do not need to assign it in the configuration user interface. The GPIO router will allow you to manipulate this field much more easi by just making route changes.

ⓘ Important Note: By manipulating **IRoute Mode** and **ORoute Mode**, it is possible to make a multicast or unicast GPIO route where one port's inputs and outputs mirror another port.

RestApi

PathfinderCore PRO also allows for the addition of Rest API devices which can be used to send rest api commands to a rest api enabled device. In order to add a rest api device, click on the devices link and then click the plus icon. This will display the Address to Investigate dialog.



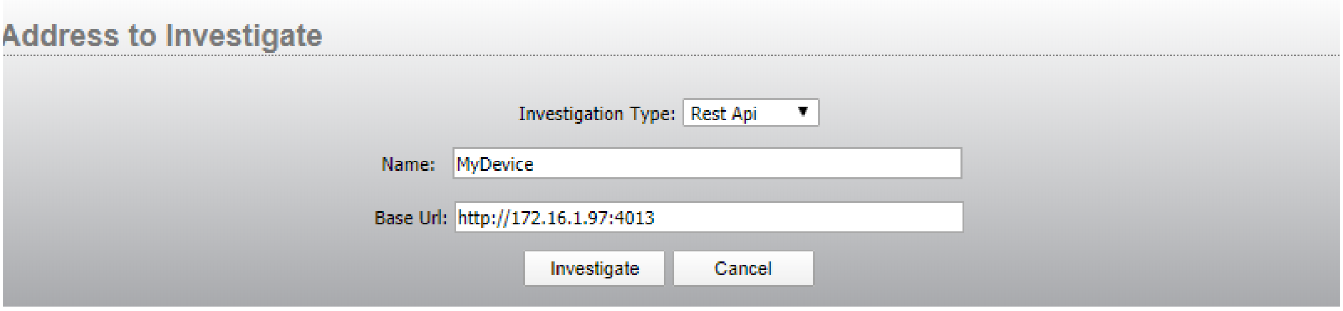
Address to Investigate

Investigation Type: Lwrp

IP Address: To

From the Investigation Type drop down, select Rest API. The dialog will change to show a name and a URL field.

Enter a name for the device and the http address to which to send rest api commands. Alternate ports may be used by adding a colon and the port number:



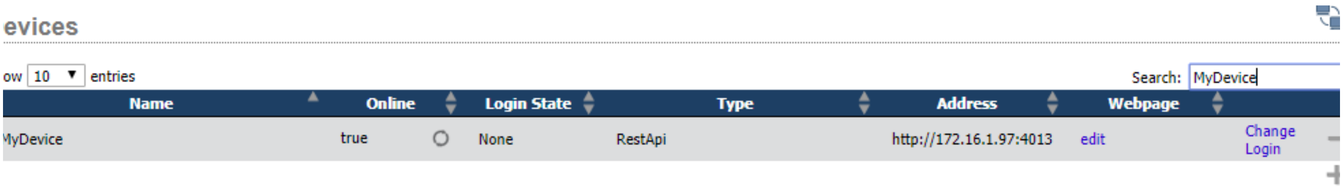
Address to Investigate

Investigation Type: Rest Api

Name: MyDevice

Base Url: http://172.16.1.97:4013

After clicking Investigate, the device will be added to the system.



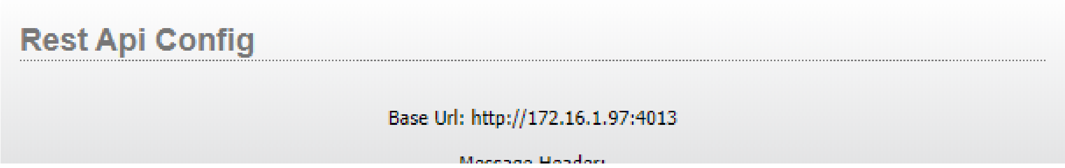
evices

ow 10 entries

Search: MyDevice

Name	Online	Login State	Type	Address	Webpage	
MyDevice	true ○	None	RestApi	http://172.16.1.97:4013	edit	Change Login

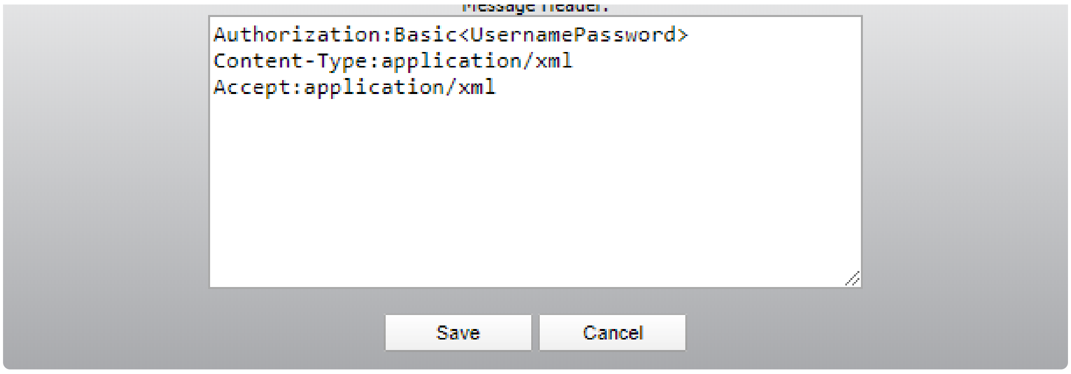
You will notice that for rest api devices, you will see an edit link instead of a web page link. The edit link will bring up a dialog for customizing the message header used by each rest api command:



Rest Api Config

Base Url: http://172.16.1.97:4013

Message Header:



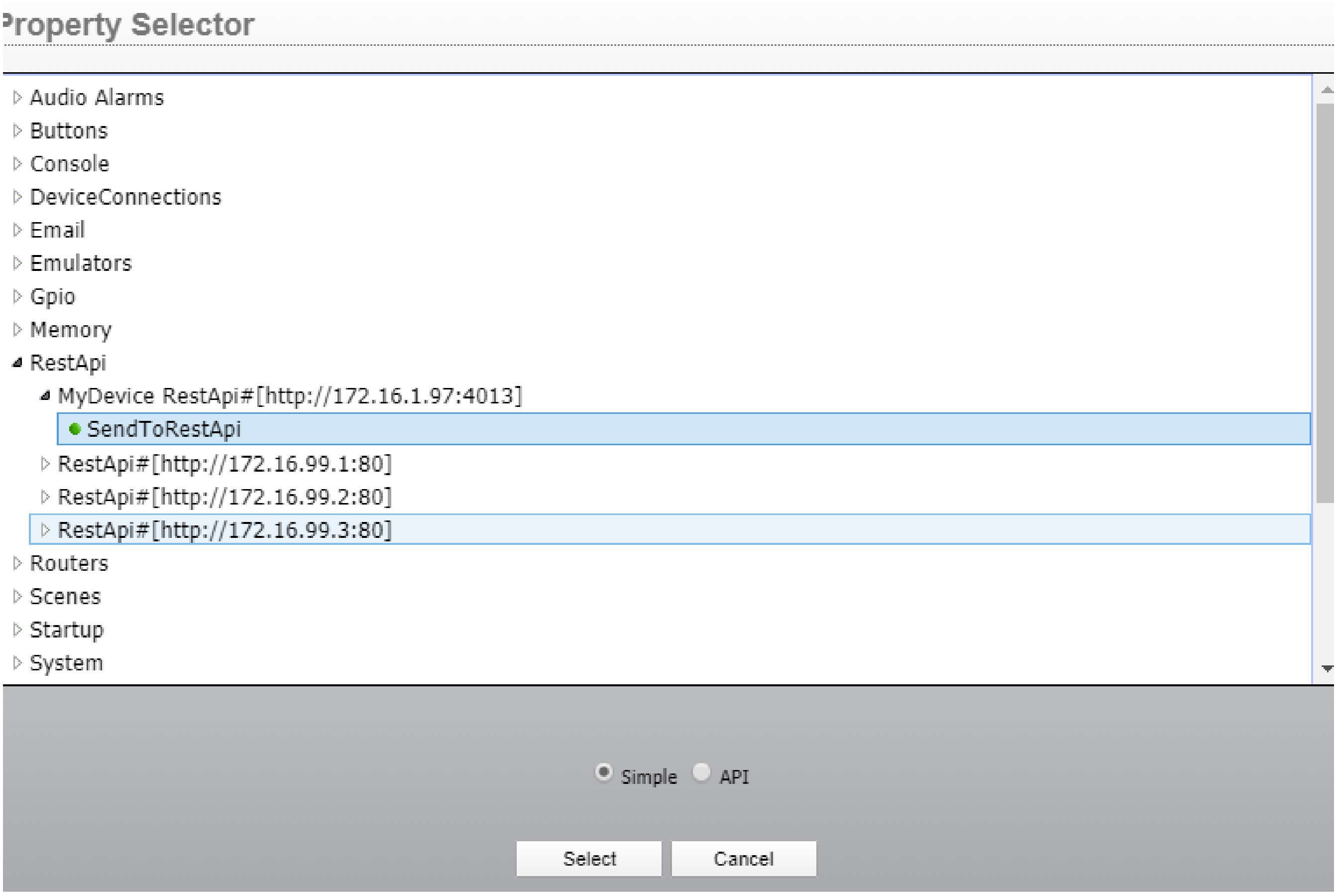
```
Authorization:Basic<UsernamePassword>
Content-Type:application/xml
Accept:application/xml
```

The <UsernamePassword> chunk will be replaced by a basic authentication conversion of the user name and password. If no user name and password is defined, the Authorization may be removed. Currently this device only supports basic authentication. You can use this dialog to add custom header messages if they are needed by the rest api device.

The ChangeLogin link allows you to assign a username and password to be used by rest api commands in the Authorization portion of the header.

Sending a command to the Rest API device

Once the device has been added to the system, it will be available in logic flows.



The SendToRestApi property needs to be structured in a specific way. The property value to send needs to have an operator (usually GET or PUT), a path, and optionally, some data. Note that the path portion should be only the data after the device path. For example:

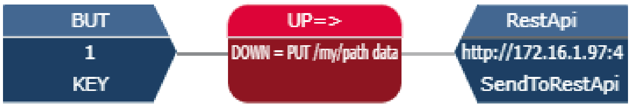
PUT /my/path/url hello

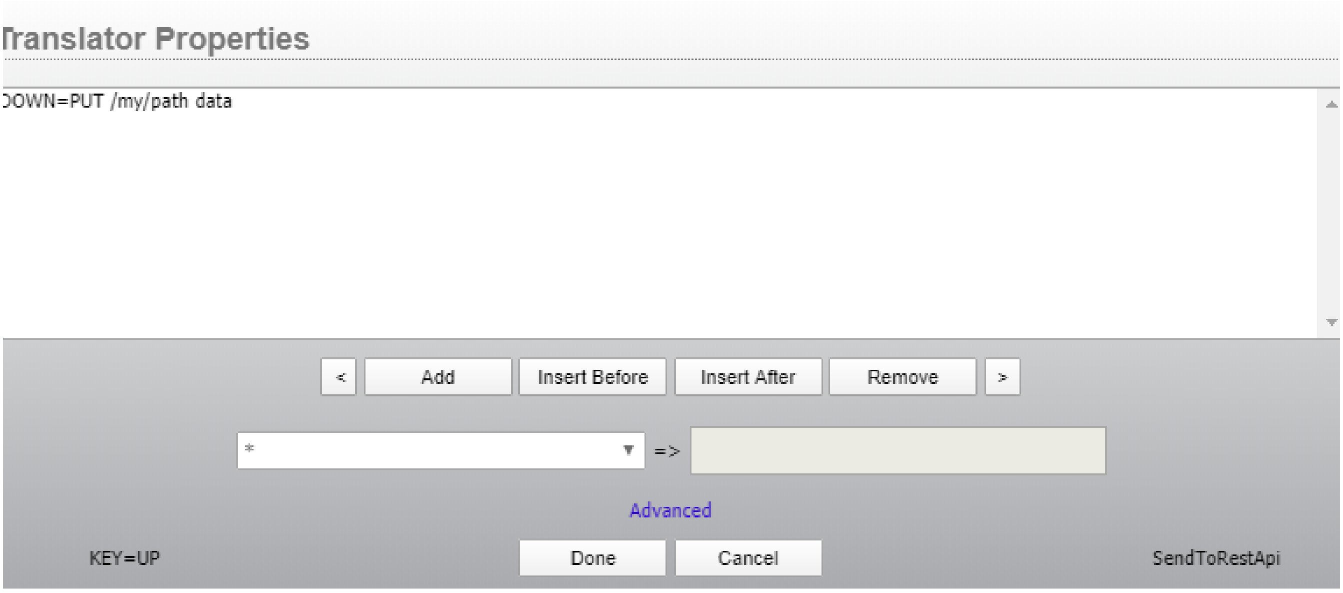
When the device sends data it will prepend the device path when it sends to the device so that the http command becomes:

PUT http://172.16.1.97:4013/my/path/url hello

The data portion is optional and depends on the specification of the api you are trying to communicate with.

For example, a logic flow that sends to a rest api device when a button is pushed might look like:



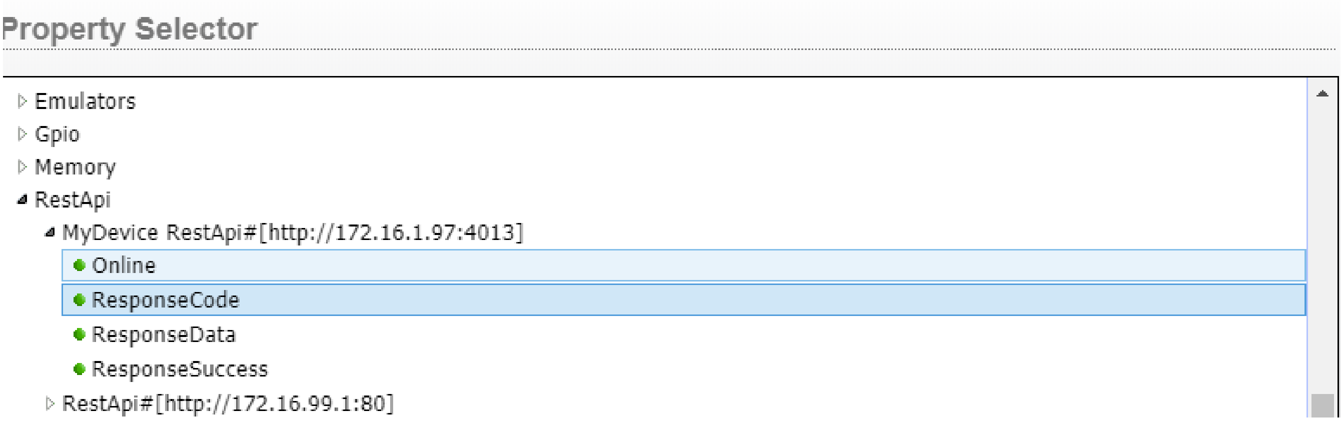

Translator Properties

DOWN=PUT /my/path data

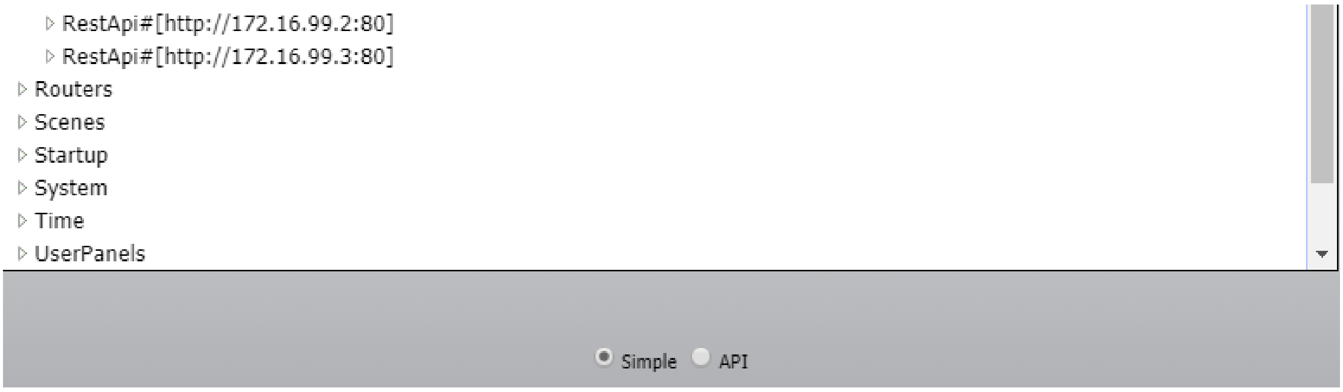
SAP property memory slots and string builder memory slots can also be useful to build rest api commands according to the device's specification and then pass the command through to the device whenever the memory slot changes.

Receiving data from the Rest API device

Please review the rest api technical details for more information on the pitfalls of using rest api to capture information from a device. In this case you would use the example above to send a request to the device probably using the GET operator. The information that is returned can be found in three read only properties (only seen if a start point is selected):


Property Selector

- Emulators
- Gpio
- Memory
- RestApi
  - MyDevice RestApi#[http://172.16.1.97:4013]
    - Online
    - ResponseCode
    - ResponseData
    - ResponseSuccess
  - RestApi#[http://172.16.99.1:80]



▷ RestApi#[http://172.16.99.2:80]
▷ RestApi#[http://172.16.99.3:80]
▷ Routers
▷ Scenes
▷ Startup
▷ System
▷ Time
▷ UserPanels
● Simple ○ API

Response code returns the http response code that results from the last message sent. For example, it might return 200 if the request succeeded. Or it might return 404 if the path was not found. For more information on http response codes, see: https://developer.mozilla.org/en-US/docs/Web/HTTP/Status.

The response data property will contain the actual data returned from the command. In many cases you may need a regex to filter the data for what you want to match in the logic flow.

The Response Success property will return true or false depending on whether the last rest api command was successful or not.

Rest API technical details

While rest APIs are currently very popular (thus the reason we are adding them to PathfinderCore PRO), we regard them as generally ill-suited for many broadcast applications.

For example, we recently did some work with a product whose control protocol was exclusively Rest API. One of the first things our customer wanted to do was have PathfinderCore Pro do an action whenever a user pressed a button. The problem is that in order to detect that button push we would have needed to poll at 2.5 times the minimum amount of time the button might be held down. For computers, this is not difficult until you start considering 25 buttons to a device and deployments of 600 to 700 devices. Now you have a situation where you are spending massive amounts of compute cycles on messages which basically say "nothing has changed".

The point of that story is to say that applications that need low latency responses to state changes do not scale well with non-event driven protocols like Rest API. The consumer of such protocols needs to be cognizant of the CPU load ramifications for what they are doing. Currently, if the user wants to poll, they would need to use an interval timer and a logic flow. We may at some point add a way to pre-program parameters that should be polled which would simplify that task, but we opted not to in this incarnation because of the potential abuse possibilities described above.

If all you care about is triggering a change and perhaps getting a response back that the change was made, then Rest APIs work well. But if real-time state change information is required, then care needs to be taken and other options employed if available.

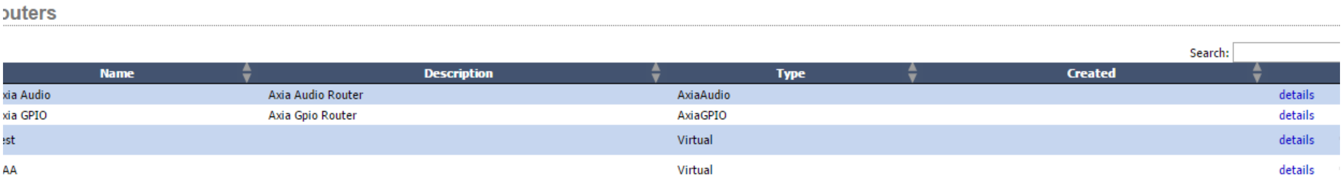
This article provides a great analogy using a General and his army to illustrate the differences between REST and a socket-based architecture:

https://www.pubnub.com/blog/websockets-vs-rest-api-understanding-the-difference/

In short: Use this feature with care.

Routers

Clicking on the **Routers** link in the navigation bar will present a list of routers in the system:



ɔuters

Search:

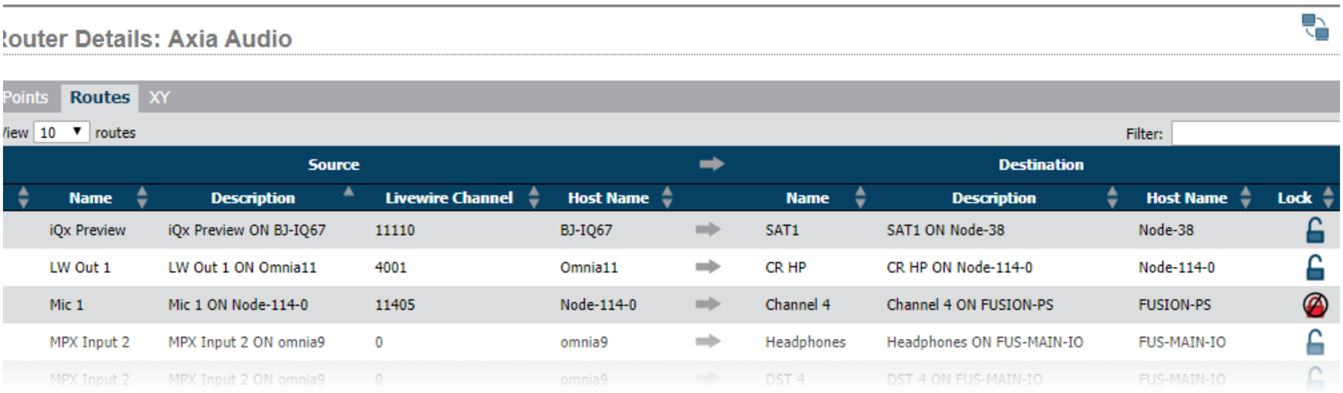
Name	Description	Type	Created	
xia Audio	Axia Audio Router	AxiaAudio		details
xia GPIO	Axia Gpio Router	AxiaGPIO		details
ɛst		Virtual		details
AA		Virtual		details

Pathfinder Core PRO currently supports:

- Axia Audio Routers
- Axia GPIO Routers
- Imagine Routers
- Virtual Routers
- Sap Property Routers

The Axia Audio Router and Axia GPIO Router are automatically generated based on devices discovered in the system. These two routers will always be in the system and may not be removed. Imagine Routers, Virtual Routers, and Sap Property Routers may be created and removed from the Routers page.

Clicking on the **Details** link for any router will present a new screen with the routing status of the router:



Ɩouter Details: Axia Audio

/iew 10 ▼ routes

Filter:

Source					Destination			
Name	Description	Livewire Channel	Host Name		Name	Description	Host Name	Lock
iQx Preview	iQx Preview ON BJ-IQ67	11110	BJ-IQ67	➡	SAT1	SAT1 ON Node-38	Node-38	🔓
LW Out 1	LW Out 1 ON Omnia11	4001	Omnia11	➡	CR HP	CR HP ON Node-114-0	Node-114-0	🔓
Mic 1	Mic 1 ON Node-114-0	11405	Node-114-0	➡	Channel 4	Channel 4 ON FUSION-PS	FUSION-PS	⊘
MPX Input 2	MPX Input 2 ON omnia9	0	omnia9	➡	Headphones	Headphones ON FUS-MAIN-IO	FUS-MAIN-IO	🔓
MPX Input 2	MPX Input 2 ON omnia9	0	omnia9	➡	DST 4	DST 4 ON FUS-MAIN-IO	FUS-MAIN-IO	🔓

This screen has three tabs on the top: **Points**, **Routes,** and **XY**.

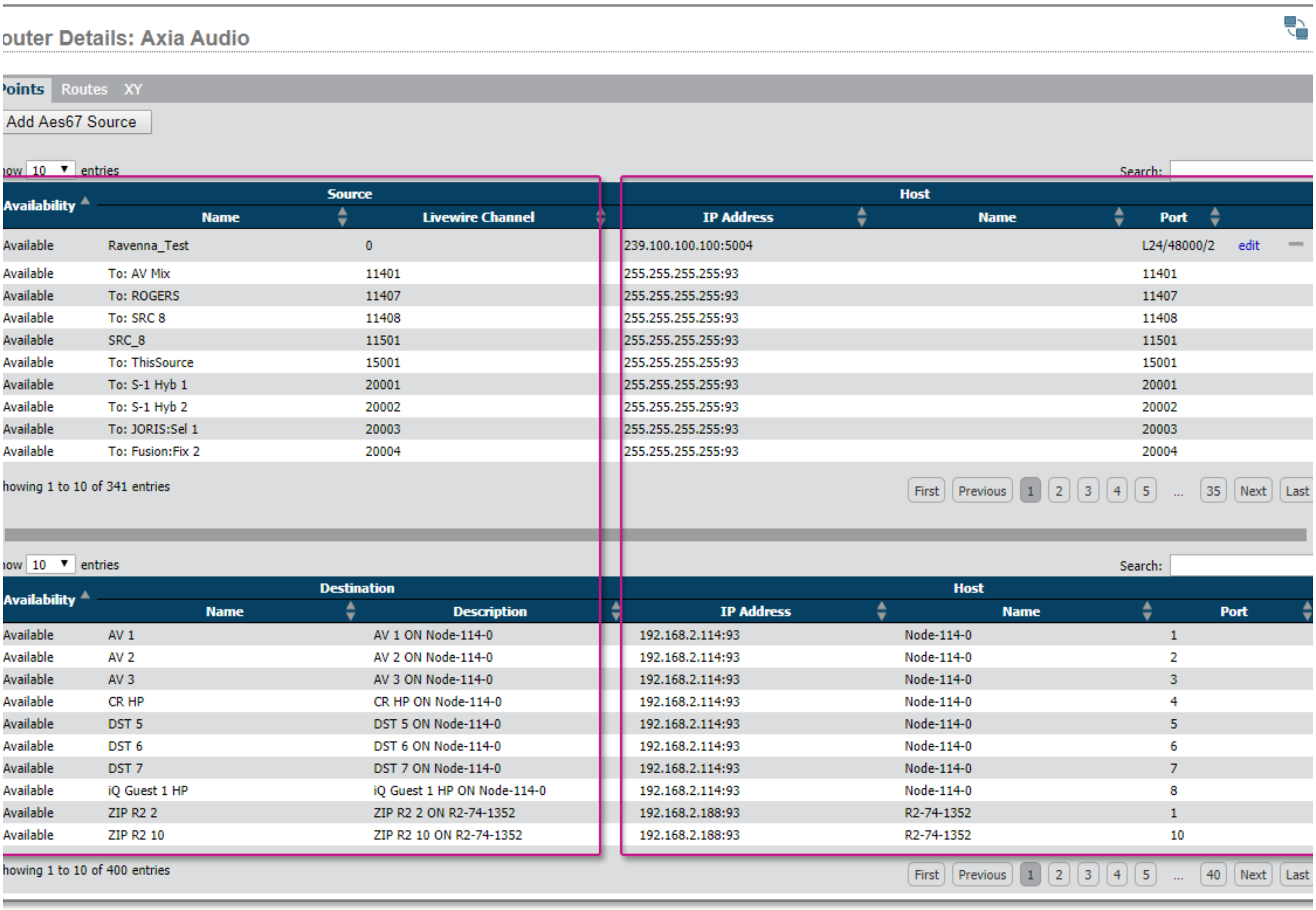
The Points Tab

The **Points** tab simply lists sources and destinations in the router along with relevant information for each. The columns in each list are broken into two major sections:

1. **Source/Destination:** Details related directly to the source or destination such as the source’s name,

Livewire channel, and availability.

2. **Host**: Information about the host device where a source or destination resides.



outer Details: Axia Audio

	Source		Host		
Availability	Name	Livewire Channel	IP Address	Name	Port
Available	Ravenna_Test	0	239.100.100.100:5004		L24/48000/2 edit
Available	To: AV Mix	11401	255.255.255.255:93		11401
Available	To: ROGERS	11407	255.255.255.255:93		11407
Available	To: SRC 8	11408	255.255.255.255:93		11408
Available	SRC_8	11501	255.255.255.255:93		11501
Available	To: ThisSource	15001	255.255.255.255:93		15001
Available	To: S-1 Hyb 1	20001	255.255.255.255:93		20001
Available	To: S-1 Hyb 2	20002	255.255.255.255:93		20002
Available	To: JORIS:Sel 1	20003	255.255.255.255:93		20003
Available	To: Fusion:Fix 2	20004	255.255.255.255:93		20004

howing 1 to 10 of 341 entries

	Destination		Host		
Availability	Name	Description	IP Address	Name	Port
Available	AV 1	AV 1 ON Node-114-0	192.168.2.114:93	Node-114-0	1
Available	AV 2	AV 2 ON Node-114-0	192.168.2.114:93	Node-114-0	2
Available	AV 3	AV 3 ON Node-114-0	192.168.2.114:93	Node-114-0	3
Available	CR HP	CR HP ON Node-114-0	192.168.2.114:93	Node-114-0	4
Available	DST 5	DST 5 ON Node-114-0	192.168.2.114:93	Node-114-0	5
Available	DST 6	DST 6 ON Node-114-0	192.168.2.114:93	Node-114-0	6
Available	DST 7	DST 7 ON Node-114-0	192.168.2.114:93	Node-114-0	7
Available	iQ Guest 1 HP	iQ Guest 1 HP ON Node-114-0	192.168.2.114:93	Node-114-0	8
Available	ZIP R2 2	ZIP R2 2 ON R2-74-1352	192.168.2.188:93	R2-74-1352	1
Available	ZIP R2 10	ZIP R2 10 ON R2-74-1352	192.168.2.188:93	R2-74-1352	10

howing 1 to 10 of 400 entries

Source/Destination Columns

Column	Description
Availability	This column indicates whether the source or destination is currently available for use. Options include: • **Available**—The device is operating normally • **Offline**—The device is offline or unreachable • **SystemLocked**—Destinations whose source cannot be changed by the router; for example mixer faders cannot be changed using standard routing commands and must be changed using the console or by issuing special source profile load commands
Name	The **Name** of the source or destination.
Livewire Channel (Axia Audio Sources Only)	The Livewire channel number assigned to the source.

Description (Destinations)	The **description** of the destination.

Host Columns

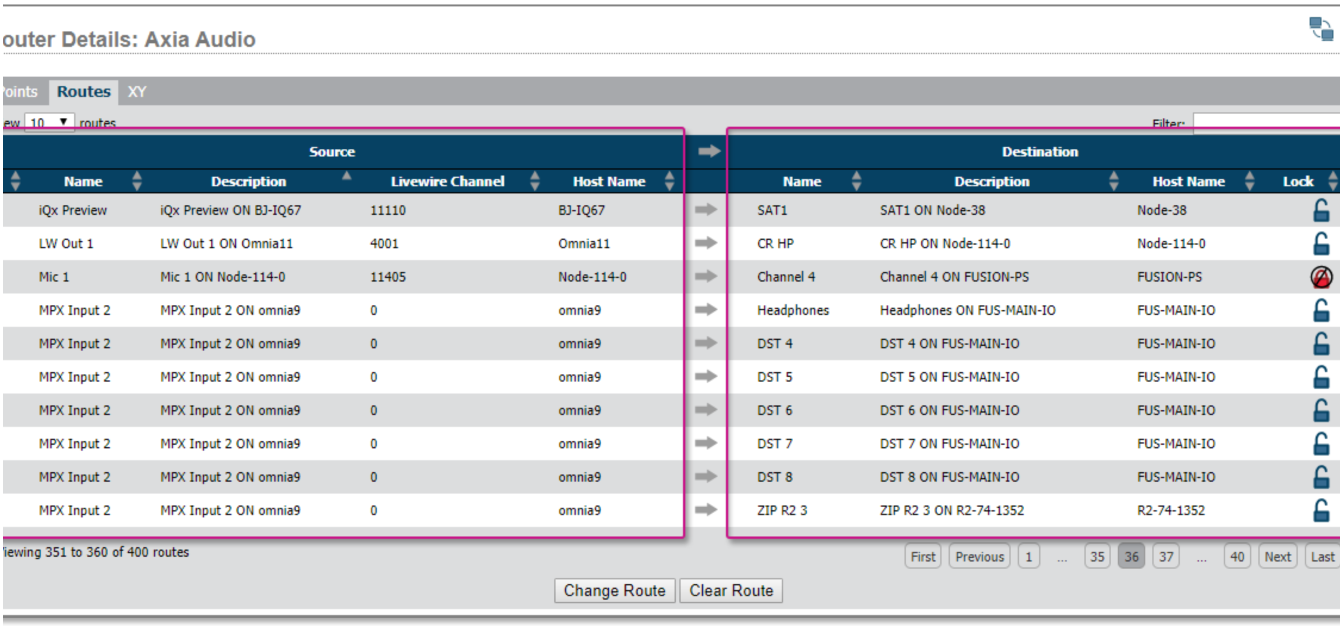
Column	Description
IP Address	The **IP Address** of the source or destination host device.
Name	The **name** of the source or destination host devic
Port	The physical (or virtual) **port** number that identifie the source or destination on the device.

Since these lists can contain many entries, the search box can be used to narrow the list.

For users that have administrative rights to the router, some additional configuration tools may also be present depending on the type of router. For example, when viewing the Axia Audio router, a button will be present for adding AES67 sources to the router. In the Virtual router, controls will exist for adding additional virtual sources and destinations to the router. These additional controls will be discussed below under the section for each router type.

The Routes Tab

The **Routes** tab displays the current source-to-destination routing of the router.



outer Details: Axia Audio

Source					Destination			
Name	Description	Livewire Channel	Host Name		Name	Description	Host Name	Lock
iQx Preview	iQx Preview ON BJ-IQ67	11110	BJ-IQ67	➡	SAT1	SAT1 ON Node-38	Node-38	🔓
LW Out 1	LW Out 1 ON Omnia11	4001	Omnia11	➡	CR HP	CR HP ON Node-114-0	Node-114-0	🔓
Mic 1	Mic 1 ON Node-114-0	11405	Node-114-0	➡	Channel 4	Channel 4 ON FUSION-PS	FUSION-PS	⊘
MPX Input 2	MPX Input 2 ON omnia9	0	omnia9	➡	Headphones	Headphones ON FUS-MAIN-IO	FUS-MAIN-IO	🔓
MPX Input 2	MPX Input 2 ON omnia9	0	omnia9	➡	DST 4	DST 4 ON FUS-MAIN-IO	FUS-MAIN-IO	🔓
MPX Input 2	MPX Input 2 ON omnia9	0	omnia9	➡	DST 5	DST 5 ON FUS-MAIN-IO	FUS-MAIN-IO	🔓
MPX Input 2	MPX Input 2 ON omnia9	0	omnia9	➡	DST 6	DST 6 ON FUS-MAIN-IO	FUS-MAIN-IO	🔓
MPX Input 2	MPX Input 2 ON omnia9	0	omnia9	➡	DST 7	DST 7 ON FUS-MAIN-IO	FUS-MAIN-IO	🔓
MPX Input 2	MPX Input 2 ON omnia9	0	omnia9	➡	DST 8	DST 8 ON FUS-MAIN-IO	FUS-MAIN-IO	🔓
MPX Input 2	MPX Input 2 ON omnia9	0	omnia9	➡	ZIP R2 3	ZIP R2 3 ON R2-74-1352	R2-74-1352	🔓

iewing 351 to 360 of 400 routes

Again, this screen is divided into two sides. The **source** information for each active route is on the left, and the **destination** information is on the right. The columns display information about each active route:

Source Columns

Column	Description
Name	The **name** of the source routed to this destination.
Description	The **description** of the source routed to this destination.
IO (Virtual Router Only)	The **IO number** of the source in the virtual router.
Livewire Channel (Axia Audio Router Only)	The **Livewire channel number** of the source routed to this destination.
HostName	The **name** of the device where the source resides

Destination Columns

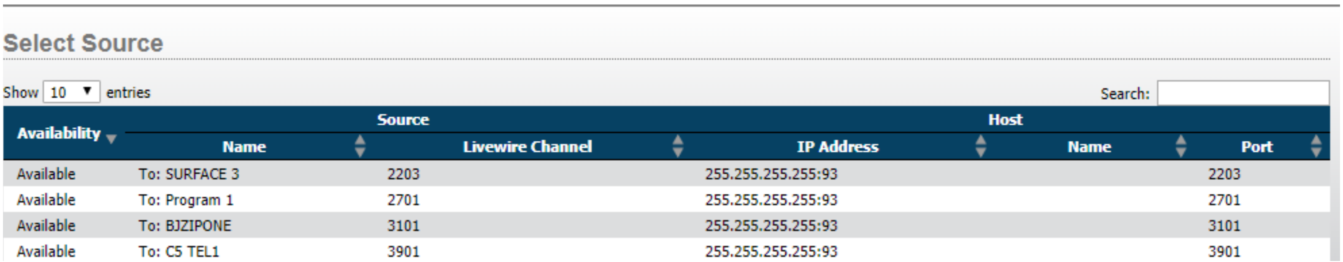
Column	Description
Name	The **name** of the destination.
Description	The **description** of the destination.
IO (Virtual Router Only)	The **IO number** of the destination in the virtual router.
HostName	The **name** of the device where the destination resides.
Lock	The **lock** state of the destination.

Route Changes

Route changes may be made from the Routes tab by either:

- Selecting an **entry** in the routing list and clicking **Change Route**
- Double-clicking on an **entry** in the routing list

Either of these actions will present a dialog with a list of sources that may be assigned to the currently selected destination:

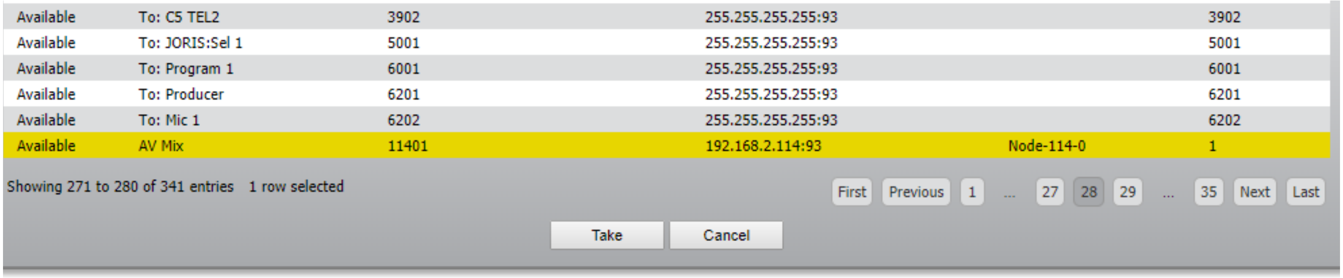


Select Source

Show 10 entries

Search:

Availability	Source		Host		
	Name	Livewire Channel	IP Address	Name	Port
Available	To: SURFACE 3	2203	255.255.255.255:93		2203
Available	To: Program 1	2701	255.255.255.255:93		2701
Available	To: BJZIPONE	3101	255.255.255.255:93		3101
Available	To: C5 TEL1	3901	255.255.255.255:93		3901



Available	To: C5 TEL2	3902	255.255.255.255:93		3902
Available	To: JORIS:Sel 1	5001	255.255.255.255:93		5001
Available	To: Program 1	6001	255.255.255.255:93		6001
Available	To: Producer	6201	255.255.255.255:93		6201
Available	To: Mic 1	6202	255.255.255.255:93		6202
Available	AV Mix	11401	192.168.2.114:93	Node-114-0	1

Showing 271 to 280 of 341 entries 1 row selected

On the *Select Source* dialog, select the **source** in the source list and click **Take** to change the route or double-click on a **source** in the source list.

Click **Cancel** to exit the route selection dialog without making changes

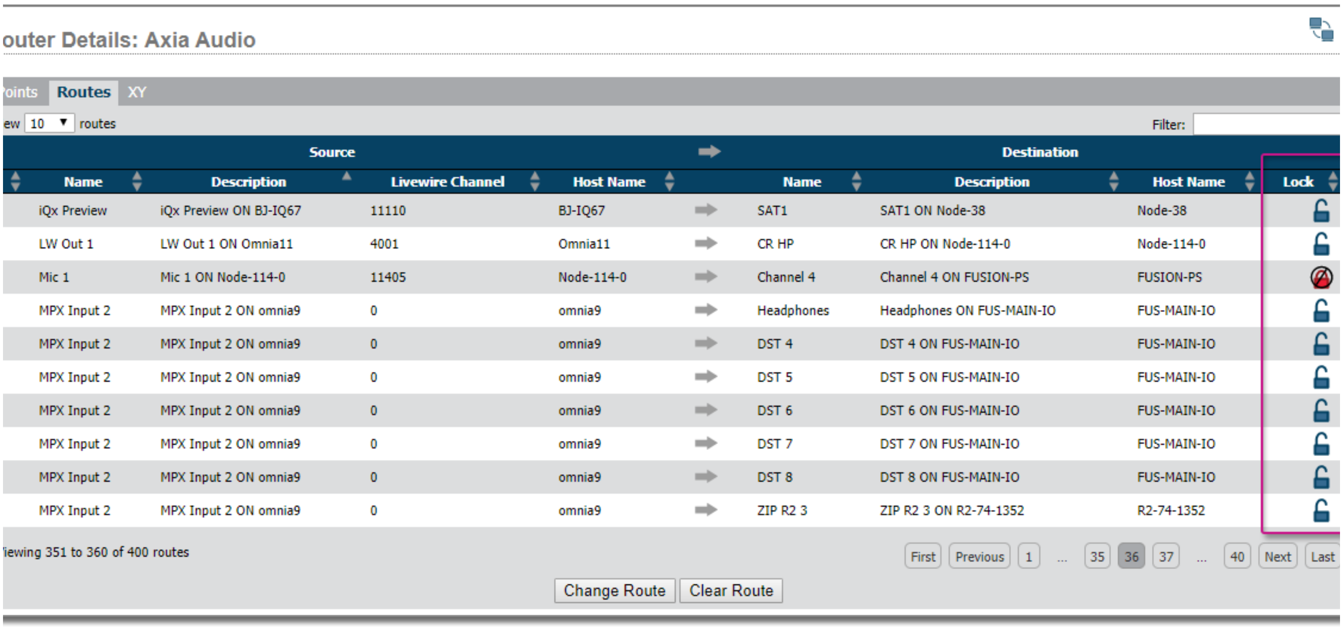
Since this list can contain many entries, the search box can be used to narrow the list.

The search filter can persist as you navigate between pages so if you return to the route page and are not seeing sources or destinations that you expect to find, check to see if there is a search filter on the list.

> ⓘ Important Note: A Source will not be available in the Routes list if it is not routed to anything. It will be present in the Take list when you go to make a change as well as in the Points tab under sources, but the search function is only searching data that is in the list. If the source is not routed to any destination, it is not in the routes list because it is not involved in any active route.

Locking

Each route (destination) in a router can be *Unlocked*, *Locked*, or *SystemLocked*.



outer Details: Axia Audio

Source					Destination			
Name	Description	Livewire Channel	Host Name		Name	Description	Host Name	Lock
iQx Preview	iQx Preview ON BJ-IQ67	11110	BJ-IQ67	➡	SAT1	SAT1 ON Node-38	Node-38	🔓
LW Out 1	LW Out 1 ON Omnia11	4001	Omnia11	➡	CR HP	CR HP ON Node-114-0	Node-114-0	🔓
Mic 1	Mic 1 ON Node-114-0	11405	Node-114-0	➡	Channel 4	Channel 4 ON FUSION-PS	FUSION-PS	⊘
MPX Input 2	MPX Input 2 ON omnia9	0	omnia9	➡	Headphones	Headphones ON FUS-MAIN-IO	FUS-MAIN-IO	🔓
MPX Input 2	MPX Input 2 ON omnia9	0	omnia9	➡	DST 4	DST 4 ON FUS-MAIN-IO	FUS-MAIN-IO	🔓
MPX Input 2	MPX Input 2 ON omnia9	0	omnia9	➡	DST 5	DST 5 ON FUS-MAIN-IO	FUS-MAIN-IO	🔓
MPX Input 2	MPX Input 2 ON omnia9	0	omnia9	➡	DST 6	DST 6 ON FUS-MAIN-IO	FUS-MAIN-IO	🔓
MPX Input 2	MPX Input 2 ON omnia9	0	omnia9	➡	DST 7	DST 7 ON FUS-MAIN-IO	FUS-MAIN-IO	🔓
MPX Input 2	MPX Input 2 ON omnia9	0	omnia9	➡	DST 8	DST 8 ON FUS-MAIN-IO	FUS-MAIN-IO	🔓
MPX Input 2	MPX Input 2 ON omnia9	0	omnia9	➡	ZIP R2 3	ZIP R2 3 ON R2-74-1352	R2-74-1352	🔓

iewing 351 to 360 of 400 routes

The lock state is represented by an icon in the Router's *Details* table:

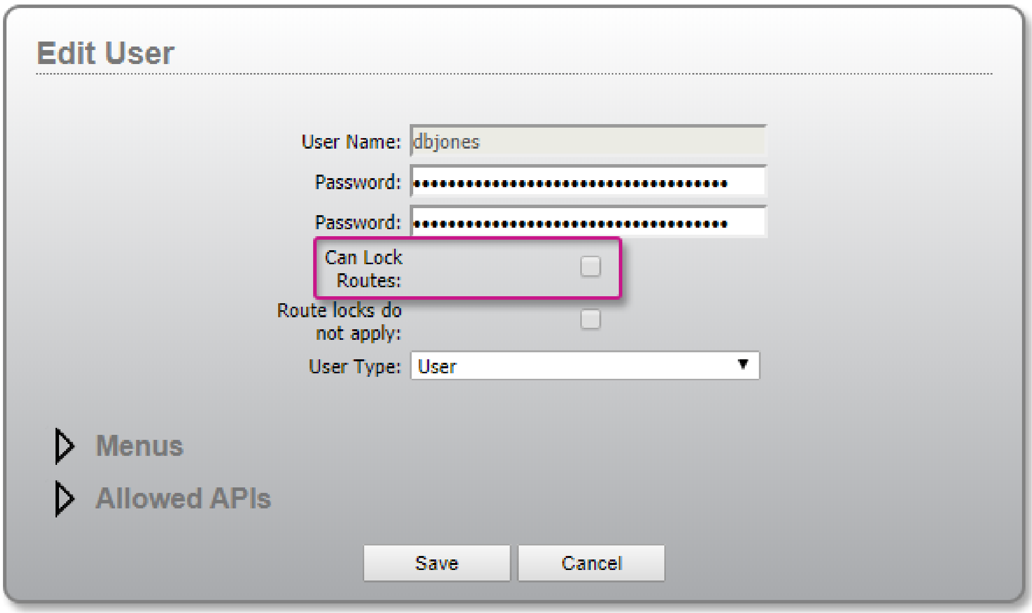
Lock State	Icon

Locked	🔒
Unlocked	🔓
System Locked	🚫

System Locked (except for virtual routers described below) indicates that the device itself does not allow route changes to be made on that IO. This is typically seen with console faders where source changes must be accomplished via a source profile load rather than a route change.

> ⓘ Important Note: Other than system locks, locking and unlocking is a Pathfinder state and does not actually change anything in the equipment as Axia equipment does not have lock parameters. Therefore, a locked route in Pathfinder Core PRO could still be changed by the device's web page.

Locking or unlocking a route is as simple as clicking on the *Destinations* icon in the route list. Locked routes must be unlocked before they can be changed. In addition, it is possible to define whether a user has permission to change route locks. When defining a user, a field now exists to configure this:



Edit User

User Name:	dbjones
Password:	••••••••••••••••••••••••••••••••••••••
Password:	••••••••••••••••••••••••••••••••••••••
Can Lock Routes:	□
Route locks do not apply:	□
User Type:	User

Checking the **Can Lock Routes** checkbox will grant this user permissions to lock and unlock routes. This allows an Administrator to lock routes and make them unchangeable by normal users.

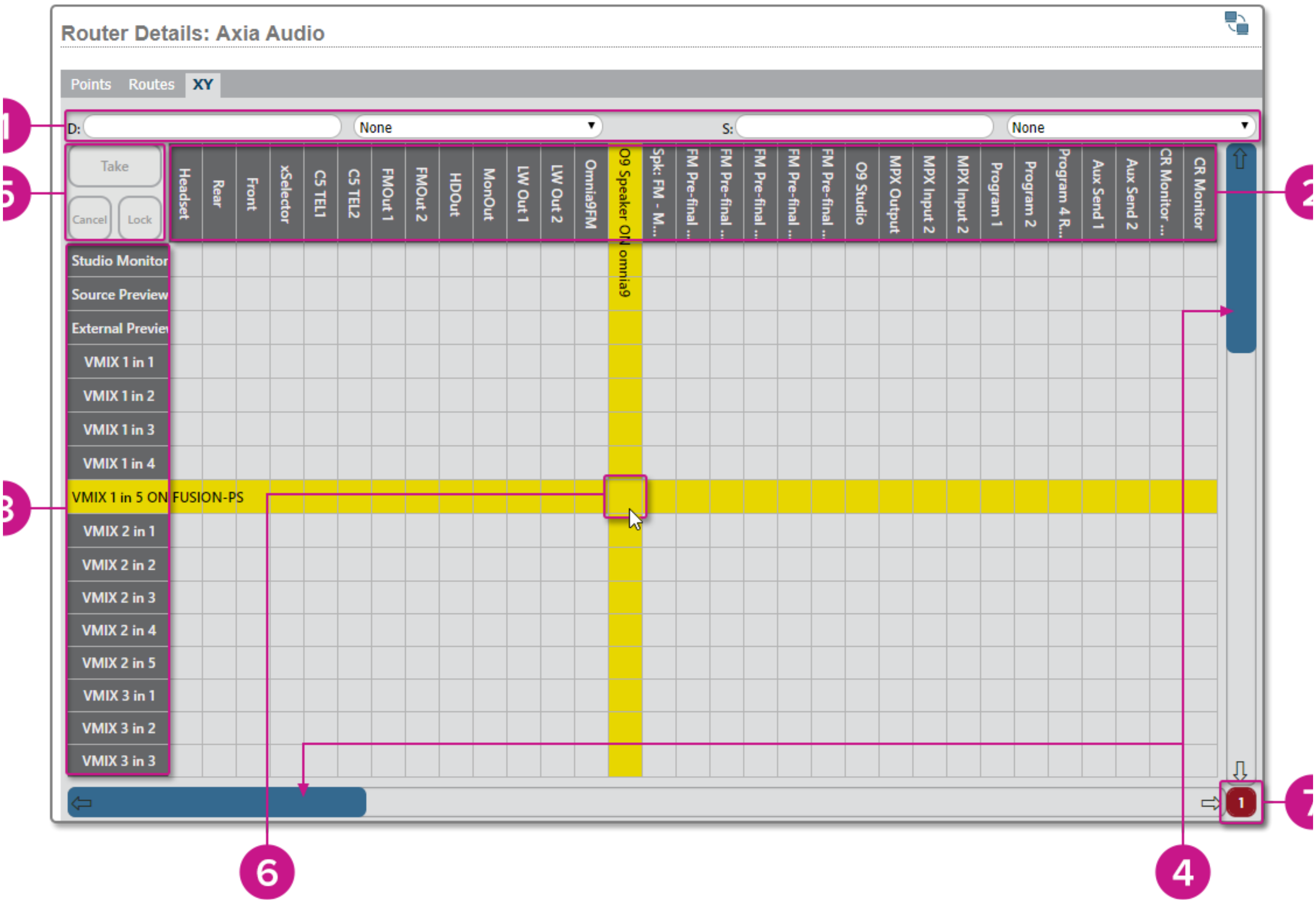
The **Route locks do not apply** option is reserved for future development.

> ⓘ Important Note: A **virtual** destination is comprised of one or more base destinations from other routers. If you lock a virtual destination, it will not lock the underlying base destination. This is by design and allows an Administrator to create a general user router with locked destinations and an engineering router where those same destinations are unlocked. However, if the base point becomes locked, then the virtual destination will display as system locked and the lock can only be removed by unlocking the lock on the base point. In the case where a virtual destination has multiple base points, the locking of any of the base points results in the entire virtual destination becoming system locked.

> The fact that locking a virtual destination does not lock its underlying base point is different from how previous versions of Pathfinder worked.

The XY Tab

The XY tab presents an XY grid which allows the user to graphically make route changes. This tab also mirrors the functionality of the XY component available in HTML5 user panels.



Ref	Description	Notes
1	Search bar	**Search** sources or destinations by typing in either the **S** or **D** field to reduce the number of columns and rows shown. You search can be filtered even further by specifying a device using the dropdowns. *The device dropdowns are not present on virtual routers and only exist on the audio and GPIO routers.*

2	Sources	**Source Names** are displayed the columns at the top of the table. Hovering over any colum header will highlight the source with a yellow bar and show the source’s full description. Double-clicking a source column header repeatedly will cause the grid to jump to each destination the source is routec to and highlight it with the yello hover bars. If the source is not routed to any destination, nothing will be highlighted.
3	Destinations	Destination Names are displayed to the left of each rov Hovering over any row header will highlight the destination wi a yellow bar and show the destination’s full description. Double-clicking on a destinatic row header will cause the grid jump to the source that is currently routed to that destination. If no source is routed to the destination, nothir will be highlighted. Greyed-out destination names indicate the destination is locke either at the user level or syste level and cannot be changed. Double-clicking on a locked rov header will still navigate to the locked cross-point if a route exists.
4	Scroll Bars	Scroll bars will appear if there are more destinations or sourc than can be displayed. The scroll bars may be manipulatec by dragging anywhere within tl bar or by clicking and/or holdin the arrows at either end of the scroll bar. The scroll bars are slightly wider than some norma windows scroll bars because tl

		grid is optimized for touch pane use.
5	Action Buttons	To **take**, **clear**, or **lock** a route, click the desired cross-point to preset it and then click the corresponding button in the top left corner of the grid. There are three action button sets: 1. Take Clear The **Take/Clear** button set will display either Take or Clear depending on wheth the selected route is currently active or not. Clic this button to either Take or Clear the route. 2. Lock Unlock The **Lock/Unlock** button set may be used to add or remove a user-level lock to an existing cross-point. Th ability to unlock a route is restricted by user permissions. 3. Cancel The **Cancel** button will clear all selected presets without making any changes.
		Clicking a cross-point will pres it and enable the action buttons Each cross-point will have one of several colors indicating the cross-point state: ⬜ No route exists at that cross-point 🟦 The route exists at that

6	Routing Grid	cross-point 🚫 The cross-point is locked at the system level 🔒 The cross-point is locked at the user level 🟥 The cross-point has beer preset for an action
7	Preset Rotate	When a preset is selected, this button will show the count of selected presets. Clicking the button will cause the hover bar to highlight the selection and move the scroll bars to the correct preset cross-point. If the matrix has been configure in multiple preset mode, this button will loop through the currently selected presets. The button will be grayed out and unavailable for use when r selection (presets) have been made.

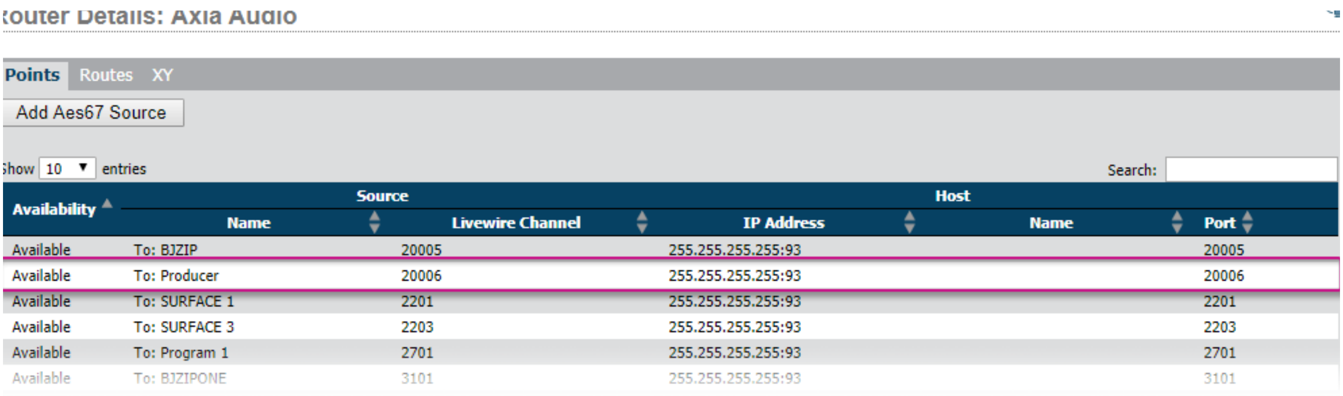
Router Types

Axia Audio Router

The Axia Audio Router is created automatically by the system. As devices are discovered or manually added to the system, they are analyzed to determine what audio sources and destinations they provide, and those sources and destinations are automatically loaded into this router.

When you use this router to make a route change, Pathfinder Core PRO is reaching out into the destination device and sending it a command to change the source assigned to that destination. This happens in much the same way as you could make the change from the device’s web page except that you can see all of the audio IOs in one location with Pathfinder Core PRO. Pathfinder Core PRO can also execute scene changes where it can make hundreds of route changes across many devices at once.

You may notice in the points tab that some sources will carry an IP Address of 255.255.255.255:93, have nothing in the host name column, and have “To:” at the start of their Name.



outer Details: Axia Audio

Points Routes XY

Add Aes67 Source

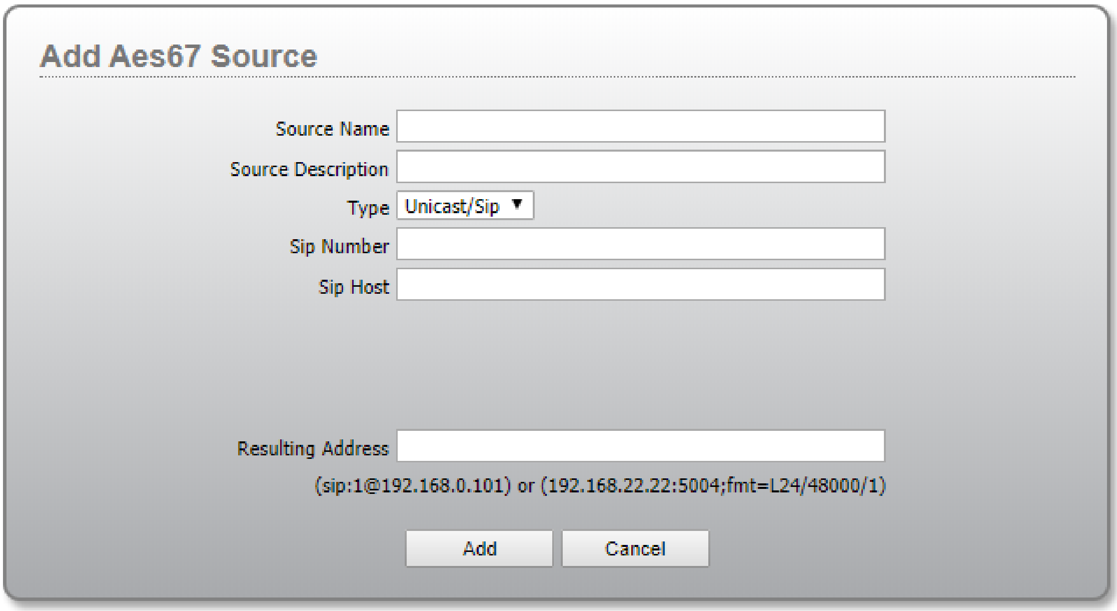
how 10 entries

Search:

Availability	Source		Host		
	Name	Livewire Channel	IP Address	Name	Port
Available	To: BJZIP	20005	255.255.255.255:93		20005
Available	To: Producer	20006	255.255.255.255:93		20006
Available	To: SURFACE 1	2201	255.255.255.255:93		2201
Available	To: SURFACE 3	2203	255.255.255.255:93		2203
Available	To: Program 1	2701	255.255.255.255:93		2701
Available	To: BJZIPONE	3101	255.255.255.255:93		3101

These sources are Axia Backfeeds generated by the mixing engines. They are not assigned a specific host because they are the one and only type of source that can dynamically change which device (Engine) is currently hosting them so they do not have a formal home. To learn more about Axia Backfeeds, see any of the Axia Mixing Console manuals.

The points tab of an Axia Audio Router will also have an Add AES67 Source button. Clicking this button will present a dialog for adding AES67 sources to the system.



Add Aes67 Source

Source Name

Source Description

Type Unicast/Sip

Sip Number

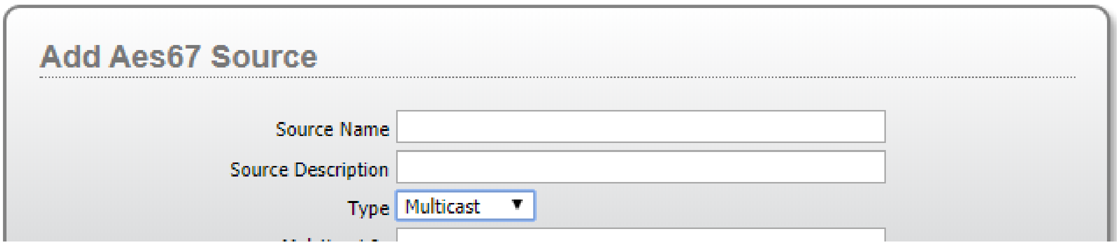
Sip Host

Resulting Address

(sip:1@192.168.0.101) or (192.168.22.22:5004;fmt=L24/48000/1)

Type the **name** and **description** by which the source will be identified in the system in the *Source Name* and *Description* fields. The AES67 specification allows for sources to be defined using a Unicast/Sip or multicast address format. The information required to define the source is different depending on the selected format and so different fields will be present depending on which definition you select. The example above shows the unicast method. Define the number and the host IP address required to define the Sip path. The resulting address field will automatically fill with the address required for this source based on these parameters.

If you select the multicast option, the required fields will change:

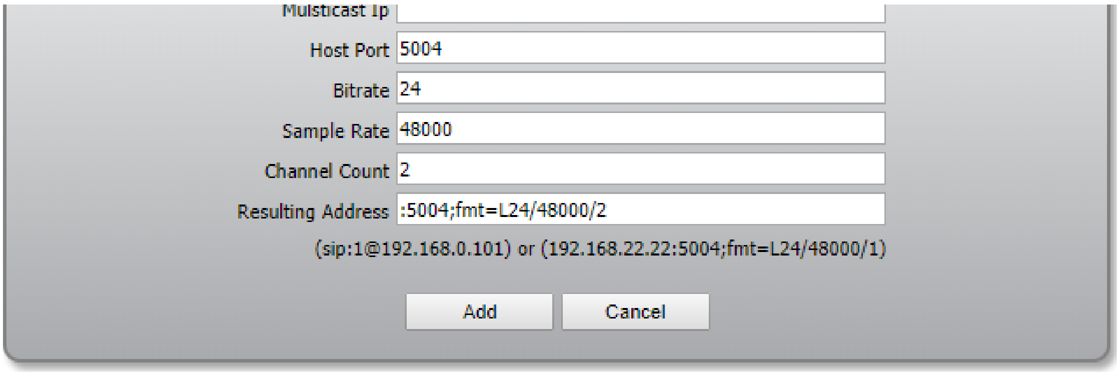


Add Aes67 Source

Source Name

Source Description

Type Multicast



Mulsticast Ip

Host Port 5004

Bitrate 24

Sample Rate 48000

Channel Count 2

Resulting Address :5004;fmt=L24/48000/2

(sip:1@192.168.0.101) or (192.168.22.22:5004;fmt=L24/48000/1)

In this case, fill in the multicast IP address and adjust the other parameters to reflect the AES67 source settings, then click add to add the source.

Once added, you will notice that in the list of points AES67 sources will have a minus icon to allow deletion of the source and an edit field to return to the editing dialog shown above.

Once these sources are added to the router, Pathfinder Core PRO can instruct any Axia device destination that supports AES67 streams to receive a specific AES67 stream in the same way as normal Axia route changes are made.

Axia GPIO Router

The Axia GPIO Router is very similar to the Axia Audio Router. As devices get discovered which have GPIO ports, those ports are added to this router. And in a similar fashion to the Axia Audio Router, when you make route changes, Pathfinder Core PRO is reaching into the destination node and sending it a command which tells a specific GPO on that device to follow the GPIs of the selected source device and port. Due to the nature of GPIOs on an Axia network, this routing becomes a little more nuanced and could use some explanation.

There are three ways to route GPIO closures across an Axia network. The Axia GPIO Router in Pathfinder Core PRO only uses one of these methods. It is still important to understand all of the methods however as attempting to use conflicting methods at the same time on a GPIO port can cause erroneous and unpredictable closures.

First, within the Axia device itself, you can assign a channel number to a GPIO port. When you do this, you are creating a special situation where you are telling the port to bind itself to Axia Mixing Console GPIO signals from whatever console currently has loaded that Livewire channel number.

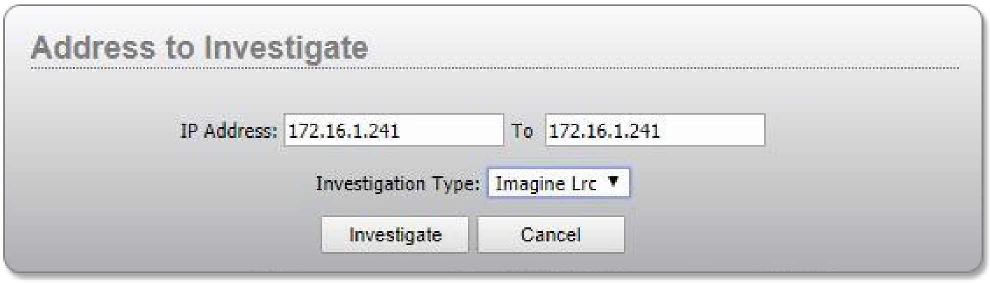
Second, you can assign an IP address and port number of a GPI to a GPO. This is called "snake mode" and it is what the GPIO Router uses to tell a device’s GPOs to mirror another port’s GPIs. Therefore, when you make a route change in the Pathfinder Axia GPIO Router from Node 1 Port 1 to Node 2 Port 4, any GPI closures on Node 1 Port 1, will be mirrored as GPOs in Node 2 Port 4. These route changes can also be updated dynamically, just as with any other router.

The third way that GPIOs can be used is to leave both the Livewire channel number and the routing assignment unassigned, then directly trip the closures using Pathfinder’s (or some other automation system’s) event system by sending closure commands to the device.

The reason it is important to understand these three methods of GPIO closures is that it is possible to do multiples of them at the same time which is occasionally useful but more often just a mistake. For example, we often see users who think that the GPIO ports should have a Livewire channel number and so just arbitrarily assign them. This can lead to a situation where the user has created Pathfinder events to make and respond to changes on those ports, but those same ports are also being fired by changes on the mixing console causing confusion and generating calls to Axia Support. It is important to first understand how you intend to use a given GPIO port then configure it correctly for that task.

Imagine Logical Router

To add an Imagine *Router* into Pathfinder Core PRO, you first must add the Imagine *Device* to the Devices list. In *Devices*, click **Add**, type the **IP Address**, and select **Imagine LRC** from the *Investigation Type* drop-down list.

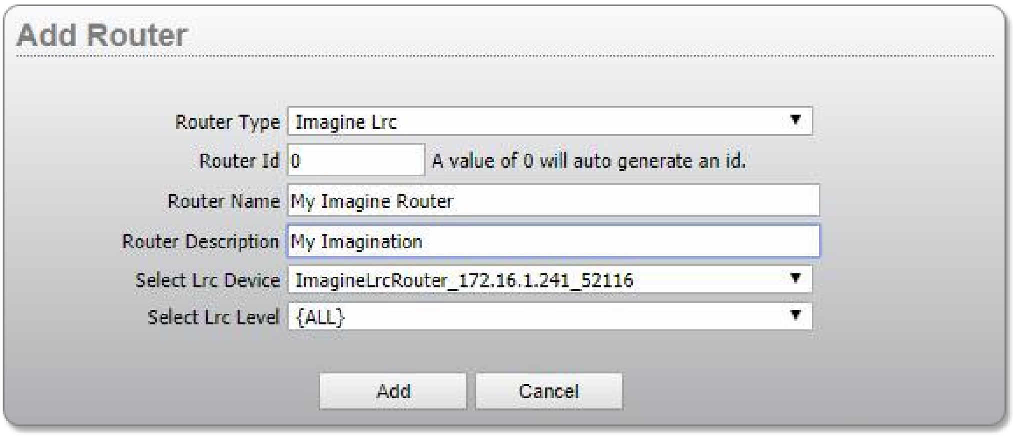


Address to Investigate

IP Address: 172.16.1.241 To 172.16.1.241

Investigation Type: Imagine Lrc ▼

Once the device has been successfully added, click on the **Routers** tab and click **Add a new Router**.



Add Router

Router Type: Imagine Lrc ▼

Router Id: 0 A value of 0 will auto generate an id.

Router Name: My Imagine Router

Router Description: My Imagination

Select Lrc Device: ImagineLrcRouter_172.16.1.241_52116 ▼

Select Lrc Level: {ALL} ▼

Field	Description
Router Type	Select **Imagine Lrc**.
Router Id	Id number of the router. Leave a value of **0** for an auto-generated Id.
Router Name	Type a descriptive **name** for this Router.
Router Description	Optionally, type a **description** for this Router.
Select Lrc Device	From the drop-down list, select the specific **devic**
Select Lrc Level	Select the LRC level in the router to control as described below.

Imagine routers have multiple levels and Pathfinder Core PRO can control full routes that route all levels at once by adding the {ALL} level. Any given breakaway level may also be added separately. Or, both may be added in different routers. This allows routes that should target all levels to be made on the router with the All target and routes of individual breakaway levels to be made on the breakaway routers.

Once added, the router will discover the IOs made available by the selected level and routes may be manipulated in the same way as other routers in the system

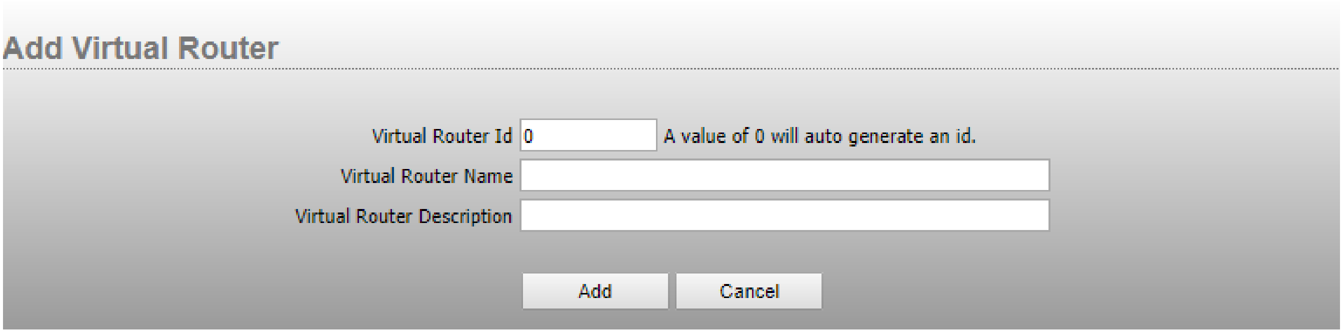
Virtual Routers

Virtual routers are a special kind of router. You can think of a virtual router as a way to make a subset of routes you wish a specific user to use. For example, you could create a router that only contains the routes from the Axia Audio Router that are related to a specific studio. Or you could create a router that only contains the routes that are relevant to the final air chains. This allows you to create purpose-driven routers that display only the routes that are relevant to a specific set of tasks and are therefore much easier to navigate than browsing every source and destination on the network.

However, virtual routers can also go much deeper than just subsets of routes. For any specific virtual source or destination, you can also associate multiple real sources or destinations with that virtual source. That sounds complicated but is actually relatively simple and extremely powerful.

For example, let's say that every time you route a specific Audio Source to a specific Audio Destination, you also want to route a GPIO source that carries signaling information related to that audio source to a specific GPIO destination. An example of where this might be useful would be switching from a primary automation system to a backup automation system by creating a virtual source that contains both an audio and a GPIO source plus a virtual destination that contains both an audio and a GPIO destination. Now, when you use the virtual router to route that source to the destination, it will tell the underlying audio and GPIO routers to make both routes.

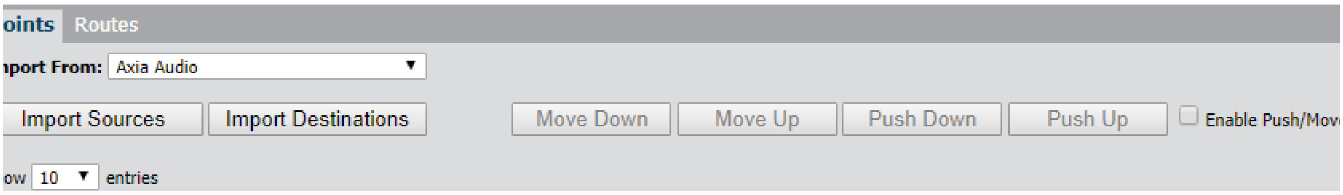
To create a virtual router, click the plus icon on the main routers page.


Add Virtual Router

Virtual Router Id: 0 A value of 0 will auto generate an id.
Virtual Router Name:
Virtual Router Description:

Provide a name and description for the router and click **Add**. The router id will autogenerate the next available router id number but can be overridden with your own id number. Be sure to use a unique id that is not already in use by another router.

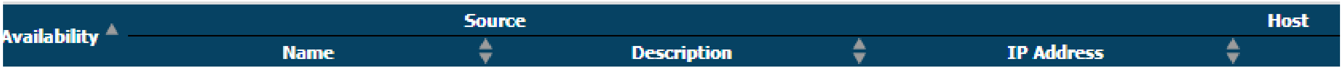
After clicking Add, the virtual router will appear in the routers list but is currently empty as it does not have any sources or destinations. Click the **Details** button and select the **Points** tab.



nport From: Axia Audio

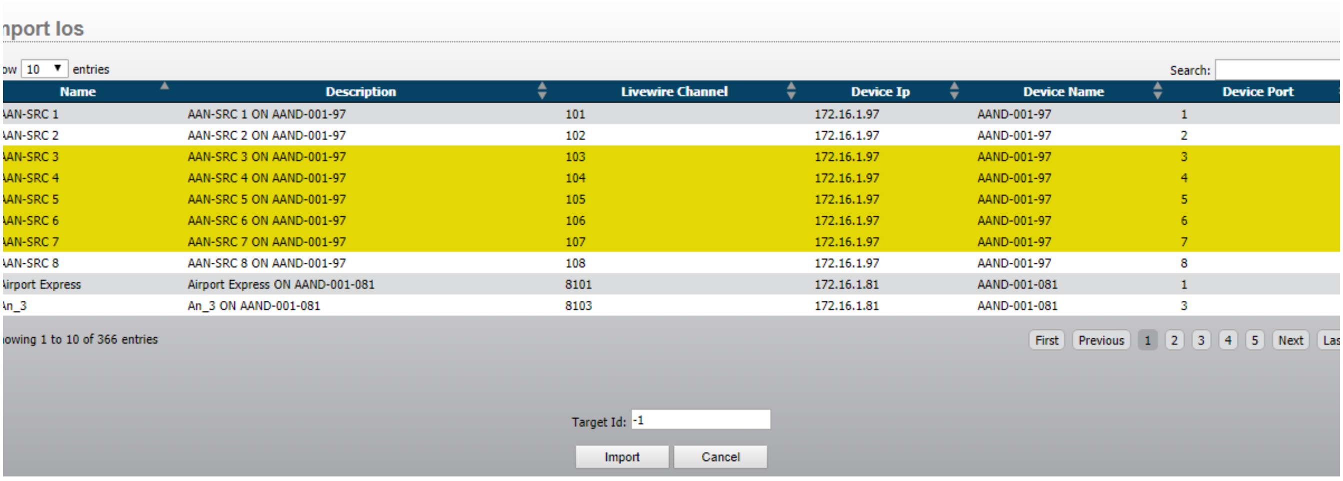
□ Enable Push/Mov

ow 10 entries



Availability	Source			Host
	Name	Description	IP Address	

The easiest way to populate a virtual router is to import some sources and destinations from a real router. In this case, we are going to select the Axia Audio Router from the import drop-down and then click Import Sources. You will then be presented with a list of importable sources.



nport Ios

ow 10 ▾ entries

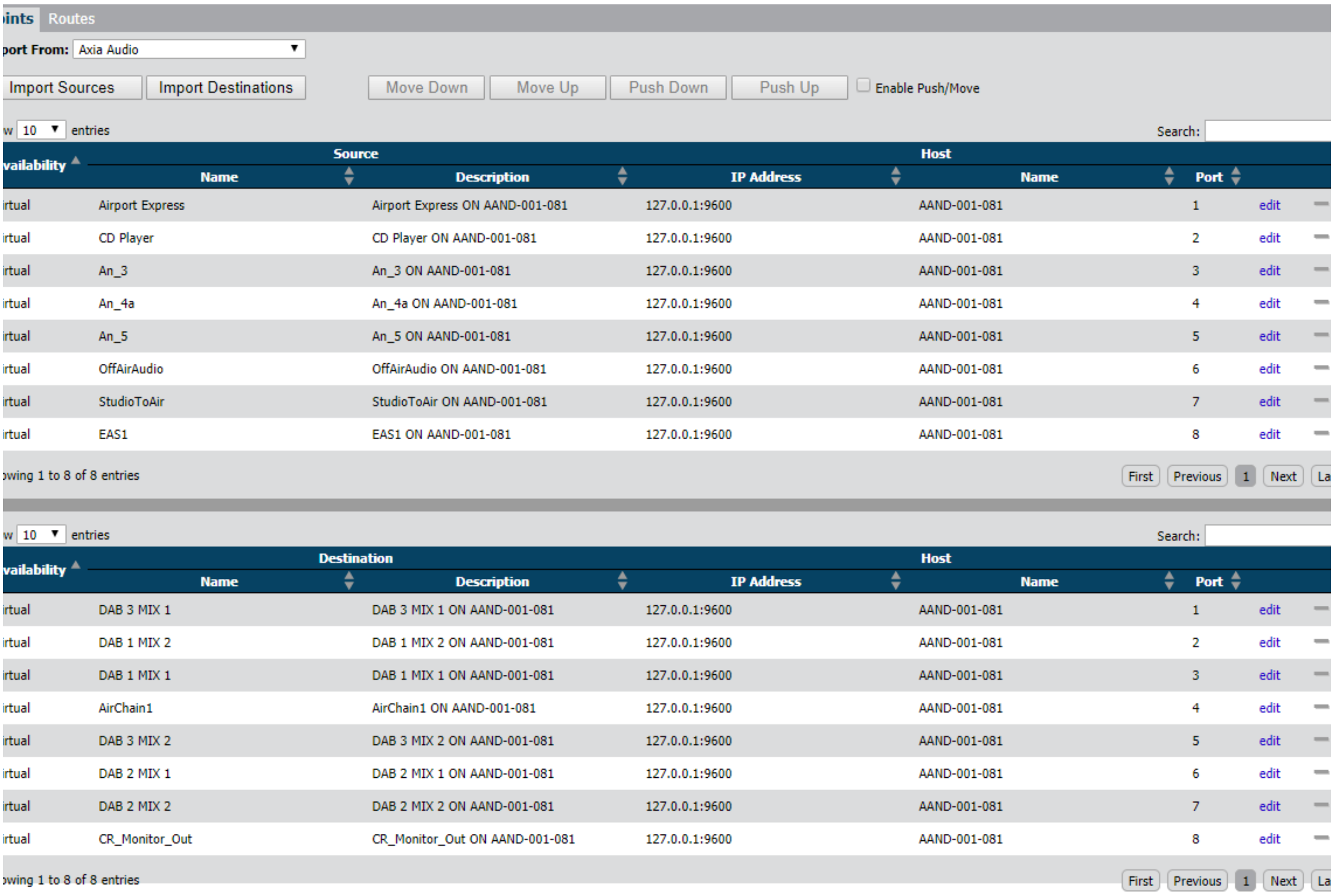
Search:

Name	Description	Livewire Channel	Device Ip	Device Name	Device Port
AN-SRC 1	AAN-SRC 1 ON AAND-001-97	101	172.16.1.97	AAND-001-97	1
AN-SRC 2	AAN-SRC 2 ON AAND-001-97	102	172.16.1.97	AAND-001-97	2
AN-SRC 3	AAN-SRC 3 ON AAND-001-97	103	172.16.1.97	AAND-001-97	3
AN-SRC 4	AAN-SRC 4 ON AAND-001-97	104	172.16.1.97	AAND-001-97	4
AN-SRC 5	AAN-SRC 5 ON AAND-001-97	105	172.16.1.97	AAND-001-97	5
AN-SRC 6	AAN-SRC 6 ON AAND-001-97	106	172.16.1.97	AAND-001-97	6
AN-SRC 7	AAN-SRC 7 ON AAND-001-97	107	172.16.1.97	AAND-001-97	7
AN-SRC 8	AAN-SRC 8 ON AAND-001-97	108	172.16.1.97	AAND-001-97	8
irport Express	Airport Express ON AAND-001-081	8101	172.16.1.81	AAND-001-081	1
n_3	An_3 ON AAND-001-081	8103	172.16.1.81	AAND-001-081	3

owing 1 to 10 of 366 entries

Target Id: -1

Click on the sources you wish to import (Shift-click to select a range) then click the **Import** button. Repeat the procedure with the Import Destinations button to import additional destinations into the virtual router. Note that in general, you will leave the target id at -1. This will import the IOs to the end of the router. Setting a number other than -1 will try to import the new IOs at the number selected. This will fail if there are not enough open numbers at the target id to support the number of selected IOs.



port From: Axia Audio ▾

□ Enable Push/Move

w 10 ▾ entries

Search:

vailability	Source			Host			
	Name	Description	IP Address	Name	Port		
irtual	Airport Express	Airport Express ON AAND-001-081	127.0.0.1:9600	AAND-001-081	1	edit	—
irtual	CD Player	CD Player ON AAND-001-081	127.0.0.1:9600	AAND-001-081	2	edit	—
irtual	An_3	An_3 ON AAND-001-081	127.0.0.1:9600	AAND-001-081	3	edit	—
irtual	An_4a	An_4a ON AAND-001-081	127.0.0.1:9600	AAND-001-081	4	edit	—
irtual	An_5	An_5 ON AAND-001-081	127.0.0.1:9600	AAND-001-081	5	edit	—
irtual	OffAirAudio	OffAirAudio ON AAND-001-081	127.0.0.1:9600	AAND-001-081	6	edit	—
irtual	StudioToAir	StudioToAir ON AAND-001-081	127.0.0.1:9600	AAND-001-081	7	edit	—
irtual	EAS1	EAS1 ON AAND-001-081	127.0.0.1:9600	AAND-001-081	8	edit	—

owing 1 to 8 of 8 entries

w 10 ▾ entries

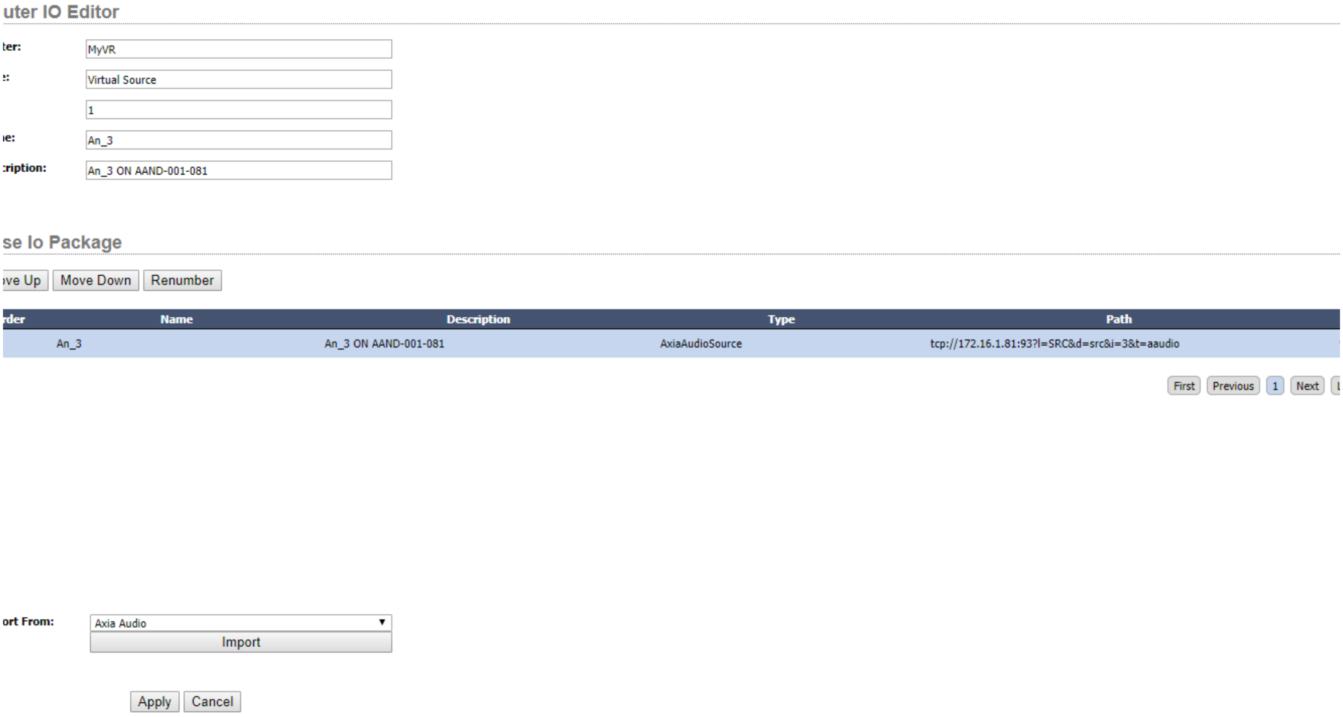
Search:

vailability	Destination			Host			
	Name	Description	IP Address	Name	Port		
irtual	DAB 3 MIX 1	DAB 3 MIX 1 ON AAND-001-081	127.0.0.1:9600	AAND-001-081	1	edit	—
irtual	DAB 1 MIX 2	DAB 1 MIX 2 ON AAND-001-081	127.0.0.1:9600	AAND-001-081	2	edit	—
irtual	DAB 1 MIX 1	DAB 1 MIX 1 ON AAND-001-081	127.0.0.1:9600	AAND-001-081	3	edit	—
irtual	AirChain1	AirChain1 ON AAND-001-081	127.0.0.1:9600	AAND-001-081	4	edit	—
irtual	DAB 3 MIX 2	DAB 3 MIX 2 ON AAND-001-081	127.0.0.1:9600	AAND-001-081	5	edit	—
irtual	DAB 2 MIX 1	DAB 2 MIX 1 ON AAND-001-081	127.0.0.1:9600	AAND-001-081	6	edit	—
irtual	DAB 2 MIX 2	DAB 2 MIX 2 ON AAND-001-081	127.0.0.1:9600	AAND-001-081	7	edit	—
irtual	CR_Monitor_Out	CR_Monitor_Out ON AAND-001-081	127.0.0.1:9600	AAND-001-081	8	edit	—

owing 1 to 8 of 8 entries

At this point, you have a virtual router with only the sources and destinations you want available. This router can then be made available as the only router certain users have access to by using the user rights section, or it can be used to easily see and change certain critical routes without viewing the entire Audio router.

However, let's dig a little deeper into each individual virtual IO. If you click the **Edit** link on one of the imported virtual sources, you will be presented with more details about the virtual source.



uter IO Editor

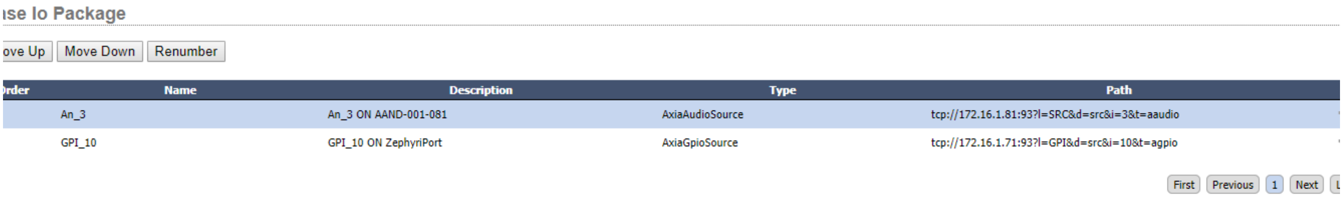
ter:	MyVR
:	Virtual Source
	1
ne:	An_3
cription:	An_3 ON AAND-001-081

se Io Package

rder	Name	Description	Type	Path
	An_3	An_3 ON AAND-001-081	AxiaAudioSource	tcp://172.16.1.81:93?l=SRC&d=src&i=3&t=aaudio

ort From: Axia Audio

By default, when you import a source the name and description fields of the virtual source inherit the name of the imported source. But you can change this to any name or description you want using the name and description field. The Base Io Package allows you to define which real sources are a part of this virtual source. As mentioned earlier, it is possible to tie multiple sources together into a single virtual source package. For example, we could select the Gpio router from the Import From dialog, click Import, then select a gpio source to include in this package.



ıse Io Package

Order	Name	Description	Type	Path
	An_3	An_3 ON AAND-001-081	AxiaAudioSource	tcp://172.16.1.81:93?l=SRC&d=src&i=3&t=aaudio
	GPI_10	GPI_10 ON ZephyriPort	AxiaGpioSource	tcp://172.16.1.71:93?l=GPI&d=src&i=10&t=agpio

The Move Up, Move Down, and Renumber buttons allow you to change the order of the sources in the virtual source package. Once applied, this virtual source now has both a gpio and audio source and so routing that virtual source to a virtual destination will attempt to make both a gpio and audio route change underneath. The same process can be used to edit and manipulate the virtual destinations.

The Points tab on the virtual router also includes a Plus icon for manually adding virtual sources and destinations and a Minus icon for deleting a virtual source or destination from the router.

Deeper Tech: Packaging

For those who are interested in what happens under the hood with virtual routers, this section is for you.

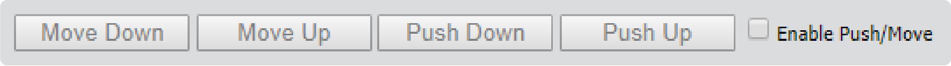
Every virtual source or destination is a package. Within that package is a list of pointers to real sources or destinations. When you make a route change, the system does a number of things.
First, it attempts to organize the package according to the IO types of the pointers in both the source and destination package. For example, it will lump all of the GPIO points together and all of the Audio points together in each package. Then it will try to make one to one routes based on the order of the routes in the package.

This also allows you to lump multiple Audio sources and destinations into a single package. For example, if you were using some sort of immersive sound configuration that required multiple channels of audio (7.1, 22.2) to be routed at once, you could package all of the sources together into a virtual source package and all of the destinations together into a virtual destination package. When the virtual route is made, Pathfinder Core PRO will step through each line in the source package and try to apply it to the same line in the destination package in the order which they exist in the package. If the count in the source and destination packages does not match, the system will go as far as it can to create the matches. For example, if the source package has 1 source and the destination package has 2 destinations, only the first destination will get a route. If the source package has 2 sources and the destination package only has a single destination then only the first source will get applied anywhere.

In the same fashion, the state of virtual routes gets updated automatically as base routes in the packages change. If an audio route changes and the source and/or destination exists in a virtual router, an analysis is done to determine if the current route states of any of the source packages match what would happen if the source were applied to the destination. If they do, the virtual route is updated to reflect that source as the current route. If multiple source packages match the destination, then the first one with the most matches is displayed as the currently routed source.

If no source package matches a destination package, the route may have two other states. The first is “None”, which means that none of the destination package's destinations are assigned to anything. The other state is “Other”, which means that some of the destination package’s destinations have some route applied to them but it does not match up with any of the source packages in the virtual router. This explicitly displays a differentiation between a destination that is cleared and one that might be in use in some fashion, but not one that the virtual router can represent given its existing configuration.

IO Ordering – Move Up/Down and Push Up/Down



The virtual router page also has a block of controls used for manipulating the order of the IOs in the router. These controls are only available to Administrative users. These buttons effectively alter the number of the IO in the virtual router. Since changing the IO numbering of a virtual router can have deeper consequences if the IOs are already in use by Logic Flows or device emulators, these buttons must be enabled by manually selecting the Enable Push/Move checkbox in order to prevent accidental changes.

It is recommended to sort by Port when using this feature as it manipulates up and down based on the port number and not the sort order. This is a design tool to create specific numbering for the virtual IOs and is most useful when those IOs will be exposed to emulated routing protocols that expect a number to be assigned to each IO.

Move Up – moves the IO up in the router effectively decreasing its IO number and swapping with its lower next-door neighbor if necessary.

Move Down – moves the IO down in the router effectively increasing its IO number and swapping with its higher next-door neighbor if necessary.

Push Up - moves the IO up in the router effectively decreasing its IO number and pushing any lower-numbered IOs it runs into up as well decreasing their numbers until it reaches a hole in the numbering or 0.

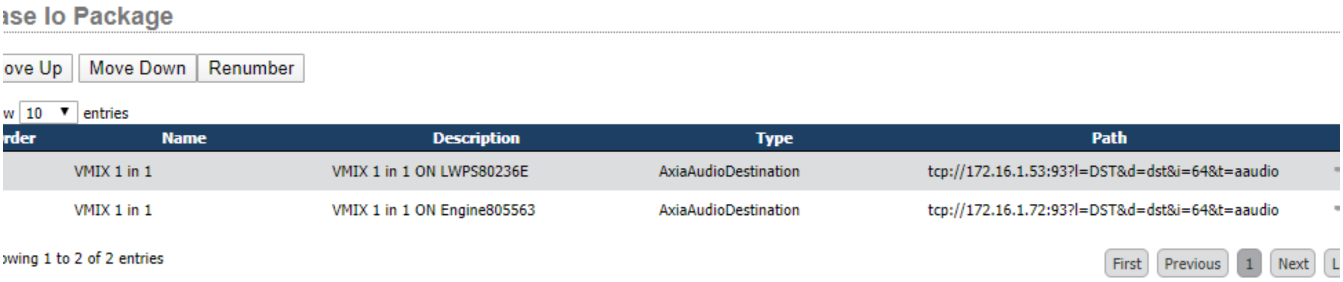
Push Down - moves the IO down in the router effectively increasing its IO number and pushing any higher-numbered IOs it runs into down as well increasing their numbers until it reaches a hole in the numbering or the highest value in the router.

Advanced Base IOs and the Enable Property

Each virtual source and destination's base points in a virtual router also include an enable property which could be true or false. This can be used to disable the base point from being considered during route changes and route state analysis. This property is only available via the API tree in logic flows as it is considered an advanced property.

Here is an example of where this feature might be useful: Let's say we are mixing audio for an airchain using a vmixer. We might create a virtual router with the virtual mixer destinations and the possible studio sources. However, what if for redundancy's sake we wished to duplicate the settings on a vimixer on a second engine thereby allowing us to feed the air chain off of either engine and take one or the other out of service? To accomplish this, we could create a virtual router with destination base points using vmixer channels from both engines and where the source points are the same source base point duplicated.

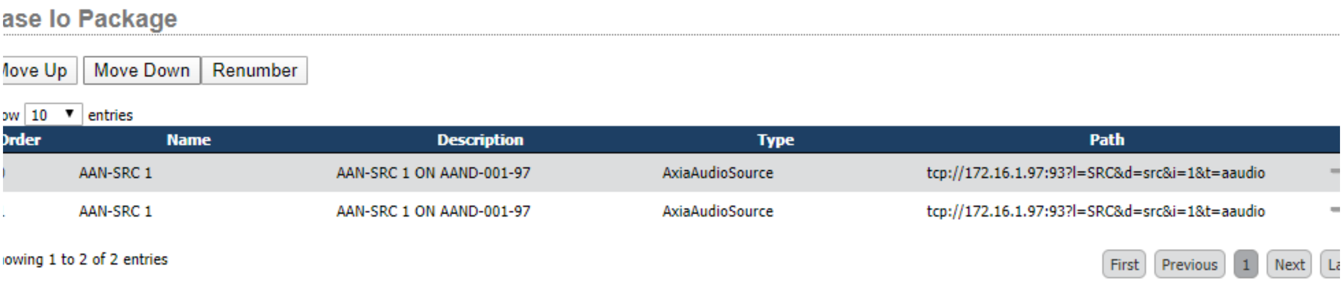
Destination (note the different engine paths):



rder	Name	Description	Type	Path
	VMIX 1 in 1	VMIX 1 in 1 ON LWPS80236E	AxiaAudioDestination	tcp://172.16.1.53:93?l=DST&d=dst&i=64&t=aaudio
	VMIX 1 in 1	VMIX 1 in 1 ON Engine805563	AxiaAudioDestination	tcp://172.16.1.72:93?l=DST&d=dst&i=64&t=aaudio

ɔwing 1 to 2 of 2 entries

Source:

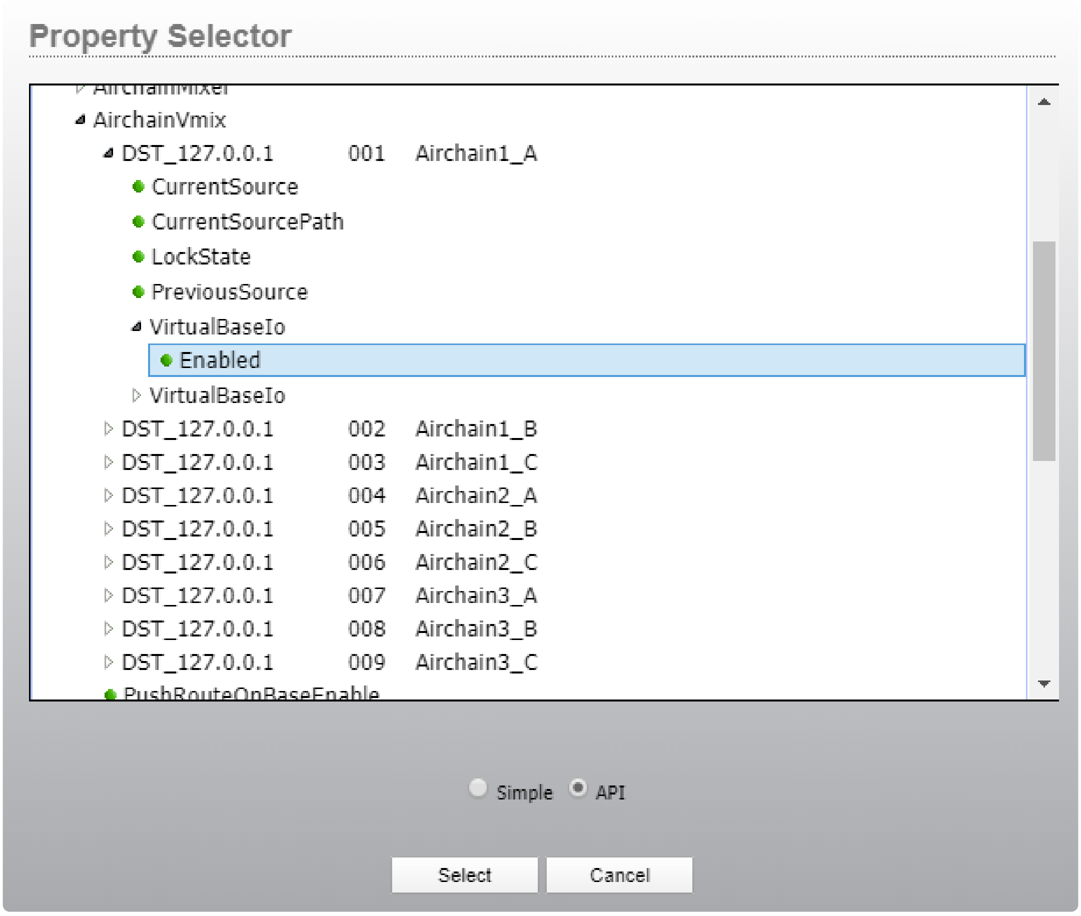


Order	Name	Description	Type	Path
	AAN-SRC 1	AAN-SRC 1 ON AAND-001-97	AxiaAudioSource	tcp://172.16.1.97:93?l=SRC&d=src&i=1&t=aaudio
	AAN-SRC 1	AAN-SRC 1 ON AAND-001-97	AxiaAudioSource	tcp://172.16.1.97:93?l=SRC&d=src&i=1&t=aaudio

ıowing 1 to 2 of 2 entries

If this virtual source was routed to the virtual destination, the same base source would get routed to the vmix destination on both engines, and the route would only show active if it was successfully made on both engines. However, if we wanted to take engine number 2 offline for maintenance, our route changes would still be sent to both engines, but the state would not be displayed since the offline engine did not show the

Using the enable property of the base points, we could create a scene or a logic flow that disabled the destination base points for the second engine while it is in maintenance. The virtual router would continue to work as if those base points did not exist in the destination. Then we could execute a second scene when the engine comes back online to re-enable those base points.



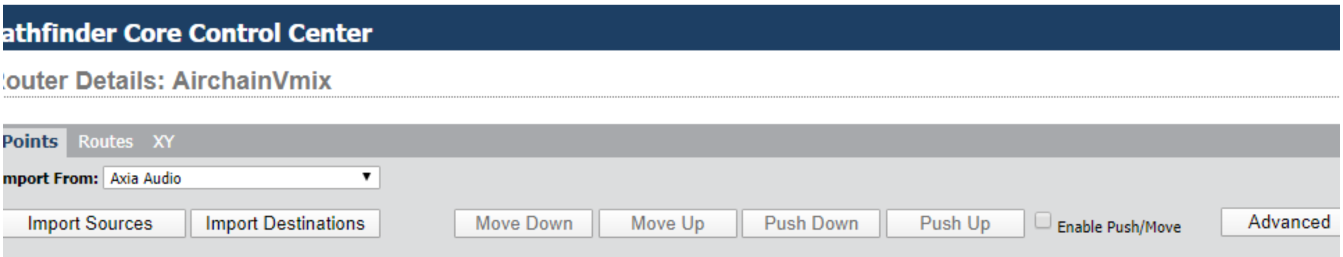
Property Selector

```
▷ AirchainMixer
◢ AirchainVmix
  ◢ DST_127.0.0.1        001   Airchain1_A
      ● CurrentSource
      ● CurrentSourcePath
      ● LockState
      ● PreviousSource
      ◢ VirtualBaseIo
          ● Enabled
      ▷ VirtualBaseIo
  ▷ DST_127.0.0.1        002   Airchain1_B
  ▷ DST_127.0.0.1        003   Airchain1_C
  ▷ DST_127.0.0.1        004   Airchain2_A
  ▷ DST_127.0.0.1        005   Airchain2_B
  ▷ DST_127.0.0.1        006   Airchain2_C
  ▷ DST_127.0.0.1        007   Airchain3_A
  ▷ DST_127.0.0.1        008   Airchain3_B
  ▷ DST_127.0.0.1        009   Airchain3_C
  ● PushRouteOnBaseEnable
```

○ Simple ● API

Advanced: Push Route on Base Io Enable

This feature works with the base Io enable/disable feature above. When viewing a virtual router on the points tab, there is now an Advanced button on the far right side:



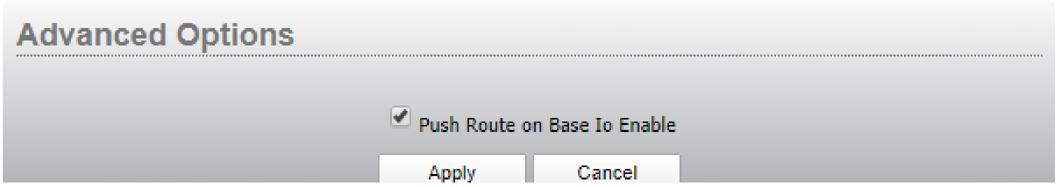
athfinder Core Control Center

:outer Details: AirchainVmix

mport From: Axia Audio

□ Enable Push/Move

Clicking the Advanced button will open a dialog for advanced options. Currently the only option is a checkbox for Push Route on Base Io Enable which by default is off.



Advanced Options

☑ Push Route on Base Io Enable

This option defines what happens when a virtual source or destination's base point was set to enabled=false and changes to enabled=true. If the option is checked, then the existing route state will be re-applied in order to update the state of the base point that was just re-enabled with the current source. If this option is not enabled, no change will be made when the base point is enabled. That is, the route state may change to something like Other since the newly enabled base point does not fit the current source to destination pairing.

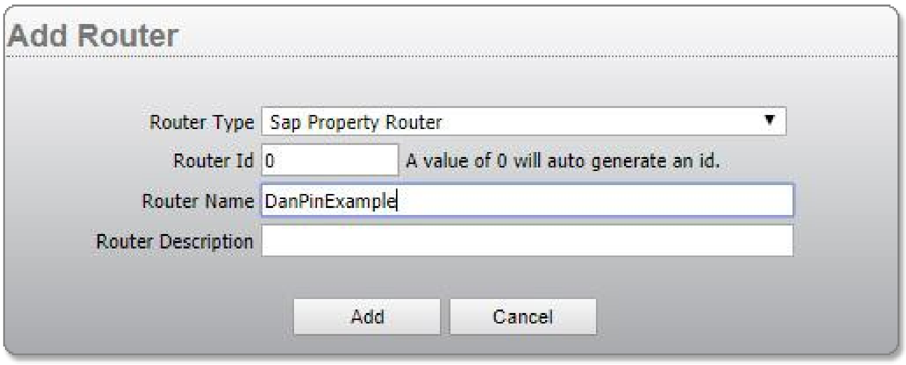
Sap Property Router

By selecting the Sap Property Router type, any property that can be read from in Pathfinder Core PRO can become a router source and any property that can be written to can become a router destination.

For example, in the legacy Pathfinder Pro it was possible to create a router of individual GPIO pins instead of routing entire ports using the GPIOByPin router. This same functionality can be realized with SapPropertyRouter in Pathfinder Core PRO.

Creating a Sap Property Router

To create a Sap Property Router, browse to the **Routers** navigation link and click the **Plus** icon to create a new router.



Add Router

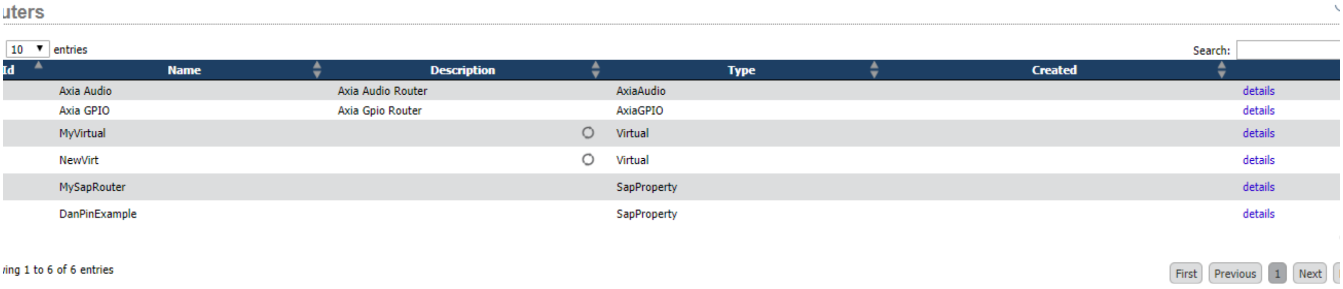
Router Type: Sap Property Router

Router Id: 0 A value of 0 will auto generate an id.

Router Name: DanPinExample

Router Description:

Select the Sap Property Router from the drop-down list of available Router Types. Click **Add** to create the new router.



uters

10 entries

Search:

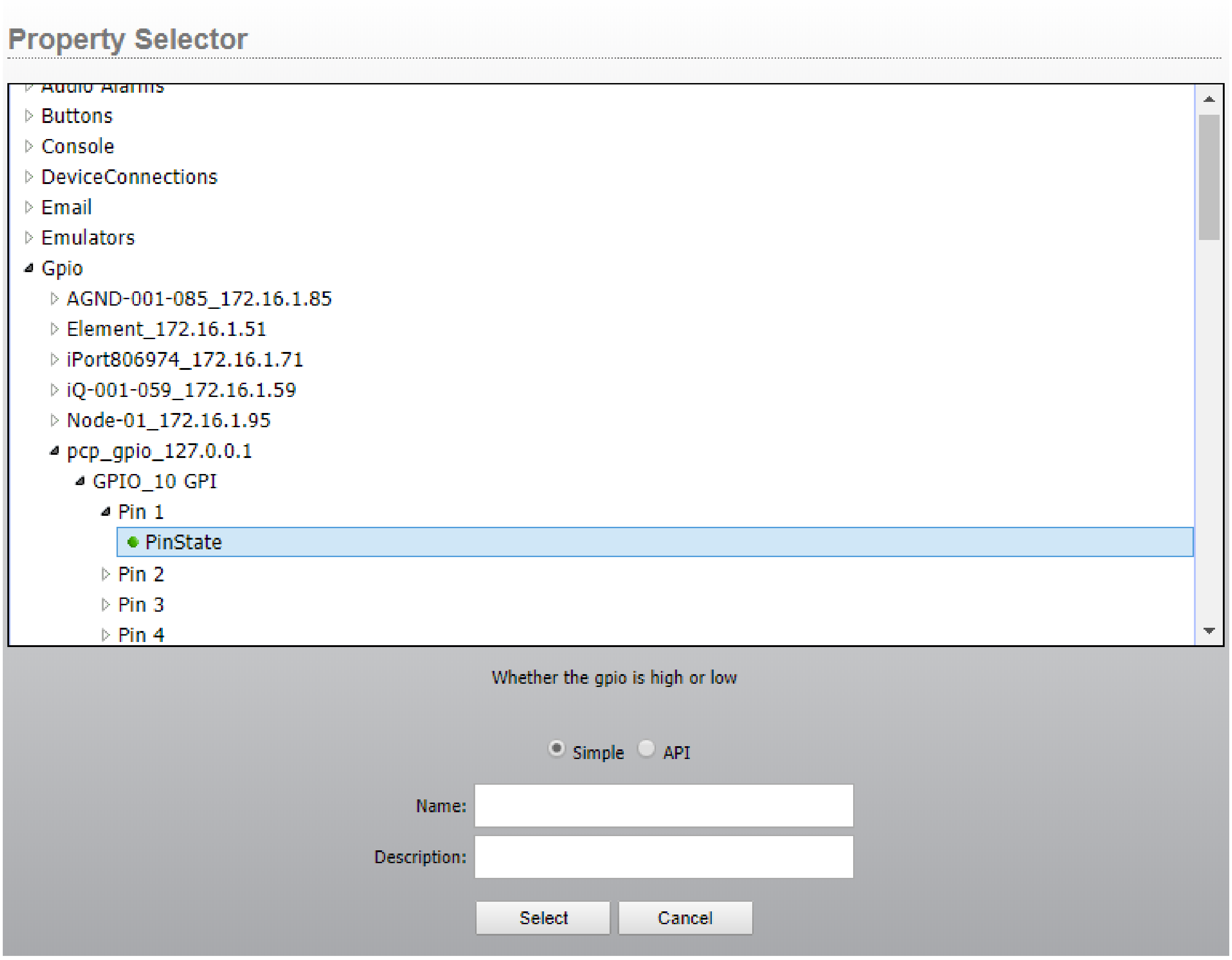
Id	Name	Description	Type	Created	
	Axia Audio	Axia Audio Router	AxiaAudio		details
	Axia GPIO	Axia Gpio Router	AxiaGPIO		details
	MyVirtual	○	Virtual		details
	NewVirt	○	Virtual		details
	MySapRouter		SapProperty		details
	DanPinExample		SapProperty		details

ving 1 to 6 of 6 entries

Adding IOs

To add IOs to your new router, click the **Details** link and then click the **Points** tab.

Like the virtual router, you will see Import Sources and Import Destinations buttons as well as a Translation button. However, clicking these buttons for this router will display the property selector used in Logic Flows.

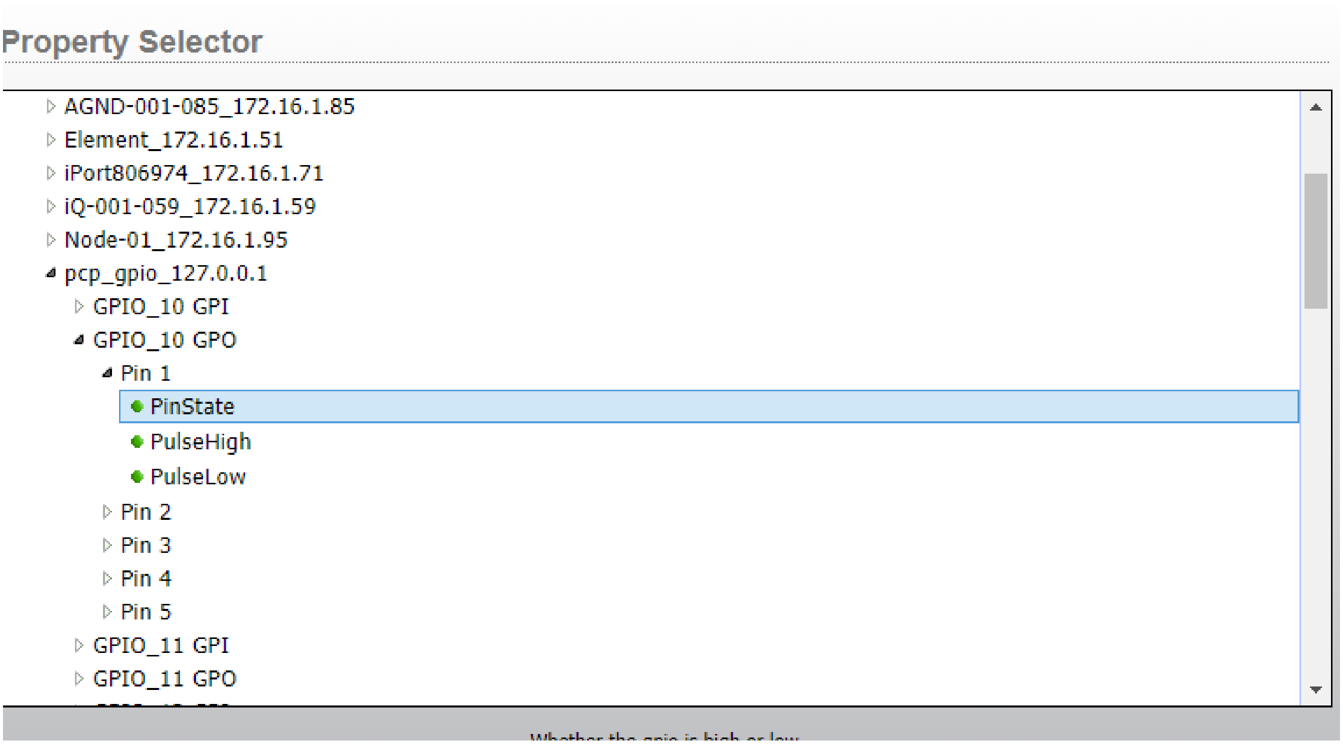


Whether the gpio is high or low

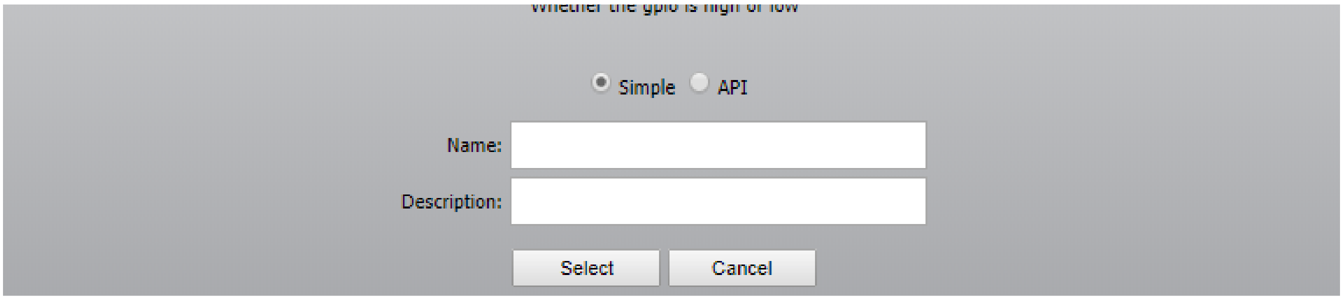
● Simple ○ API

In the example above, we are going to select a GPI PinState property.

Type a name and description for the IO (or use the default) and click **Select**. Repeat for other pins you want as sources, then import the destinations.



Whether the gpio is high or low



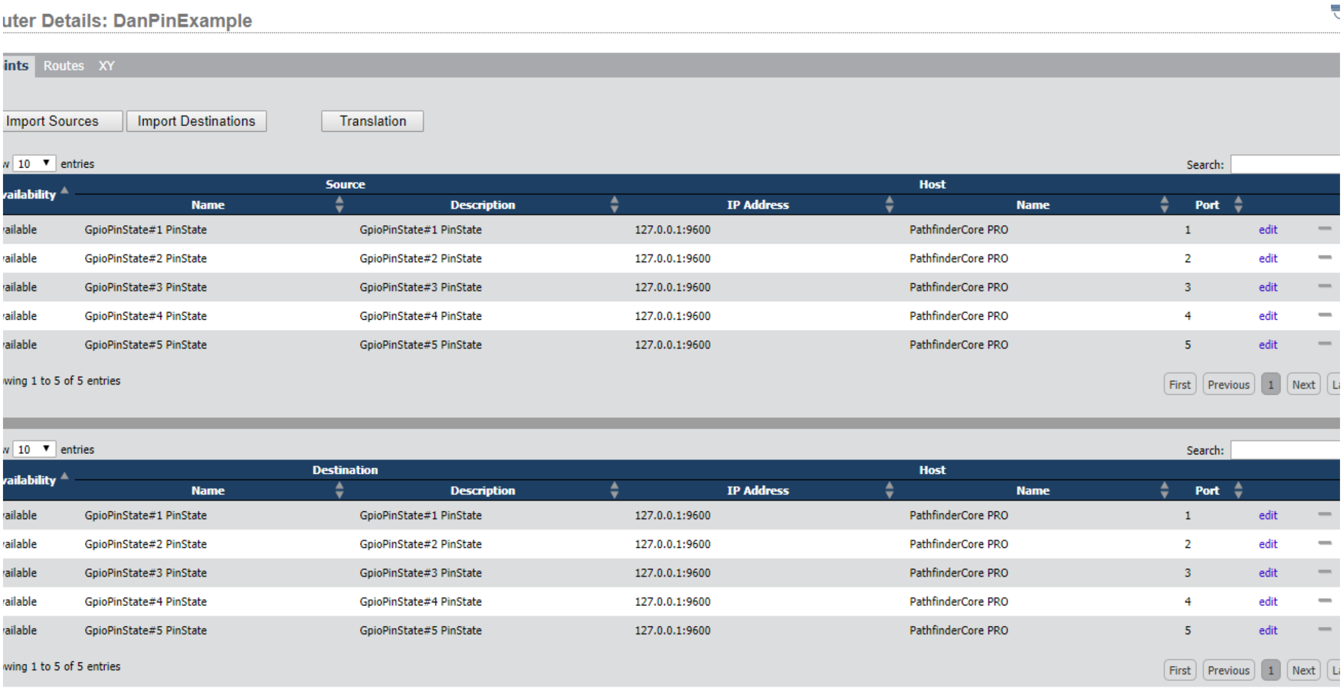
Whether the gpio is high or low

● Simple ○ API

Name:

Description:

Note that if the ports you are using for destinations are on a Livewire driver or Pathfinder Core PRO GPIO node, you can use GPI pins as the destination and route GPI to GPI. When you are done, it might look something like this:



uter Details: DanPinExample

w 10 entries Search:

	Source		Host				
vailability	Name	Description	IP Address	Name	Port		
ailable	GpioPinState#1 PinState	GpioPinState#1 PinState	127.0.0.1:9600	PathfinderCore PRO	1	edit	—
ailable	GpioPinState#2 PinState	GpioPinState#2 PinState	127.0.0.1:9600	PathfinderCore PRO	2	edit	—
ailable	GpioPinState#3 PinState	GpioPinState#3 PinState	127.0.0.1:9600	PathfinderCore PRO	3	edit	—
ailable	GpioPinState#4 PinState	GpioPinState#4 PinState	127.0.0.1:9600	PathfinderCore PRO	4	edit	—
ailable	GpioPinState#5 PinState	GpioPinState#5 PinState	127.0.0.1:9600	PathfinderCore PRO	5	edit	—

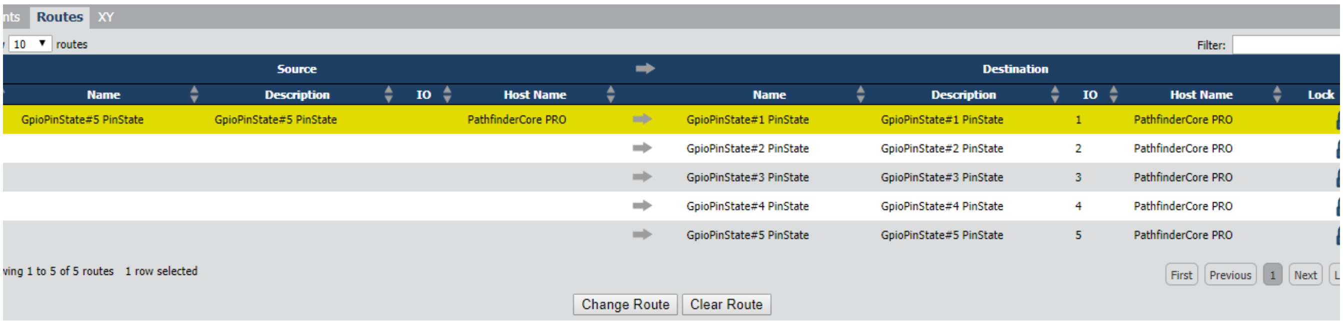
wing 1 to 5 of 5 entries

w 10 entries Search:

	Destination		Host				
vailability	Name	Description	IP Address	Name	Port		
ailable	GpioPinState#1 PinState	GpioPinState#1 PinState	127.0.0.1:9600	PathfinderCore PRO	1	edit	—
ailable	GpioPinState#2 PinState	GpioPinState#2 PinState	127.0.0.1:9600	PathfinderCore PRO	2	edit	—
ailable	GpioPinState#3 PinState	GpioPinState#3 PinState	127.0.0.1:9600	PathfinderCore PRO	3	edit	—
ailable	GpioPinState#4 PinState	GpioPinState#4 PinState	127.0.0.1:9600	PathfinderCore PRO	4	edit	—
ailable	GpioPinState#5 PinState	GpioPinState#5 PinState	127.0.0.1:9600	PathfinderCore PRO	5	edit	—

wing 1 to 5 of 5 entries

If you go to the **Routes** tab you will see the destinations with a "take table" of the sources where you can make route changes.



10 routes Filter:

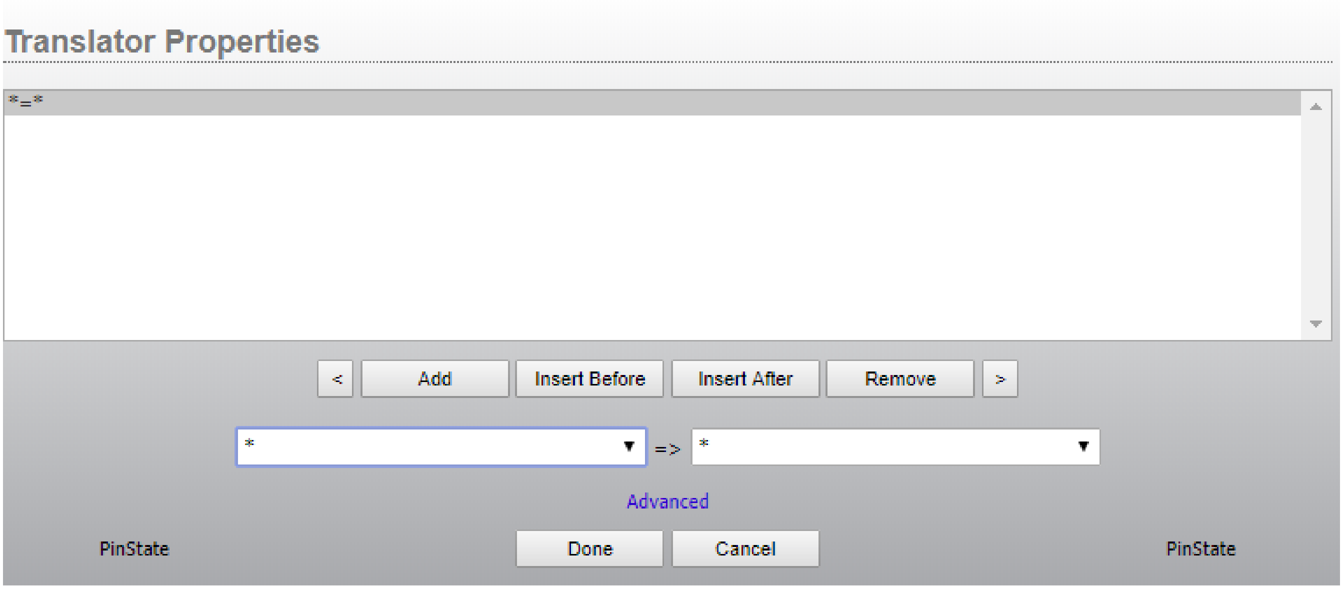
Source				➡	Destination				
Name	Description	IO	Host Name		Name	Description	IO	Host Name	Lock
GpioPinState#5 PinState	GpioPinState#5 PinState		PathfinderCore PRO	➡	GpioPinState#1 PinState	GpioPinState#1 PinState	1	PathfinderCore PRO	
				➡	GpioPinState#2 PinState	GpioPinState#2 PinState	2	PathfinderCore PRO	
				➡	GpioPinState#3 PinState	GpioPinState#3 PinState	3	PathfinderCore PRO	
				➡	GpioPinState#4 PinState	GpioPinState#4 PinState	4	PathfinderCore PRO	
				➡	GpioPinState#5 PinState	GpioPinState#5 PinState	5	PathfinderCore PRO	

wing 1 to 5 of 5 routes 1 row selected

In this example, GPI pin 5 has been routed to GPO pin 1. Therefore, any closures that take place on pin 5 will be replicated on pin 1.

Translation Patterns

Clicking the **Translation** button on the **Points** tab will open the translation pattern used for any values that pass from the source property to the routed destination properties. There is a single translation pattern that is used between the source values and destination values



Translator Properties

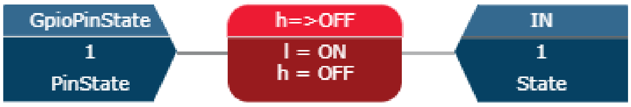
=

* => *

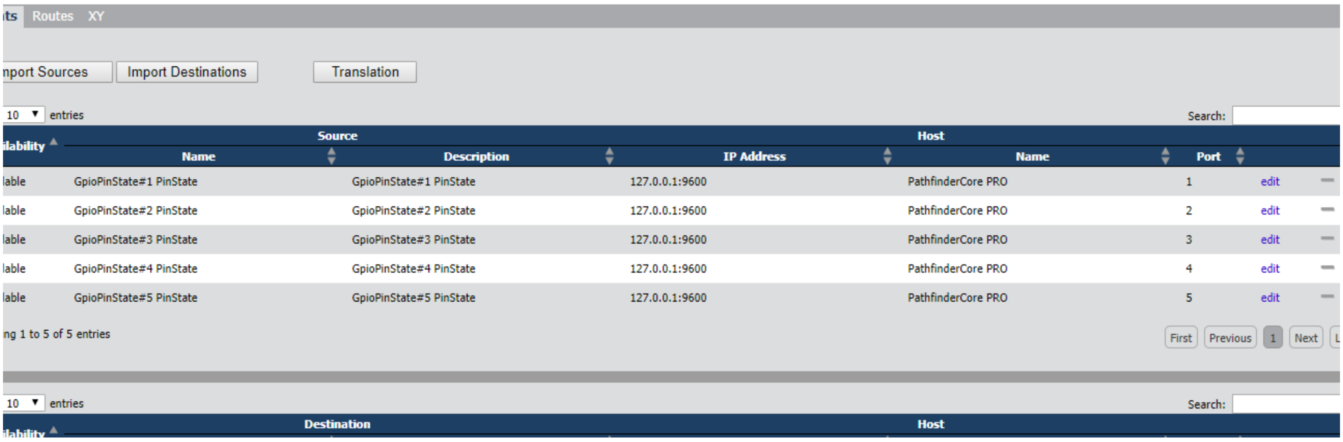
In the example above, the translation pattern is *=*. Using this pattern, a pin low on the source side will cause a pin low on all destinations to which the source is routed. By changing this translation to l=h and h=l, lows on the source would generate highs on any destination to which that source was routed.

Translation patterns may be applied to other properties. Applying a translation pattern of l=5000 to the Pulse state would force the output to go low for 5000ms each time the input went low. Without defining a h= value, nothing would happen when the input went high.

While we could achieve similar results with Logic Flows, configuring a router allows the start point of a flow to change dynamically. For example, suppose we have a Vmixer assigned to each air chain with channel 1 representing the feed from the studio console. There is a GPI pin in the studio which turns the Vmix channel on. This is simple to accomplish with a Logic Flow:



What if we have 5 studios and any one of them could be assigned to the air chain? Using Logic Flows this becomes much more complex because we must build flows for each studio-to-airchain variation. However, this becomes easy with a property router. On the source side of the router we select the GpiPinState property, and on the destination side we select all airchain Vmix IN state properties:



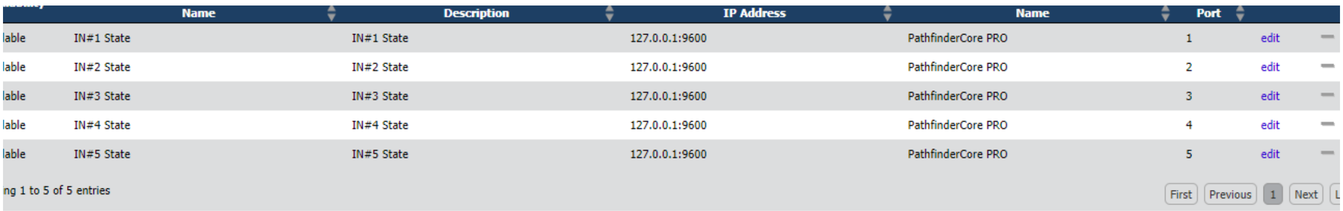
10 entries Search:

	Source			Host			
ilability	Name	Description	IP Address	Name	Port		
lable	GpioPinState#1 PinState	GpioPinState#1 PinState	127.0.0.1:9600	PathfinderCore PRO	1	edit	—
lable	GpioPinState#2 PinState	GpioPinState#2 PinState	127.0.0.1:9600	PathfinderCore PRO	2	edit	—
lable	GpioPinState#3 PinState	GpioPinState#3 PinState	127.0.0.1:9600	PathfinderCore PRO	3	edit	—
lable	GpioPinState#4 PinState	GpioPinState#4 PinState	127.0.0.1:9600	PathfinderCore PRO	4	edit	—
lable	GpioPinState#5 PinState	GpioPinState#5 PinState	127.0.0.1:9600	PathfinderCore PRO	5	edit	—

ng 1 to 5 of 5 entries

10 entries Search:

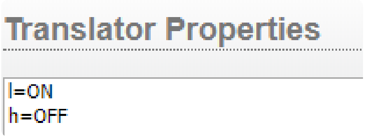
	Destination	Host
ilability		



	Name	Description	IP Address	Name	Port		
lable	IN#1 State	IN#1 State	127.0.0.1:9600	PathfinderCore PRO	1	edit	—
lable	IN#2 State	IN#2 State	127.0.0.1:9600	PathfinderCore PRO	2	edit	—
lable	IN#3 State	IN#3 State	127.0.0.1:9600	PathfinderCore PRO	3	edit	—
lable	IN#4 State	IN#4 State	127.0.0.1:9600	PathfinderCore PRO	4	edit	—
lable	IN#5 State	IN#5 State	127.0.0.1:9600	PathfinderCore PRO	5	edit	—

ng 1 to 5 of 5 entries

Once the items are selected, we can change the translation to:



Translator Properties

```
l=ON
h=OFF
```

Now when we route a studio pin to the airchain Vmix, the low from the studio will control the Vmix state of whichever airchain that studio happens to be routed to.

We can extend this example by using a virtual router where the virtual source package includes the audio console program buss audio source and the pin source from the SapProperty router, and the destination package includes the Vmix channel input audio destination and the Vmix state property from the property router.

These signals get married together such that a single virtual route change will switch both audio and signaling seamlessly. That virtual router can then be controlled by a third-party application using something like Probel without that application needing to know that we are actually switching numerous things under the hood. This opens up a whole new world of routing possibilities.

Another interesting example uses many of the new features described above. What if we wanted to route a TCP stream of song data to different destinations depending on what is currently on the air? We could create generic device emulators for each destination and each song data source. Using the regex capability, we could then add a watcher to the emulators on the source side that looks like:

Regex.Match((?s).*)

This will match any data coming in. Then, we create a new property router and choose the watcher TranslationValues for the soruces. For the destinations, we select the *ToSend* property of the destination generic emulators. This will essentially pass any data that comes into the source emulator to the output of destinations to which that source is routed. Then we could also marry those points into the air chain virtual router such that audio, studio pin to Vmix state, and songdata are all married together as a single virtual route.

Another possibility might be to route memory slots. Or, we could use the duration regex example above to capture different satellite duration data as source points in a router that then gets passed on to countdown clocks.

The point is, you can now route anything Pathfinder knows about.

In each example above, all sources and destinations in a given property router are of the same type. While this is not strictly required, it is usually good practice. Once you start mixing IO property types in the same router’s sources or destinations, the translation table can become non-intuitive and it is possible to output the wrong data to devices.

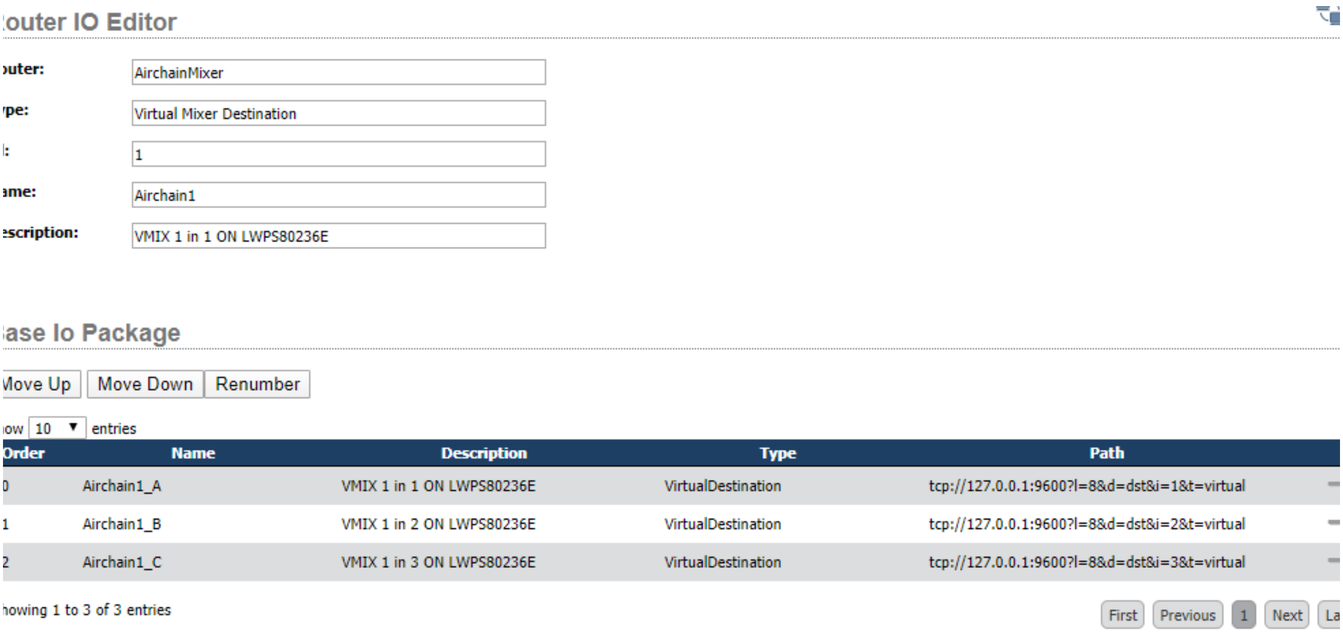
As there is no route state in the equipment for this type of routing, in order to retain the route state between restarts, this router must store all route changes (not the values transitioning through the routes) to the backing storage. In the case of the fanless engine which uses compact flash for this storage, we have implemented some protections to reduce the write cycles, though it is not recommended to make recursive millisecond route changes on Property routers if it can be avoided; for example, a rotation circuit that continuously loops through changing property routes. This is less of a concern with the R2 platform which uses SSD and the VM version.

Virtual Mixing Router

The virtual mixing router can be used to dynamically add and remove sources from a destination that accepts multiple sources like a vmixer. It's usage will be become more clear as we work through an example. To create a virtual mixing router, browse to the routers page and click the plus icon to add the router. From the Router Type, select Virtual Mixing Router, provide a name and description, and click Add.

The virtual mixing router is created and populated in the same way as a virtual router but it behaves differently. A virtual mixing router only allows a single base point for each source and assumes that the base points of a destination mix together. Therefore, you can apply multiple sources to the same destination up to the number of base points you have defined for the destination. Let's see if we can make this clearer through an example. Let's say you have 10 studios and 4 air-chains. However, some of these studios are news rooms which may need to be mixed in with the main studio. In this case you can use a vmixer on an engine to mix these together. And we could use the Virtual Mixing Router to dynamically route studios into the vmixer. To do this we would create the virtual mixing router like this:

Destination:



:outer IO Editor

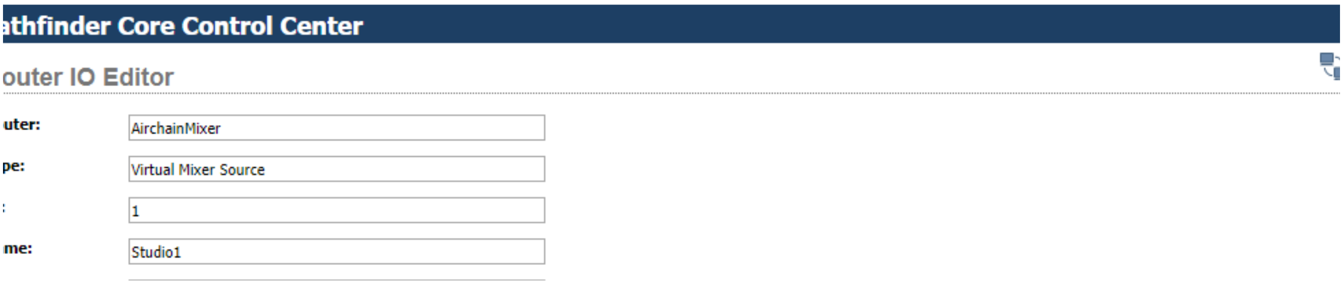
outer:	AirchainMixer
'pe:	Virtual Mixer Destination
l:	1
ame:	Airchain1
escription:	VMIX 1 in 1 ON LWPS80236E

:ase Io Package

Order	Name	Description	Type	Path
0	Airchain1_A	VMIX 1 in 1 ON LWPS80236E	VirtualDestination	tcp://127.0.0.1:9600?l=8&d=dst&i=1&t=virtual
1	Airchain1_B	VMIX 1 in 2 ON LWPS80236E	VirtualDestination	tcp://127.0.0.1:9600?l=8&d=dst&i=2&t=virtual
2	Airchain1_C	VMIX 1 in 3 ON LWPS80236E	VirtualDestination	tcp://127.0.0.1:9600?l=8&d=dst&i=3&t=virtual

howing 1 to 3 of 3 entries

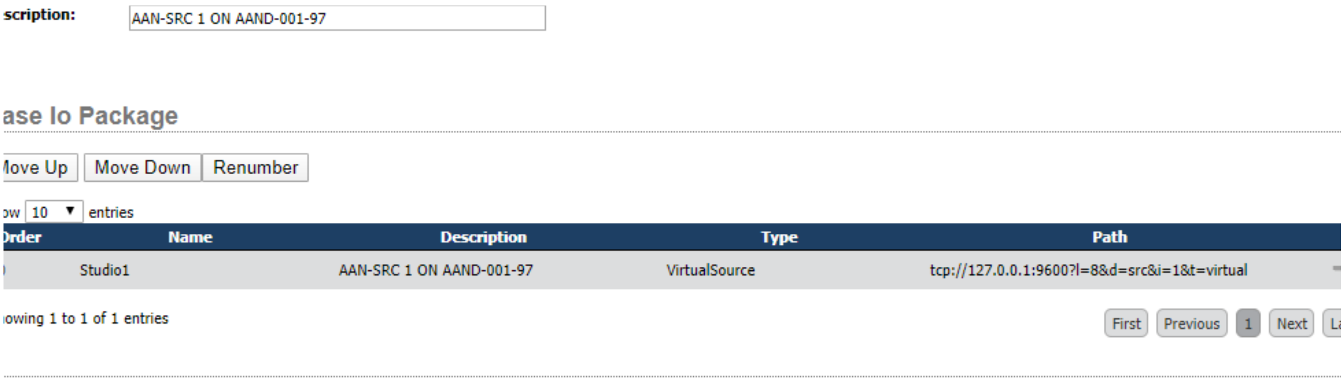
Source:



athfinder Core Control Center

outer IO Editor

uter:	AirchainMixer
pe:	Virtual Mixer Source
:	1
me:	Studio1



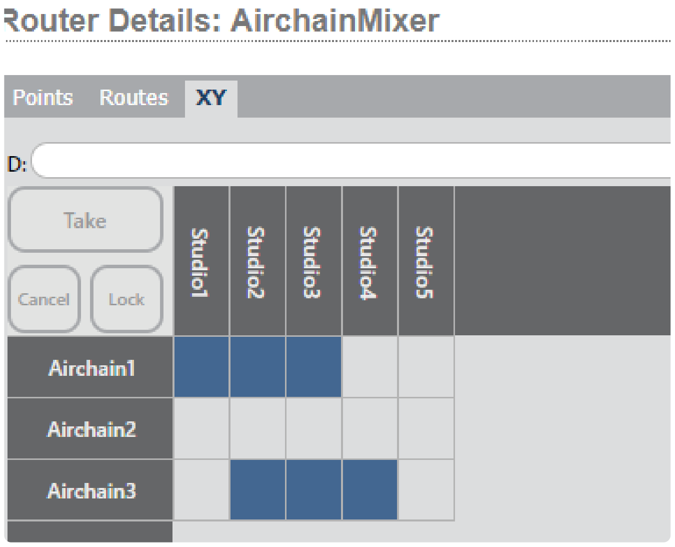
scription: AAN-SRC 1 ON AAND-001-97

ase Io Package

)rder	Name	Description	Type	Path
)	Studio1	AAN-SRC 1 ON AAND-001-97	VirtualSource	tcp://127.0.0.1:9600?l=8&d=src&i=1&t=virtual

iowing 1 to 1 of 1 entries

If we added additional sources and destinations where the destinations were comprised of multiple mixed (vmix) destinations and the source was a single source, we end up with a router that can look like this:



Router Details: AirchainMixer

	Studio1	Studio2	Studio3	Studio4	Studio5
Airchain1	■	■	■		
Airchain2					
Airchain3		■	■	■	

You will notice that the destination AirChain1 actually has three sources assigned to it and since under the hood the destination's base points are in the same vmixer, these are mixed together. It is important to note that PathfinderCore PRO does not actually do any audio mixing. No audio passes through PathfinderCore PRO. Rather you are telling PathfinderCore PRO that these destinations are all a part of a mixer.

Routing a source to a destination in a Virtual Mixing Router is accomplished differently than in a normal router. In a normal router we assign a source to a destination using the CurrentSource or CurrentSourcePath property. With a Virtual Mixing Router we use a different set of properties called AddSource, RemoveSource, and SwapSource.

When you apply a source number to a virtual mixing destination's AddSource Property, the system will review the destination's base points. If the source is already assigned to one of the destination's base points then no change is made. If it is not assigned and all of the base points already have a source assigned to them, then no change is made. If it is not assigned and once of the destination's base points is cleared, then the source will be applied to that base point. Therefore, you can route as many sources to the destination as the destination has base points and they will be dynamically used.

Conversely, when a source number is applied to a destination's RemoveSource property, the system will check to see if that source is applied to any of the destination's base points, and if so it will clear that base point.

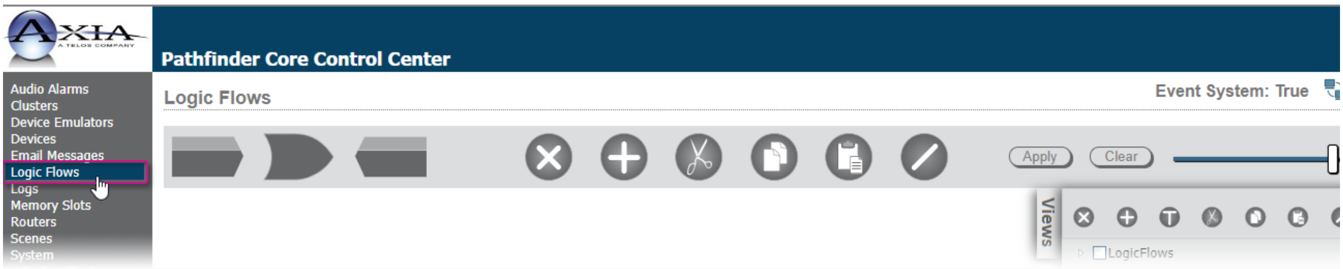
The SwapSource property checks to see if a source is applied to one of the destination's base points and if not, it attempts the same actions as AddSource. If it is applied, then it executes a remove source. This is useful for toggle buttons that add and remove sources from a destination.

While this router's sources can only have a single base point. It is possible to marry points together by putting the base points in a virtual router and then targeting the virtual router as the base points of a virtual mixing router.

Logic Flows

Pathfinder Core PRO's event system is called **Logic Flows**, and it enables information visualization so that you can see how things work within your system.

Click the **Logic Flows** link in the navigation bar to open the Logic Flows user interface.



Each Logic Flow can be thought of as an event. More specifically, a Logic Flow is a set of property translations. During an event, we convert an input property value to an output property value.

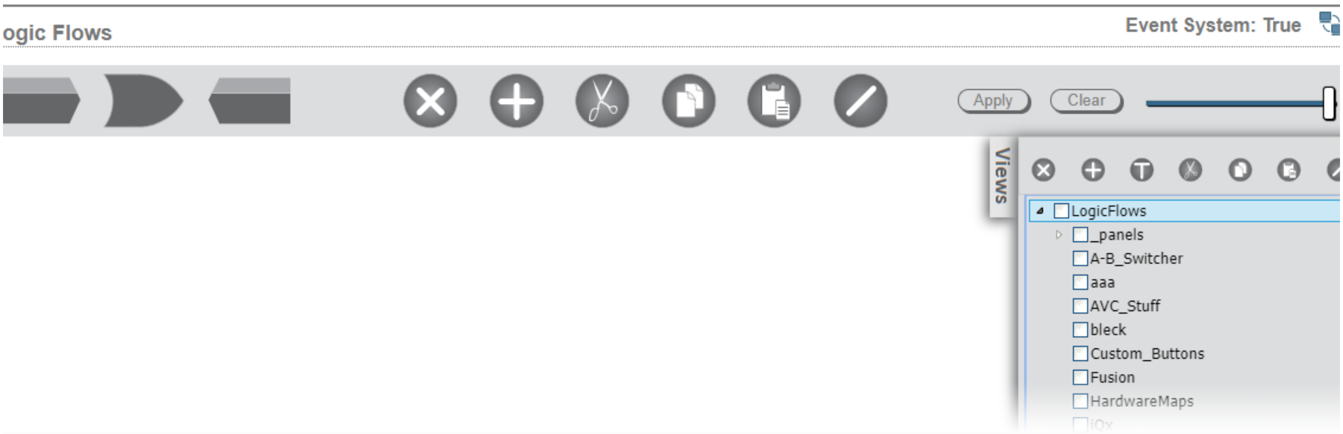
There are four components that may be used to design a Logic Flow:

Name	Description
Start Point	A property in the system being monitored for changes.
End Point	A property in the system which will be changed by the Logic Flow.
Translator	A list of changes that can be applied to the value of a property.
Combiner	A way to logically combine multiple states using logical operators such as AND, OR, NOT, etc.

Using these tools, you can visually build Logic Flows to accomplish just about any task you want.

Organizing Logic Flows with Views

When you first look at the Logic Flows screen you will see an organizational panel on the right called **Views**. Think of Views as folders in your computer's file system - organizational containers into which you can place Flows, allowing you to keep all Flows related to a certain subject in the same place.

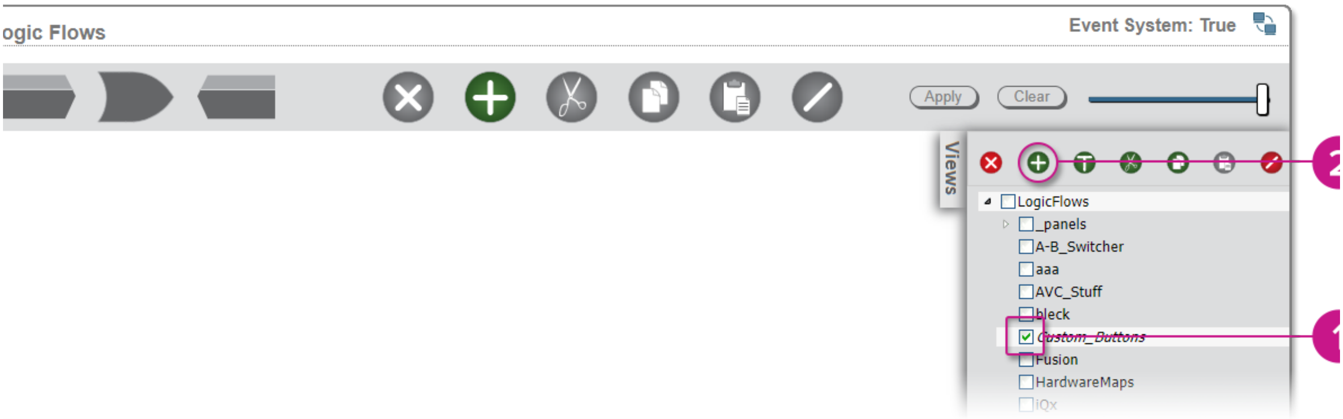


ⓘ Important Note: Clicking the **Views** tab allows the panel to slide in and out.

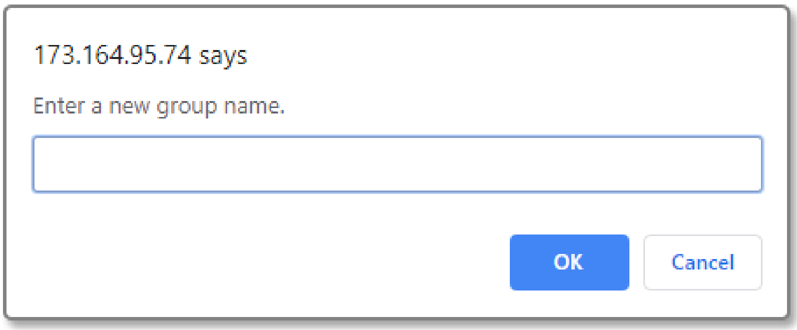
Creating a New View

To create a new View:

1. Check an existing **View** (this will become the *parent* View).
2. Click the **Add (plus)** button in the Views panel.



1. Type a **Name** for the new View and click **OK**.


173.164.95.74 says

Enter a new group name.

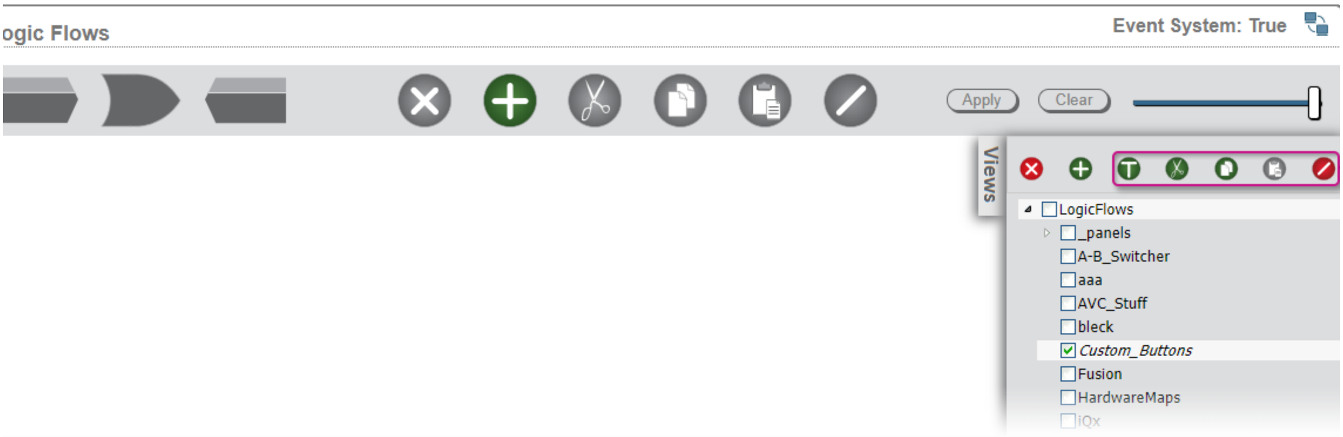
Like your computer folder structure, you can nest Views as deep as you like. While you can add Flows to the master *LogicFlows* View (your “Root” folder), this is not recommended as *LogicFlows* is the one View that you cannot completely delete. Instead, devise a View naming strategy and create additional Views underneath *LogicFlows* to store your Flows.

> ⓘ Important Note: Each View is limited to 100 Flows.

There are two special Views called **HardwareMaps** and **_panels**. These Views are created by the system and used to manage the mapping of software buttons to hardware controls and bindings in HTML5 user panels. This is described in the chapter on HTML5 panels, but it is important to understand that these Views are managed by the system and should not be deleted.

Manipulating Existing Views

With a View selected:



Icon	Function	Description
T	Rename	Clicking this icon gives you the option to Rename the selected View.
✂	Cut	With a single View selected, click this icon to move the selected View to the clipboard.
⧉	Copy	With a single View selected (se *note below*), click this icon to copy the selected View to the clipboard.
📋	Paste	After pasting a *cut* View back into the system, the original View is deleted. When pasting *copied* View back into the system, you will be prompted to give the pasted View a new name.

⊘	State Toggle	Clicking this button will toggle all Logic Flows in the selected Views between *Enabled* and *Disabled* state. The recursion checkbox, if enabled, will also enable or disable flows in sub views if desired.

> ⓘ Important Note: You cannot select multiple Views for copy and paste operations, though if you select a View with child Views, those child Views are included in the copy operation. All pasted Logic Flows will be in a disabled state whether the operation started as a cut or a copy.

It is assumed that after completing a copy and paste you will want to modify the Start and End Points of the Flow to reproduce behavior using different End Points.

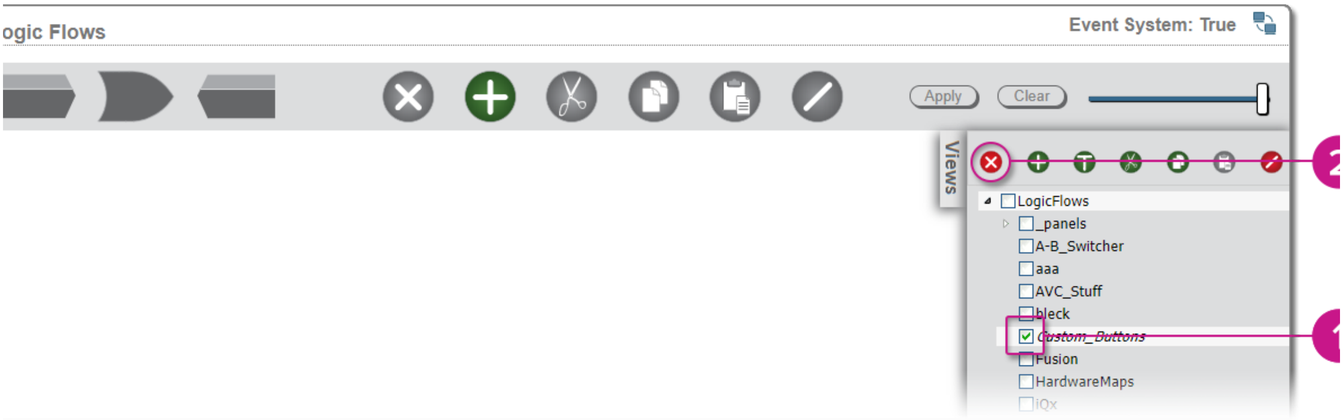
View, copy, and paste operations are completed by the Pathfinder Core PRO appliance rather than by the browser and so they do not require an Apply.

Under the hood, the browser sets a write-only *CopyTo* property on the copied View folder object via a SapV2 *Set* message. The value used in the set message is the path to which the View should be copied. The Pathfinder Core PRO Logic Flow engine then handles the data cloning.

Deleting a View

To delete a View:

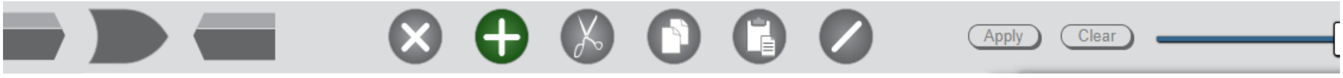
- Select the **View**
- Click the **X** button in the Views panel



- The system will ask for confirmation; all Logic Flows and descendent Views of the View selected for deletion will be deleted as well

Toolbar Controls

From left to right, the Toolbar controls are:



Control	Description
Start Point	Allows you to add a Start Point and a Translator to the left side of a Combiner or to extend the Flow’s start to a new Flow.
Combiner	Allows you to add a Combiner.
End Point	Allows you to add an End Point and a Translator t the right side of a Combiner or to extend the Flow' end into a new Flow.
Delete	Allows you to delete a Flow, Flow Combiner, or Translator.
Add	Creates a new Flow.
Cut	Cuts a Flow to the clipboard.
Copy	Copies a Flow to the clipboard.
Paste	Pastes a Flow from the clipboard.
Disable/Enable	Disables or Enables a Flow, Combiner, or Translator.
Apply Button	Applies pending changes to the system.
Clear Button	Clears pending changes and reverts to the system’s actual state.
Size Slider	Increases or decreases the size of the Flow objec in order to display more items on one screen.

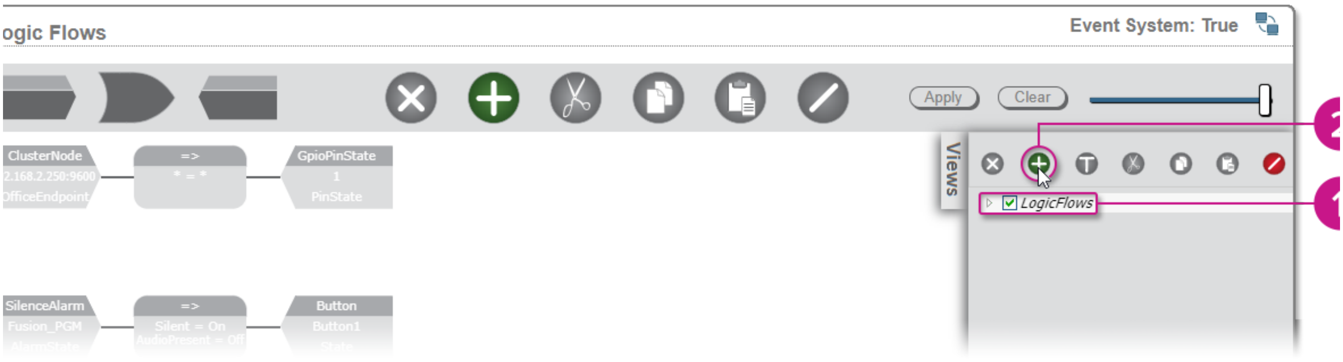
Toolbar controls will be enabled or disabled depending on whether the control is currently valid. For example, the *Cut* icon will not be enabled if you do not have a Flow selected that can be cut.

First Flow

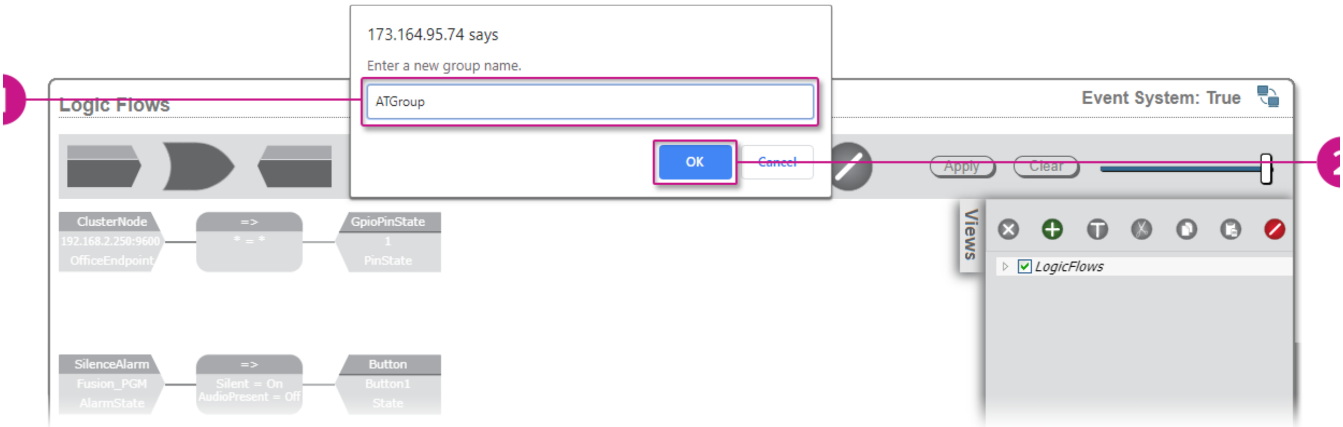
Understanding how these tools are used will be easier if we walk through creating a Flow.

Create a New View

1. Click the **Logic Flows** checkbox then click the **Plus icon** in the *Views* panel



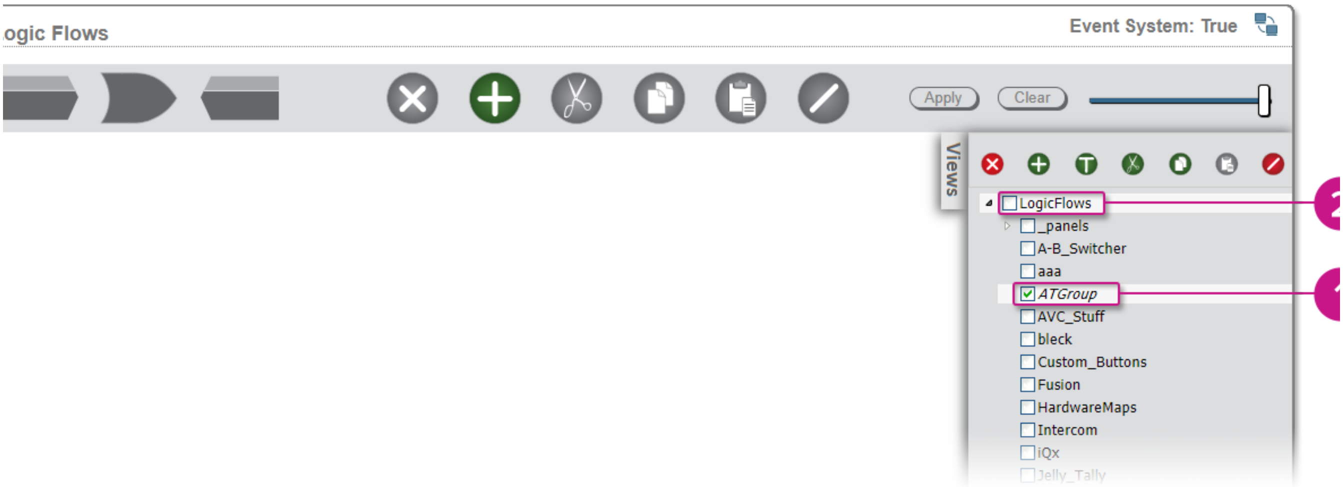
- Type a **name** for the new View and click **OK**


173.164.95.74 says

Enter a new group name.

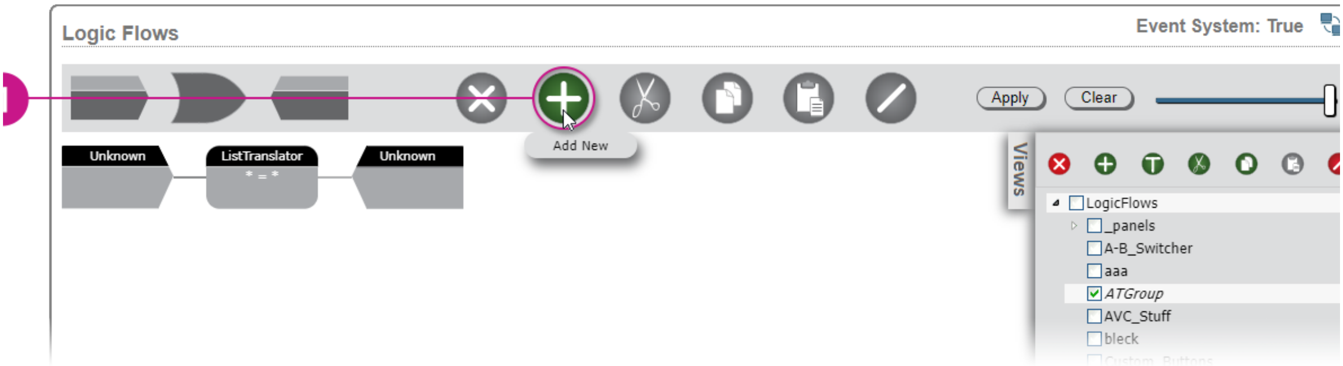
ATGroup

- You can select "multiple Views" to view the Flows from multiple Views together on the same screen; in order to create new Flows, you must select a single View since the system has to know into which View to place the new Flow; click the newly created View's **checkbox** to select the new View; click the root Logic Flows **View** to uncheck its **checkbox** so only your new View is selected



- Once a single View is selected, the **Add** button (plus) in the main control panel for Logic Flows will switch from its disabled grey state to green; click the **Add** button to create a new Flow consisting of a

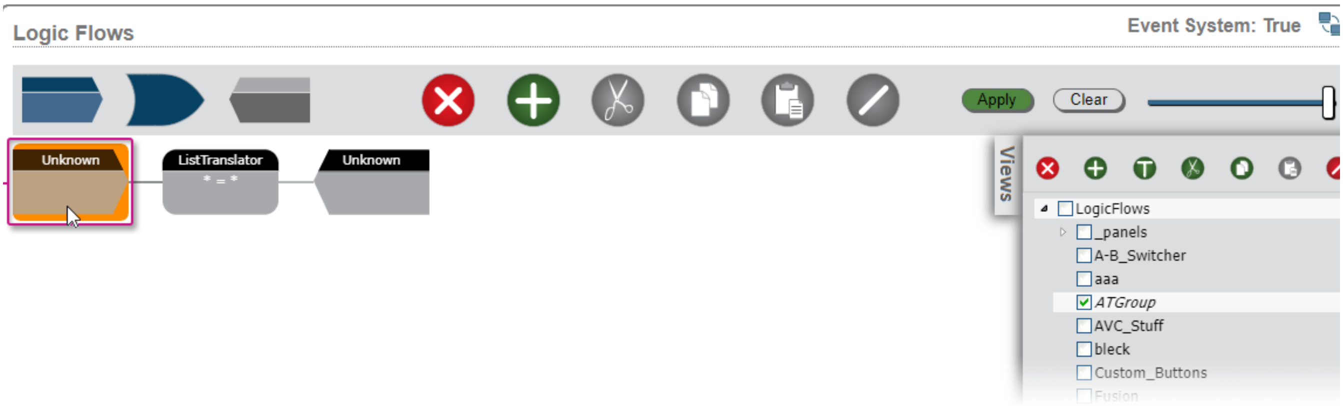
Start Point, a Translator, and an End Point



Important Note: This Flow is in black and gray indicating that these are pending changes that have not been submitted to the system yet. So far, they only live in your local browser.

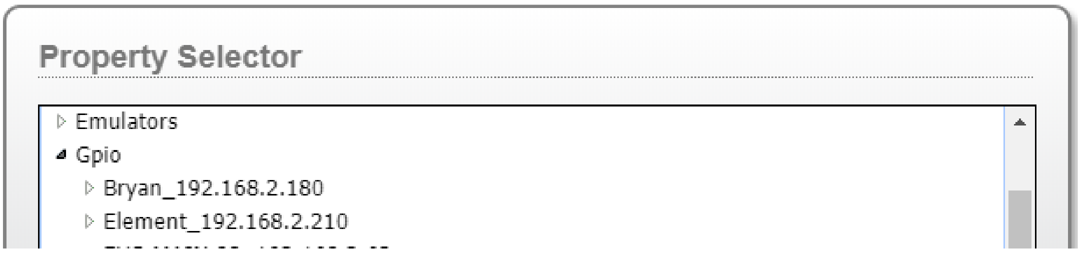
All Flows must have at least one Start Point, one Translator, and one End Point, which is the simplest Flow you can creat. However, we have not yet set parameters to define what this Flow will do. You can define the options for each block in this Flow by double-clicking on the block, though double-clicking on the Translator itself will not do anything until the Start Point and End Point are defined. The reason for this will be explained shortly.

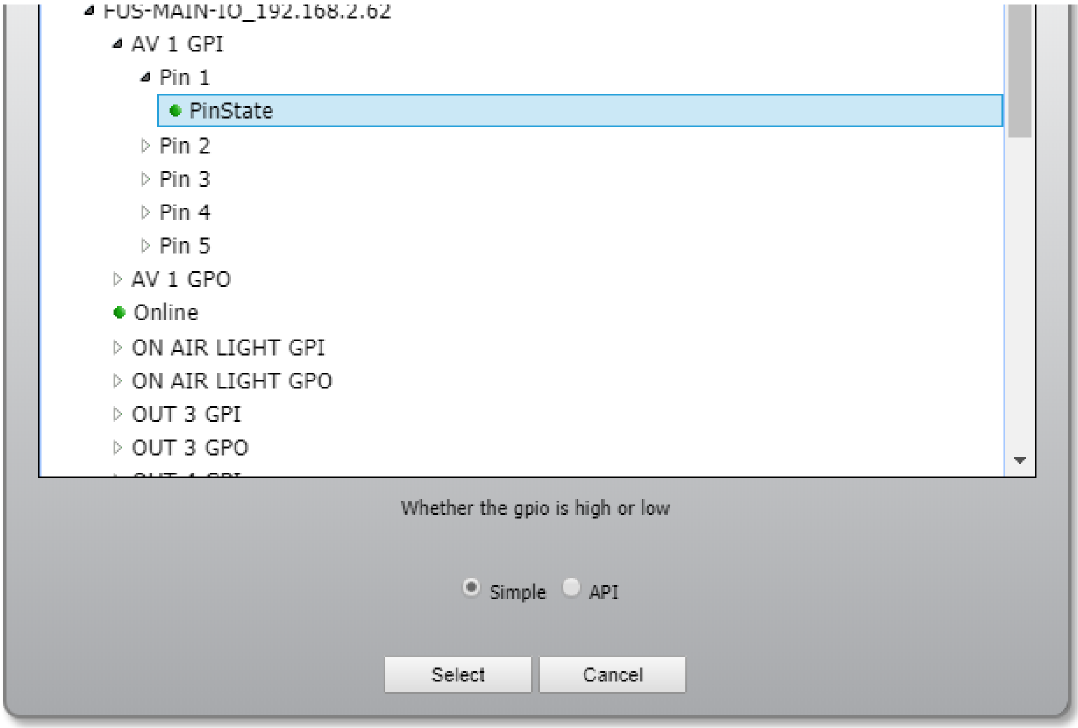
Double-click on the **Start Point block;** its background will turn orange on the first click to indicate that it is selected.



The second click opens the property selector dialog. This dialog is used to select a property in the system that the new Flow will monitor for changes.

Use the arrows to expand and navigate into the objects in the system until you find the object and property you want to select. Properties will have a green dot next to them as opposed to the expansion arrow.





◢ FUS-MAIN-IO_192.168.2.62
◢ AV 1 GPI
◢ Pin 1
● PinState
▷ Pin 2
▷ Pin 3
▷ Pin 4
▷ Pin 5
▷ AV 1 GPO
● Online
▷ ON AIR LIGHT GPI
▷ ON AIR LIGHT GPO
▷ OUT 3 GPI
▷ OUT 3 GPO

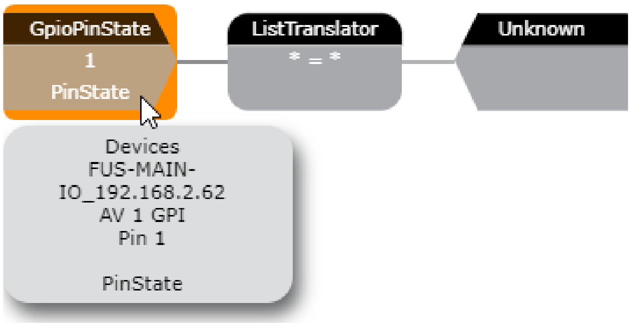
Whether the gpio is high or low

● Simple ○ API

You can either click the property you want to monitor and then click **Select**, or just double-click the **property** to select it.

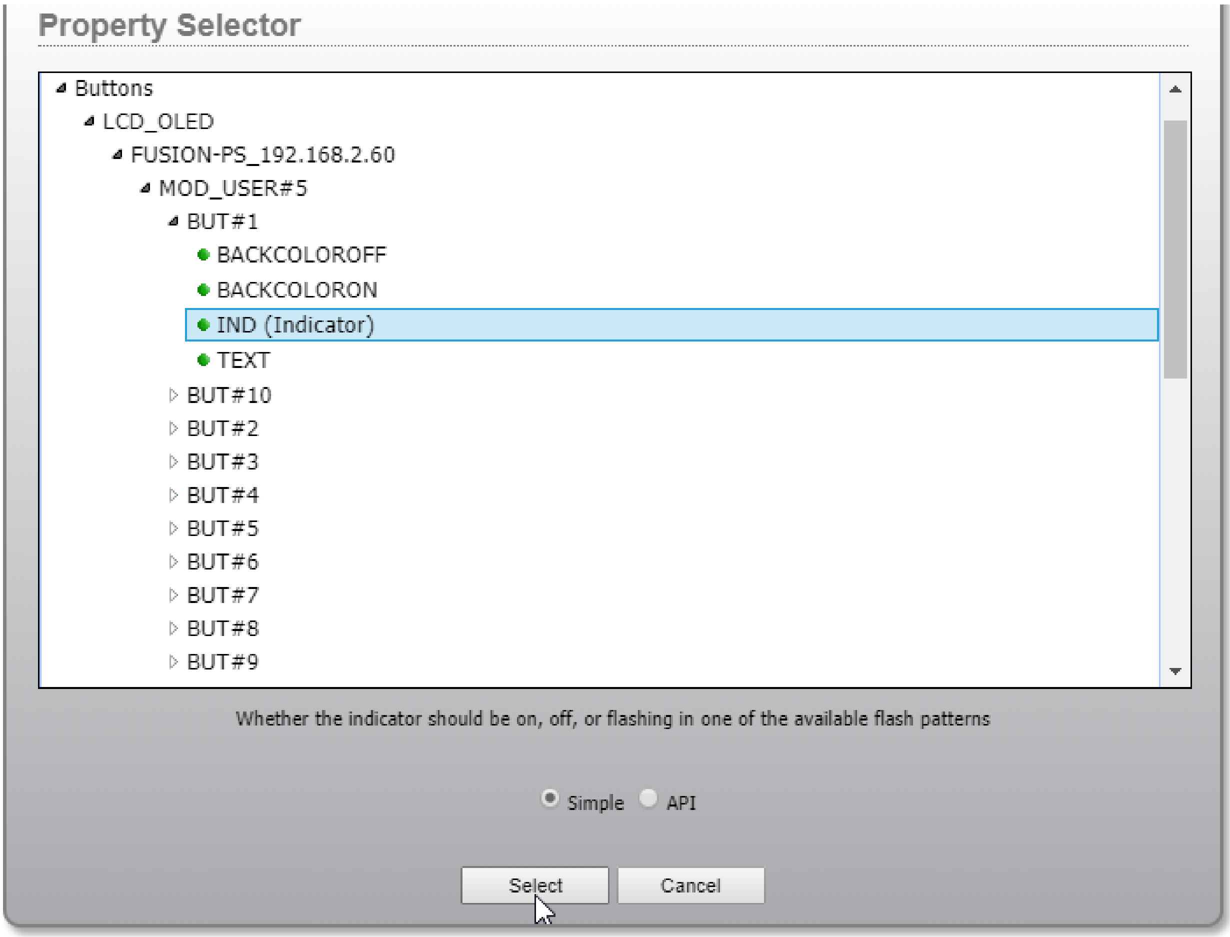
In this example, we are selecting the *PinState* property of GPI#1, Pin#1 on the Fusion device at 192.168.2.62. Notice that as you select properties in this dialog, a description of the property will usually appear underneath the tree window providing more detail about the property's purpose.

> ⓘ Important Note: The property selector dialog has a simple list and an API list. The API list is only recommended for advanced users and so will not be discussed until later in the documentation.



> ⓘ Important Note: Hovering over the Start Point that has been assigned a property will cause a pop-up hover box to appear with more detailed information about the selected property.

- Double-click on the **End Point block** and repeat the procedure to select the End Point or property we want this Flow to change; in this example, we are going to pick an LCD button, but you could also pick a GPO in order to make the GPO state follow the GPI state



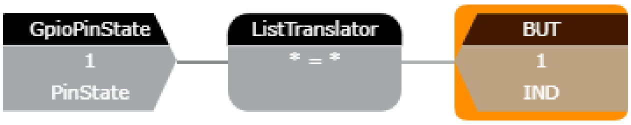
Property Selector

- Buttons
  - LCD_OLED
    - FUSION-PS_192.168.2.60
      - MOD_USER#5
        - BUT#1
          - BACKCOLOROFF
          - BACKCOLORON
          - IND (Indicator)
          - TEXT
        - BUT#10
        - BUT#2
        - BUT#3
        - BUT#4
        - BUT#5
        - BUT#6
        - BUT#7
        - BUT#8
        - BUT#9

Whether the indicator should be on, off, or flashing in one of the available flash patterns

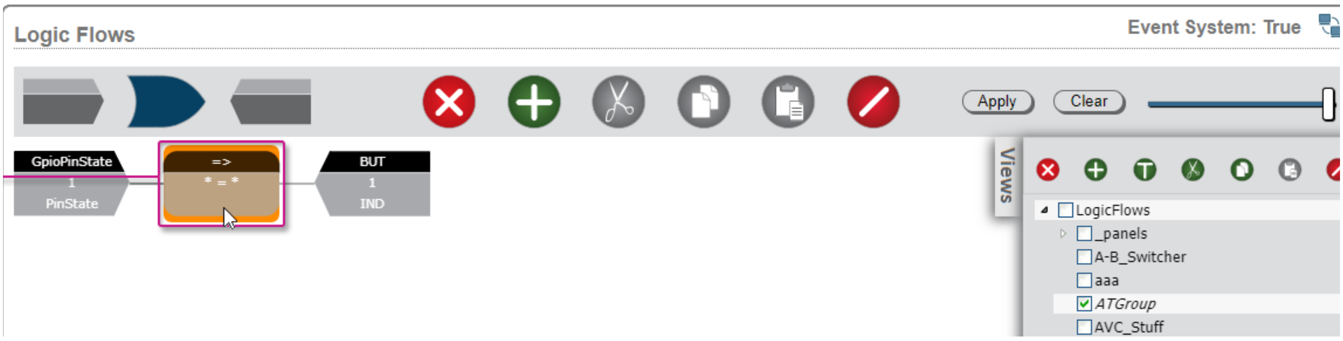
● Simple ○ API

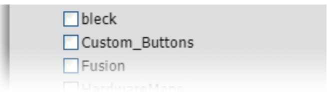
At this point, the Flow looks something like:



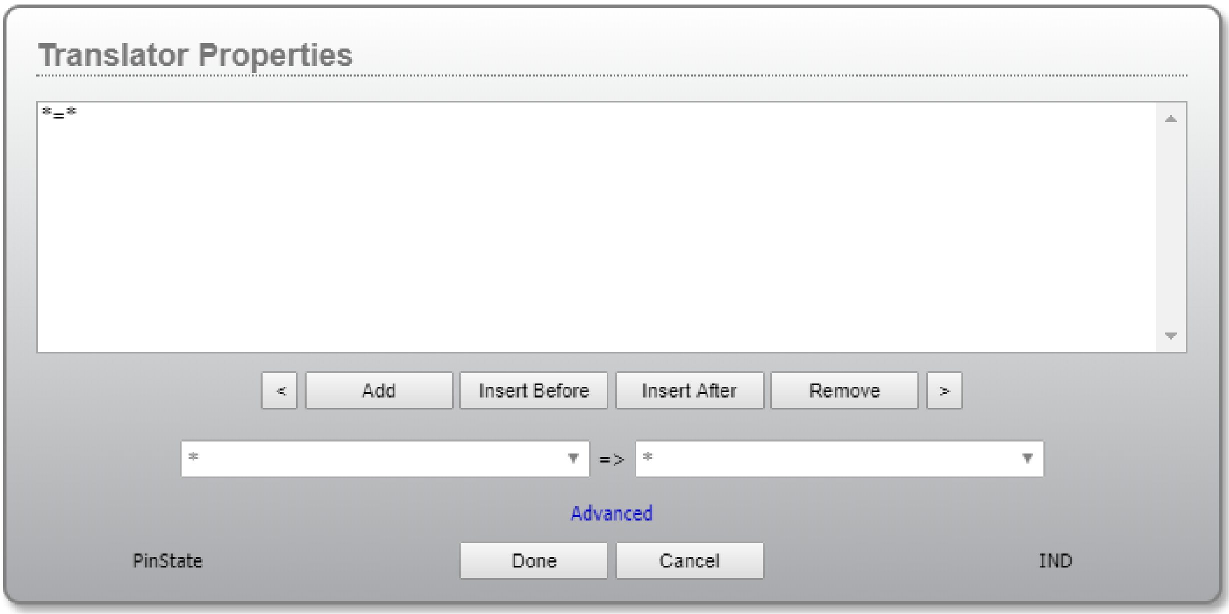
> ⓘ Important Note: The list of available properties differs depending on whether you are editing a Start Point or an End Point. This is because the system is intelligent enough to know if a property is *read only*, *read write*, or *write only*. Properties that are *read only* may not be used as an End Point (something you want to change). The system will only display properties that make sense for the type of block you are editing.

- With configured Start and End Points, you can edit the translation table by double-clicking the **ListTranslator block**

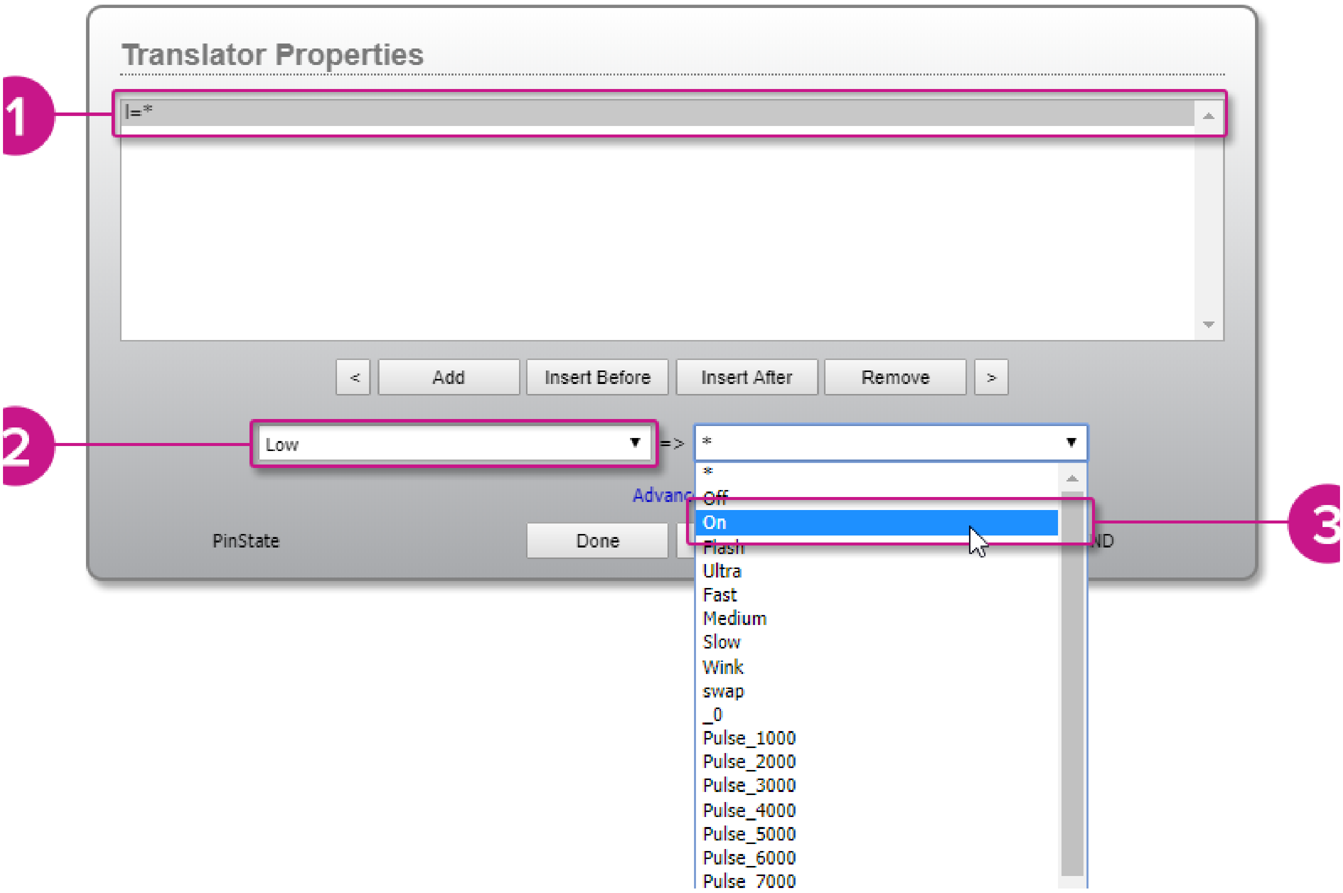


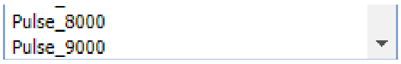


The dialog that opens allows you to enter a list of translations you want to execute as part of this Logic Flow. The other buttons on the translation list screen can be used to move translation definitions up and down in the list, insert new ones, and remove them.

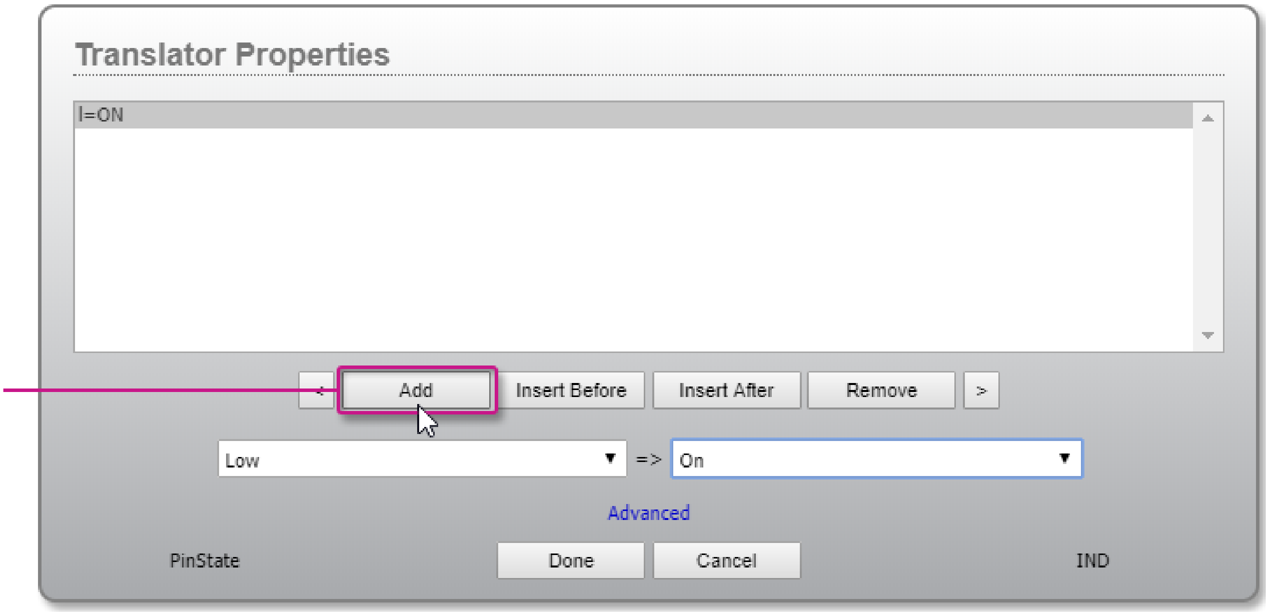


- Select the top translation point (***=***) to enable the drop-down boxes below; the left drop-down has *Low* and *High* as options, and the right drop-down has a list of indicator states; for this example, select ***Low*** in the left drop-down and ***On*** in the right drop-down

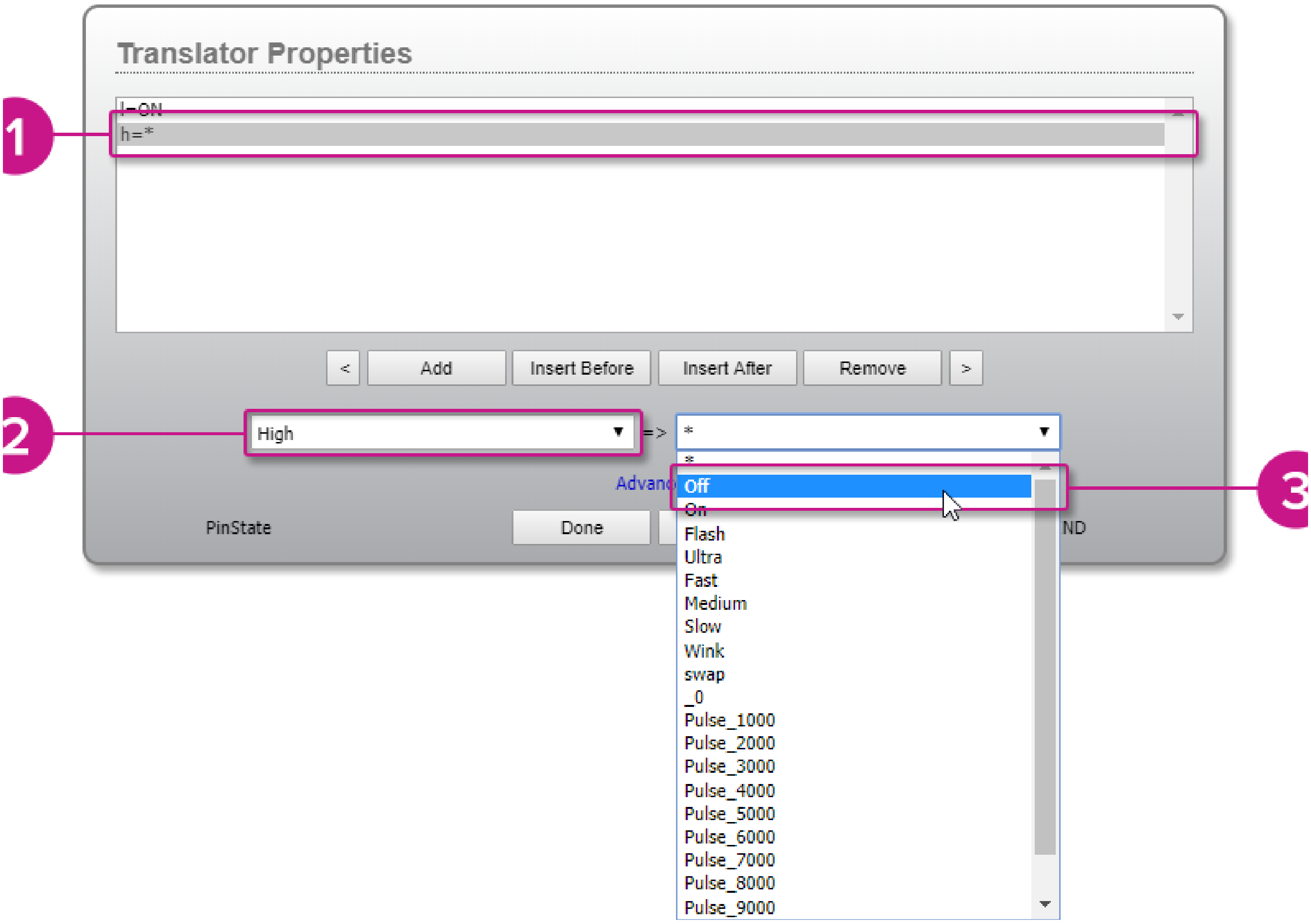




- Click **Add** to add another translation to the list

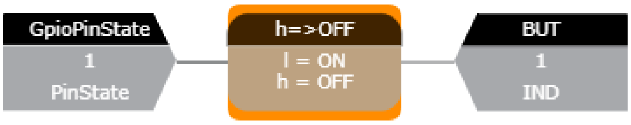


- For this example, select ***High*** in the left drop-down and ***Off*** in the right drop-down



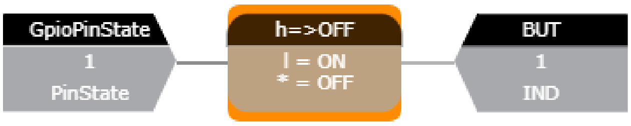
You cannot edit the translation block until the Start and End Point properties are configured because the system will filter the list of possible options depending on the property selected. For example, in this case, *Low* and *High* were available options on the left, and indicator states were available on the right. If the Start Point had been the *CurrentSource* property of a destination, the right drop-down would present a list of available router sources.

- Your Flow should now look something like this:



What we have just defined is that if the Start Point (GPI) is *Low*, then we are going to set the LCD Button indicator on the button to *On*. If the GPI is *High*, then we are going to set the LCD button indicator to *Off*. We are essentially defining a list of rules as to what the End Point value should be depending on the Start Point property value.

When creating translation items, the asterisk serves as a special wildcard that will be evaluated as "any" value. For example, we could have made the second entry in this translation list *=OFF.



Since the list is analyzed in order, this would mean that if the GPI is *Low*, then the indicator should be turned *On*. If the GPI is anything else other than *Low,* then the indicator should be turned *OFF.* In this case, using * or explicitly defining the *High* are equivalent because the GPI only has two possible states.

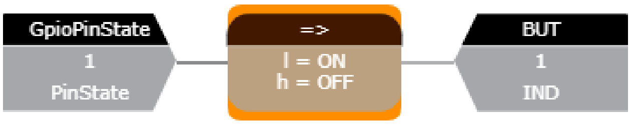
Where the * comes in useful is in situations where there are multiple options on the Start Point side. For example, if the Start Point was the current source routed to a destination, we could build the translation such that if a specific source is selected the indicator would be *On*, and then use the * to say if there is any other source routed, turn it *Off*.

Order is important. If you were to move ***=Off** to the top of the list, the indicator would always be *Off* because the first item in the list says if anything, then *Off*, and the system would never analyze any of the other items in the list.

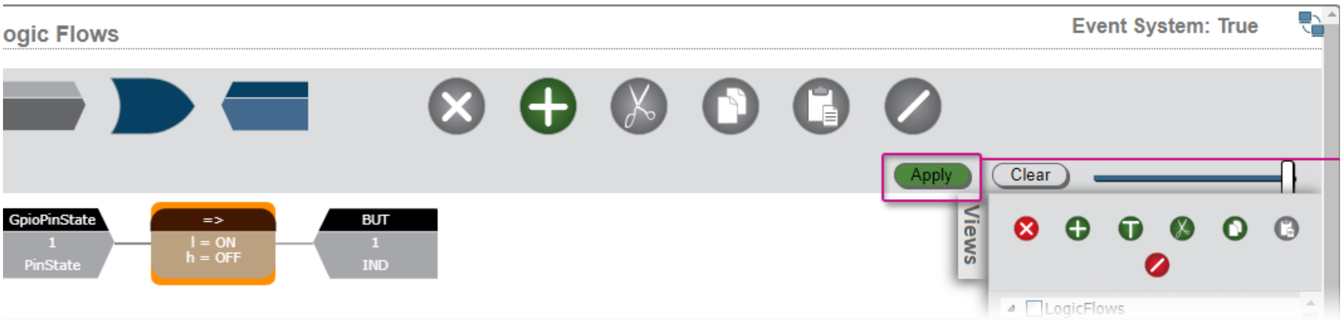
Generally, a * should almost always appear at the end of a list of translations as a catchall. If there is a * on the right-hand side of the translation it means to pass the value from the left on unchanged. So ***=*** would pass anything from the Start Point to the End Point (or next step in the Flow) unchanged.

For those of you familiar with programming languages, this list works very much like a *select case* or *switch* statement. When the Start Point property changes, the value is analyzed in order through this list until a match is found and then the correct value is applied to the End Point or next step in the Flow. If no match is found, the End Point is not changed.

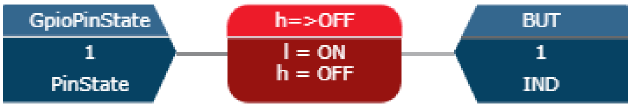
- Once you are finished, click **Done**. The Flow will now look something like this:



- This Flow is not active yet, and the black and gray states are showing us that this Flow has not been committed to Pathfinder Core PRO; in the top toolbar, the flashing **Apply** button indicates changes are pending; click **Apply** to commit the changes to Pathfinder Core PRO

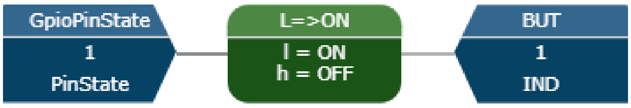


The Flow will now change colors to indicate that it is a live and active Flow:

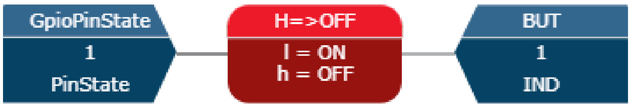


Dynamic Data

The Flow diagram does not simply show the design of the Logic Flows in your system. It also shows the live up-to-the-second state of each of the Flows. Using the Flow example created above, if we trip the source GPI to *Low*, we will see the diagram change to:



And if we switch it to *High*, it will change to:



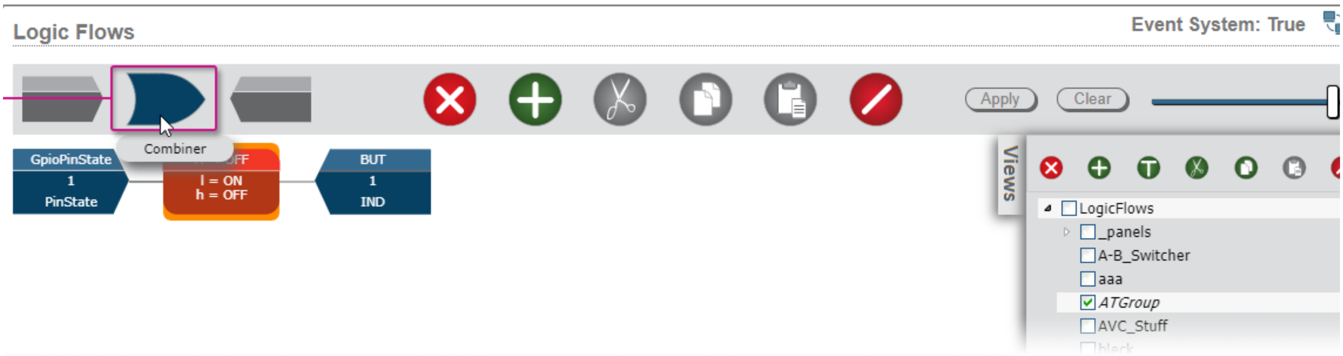
Within the top of the translation box, we see the currently active translation state in text and a color change depending on that state. These dynamic changes indicating the state of the Flow makes for much easier monitoring and troubleshooting.

> ⓘ Note: These colors come from a file that may be edited in an advanced web page if necessary.

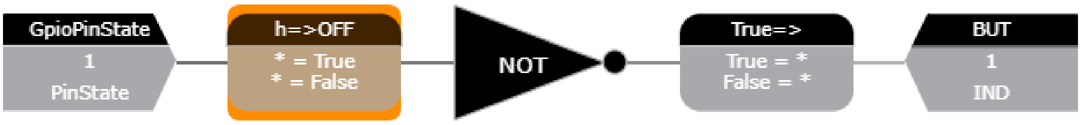
Adding Combiners

Translation points are one-to-one only. In other words, we are translating one value from a property to another value on another property. Combiners, on the other hand, allow us to pass values to multiple End Points and make logical decisions based on more than one Start Point.

- To add a Combiner, click a **Translator** in an existing Flow to enable the *Combiner* tool in the toolbar; click the **tool** to insert a Combiner into the Flow



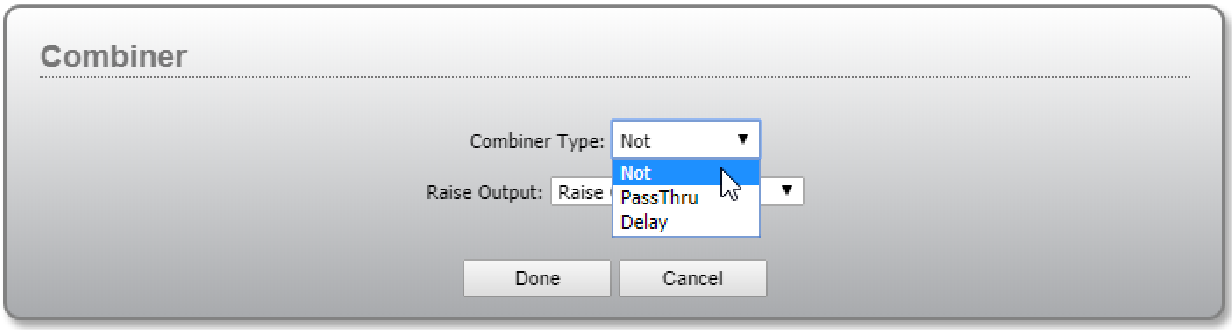
Using the Flow we created in the example above, it will look like this:



Combiners can perform many useful functions, but each Combiner type both require a minimum number of inputs and are limited to a maximum number of inputs.

Type	Minimum Inputs	Maximum Inputs
AND	2	Any
OR	2	Any
NAND	2	Any
NOR	2	Any
XOR	2	Any
XNOR	2	Any
NOT	1	1
RELAY	2	2
PASSTHRU	1	1
EQUALITY	2	Any
DELAY	1	1

As you attempt to change Combiner types, you will only see options that meet these restrictions. For example, with the single input in our sample Logic Flow, we are restricted to NOT, PASSTHRU, and DELAY Combiner types:



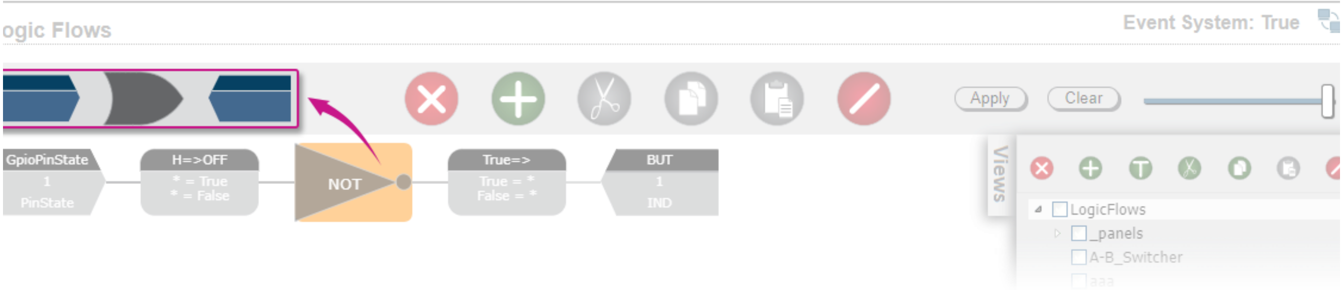
Combiner

Combiner Type: Not

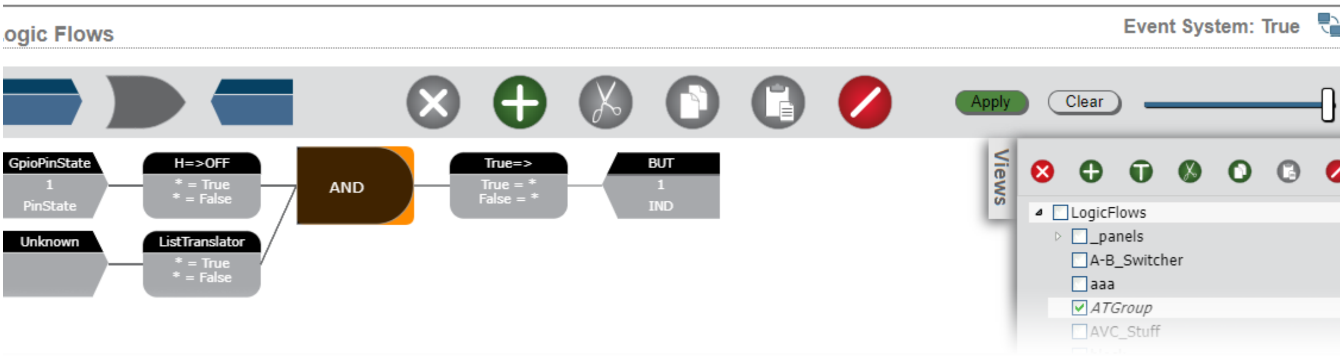
Raise Output: Raise

Type	Minimum Inputs	Maximum Inputs
NOT	**1**	**1**
PASSTHRU	**1**	**1**
DELAY	**1**	**1**

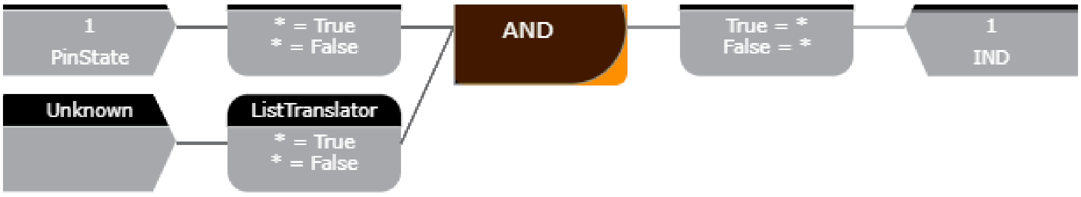
- We can access the other Combiner types by adding the requisite number of inputs by clicking on the **Combiner** to select it; notice that both the *Start Point* and *End Point* tools then become available



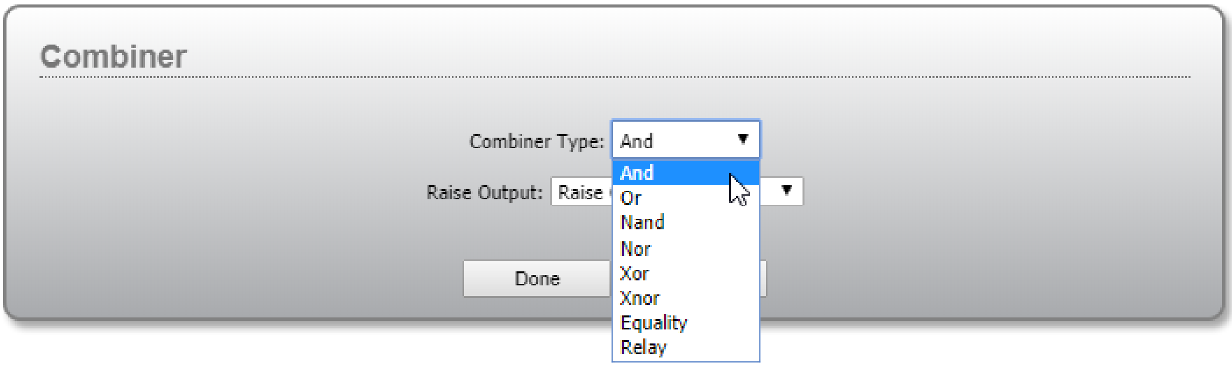
- Click on the **Start Point tool** to add a second input to our Flow



By adding a second input, you will notice the Combiner switches to AND.

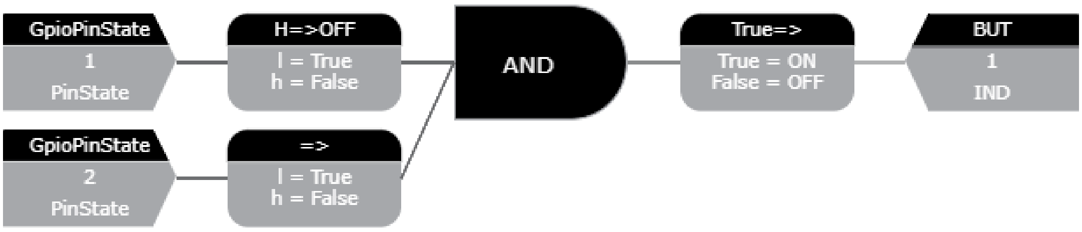


With two inputs, we can perform other Combiner operations, including: AND, OR, NAND, NOR, XOR, XNOR, EQUALITY, and RELAY.



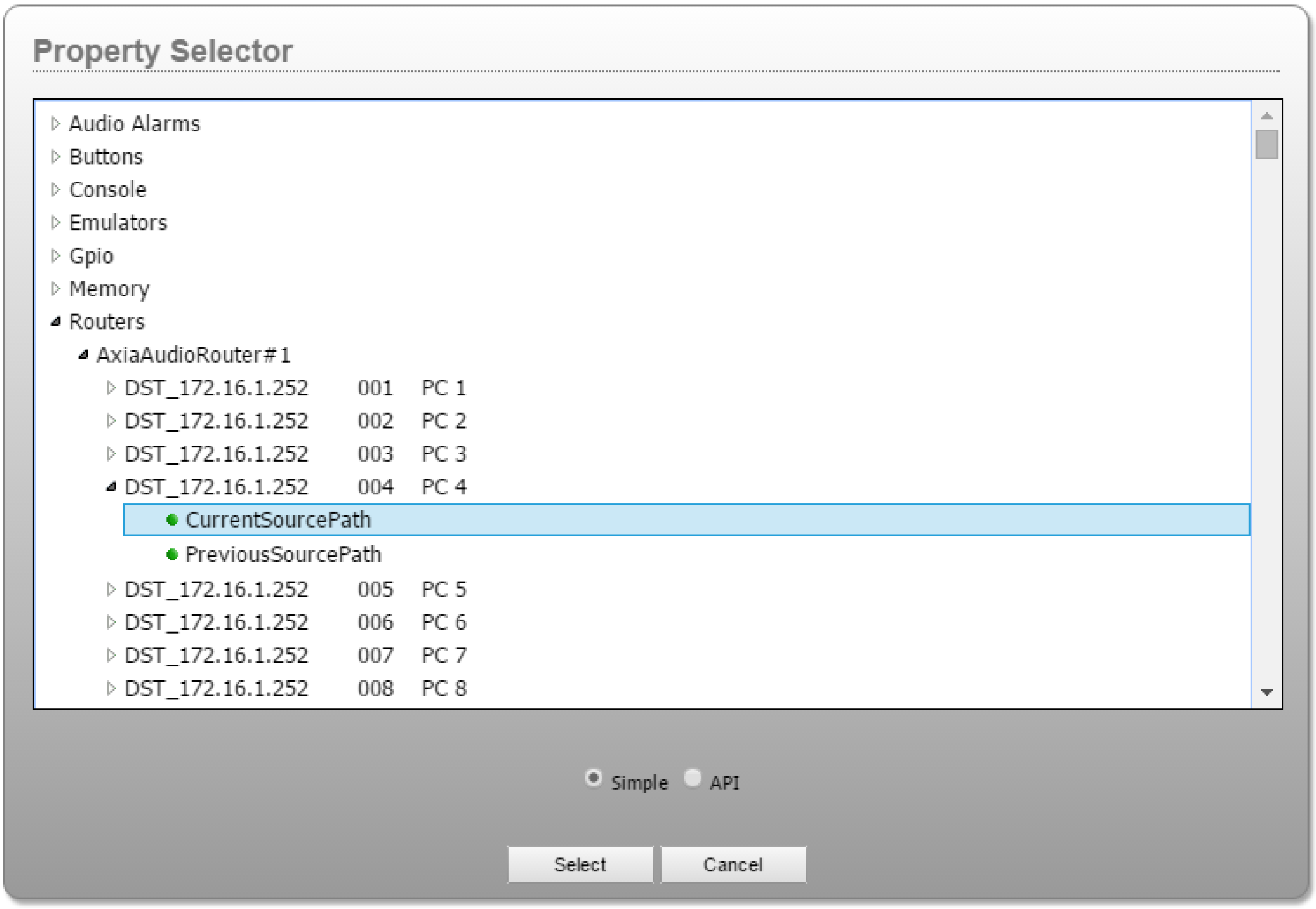
Type	Minimum Inputs	Maximum Inputs
AND	**2**	Any
OR	**2**	Any
NAND	**2**	Any
NOR	**2**	Any
XOR	**2**	Any
XNOR	**2**	Any
RELAY	**2**	2
EQUALITY	**2**	Any

In this case, since the default AND Combiner is a logic gate, the translation types entering and exiting the Combiner are automatically switched to *True/False*. By assigning a second GPI pin to the second Start Point…



…we can change this event so that two buttons must be held simultaneously for the End Point change to happen. This Flow will light the indicator light on the button only if both GPI 1 and 2 are *Low*.

- As it is, this Flow is not very useful, but if we were to change the End Point to be a route destination, it becomes more interesting; to do so, double-click on the **End Point** and instead of a GPIO, expand the Routers section and select the **CurrentSourcePath** property of one of the router destinations

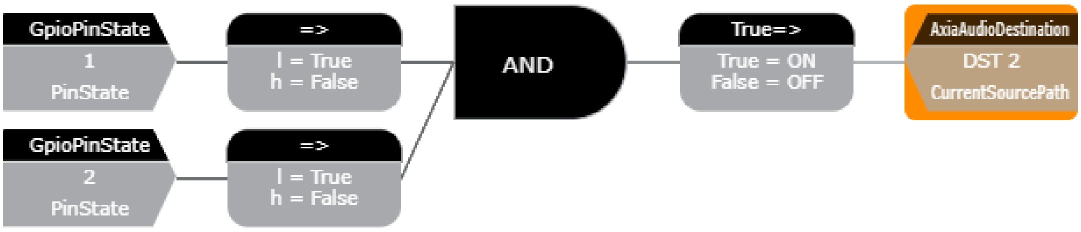


Property Selector

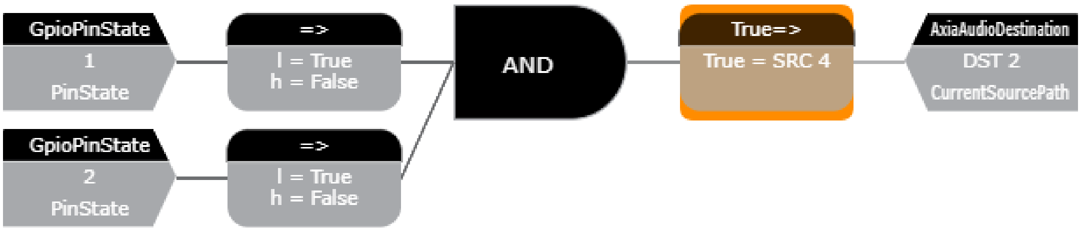
- Audio Alarms
- Buttons
- Console
- Emulators
- Gpio
- Memory
- Routers
  - AxiaAudioRouter#1
    - DST_172.16.1.252 001 PC 1
    - DST_172.16.1.252 002 PC 2
    - DST_172.16.1.252 003 PC 3
    - DST_172.16.1.252 004 PC 4
      - ● CurrentSourcePath
      - ● PreviousSourcePath
    - DST_172.16.1.252 005 PC 5
    - DST_172.16.1.252 006 PC 6
    - DST_172.16.1.252 007 PC 7
    - DST_172.16.1.252 008 PC 8

● Simple ○ API

Your Flow should now look something like this:



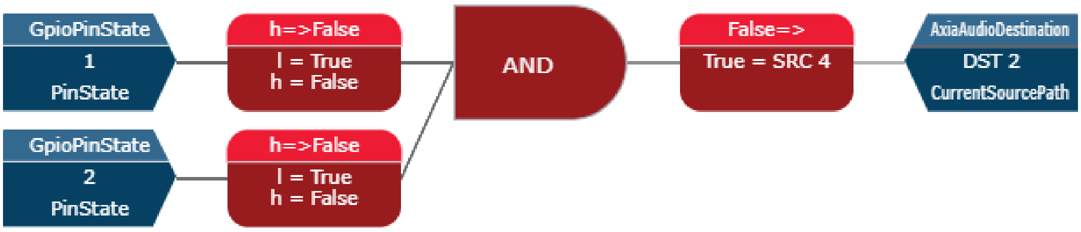
- Edit the translation which is after the Combiner and change the translation so that *True*= one of the sources in the right-hand drop-down of the translation dialog; remove the *False* translation as we do not want to do anything if the buttons are released, and you end up with:



This essentially means a user must hold two GPIO buttons down simultaneously and then source SRC 4 will be routed to destination DST 2. This is a common scenario where we don't want an accidentally bumped button to change the air chain routing.

Logical Combiners

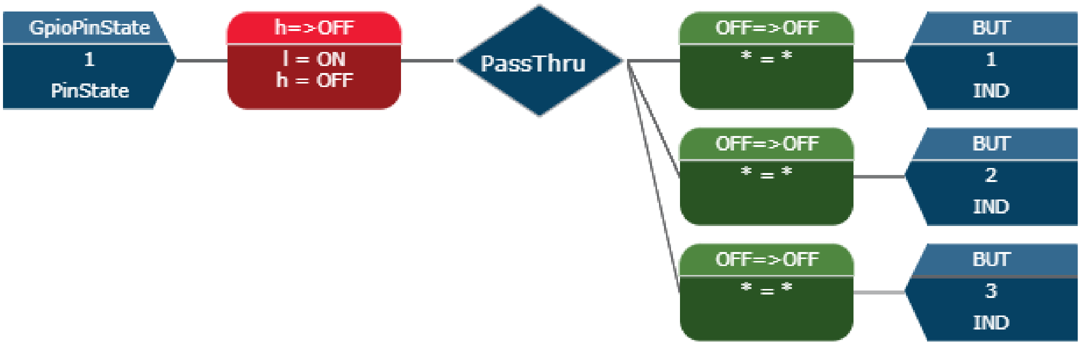
Logical Combiners include AND, OR, NAND, NOR, XOR, XNOR, and NOT. All of these except NOT accept multiple inputs which must be translated to *True* or *False* on the input side. NOT only accepts a single input. In each case, the translation that feeds the input of these Combiners should be converted to a *True* or *False* like we did in the example above.



The output of the Translator then becomes either *True* or *False* based on the truth table associated with the logical type selected.

PASSTHRU

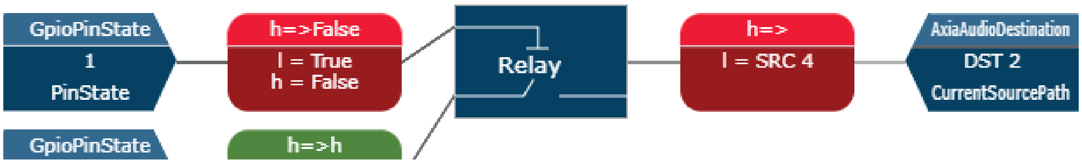
The PASSTHRU Combiner allows only one input but can pass the value from the Start Point translation to as many output Translators as you like, allowing you to change multiple output properties based on a single input property.

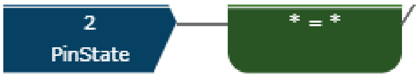


RELAY

The RELAY Combiner allows you to define one Start Point property that will translate to a *True* or *False*. If it evaluates to *True,* the value from the second Start Point is passed through the Combiner to any translations on the output of the Combiner.

The diagram within the combination displays which Start Point translation is acting as the trigger and which is acting as the PASSTHRU:

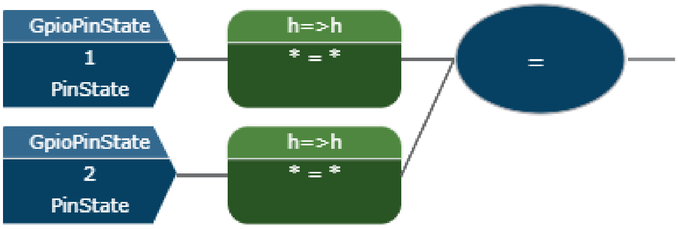




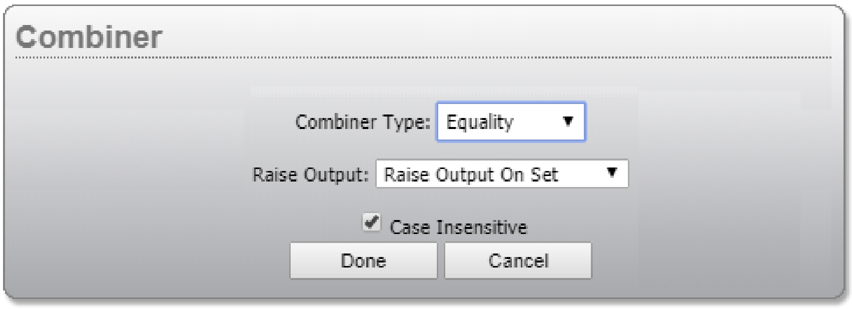
This will achieve the same functionality as our previous example but in a different way. GPI 1 must be *Low* in order for changes taking place on GPI 2 to pass thru. If it is *Low* then changes from GPI 2 will be passed to the output translation which converts the *Low* to a route change.

EQUALITY

The EQUALITY Combiner takes multiple inputs and will result in an output value of *True* or *False* depending on whether all input values match.



The EQUALITY Combiner also has a *Case Insensitive* checkbox which defines whether comparisons are done in a case sensitive or insensitive way:



Combiner

Combiner Type: Equality ▼

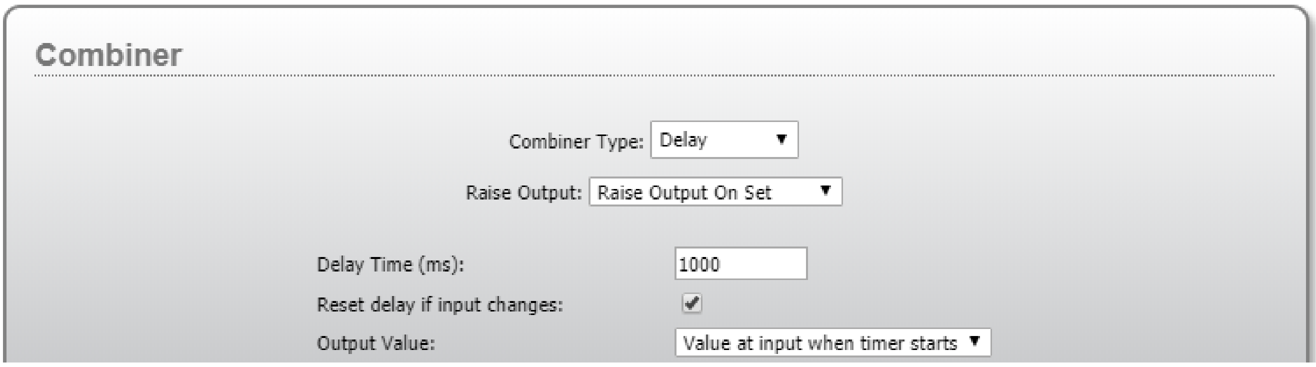
Raise Output: Raise Output On Set ▼

☑ Case Insensitive

This Combiner can be used in situations where the primary concern is whether the property states are equivalent or not.

DELAY

The DELAY Combiner takes a single input and introduces delay directly into the Logic Flow. It works like a PASSTHRU Combiner in that it takes the input value and passes it through to the output but only after a configured number of milliseconds. There are also some additional parameters that can affect how this Combiner functions:



Combiner

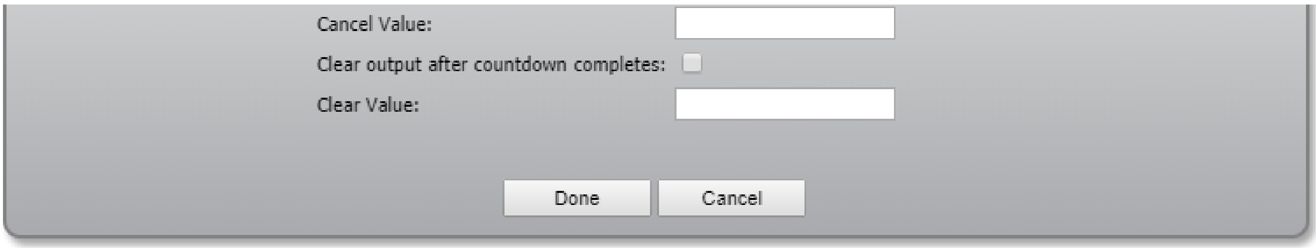
Combiner Type: Delay ▼

Raise Output: Raise Output On Set ▼

Delay Time (ms): 1000

Reset delay if input changes: ☑

Output Value: Value at input when timer starts ▼



Cancel Value:

Clear output after countdown completes: □

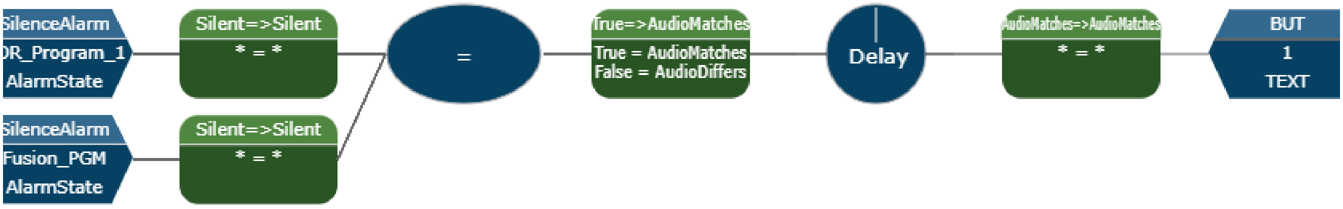
Clear Value:

Field	Description
Delay Time	Type the **Delay duration** in milliseconds.
Reset delay if input changes	The delay countdown starts every time the input changes its value. If this option is selected the countdown will be reset whenever the input value changes. • If you have a parameter that is fluttering and you only want an action to happen if it settles down to a fixed value for more than **x** milliseconds, this option should be checked. • If you want the value to happen **x** millisecond after a change even if additional changes happen during the countdown (delay time) the this option should be unchecked.
Output Value	From the drop-down list, select the option that defines whether the value that is passed through i the input value at the start of the delay countdown (**Value at input when timer starts**) or the end of the countdown (**Value at input when timer ends**) This option accounts for the possibility that the input value could change again during the delay countdown and allows you to define which value gets passed through.
Cancel Value	Type an **input value** that will cancel the timer and not make any change to the Combiner output (not pass any different value through).
Clear output after countdown completes	Used to make the delay Combiner function in a momentary fashion. *See note below.*
Clear Value	Type a specific **value** as the clear state when the clear output after countdown completes option is selected.

DELAY Combiners can be used in many situations where you need to introduce some delay into a Logic Flow. Previously this required using an interval timer. Therefore, there will be many situations where an interval timer is no longer necessary to accomplish the task as the delay can be built directly into the Flow. This also reduces licensing requirements as the DELAY Combiner does not require an intermediary timer End Point.

Interval timers are still useful in situations where a single delay needs to be stopped, started, reset, or manipulated by different Flows.

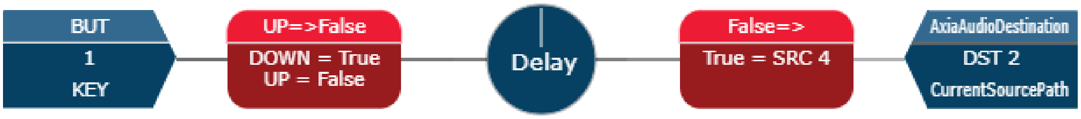
DELAY/EQUALITY Flow Example



In this example, the EQUALITY Combiner outputs *True* or *False* depending on whether two silence alarms are in the same state. This can be useful if two audio channels should have the same audio. Silence is expected in certain situations, but we are trying to make sure that if audio is present on one, it is present on both and if it is silent on one then it is silent on both. *True* or *False* is translated into a message of *AudioMatches* or *AudioDiffers*. A delay is introduced to expunge short variations of alarm states and only pass on definitive states. This is one example of how the DELAY and EQUALITY Combiners might be used.

DELAY Momentary Example

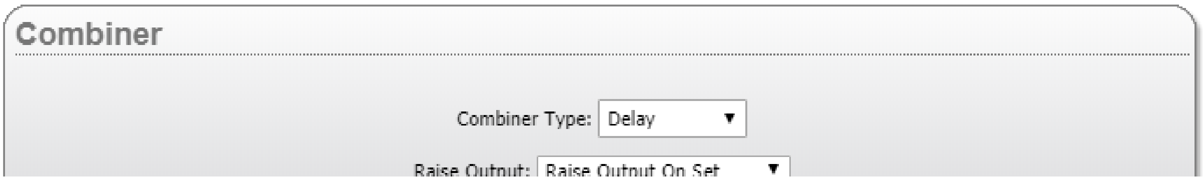
In many cases you may want the DELAY to pass a value through after the delay but then reset the output for the next change. For example, if we want a Flow that requires a user to hold a button down for a length of time before making a change, the Flow might look something like:



Our goal with this Flow is the following:

- If the button is pressed, start a 5 second timer
- If the button is released cancel the timer
- If the button is held for 5 seconds make the route change

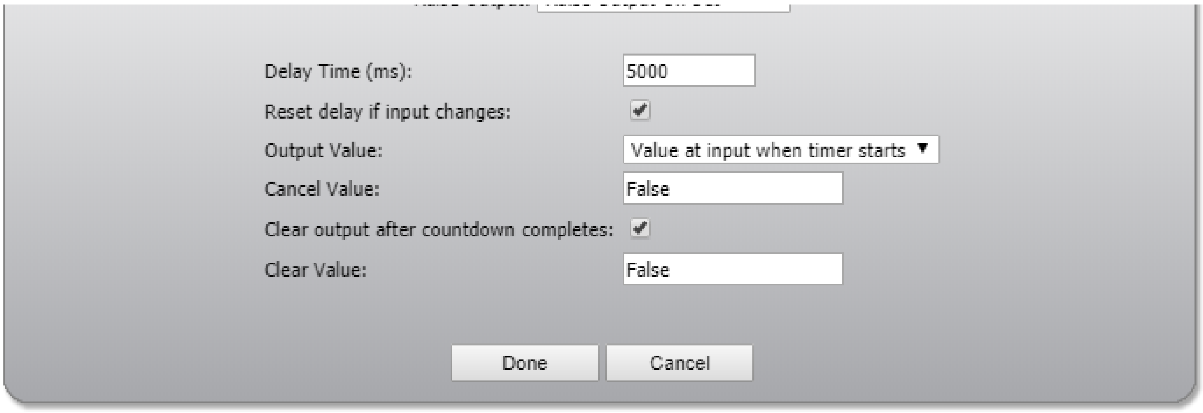
The DELAY Combiner has an *input* value and an *output* value. Changes only get passed to the output Translator when the Combiner's output value changes. And the delay is only analyzed for countdown when the Combiner's input value changes. For this example, we would set the DELAY Combiner parameters as follows:



Combiner

Combiner Type: Delay ▾

Raise Output: Raise Output On Set ▾



Delay Time (ms): 5000

Reset delay if input changes: ☑

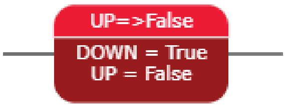
Output Value: Value at input when timer starts ▼

Cancel Value: False

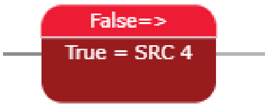
Clear output after countdown completes: ☑

Clear Value: False

- If the user presses the button, the input value gets set to *True* based on the inbound translation:



- The timer starts counting
- If the user releases the button before the timer elapses, the input value of the Combiner gets set to *False*. Since this matches the cancel value, the delay timer stops counting
- If the user does not release the button for 5 seconds, then the *True* value gets passed to the Combiner output which is being monitored by the output Translator:



- A route change is made
- Since the *Clear Value* and *Clear Output after countdown completes* options are set, the output value of the Combiner is then set to *False* again so it is ready for the next button press
- When the user releases the button, the Combiner input is set to *False* but since that is the Cancel field no change to the Combiner output is made

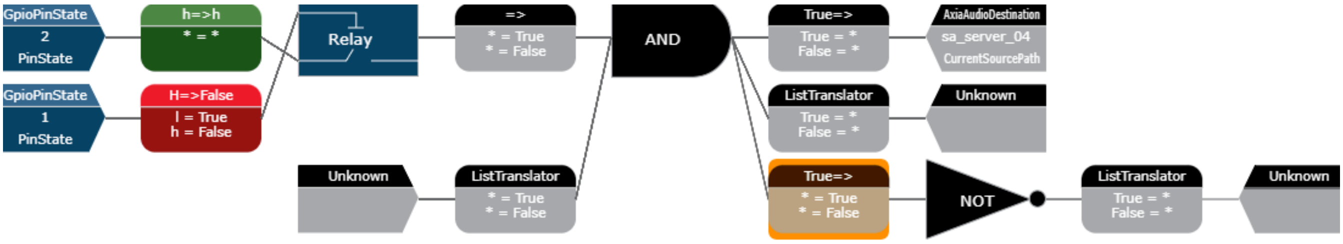
Without the *Clear Value* and *Clear Output after countdown completes* options, the output of the Combiner would remain *True* and so the next press would not change the output and therefore would not trigger another route.

If we cleared the *Cancel* value and did not use the *Clear* options, releasing the button would pass *False* to the output but 5 seconds after the button was released. It is important to note that a *Cancel* value if set will cancel the countdown, but the *Cancel* value does not get passed through.

These parameters will allow you to produce differing effects using the DELAY Combiner depending on the required goal.

Going Crazy

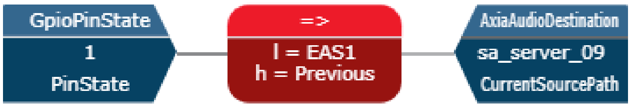
By adding Combiners and Translators you can make any individual Flow as deep as you need it to be.



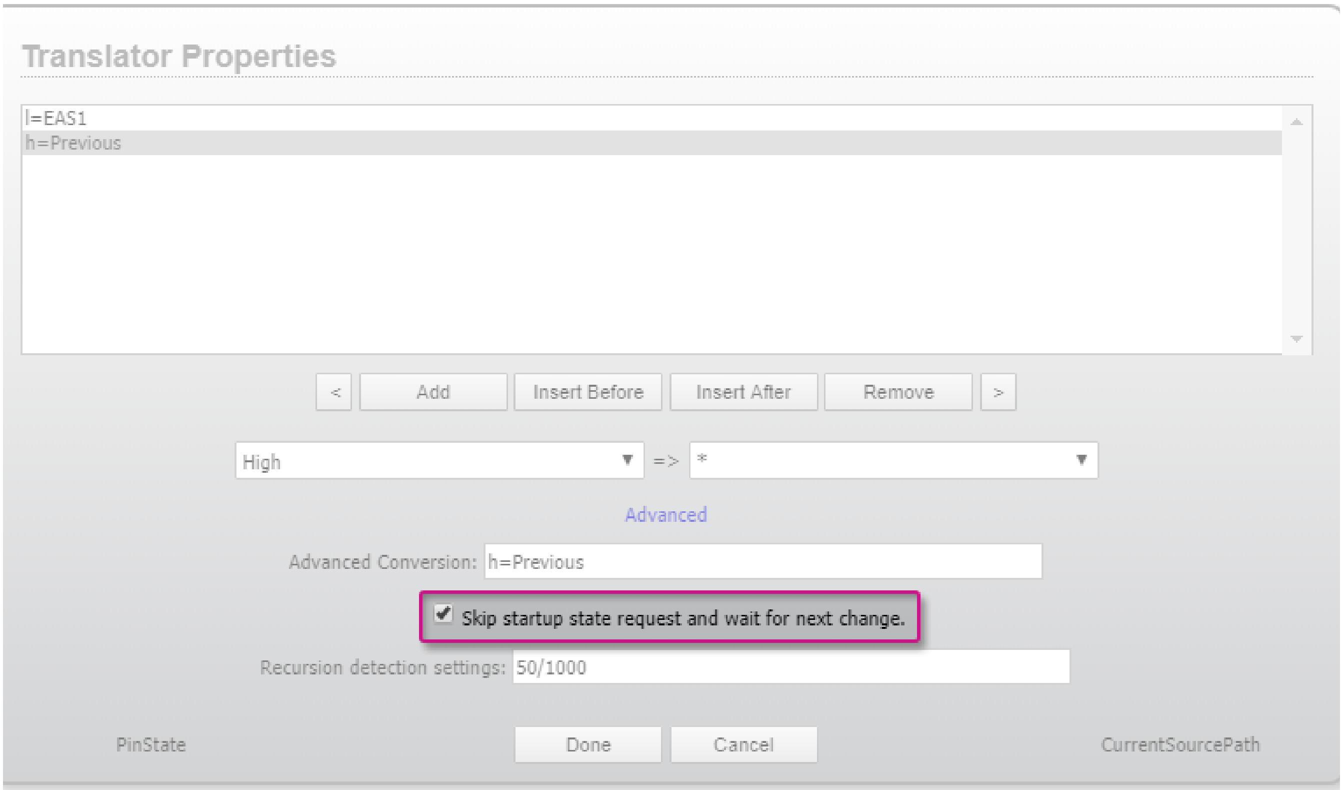
This example is not created to do anything specific; it is just shown to illustrate that you can theoretically continue to add Translators and Combiners as needed to get the logical job done. Translators and Combiners must alternate. You cannot have a Combiner next to a Combiner or a Translator next to a Translator, but the user interface helps with that by only enabling the controls you can use depending on which logic block you currently have selected.

EAS Example

Another relevant example is EAS. This is a pretty common EAS Flow that we see customers make:



Important Note: if you are going to use a Flow like this, it is critically important that you also turn on the **Skip startup state request and wait for next change** option. See the *Advanced Options* section on this option.



Translator Properties

l=EAS1
h=Previous

High => *

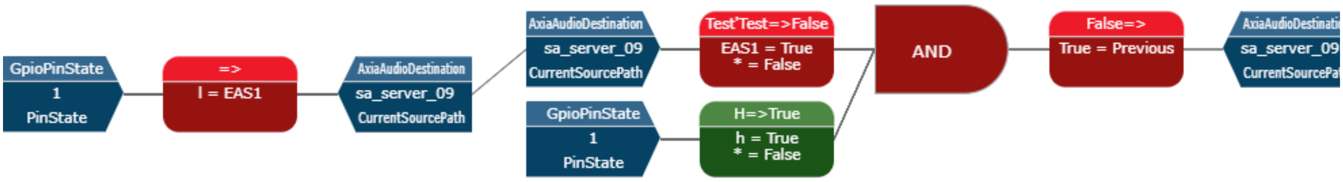
Advanced Conversion: h=Previous

☑ Skip startup state request and wait for next change.

Recursion detection settings: 50/1000

Without enabling this option, a restart of the system will request the current GPI state which is *High* and will therefore toggle the *Previous* route. In many cases this will route “nothing” to the air chain because no previous state exists yet.

However, even with that option enabled this Flow may produce unforeseen results. For example, what happens if the GPI goes *Low* and someone manually changes the route before it goes *High* again? At that point EAS is now the previous source when it does go *High*. This can be solved by using a slightly more complex Flow and *Previous:*



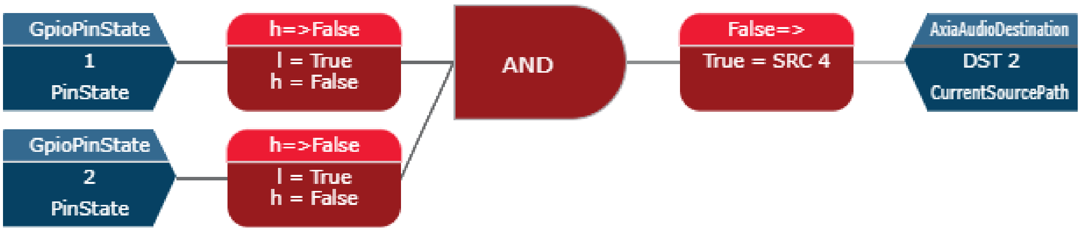
In this example, a route to *Previous* will only happen if the GPI is *High* and we are currently on the EAS source. This will function in a much more reliable manner.

While these examples have been EAS related the same rules apply to other uses of previous. More importantly, if you can avoid the use of *Previous* and design the Flows to be more specific, the outcome will also be more reliably specific.

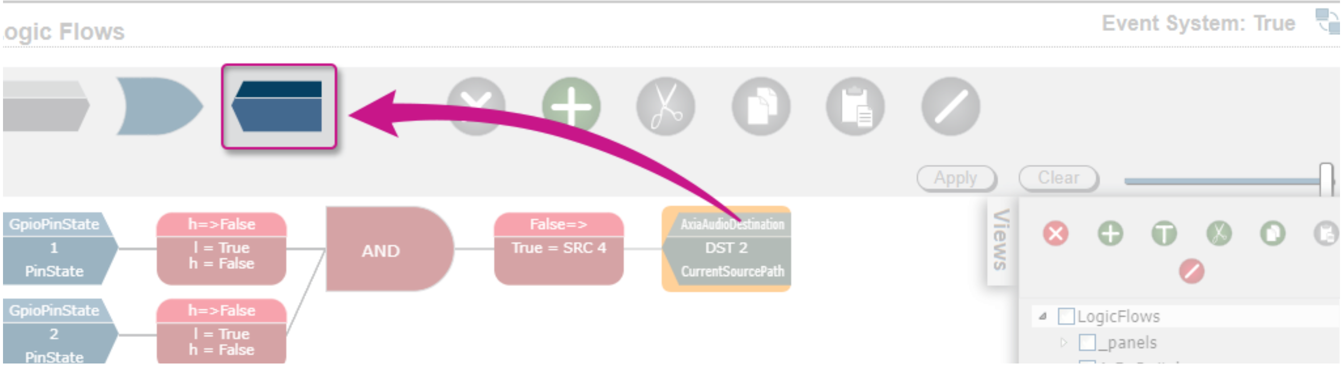
Extending Flows

If you select the End Point of a Flow, you will notice that the right Translator control in the control panel is available and not disabled. This is what is referred to as "extending a Flow". You can create a second Flow whose Start Point is the same as the End Point of the previous Flow.

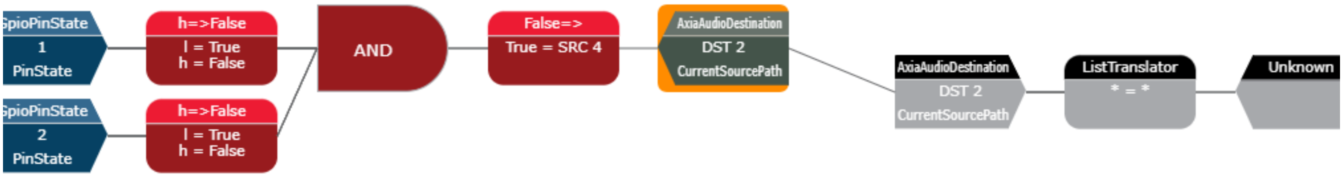
If we go back to this example Flow:



Click on the **End Point** and then click the **End Point control** in the tool bar.

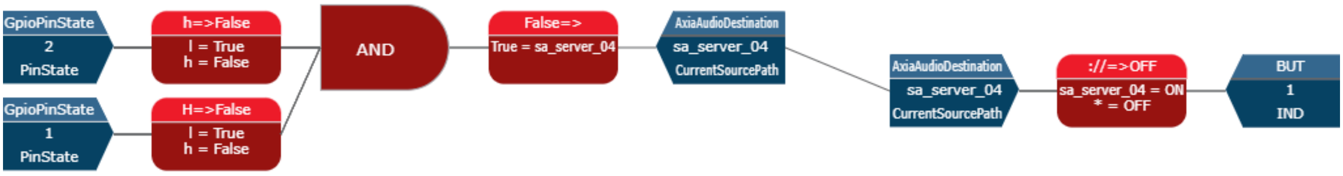


Our Flow now looks something like this:



In this case, the second Flow is joined to the first Flow because the Start and End properties are the same. Likewise, if we had added a second Flow using the + button and then set the Start Point to the DST 2 destination's *CurrentSourcePath*, after we applied the change, the system would detect that the Flows should be joined and join them.

In this example we could now set the indicator of a button at the End Point of the second Flow:



What we have designed is a Flow where if you hold GPI buttons 1 and 2 down, PC4 source will be routed to PC4 destination. And if PC 4 source is routed to PC4 destination then LCD button 1's indicator will be turned on. If anything else is routed to that destination then the Indicator will be off.

The system is intelligent enough to recognize when Flows should be joined into a single logical entity.

Cut, Copy, and Paste

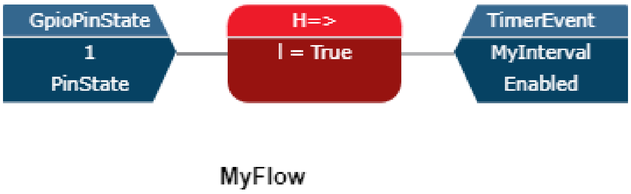
Cut, *Copy*, and *Paste* are standard tools and require little explanation. You can select a Flow by clicking in the area surrounding a Flow at which point the cut and copy icons become available. "Cut" will remove the Flow from the View and place it on the clipboard while "copy" will just make a copy of it on the clipboard. Once a Flow exists on the clipboard it can be pasted back into the current View or any other View.

The system does make some assumptions regarding pasting. If you copy a Flow and paste it back into the same View, the system assumes you are trying to duplicate the functionality with a new set of End Points. It will clear the outer Start and End Points of the pasted Flow so you can select new ones.

It is important to reiterate that none of the changes involved in a cut/copy/paste operation ever become a reality on the system until you click the **Apply** button, and that clicking the **Cancel** button will return to the last known applied state.

Flow Titling

Each Flow can be assigned a title. If you double-click in the whitespace surrounding a Flow, the system will ask you for a title for the Flow. Click **Apply** to apply the title. Flows that have titles will display the titling text underneath the Flow.

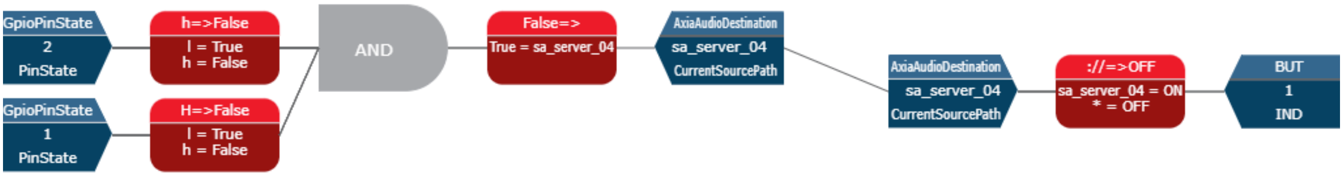


Titling is a very important habit to form as it makes it much easier to glance at a Logic Flow View and understand the states of the Flows and which Flows are performing which functions. Additionally, it is useful as an organizational tool as Flows with titles will be displayed alphabetically within the View. Otherwise the Flow order within the View is somewhat arbitrary.

Disabling/Enabling

Translators, Combiners, and entire Flows may be enabled or disabled. Select the Translator, Combiner, or entire Flow (by clicking in the whitespace within the Flow), and then use the disable/enable button on the control panel.

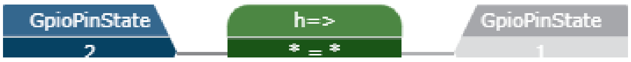
This change is the only one that takes place immediately within Pathfinder Core PRO Logic Flows without the need to click **Apply**. The object will become greyed out and that particular object will no longer function, stopping the Logic Flow at that point. For example:



If you trigger the GPIs, their states will change within the attached Translators, but the End Point in the first half of the Flow will not change because the Combiner is disabled. This is a useful tool if you are testing logic up to a point but do not want the End Point (which might be an air chain change) to change. It is also useful if you want to temporarily disable a Flow during testing or troubleshooting.

Cluster Disabling

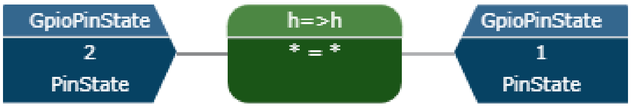
If your system is clustered and you are Viewing Logic Flows on the secondary system, you may see a lighter gray form of disabling:



Also with clustering the upper right corner will show an *Event System* state:

Event System: False

In this case you are being shown that the final output will not actually execute on this server because the other server is currently in control of the event system. On that server, you will see:



And in the right corner:

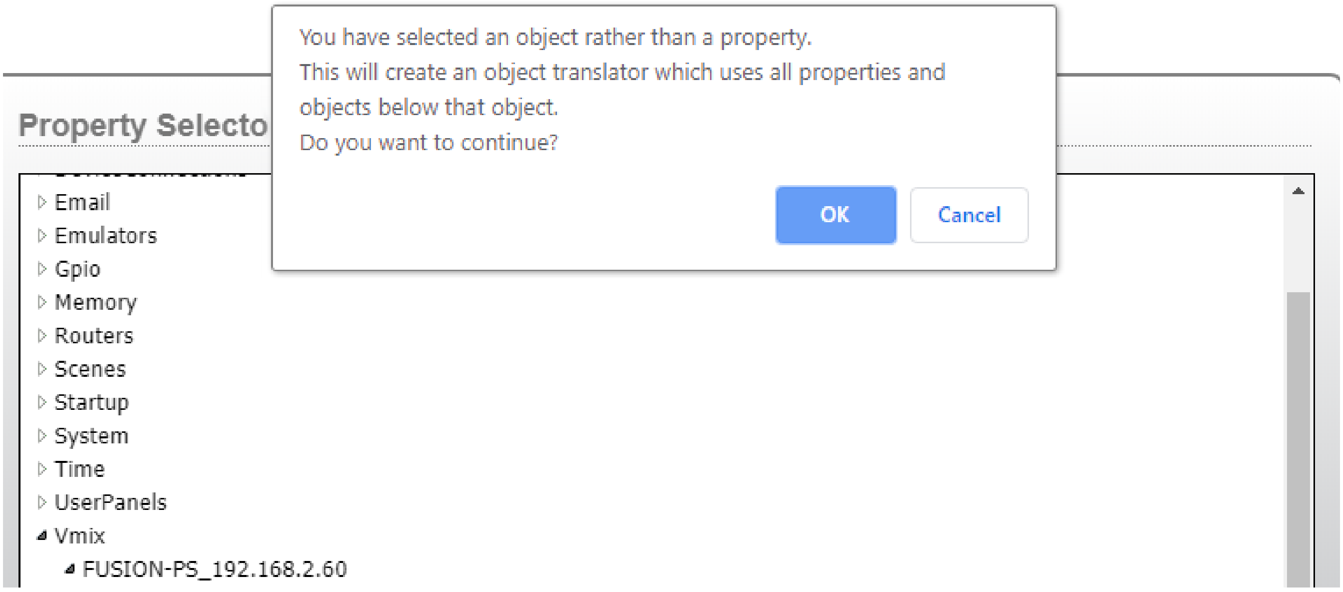
Event System: True

> ⓘ Important Note: The event system is never actually turned off in Pathfinder Core PRO. When a Logic Flow needs to make a change to an End Point, it first checks to see if it has the lowest IP address of the currently active and online servers in the cluster. If so then it executes the action. If not, then it does not.

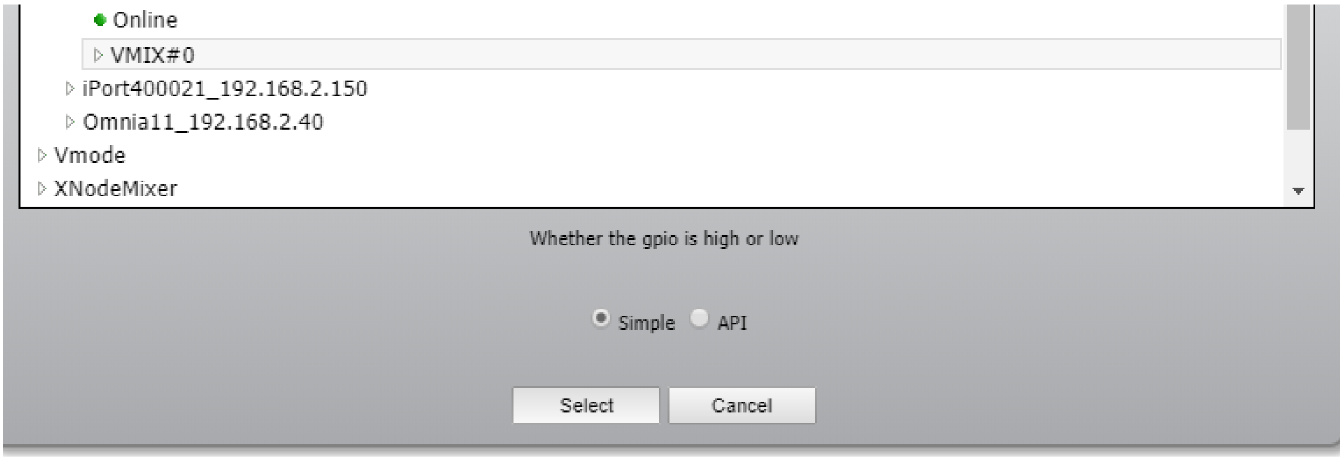
The Event System *True* or *False* will be hidden on systems that are not in a cluster because in that case the Logic Flow End Points will not be disabled by the clustering system.

Object Translators

You may notice when editing the Start or End Point of a Flow that if you select an *object* in the property selection tree (arrow icon) instead of a *property* (green dot icon), you will get a warning message that looks like:



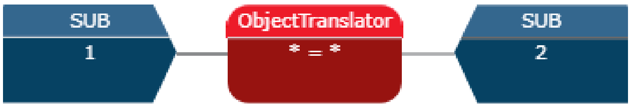
You have selected an object rather than a property.
This will create an object translator which uses all properties and objects below that object.
Do you want to continue?



Whether the gpio is high or low

● Simple ○ API

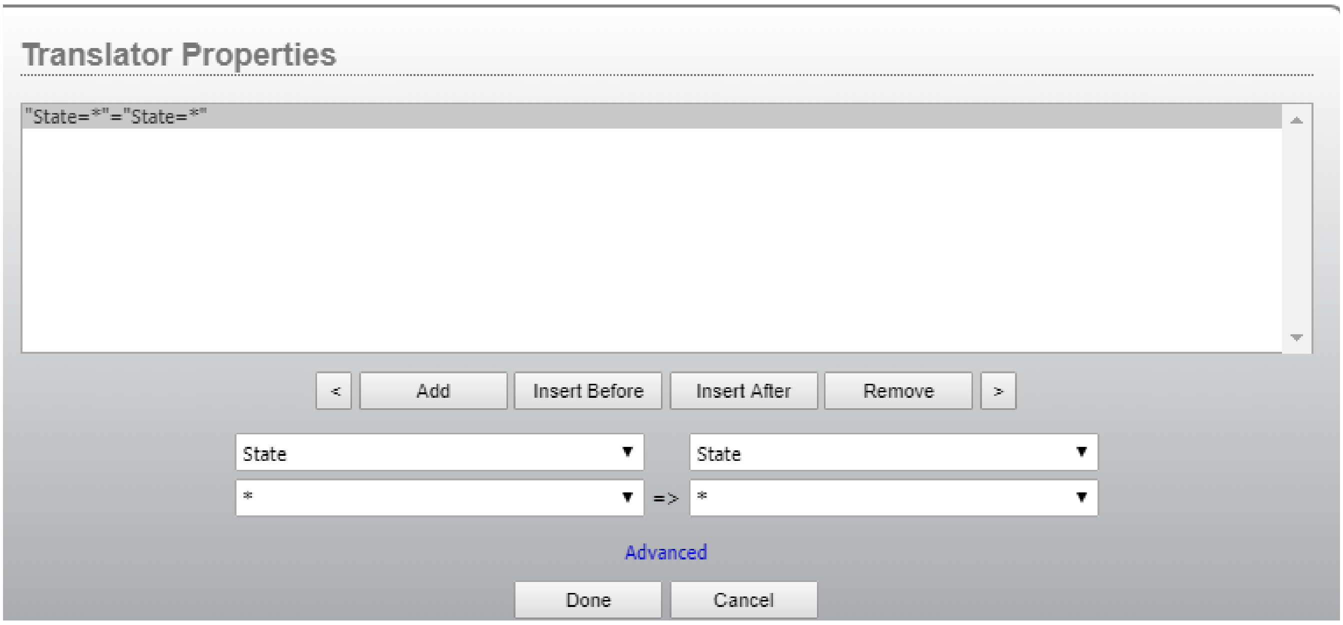
Object translations are a slightly more advanced subject and should be used with caution (hence the warning), but they can also be extremely powerful. Object translations will most likely be used in situations where you want to mirror several different property states across a couple of objects. We could create a Translator Logic Flow where the Start Point is a VMIX sub mixer on Engine 1 and the End Point is a different VMIX sub mixer on a different engine:



This Translator looks slightly different in that the Start and End Points only have two pieces of information rather than the usual three.

If we do not alter the translation pattern from the default ***=***, every change to every property of every sub-object under the sub mixer 1 will get mirrored to the equivalent object and property under sub mixer 2. For example, if sub mixer 1, channel 1 gets turned *On*, then sub mixer 2, channel 1 would also be turned *On*. If sub mixer 1, channel 3 has its *TimeDown* property changed, then sub mixer 2, channel 3 will also get its *TimeDown* property changed. This allows us to mirror an object's settings.

When you edit the translation list and select an object translation, there are additional drop-downs that allow us to specify the object's *properties* as well as its *values*. For example, we could change the translation pattern from ***=*** to:


Translator Properties

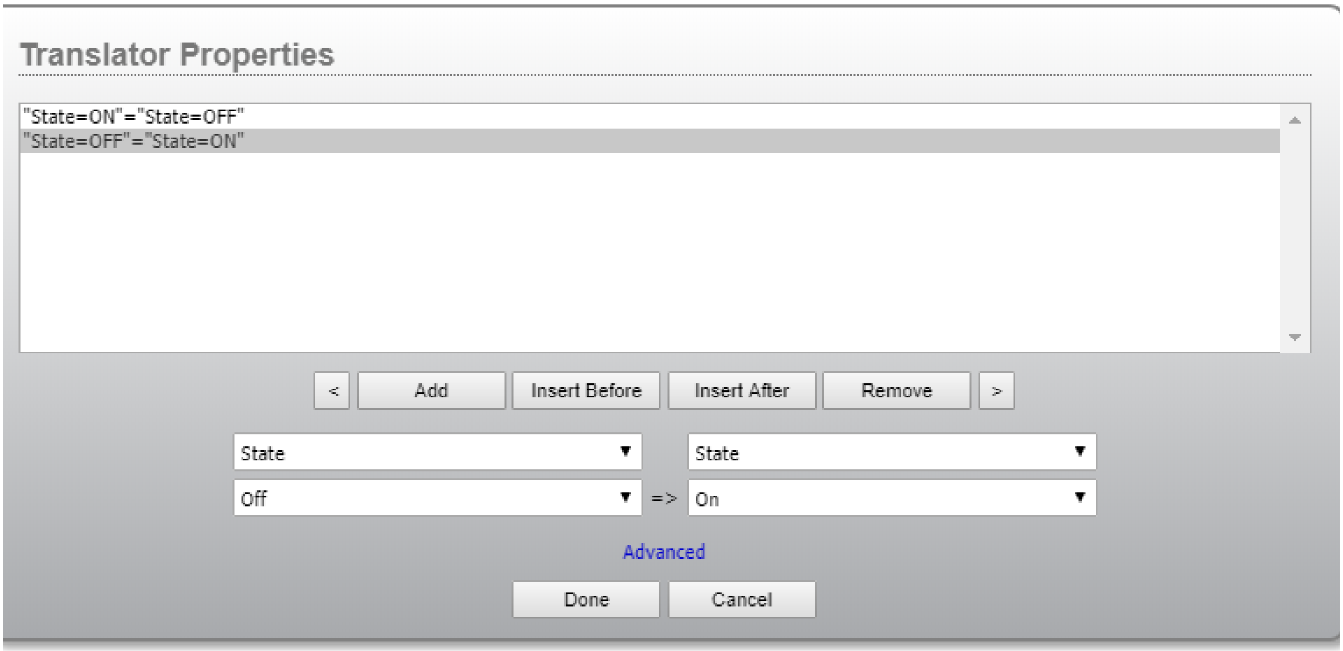
"State=*"="State=*"

State ▼ State ▼

* ▼ => * ▼

In this case, we are specifying that we only want this translation to operate on *State* properties. If sub mixer 1, channel 4's *State* property gets turned on, so would sub mixer 2, channel 4's state property. Other properties would not be affected.

Or, we could apply values such as:



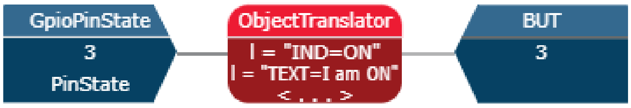
Translator Properties

"State=ON"="State=OFF"
"State=OFF"="State=ON"

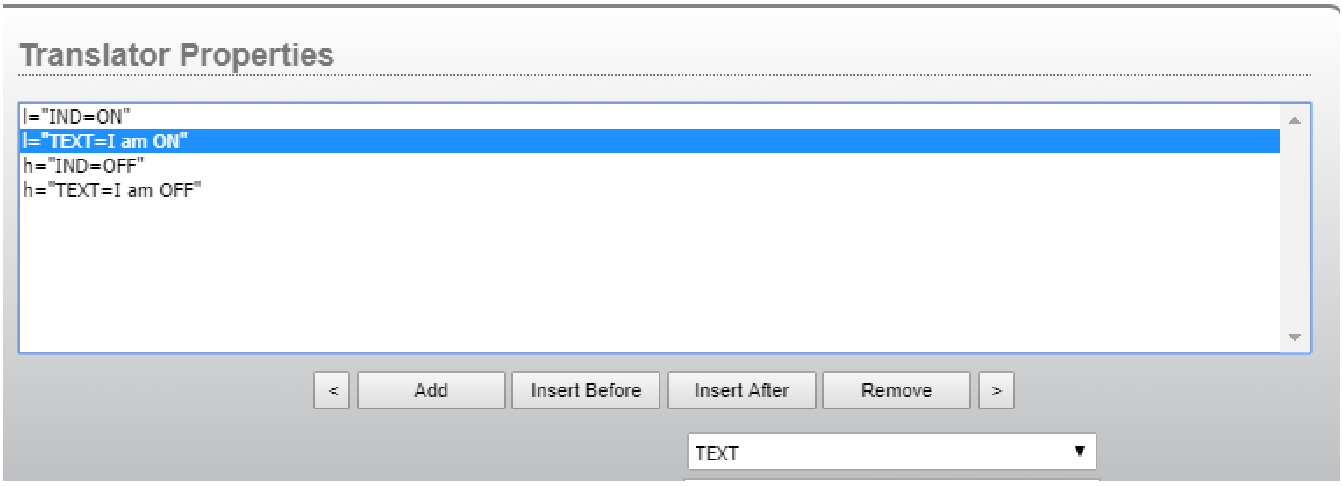
State ▼ State ▼
Off ▼ => On ▼

This allows us to invert the states on the second sub mixer. Now when sub mixer 1, channel 5's *State* property gets turned on, sub mixer 2 channel 5's *State* property gets turned off and vice versa.

Another useful way to use Object Translators is when you want to set multiple properties on an object. For example:



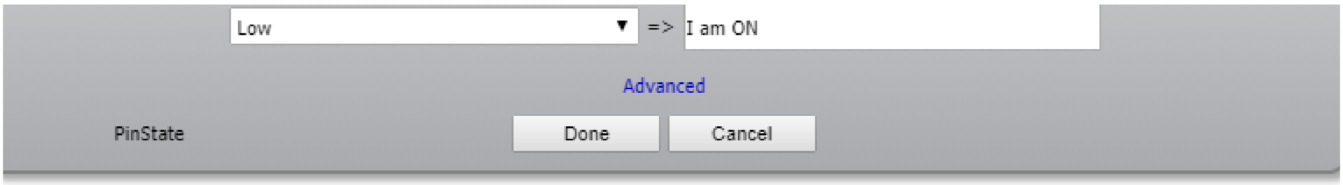
In this case the Start Point is a normal property but the End Point is an object selection. If we click on the translation properties we can make the configuration look like this:



Translator Properties

l="IND=ON"
l="TEXT=I am ON"
h="IND=OFF"
h="TEXT=I am OFF"

TEXT ▼



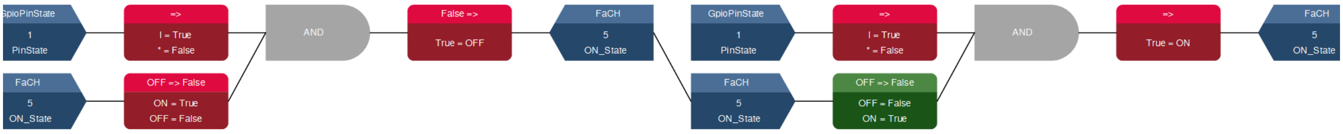
In this case, we are setting two different properties on the button each time the GPIO pin state changes *High* or *Low*. This can simplify the Flows and greatly reduce the licensing counts used when there are many button states that need to be set.

Object translation is extremely powerful in situations where you want to match multiple properties across two similar objects or multiple properties on a single object.

Latching

It is common to need a button to latch, where each press of the button will toggle something back and forth. Using the kind of logic shown above to accomplish this can get complicated because the *state* becomes a condition which is also an *action*. For example, many novice Pathfinder Core PRO users may try to create a Flow that looks like this:

> ⚠ *Warning: Do not use this Flow. This is an example of what NOT to do!*



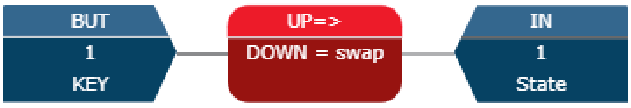
The goal of this Flow is that each time a GPI goes *Low*, the user wants to toggle the on state of the fader back and forth between on and off.

The user has tried to create a Flow where if the GPI is *Low* and the button is on, it turns off, and if the GPI is *Low* and the button is off, then it turns on. The problem is that this creates a loop that lasts as long as the GPI is *Low*.

Setting the GPI to *Low* changes the state of the console channel which in turn causes the second half of the Flow to change it again and so on until the GPI is returned to *High*. This leads to interesting and convoluted loops resulting from trying to solve what on the surface appears to be a simple problem. To avoid this scenario, we have crafted two much simpler approaches to latching in Pathfinder Core PRO.

Swap

Many of the properties that have two states (*On/Off*, *Low/High*, etc.) also have a value that you can set called *Swap*. For example:



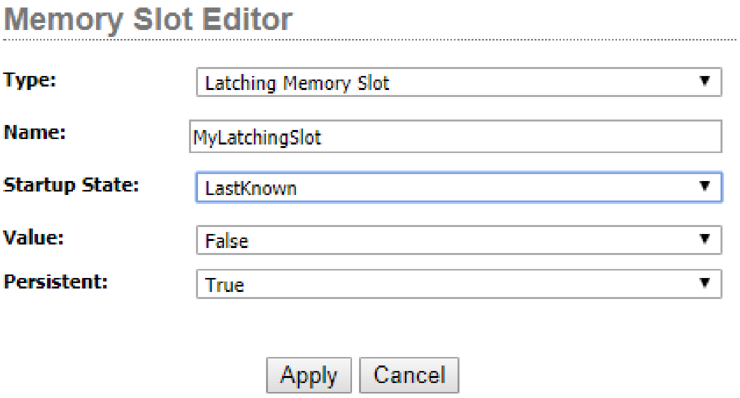
In this case, each time the button is pressed, the button state will be swapped (*On* to *Off* or *Off* to *On*).

If a *Swap* value is available, it is always preferable to other options.

Latching Memory Slot

To achieve similar behaviors for properties that do not have the *Swap* value, you can use the *Latching Memory Slot*. While basic memory slots can be created using the Logic Flows property tree as a shortcut, there are additional types that can only be created from the memory slots page as outlined in the Memory Slots chapter.

To create latching functionality, go to the memory slots link in the Navigation Bar and add a new memory slot using the plus icon. In the resulting dialog, switch the memory slot type to “Latching Memory Slot”. Provide a name for the slot and click **Apply** to create the slot.



Memory Slot Editor

Type: Latching Memory Slot

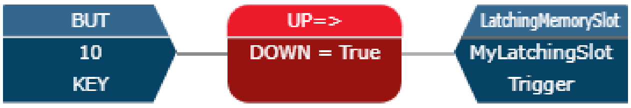
Name: MyLatchingSlot

Startup State: LastKnown

Value: False

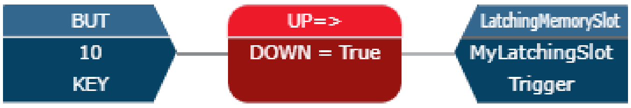
Persistent: True

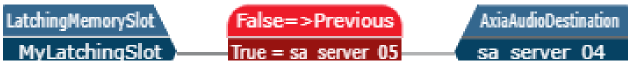
The Latching memory slot can only have a value of *True* or *False*. For example, in Logic Flows we could now add a Flow such that whenever a button is pressed on the console, the Translator sets the trigger property to *True:*



It also has a property called "Trigger" which is write-only and cannot be read. Whenever you apply a *True* value to it, it automatically switches the *SlotValue* property of the memory slot from *True* or *False* to the opposite state. This simple Logic Flow will cause the latching memory slot to switch its value back and forth every time button 10 is pressed.

Next, we can create a second Logic Flow that defines what we want to happen depending on whether the Latching memory slot is *True* or *False*. For example:

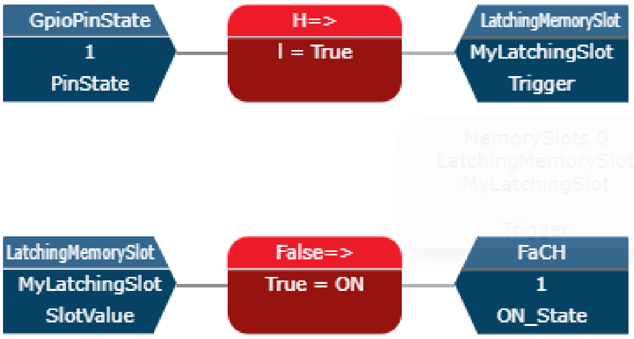




Now the button will cause the latching memory slot to switch back and forth between *True* and *False. If* the slot's value is *True* source sa_server_05, it will be routed to Destination sa_server_04. If the slot value is *False*, then whatever the previous source contains will be routed back to Destination sa_server_04.

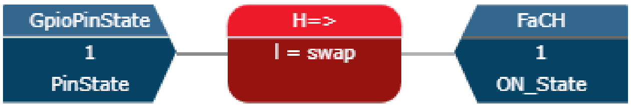
Swap over Latching Memory Slot

It is important to reiterate that *Swap* is preferable where it is available. For example:



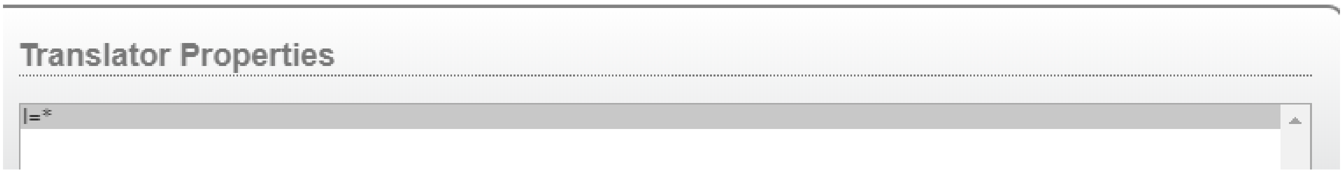
In this case, the Trigger property causes the latching memory slot to *Swap* between *True* and *False* each time the GPI goes *Low,* and then that *True* or *False* is translated to the on/off state of the fader. Engaging the GPIO repeatedly will cause the fader to turn on and off. The problem with this Flow is that the fader can also be turned on and off with the actual console button which causes the latching memory slot to get out of sync with the fader state. This could be solved with a third Flow if we allowed you to force the state of the memory slot each time the fader state changed, but this requires numerous Flows and the creation of a memory slot for every instance of this functionality. This approach feels too complicated to accomplish what should be a simple task.

The *Swap* value makes this much simpler and is all that is needed to solve this problem:



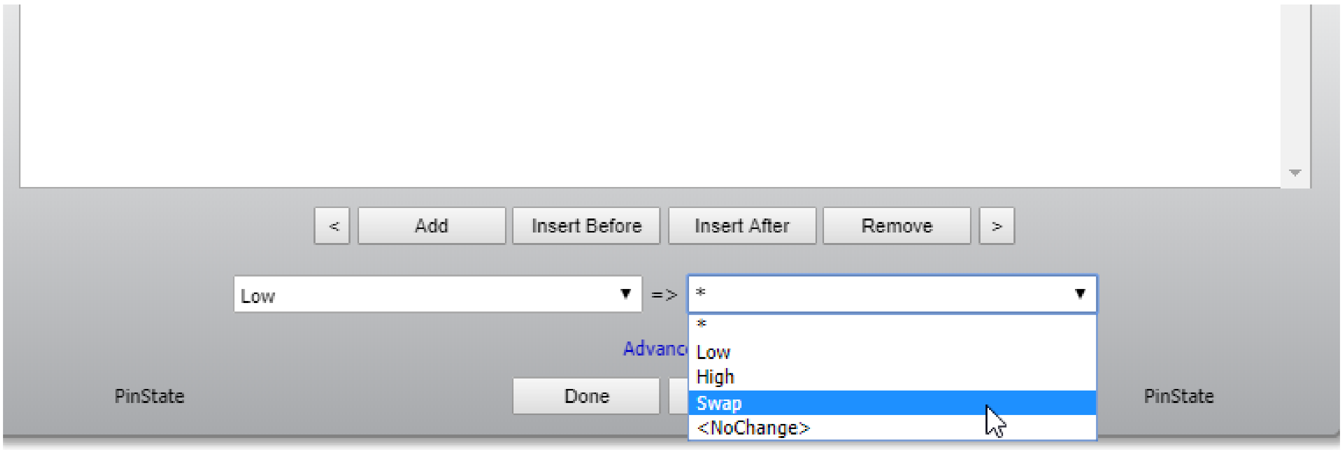
Each time the GPI goes *Low,* a value of *Swap* is sent to the fader. The device manager picks up this request, checks to see whether the fader is currently on or off, and sends the opposite state as a command to the equipment. The problem with the earlier Flows is that the Flow logic had to encapsulate all viable options of the existing state and specifically request what state to move to accordingly. The *Swap* property removes that complexity from the Flow logic and allows the system to handle it for you.

It is important to note that the *Swap* value is an action and so will only appear in the translation options for an End Point. It has no meaning for a Start Point because placed there, there it is not an actual state accordingly will not appear in the options on a Start Point. The user interface will present the option only where it is available. For example:



Translator Properties

I=*



Important Note: Axia LCD buttons and user button indicators represent a special case in that they have more than two options including *On, Off, Flash, Wink,* and a variety of other flashing states. However, we chose to add a *Swap* value to this property because *On* and *Off* are the states that are most often used. If a button indicator is *Off* and the *Swap* value is sent, it will turn the indicator on. If it is in any other state when the *Swap* value is sent, the indicator will be turned off.

Device-Specific Parameters

Infinity Product Support

Detected devices identified as Infinity products have a *SendToRestAPI* property which can be used by Logic Flows to address the Infinity Rest API. The format of the commands addressed to this property should be as follows:

```
Operator Path Value
```

For example:

```
PUT /api/s/main/audio/speakerMute True
```

Note that *Gets* have not yet been implemented as we do not want to encourage polling logic. It is also important to know whether or not the value being sent needs to be wrapped in quotes.

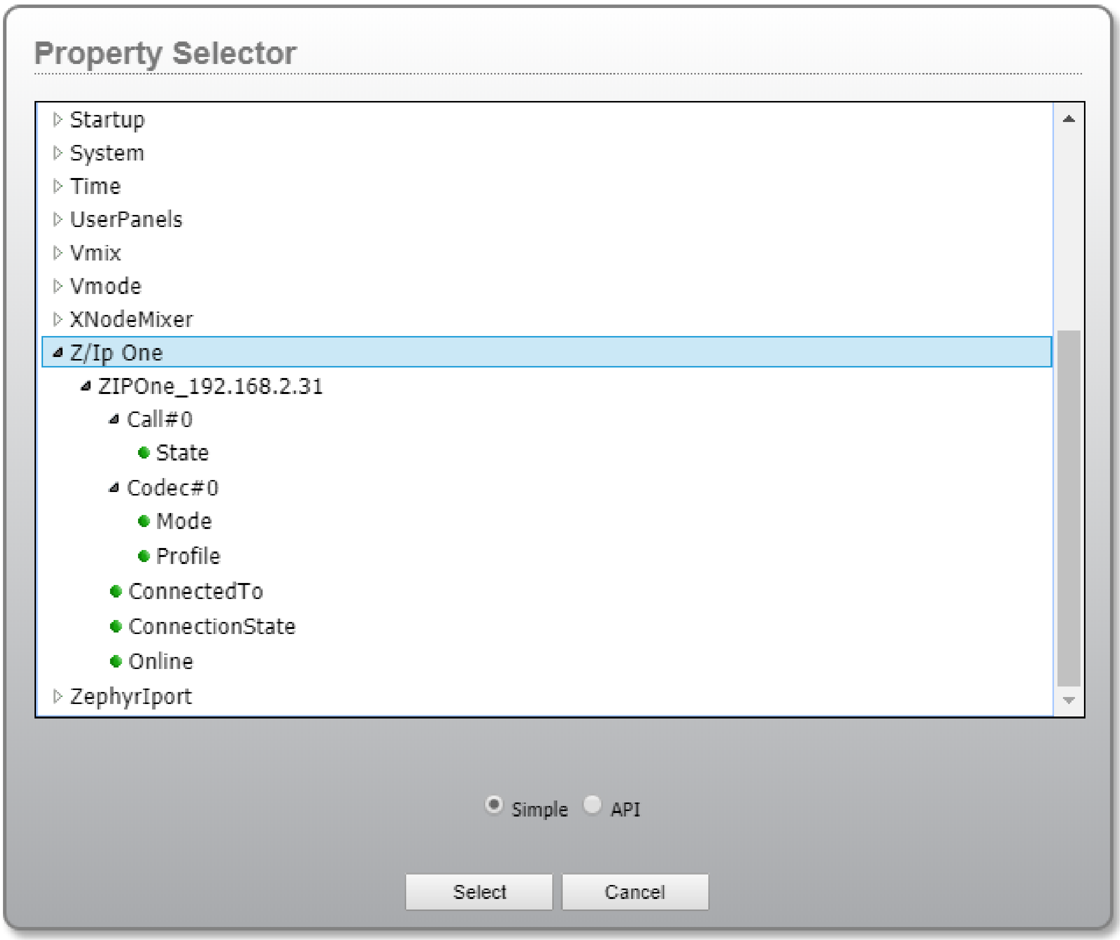
Contact Telos support for details on properties that may be manipulated using the Infinity Rest API.

Z/Ip One Control Points

Dialing and status items are exposed in the Logic Flows simple tree under the Z/Ip One branch.

Selecting Start Points

When viewing a Start Point there are several options:



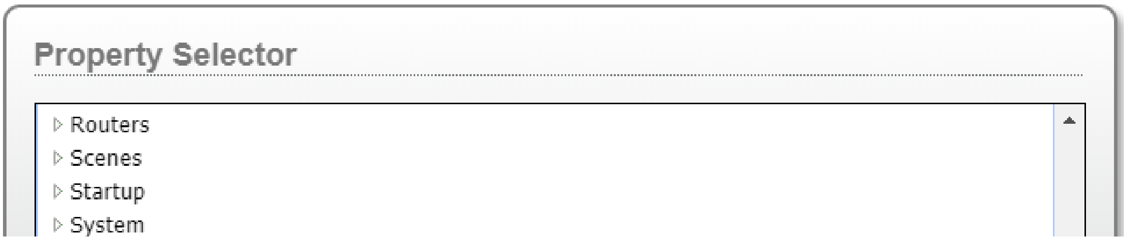
It is important to understand the difference between **Call#0 > State** and **ConnectionState**. **Call#0 > State** lists all possible connection statuses including multiple options for *Idle* and options for how the connection was established. **ConnectionState** offers only *Idle*, *Connecting*, and *Connected* options making it suitable for situations where logic needs to be fired based on the connection being active or disconnected.

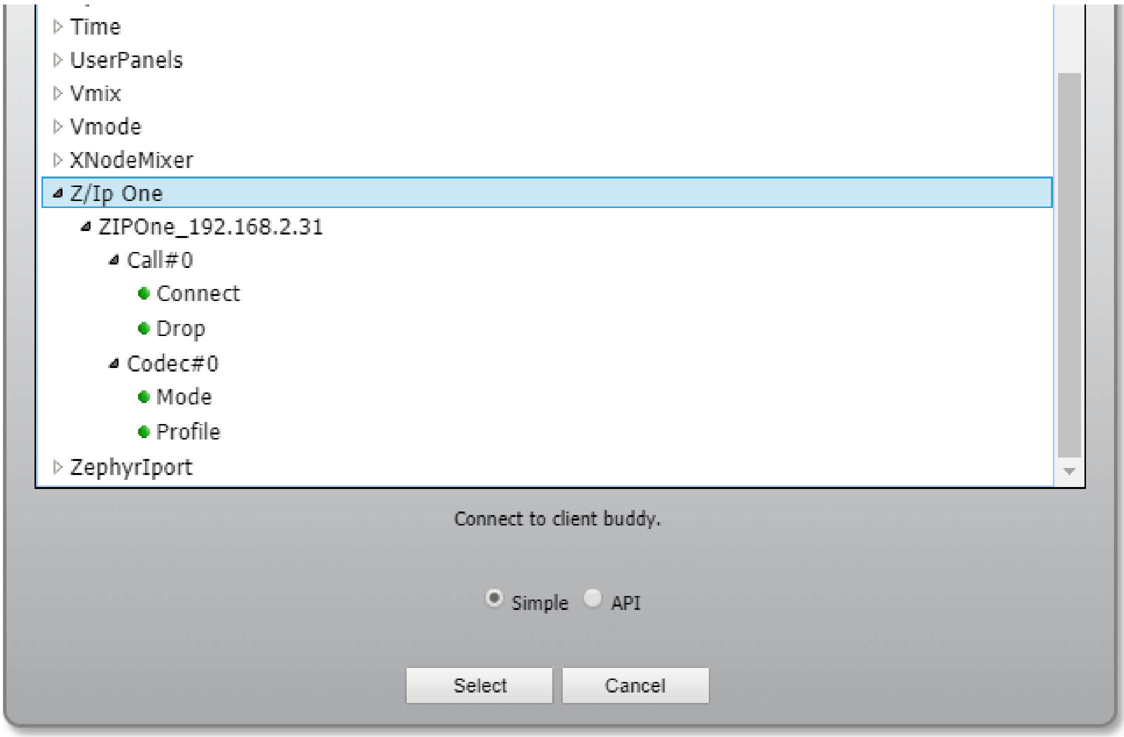
The **ConnectedTo** property will contain the phone book entry matching the one selected in the *Call Connect* property allowing you to display the current call connection phonebook entry name.

> Important Note: In addition to these properties, the API tree contains a much larger set of properties matching those from the Z/Ip One Lwcp API. Consult the Z/Ip One documentation for details on these other properties.

Selecting End Points

When selecting an End Point in the Simple tree, the Z/Ip One branch includes several properties for any Z/Ip One in the system:

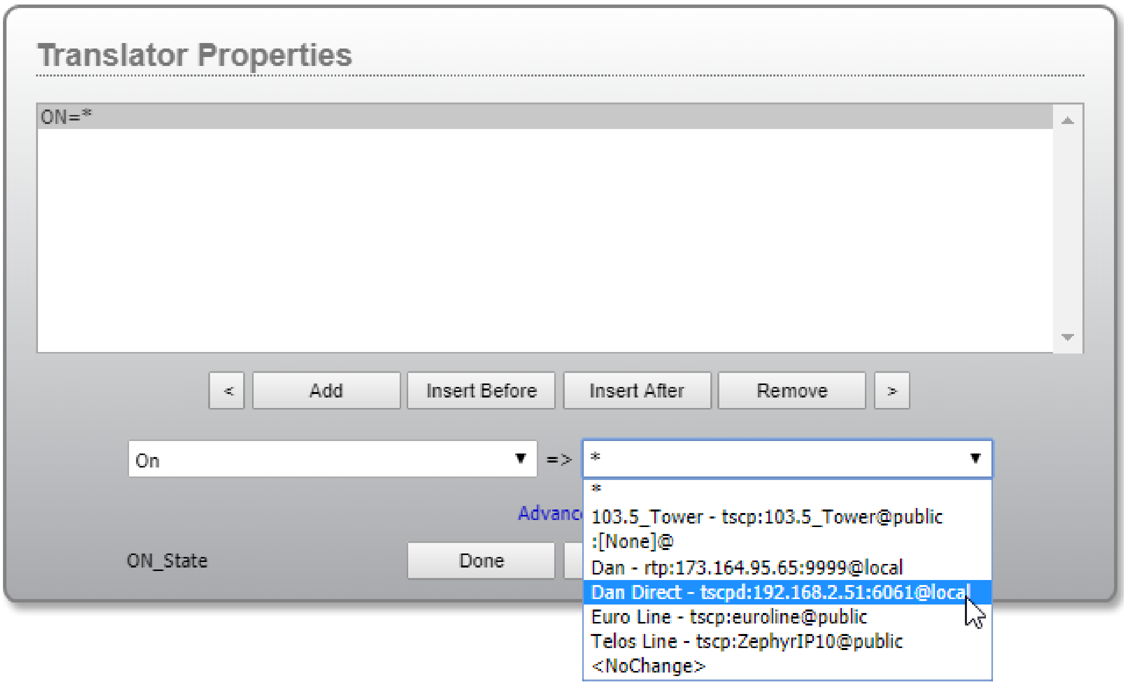




Connect to client buddy.

● Simple ○ API

Connect and *Drop* can be used to establish and drop connections. After selecting the *Connect* property as a Logic Flow End Point, the translation list will display the list of entries in the Z/Ip One phone book. Currently, only phone book entries may be used for dialing:


Translator Properties

ON=*

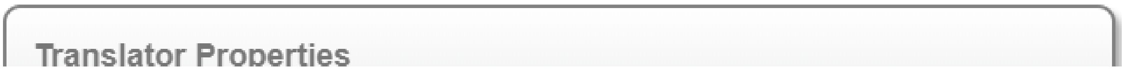
On => *

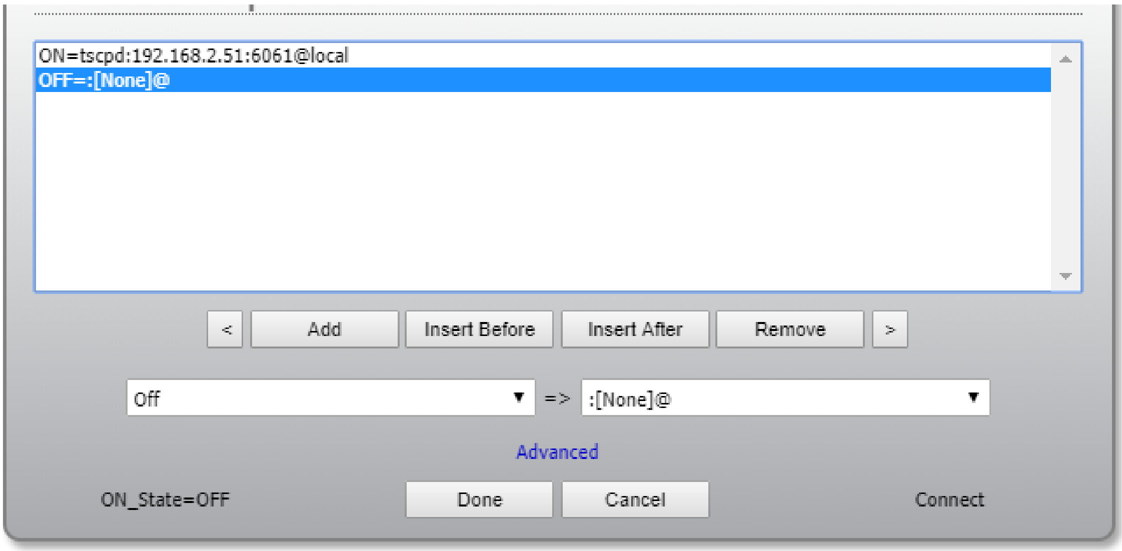
ON_State

Add
Insert Before
Insert After
Remove
Advanc
Done
*
103.5_Tower - tscp:103.5_Tower@public
:[None]@
Dan - rtp:173.164.95.65:9999@local
Dan Direct - tscpd:192.168.2.51:6061@local
Euro Line - tscp:euroline@public
Telos Line - tscp:ZephyrIP10@public
<NoChange>

Important Note: In general, the *Codec* settings will not be used as an End Point since those settings are normally stored within the call phone book entries.

There is also a special phonebook entry that only lives in Pathfinder Core PRO called **None** which, when dialed, will behave the same as the *Drop* item.


Translator Properties



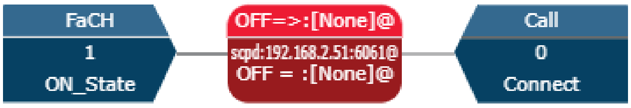
ON=tscpd:192.168.2.51:6061@local
OFF=:[None]@

Off => :[None]@

ON_State=OFF

Connect

For example:



FaCH
1
ON_State

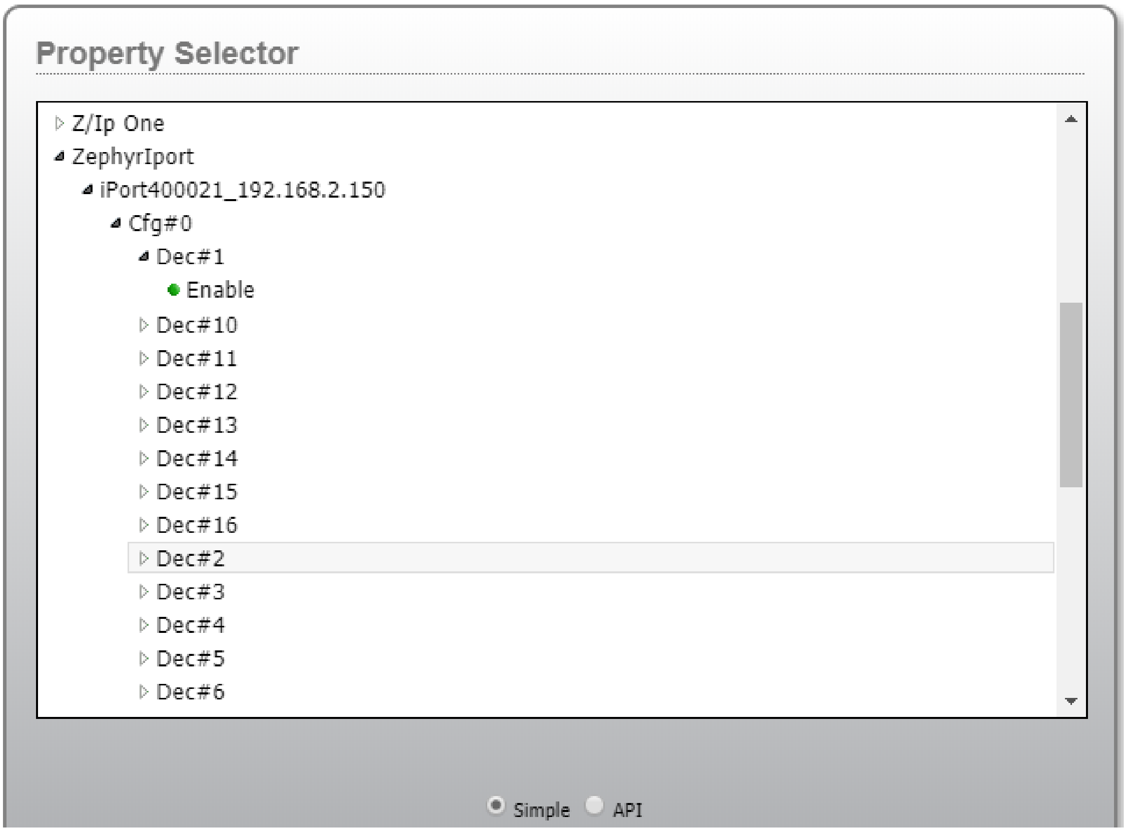
OFF=>:[None]@
scpd:192.168.2.51:6061@
OFF = :[None]@

Call
0
Connect

This Flow will dial the Z/Ip One every time Fader 1 is turned on and drop the call every time it is turned off.

Zephyr iPort Control Points

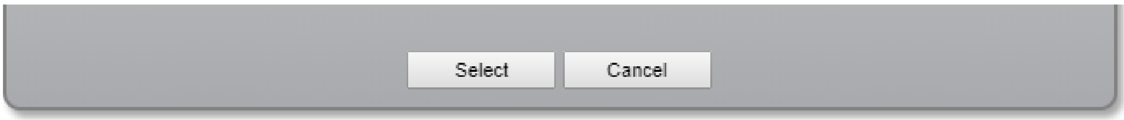
The iPort branch in the simple tree exposes the Encoder and Decoder *Enable* and *Disable* options. We have limited those options to the simple tree as most other properties related to the iPort MPEG side of the configuration are considered advanced and are not normally dynamically adjusted.



Property Selector

- Z/Ip One
- ZephyrIport
  - iPort400021_192.168.2.150
    - Cfg#0
      - Dec#1
        - Enable
      - Dec#10
      - Dec#11
      - Dec#12
      - Dec#13
      - Dec#14
      - Dec#15
      - Dec#16
      - Dec#2
      - Dec#3
      - Dec#4
      - Dec#5
      - Dec#6

● Simple ○ API



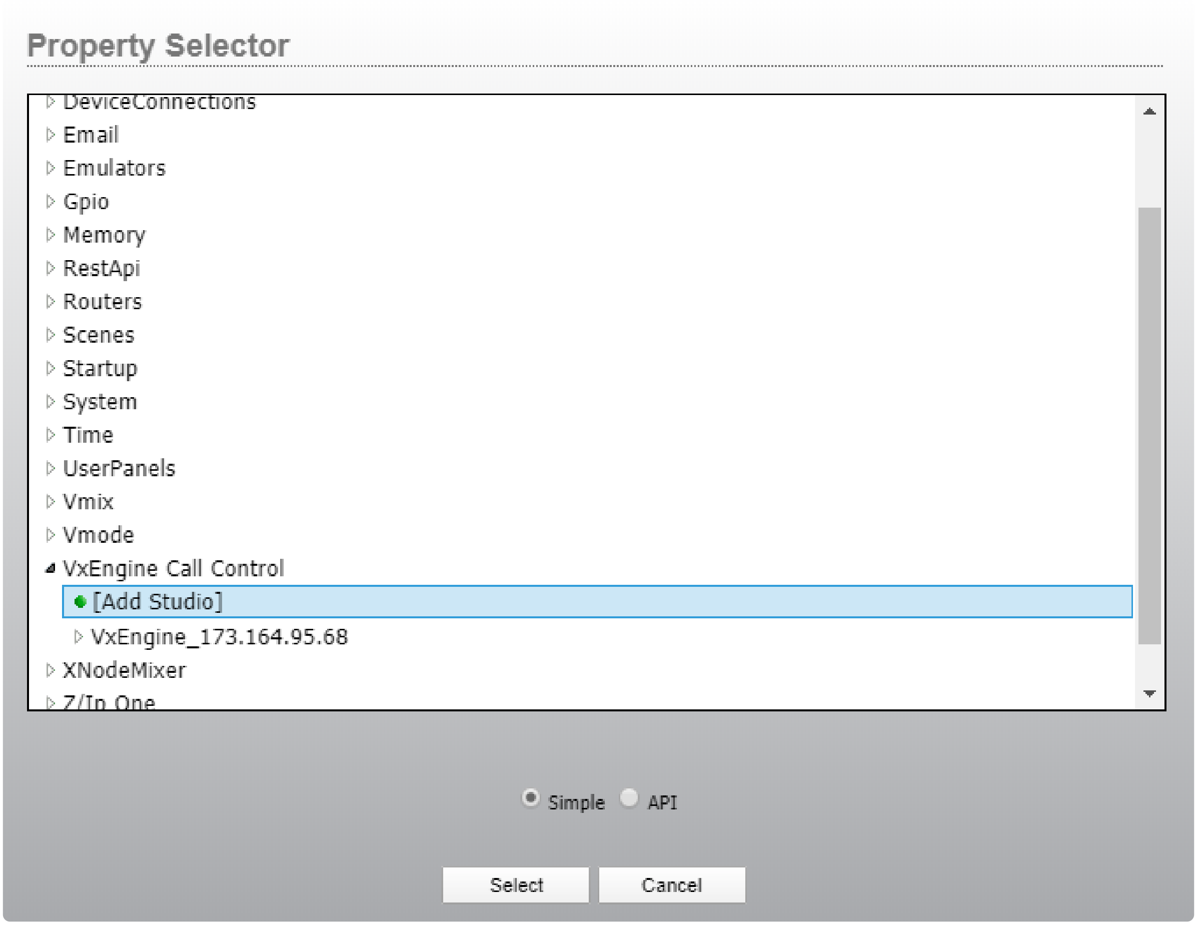
Most of the other settings found on the Configuration pages of the iPort are available via the API.

VX Control Protocol

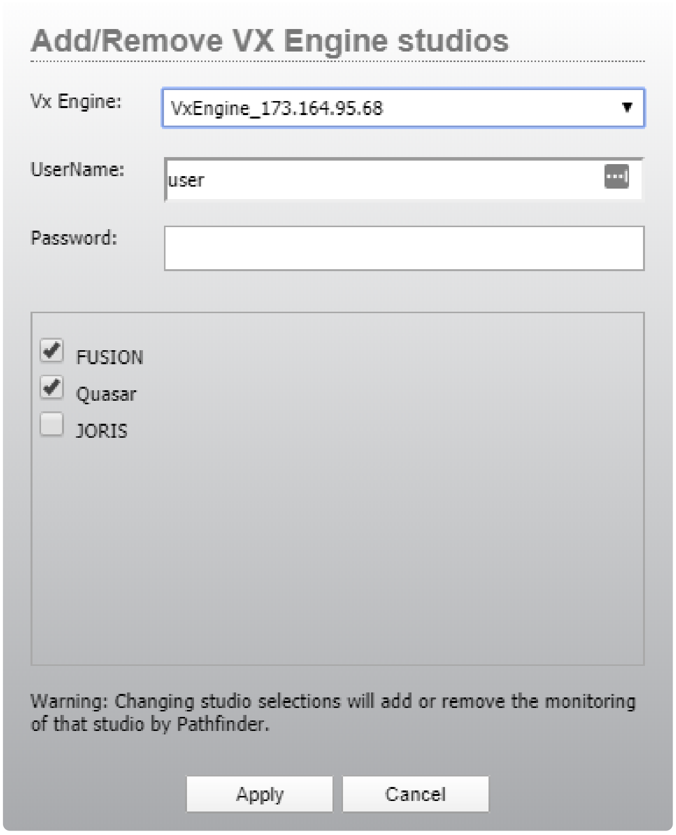
Some additional explanation is needed regarding the VX and its control protocol. This protocol allows Pathfinder Core PRO logic flows and user panel controls to monitor and change Vx calls in a studio. Control points such as lines, studios, call states, comments, and call actions are available to be used by Pathfinder Core PRO via this protocol.

It is important to understand that when you connect to a VX engine on the control port, you have to select which studio to interact with. As a result, if you want to monitor multiple VX engine studios in Pathfinder Core PRO at the same time, a separate TCP connection is required for each studio. Rather than enabling this by default when many customers may not need this functionality, we decided to allow the administrator to enable the studios they wish PathfinderCore PRO to interact with and leave the others disconnected. This conserves resources.

To begin using the VX Control protocol, go to the Logic Flows tab and select a logic flow view. Add a flow and double click on the end point. Within the tree you will see a new branch called VxEngine Call Control.



Clicking on the [Add Studio] branch will bring up a dialog where you can select which studios you want Pathfinder Core PRO to use and monitor.



Add/Remove VX Engine studios

Vx Engine: VxEngine_173.164.95.68

UserName: user

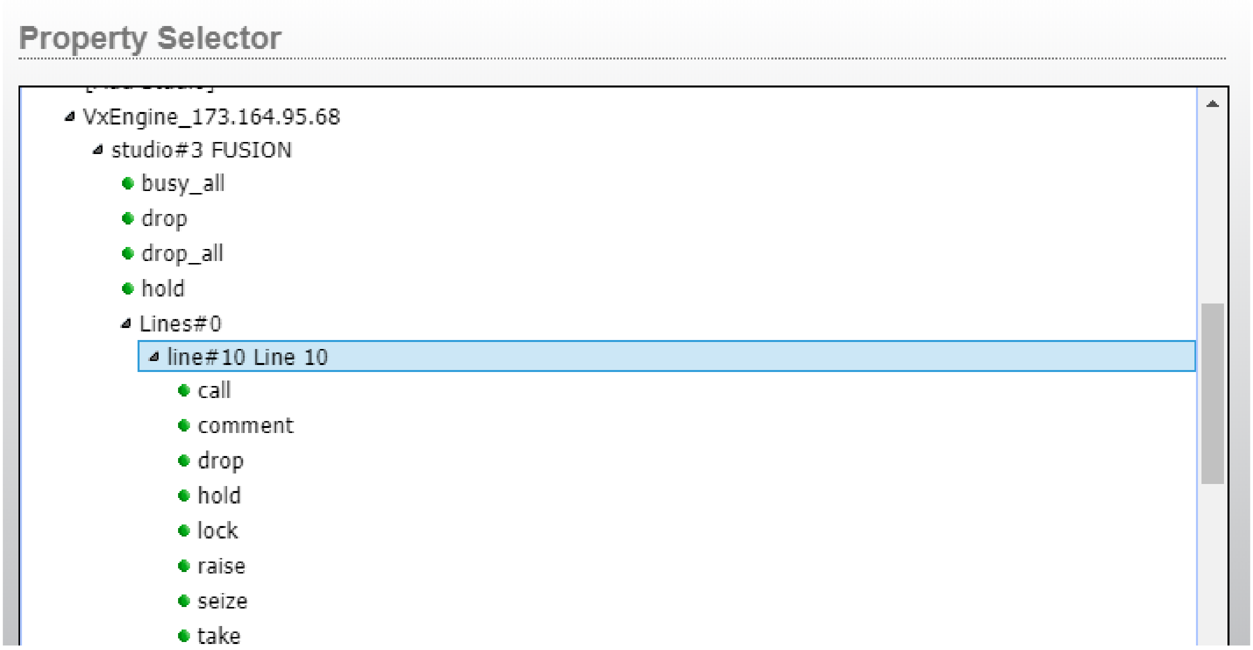
Password:

☑ FUSION
☑ Quasar
☐ JORIS

Warning: Changing studio selections will add or remove the monitoring of that studio by Pathfinder.

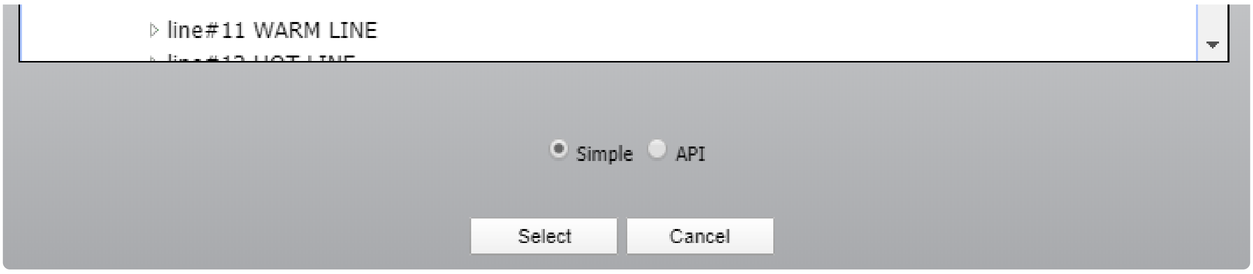
Select the VxEngine whose studios you wish to enable from the drop down and then enable the check boxes for the studios you wish to have available. You need to click apply for each engine whose studios you enable or disable. If you have changed the Lwcp password in the VxEngine, you need to supply the correct password in this dialog as well. If you change the password after adding the studios into PathfinderCore PRO, come back to this dialog, deselect the studios and apply and then reselect them using the new password and apply again. This is not a dialog that is specific to this instance of logic flow editing. You are adding or removing that studio support from the system and so it is generally a one-time change, and does require some cpu resources while adding or removing the studio. We decided to place this dialog within logic flow because that will be where customers will most frequently be interacting with the VX Control protocol data points.

Once added, the studio and property items will then be available in logic flows under the specified engine:



Property Selector

```
◢ VxEngine_173.164.95.68
  ◢ studio#3 FUSION
    ● busy_all
    ● drop
    ● drop_all
    ● hold
    ◢ Lines#0
      ◢ line#10 Line 10
        ● call
        ● comment
        ● drop
        ● hold
        ● lock
        ● raise
        ● seize
        ● take
```

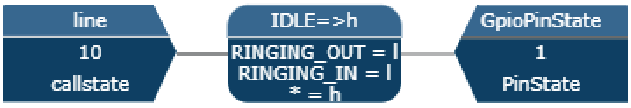


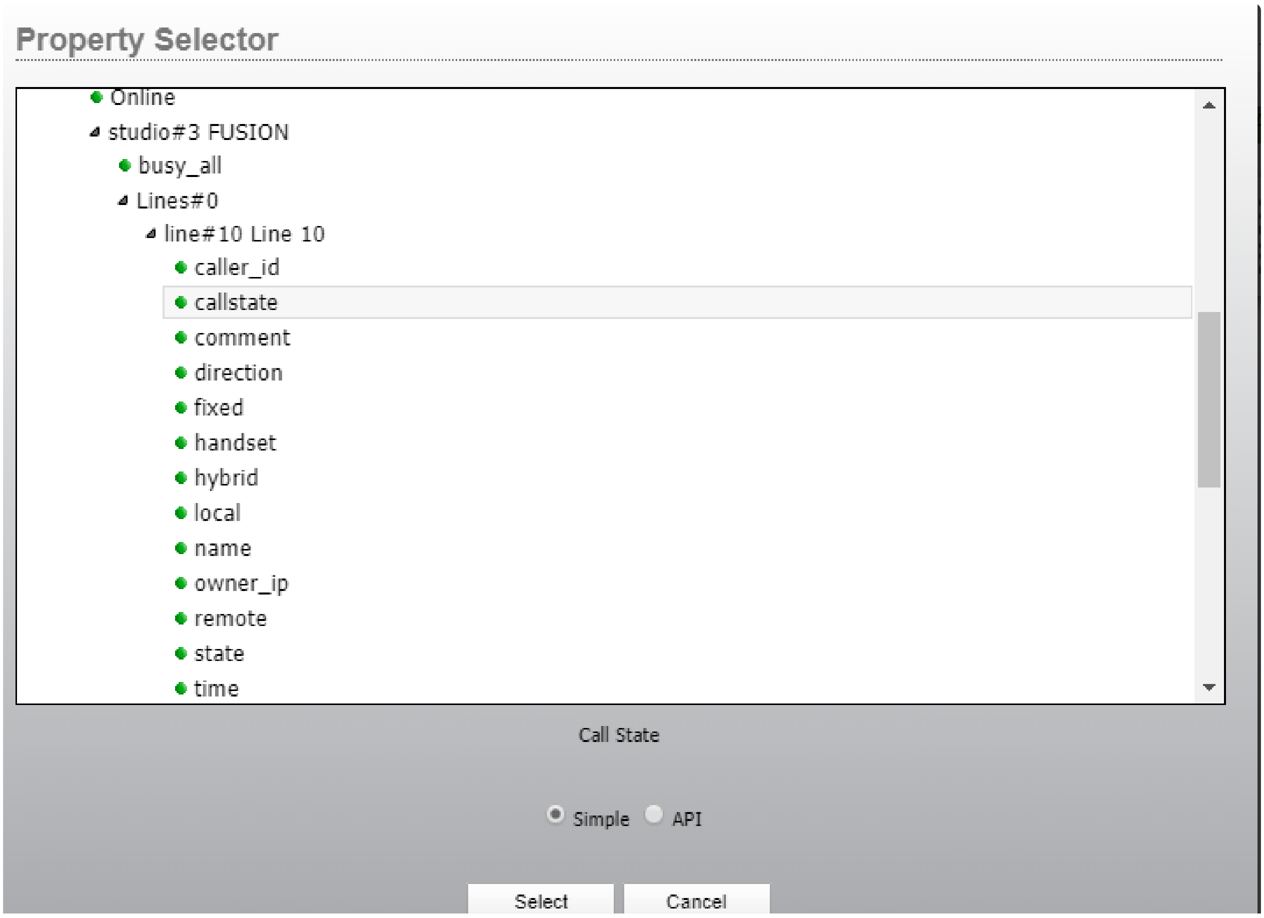
> ⓘ Warning: If you add new studios to an engine, you need to collapse and re-expand the engine branch to see the new studio. Also if you uncheck (remove) a studio, you may need to refresh the logic flows web page before it is removed from the property tree. Removing (unchecking) a studio removes it from Pathfinder's control and monitoring entirely which may affect any other flows that use that studio.

Gpio Ring Example

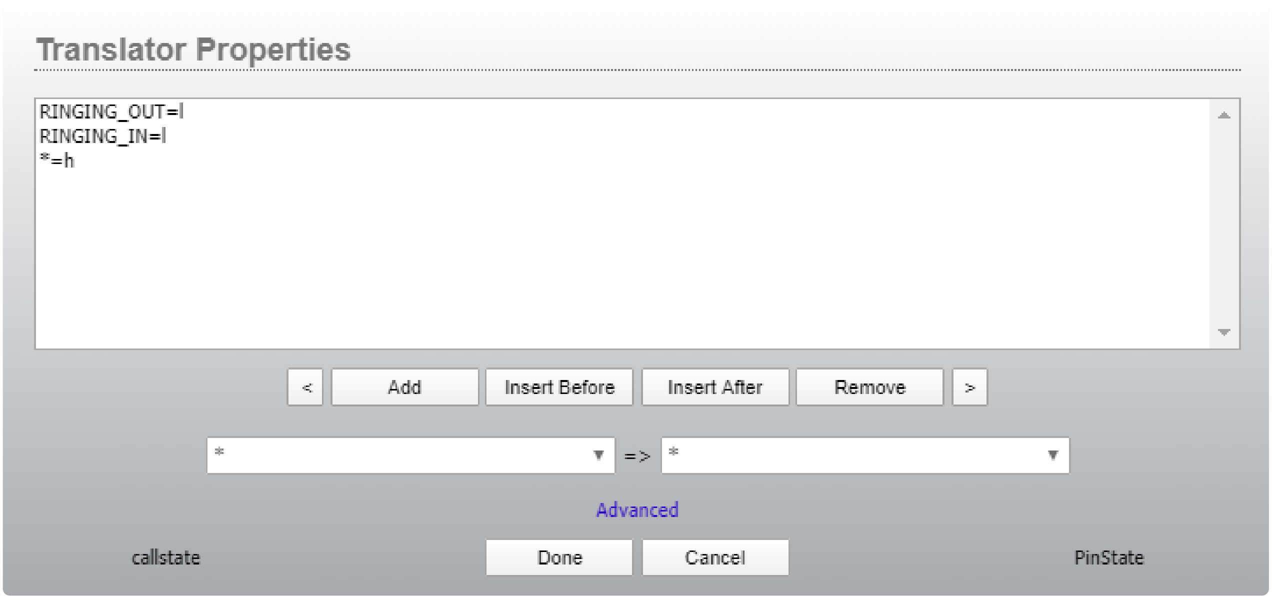
As an example as to how this might be used, you could cause a gpio to light an indicator whenever a given line rings. To accomplish this, you would create a flow where the start point is the line's callstate property and the endpoint is a gpio.

The flow would look like:





For the translation you would create something like:



Translator Properties

```
RINGING_OUT=l
RINGING_IN=l
*=h
```

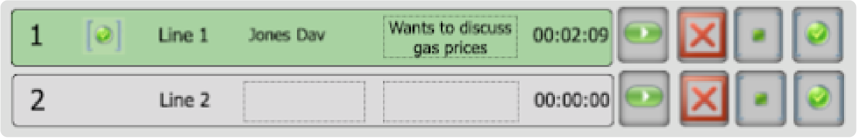
* => *

Additional flows could be added to trip other gpio pins for different lines.

It is important to note that the properties will differ significantly depending on whether you are selecting a start or end point. There are numerous properties that are “read only” or “write only” and so will only display in either the start or end point editing. Selecting a property in the logic flows tree will also display a brief description as to what the property does.

Panel Example

You can also use these properties in user panels. So for example, you could use a label object and bind the label’s background color to the line call state and the textContent to the caller_id field. A button could then be used to answer and drop the calls. One of our support engineers crafted the following using the html 5 user panel designer during our initial testing.



This uses a large label where the call state is bound to the background color, and smaller transparent labels on top of that are bound to the caller_id, comment, and time properties. An image was also overlaid that swaps between multiple images to represent call state. Finally, additional buttons were added for the additional call actions such as seize, drop, hold, etc. Using these items you can develop a user interface to your liking that works with the vx phone system. This also allows you to create a user interface that mixes console fader control, call control, and other user interface items to achieve your specific needs.

Available Properties

Below find a list of the commonly used properties. Additional properties may be available that are not included in this list, so review the logic flow tree and property descriptions within the tree. Items in this table that are marked as true only in the syntax are write only actions where setting the property to true is a

Object	Property	ReadWrite	Description	Syntax
Line	call	W	Call out using a specific hybrid.	<hybridnumber> <numbertocall>
Line	caller_id	R	Caller Id	Text
Line	callstate	R	Call State	List of states
Line	comment	RW	Comment for the Line	Text
Line	direction	R	Direction for the Call	NONE INCOMING OUTGOING
Line	drop	W	Drop a line	True
Line	fixed	R	Is line fixed or selectable	True/False
Line	handset	R	Is call on the handset	True/False
Line	hold	W	Hold a line	ON_HOLD_RE DY ON_HOLD
Line	hybrid	R	Take the next call to the specified Hybrid	List of hybrid ids by name
Line	local	R	Local Phone Number	Text
Line	lock	W	Lock a line	True/False
Line	name	R	Name of the Line	Text
Line	owner_ip	R	IP address of the current line owner	Text
Line	raise	W	Raise a line in priority	True
Line	remote	R	Remote Phone Number	Text
Line	seize	W	Seize a line to reserve it for this	True

			client	
Line	state	R	Line State	List of states
Line	take	W	Take the next call to the specified hybrid	List of hybrid ids by name

Quasar 4 fader panel QSCONN Lwrp Object

Each Quasar 4 fader panel device type has a QSCONN property within the lwrp protocol. If you already have Quasar fader panels in your system, before upgrading to 1.18, you may need to remove then and re-add them in order to get the correct object type. If they are appearing in your devices list as Quasar_4-Fader as opposed to QuasarFader4, you might need to remove and add the panel to PathfinderCore PRO again.

The QSCONN object is accessible via the API tree in logic flows under the Devices\<QuasarFaderPanelDevice>\LwrpInterpreter\LwrpRoot\QSCONN path. It is not a normally used object and so does not appear in the simple tree. It has two properties:

- Master: (RW): Holds the IP address of the master module to which this fader bank is attached.
- Status: (RO): Whether the Fader panel is currently connected or disconnected from the master panel.

The master property may be changed with a new ip address to change what master module the fader panel is connected to. This functionality should be used with care and discussed with support to be sure you understand the ramifications, but it can be used to make a fader module address a different master module dynamically.

Additionally, in the current version of PathfinderCore PRO code and Quasar code, this Fader Module does not push changes to Core PRO. As a result status and master changes made in the web page of the device may not be reflected in Core PRO until the next poll time. Additionally changes made by one Core PRO server of this object may not be detected by a second server until its next poll time. This limitation may

XNodes and FPSTAT

This version adds support for the FPSTAT object in xNodes. The FPSTAT object carries several read-only properties related to temperature, power supply status, network link, etc. It's found in the Simple tree under the DeviceConnections branch and the API tree under the device's LWRP path.

In the current version of xNode software, these parameters require polling to obtain state and do not automatically push changes. By default, PathfinderCore PRO will poll these properties every 15 seconds.

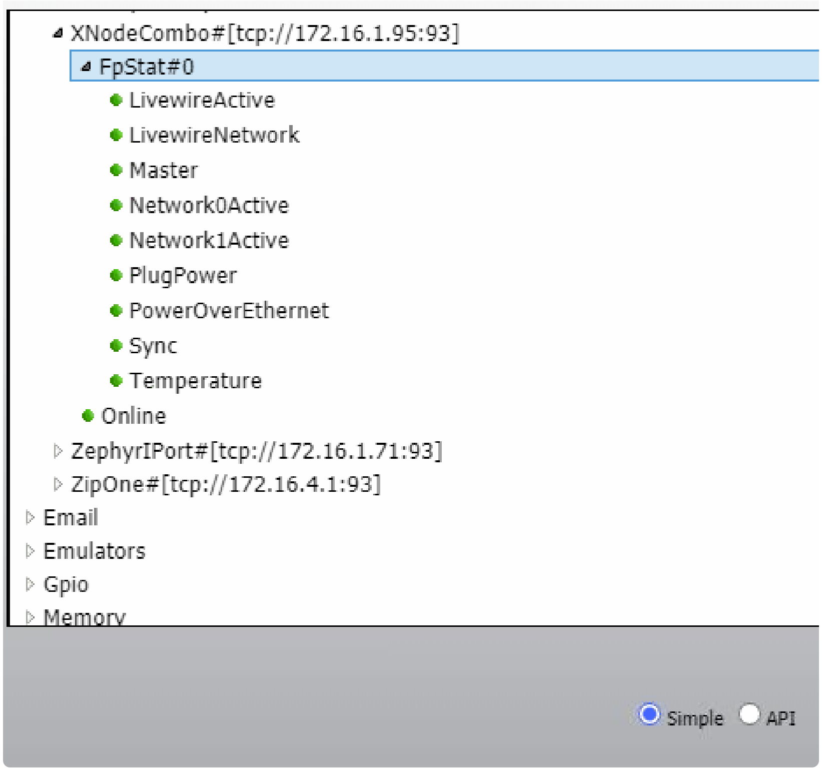
> ⓘ There is an advanced option in the Configuration, Advanced Options settings to change this called FpStatePollRate. You may need to view the default settings to obtain and override its syntax. **FpStatePollRate is a global setting and should only be changed with caution and understanding of the consequences. In networks with many xNodes, decreasing this poll rate can affect the amount of communication that PathfinderCore PRO needs to process.**

The properties available are:

- **Master:** Whether this node is clock master. Without respect to the actual setting. The xNode could be set to allow it to become a master but is not currently. 1 indicates Master.
- **Sync:** Status information regarding the clock sync state. A numerical value of the xNodes SYNC indicator. Blank is no SYNC. (Normal if this device is Master). 12 is fully synced (SYNC is not blinking). 1 indicates furthest away from the synced state. The closer the xNode gets to a fully synchronized state, the higher the number.
- **LivewireActive:** Whether the device is receiving livewire advertisements
- **Temperature:** Temperature in Celcius of the main board
- **Plug Power:** Whether the power supply has power from the IEC plug port.
- **PowerOverEthernet**: Whether the power supply is receiving power from the Ethernet port.
- **Network0Active:** Whether ethernet port 0 is active (this is NET 1 on the xNode).
- **Network1Active:** Whether ethernet port 1 is active (this is NET 2 on the xNode).
- **LivewireNetwork:** Which Nic (0 or 1) is the livewire network

It is important to note that you should use these settings in conjunction with the *online* property and not as a replacement. For example, if all power is lost from the xNode, we will likely not see the Plug Power or POE properties change because we will have lost connection to the object that hosts those properties before we could read the change. So we must rely on the *online* state to know we are no longer connected. On the other hand, if you have both Plug Power and POE power applied and one of them is lost, the status shows online, but we get an alert that one of the supplies is down. Similarly, if you lose connection on the NIC that you are using to monitor this object, then the NIC states in Core PRO for this object will not change. We have to rely on the Online state for that. On the other hand, if you set up the NICs in the xNode to be redundant and one fails, we can detect and alarm this condition.

Property Selector

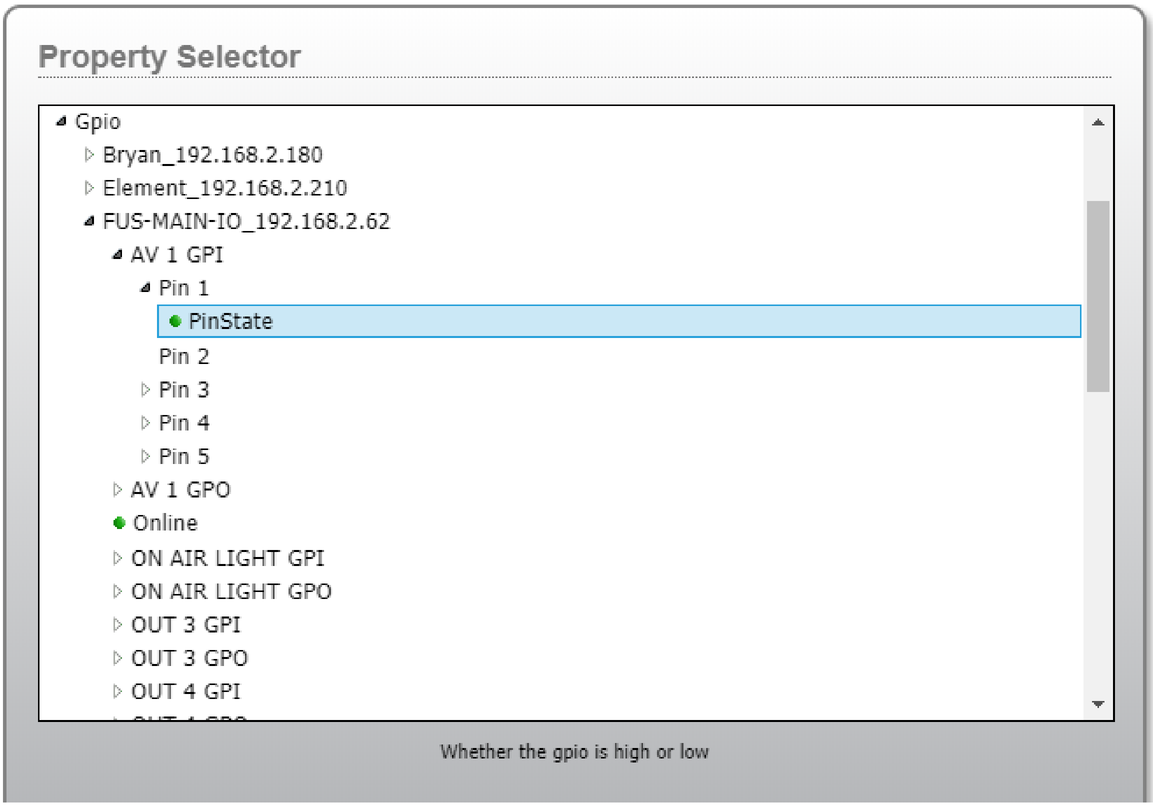


- XNodeCombo#[tcp://172.16.1.95:93]
  - FpStat#0
    - ● LivewireActive
    - ● LivewireNetwork
    - ● Master
    - ● Network0Active
    - ● Network1Active
    - ● PlugPower
    - ● PowerOverEthernet
    - ● Sync
    - ● Temperature
  - ● Online
- ZephyrIPort#[tcp://172.16.1.71:93]
- ZipOne#[tcp://172.16.4.1:93]
- Email
- Emulators
- Gpio
- Memory

Advanced Options

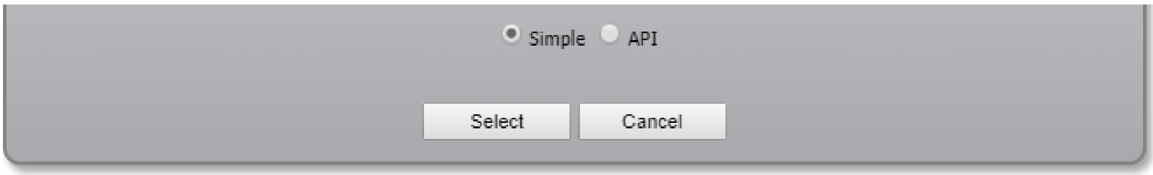
Endpoints and the API Property List

When editing a Start Point or End Point, the property selection dialog has two modes: Simple and API.



- Gpio
  - Bryan_192.168.2.180
  - Element_192.168.2.210
  - FUS-MAIN-IO_192.168.2.62
    - AV 1 GPI
      - Pin 1
        - ● PinState
      - Pin 2
      - Pin 3
      - Pin 4
      - Pin 5
    - AV 1 GPO
    - ● Online
    - ON AIR LIGHT GPI
    - ON AIR LIGHT GPO
    - OUT 3 GPI
    - OUT 3 GPO
    - OUT 4 GPI

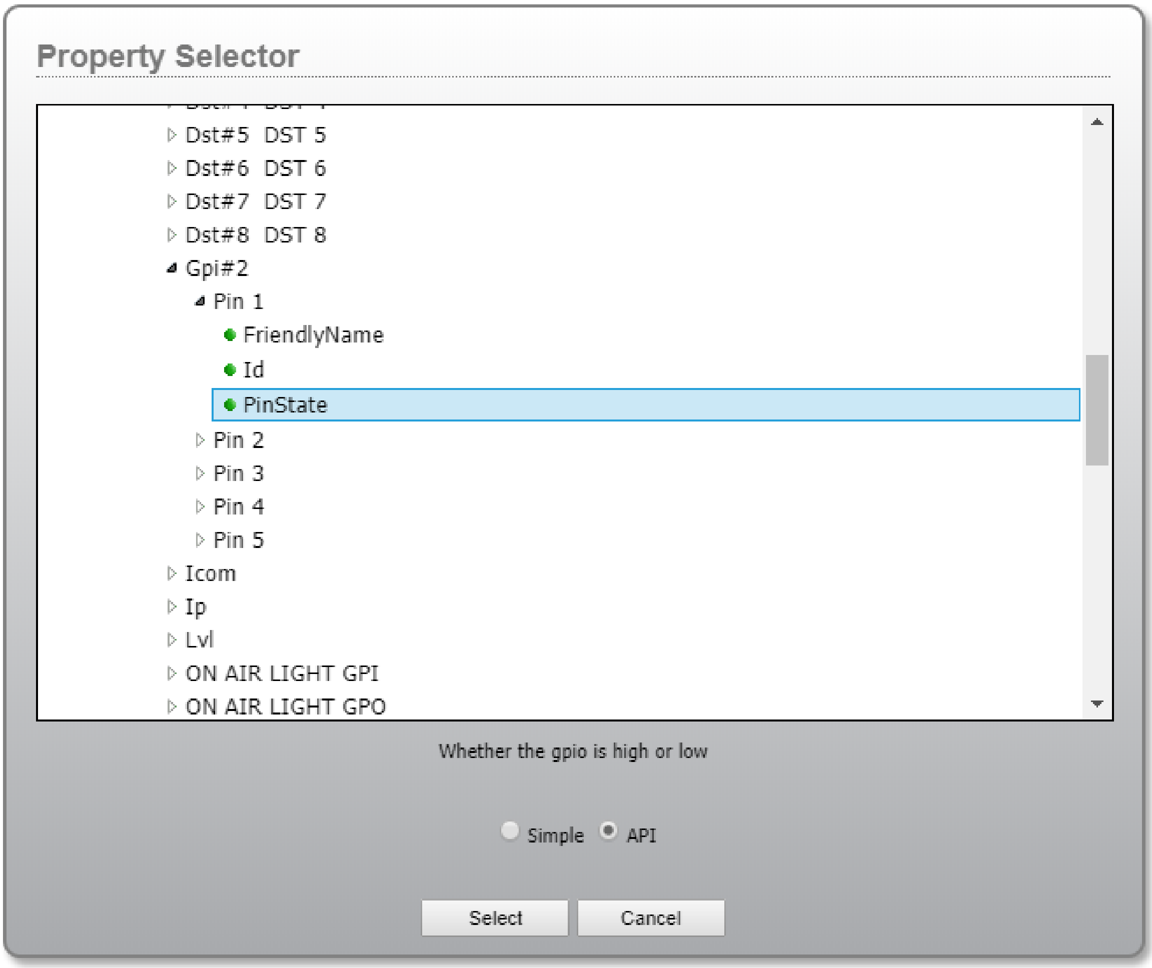
Whether the gpio is high or low


● Simple ○ API

The simple tree is what most users should use. It is designed to present the most commonly used objects and properties organized for quick access. Occasionally there may be a job that requires reaching beyond the normal options, and selecting the API tree will present a much wider array of objects and properties.

Pathfinder Core PRO has an advanced protocol called SapV2 which extends beyond most typical protocols and reaches the level of an API (application programming interface). It allows virtually complete control over the system. This API is the protocol used for all internal messaging between system services. If you want to learn more about this API, see the SapV2 appendix.

For this API to do its job successfully, almost every property and object must be available. The API section of the property selection dialog exposes the object and property tree available via that API.

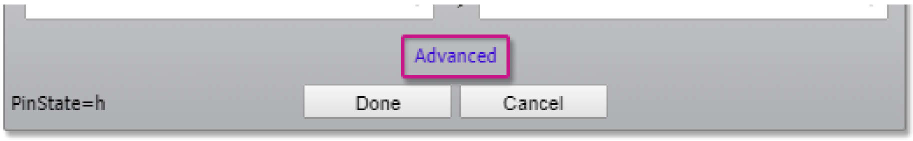


Whether the gpio is high or low

○ Simple ● API

Translator

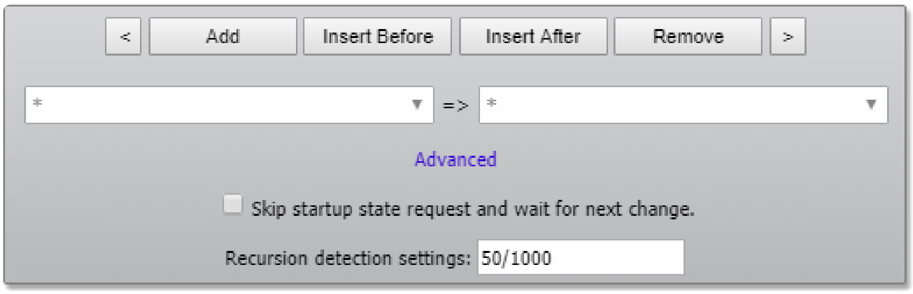
To configure advanced options on a Translator, click the **Advanced** link on the *Translator Properties* window.



PinState=h

Translator Skip Startup State

In the expanded Translator dialog is the Advanced *Skip startup state request and wait for next change* option:



* => *

□ Skip startup state request and wait for next change.

Recursion detection settings: 50/1000

By default, when the system starts (or when a Flow is applied to the system) the Flow asks for the current state of the Start Point and then applies the correct result to the End Point. In most cases this is desirable, but not always. For example, what if the Logic Flow in question was a latching GPIO on the Start Point and an automation system that gets tripped on the End Point?

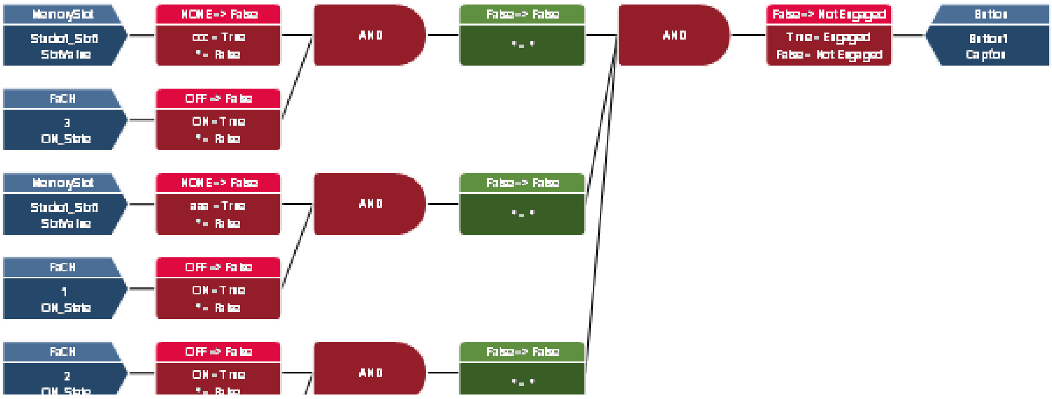
If we needed to restart Pathfinder during that event, we would not want the automation system to be tripped again. Checking the *Skip startup state request and wait for next change* option overrides the default behavior and forces the system to wait for the next change that would cause the Flow to execute.

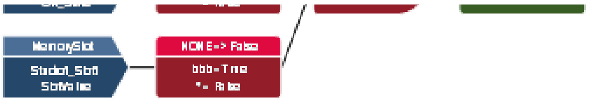
> ⓘ Important Note: When designing a system, it is important to test your logic before it goes live on air in case of a system restart and/or device restart.

Another highly relevant example for where the Skip Startup State option becomes important is with the commonly-used EAS example shown earlier in this section.

Recursion Detection

If the Logic Flow system detects a given Translator being activated more than 20 times in 1000 milliseconds, it will assume there is recursion involved and the Translator will be disabled to prevent excess CPU load. In some situations, this can create a *False* positive. For example:

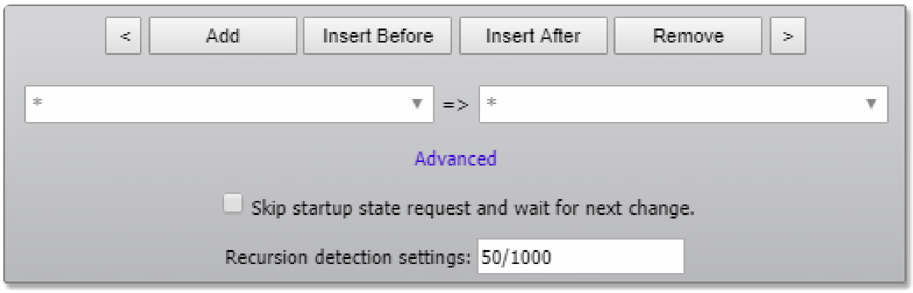




In this Flow, the same memory slot is used for multiple entry points to a Flow, so a change to that memory slot could cause multiple pass-throughs of some Translators. If this Flow were to grow to a much larger variation with many more entry points of the same memory slot, it is possible to trip the recursion detection incorrectly.

> ⓘ Important Note: If a Translator gets disabled via the recursion detection, a log message is generated and an attempt will be made to send an email message to the critical event email address if configured.

Recursion detection threshold settings are modified in the expanded Translator dialog.



□ Skip startup state request and wait for next change.

Recursion detection settings: 50/1000

This value includes two parts separated by a forward slash. The first value indicates the analysis frequency; the second indicates the analysis duration in milliseconds. For example, typing **50/1000** will disable the Translator if it is being analyzed more than **50** times per **1000** milliseconds.

Additionally, there is an API property that that can change the default recursion settings for all new Translators:

```
Set LogicFlows#0 DefaultRecursionSettings="50/1000"
```

This setting should be modified with extreme care as it will also update any existing Translators that are currently at the default settings. This can be CPU intensive and require a significant amount of writing to disk. Please make a backup before changing this value.

Combining the * Wildcard with Other Values

It is common to have a conversion list item in a translator of:

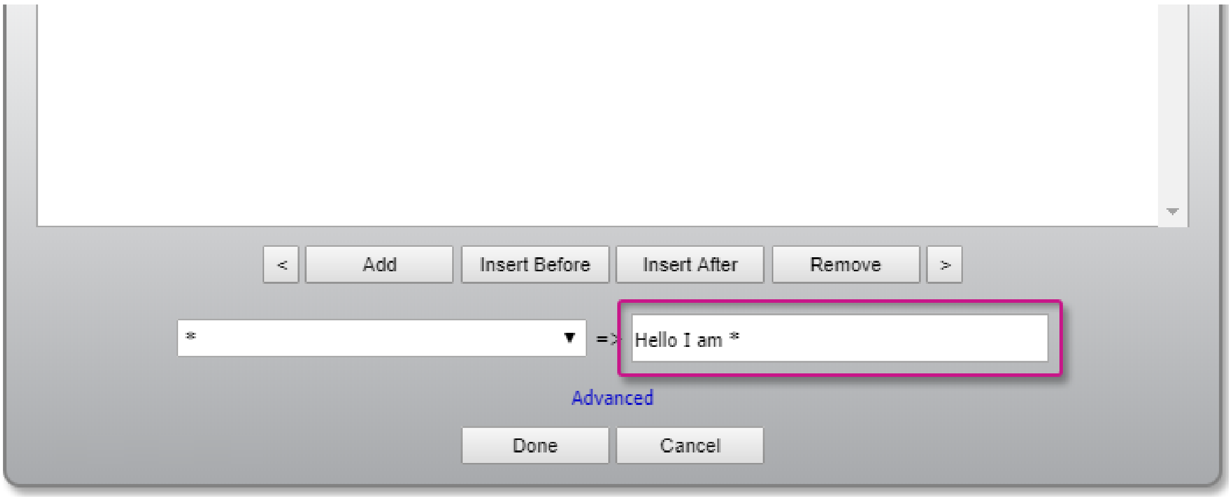
=

However, the ***** on the *ToValue* side does not need to be by itself. For example, you could have:

***=Hello I am ***

Translator Properties

*=Hello I am *



* => Hello I am *

In this case, the input value would replace the ***** in the output value. If the input value is **Dan**, the *ToValue* example above would output the string:

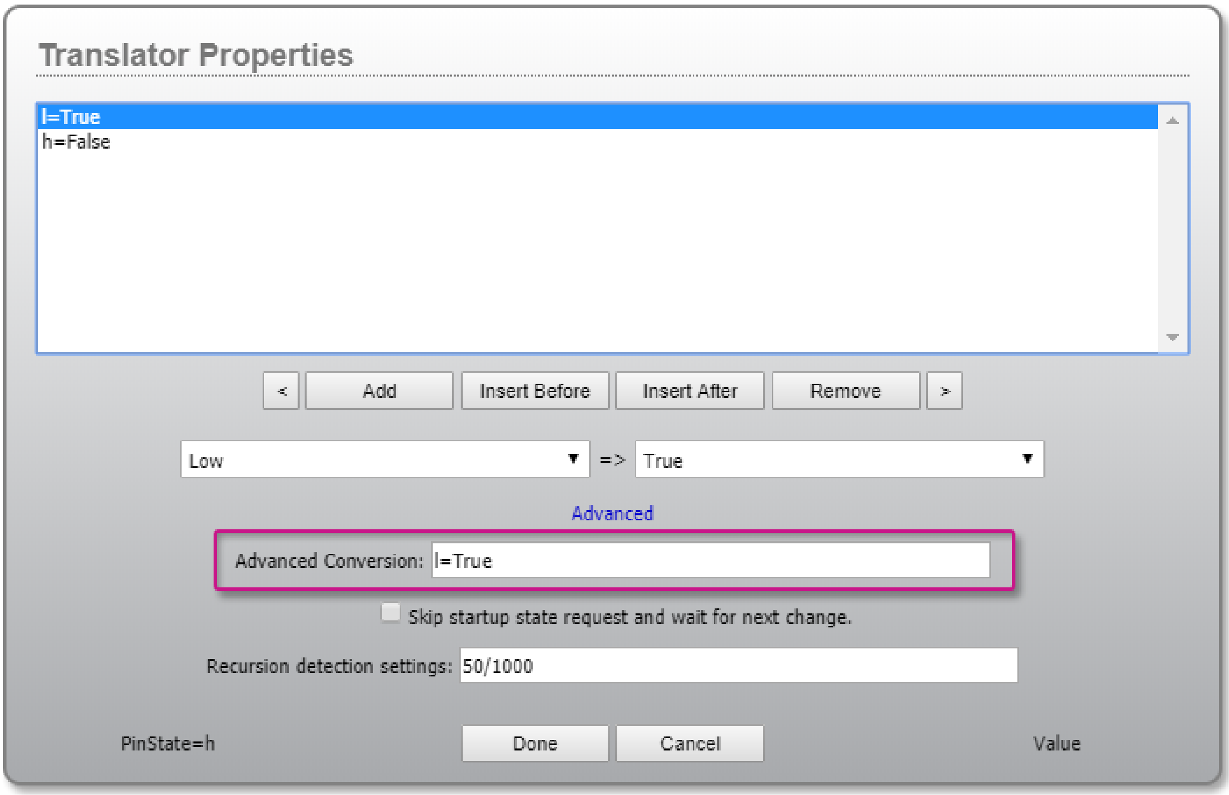
Hello I am Dan

This can provide similar functionality natively in the Flow translation as string builder memory slots.

***** cannot be used in the middle of data on the "from" side. Use regular expressions for complex matching if needed.

Advanced Conversion Field

The **Advanced Conversion** field, available in the expanded Translator dialog, allows users to bypass some of the helper dialogs and enter conversion statements directly.



Translator Properties

l=True
h=False

Low => True

Advanced Conversion: l=True

□ Skip startup state request and wait for next change.

Recursion detection settings: 50/1000

PinState=h

Value

While bypassing the helper dialogs can be useful in certain unique scenarios, those dialogs present information that is helpful when selecting a property but may or may not be the underlying value needed in the conversion. For example, when using the *CurrentSourcePath* property, the conversion list and normal drop-down list will show you the friendly name of the source. However, the actual conversion list item needs the **uri** based **pathio** which looks something like:

"tcp://172.16.1.53:93?l=SRC&d=src&i=6&t=aaudio"

If you are uncertain about the value requirement, create an example of what you are trying to do using the normal conversion list selection methods and then select it in the Advanced field to see the actual values that need to be applied. If you have questions, contact support.

> ⓘ Important note: Incorrectly entering data into this field can cause unpredictable results.

Conversion text syntax is:

<FromValue>=<ToValue>

For example:

l=True

If the *From* or *True* value needs to include a literal *equals* sign, then enclose the value in quotes. The *From* value will be **abc=frz** in this example:

"abc=frs"=True

If **<NoChange>** is applied to the *ToValue* portion of a translation, the output of the translator will not be changed. For example:

AAA=True

BBB=<NoChange>

***=False**

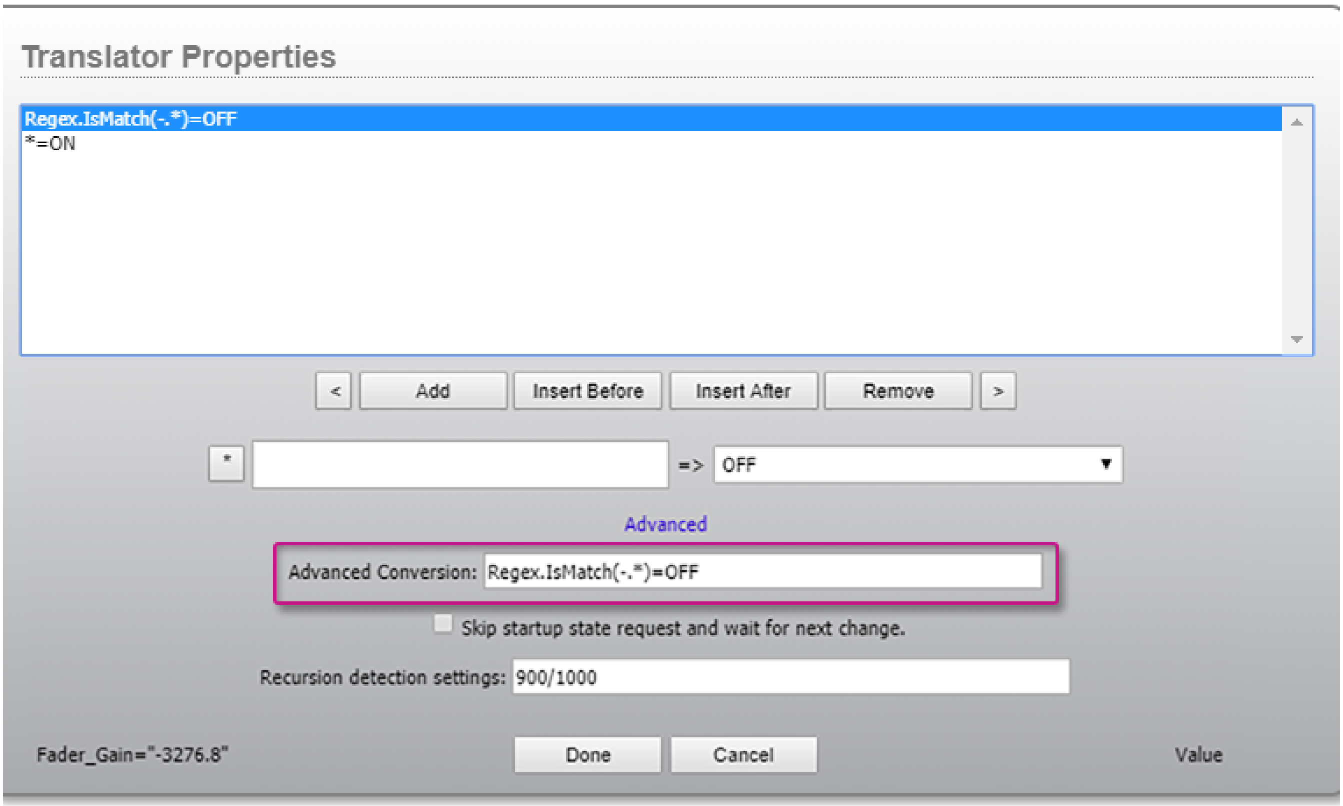
In this case, an input *FromValue* of **AAA** will result in **True**, **BBB** will not cause anything to change on the output of the translator, and any other *FromValue* will result in **False**.

In many cases, a **<NoChange>** is not required because you can just opt not to supply the input which will result in nothing happening when that input occurs. But in a situation where you are using the ***** wildcard as a catchall but want to explicitly *exclude* a value from making a change, this can be used.

> ⓘ Important Note: **<NoChange>** is not applicable to the *FromValue* side of a translation and will be interpreted literally if entered on the *FromValue* side.

Translation Regular Expressions

Regular expressions as described in the *Watchers and Regular Expressions* section in the *Device Emulators* chapter can also be used in the *FromValue* portion of a conversion list item. Regular expressions are added to the *Advanced Conversion* field as follows:



Translator Properties

Regex.IsMatch(-.*)=OFF
*=ON

* => OFF

Advanced

Advanced Conversion: Regex.IsMatch(-.*)=OFF

□ Skip startup state request and wait for next change.

Recursion detection settings: 900/1000

Fader_Gain="-3276.8"

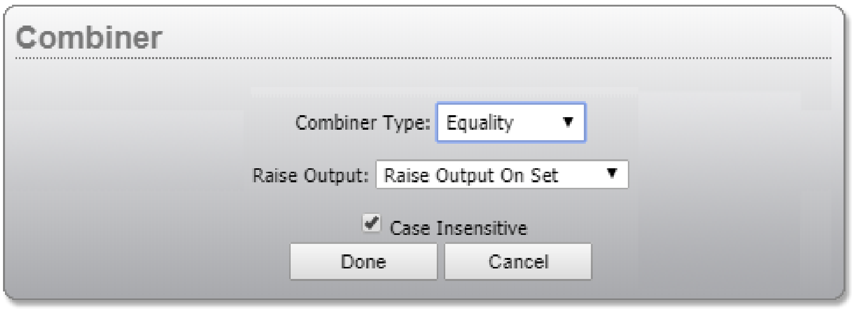
Value

In this example, we are using the **IsMatch** property. *Fader_Gain* values that are not negative will result in a button turning *On;* otherwise, the button indicator will be *Off*. Because *Fader_Gain* generates significant data as it slides, we have also increased the recursion detection value to prevent this Flow from getting disabled.

> ⓘ Important Note: Using regular expressions can be slower and more CPU intensive. Use them only where needed and use the normal translations elsewhere.

Combiner Raise Output Property

Over the course of the examples above, you may have noticed an additional property in Combiners that we have not discussed yet called *Raise Output*.



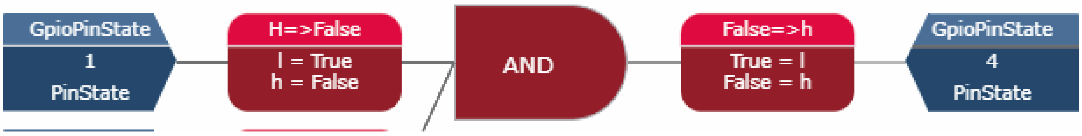
Combiner

Combiner Type: Equality

Raise Output: Raise Output On Set

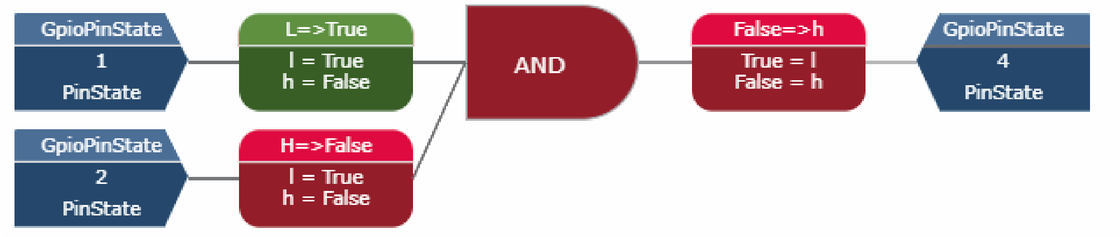
☑ Case Insensitive

The values for this property are *Raise Output on Set* and *Raise Output on Change*. The effects of this property are subtle but are important to understand:



GpioPinState
2
PinState
H=>False
l = True
h = False

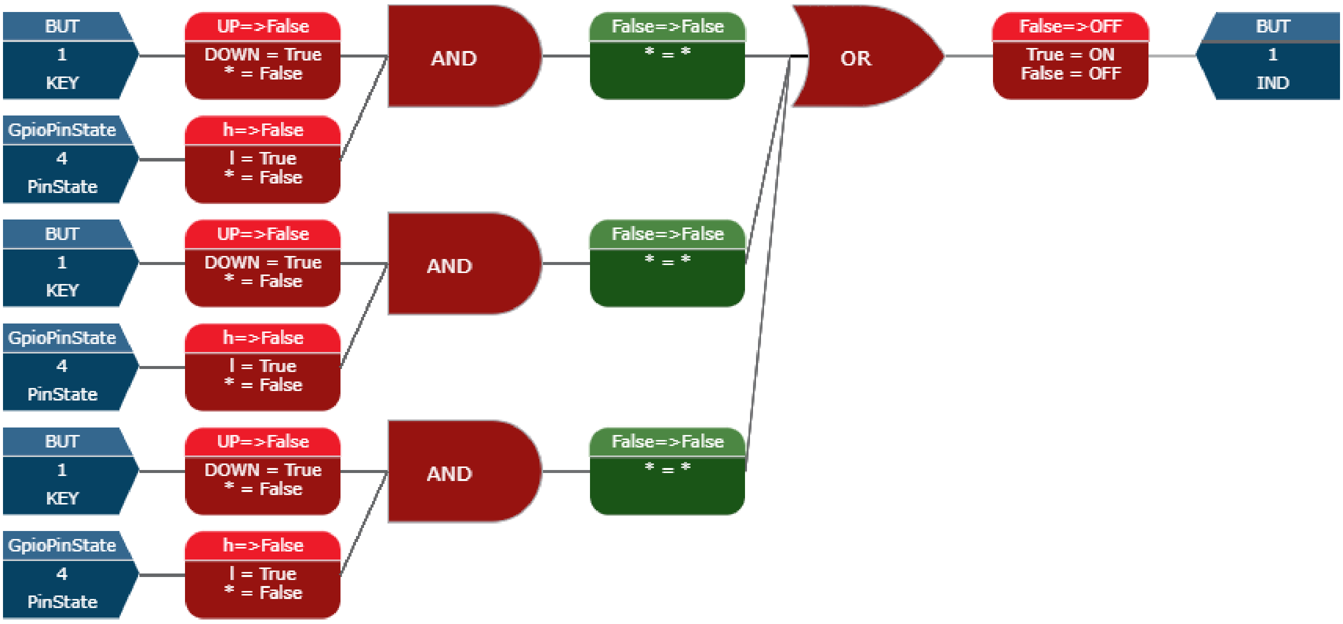
In this Flow, if both pin 1 and pin 2 are *Low*, then pin 4 will be set *Low*. If either pin 1 or pin 2 is *High*, pin 4 will be set *High*. If the Combiner is set to Raise Output on Set, then any time either pin 1 or pin 2 is changed a message will be sent to set pin 4 to the corresponding value. So, if we assume pin 4 is currently *High* or *Low* and we set pin 1 to *Low* when pin 2 is still *High*, a message will get sent to pin 4 to go *High*.



However, if the setting is set to Raise Output on Change, no message will be sent to pin 4. When both pin 1 and 2 were *High* the resulting output of the And Combiner is *False*. When pin 1 goes *Low*, the resulting output of the And Combiner is still *False*. Its output value has not changed and so no message is sent.

The difference is subtle, but essentially with “Raise Output on Set”, a change message will be sent to the End Point any time the input of the Combiner changes even if it evaluates to the same output the Combiner was at before, whereas with “Raise Output on Change”, a change message will only be sent when the Combiner’s output changes.

Let’s look at another example of when Raise Output on Change could be useful:



In this example, GPIO pin state 4 appears at each pair of And Combiners on the left-hand side of the Flow. The buttons are button 1 of three different LCD panels. If all the Combiners in this Flow are set to Raise Output on Set, then the resulting End Point may get set 3 times when GPIO pin 4 changes. This is because a change to the pin would get fully passed through the Flow for each instance where it appears as a Start Point. However, if the left And Combiners are all set to Raise on Change, then the output will only get set for Combiners where the pin state change actually causes a different state on the output of the Combiner. This can greatly reduce load and improve performance in large and complex systems.

It is also important to note that this property is a new variable that must be stored in the backing storage. For backward compatibility with older software versions, Raise on Set will store the same way it historically has. However, any Flows that are changed to use Raise on Change will not be loadable by older software versions that do not support this property.

For advanced users, it is possible to change all of the Combiners to a particular setting using the port 9600 API. Each Logic Flow View has a write-only property called ChangeAllCombinerRaiseOutput. Using this property, all Combiners in a View and that View’s sub-Views can be set to the same value. This is useful for changing many Combiners at once but should be used with caution. It is recommended that a backup be taken before using this api command. For example:

```
set LogicFlows#0.LogicFlowFolder#yyy ChangeAllCombinerRaiseOutputSettings=RaiseOutputOnSet
set LogicFlows#0.LogicFlowFolder#yyy ChangeAllCombinerRaiseOutputSettings=RaiseOutputOnChar
```

See the API reference in Appendix 2 for details on how to use these commands.

Simple Tree Commonly Used Properties

Providing a list of every object and property in the system is beyond the scope of this documentation. One of the reasons we provide the description within the property selection dialog is to make the software self-documenting. However, it is useful to describe some of the more commonly used properties here if for no other reason than to generate ideas for how to use Flows within your environment. While this list is not inclusive, we encourage you to spend some time browsing the tree and examining the various available properties and their descriptions. And as always, our support staff stands ready to assist. Remember that due to the read/write nature of some of the properties, they may only appear and be available for use with a Start Point or an End Point.

- Audio Alarms
  - **AlarmState**: Once an audio alarm has been configured, this property exhibits the current state of the alarm; it can be Unknown, AudioPresent, AudioSilent, or Clipping; logical decisions can be made when this state changes
  - **LvlState**: When a silence alarm is configured, Pathfinder Core PRO requests updates whenever the threshold value is passed for more than 250ms; this property carries the current state regardless of the current alarm countdown
- Buttons
  - **BackColorOn**: Used to set the backcolor value when the button is on
  - **BackColorOff**: Used to set the backcolor value when the button is off
  - **Ind**: Used to set the indicator of the button on, off, or in a variety of flash states
  - **Text**: Used to set the Text value of a button
  - **Key**: This property will be UP or DOWN depending on whether the button is pressed or released
- Console
  - **ShowProfId**: The id of the show profile loaded on the console; this can also be used to change the

currently loaded show profile

  - **ShowProfName**: The name of the show profile loaded on the console
  - **Asg_PGM1-4**: Used to set or determine whether a specific console fader is assigned to any of the 4 program busses
  - **Asg_Prev**: Used to set or determine whether a specific console fader is assigned to the preView buss
  - **Fader_Gain**: Used to change the gain of a fader
  - **MUTE_State**: Used to set or determine if the mute state on a fader is engaged
  - **OFF_But**: Whether the Off button on a fader is up or down.
  - **ON_But**: Whether the On button on a fader is up or down
  - **ON_State**: Used to set or determine if a console fader is on or off
  - **src_lwch**: Carries the Livewire channel number loaded to a fader

  - **talkback**: Used to engage or sense the engagement of the talkback button on a fader
  - **tt_cr**: Used to engage or sense the engagement of the talk to control room button
  - **tt_st**: Used to engage or sense the engagement of the talk to studio button
  - **tt_prev**: Used to engage or sense the engagement of the talk to preView button
  - **IND**: Used to change the indicator state of user definable buttons on the console
  - **KEY**: Used to sense the key state of certain user definable buttons on the console
- DeviceConnections
  - **Connected**: This property can be used to sense and react to connection problems with any of the Axia devices
  - **Online**: This property senses whether a device is responding to commands
- Email
  - **Body**: Used to change the body text of an email message
  - **Subject**: Used to change the subject of an email message
  - **Send**: Used to send a predefined email message
- Emulators
  - **Triggered**: Used to determine when a Generic emulator’s watcher detects a defined input value
  - **ToSend**: Used to send data out a generic emulator port
- **Gpio**
  - PinState: Used to change or sense changes on a GPIO pin
- Memory
  - **SlotValue**: Used to sense changes or change the value of a memory slot
  - **Trigger**: Used to change the value of a latching memory slot from *True* to *False* or *False* to *True*
- Routers
  - **CurrentSourcePath**: Used to sense or activate a route change on a specific destination
  - **CurrentChannelNumber**: Used to sense the Livewire channel number assigned to a destination. This property can also be used to make router changes using Livewire channel numbers for the

source

  - **CurrentSourceName**: This property is updated with the name of the source that is currently routed to the selected destination
- Scenes
  - **IsActive**: This property will be either *True* or *False* depending on whether all states in the scene are currently active
  - : Setting this property to *True* will make Pathfinder Core PRO initiate all of the changes in a scene
- Startup
  - **StartupFileProcessed**: This property can be used to set up certain states after Pathfinder Core PRO has been restarted

- Time
  - **Enabled**: Used to enable or disable a timer. Logical decisions can also be made depending on whether a timer is enabled or disabled
  - **Elapsed**: Becomes *True* when a timer elapses
- VMIX
  - **Gain**: Used to sense or make changes to a VMIX fader’s gain setting
  - **State**: Used to turn VMIX channels on or off and sense the same state changes
  - **TimeDown**: Used to sense or make changes to the length of the fade-out when a VMIX fader is turned off
  - **TimeUp**: Used to sense or make changes to the length of the fade-in when a VMIX channel is turned on
  - **Master_Gain**: Used to sense or make changes to the master gain of a VMIX submixer
- VMODE
  - A**UDIO_MODE**: Used to sense changes to the audio mode state of a VMODE IO
  - IN_SELECT: Used to sense changes to the IN_SELECT state of a VMODE IO

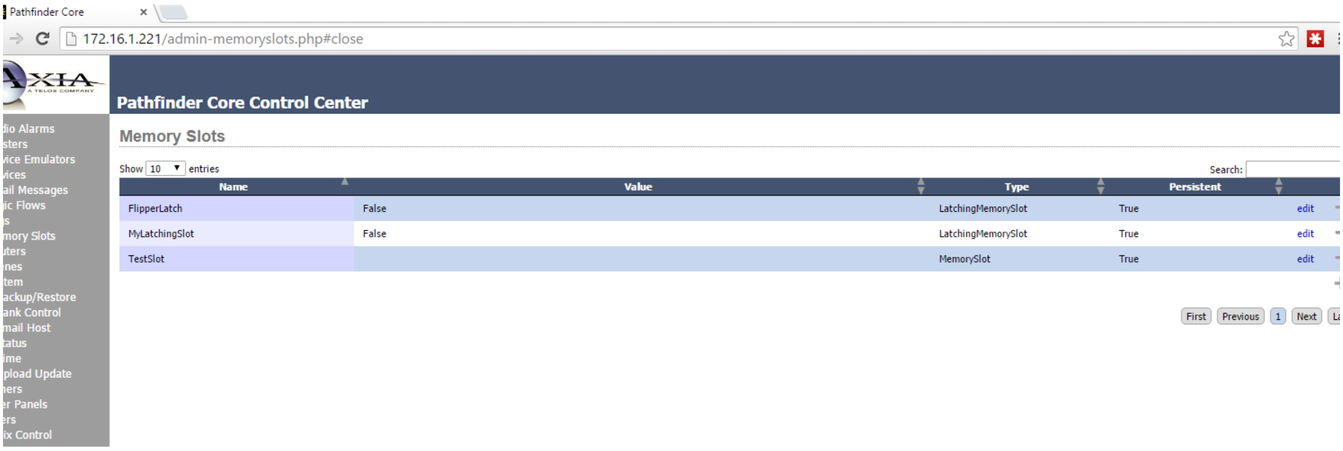
Flows, Flows, and More Flows

While this chapter has reviewed the details of Logic Flows, many of the other chapters of this manual will touch more on their uses as we detail how the aspects of the system functions and therefore what capabilities may be used.

Memory Slots

Overview

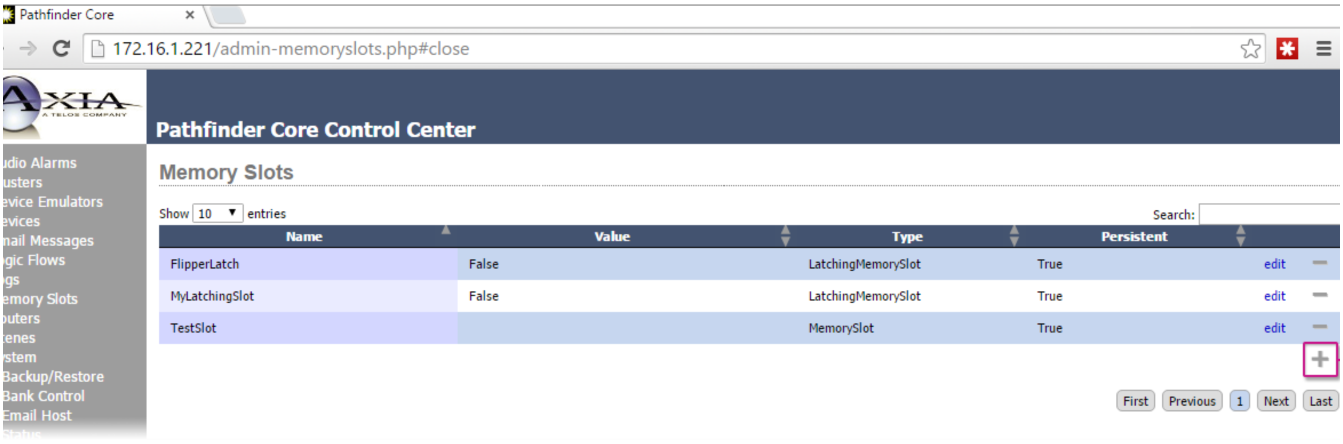
Pathfinder Core Pro's memory slots provide a named location for storing data that can be used in Logic Flows. You can create and view memory slots via this link. You can also monitor the slot’s changes in real-time and alter the value of a memory slot directly from this page.



Memory Slots

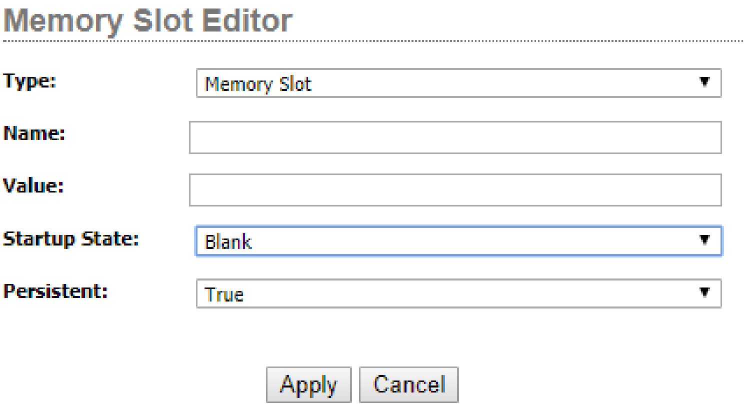
Name	Value	Type	Persistent	
FlipperLatch	False	LatchingMemorySlot	True	edit
MyLatchingSlot	False	LatchingMemorySlot	True	edit
TestSlot		MemorySlot	True	edit

Add a memory slot by clicking the **plus** icon on the bottom-right corner of the memory slot list.



Memory Slots

Name	Value	Type	Persistent	
FlipperLatch	False	LatchingMemorySlot	True	edit
MyLatchingSlot	False	LatchingMemorySlot	True	edit
TestSlot		MemorySlot	True	edit



Memory Slot Editor

Type: Memory Slot

Name:

Value:

Startup State: Blank

Persistent: True

Select the memory slot type you wish to create from the Type drop-down. The available properties will change depending on the memory slot type. Several of the memory slot properties are present in all or most of the memory slots as described here.

Common Memory Slot Properties

Property Name	Description
Name	Each memory slot must have a unique name with the system

Value	Except in cases where the memory slot's value is derived from other information, the value property allows you to define an initial value for the memor slot and/or to change the current value
Startup State	The startup state defines how an initial value for th memory slot is obtained when the system first starts; for details see the section on startup states.
Persistent	Defines whether the memory slot will be retained between restarts or is just a temporary memory slo the default and by far most used option here is persistent

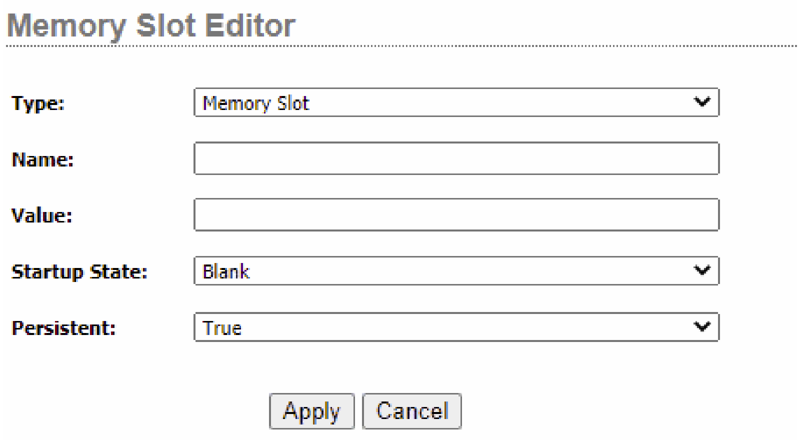
Each memory slot may also have additional properties according to their type as described below.

Memory Slot Types

Pathfinder Core PRO has six different types of memory slots, each with its own options and properties.

Memory Slot

This is the traditional memory slot into which any kind of data may be stored. Its parameters are shown below:



Memory Slot Editor

Type: Memory Slot

Name:

Value:

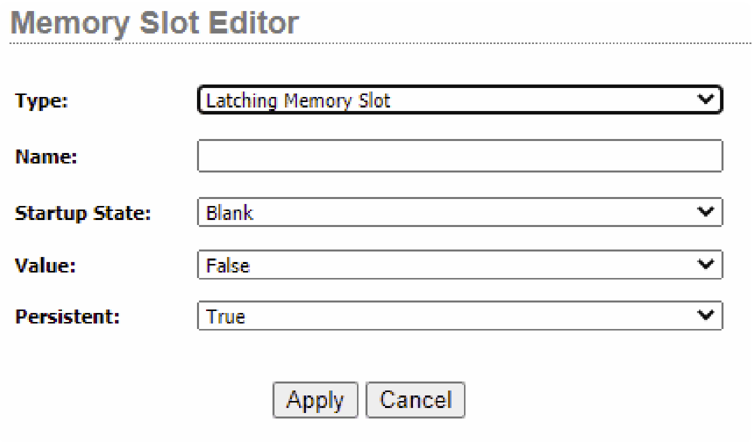
Startup State: Blank

Persistent: True

All parameters for this memory slot are described in the common properties section above.

Latching

The Latching memory slot can only have a value of True or False.   It has a write-only property called Trigger which will change the memory slot's value. Using a Logic Flow end-point, you can set the trigger property to true and the slot value will toggle from true to false, or from false to true depending on its current value. See the latching example in the Logic Flows section of this manual for more details. The configuration properties for this slot type are shown below:



Memory Slot Editor

Type: Latching Memory Slot

Name:

Startup State: Blank

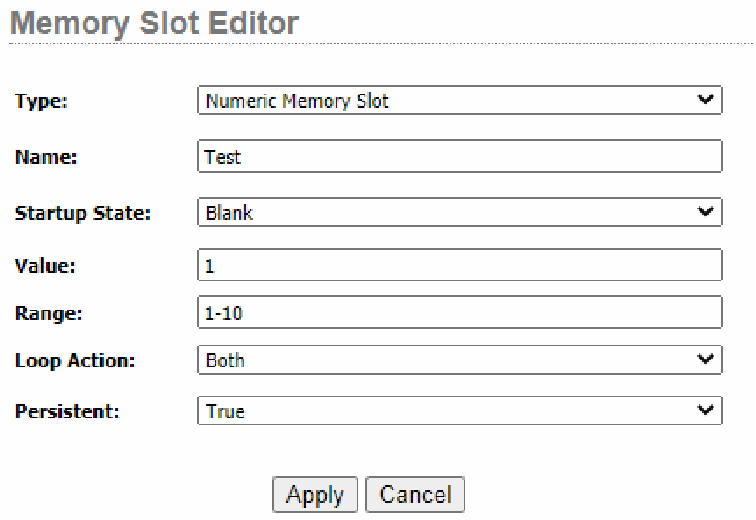
Value: False

Persistent: True

All parameters for this memory slot are described in the common properties section above. However, note that the value property field becomes a drop-down that only allows a value of True or False.

Numeric

Numeric memory slots allow you to store, retrieve, increment, and decrement a numeric value. Its write-only properties are increment and decrement, and can be used in Logic Flows to change the value of a memory slot. This type of memory slot can also accept a range to specify what happens when the value increments or decrements outside of the range using the loop action option. Its parameters are shown below:



Memory Slot Editor

Type: Numeric Memory Slot

Name: Test

Startup State: Blank

Value: 1

Range: 1-10

Loop Action: Both

Persistent: True

In addition to the properties outlined in the common properties section above, this memory slot type includes two additional properties: **Range** and **Loop Action**. If the range field is left blank, then the numeric memory slot may accept any numeric value. If you apply a range by filling the range field with a low and high value separated by a hyphen, then increment and decrement will not exceed the specified range. The Loop Action property only works in conjunction with a range and does nothing if no range is specified. The loop action field defines what happens when you try to increment or decrement beyond the specified range as follows:

- **None** = Incrementing beyond the max or decrementing below the min will do nothing and no change will be applied to the Value.
- **Forward** = Incrementing beyond the max will cause the value to loop back to the min, but decrementing below the min will make no changes.
- **Backward** = Decrementing below the min will cause the value to loop to the max, but incrementing

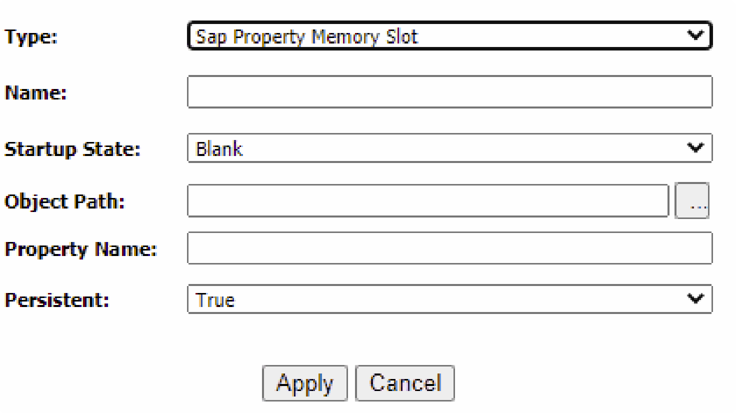
beyond the max will make no changes.

- **Both** = Incrementing beyond the max will cause the value to loop to the min value, and decrementing below the min value will cause it to loop to the max value.

These properties make it easy to assert a valid range to a numeric memory slot as well as to implement things like paging functions where paging beyond the maximum number of pages will take you back to page 1.

SapProperty

SapProperty memory slots allow you to map the value of any Sap Property in the system to this memory slot's value. When this Sap Property value changes, the slot is updated with that value. Its configuration parameters are shown below:



Type: Sap Property Memory Slot

Name:

Startup State: Blank

Object Path:

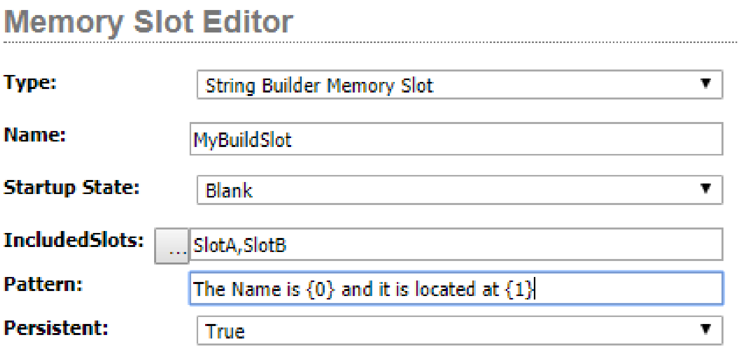
Property Name:

Persistent: True

In addition to the properties outlined in the common properties section above, this memory slot type includes two additional properties: **Object Path** and **Property Name**. The ellipsis button next to Object Path can be used to select the Sap Property to be used with this slot and when selected will present a property selection dialog similar to that of the dialog in Logic Flow endpoint editing. Once a property is selected, the object path and property name will be filled in based on the dialog selection. The value property is not present in the configuration as it gets filled dynamically over time according to the value of the selected Sap Property. This value is then available to be used with Logic Flows. It can also be used to create custom values when used in conjunction with the String Builder memory slot.

String Builder

The string builder slot allows you to build a string based on values in other Memory Slots. The parameters that will be presented when selecting this type of memory slot are shown below:



Memory Slot Editor

Type: String Builder Memory Slot

Name: MyBuildSlot

Startup State: Blank

IncludedSlots: SlotA,SlotB

Pattern: The Name is {0} and it is located at {1}

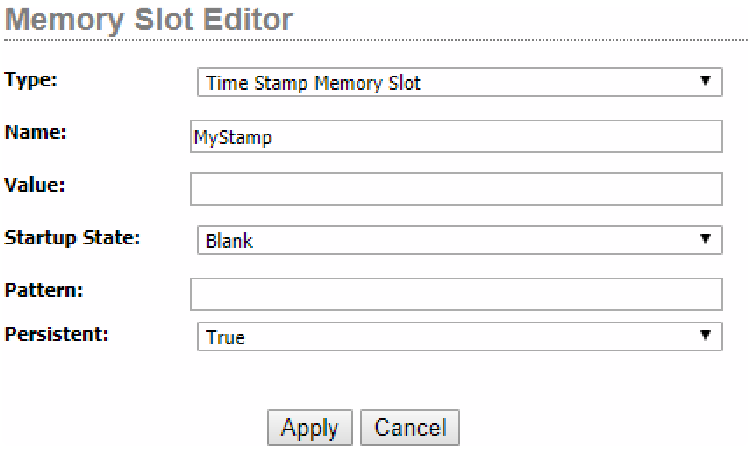
Persistent: True

In addition to the properties outlined in the common properties section above, this memory slot type includes two additional properties; **IncludedSlots** and **Pattern**. The **IncludedSlots** field is used to enter a comma-delineated list of memory slot names to use in the builder. The ellipsis button will present a list of slots to select from. The order is important as described by the **Pattern**. The **Pattern** field is a text pattern with bracketed numbers used where slot values from the comma-delineated list should be inserted.

In the example above, case {0} will be replaced with the value of slot A and {1} will be replaced with the value of slot B. If Slot A's value is "Fred" and Slot B's value is "Telos Alliance", the value of this new memory slot called MyBuiltSlot would be: The Name is Fred and it is located at Telos Alliance. The value of this slot will update whenever Slot A or Slot B updates. This slot can be used to build text which is then used as labels on buttons or commands to generic emulators. When the included slot list contains slots that are Sap Property Memory Slots, this memory slot type can become even more powerful.

Time Stamp

The Time Stamp Memory Slot can be used to grab time stamps, which can be useful in displaying the last time something happened in a user panel. The parameters that will be presented when selecting this type of memory slot are shown below:



Memory Slot Editor

Type: Time Stamp Memory Slot

Name: MyStamp

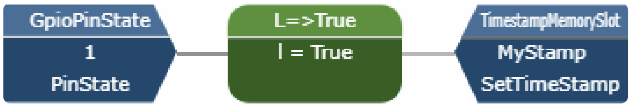
Value:

Startup State: Blank

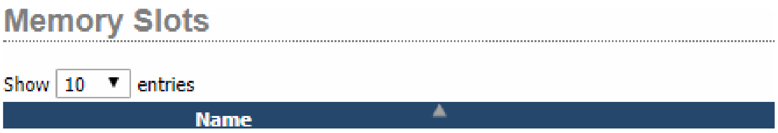
Pattern:

Persistent: True

In addition to the properties outlined in the common properties section above, this memory slot type includes one additional property: **Pattern.** When you first create a time stamp memory slot, you will probably not apply a value. The pattern field can also be left blank. It can be used to specify the format of the time stamp which we will describe in more detail below. After creating the timestamp memory slot, it can be used in Logic Flows by applying an endpoint to the SetTimeStamp write-only property of the memory slot. For example:



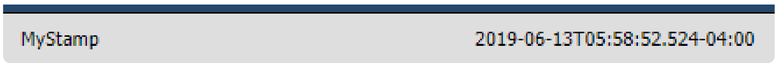
This Logic Flow will set the memory slot to the current date and time every time the specified GPIO pin goes low.



Memory Slots

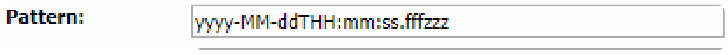
Show 10 entries

Name



MyStamp 2019-06-13T05:58:52.524-04:00

The format of the date time stamp can be changed using the pattern field. If you edit the memory slot again you will see that the default format has been assigned to the memory slot:



Pattern: yyyy-MM-ddTHH:mm:ss.fffzzz

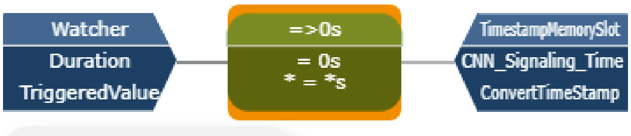
The time stamp memory slot can also be used to convert values from one time format to the pattern specified in the time stamp memory slot. This is accomplished using the write only CalculateTime property. For example, you could assign a number with either m, h, or s to the end, and it would convert the value to the requested format in minutes hours or seconds. If the time stamp memory slot pattern was:

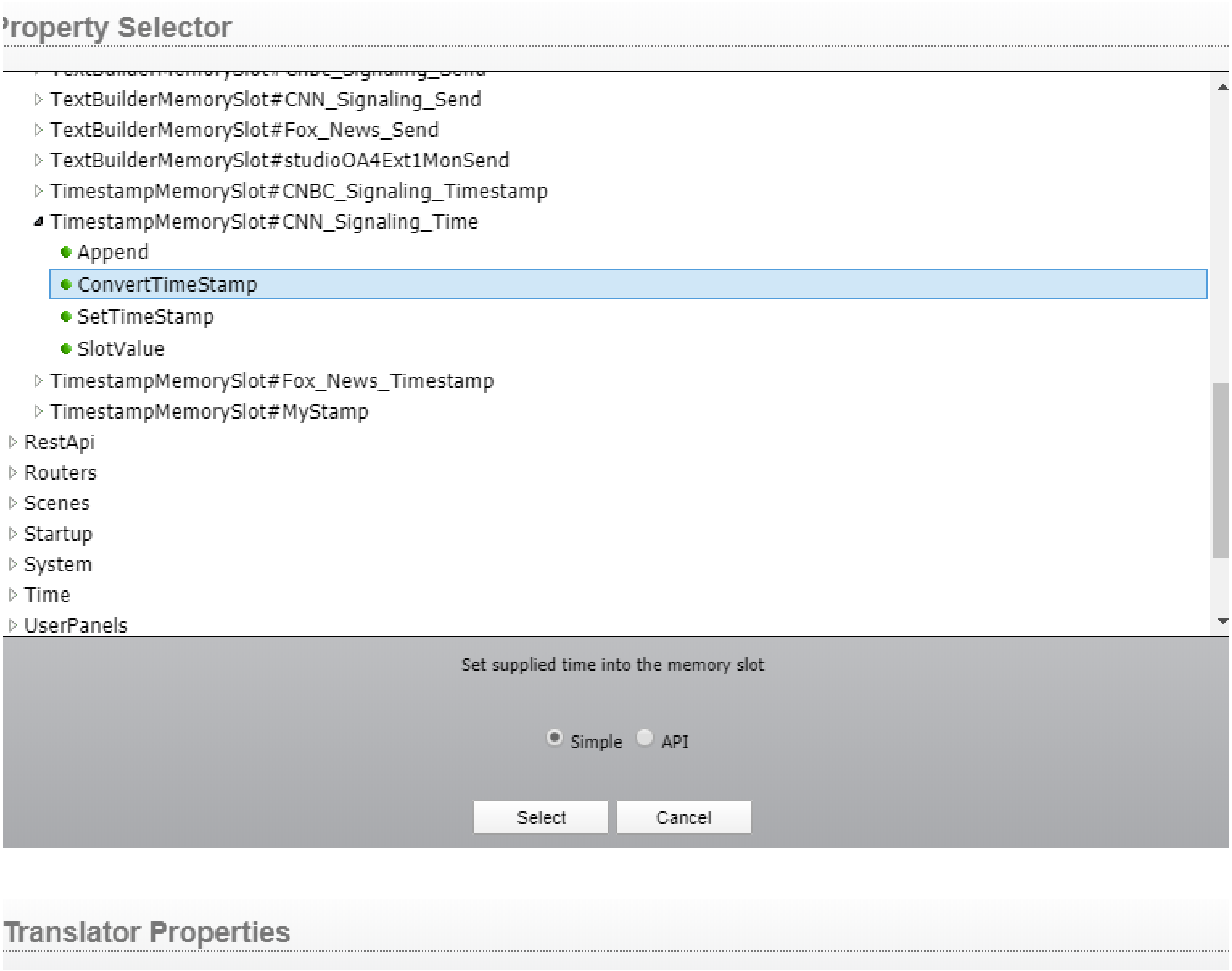
hh:mm:ss

and you passed in

100s

to the CalculateTime property, the value of the memory slot would become 00:01:40.





Property Selector

- TextBuilderMemorySlot#CNBC_Signaling_Send
- ▷ TextBuilderMemorySlot#CNN_Signaling_Send
- ▷ TextBuilderMemorySlot#Fox_News_Send
- ▷ TextBuilderMemorySlot#studioOA4Ext1MonSend
- ▷ TimestampMemorySlot#CNBC_Signaling_Timestamp
- ◢ TimestampMemorySlot#CNN_Signaling_Time
  - ● Append
  - ● ConvertTimeStamp
  - ● SetTimeStamp
  - ● SlotValue
- ▷ TimestampMemorySlot#Fox_News_Timestamp
- ▷ TimestampMemorySlot#MyStamp

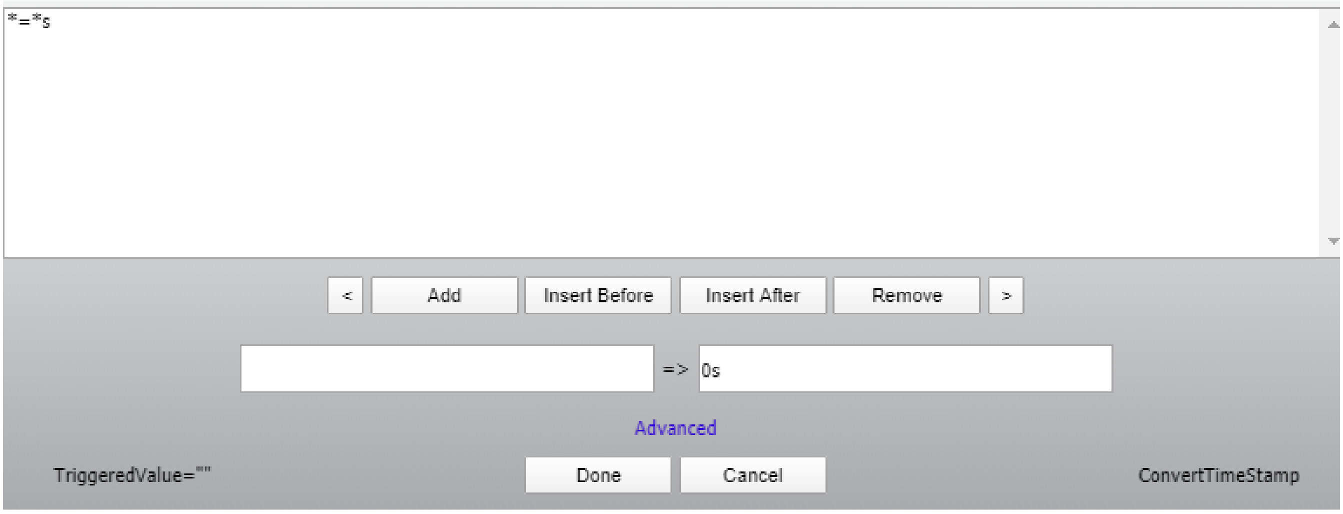
▷ RestApi
▷ Routers
▷ Scenes
▷ Startup
▷ System
▷ Time
▷ UserPanels

Set supplied time into the memory slot

● Simple ○ API

Select Cancel

Translator Properties



```
*=*s
```

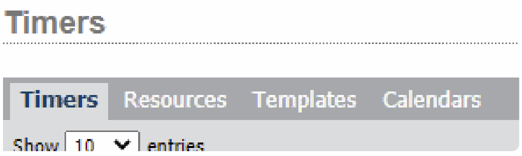
Passing other time values than just numbers will try to interpret the value as time, convert it, and then express it in the correct pattern.

Startup State Drop-down

The startup state drop-down applies to each of the memory slot types above and requires some explanation. This option allows you to determine the value of the slot when the system first starts up and before any flows are executed. It can be blank, a fixed value as defined when you create the slot, or last known. It is important to understand the last known option will attempt to store the state in the database each time the memory slot state changes. This can be more CPU- and time-intensive because of the compact flash storage within the Pathfinder Core PRO. As a result, this option should not be used in situations where the existing state of Axia devices will reset the memory slot to the correct state via the flows as the flows start-up anyway.

Timers

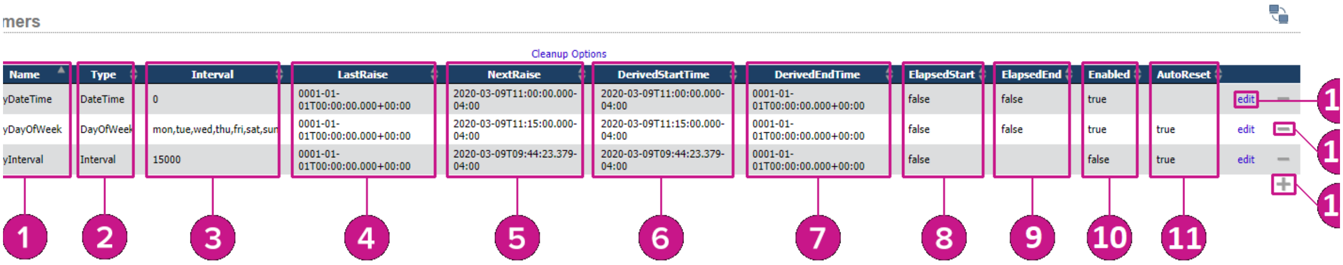
After clicking on the Timers navigation link, you will be presented with 4 tabs underneath the Timers section:



The timers tab can be used to create interval, date/time, and recurring timers which can either execute changes at specific times or after an interval. They can be used by assigning property changes in the time event and/or by hooking the timer to a logic flow. The resources, templates, and calendars tabs are for use with the additional scheduling license.

Timers Tab

Timers allow Flows and or property changes (such as a route change) to be executed at specific times. This list shows the Timers that have been created and may be used in Logic Flows. The page shows up to date information as the Timer states changes.



mers

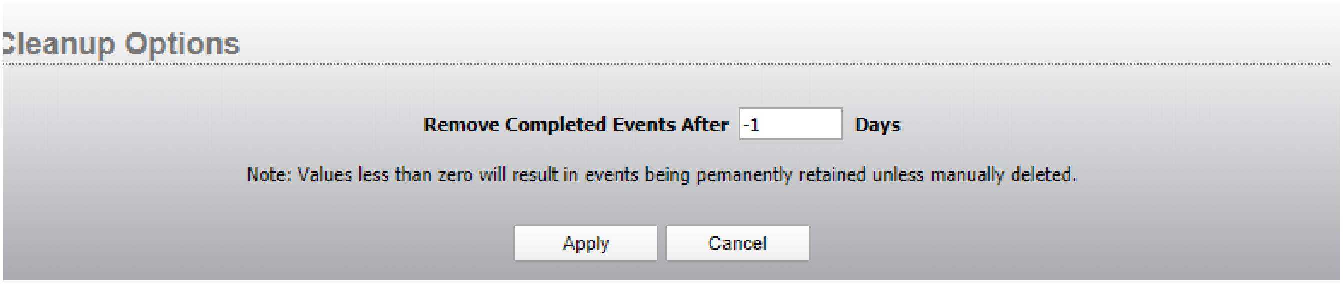
Cleanup Options

Name	Type	Interval	LastRaise	NextRaise	DerivedStartTime	DerivedEndTime	ElapsedStart	ElapsedEnd	Enabled	AutoReset		
yDateTime	DateTime	0	0001-01-01T00:00:00.000+00:00	2020-03-09T11:00:00.000-04:00	2020-03-09T11:00:00.000-04:00	0001-01-01T00:00:00.000+00:00	false	false	true		edit	−
yDayOfWeek	DayOfWeek	mon,tue,wed,thu,fri,sat,sun	0001-01-01T00:00:00.000+00:00	2020-03-09T11:15:00.000-04:00	2020-03-09T11:15:00.000-04:00	0001-01-01T00:00:00.000+00:00	false	false	true	true	edit	−
yInterval	Interval	15000	0001-01-01T00:00:00.000+00:00	2020-03-09T09:44:23.379-04:00	2020-03-09T09:44:23.379-04:00	0001-01-01T00:00:00.000+00:00	false		false	true	edit	−

Reference	Column/Button	Notes
1	Name	The **name** of the Timer.
2	Type	The **type** of Timer.
3	Interval	In the case of *Interval* Timers, this column will display the **tim** in milliseconds. In the case of *Day of week* Timers, the colum will display the **days per week** that are selected. This column not used for *Date/Time* Timers.
4	LastRaise	This column displays the date and time when the Timer last elapsed. If the Timer has never elapsed, it will display a defaul minimum date time value.
5	NextRaise	This column displays the date and time when the Timer is ne› scheduled to elapse. If the Tim has never been enabled, this column may also show a defau minimum date time value. If a timer's start time has passed bı the end time has not, this will show the end time.
6	DerivedStartTime	This column displays the time when the event will start. This will shift for day of week (recurring) events as days progress.
		This column displays the time when the event will end. This

7	DerivedEndTime	will shift for day of week (recurring) events as days progress.
8	ElapsedStart	The ElapsedStart column will turn **true** if the Start time of the Timer has elapsed. It is important to note that if a Timer is set to *Autoreset*, this value w immediately turn **false** again.
9	ElapsedEnd	The ElapsedEnd column will turn **true** if the End time of the timer has elapsed.
10	Enabled	This column displays whether the Timer is currently enabled. Timer which is not enabled will never elapse.
11	Autoreset	A value of true indicates an Interval Timers set to auto rese When the Timer elapses, it will automatically update its next raise time and start counting down again.
12	Edit	Clicking the **Edit** link allows yo to edit and change the parameters of a Timer.
13	Delete	Clicking the **minus** icon allows you to delete a Timer from the system.
14	Add	Clicking the **add** icon allows th user to create a new Timer.

Additionally the cleanup timers link can be used to specify what should happen with non recurring Date/Time timers that have already elapsed. Clicking the cleanup option link will present the following dialog:



Cleanup Options

Remove Completed Events After -1 Days

Note: Values less than zero will result in events being pemanently retained unless manually deleted.

By default, the value will be -1 which means DateTime timers which have completed their execution will not be deleted from the system. Setting a number of days into this dialog and applying that number will cause single execution datetime events to be deleted from the system after the requested number of days has passed after the timer has completed executing. A value of 0 will cause the timer to be deleted sometime within the hour or two after the timer has completed execution.

Timer Types

To create a new Timer, click the **Add** icon on the Timers dialog to open the Timer Editor.

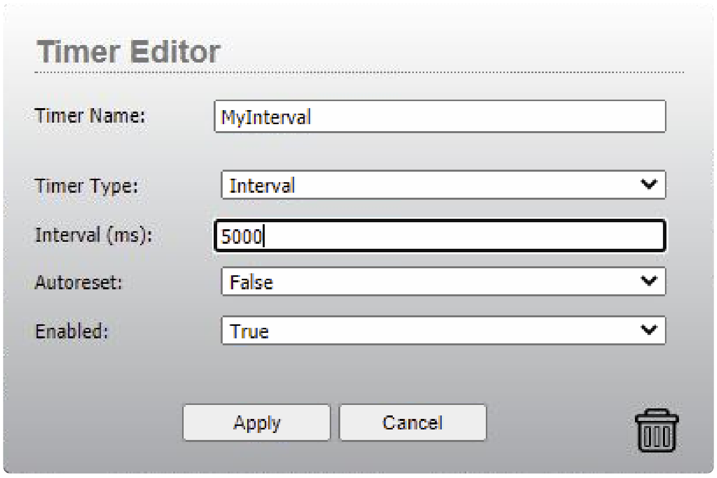
Three types of Timers can be created in Pathfinder Core PRO: **Interval**, **Day of Week**, and **Date/Time**.

Interval Timers

If you want to create a Logic Flow with a delayed action, you could use a delay directly in the Flow, or you could create an Interval Timer. An Interval Timer raises an *Elapsed* property after a specified number of milliseconds.

Interval Timers can also be programmed to autoreset. An autoresetting Interval Timer will set the *Elapsed* flag to **true** after the Interval, and then immediately set it back to **false** and restart the countdown.

Leave the *Autoreset* and *Enabled* properties as **false**. Type a name for this Timer and apply it.



Timer Editor

Timer Name: MyInterval

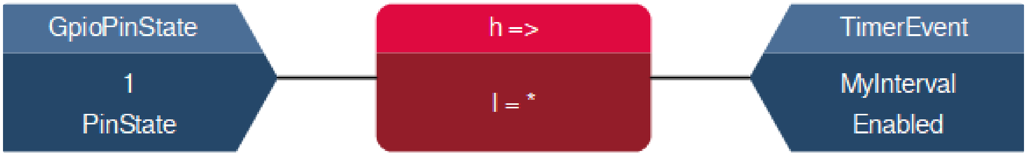
Timer Type: Interval

Interval (ms): 5000

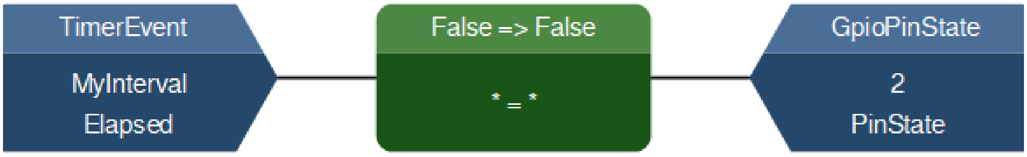
Autoreset: False

Enabled: True

Now we can use this Timer in Logic Flows:



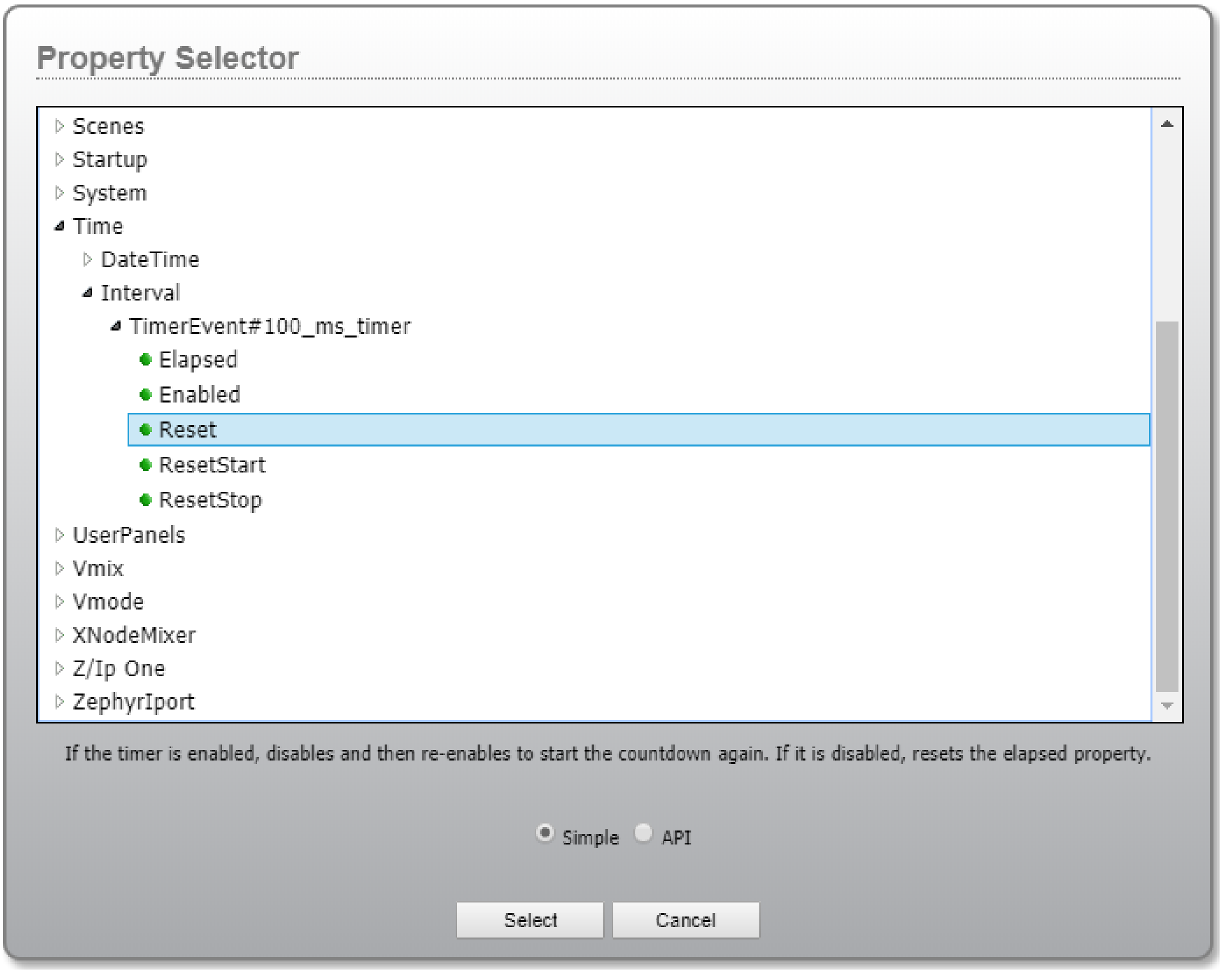
For example, an Interval Timer can be used in a Logic Flow so when GPI 1 goes low, it enables the Timer. Then the Elapsed property 5-seconds later causes GPI 2 pin to be set low:



Interval Timers have three additional write-only properties:

1. **Reset**: If the Timer is enabled, this property will disable and then re-enable the Timer to start the countdown over. If the Timer is disabled, this will set the elapsed property to false.
2. **ResetStart**: Stops the Timer (if it is running) and then starts the Timer to reset the countdown.
3. **ResetStop**: Disables the Timer and changes the elapsed state to false.

These are available in the Logic Flows simple tree when an Interval Timer endpoint is selected in the Logic Flow. Since these are write-only properties they will not be shown when Interval Timer start points are selected – only endpoints.

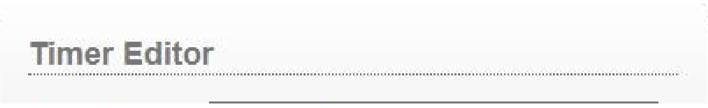


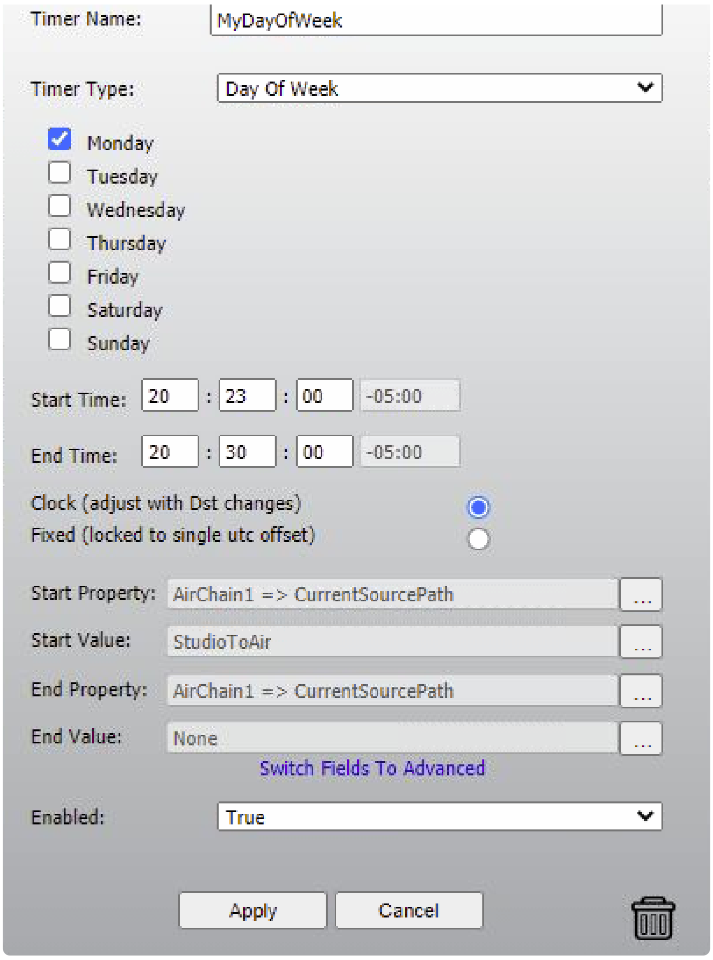
If the timer is enabled, disables and then re-enables to start the countdown again. If it is disabled, resets the elapsed property.

All three properties accept true/false values and setting any of them to true will initiate the reset. They do not need to be set to false again afterwards as passing the true value is just a trigger to initiate the reset. The value is not retained. These properties can be thought of more as actions than traditional properties.

Day of Week Timers

If we want something to happen at a specific time every Monday, we could create a **Day of Week** Timer. With this type of Timer, you can specify a time and the days of the week on which the Timer should set its *Elapsed* property to **true**.





Timer Name: MyDayOfWeek

Timer Type: Day Of Week

☑ Monday
□ Tuesday
□ Wednesday
□ Thursday
□ Friday
□ Saturday
□ Sunday

Start Time: 20 : 23 : 00 -05:00

End Time: 20 : 30 : 00 -05:00

● Clock (adjust with Dst changes)
○ Fixed (locked to single utc offset)

Start Property: AirChain1 => CurrentSourcePath ...

Start Value: StudioToAir ...

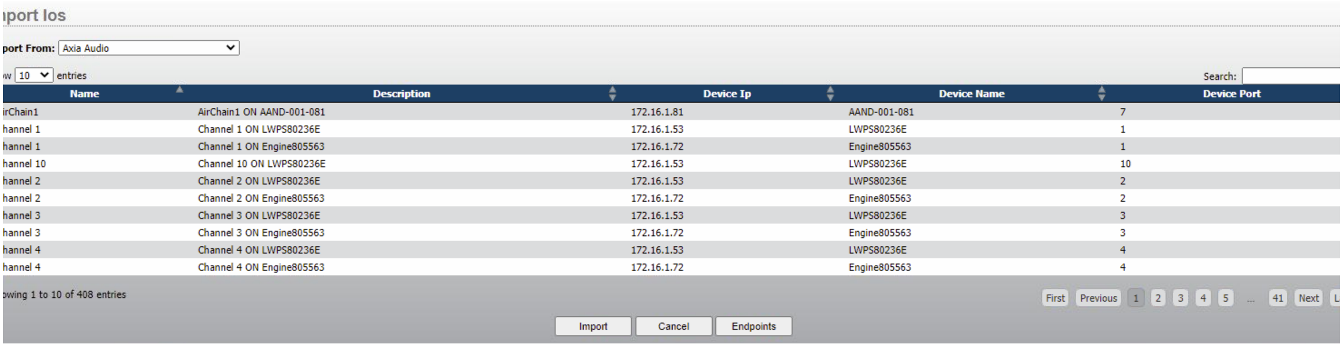
End Property: AirChain1 => CurrentSourcePath ...

End Value: None ...

Enabled: True

Each Day of Week timer actually represents a range of time. It has a start time and an end time. And in logic flows there is an Elapsed and an ElapsedEnd property to match the start and end times. If you only want to execute a single action at a specific time, you could just use the Elapsed property and ignore the ElapsedEnd property. The end time must be specified as later than the start time when configuring the event however.

Instead of or in addition to using this with logic flows, you can also specify a change directly in the event itself. Clicking the ellipsis button next to the start property will open a route selection dialog where you can select a destination.



ıport Ios

port From: Axia Audio

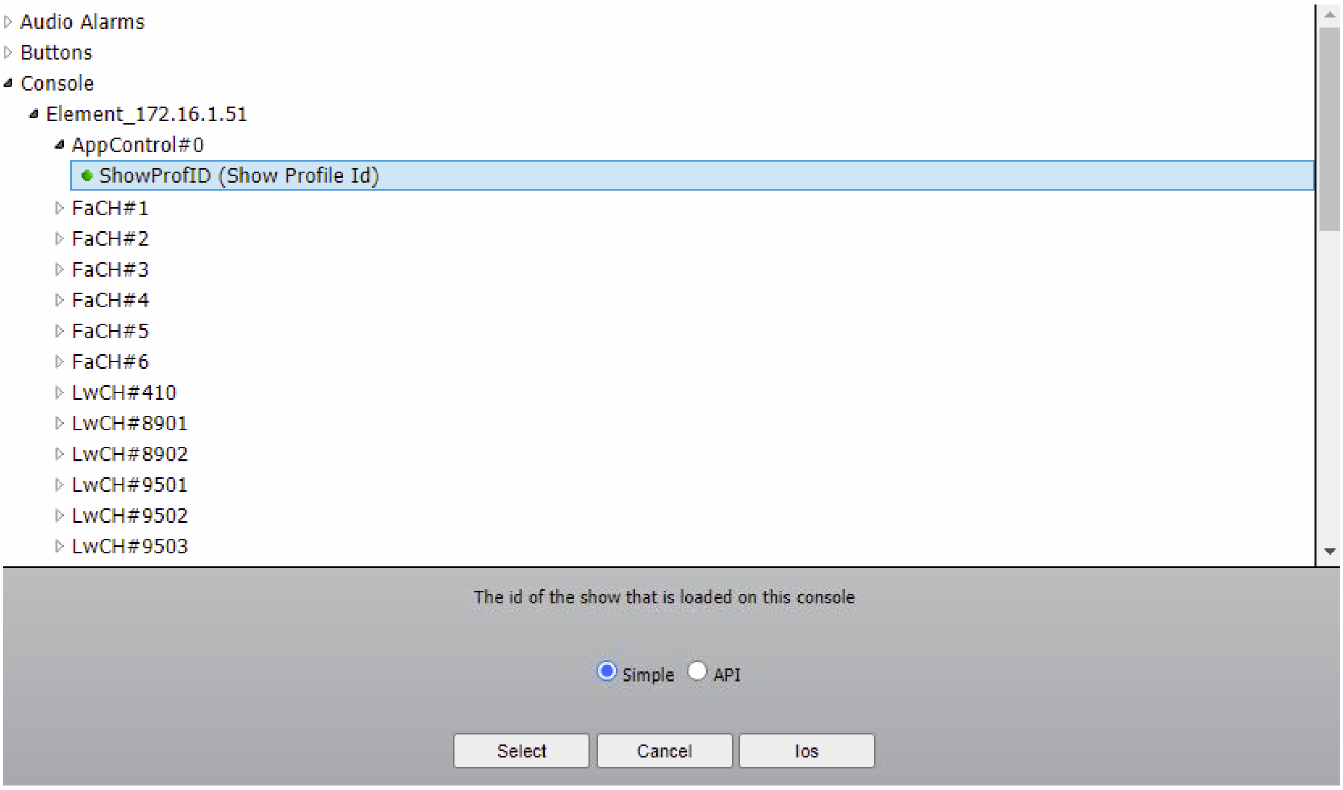
w 10 entries Search:

Name	Description	Device Ip	Device Name	Device Port
irChain1	AirChain1 ON AAND-001-081	172.16.1.81	AAND-001-081	7
hannel 1	Channel 1 ON LWPS80236E	172.16.1.53	LWPS80236E	1
hannel 1	Channel 1 ON Engine805563	172.16.1.72	Engine805563	1
hannel 10	Channel 10 ON LWPS80236E	172.16.1.53	LWPS80236E	10
hannel 2	Channel 2 ON LWPS80236E	172.16.1.53	LWPS80236E	2
hannel 2	Channel 2 ON Engine805563	172.16.1.72	Engine805563	2
hannel 3	Channel 3 ON LWPS80236E	172.16.1.53	LWPS80236E	3
hannel 3	Channel 3 ON Engine805563	172.16.1.72	Engine805563	3
hannel 4	Channel 4 ON LWPS80236E	172.16.1.53	LWPS80236E	4
hannel 4	Channel 4 ON Engine805563	172.16.1.72	Engine805563	4

owing 1 to 10 of 408 entries

The import from button allows you to select the router you want to pick a destination from. Additionally the endpoints button in this dialog will switch to the property selector similar to the one in Logic flows so that you can select some other property like a gpio or console show profile load property:

ɾroperty Selector

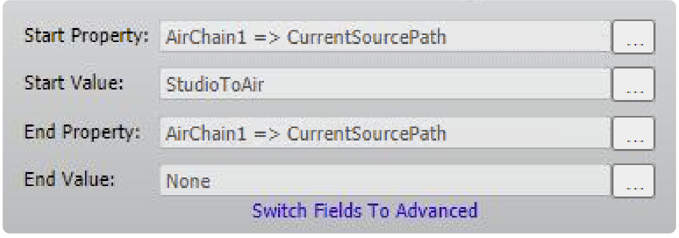


- Audio Alarms
- Buttons
- Console
  - Element_172.16.1.51
    - AppControl#0
      - ● ShowProfID (Show Profile Id)
    - FaCH#1
    - FaCH#2
    - FaCH#3
    - FaCH#4
    - FaCH#5
    - FaCH#6
    - LwCH#410
    - LwCH#8901
    - LwCH#8902
    - LwCH#9501
    - LwCH#9502
    - LwCH#9503

The id of the show that is loaded on this console

● Simple ○ API

The Ios button will take you back to audio io selection. Since many times audio routing is what will be used with date/time events we present both dialogs for ease of selection. Once the property is selected, the value field will be filled with the current value of the destination's source or property value. Selecting the ellipsis button next to the value will allow you to change the property value that will be applied. If you only want a change to happen at the start, then do not select a property and value for the end. In the example below, a studio source will be routed to air at the start time of the event and then the airchain will be cleared at the end of the event.



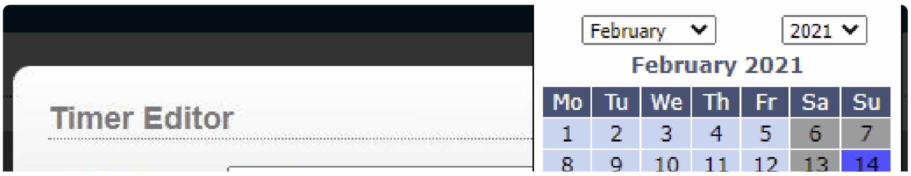
Start Property:	AirChain1 => CurrentSourcePath
Start Value:	StudioToAir
End Property:	AirChain1 => CurrentSourcePath
End Value:	None

The Switch Fields To Advanced button should only be used by advanced users. It changes the input boxes to the literal API commands associated with the selections for manual adjustment.

Note that the clock option buttons will be described below as they apply to both the Day Of Week and Date/Time Timers.

Date/Time Timers

If we want something to happen on a specific day and time, we could create a **Date/Time** Timer. The Timer’s *Elapsed* property will be set to **true** at that day and time.

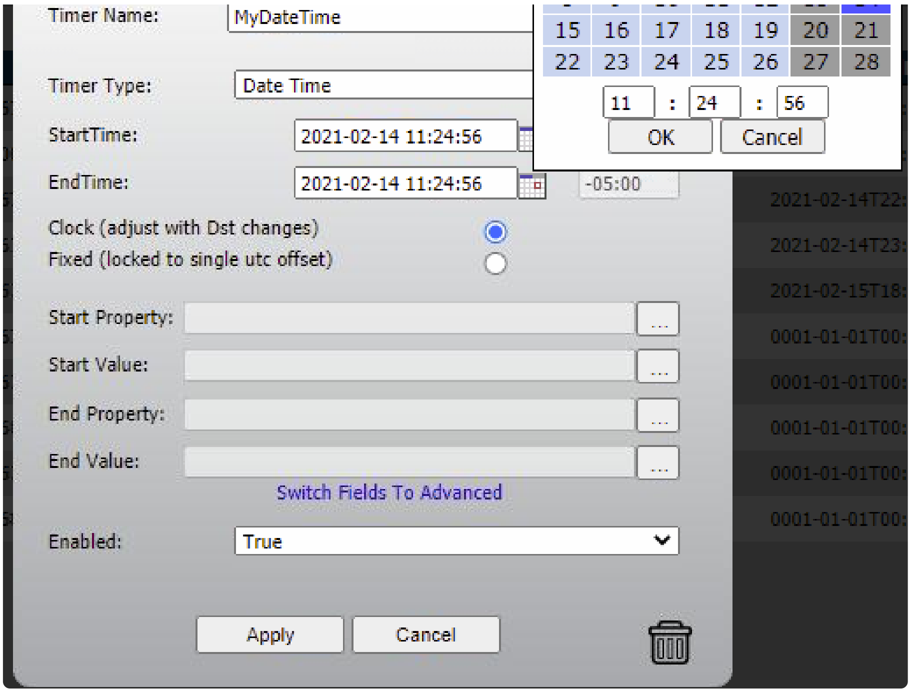


Timer Editor

February 2021

February 2021

Mo	Tu	We	Th	Fr	Sa	Su
1	2	3	4	5	6	7
8	9	10	11	12	13	14



Timer Name: MyDateTime

Timer Type: Date Time

StartTime: 2021-02-14 11:24:56

EndTime: 2021-02-14 11:24:56 -05:00

Clock (adjust with Dst changes) ●

Fixed (locked to single utc offset) ○

Start Property:

Start Value:

End Property:

End Value:

Enabled: True

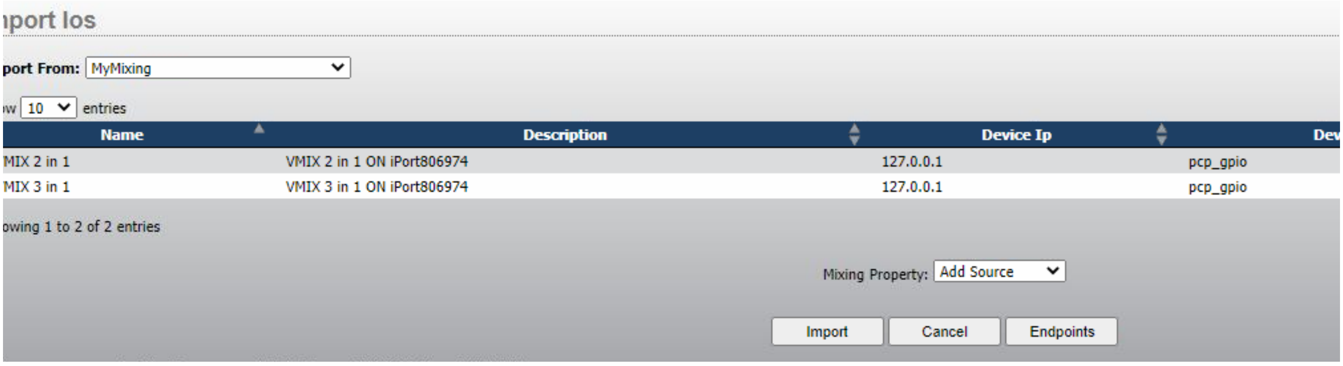
15	16	17	18	19	20	21
22	23	24	25	26	27	28

11 : 24 : 56

Clicking the calendar icon will present a user-friendly calendar for configuring the date and time. This event has the same parameters as the *Day of Week* event above for selecting properties to change at the start and end time of the event. The primary difference is that this event executes it's start and end times only once at the specified date and time. It will then be deleted or retained according to the settings in the cleanup options above.

Virtual Mixing Routers and Property Selection

Virtual mixing routers are inherently different in how they function from other routers. For details on Virtual Mixing Routers, see their section in the Routers section of the manual. With Virtual mixing destinations you do not change the source that is routed to the destination. Instead you add sources that will be mixed together by the mixing destination up to the supported quantity of the destination. As a result it uses different routing properties than a normal router. For example, AddSource adds a source to the mixing destination, and RemoveSource removes a source from the destination freeing that slot up for another source. Therefore when importing a destination as a property change in a timer event, you need to be able to define whether the change will add or remove a source. To support this, a drop down will appear in the import destination list when defining a start or end property in a timer.



ıport Ios

port From: MyMixing

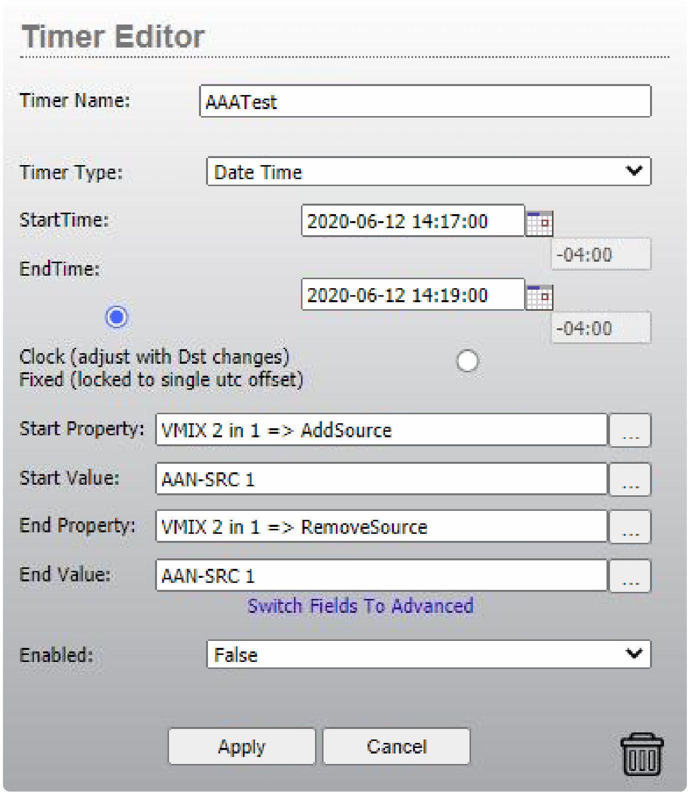
w 10 entries

Name	Description	Device Ip	Dev
MIX 2 in 1	VMIX 2 in 1 ON iPort806974	127.0.0.1	pcp_gpio
MIX 3 in 1	VMIX 3 in 1 ON iPort806974	127.0.0.1	pcp_gpio

owing 1 to 2 of 2 entries

Mixing Property: Add Source

The Mixing property drop down allows you to select whether the operation will add or remove a selected source to/from the destination. For example:



Timer Editor

Timer Name: AAATest

Timer Type: Date Time

StartTime: 2020-06-12 14:17:00 -04:00

EndTime: 2020-06-12 14:19:00 -04:00

● Clock (adjust with Dst changes)
○ Fixed (locked to single utc offset)

Start Property: VMIX 2 in 1 => AddSource

Start Value: AAN-SRC 1

End Property: VMIX 2 in 1 => RemoveSource

End Value: AAN-SRC 1

Enabled: False

In this example the timer will add Source 1 to the destination at the start of the event time range, and it will remove it at the end.

Clock Time and Fixed Time

In both the *Day of Week* and *Date/Time* Timers, it is important to understand the difference between clock time and fixed time. Within Pathfinder Core PRO, all times are stored with a UTC offset. **Clock time** means that the event will happen at a specific time (say 7PM) according to the clock regardless of the UTC offset. If you want an event to happen at 7PM according to what the clock shows both before and after a daylight savings shift, you should pick the clock time option.

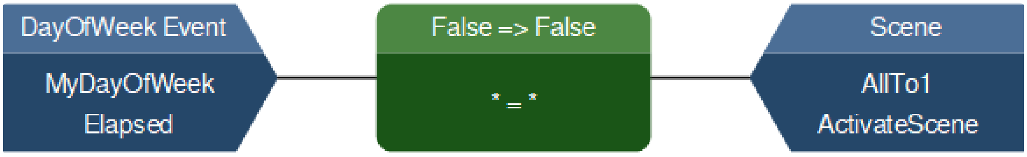
However, if you are picking up a satellite feed and the region from which the feed originates does not shift their clock to daylight savings time, then you should select **fixed time** because the event will happen at a different time (according to your wall clock) before and after the daylight savings time shift. In this case, it is fixed to the time and UTC offset.

UTC Offset

Previous versions of the PathfinderCore PRO would use the browser's UTC offset when creating a new clock based *Date Time* or *Day of Week* timer. This also caused odd display issues when editing events from a Core PRO in one timezone while using a browser in another. This version will always display the offset of the event, and the specific time as it is set in Core PRO. Additionally, new events will be created using the offset defined in Core PRO by its current time zone rather than the browser's offset.

Using Timers in Flows

In addition to specifying an action to take place directly in the event, you can also hook the event to a logic flow. And in fact this is the primary way to make use of an interval type timer. To do that, you would need to create a Logic Flow with a start point that uses the Timer's elapsed property to then change some other property in the system. For example, if you wanted to execute a scene change every Monday at a specific time, you would create a Logic Flow that uses the Day of Week Timer shown above.



In this way, we can use Timers to cause delays, do repetitive operations (using the autoreset property of Interval Timers), or execute actions at specific days/times and/or on a recurring daily schedule. This particular example could also be accomplished by selecting the Activate Scene property directly in the event, but if you wanted to create more complex logic using logic gates, then you would need to build the logic in a logic flow. Both options are available. However, property changes that are defined within the actual event (as opposed to a logic flow) so not count against the logic licensing of the system.

Scheduling

PathfinderCore PRO Scheduling Module

> ⓘ **Note -** This feature requires an additional license as described below.

Overview

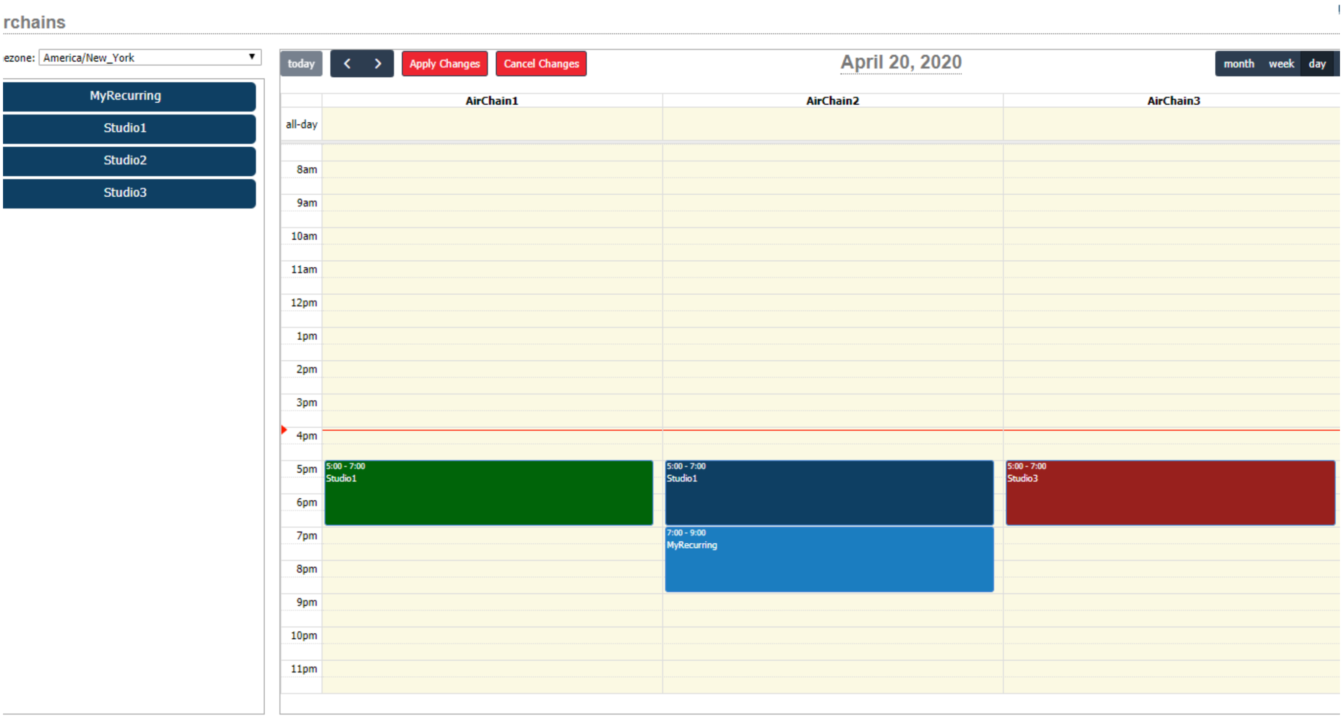
The Scheduling module is an additional module that may be enabled in PathfinderCore PRO through a license add-on. It works as an addendum to the *Day of Week* and *Date/Time* timers. Specifically, this module allows you to define resources and event templates and then apply them on an easy-to-use drag and drop calendar in order to easily schedule changes.

Scheduling License

This feature requires a scheduling license. For evaluation, we can also supply a 90-day demo license. Events created using this demo license will continue to function properly after the demo expires but without the ability to view them in the calendar. After demo expiration, the events will still be viewable and editable via the normal timers editor dialog. Because these timer events use a different object type, events created with the scheduling demo feature may not work if you downgrade the software to a previous version on PathfinderCore PRO software. In addition to the demo license, there is the ability to open a demo calendar that has non-functioning data so that the concepts may be evaluated; this does not require a license.

Calendar

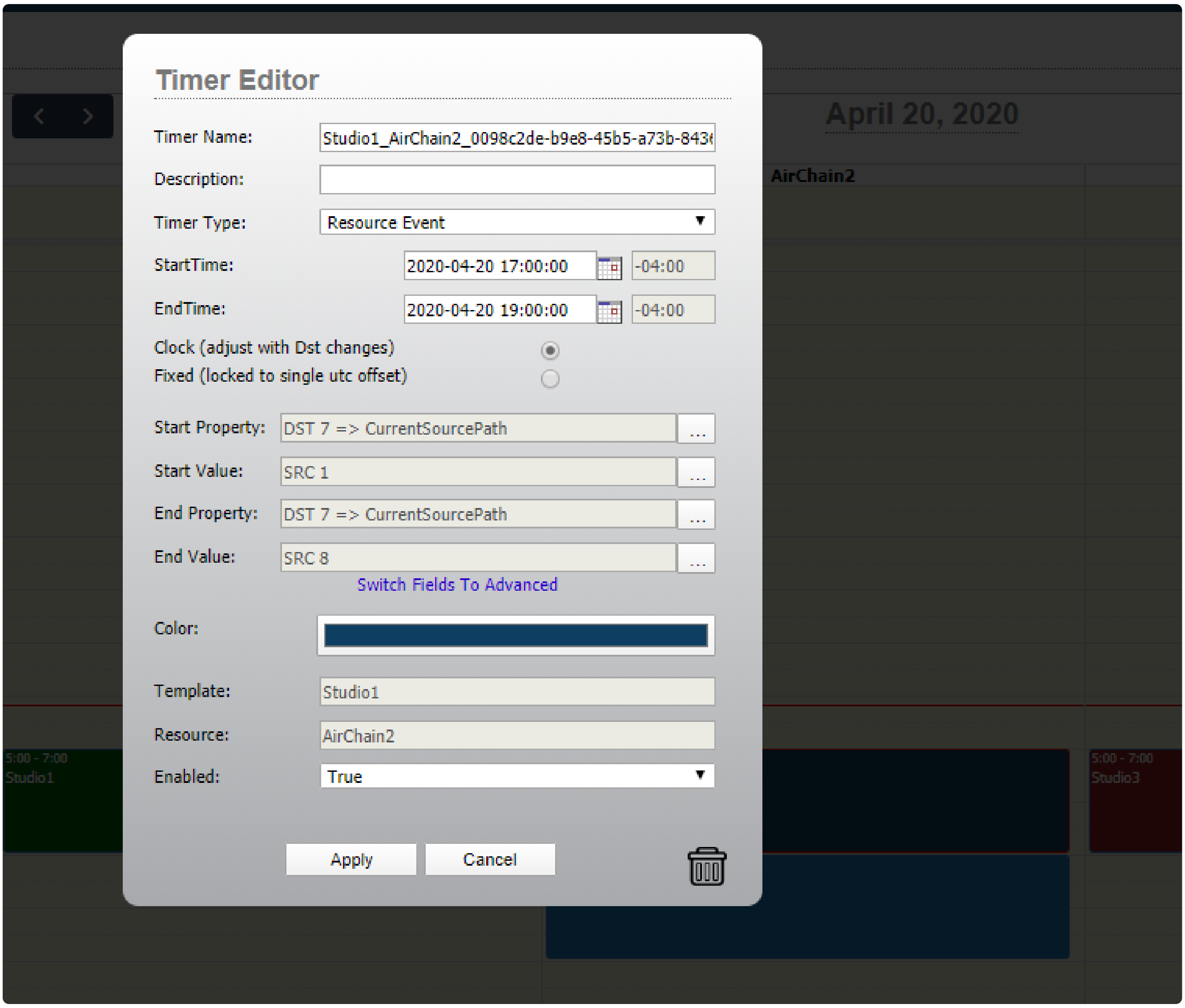
The scheduling module adds an optional calendar-based scheduling system into PathfinderCore PRO. While scheduling of time-based events has always been possible, the scheduling module simplifies this process with a drag and drop calendar user interface and the ability to dynamically combine properties and values during that process.



The scheduling module is comprised of 4 objects:

- **Resources**: A container for a set of events which also may include properties and values that interact with templates that are dropped onto the resource.
    - AirChains1, 2, and 3 are resources in the picture above.
- **Templates:** An event template with duration and start and end actions.
    - Templates are displayed in the list of rectangles on the left side of the screen in the picture above and can be drug onto the calendar.
- **Calendars:** A calendar-based visualization of resources and their associated events along with drag and drop visualizations of templates.
    - This image above is a calendar with three resources, 4 currently displayed ResourceEvents, and 4 templates.
- **ResourceEvents:** An event with a start and end time and action to perform at the start and end. The action data may be comprised of information from both the resource and template used to create the event.
    - The events are displayed at the time values in the calendar.

The user workflow involves selecting an event template and dragging it into the correct resource in the calendar, then adjusting the time duration or placement by dragging the event or event bottom edge. Details of the event may be further manipulated if the user has the rights to do so by clicking on the event as shown



Timer Editor

Timer Name: Studio1_AirChain2_0098c2de-b9e8-45b5-a73b-843

Description:

Timer Type: Resource Event

StartTime: 2020-04-20 17:00:00 -04:00

EndTime: 2020-04-20 19:00:00 -04:00

● Clock (adjust with Dst changes)

○ Fixed (locked to single utc offset)

Start Property: DST 7 => CurrentSourcePath

Start Value: SRC 1

End Property: DST 7 => CurrentSourcePath

End Value: SRC 8

Switch Fields To Advanced

Color:

Template: Studio1

Resource: AirChain2

Enabled: True

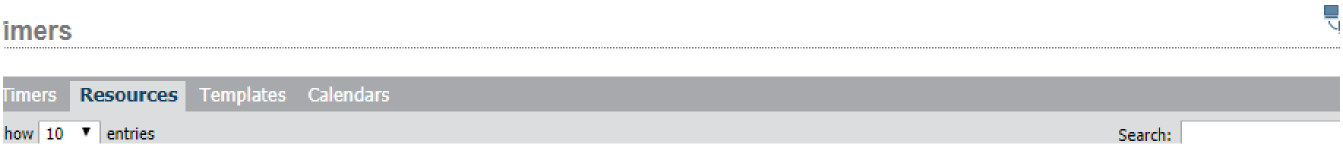
Apply Cancel

In the calendar shown earlier, each event makes a route change of the studio source defined in the template to the AirChain destination which is defined by the resource. So when we drop a studio onto the resource, the destination from the resource is combined with the source from the event template in a new event with the correct time parameters that we define by dragging the duration. This allows the scheduling of studio to airchain routing to be a simple drag and drop procedure.

It is also important to note that no changes take effect until the Apply button is selected. The apply and cancel buttons will flash when there are pending changes that have not been committed.

Creating a Resource

Let’s dig a little deeper into how resources, templates, calendars, and resource events are created and interact. As an Administrator to create a calendar such as the one we see above, we would first need to define the resources and templates. To do this, click on the Timers navigation bar link and then click on the Resources tab.

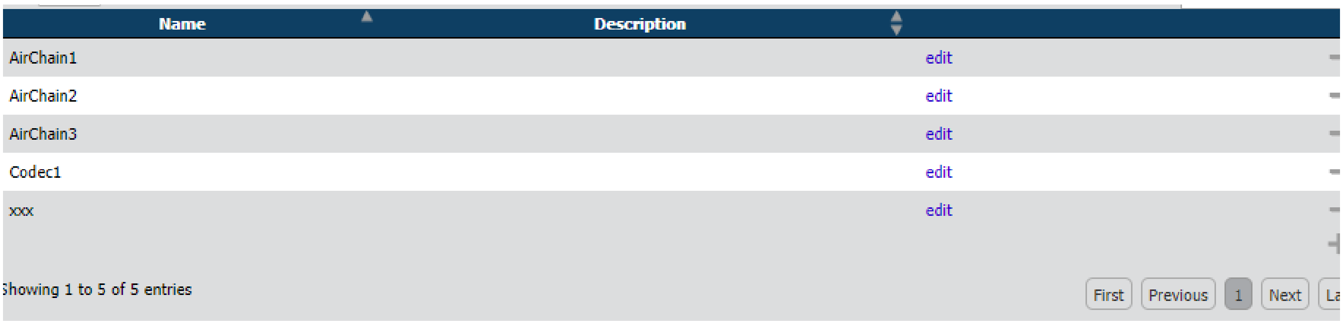


imers

Timers Resources Templates Calendars

how 10 entries

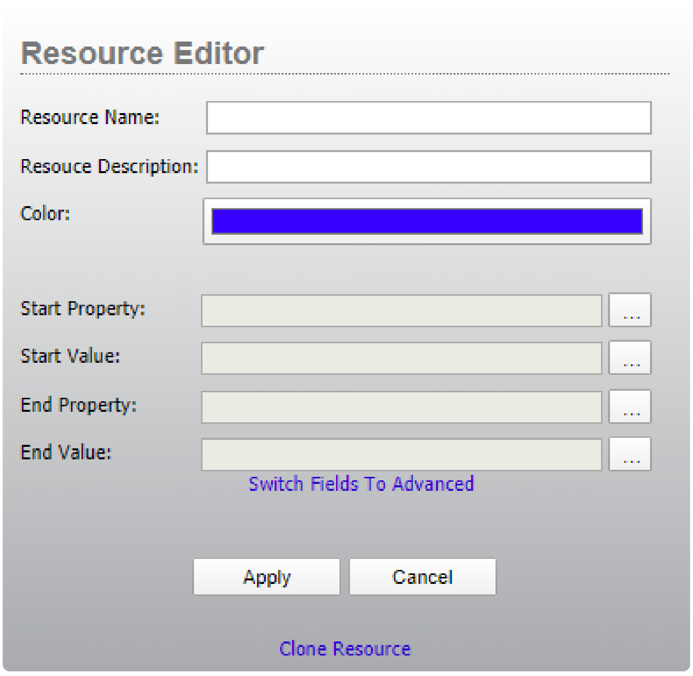
Search:



Name	Description	
AirChain1		edit
AirChain2		edit
AirChain3		edit
Codec1		edit
xxx		edit

Showing 1 to 5 of 5 entries

Click on the plus icon to add a new resource:



Resource Editor

Resource Name:

Resouce Description:

Color:

Start Property:

Start Value:

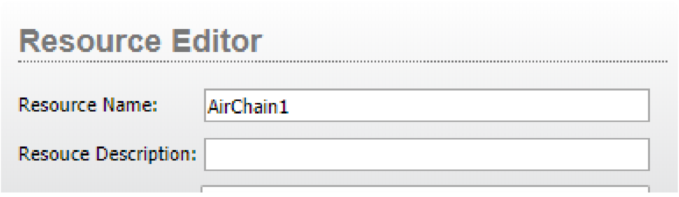
End Property:

End Value:

Each resource can have a name and description, a color, a start property and value, and an end property and value. The start property and value determine what happens at a newly created event start, and the end property and value set what happens at a newly created event end. In each case where the start and end property fields are used (resources, templates, and events), if an end action is not defined, nothing will happen at the end of the event time range.

It is important to understand that the values defined here are designed to be overridden by the template when a new template is dropped on the resource unless the template says to take data from the resource. We will see more of this when we get to the templates section below. So for example, we could leave all of the Start and End fields blank and define what happens only in the template. In the Studio and AirChain example above, we clicked on the ellipsis button and defined destinations and blank sources in these fields. The dialogs for property and value selections are the same as the *Date/Time* and *Day of Week* timers described in the Timers section of the manual.

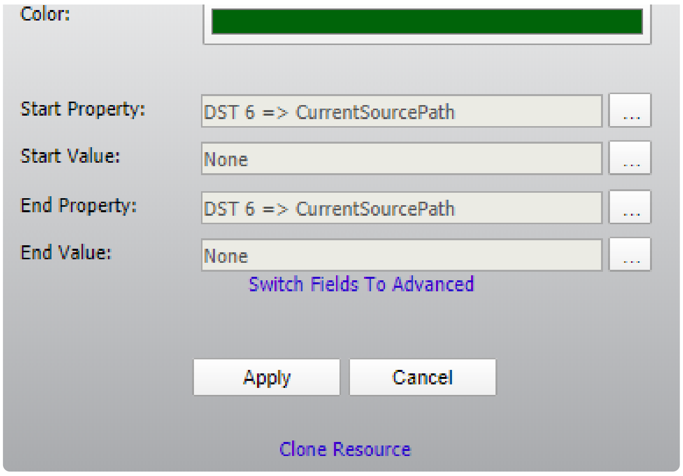
Click on the ellipsis buttons for each field and select the air chain destinations. The sources can be set to none or anything you like as we will override them in the templates section below.



Resource Editor

Resource Name: AirChain1

Resouce Description:



Color: 🟩

Start Property: DST 6 => CurrentSourcePath

Start Value: None

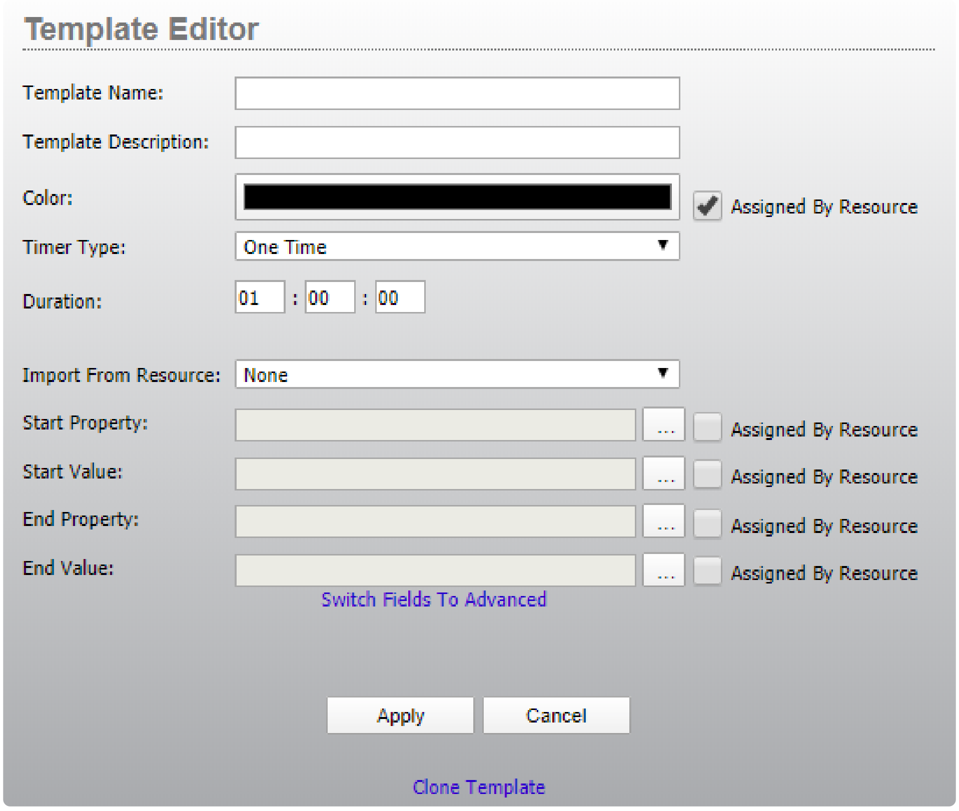
End Property: DST 6 => CurrentSourcePath

End Value: None

A color can also be picked. The clone resource link allows for quick duplication where you need to create several resources with only slight differences between them. Each resource requires a unique name.

Creating a Template

After creating several resources with different destinations for your Airchains, click on the templates tab and then use the plus icon to define a new template.



Template Editor

Template Name:

Template Description:

Color: ■ ☑ Assigned By Resource

Timer Type: One Time

Duration: 01 : 00 : 00

Import From Resource: None

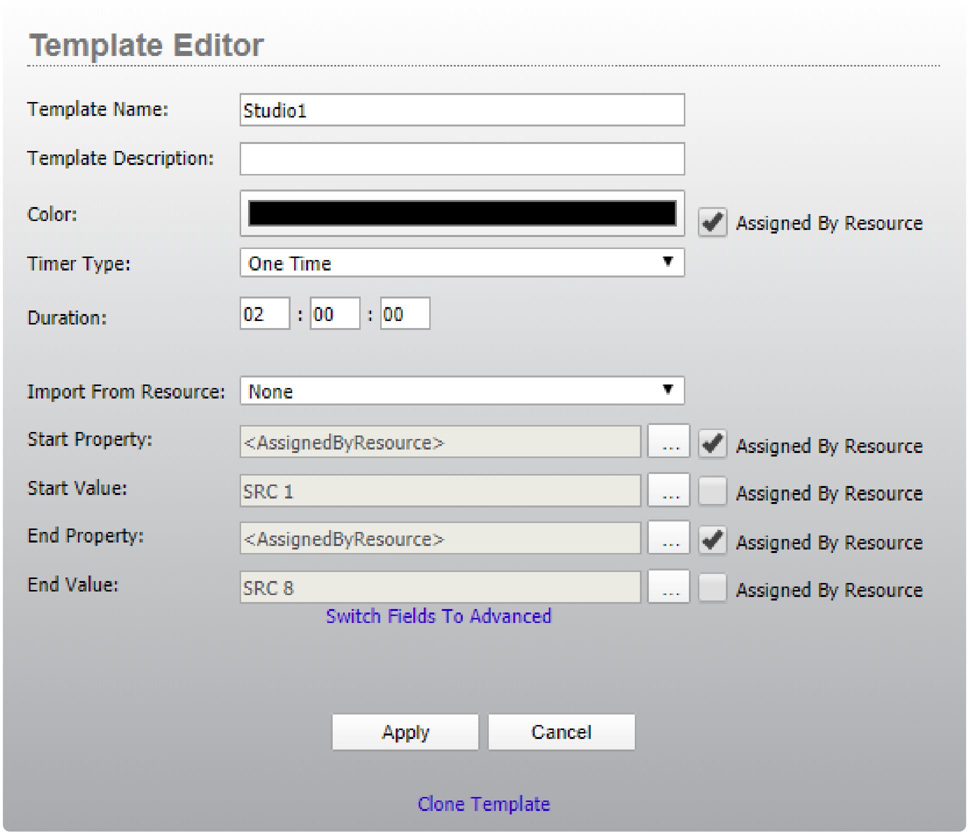
Start Property: □ Assigned By Resource

Start Value: □ Assigned By Resource

End Property: □ Assigned By Resource

End Value: □ Assigned By Resource

You will notice a checkbox next to several fields called Assigned by Resource. Enabling this checkbox indicates that the field will be obtained from the equivalent field in the resource when the template is dropped onto the resource. So for example, use the import from resource drop-down to import the settings for the properties and values from one of the Airchain resources you created earlier as it is easier than manually selecting them again. Then provide a name and enable the checkboxes on the start and end properties. The duration is the default duration that will be used when dropping the template in the calendar but then can be adjusted by dragging the event. The result will look like this:



Template Editor

Template Name: Studio1

Template Description:

Color: ☑ Assigned By Resource

Timer Type: One Time

Duration: 02 : 00 : 00

Import From Resource: None

Start Property: <AssignedByResource> ... ☑ Assigned By Resource

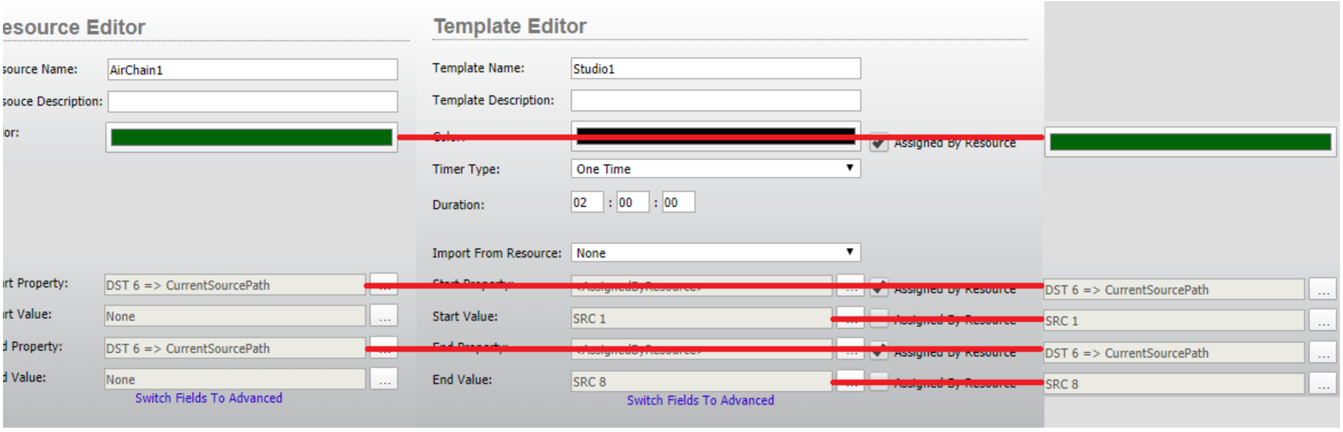
Start Value: SRC 1 ... □ Assigned By Resource

End Property: <AssignedByResource> ... ☑ Assigned By Resource

End Value: SRC 8 ... □ Assigned By Resource

Template to Resource Relationship

What we are saying is that when this template is dropped onto the resource, the template will provide the source values and the resource will provide the property (destination) and color values.



esource Editor

source Name: AirChain1

souce Description:

or:

rt Property: DST 6 => CurrentSourcePath

rt Value: None

d Property: DST 6 => CurrentSourcePath

d Value: None

Template Editor

Template Name: Studio1

Template Description:

Color: ☑ Assigned By Resource

Timer Type: One Time

Duration: 02 : 00 : 00

Import From Resource: None

Start Property: <AssignedByResource> ☑ Assigned By Resource

Start Value: SRC 1 □ Assigned By Resource

End Property: <AssignedByResource> ☑ Assigned By Resource

End Value: SRC 8 □ Assigned By Resource

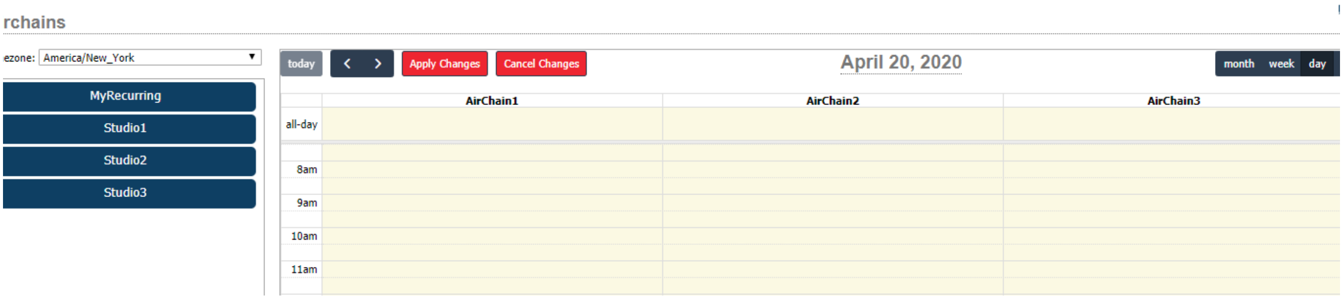
DST 6 => CurrentSourcePath

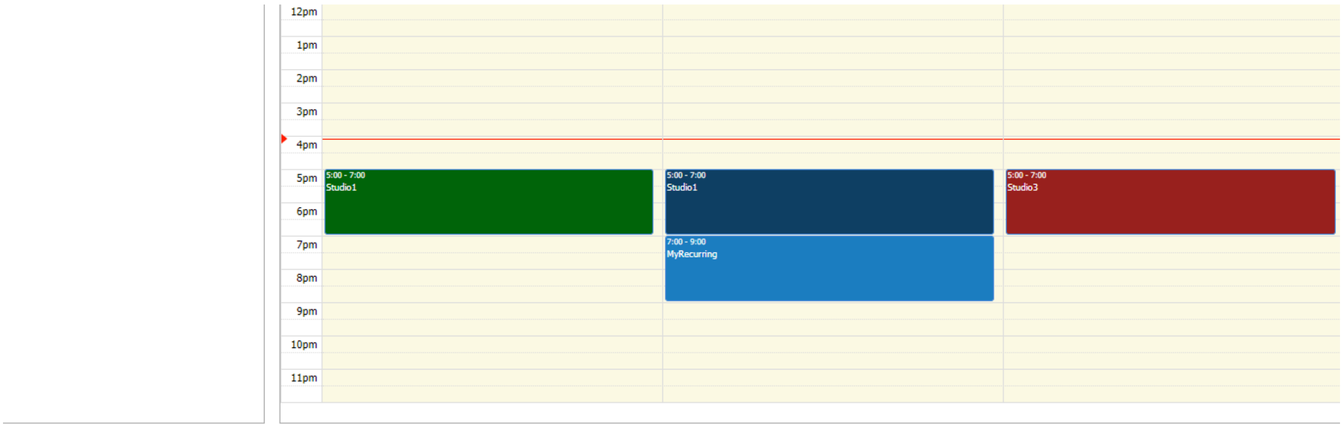
SRC 1

DST 6 => CurrentSourcePath

SRC 8

It is important to understand that if we had not turned on any of the “Assigned by Resource” checkboxes, all of the parameters from the template would be used. Why is it useful to split the parameters up in this way? If we look at our original calendar again we will see its usefulness.

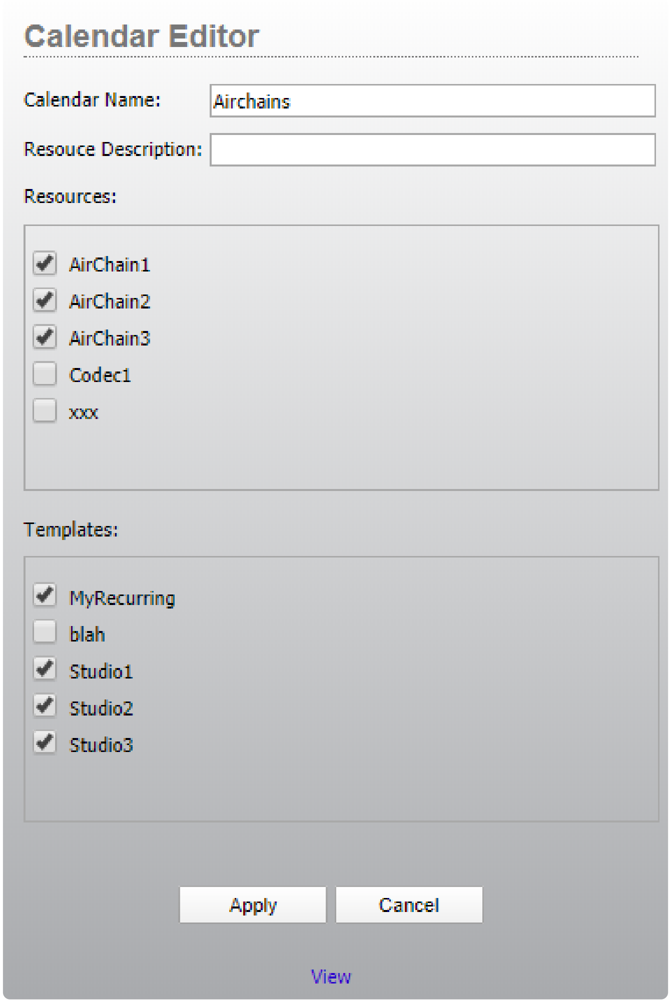




Here if you drag studio 1 onto Airchain 1, then the event will have the airchain destination and the studio source. If you decide you made a mistake, you can drag the event to Airchain 2, and the routing destination for the event will update to reflect the change.

Creating a Calendar

Going back to our Administrative configuration again, we still need to actually create the calendar. To do that, click on the calendars tab and click the Plus button. The calendars dialog allows us to create a new named calendar and configure what resources and templates are allowed to be used by the calendar.



Calendar Editor

Calendar Name: Airchains

Resouce Description:

Resources:

- ☑ AirChain1
- ☑ AirChain2
- ☑ AirChain3
- □ Codec1
- □ xxx

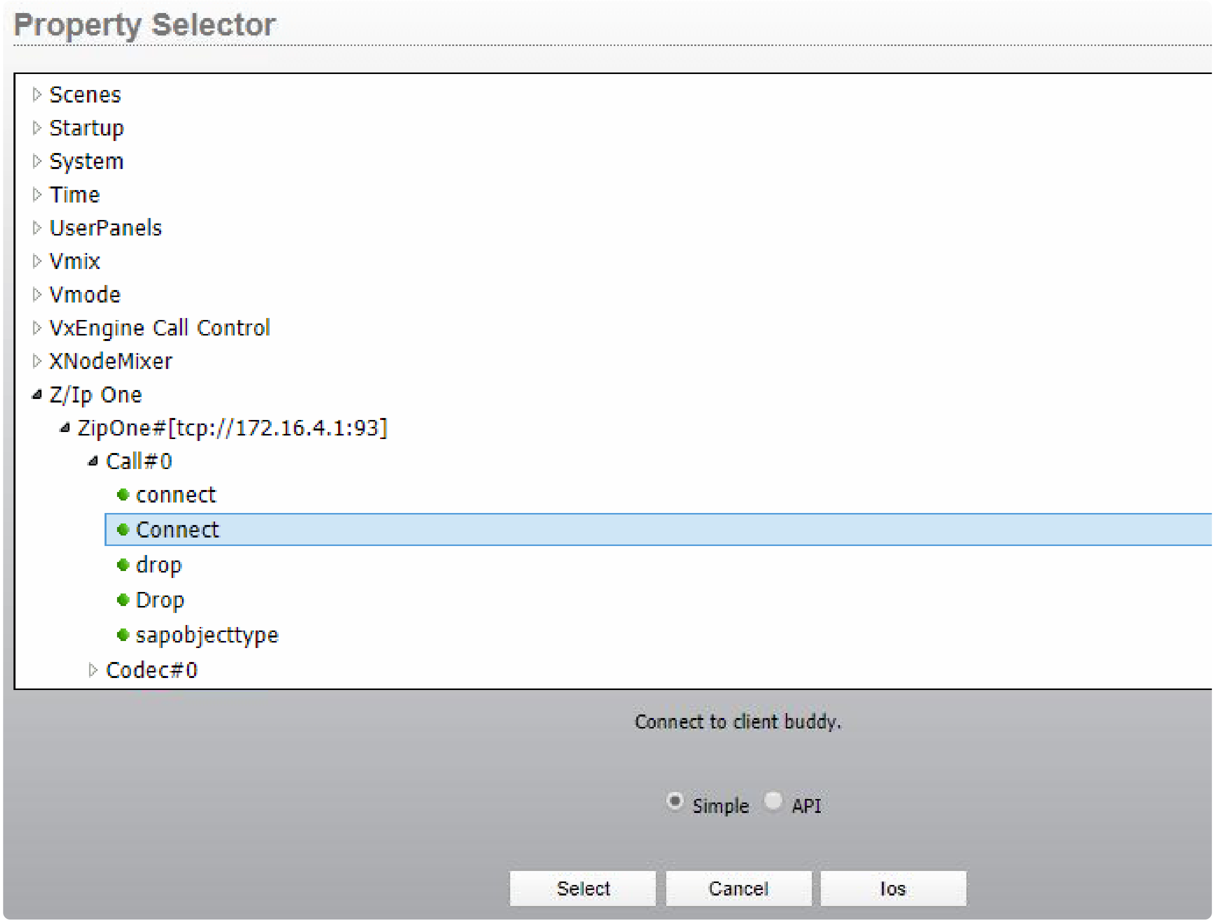
Templates:

- ☑ MyRecurring
- □ blah
- ☑ Studio1
- ☑ Studio2
- ☑ Studio3

Supply a name and description for the calendar and then select the resources and templates that are available for use. It is important to note that the same resources and templates can be used on multiple calendars. Think of a calendar as a view of resource based events and the templates as the available events that may be applied. After applying the new calendar settings, it becomes available in the calendar list and the view button will allow a user to start working with it.

Zip One Dialer Example

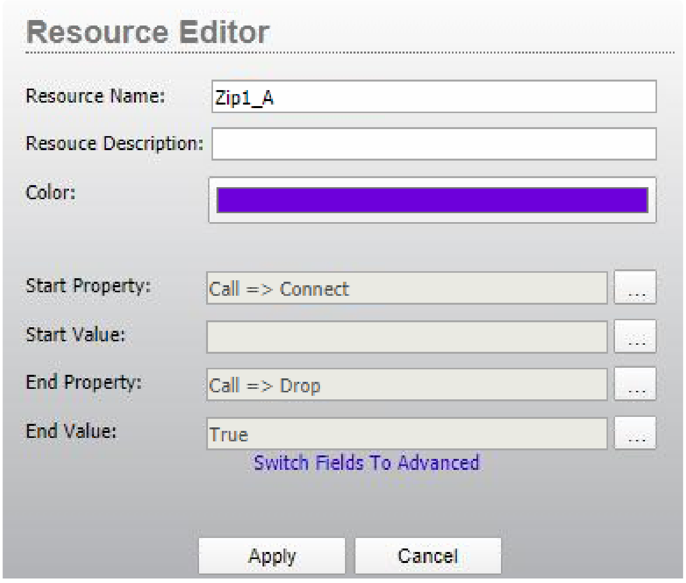
While all of the examples so far have been to make route changes, they do not have to be. Any property in the system can be used in a resource or template and therefore scheduled. So for example we could change the resource and template to use ZipOne dialing parameters. When creating the resource after clicking the ellipsis button, use the Endpoints button to obtain the normal logic flows endpoint selector.



Connect to client buddy.

● Simple ○ API

Therefore if we create our resource to look like:



Resource Name: Zip1_A

Resouce Description:

Color:

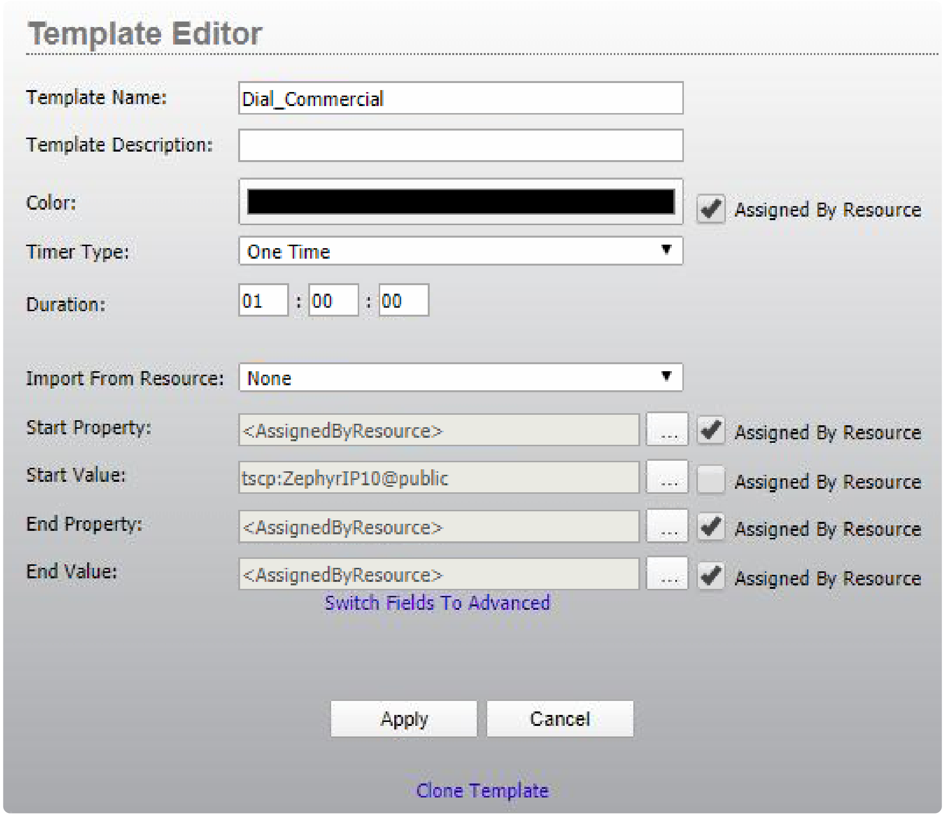
Start Property: Call => Connect ...

Start Value: ...

End Property: Call => Drop ...

End Value: True ...

And our template to look like:



Template Editor

Template Name: Dial_Commercial

Template Description:

Color: ■ ☑ Assigned By Resource

Timer Type: One Time

Duration: 01 : 00 : 00

Import From Resource: None

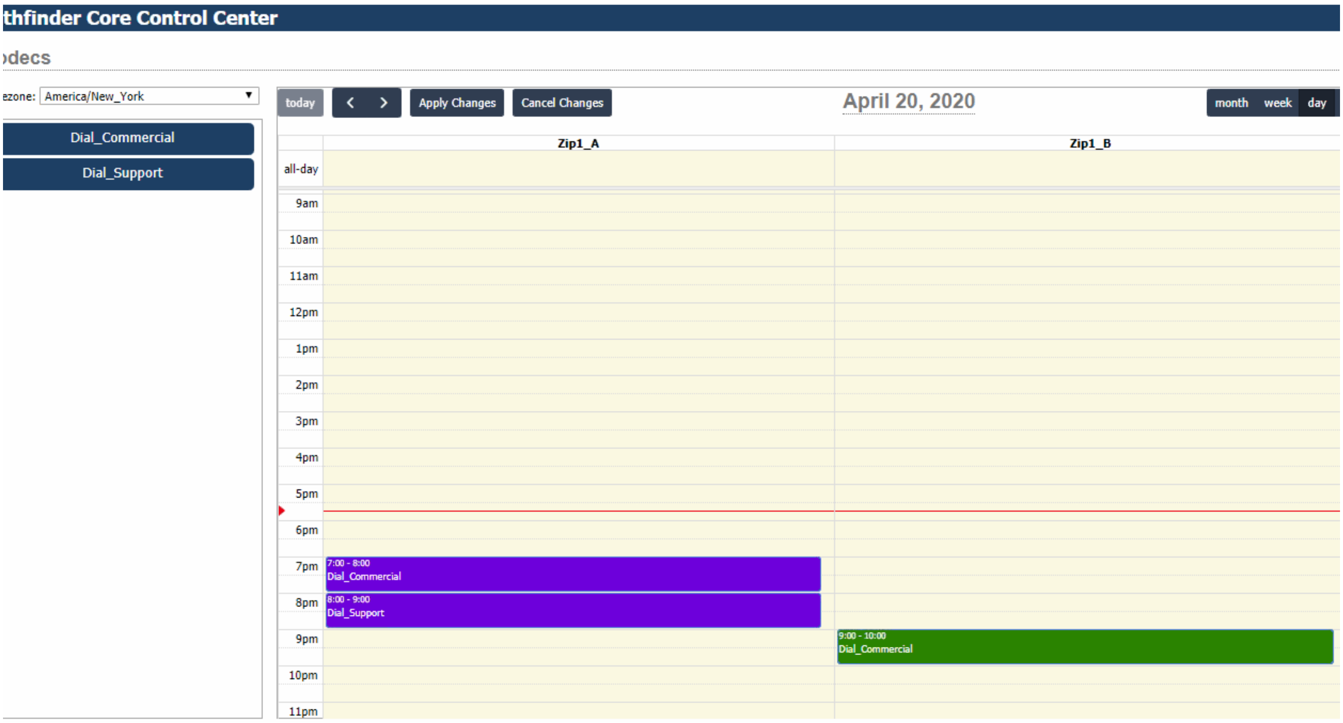
Start Property: <AssignedByResource> ☑ Assigned By Resource

Start Value: tscp:ZephyrIP10@public □ Assigned By Resource

End Property: <AssignedByResource> ☑ Assigned By Resource

End Value: <AssignedByResource> ☑ Assigned By Resource

In this instance the resource defines the Zip one Connect property as well as the Zip one drop property and value. And the template only supplies the dial location. Using this methodology we could create a calendar for scheduling Zip one dialing such as:



thfinder Core Control Center

ɔdecs

ezone: America/New_York

April 20, 2020

Zip1_A: 7:00 - 8:00 Dial_Commercial

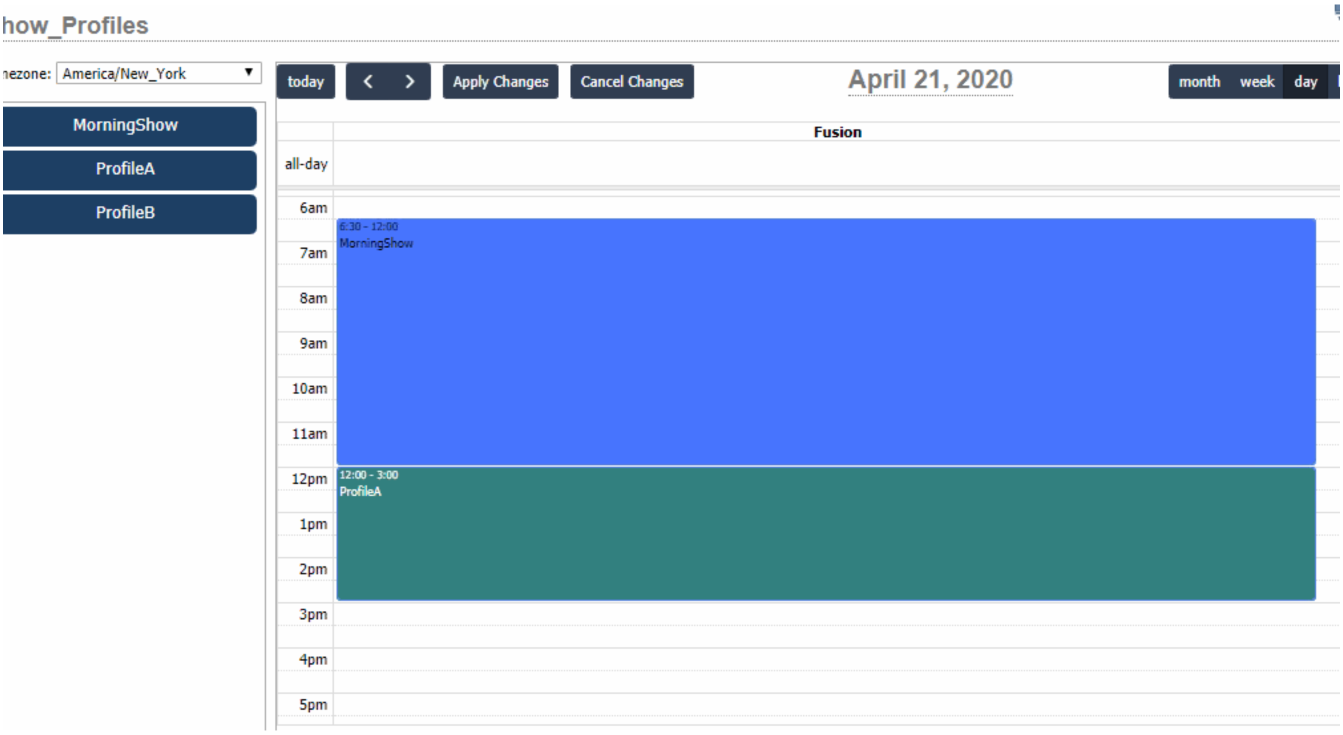
Zip1_A: 8:00 - 9:00 Dial_Support

Zip1_B: 9:00 - 10:00 Dial_Commercial

In this example our event templates are dial locations and our resources are the Zip One to use for the dialing.

Fusion Show Profiles

As one last example, we could create a calendar to schedule the loading of show profiles on a console:



In this example, the templates are show profiles and the only resource on the calendar is the fusion console on which they are being scheduled.

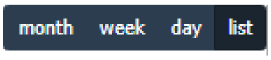
The point we are trying to make with these examples is that anything Pathfinder Core PRO knows about can be used for scheduling and multiple calendars may be created in order to schedule different types of things.

Timezone Selector

The calendar has a time zone selector in the top left corner. By default when the calendar is displayed, it queries PathfinderCore PRO for its current time zone and sets the calendar time zone to that time zone. Changing the time zone changes how the events are viewed. Adding events while in a different time zone than the Pathfinder Core PRO time zone will cause the event times to be converted to the Pathfinder Core PRO time zone when the apply button sends the changes to the system.

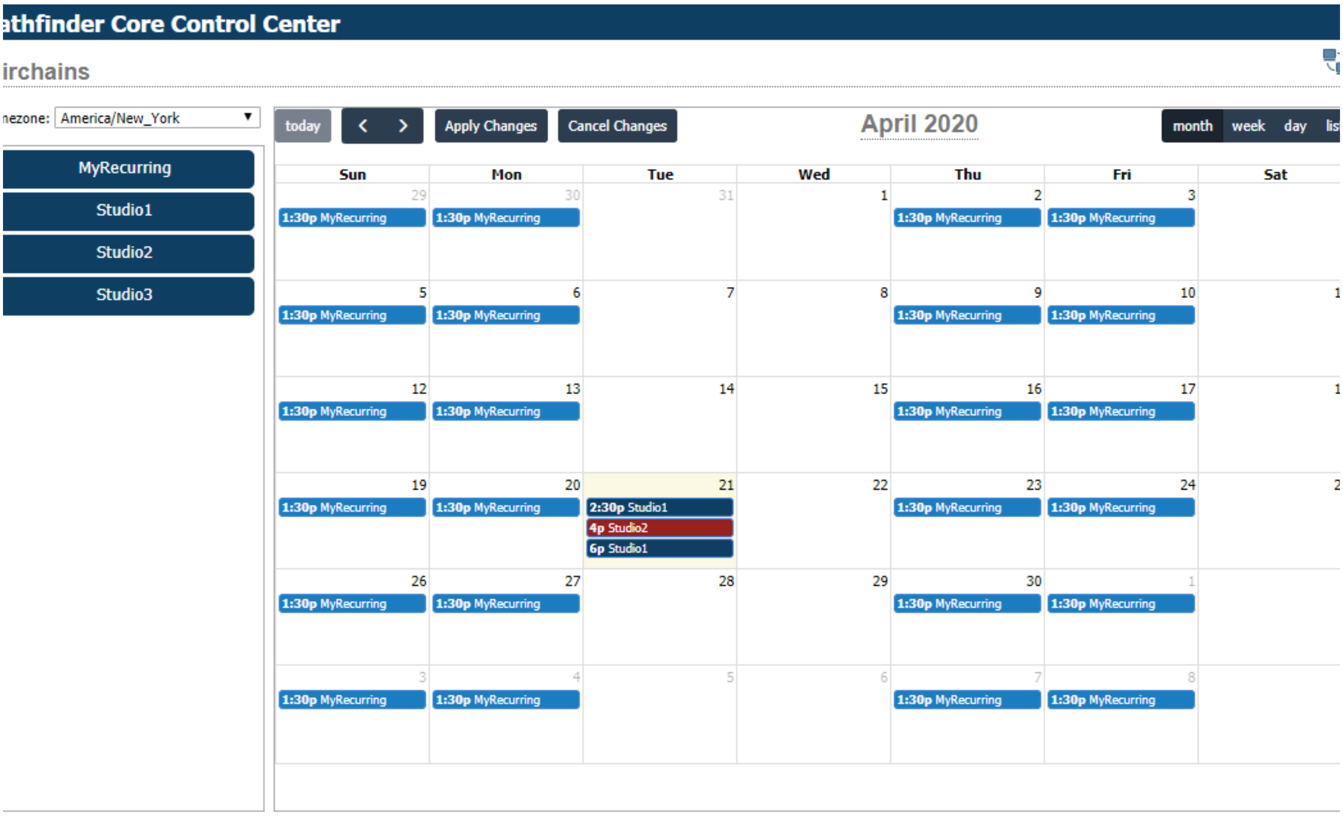
Calendar Controls and Views

Several controls exist across the top of the calendar. On the right hand corner, there are several buttons which change the view.

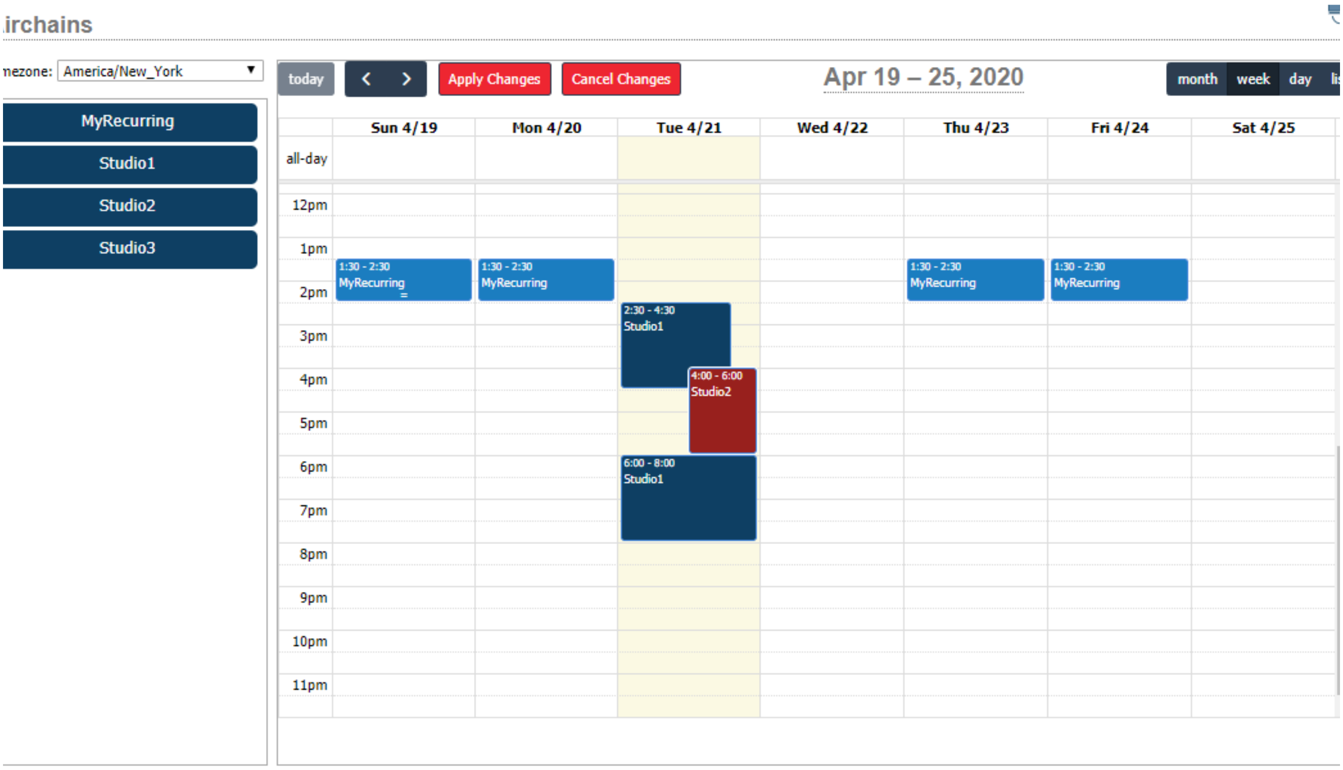


Clicking these buttons will change the calendar view to one of the following:

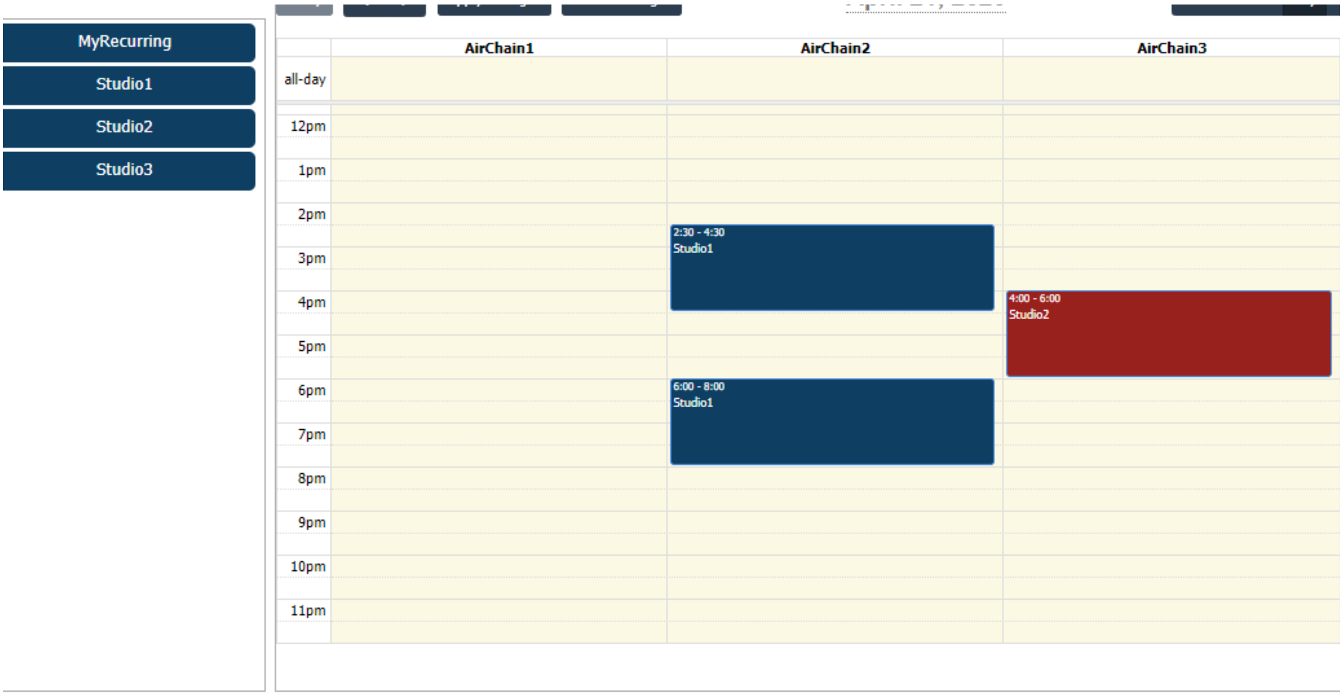
Month: Dropping templates onto the month view is prohibited as there is not enough resolution for time selection.



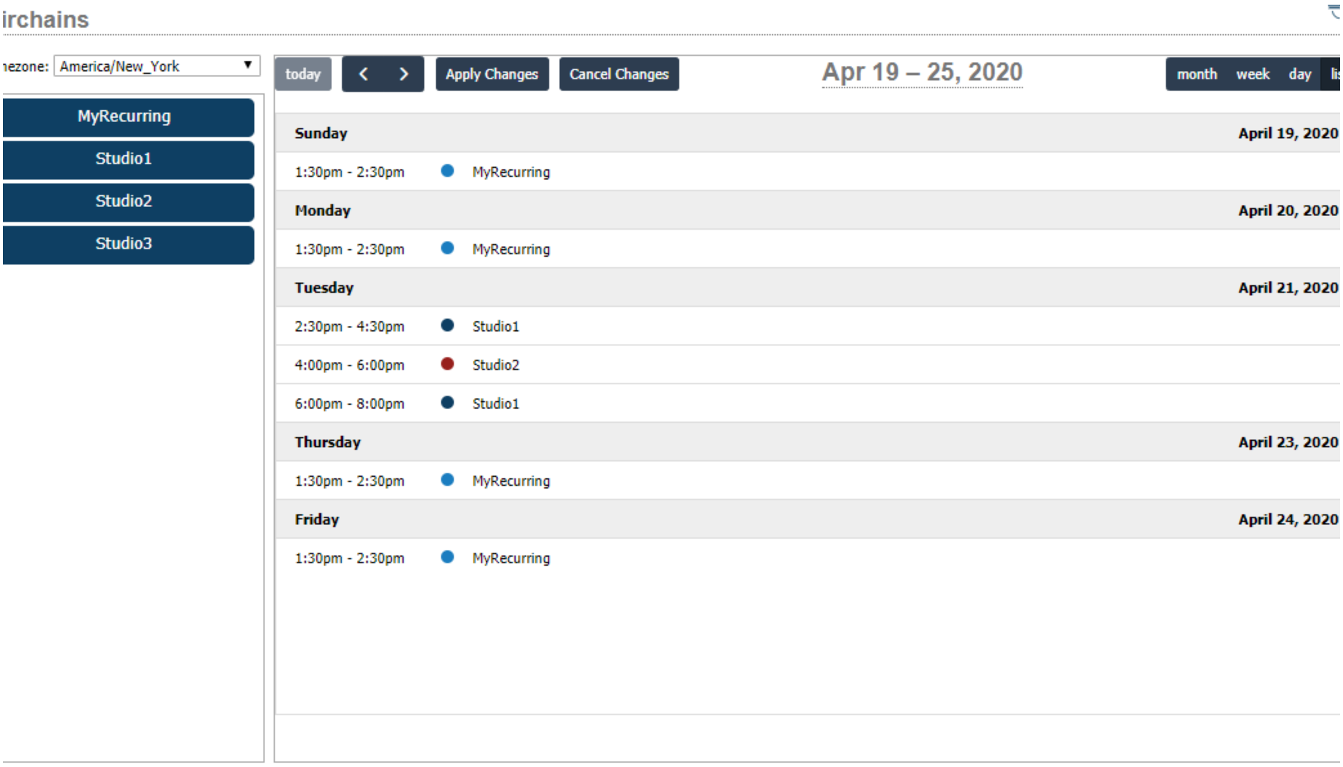
Week: Dropping a template onto a week view will always assign the template to the first resource assigned to the calendar. This can then be adjusted if necessary in the day view.



Day: This is the primary and default scheduling view.



List: This view is for informational purposes only and cannot be a drag and drop target.



Sunday April 19, 2020

1:30pm - 2:30pm ● MyRecurring

Monday April 20, 2020

1:30pm - 2:30pm ● MyRecurring

Tuesday April 21, 2020

2:30pm - 4:30pm ● Studio1

4:00pm - 6:00pm ● Studio2

6:00pm - 8:00pm ● Studio1

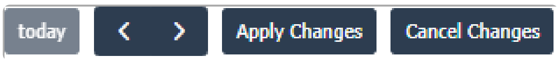
Thursday April 23, 2020

1:30pm - 2:30pm ● MyRecurring

Friday April 24, 2020

1:30pm - 2:30pm ● MyRecurring

On the left hand side is a today button that will bring the view back to the current day as well as left right arrows for navigating to the next day/week/month depending on the view.



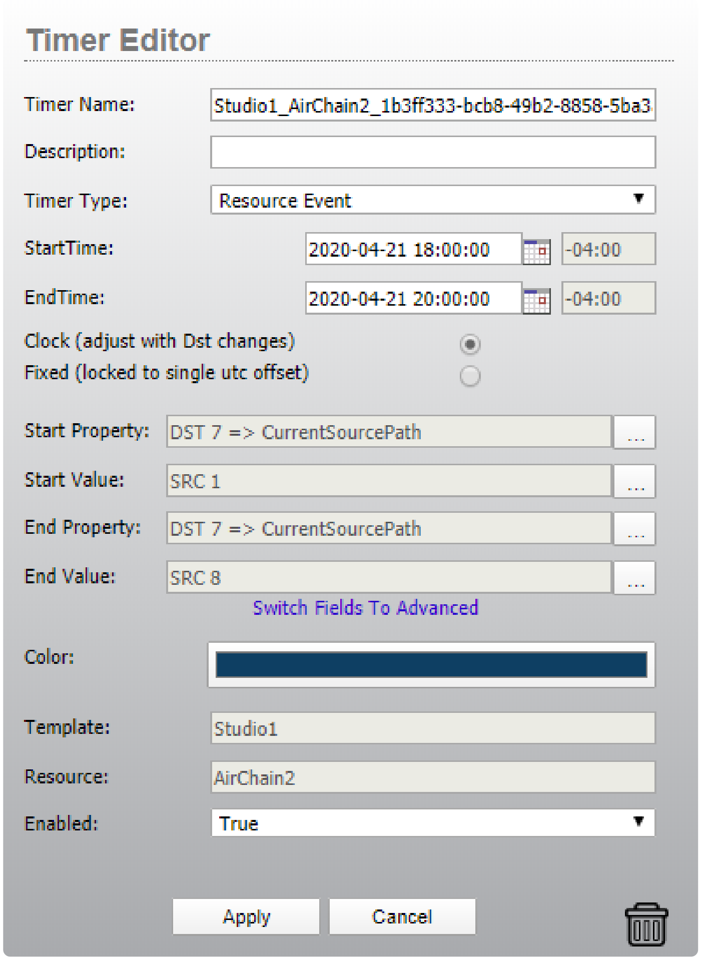
The apply and cancel buttons apply or cancel any pending changes to the Pathfinder Core PRO timer engine. These buttons will flash if there are pending changes. Events which have not been applied will not exist or be acted upon by PathfinderCore PRO.

Colors

You can define colors in the templates, resources, and in the events themselves and there is some internal logic as to which color is used when an event gets created. If the template has the Assigned by Resource checkbox enabled for the color then the resource color will be used. This is the default because when viewing the calendar using a week or month view where the resource columns are not visible, events associated with a specific resource will all have the same color making it easy to tell which events belong to which resource. However you may prefer to have the color relate to the template that created the event, in which case deselecting the Assigned by Resource checkbox in the template will cause newly created events to use the template color. Finally, once an event is created it can be edited by clicking on the event and a color specific to that event instance can be selected.

Event Editing

After an event is created, if the settings generated by the resource/template combination are not what you want, you can click on the event and edit it to do whatever you wish.



Timer Editor

Timer Name:	Studio1_AirChain2_1b3ff333-bcb8-49b2-8858-5ba3
Description:	
Timer Type:	Resource Event
StartTime:	2020-04-21 18:00:00 -04:00
EndTime:	2020-04-21 20:00:00 -04:00

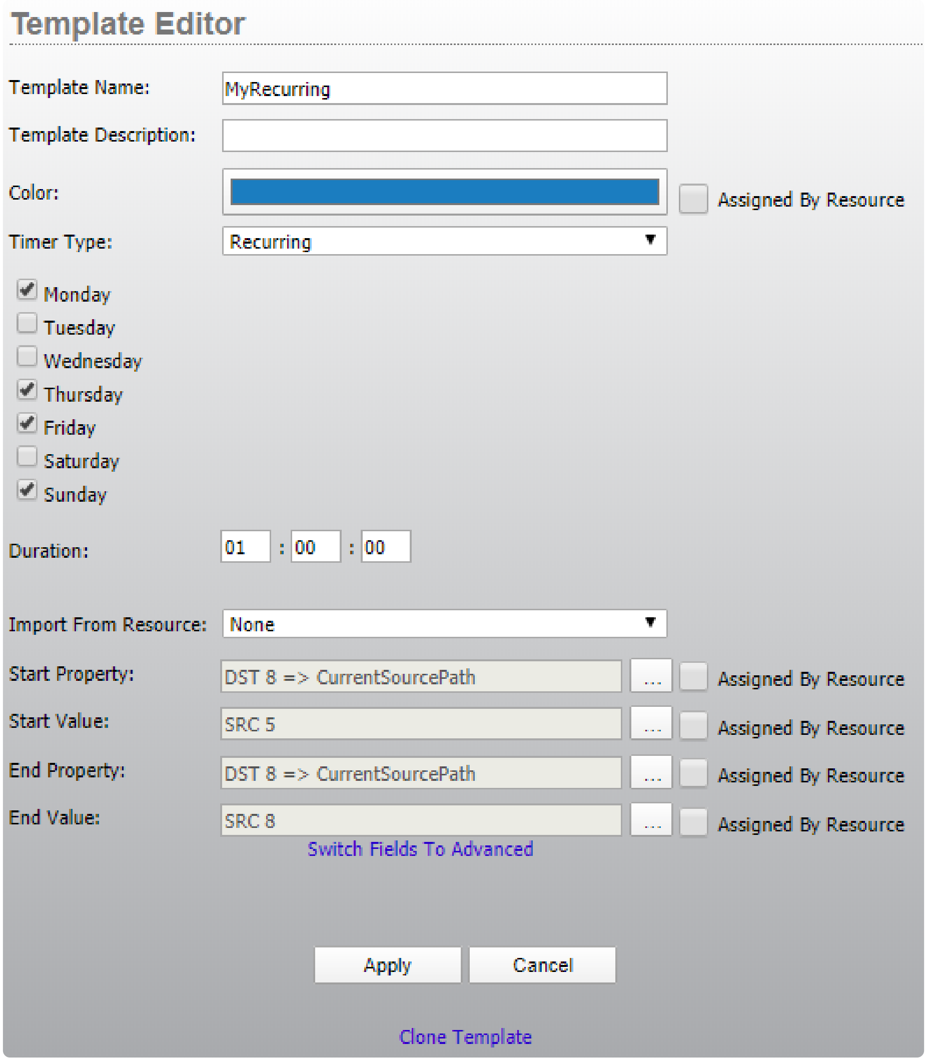
Clock (adjust with Dst changes) ●
Fixed (locked to single utc offset) ○

Start Property:	DST 7 => CurrentSourcePath
Start Value:	SRC 1
End Property:	DST 7 => CurrentSourcePath
End Value:	SRC 8
Color:	
Template:	Studio1
Resource:	AirChain2
Enabled:	True

This is basically the same timer editing dialog that is discussed in the timers section of the manual above. The only differences are the presence of the color, template, and resource fields and different options in the timer type field. Calendar based timers must be of type Resource Event or Recurring Resource Event. These two types equate to the DateTime event type and the DayOfWeek type but are designed to work with the calendar. You can change the name of the event from the default one that is generated, but the name must be unique within the timers in the system.

Recurring Events

When creating an event template or editing an event, it is possible to create it as a recurring event rather than a one-time event by selecting the Recurring timer type.



Template Editor

Template Name: MyRecurring

Template Description:

Color: 🟦 □ Assigned By Resource

Timer Type: Recurring ▼

☑ Monday
□ Tuesday
□ Wednesday
☑ Thursday
☑ Friday
□ Saturday
☑ Sunday

Duration: 01 : 00 : 00

Import From Resource: None ▼

Start Property: DST 8 => CurrentSourcePath ... □ Assigned By Resource

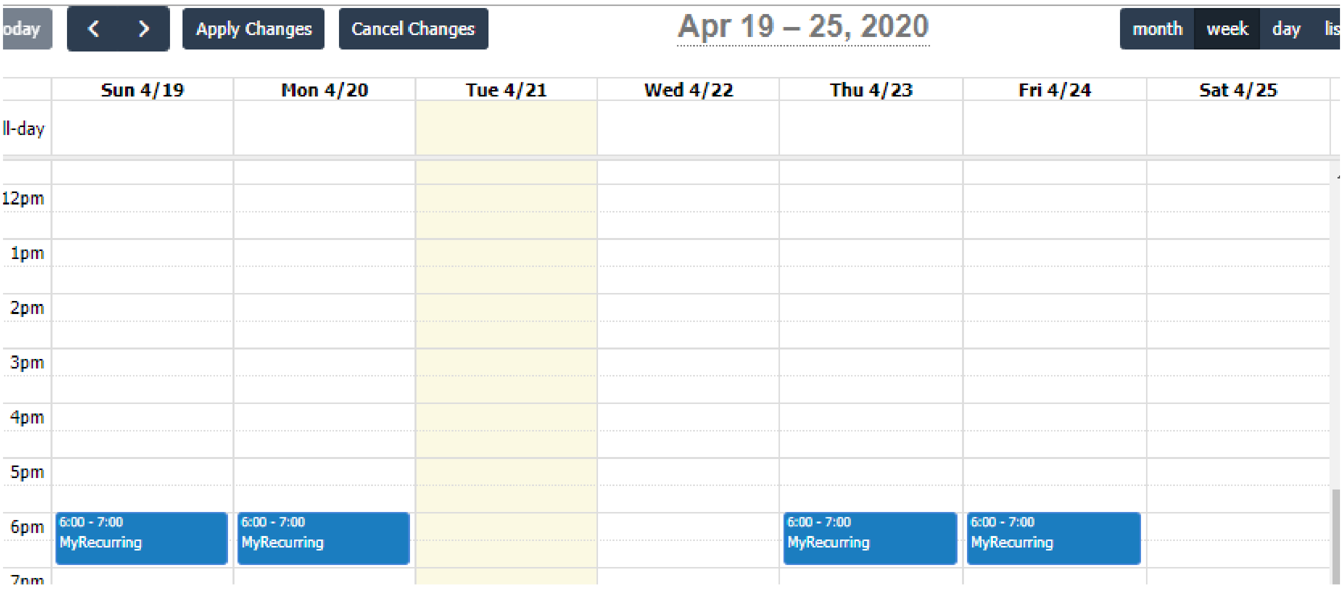
Start Value: SRC 5 ... □ Assigned By Resource

End Property: DST 8 => CurrentSourcePath ... □ Assigned By Resource

End Value: SRC 8 ... □ Assigned By Resource

This will add fields for selecting the days of the week on which the event should occur.

Dragging a recurring template onto the calendar will cause it to appear at the selected timeslot on all days of the week for which the template is configured.



Apr 19 – 25, 2020

	Sun 4/19	Mon 4/20	Tue 4/21	Wed 4/22	Thu 4/23	Fri 4/24	Sat 4/25
ll-day							
12pm							
1pm							
2pm							
3pm							
4pm							
5pm							
6pm	6:00 - 7:00 MyRecurring	6:00 - 7:00 MyRecurring			6:00 - 7:00 MyRecurring	6:00 - 7:00 MyRecurring	
7pm							

One thing that can cause confusion is if you drag a recurring event to a day view on a day of the week that the recurring event is not scheduled to take place. This will appear to have no effect, but if you use the week view to zoom out you will see that the event has been added and applied to the correct days.

It is also important to note than changing the length or any of the properties of a recurring event will make that change in the base recurring event and will therefore affect all days the event takes place. For example, in the week view above, if you grab the duration handle on the bottom of one of these events and make the event longer, all of them will change because they are all just views of a single recurring event.

Deleting an Event

When editing an event, you should also notice the trashcan icon in the bottom right corner.

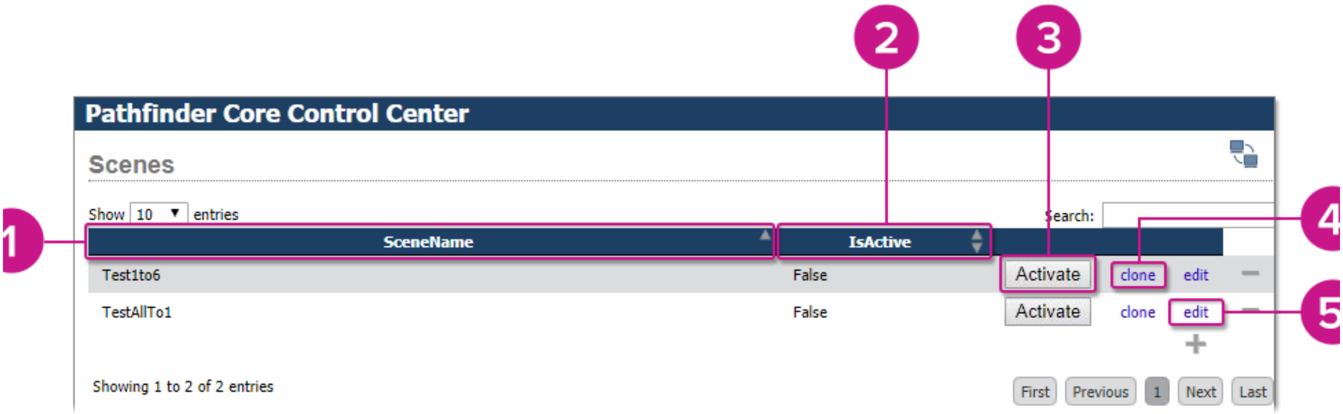
🗑

To delete an event that has been added to a calendar resource, click on the event and then click on the trash can icon. The system will ask you if you are sure you want to delete the event. It is important to note that deleting a recurring event will delete all instances of that recurring event. There is currently no way to delete a single day of a recurring event. Instead you would need to disable that day in the event and then re-enable it afterwards if you want it to happen on that day in the future.

Scenes

A **Scene** is nothing more than a list of property changes. A Scene does not have to be just a route change or even include a route change. You could create a Scene that sets memory slots to specific values or that includes memory slot changes, VMix changes, fader changes, and button state changes. Any property available in the system can be a part of a Scene.

To create a new Scene or view and edit an existing Scene, click the **Scenes** icon in the navigation bar to open the Scenes page:



SceneName	IsActive			
Test1to6	False	Activate	clone	edit
TestAllTo1	False	Activate	clone	edit

Showing 1 to 2 of 2 entries

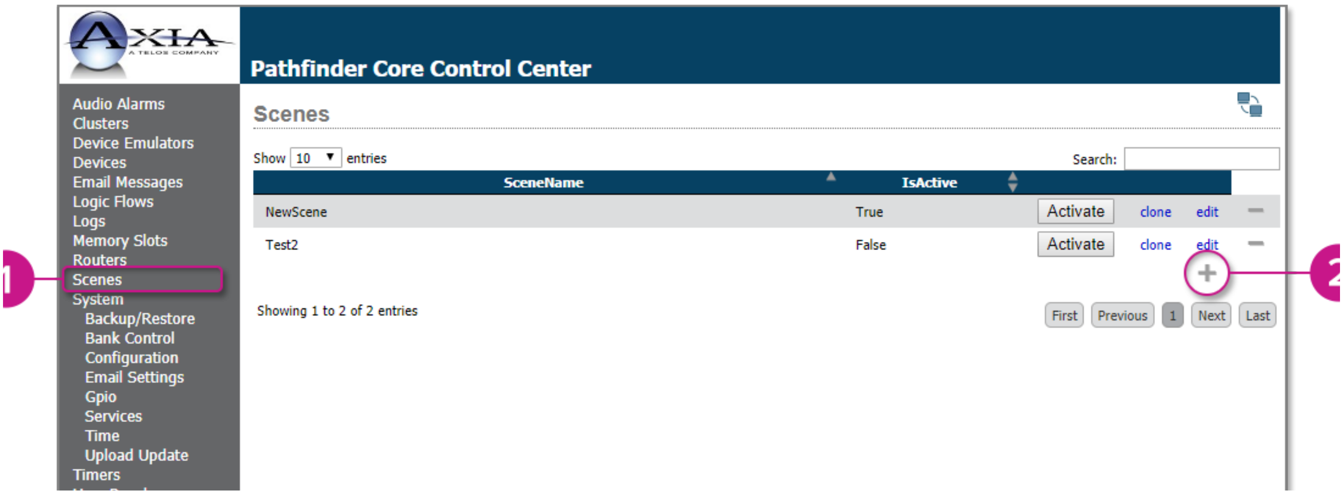
Reference	Notes

1	The **SceneName** column lists the name of the Scene. Scene names may be preceded by a number and an underscore in some cases, indicating Scenes created by *PathfinderPC_Core Client* related to a specific router number.
2	The **IsActive** column displays either *True* or *False* depending on the state of all entries in the Scene. This column updates dynamically with each Scene's current state. This property is also available to Logic Flows so a Flow can perform actions depending on the *IsActive* state. *See note below*.
3	The **Activate** button executes all changes within a Scene.
4	The **Clone** link creates a copy of the Scene. This useful if you need to create multiple Scenes with the same items but different values for each item.
5	The **Edit** link will open the Scene Editor and display the data for an existing Scene.

ⓘ Note: The *IsActive* state depends on the state of *all* entries in the Scene. For example, if an xNode involved in the Scene is powered off, the Scene change will not be completed and the *IsActive* state will not change to *True*.

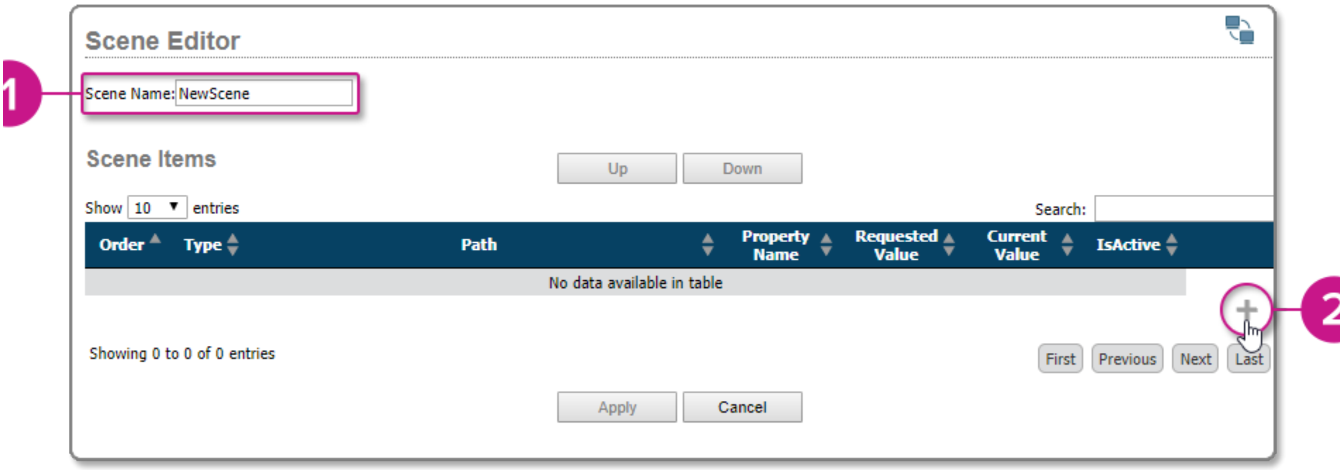
Creating Scenes

Begin by clicking the **Scenes** link in the left navigation pane in the Pathfinder Core Control Center. On the *Scenes* page, click the **plus icon** to create a new Scene.



SceneName	IsActive				
NewScene	True	Activate	clone	edit	−
Test2	False	Activate	clone	edit	−

Type a **Name** for the new Scene. To add events to your Scene, click the **plus icon**.



Scene Editor

Scene Name: NewScene

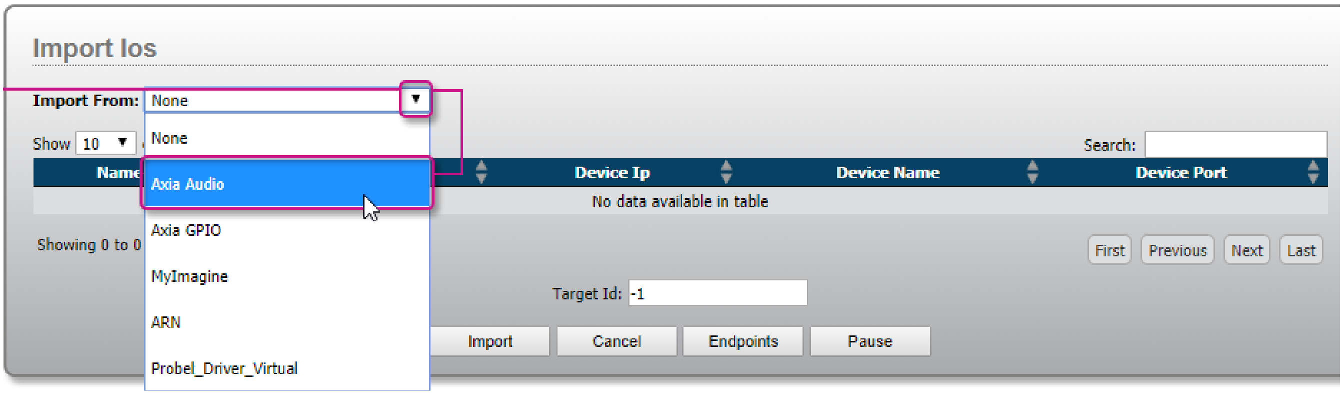
Scene Items

Show 10 entries Search:

Order	Type	Path	Property Name	Requested Value	Current Value	IsActive
No data available in table						

Showing 0 to 0 of 0 entries

On the **Import Ios** dialog, select the **router** from the *Import From* drop-down list to populate the *Destinations* list.



Import Ios

Import From: None

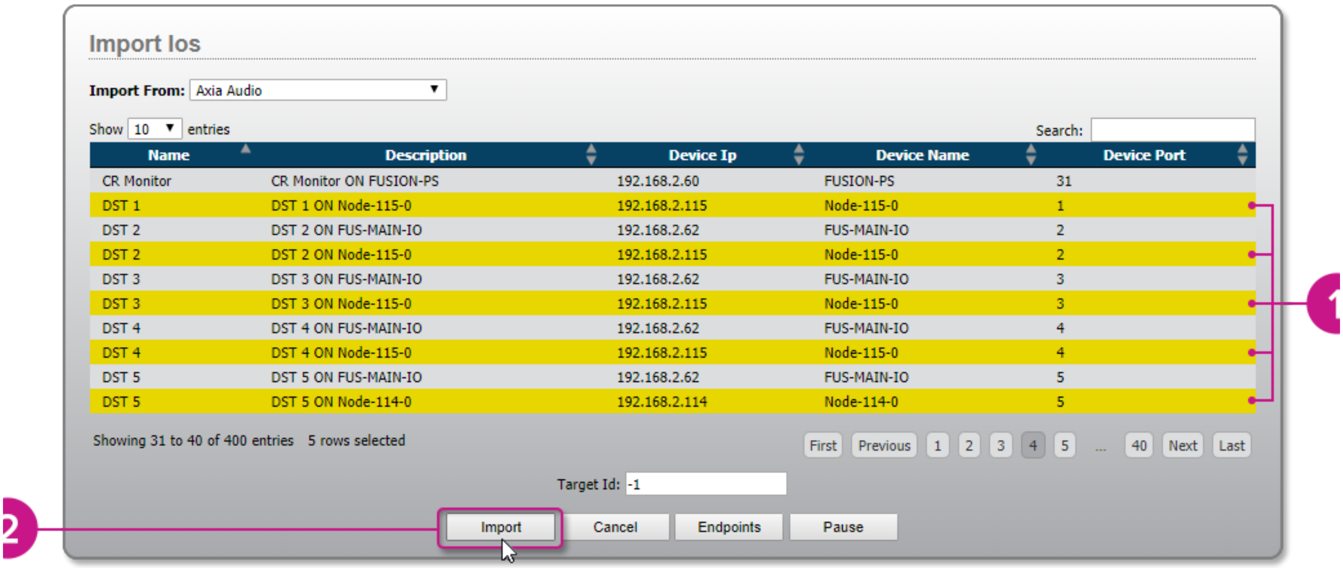
Show 10 Search:

Name	Device Ip	Device Name	Device Port
No data available in table			

Showing 0 to 0

Target Id: -1

Select the **destinations** to import, using the SHIFT key to select multiple destinations. When all destinations have been selected, click **Import**.



Import Ios

Import From: Axia Audio

Show 10 entries Search:

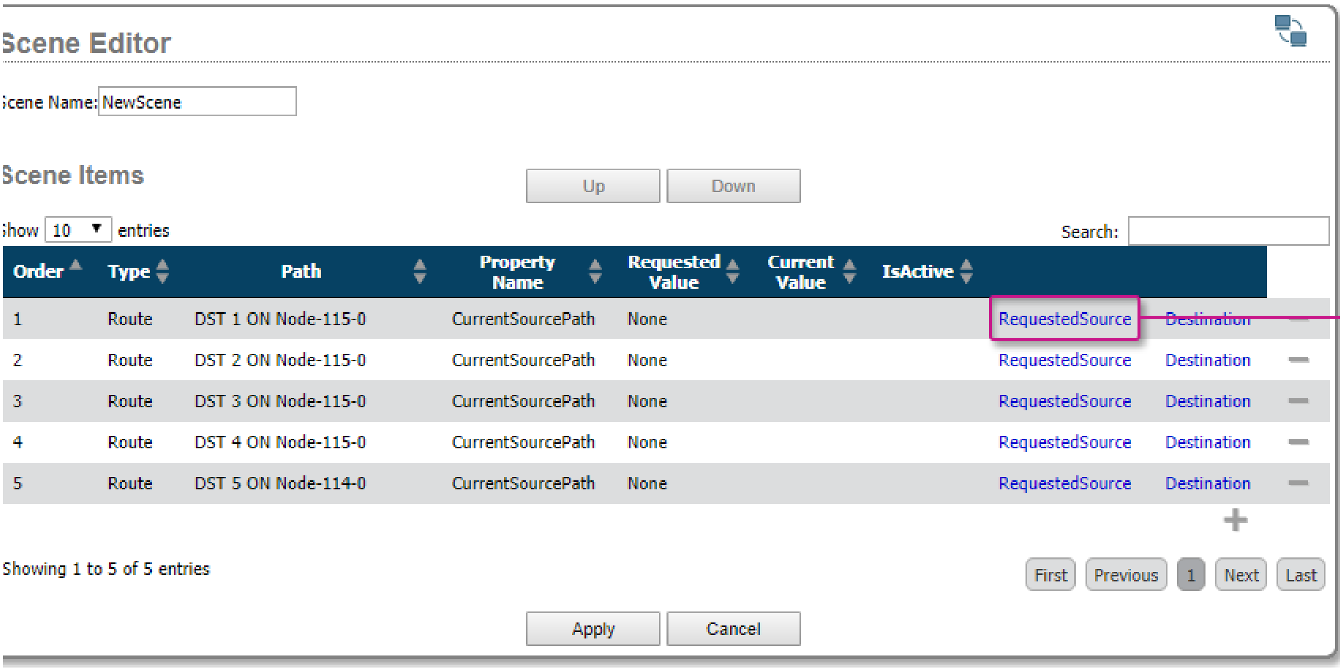
Name	Description	Device Ip	Device Name	Device Port
CR Monitor	CR Monitor ON FUSION-PS	192.168.2.60	FUSION-PS	31
DST 1	DST 1 ON Node-115-0	192.168.2.115	Node-115-0	1
DST 2	DST 2 ON FUS-MAIN-IO	192.168.2.62	FUS-MAIN-IO	2
DST 2	DST 2 ON Node-115-0	192.168.2.115	Node-115-0	2
DST 3	DST 3 ON FUS-MAIN-IO	192.168.2.62	FUS-MAIN-IO	3
DST 3	DST 3 ON Node-115-0	192.168.2.115	Node-115-0	3
DST 4	DST 4 ON FUS-MAIN-IO	192.168.2.62	FUS-MAIN-IO	4
DST 4	DST 4 ON Node-115-0	192.168.2.115	Node-115-0	4
DST 5	DST 5 ON FUS-MAIN-IO	192.168.2.62	FUS-MAIN-IO	5
DST 5	DST 5 ON Node-114-0	192.168.2.114	Node-114-0	5

Showing 31 to 40 of 400 entries 5 rows selected

Target Id: -1

ⓘ Note: Items from different routers may also be a part of the same Scene by clicking the **plus icon** again. Select the additional router from the routers drop-down, select the additional IOs, then click **Import** to bring them into the Scene.

After the import is complete, the **Requested Value** column in the Scene Editor will reflect the source that will be taken if that Scene item is executed. By default, it will be set to the current source routed to the destination. This can be changed by highlighting an item and clicking the **RequestedSource** link. This will open a list of sources present on the selected router so you can select an alternate source as the value for the route point in that Scene item.



Scene Editor

Scene Name: NewScene

Scene Items

Show 10 entries

Search:

Order	Type	Path	Property Name	Requested Value	Current Value	IsActive			
1	Route	DST 1 ON Node-115-0	CurrentSourcePath	None			RequestedSource	Destination	—
2	Route	DST 2 ON Node-115-0	CurrentSourcePath	None			RequestedSource	Destination	—
3	Route	DST 3 ON Node-115-0	CurrentSourcePath	None			RequestedSource	Destination	—
4	Route	DST 4 ON Node-115-0	CurrentSourcePath	None			RequestedSource	Destination	—
5	Route	DST 5 ON Node-114-0	CurrentSourcePath	None			RequestedSource	Destination	—

Showing 1 to 5 of 5 entries

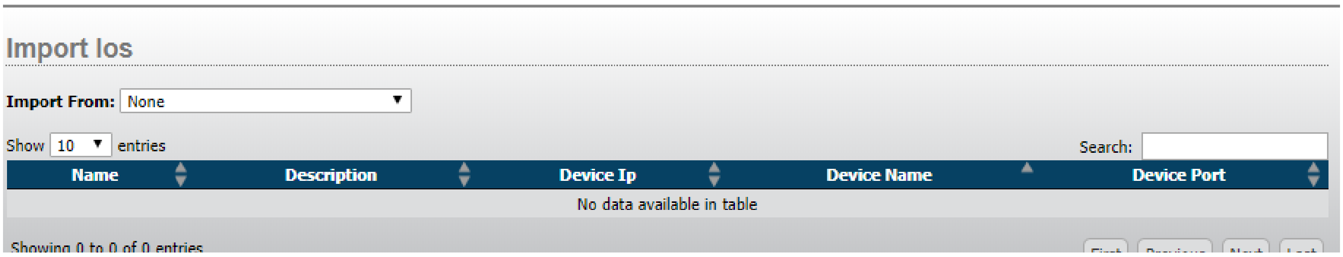
Clicking the **Destination** link will allow you to select a different destination for the Scene item if you selected the wrong one and need to change it.

Clicking the **minus icon** will delete the item from the Scene.

Once all items for this Scene are set, click **Apply** to save the changes and return to the Scene list, at which point the Scene will be available for use.

Property Scene Items

In addition to Route Points, any property in the system can be a Scene item. After clicking the Add button to add a new Scene item to a Scene, you will also find an **Endpoints** button to switch the route destination dialog to a property selection dialog.



Import Ios

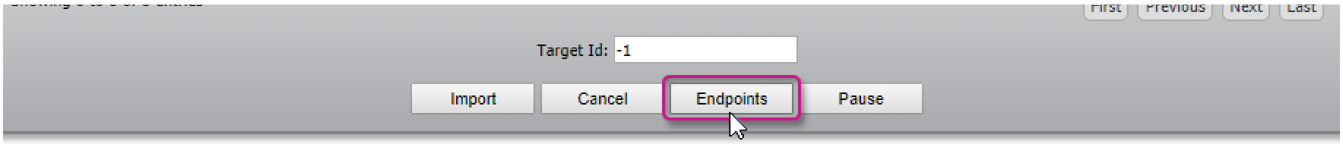
Import From: None

Show 10 entries

Search:

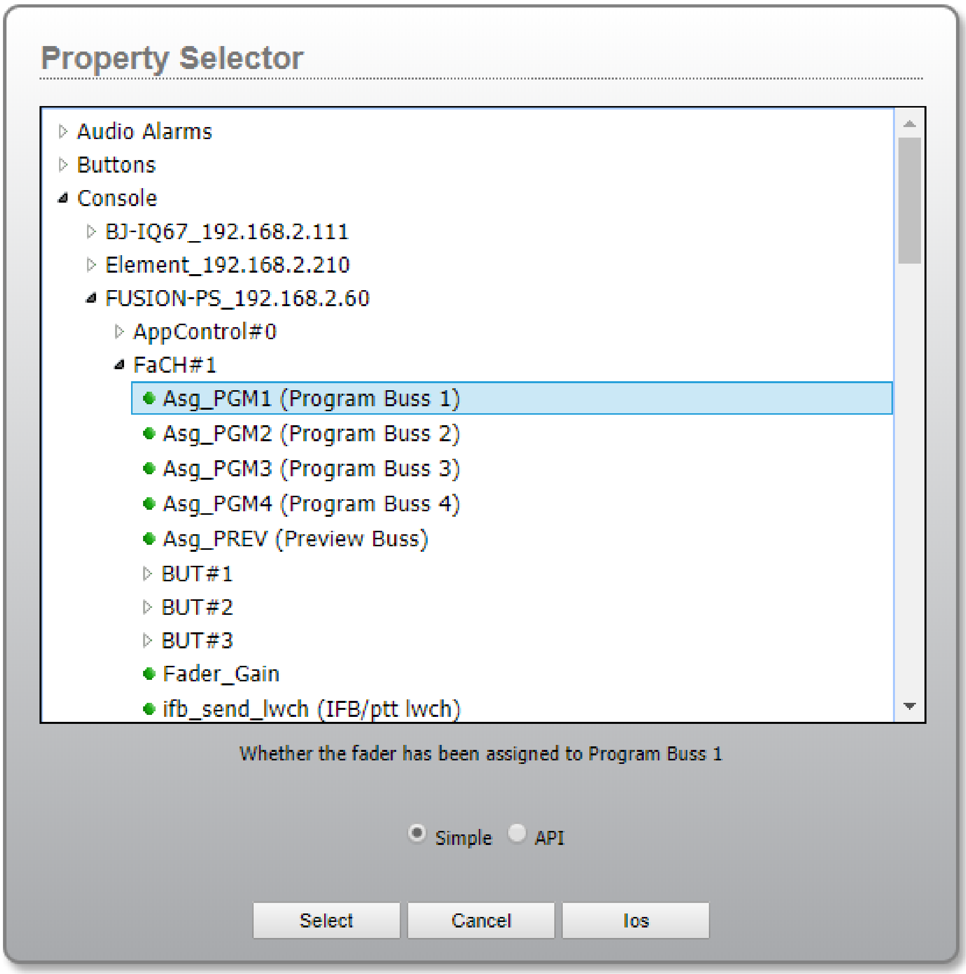
Name	Description	Device Ip	Device Name	Device Port
No data available in table				

Showing 0 to 0 of 0 entries



Target Id: -1

This is the same dialog used to select an endpoint in logic flows. In the example below, we are adding a program buss assignment property on a Fusion console to the Scene.



- Audio Alarms
- Buttons
- Console
  - BJ-IQ67_192.168.2.111
  - Element_192.168.2.210
  - FUSION-PS_192.168.2.60
    - AppControl#0
    - FaCH#1
      - ● Asg_PGM1 (Program Buss 1)
      - ● Asg_PGM2 (Program Buss 2)
      - ● Asg_PGM3 (Program Buss 3)
      - ● Asg_PGM4 (Program Buss 4)
      - ● Asg_PREV (Preview Buss)
      - BUT#1
      - BUT#2
      - BUT#3
      - ● Fader_Gain
      - ● ifb_send_lwch (IFB/ptt lwch)

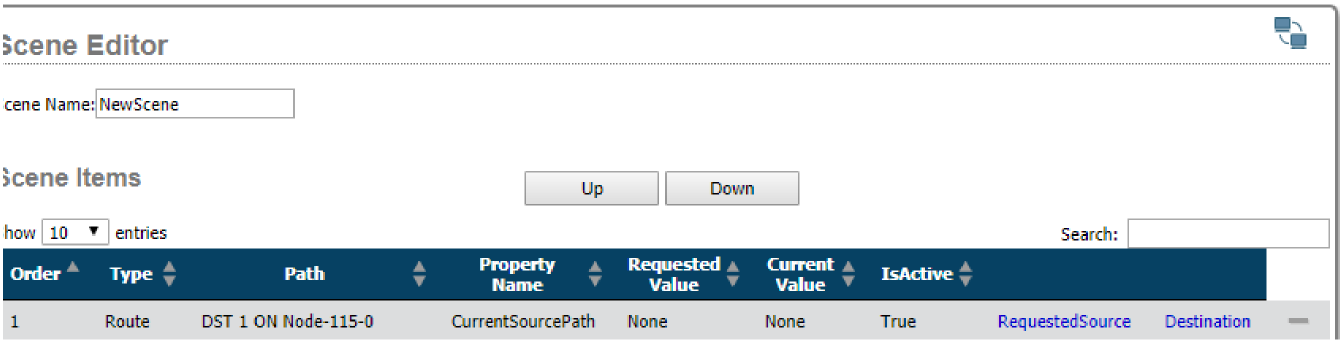
Whether the fader has been assigned to Program Buss 1

● Simple ○ API

The **Ios** button will return to the route selection dialog.

The **Select** button will select the property and import it as a Scene item into the Scene. The system will remember whether the last item imported during a given Scene editing session was a property or route point and return to the last used dialog. You can switch back and forth using the Endpoints and IOs buttons from the corresponding dialog.

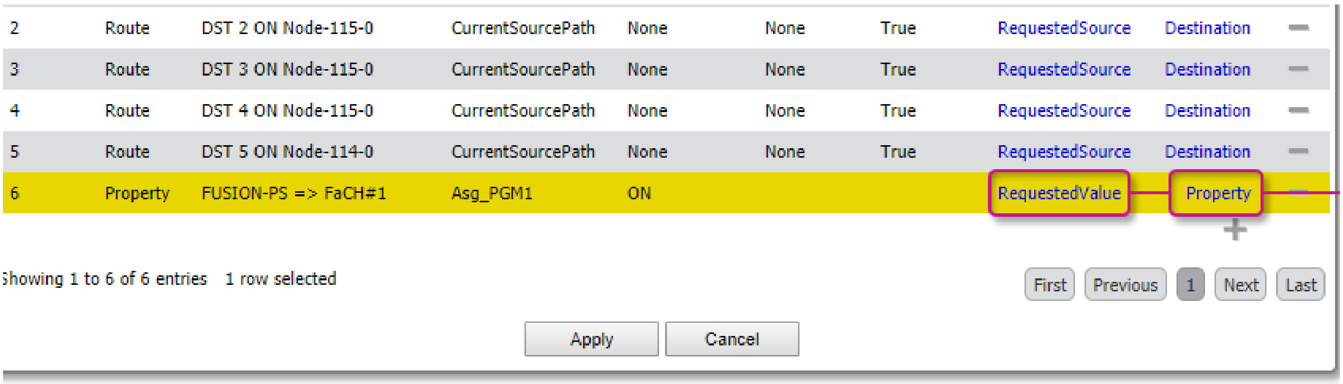
After importing a property, it will have slightly different links in the Scene itself. In this case, you will see a *RequestedValue* link and a *Property* link.



cene Name: NewScene

how 10 entries

Order	Type	Path	Property Name	Requested Value	Current Value	IsActive			
1	Route	DST 1 ON Node-115-0	CurrentSourcePath	None	None	True	RequestedSource	Destination	—

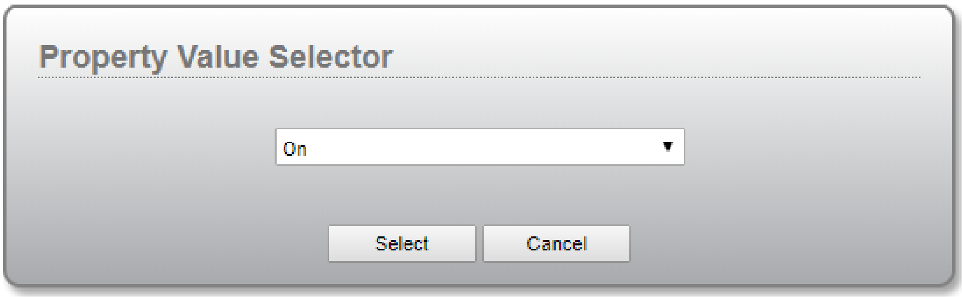


2	Route	DST 2 ON Node-115-0	CurrentSourcePath	None	None	True	RequestedSource	Destination	—
3	Route	DST 3 ON Node-115-0	CurrentSourcePath	None	None	True	RequestedSource	Destination	—
4	Route	DST 4 ON Node-115-0	CurrentSourcePath	None	None	True	RequestedSource	Destination	—
5	Route	DST 5 ON Node-114-0	CurrentSourcePath	None	None	True	RequestedSource	Destination	—
6	Property	FUSION-PS => FaCH#1	Asg_PGM1	ON			RequestedValue	Property	—

howing 1 to 6 of 6 entries 1 row selected

Click the **Property** link to change the property you selected if you selected the wrong one.

By default, the value used for the Property item is set to whatever it is at the time the Scene item is created. This can be changed by clicking the **RequestedValue** link.



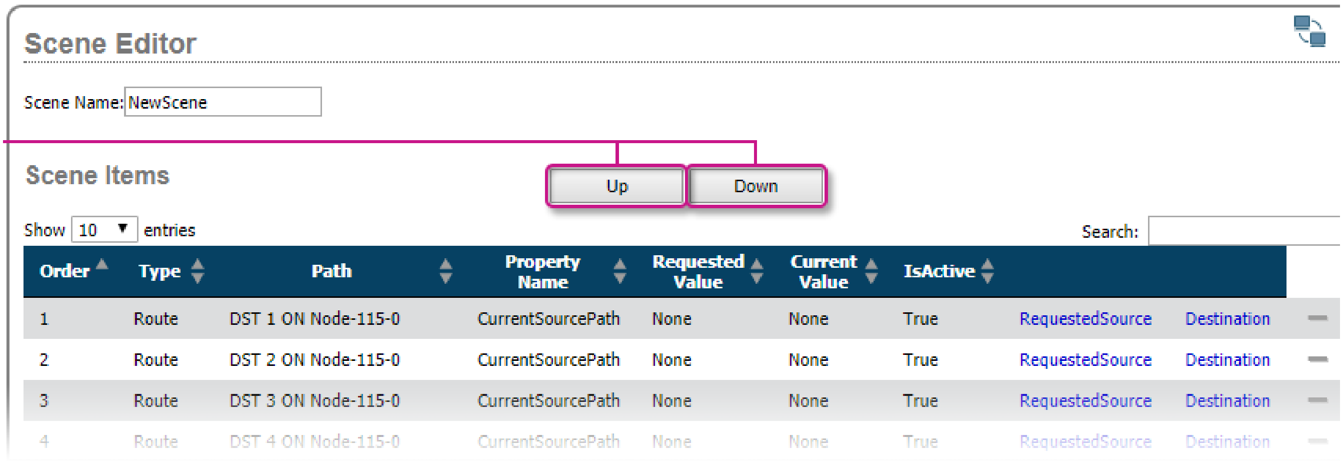
Property Value Selector

On

This will open a value selector. The value selector may be different depending on the type of property you are manipulating. For example, if the item is a color property, it might present a color selection dialog. If the property requires text, a text box may be presented. Or it could present a drop-down of possible values. In the case above, the program buss assignment could either be on or off, so select the option you want the Scene item to set. Note this does not change the current state of the console in the system unless the Scene is activated. At this point, you are only editing what would happen if you activated the Scene.

Modifying Scene Order

The Scene editor also has “move up” and “move down” buttons.



Scene Editor

Scene Name: NewScene

Scene Items

Show 10 entries

Search:

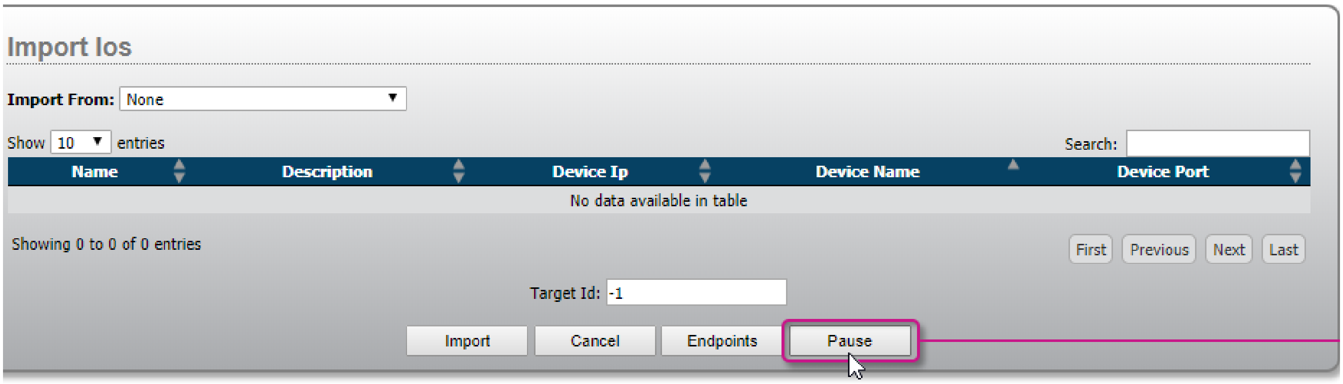
Order	Type	Path	Property Name	Requested Value	Current Value	IsActive			
1	Route	DST 1 ON Node-115-0	CurrentSourcePath	None	None	True	RequestedSource	Destination	—
2	Route	DST 2 ON Node-115-0	CurrentSourcePath	None	None	True	RequestedSource	Destination	—
3	Route	DST 3 ON Node-115-0	CurrentSourcePath	None	None	True	RequestedSource	Destination	—
4	Route	DST 4 ON Node-115-0	CurrentSourcePath	None	None	True	RequestedSource	Destination	—

These are used to organize the order in which a Scene sends its messages to the equipment when activated.

The order of items in a Scene is generally irrelevant because the system does not wait for a change to complete before sending the next change message. Changes are usually sent as a block so the actual changes may not happen in the equipment in the same order in which they were sent.

Inserting Pauses

If the execution order is important it is possible to add pause items into the Scene. This will cause the Scene to pause as it iterates through the items for a defined duration, measured in milliseconds. To add a pause, click the Scene editor **plus icon**. From the **Import Ios** dialog, click **Pause**.



Import Ios

Import From: None

Show 10 entries

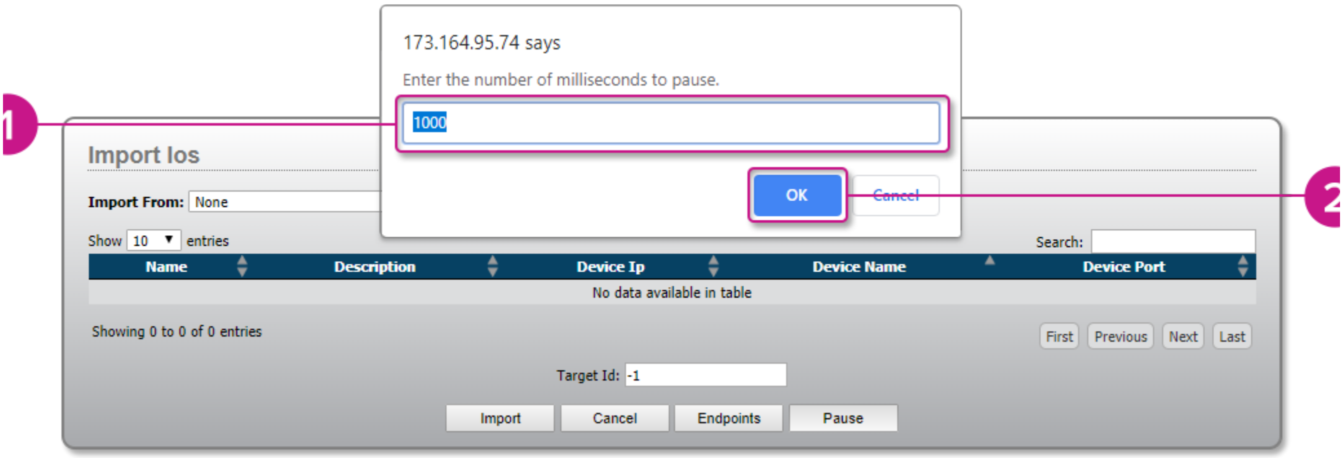
Search:

Name	Description	Device Ip	Device Name	Device Port
No data available in table				

Showing 0 to 0 of 0 entries

Target Id: -1

Type the **Pause duration** in milliseconds and click **OK** to add the Pause item to the Scene.



173.164.95.74 says

Enter the number of milliseconds to pause.

1000

Import Ios

Import From: None

Show 10 entries

Search:

Name	Description	Device Ip	Device Name	Device Port
No data available in table				

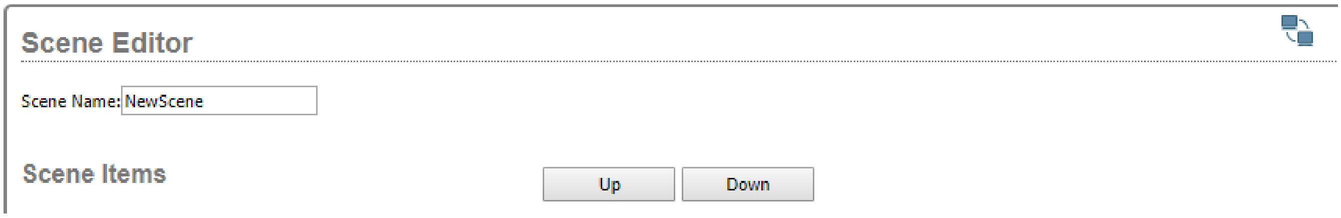
Showing 0 to 0 of 0 entries

Target Id: -1

Like all items in the Scene, you may select a Pause item and use the **Up** and **Down** arrows to manipulate the order of the Scene messages and configure when pauses occur in the Scene execution.

Saving Your Scene

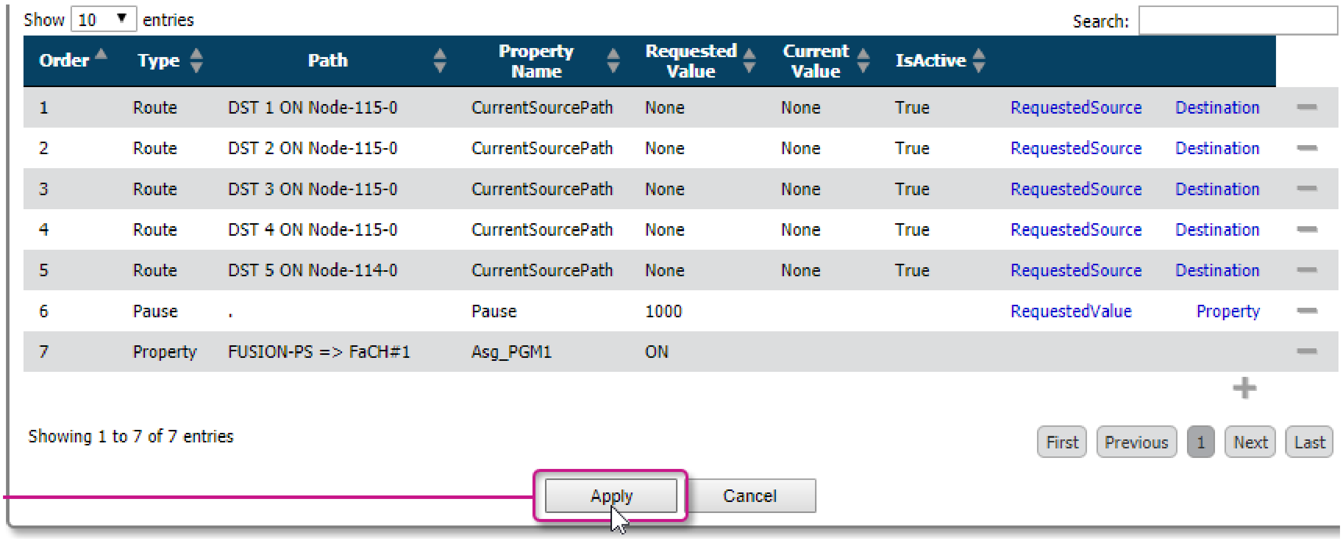
Once all items have been added to your Scene and have been ordered correctly, click **Apply** to save your changes.



Scene Editor

Scene Name: NewScene

Scene Items



Show 10 ▾ entries

Search:

Order	Type	Path	Property Name	Requested Value	Current Value	IsActive			
1	Route	DST 1 ON Node-115-0	CurrentSourcePath	None	None	True	RequestedSource	Destination	—
2	Route	DST 2 ON Node-115-0	CurrentSourcePath	None	None	True	RequestedSource	Destination	—
3	Route	DST 3 ON Node-115-0	CurrentSourcePath	None	None	True	RequestedSource	Destination	—
4	Route	DST 4 ON Node-115-0	CurrentSourcePath	None	None	True	RequestedSource	Destination	—
5	Route	DST 5 ON Node-114-0	CurrentSourcePath	None	None	True	RequestedSource	Destination	—
6	Pause	.	Pause	1000			RequestedValue	Property	—
7	Property	FUSION-PS => FaCH#1	Asg_PGM1	ON					—

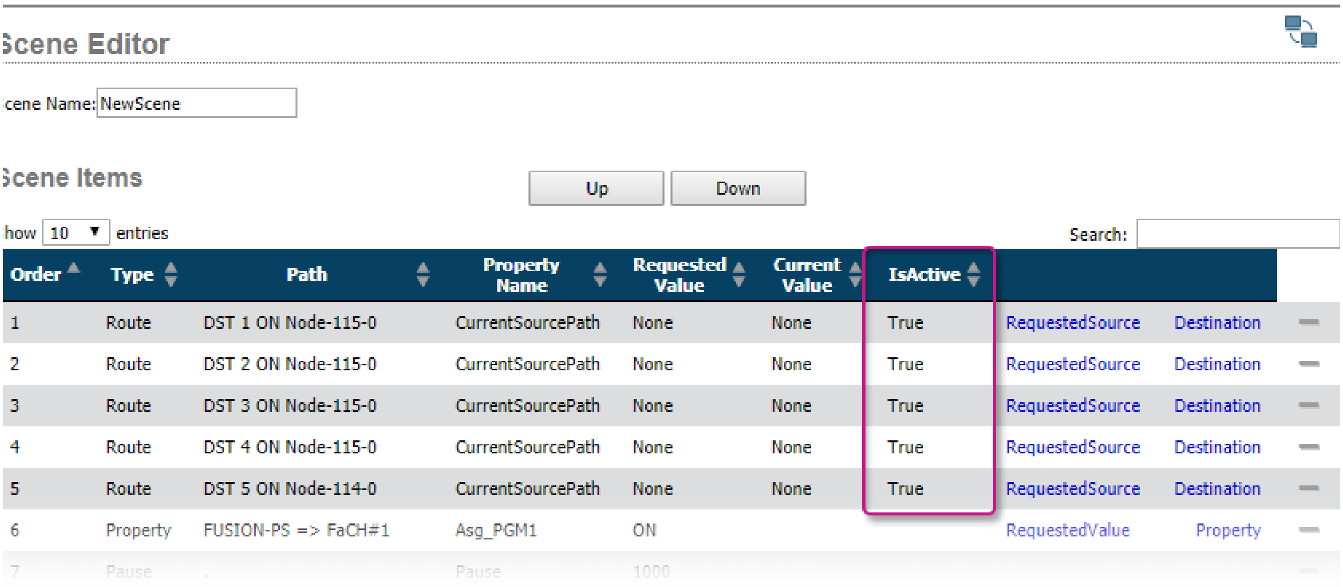
+

Showing 1 to 7 of 7 entries

Current Scene State and Troubleshooting

Once a Scene has been created, the list of Scenes will show whether the Scene is active or not in the **IsActive** field.

If you expect a Scene to be active and it is not, you can see what items failed to change by editing the Scene.



Scene Editor

cene Name: NewScene

Scene Items

how 10 ▾ entries

Search:

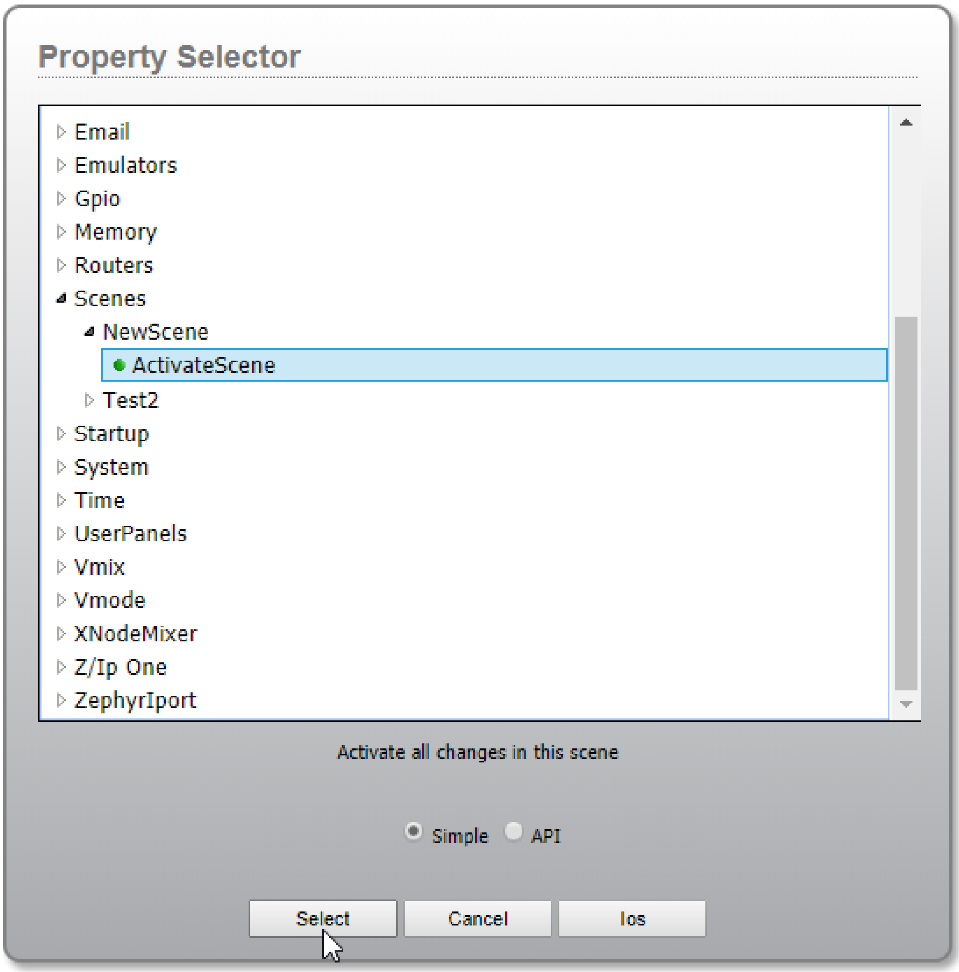
Order	Type	Path	Property Name	Requested Value	Current Value	IsActive			
1	Route	DST 1 ON Node-115-0	CurrentSourcePath	None	None	True	RequestedSource	Destination	—
2	Route	DST 2 ON Node-115-0	CurrentSourcePath	None	None	True	RequestedSource	Destination	—
3	Route	DST 3 ON Node-115-0	CurrentSourcePath	None	None	True	RequestedSource	Destination	—
4	Route	DST 4 ON Node-115-0	CurrentSourcePath	None	None	True	RequestedSource	Destination	—
5	Route	DST 5 ON Node-114-0	CurrentSourcePath	None	None	True	RequestedSource	Destination	—
6	Property	FUSION-PS => FaCH#1	Asg_PGM1	ON			RequestedValue	Property	—
7	Pause	.	Pause	1000					—

The *Requested Value* column will show the expected value for each Scene item for it to be considered active, while the *Current Value* column will show the current value of the Scene item.

> ⓘ Note: To refresh the *CurrentValue* data, you must exit and re-enter the Scene editor.

Scenes and Logic Flows

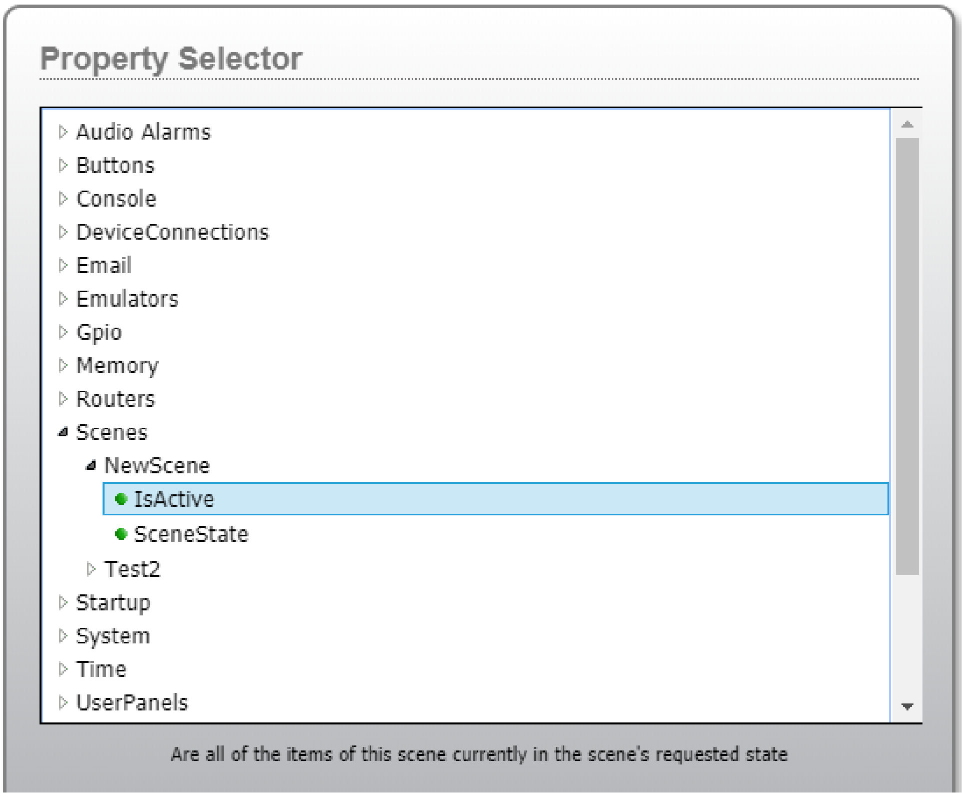
There are several properties that may be used with Scenes in Logic Flows. When an *End point* is selected each Scene has an **ActivateScene** property that can be set to **true** to cause all Scene items to be executed.



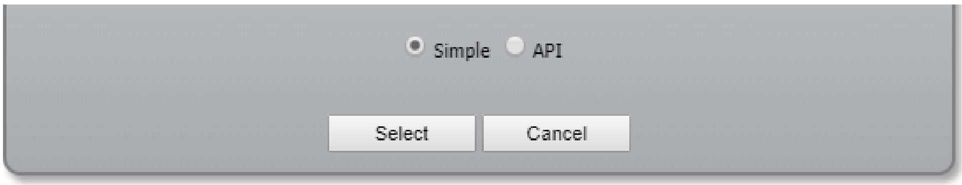
Activate all changes in this scene

● Simple ○ API

As a *Start point*, two properties are available: **IsActive** and **SceneState**.



Are all of the items of this scene currently in the scene's requested state

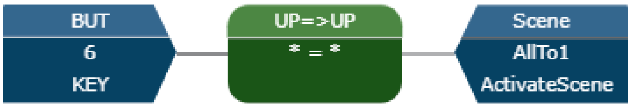


The *IsActive* property will either be **true** or **false** depending on whether all items in the Scene are currently at their requested value.

The *SceneState* property extends this slightly with three options: **All**, **Some**, or **None**. This will change depending on the Scene items' value, whether all items have their *RequestedValue*, some do, or none do.

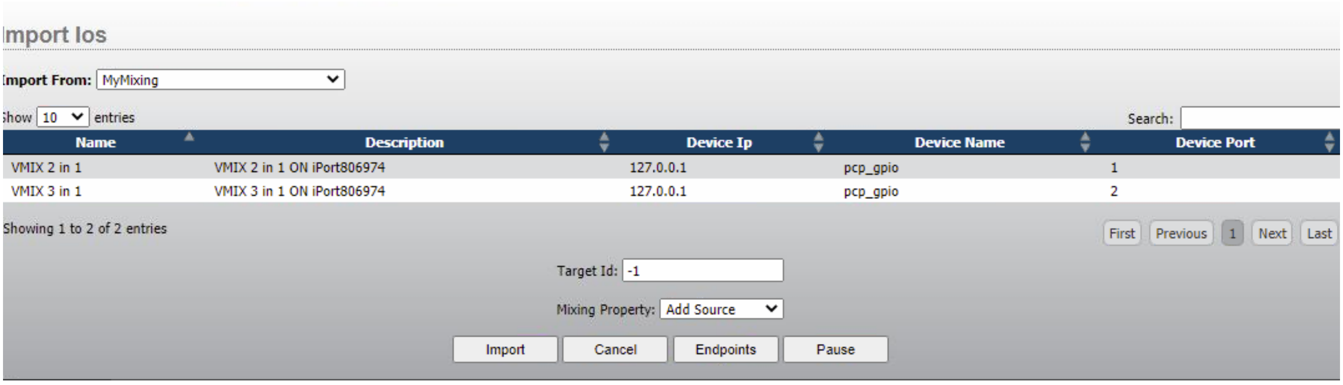
Activating Scene Changes

Scene changes may be activated using Logic Flows using the **ActivateScene** property. This property is write-only and is only available when editing End points. For example, we could create a Logic Flow that executes a Scene change every time a specific LCD button is pressed.



Virtual Mixing Routers and Scene Property Selection

Virtual mixing routers are inherently different in how they function from other routers. For details on Virtual Mixing Routers, see their section in the Routers section of the manual. With Virtual mixing destinations you do not change the source that is routed to the destination. Instead you add sources that will be mixed together by the mixing destination up to the supported quantity of the destination. As a result it uses different routing properties than a normal router. For example, AddSource adds a source to the mixing destination, and RemoveSource removes a source from the destination freeing that slot up for another source. Therefore when importing a destination to a scene, you need to be able to define whether the change will add or remove a source. To support this, a drop down will appear in the import destination list when creating a scene item.



Import Ios

Import From: MyMixing

Show 10 entries

Search:

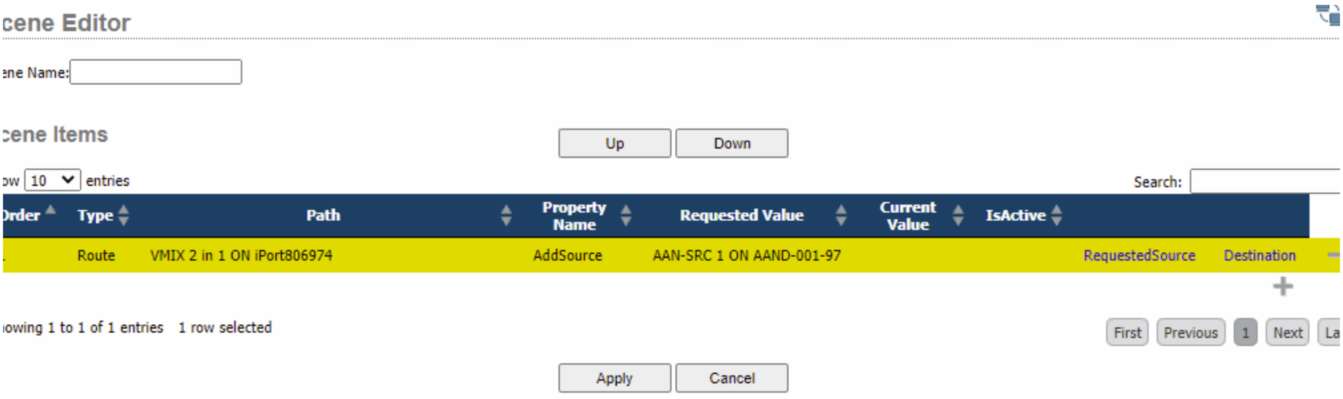
Name	Description	Device Ip	Device Name	Device Port
VMIX 2 in 1	VMIX 2 in 1 ON iPort806974	127.0.0.1	pcp_gpio	1
VMIX 3 in 1	VMIX 3 in 1 ON iPort806974	127.0.0.1	pcp_gpio	2

Showing 1 to 2 of 2 entries

Target Id: -1

Mixing Property: Add Source

The Mixing property drop down allows you to select whether the operation will add or remove a selected source to/from the destination. For example:


cene Editor

ene Name:

cene Items

Up Down

ow 10 entries

Search:

rder	Type	Path	Property Name	Requested Value	Current Value	IsActive		
	Route	VMIX 2 in 1 ON iPort806974	AddSource	AAN-SRC 1 ON AAND-001-97			RequestedSource	Destination

owing 1 to 1 of 1 entries 1 row selected

First Previous 1 Next La

Apply Cancel

In this example the scene will add Source 1 to the destination when the scene is executed.

The AddSource and RemoveSource properties of a Virtual Mixing Router destination are write only properties. You are applying a source to the property to change the underlying routing state, but the property itself will not have a value. Whenever you use write only properties in a scene, the state cannot be determined and therefore the IsActive property will never become true on such scenes. In a future version we may add some special code to handle the situation of virtual mixing routers such that the scene will detect if after the source is applied whether it exists on the destination and adjust the value of the scene line accordingly. But for now scenes will not become IsActive using these virtual mixing properties.

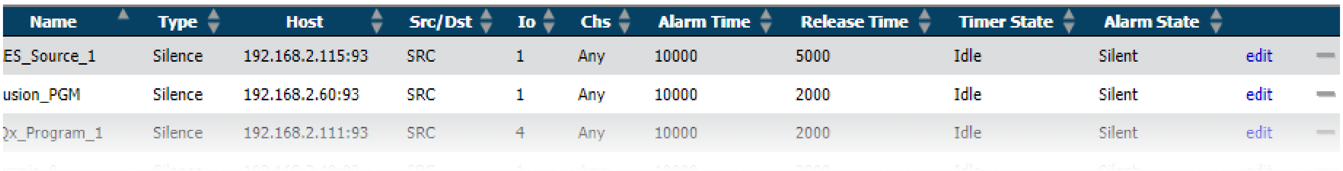
Audio Alarms

A very common use of Logic Flows in Pathfinder Core PRO is to perform critical actions when an audio source or destination falls silent. For example, you could have a Logic Flow send you an email if the feed to the transmitter becomes silent. In order to accomplish this, you should use the Audio Alarms section of Pathfinder Core PRO.

Almost any audio source or destination in an Axia network can be monitored for silence, clipping, and audio presence.

Click the **Audio Alarms** link in the Navigation Bar to create, edit, and view audio alarms in the system.

Like other Pathfinder Core PRO pages, the *Audio Alarms* page updates dynamically as the alarm states change. Nine columns are displayed for each audio alarm.



Name	Type	Host	Src/Dst	Io	Chs	Alarm Time	Release Time	Timer State	Alarm State	
ES_Source_1	Silence	192.168.2.115:93	SRC	1	Any	10000	5000	Idle	Silent	edit
usion_PGM	Silence	192.168.2.60:93	SRC	1	Any	10000	2000	Idle	Silent	edit
Qx_Program_1	Silence	192.168.2.111:93	SRC	4	Any	10000	2000	Idle	Silent	edit

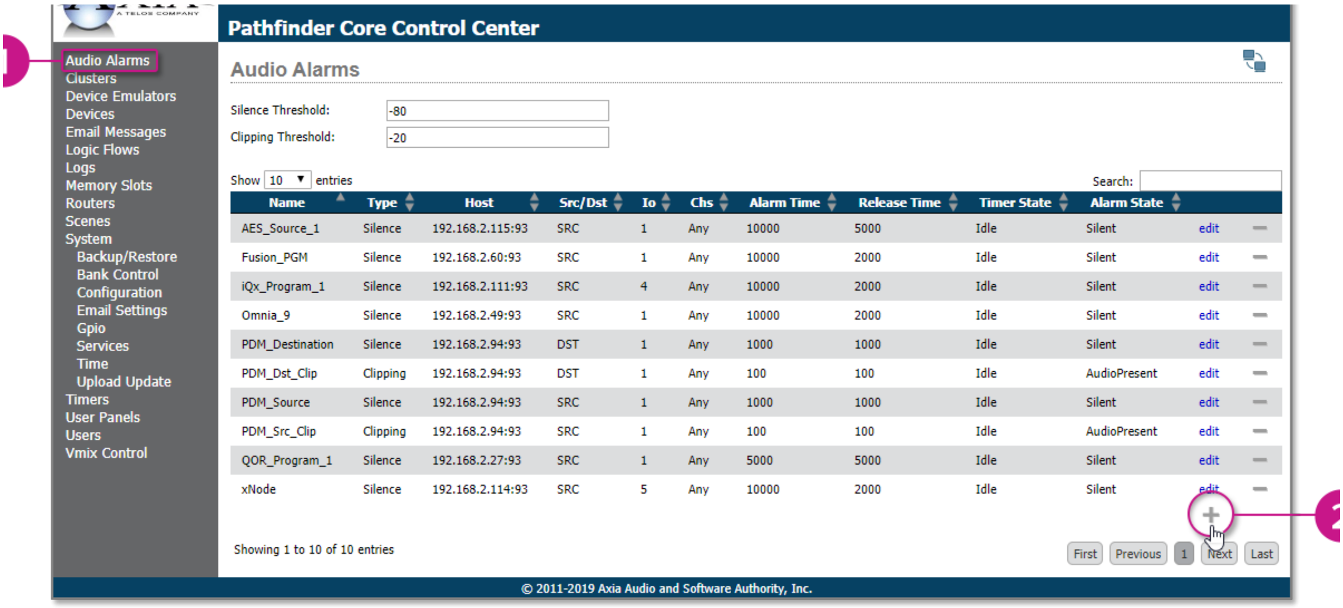
Column	Description
Name	The **name** of the alarm.

Type	The **type** of clearing alarm. Types can either be Silence or Clipping. If you wish to know when audio returns, use a silence alarm and then selec the correct value for the Alarm state in Logic Flow This will be discussed in greater detail below.
Host	This displays the **IP address** of the device where the source or destination being monitored exists.
SRC/DST	Displays whether the IO being monitored is a **source** or **destination**.
IO	Displays the **port number** on the device of the input or output being monitored.
Alarm Time	Displays the amount of **time in milliseconds** the selected source or destination must be silent (or clipping if the type is *clipping*) before the alarm is triggered.
Release Time	Displays the amount of **time in milliseconds** that the selected source or destination must be not silent (or not clipping if the type is *clipping*) before the alarm is released.
Timer State	This column shows the current **countdown** to Alarm Time or Release Time. For example, when a source being monitored goe silent, this column will show *CountingToAlarm*. *Id* means that the source or destination is currently ir the state represented by the AlarmState column.
Alarm State	This column represents the **current state** of the alarm. Possible values include *Silent*, *AudioPresent*, and *Clipping*. This value will only change after the source or destination maintains the requested state for at least the amount of time defined in the Alarm Time or Release Time properties.

Creating New Audio Alarms

Let's create an example.

Click the **Audio Alarms** link in the Navigation Bar to create, edit, and view audio alarms in the system. On the Audio Alarms page, click the **plus** icon to add an Audio Alarm.



Pathfinder Core Control Center

Audio Alarms

Silence Threshold: -80

Clipping Threshold: -20

Show 10 entries

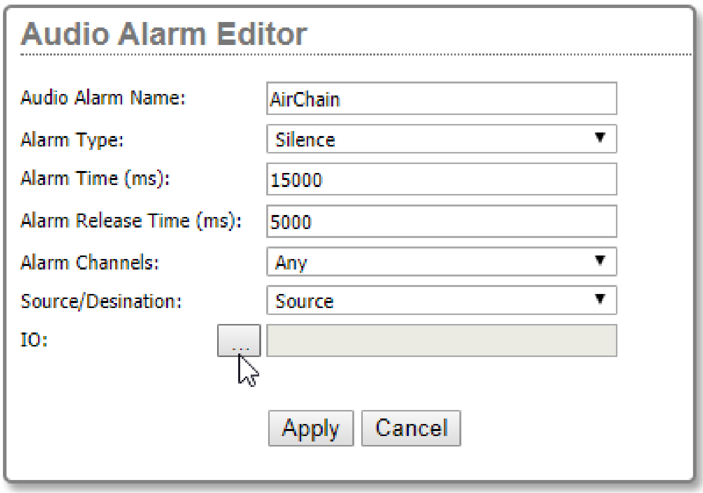
Search:

Name	Type	Host	Src/Dst	Io	Chs	Alarm Time	Release Time	Timer State	Alarm State		
AES_Source_1	Silence	192.168.2.115:93	SRC	1	Any	10000	5000	Idle	Silent	edit	—
Fusion_PGM	Silence	192.168.2.60:93	SRC	1	Any	10000	2000	Idle	Silent	edit	—
iQx_Program_1	Silence	192.168.2.111:93	SRC	4	Any	10000	2000	Idle	Silent	edit	—
Omnia_9	Silence	192.168.2.49:93	SRC	1	Any	10000	2000	Idle	Silent	edit	—
PDM_Destination	Silence	192.168.2.94:93	DST	1	Any	1000	1000	Idle	Silent	edit	—
PDM_Dst_Clip	Clipping	192.168.2.94:93	DST	1	Any	100	100	Idle	AudioPresent	edit	—
PDM_Source	Silence	192.168.2.94:93	SRC	1	Any	1000	1000	Idle	Silent	edit	—
PDM_Src_Clip	Clipping	192.168.2.94:93	SRC	1	Any	100	100	Idle	AudioPresent	edit	—
QOR_Program_1	Silence	192.168.2.27:93	SRC	1	Any	5000	5000	Idle	Silent	edit	—
xNode	Silence	192.168.2.114:93	SRC	5	Any	10000	2000	Idle	Silent	edit	—

Showing 1 to 10 of 10 entries

© 2011-2019 Axia Audio and Software Authority, Inc.

Complete the fields on the Audio Alarm Editor.



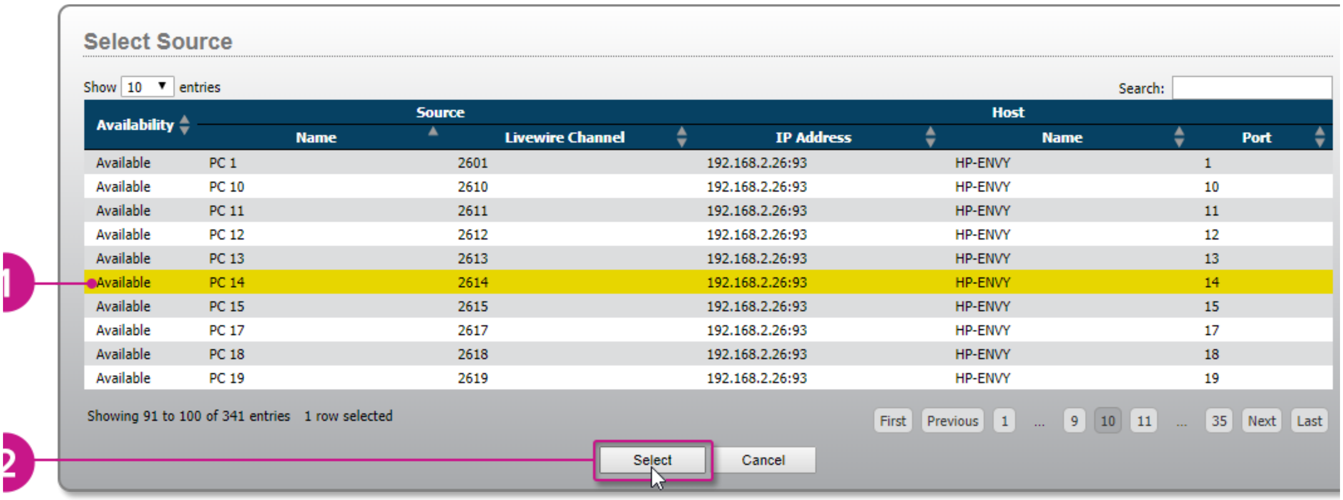
Audio Alarm Editor

Audio Alarm Name:	AirChain
Alarm Type:	Silence
Alarm Time (ms):	15000
Alarm Release Time (ms):	5000
Alarm Channels:	Any
Source/Desination:	Source
IO:	

Field	Notes
Audio Alarm Name	Type a **name** for this Alarm. In the example above *AirChain*.
Alarm Type	Select the **alarm type** from the drop-down list. Options include *Silence* and *Clipping*.
Alarm Time (ms)	**Alarm time** is the number of milliseconds the aud must be in the specified state before the alarm becomes active. The example above defines an alarm time of 15000ms or 1.5-seconds.
Alarm Release Time	**Alarm release time** is the amount of time the aud needs to be in the opposing state after an alarm has tripped before the alarm clears. The example above defines an alarm release time of 5000ms, c .5-seconds.

Alarm Channels	From the drop-down list, specify the **channel** the system should monitor to determine the alarm stat Options include *Any*, *All*, *Left*, or *Right*.
Source/Destination	Select whether the alarm will be on a **Source** or **Destination**.
IO	Click the **ellipsis button** to choose from the list of available sources or destinations.

After clicking on the ellipsis button to open the Select IO screen (either Select Source or Select Destination, depending on how this alarm is configured), select the **source** or **destination** from the IO list and click **Select**.



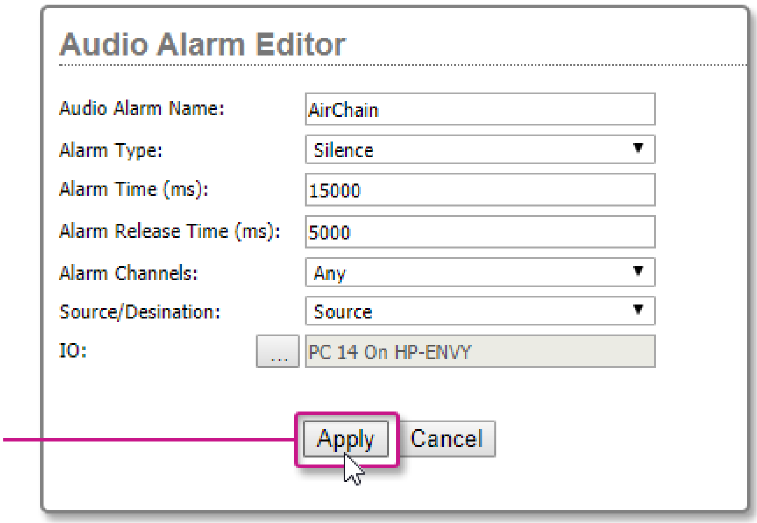
Select Source

Show 10 entries Search:

	Source		Host		
Availability	Name	Livewire Channel	IP Address	Name	Port
Available	PC 1	2601	192.168.2.26:93	HP-ENVY	1
Available	PC 10	2610	192.168.2.26:93	HP-ENVY	10
Available	PC 11	2611	192.168.2.26:93	HP-ENVY	11
Available	PC 12	2612	192.168.2.26:93	HP-ENVY	12
Available	PC 13	2613	192.168.2.26:93	HP-ENVY	13
Available	PC 14	2614	192.168.2.26:93	HP-ENVY	14
Available	PC 15	2615	192.168.2.26:93	HP-ENVY	15
Available	PC 17	2617	192.168.2.26:93	HP-ENVY	17
Available	PC 18	2618	192.168.2.26:93	HP-ENVY	18
Available	PC 19	2619	192.168.2.26:93	HP-ENVY	19

Showing 91 to 100 of 341 entries 1 row selected

Click **Apply** to save your changes to this alarm.



Audio Alarm Editor

Audio Alarm Name: AirChain

Alarm Type: Silence

Alarm Time (ms): 15000

Alarm Release Time (ms): 5000

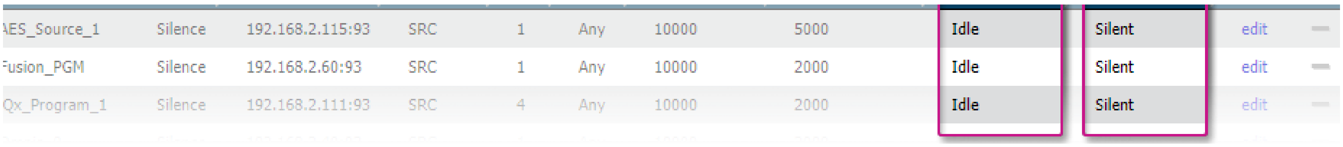
Alarm Channels: Any

Source/Desination: Source

IO: PC 14 On HP-ENVY

Monitoring Alarms from the Audio Alarms List

The Audio Alarms list view, like most lists in Pathfinder Core PRO, show alarm status changes in real time.

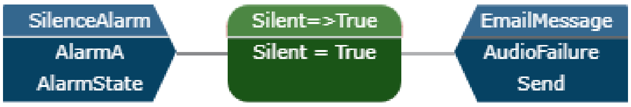


AES_Source_1	Silence	192.168.2.115:93	SRC	1	Any	10000	5000	Idle	Silent	edit	—
Fusion_PGM	Silence	192.168.2.60:93	SRC	1	Any	10000	2000	Idle	Silent	edit	—
Qx_Program_1	Silence	192.168.2.111:93	SRC	4	Any	10000	2000	Idle	Silent	edit	—

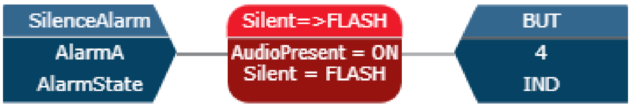
For example, the *Timer State* and *Alarm State* values will dynamically update in this list as alarm states change. This can be an excellent troubleshooting tool.

Using Alarms in Logic Flows

Once an alarm has been created, it is available for use in Logic Flows. In the Logic Flow below, we have created a Logic Flow that sends an email whenever the *SilenceAlarm* audio alarm has its *Alarm State* switch to **Silent**:



If we wanted to perform a different action when the alarm releases (audio returns), we would use the *AudioPresent* value in the translation list. For example, the following Logic Flow will light a button when audio is present and cause it to flash when there is silence.



> ⓘ Important Note: The thresholds used for silence detection and clipping are fixed in the code and cannot be changed. The silence threshold is set for -80db. The clipping threshold is set for -1db.

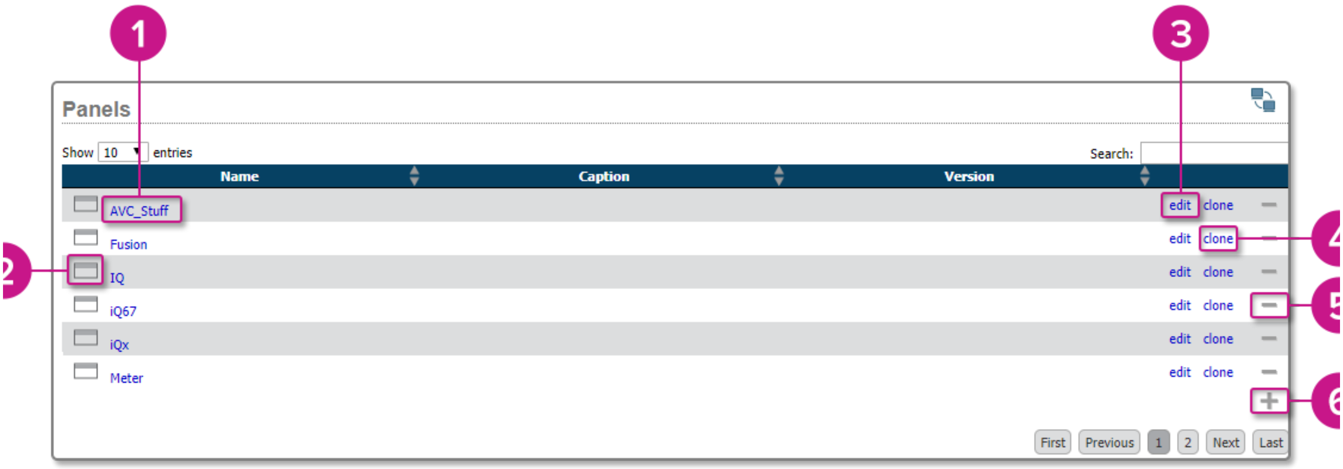
More possible audio alarm Logic Flows examples are covered in the *Email Messages* section since that is a common action to be taken when things get quiet.

User Panels

HTML5 User Panels

Pathfinder Core PRO allows you to create your own custom user interface that can be displayed and used in a web browser on any device. This allows you to define precisely how your users interact with your Axia system in a way that supports your organization’s workflow. Currently, Google Chrome is the recommended browser for this feature.

HTML User Panels may be used and edited from the **User Panels** menu item. The *Panels* list includes HTML5 Panels and legacy Panels created with the legacy Panel designer.



Panels

Show 10 entries Search:

Name	Caption	Version			
AVC_Stuff			edit	clone	—
Fusion			edit	clone	—
IQ			edit	clone	—
iQ67			edit	clone	—
iQx			edit	clone	—
Meter			edit	clone	—

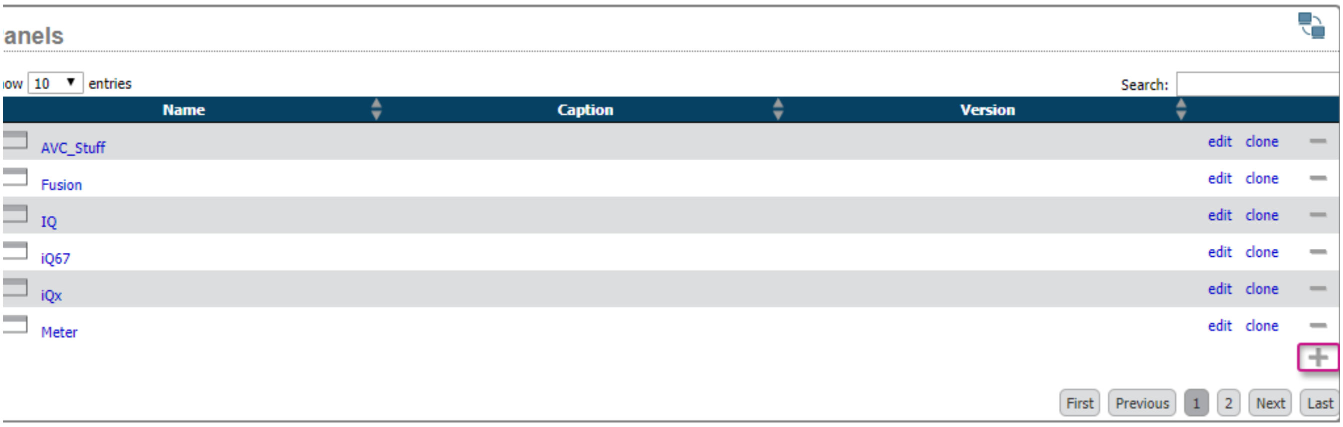
+

First Previous 1 2 Next Last

Reference	Description
1	HTML5 Panels are displayed as links. Clicking th **link** will display the Panel within the context of the Pathfinder Core PRO web pages.
2	Click this icon to open the Panel in its own windo without most of the browser menu systems.
3	Click the **Edit** link to open the Panel in the HTML Panel designer
4	Clicking the **Clone** link will make a duplicate of th Panel as it exists under a new Panel name.
5	Click the **minus icon** to delete a Panel
6	Click the **plus icon** to open the HTML Panel designer and create a new Panel.

Creating a New Panel

- Click the **plus icon** to create a new Panel and open the new Panel in the Panel Designer.



anels

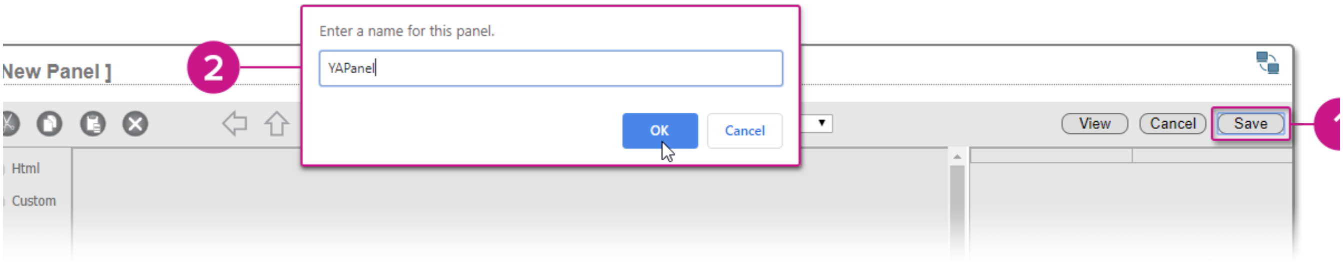
ow 10 entries Search:

Name	Caption	Version			
AVC_Stuff			edit	clone	—
Fusion			edit	clone	—
IQ			edit	clone	—
iQ67			edit	clone	—
iQx			edit	clone	—
Meter			edit	clone	—

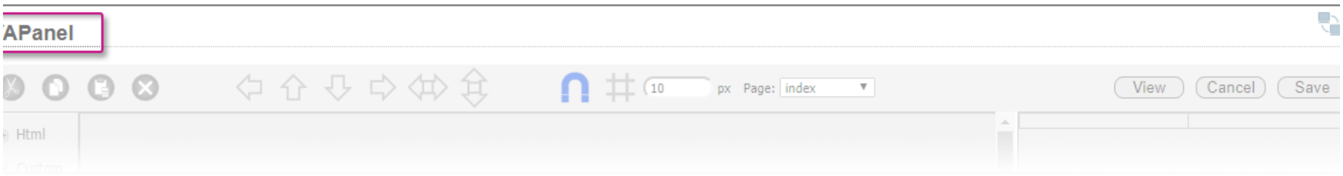
+

First Previous 1 2 Next Last

- Click **Save** and type a **name** for your Panel to instantiate it before you begin adding components.

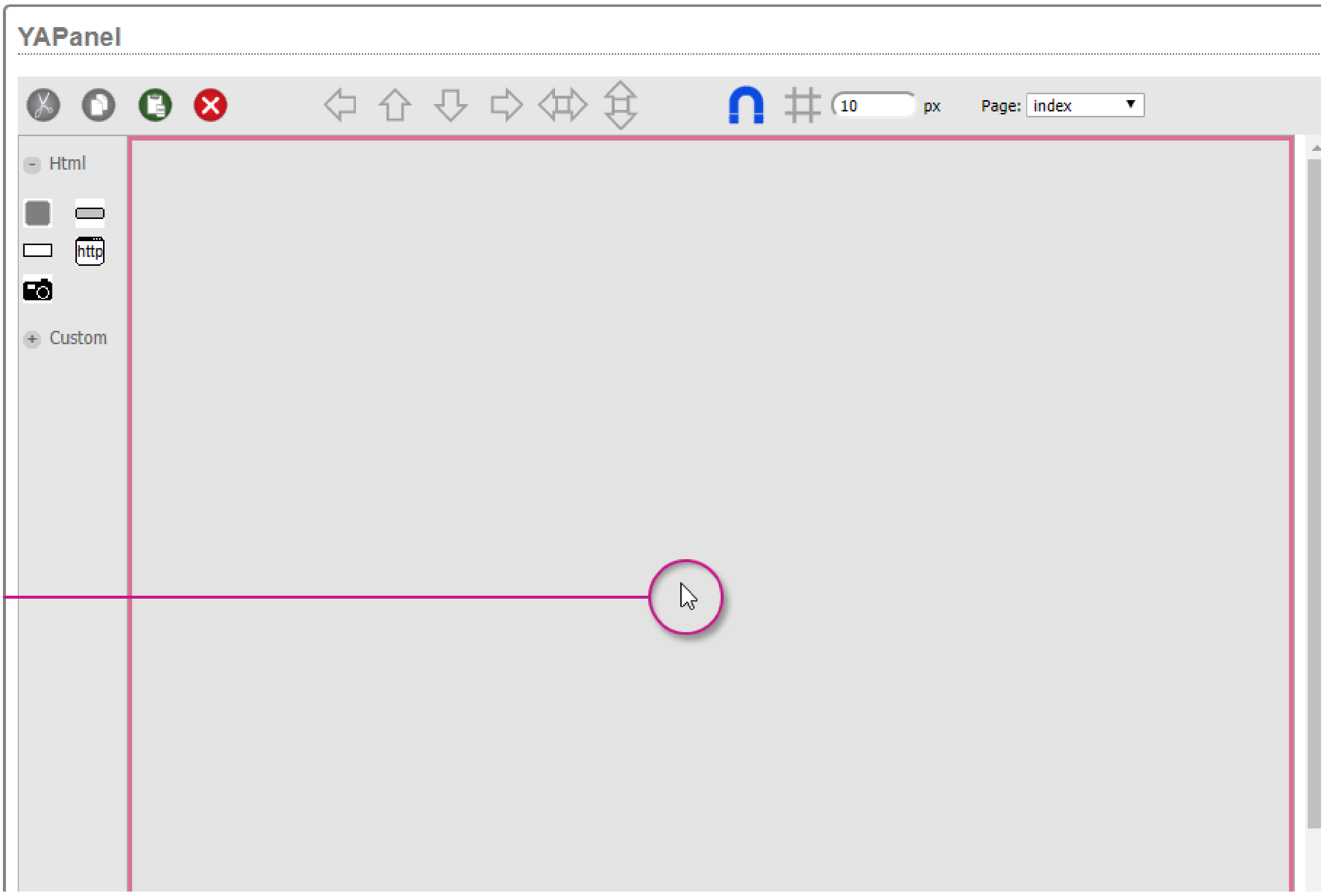


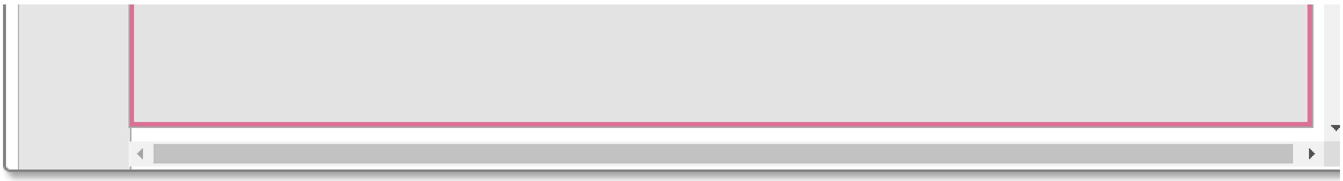
- After saving and giving the Panel a name, the name of the Panel should appear in the upper corner:



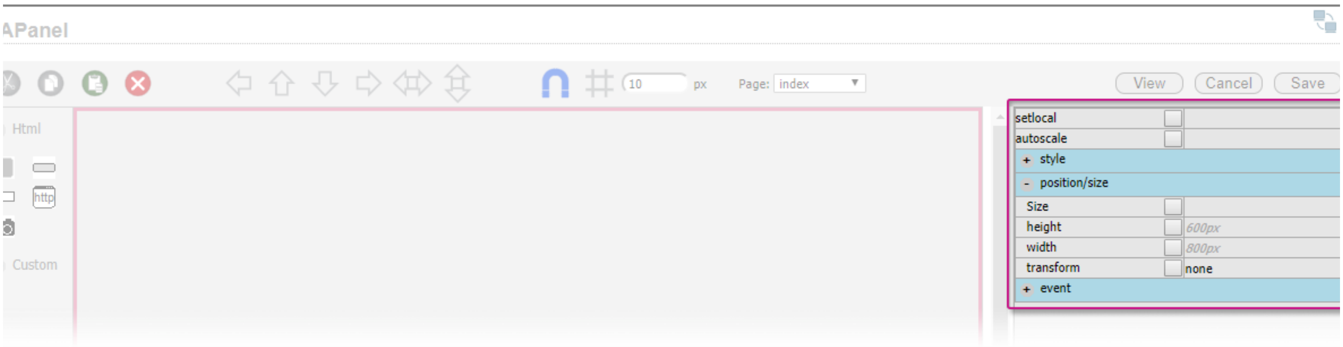
Selecting a Theme

User panel components in their default state will look differently depending on the theme selected for the panel. Currently, there are two themes available: Default and Dark. To select a theme, click the **main Panel** to select it:



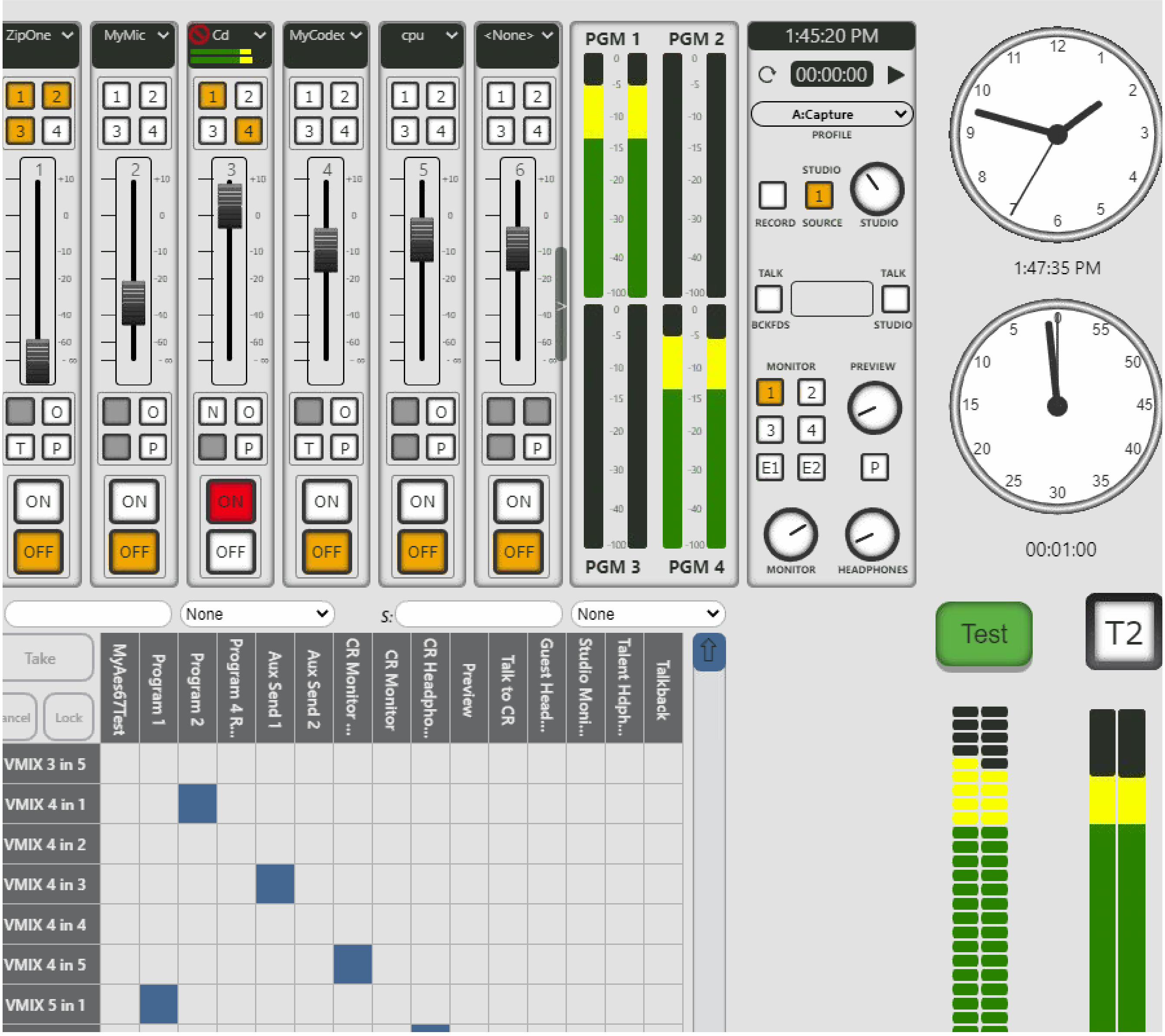


The property window on the right side of the Panel designer will display properties that can be modified for the selected component:

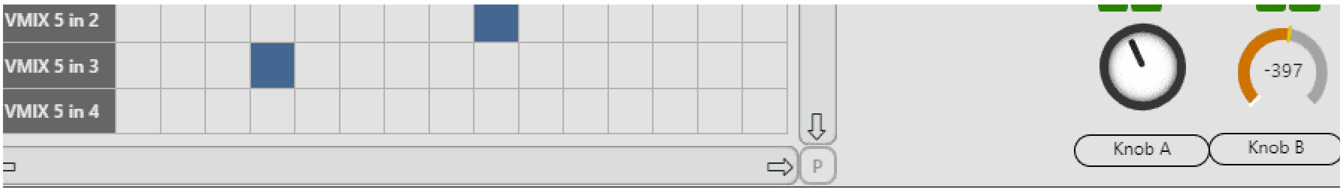


setlocal	□	
autoscale	□	
+ style		
- position/size		
Size	□	
height	□	600px
width	□	800px
transform	□	none
+ event		

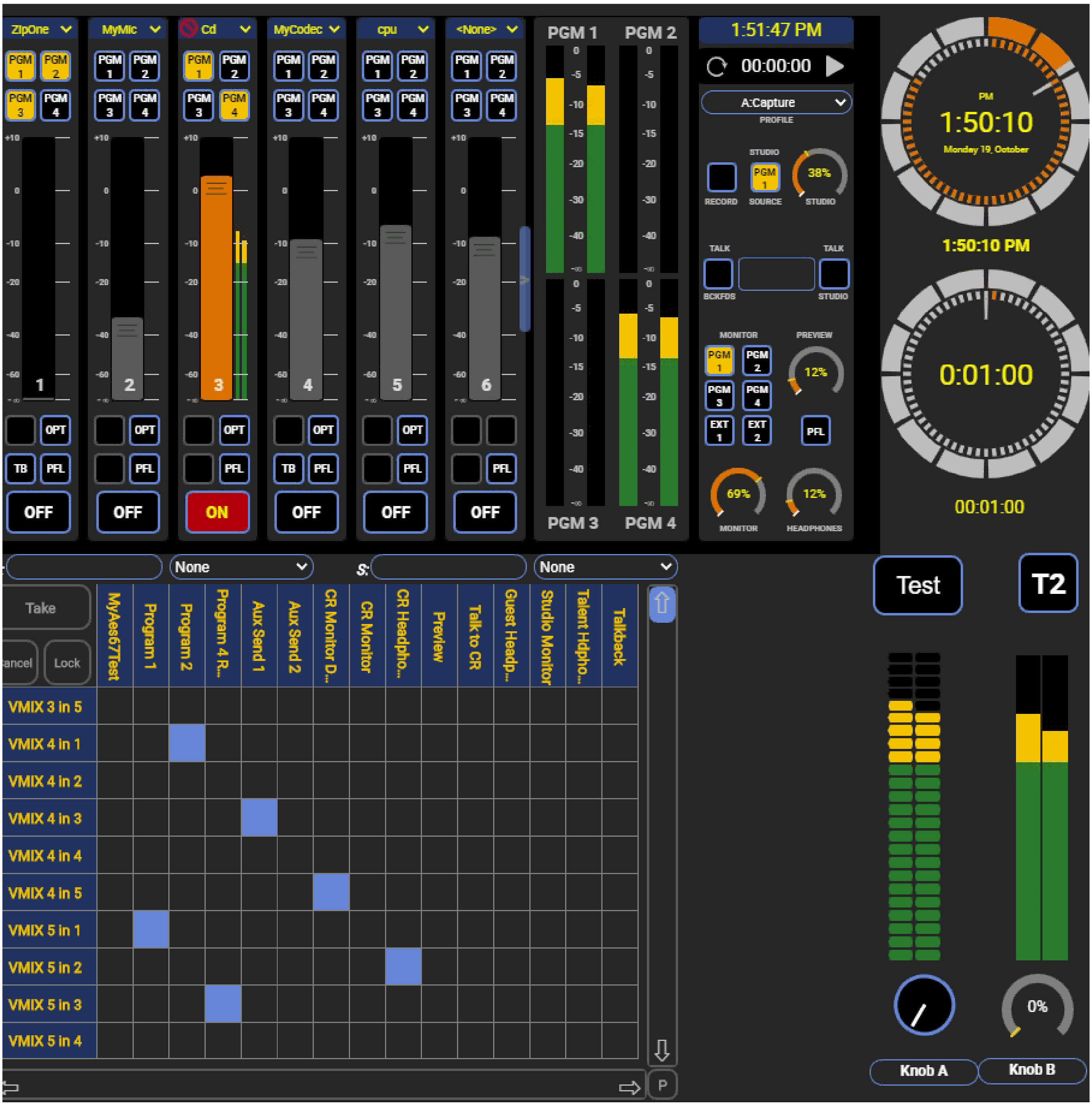
Expand the style selection and use the theme drop-down to select default or dark, then click Save. To get an idea of the difference between the two theme styles, see the two example panel pictures below:



	MyAes67Test	Program 1	Program 2	Program 4 R...	Aux Send 1	Aux Send 2	CR Monitor ...	CR Monitor	CR Headpho...	Preview	Talk to CR	Guest Head...	Studio Moni...	Talent Hdph...	Talkback
VMIX 3 in 5															
VMIX 4 in 1			■												
VMIX 4 in 2															
VMIX 4 in 3					■										
VMIX 4 in 4															
VMIX 4 in 5							■								
VMIX 5 in 1		■													



Default Theme

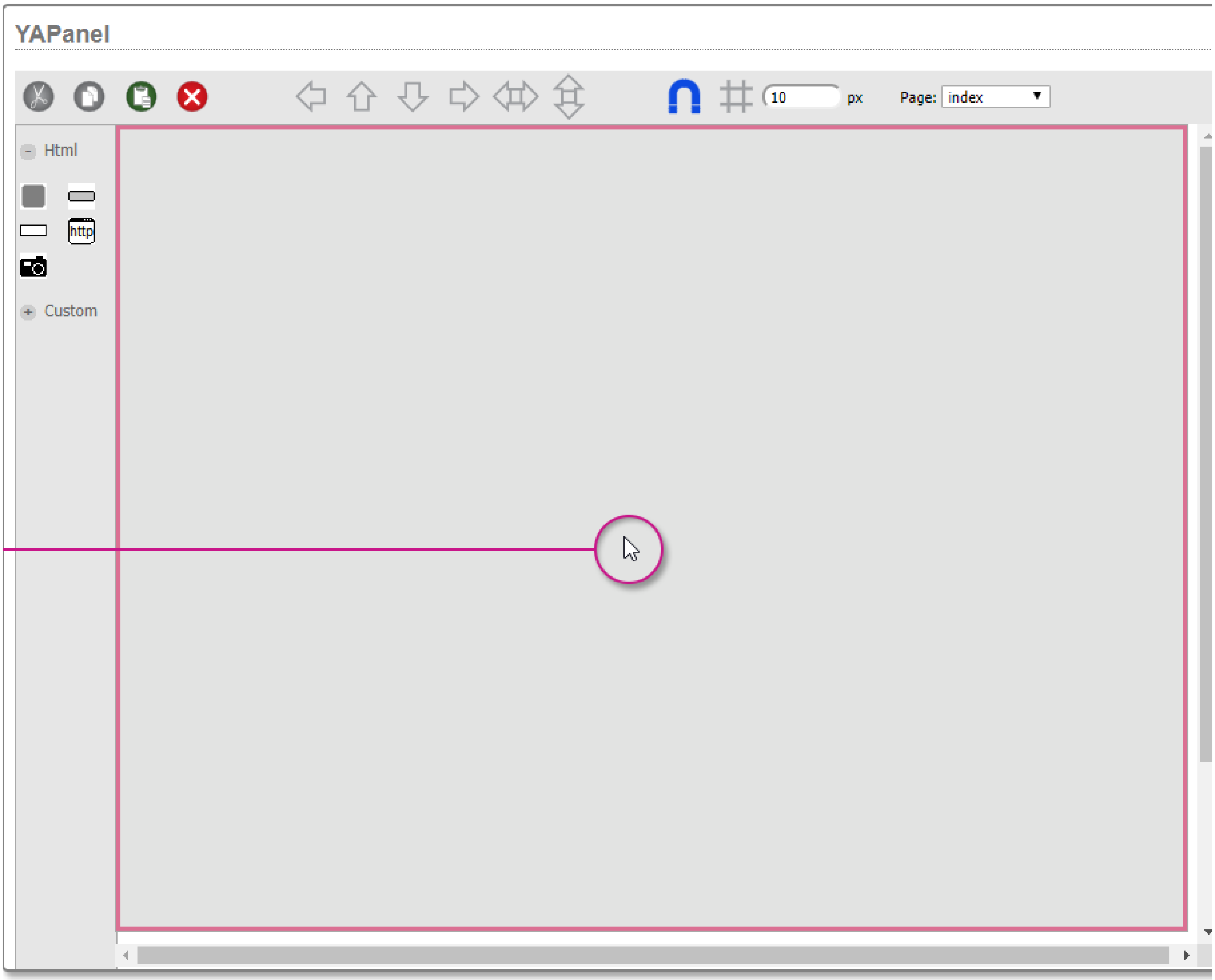


Dark Theme

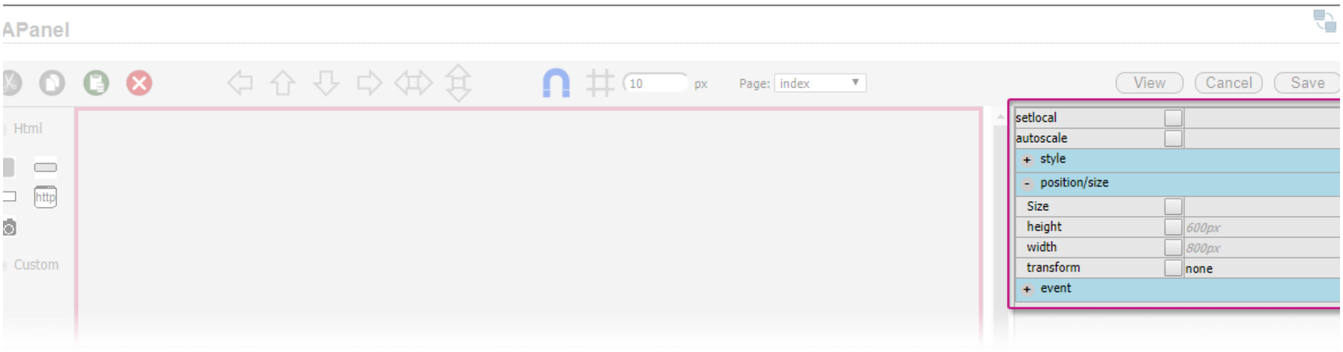
It is important to note that when changing between themes it is sometimes necessary to save the panel in the new theme before the style will be correctly displayed. Additionally, existing panels may require some adjustment when changing to the new theme. In particular, styles which you have overridden with you own settings will retain your choices which may or may not be appropriate to the new style. If you prefer a certain style to be the default theme for all new panels, see Theme Advanced Options. In the Component Reference we may display either dark or default theme, but the same components will work with either and will be styled accordingly.

Setting Panel Size

Next set the size of the Panel. Click the **main Panel** to select it:



The property window on the right side of the Panel designer will display properties that can be modified for the selected component:

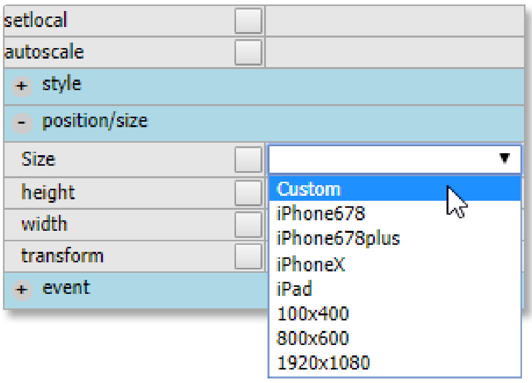


setlocal	□	
autoscale	□	
+ style		
- position/size		
Size	□	
height	□	600px
width	□	800px
transform	□	none
+ event		

Panel dimensions can be set to a **standard preset**, set to a specific **height or width**, or **autoscaled** dynamically.

Selecting a Size Preset

With the main Panel selected, click in the **Property** window's **Size** field and select from a size preset.



setlocal

autoscale

+ style

- position/size

Size

height

width

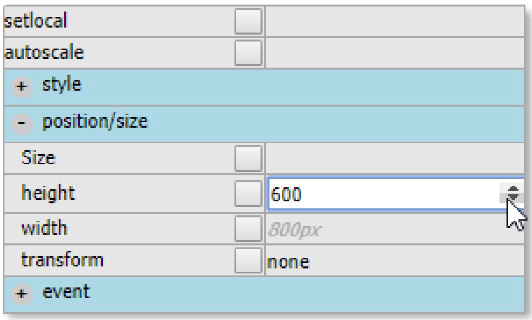
transform

+ event

The size of the Panel area in the Panel Designer should change based on your selection.

Setting the Panel Height or Width

With the main Panel selected, click in the **Property** window's **Height** or **Width** field and set a new height and width in pixels. Pixel values may be typed in, or you may click on the spinner control to adjust the value.



setlocal

autoscale

+ style

- position/size

Size

height 600

width 800px

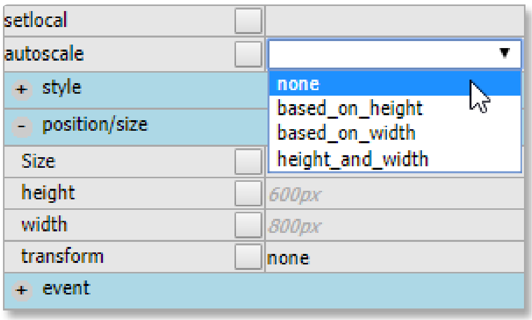
transform none

+ event

The size of the Panel area in the Panel Designer should change based on your changes.

Autoscaling

Use autoscaling if you want the components in the panel to scale to the size of the browser window as it changes. With the main Panel selected, select an option from the **autoscale** dropdown.



setlocal

autoscale

+ style

- position/size

Size

height 600px

width 800px

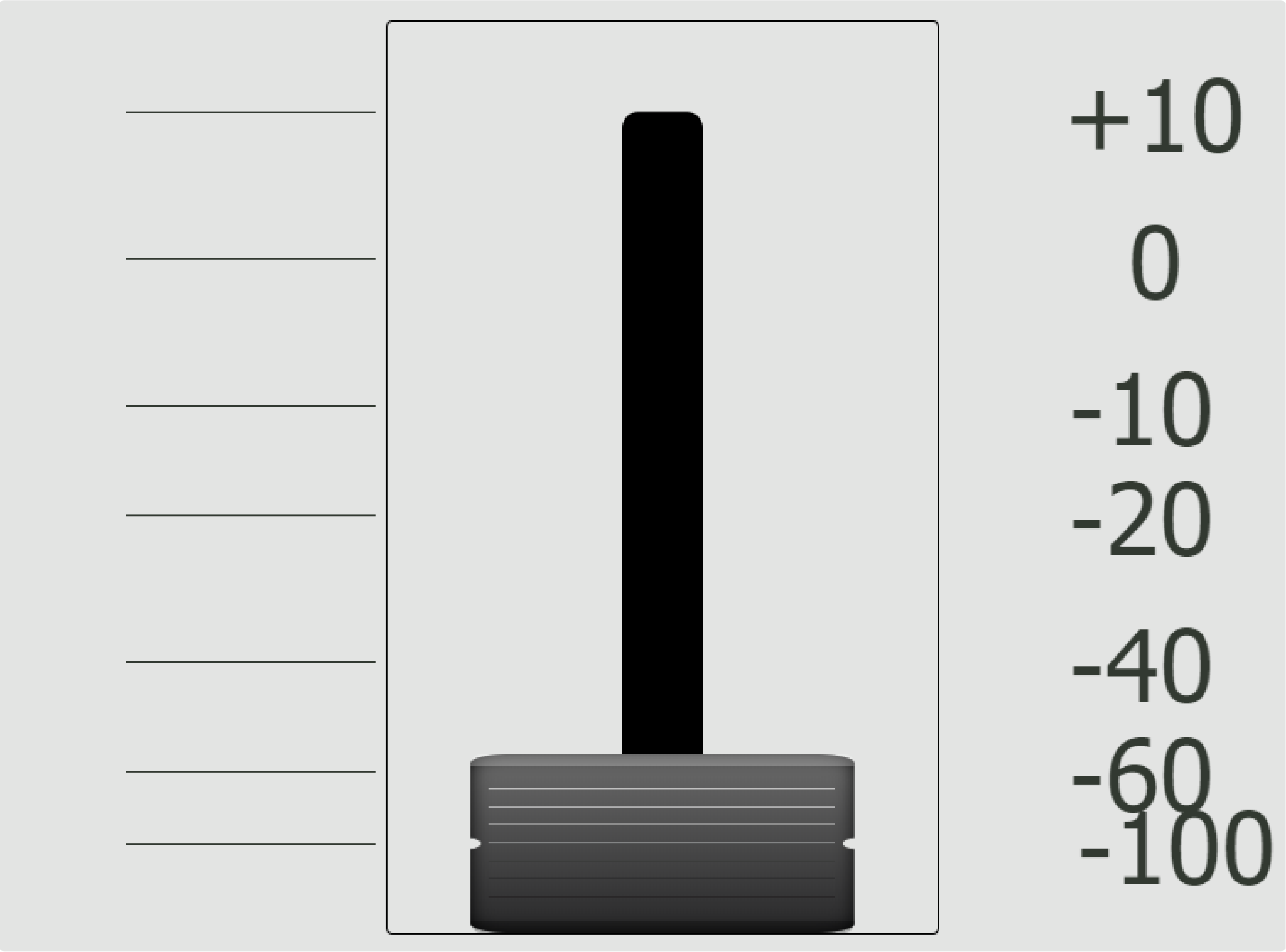
transform none

+ event

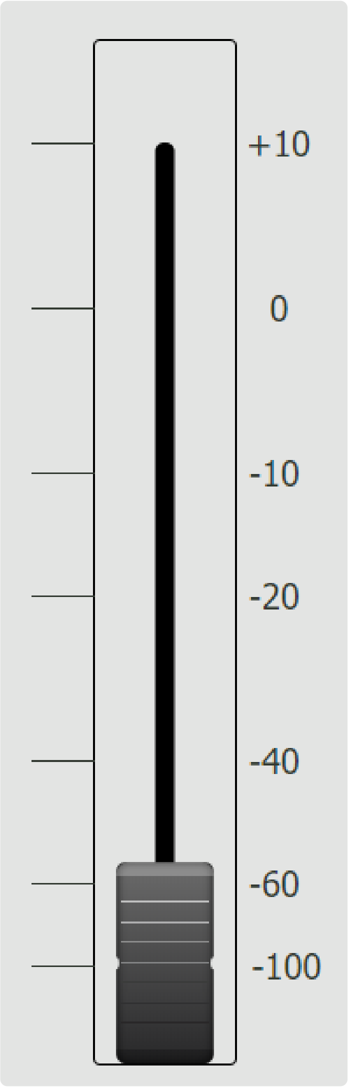
Auto Scale Setting	Description
	No value or selecting a value of **none** will disable

None	Autoscale functionality, allowing the component size and placement to remain fixed regardless of the browser window size.
based_on_height	As the user changes their browser's **height**, the amount of change will be used as the factor for scaling up the height and width of the component Changing the browser width will do nothing.
based_on_width	As the user changes their browser's **width**, the amount of change will be used as the factor for scaling up the height and width of the component Changing the browser height will do nothing.
height_and_width	As the user changes their browser's **height** or **width**, the amount of change in height will be use as the factor for scaling the height of the components, the amount of change in width will b used as the factor for scaling the width of the components.

While intuitively you might believe that **height_and_width** is the correct choice, it rarely is. This option can cause buttons and components to be stretched in strange ways.



On the other hand, scaling based on **height** alone will maintain the correct aspect of the component while just scaling it up and down:



In general, if the aspect ratio of the Panel objects is important and the Panel will be displayed on a widescreen monitor, it is best to choose scaling based on *height*. If it is a widescreen monitor turned sideways for portrait orientation, the best option is based on *width*.

If aspect ratio does not matter, for example when you are working only with square or rectangular buttons, autoscaling based on *height and width* may be used.

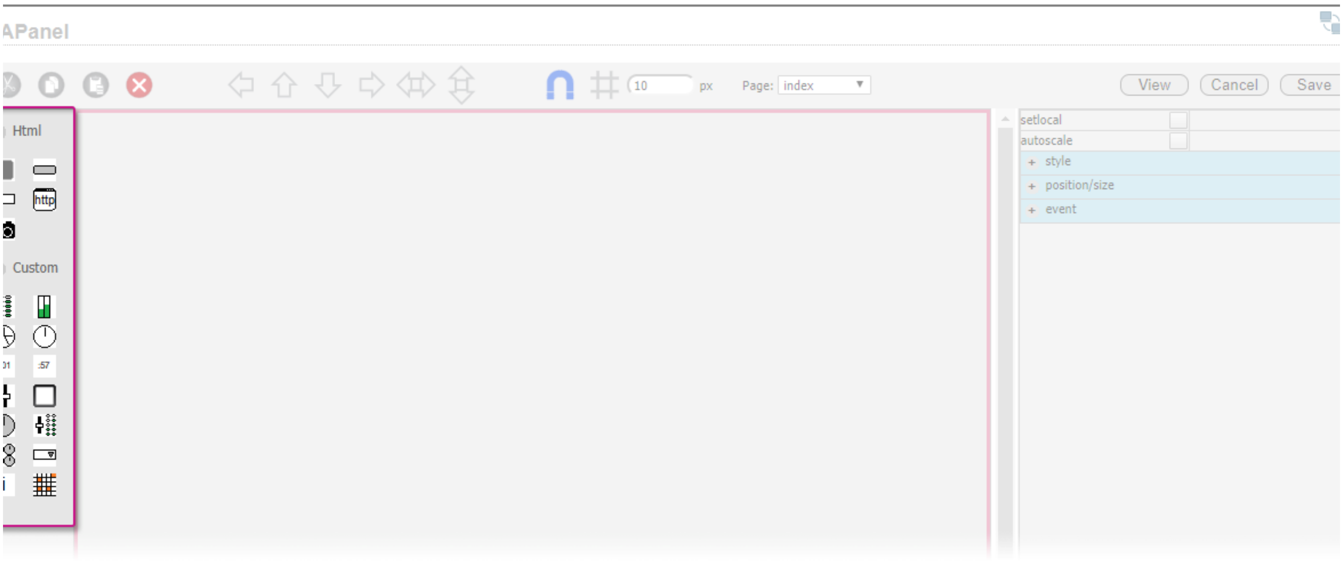
Testing Autoscaling Options

After creating your Panel, save it with one of the autoscaling options. Open the Panel in a browser and change the browser size to observe the autoscaling behavior and get a feel for which option is most appropriate.

As you evaluate the different autoscaling options, after resaving with a new option you must refresh your browser for the new option to be applied.

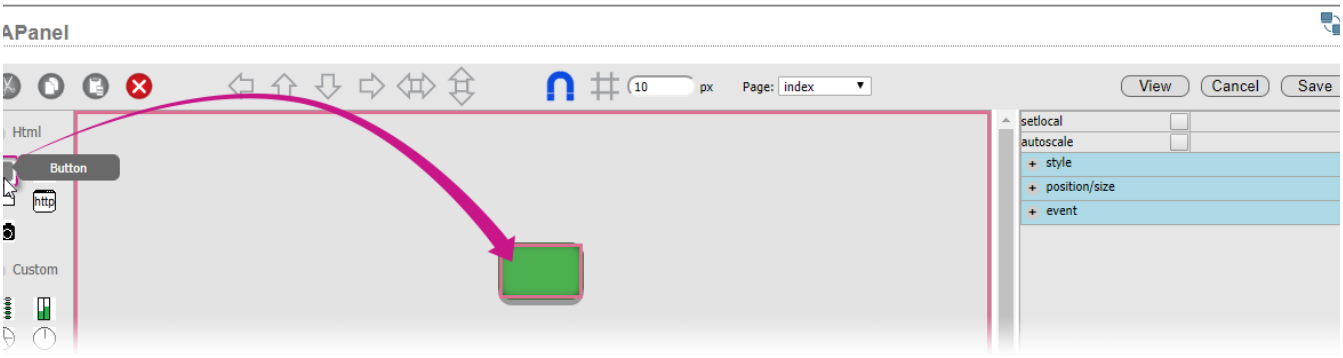
Adding Components

To add components to your Panel, expand the HTML and/or Custom sections in the left toolbar:



Hovering over any component will show a tooltip with the name of that component. We can illustrate adding and modifying components using a simple HTML button.

Click and drag the top-left HTML **Button** (the tooltip will say *Button*) from the toolbar into the Panel. After dropping the button in your Panel, it should be highlighted with a red box indicating it is the selected component.



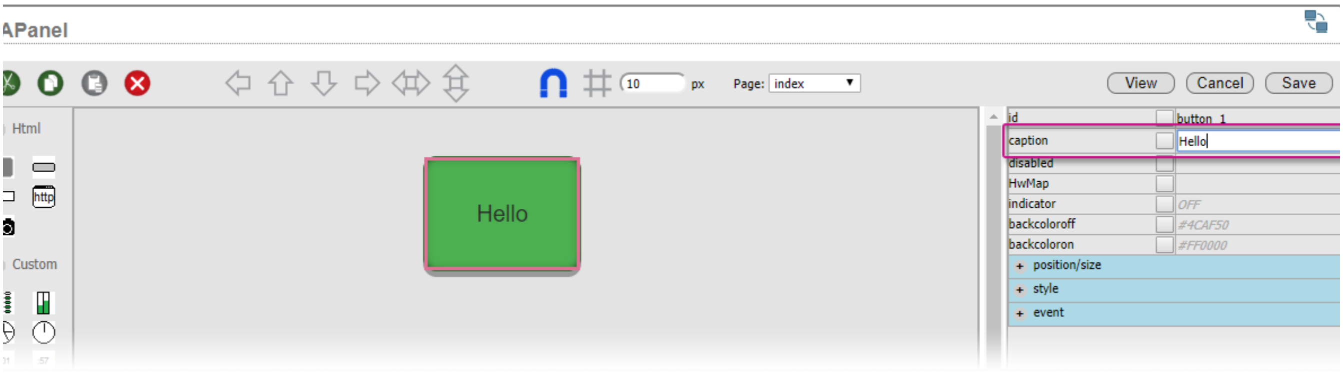
To resize the component, click and drag the component's red border. The border displays just inside the actual edges of the object by design so when aligning objects you can still see the object's actual edges.

To move the component within the Pane, click and drag the center of the component. You can also use the arrow keys on your keyboard to nudge the selected component in any direction. Each press of an arrow key will move the component one pixel at a time unless the grid is enabled in which case it will move by the grid amount. See the *Grid* topic in the *Tool Bar* section below for additional details.

> ⓘ **Important Note:** By default, some components may resize both height and width when one or the other is dragged as non-scaled resizing causes the component to look skewed and stretched if the aspect ratio is not maintained. These are generally more complex components such as the console fader. Holding the SHIFT key while resizing overrides this behavior and allows you to skew the component if desired.

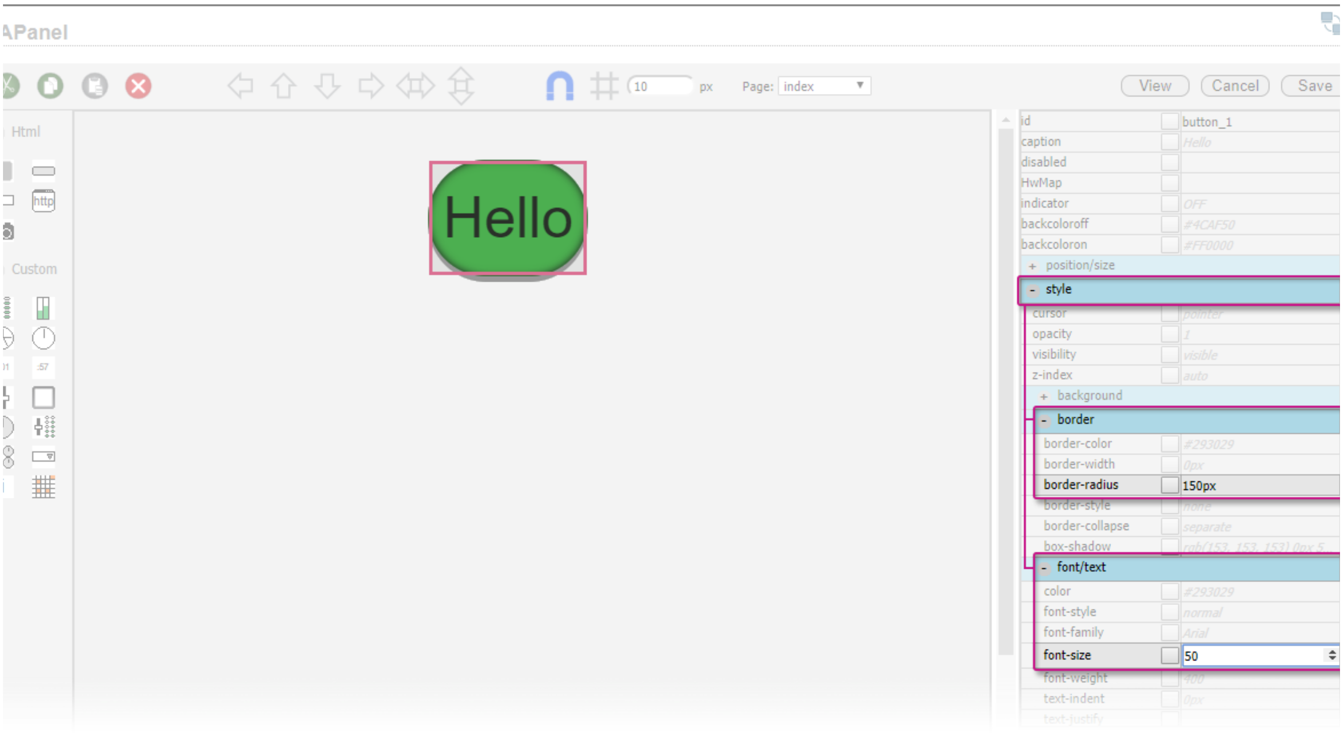
Adjusting Properties

The property list on the right will update to include properties that may be adjusted for the currently selected component type. Clicking in a field and changing the property will make the corresponding change to the component. For example, with our button still selected, click in the *Caption* field and type a **caption**:



id: button_1
caption: Hello
disabled
HwMap
indicator: OFF
backcoloroff: #4CAF50
backcoloron: #FF0000
+ position/size
+ style
+ event

Expand the *Style* section and try adjusting the **border>border-radius** value or the **font/text>font-size** value.



id: button_1
caption: Hello
disabled
HwMap
indicator: OFF
backcoloroff: #4CAF50
backcoloron: #FF0000
+ position/size
- style
cursor: pointer
opacity: 1
visibility: visible
z-index: auto
+ background
- border
border-color: #293029
border-width: 0px
border-radius: 150px
border-style: none
border-collapse: separate
box-shadow: rgb(153, 153, 153) 0px 5
- font/text
color: #293029
font-style: normal
font-family: Arial
font-size: 50
font-weight: 400
text-indent: 0px
text-justify

As you can see there is a high degree of power to achieve exactly the desired design using the properties in the property grid.

Once you’ve achieved your design objectives, click **Save** to save your changes or click **Cancel** to return the Panel to its previous save state.

> ⓘ Important Note: There is no **undo** or **redo feature,** so saving frequently as you work is recommended.

SHIFT Editing the Property List

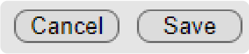
Holding the SHIFT key while clicking in the User Panel property editor fields will bypass the usual helper dialogs and allow direct editing of the text. This can be useful for things like copying and pasting color values. **SHIFT+click** to highlight the text box without the helper dialog and then click again to edit the text for the property.

Tool Bar

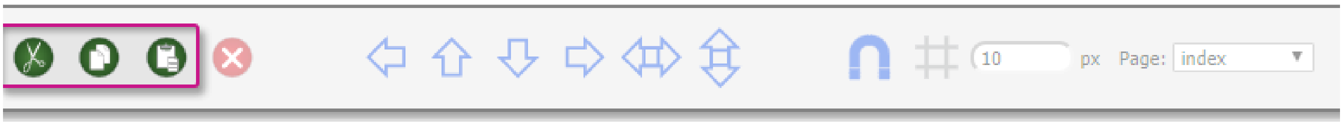
The top Tool Bar has several tools to help with the component layout:

Cancel/Save

The *Cancel* and *Save* buttons may be used to save your work or cancel pending changes and reload from the last save point. Frequent saves while working on the design are recommended.



Cut, Copy, Paste

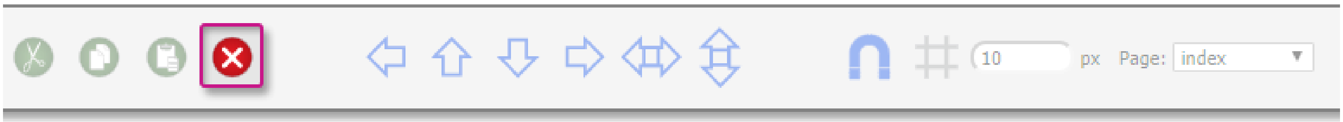


Page: index

These are the standard *Cut*, *Copy*, and *Paste* tools. These actions can be performed on any currently selected components.

You can select multiple components by holding the SHIFT key while you click additional components.

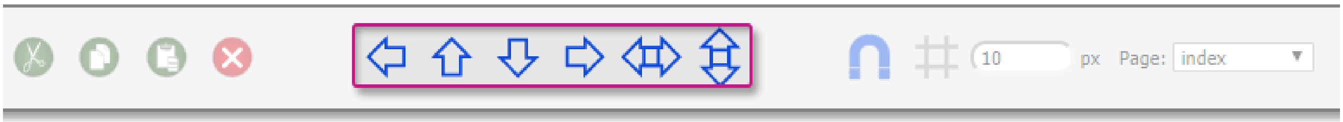
Delete



Page: index

Clicking the *Delete* button will delete the currently select components. Be careful with this function as there is no undo though you can click cancel to return to the previously saved state.

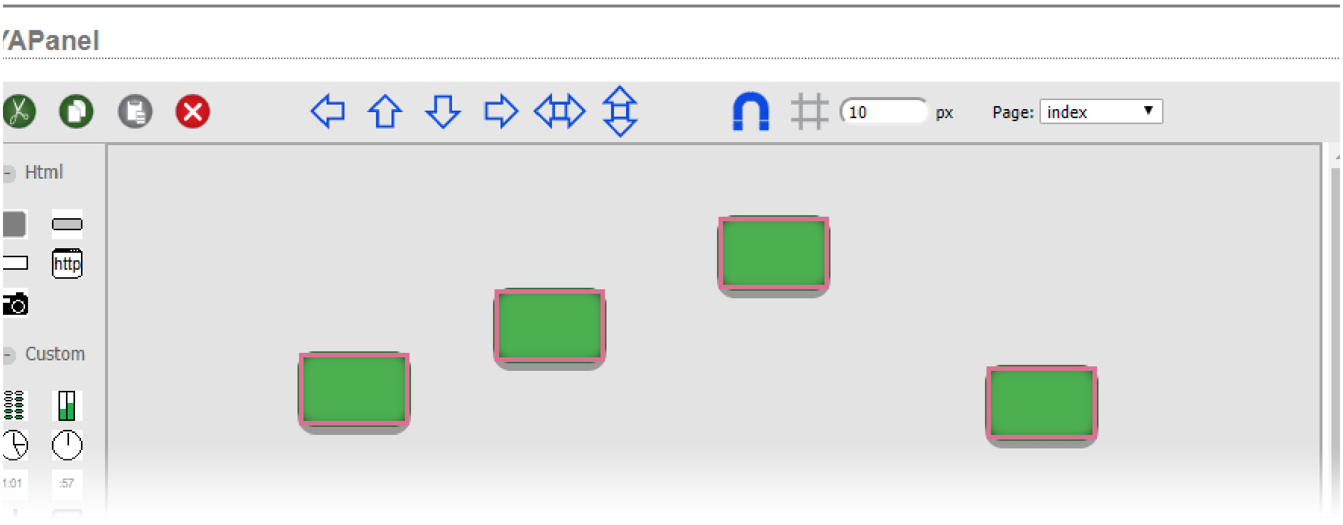
Alignment Tools



Page: index

The alignment tools will only be available for use if you have multiple components selected. You can select multiple components by holding the SHIFT key while you click additional components.

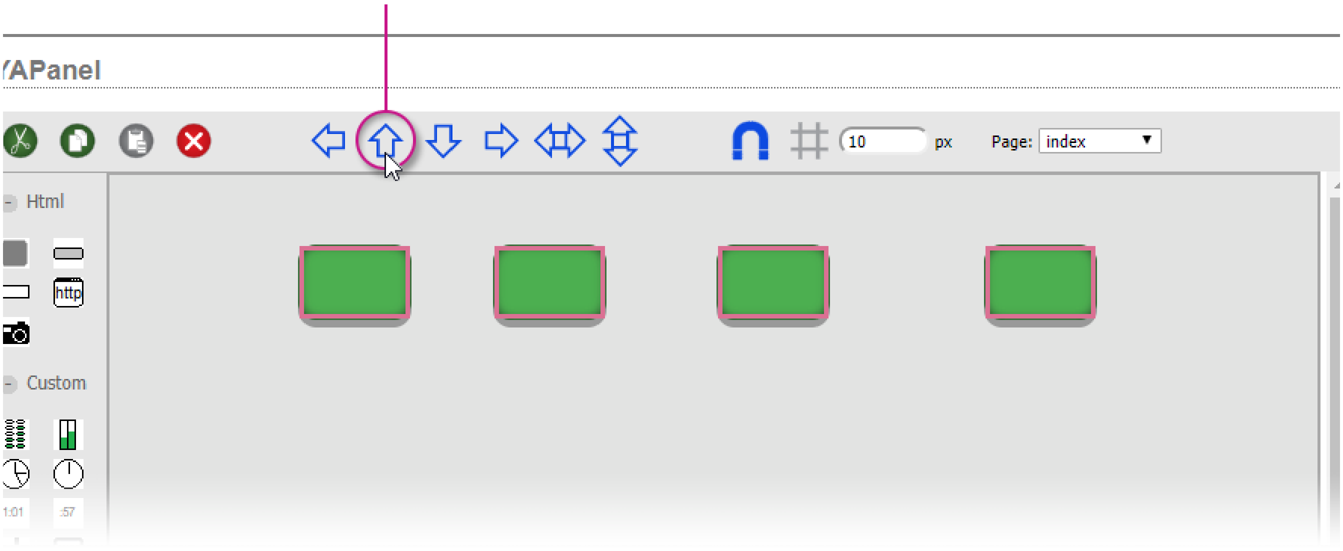
To illustrate, drag three or four buttons into the Panel. Then, while holding the SHIFT or CTRL key, click each component until the red select box is around each of them:



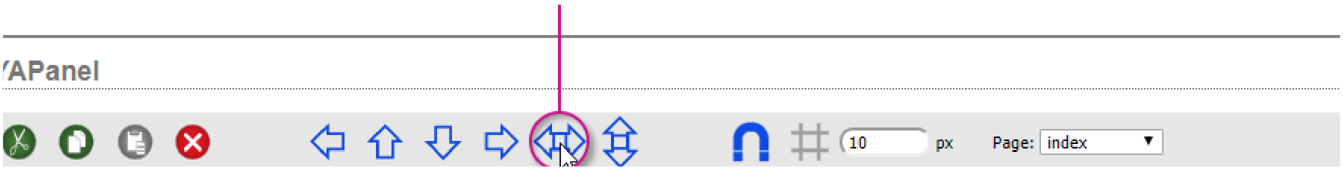
Once you have selected more than one component, the alignment tools will become available. These tools are:

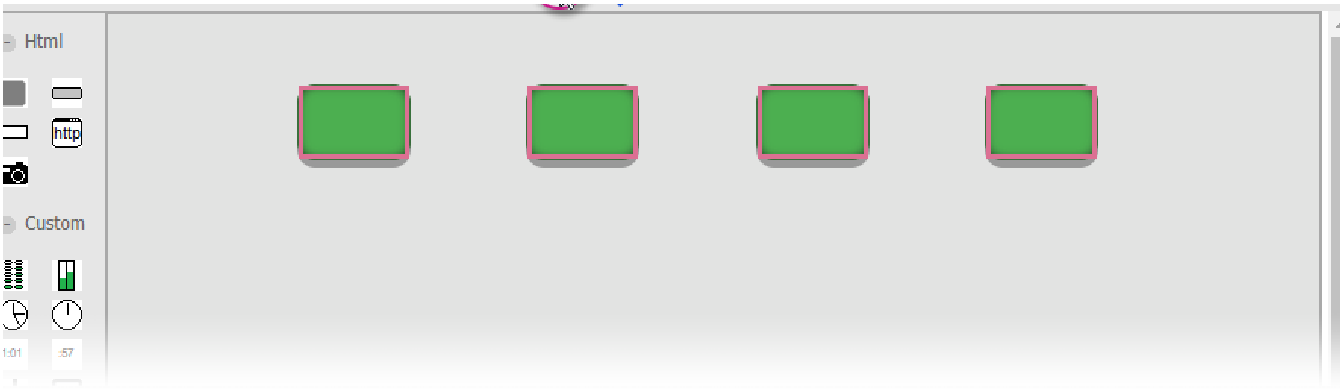
⇦	⇧	⇩	⇨	⬄	⇳
Align left	Align top	Align bottom	Align right	Distribute horizontally	Distribute vertically

In each case the system will align to the most extreme edge. For example, with the four buttons selected in the example above, clicking **Align top** will align all buttons to the top edge of the button closest to the top of the Panel.

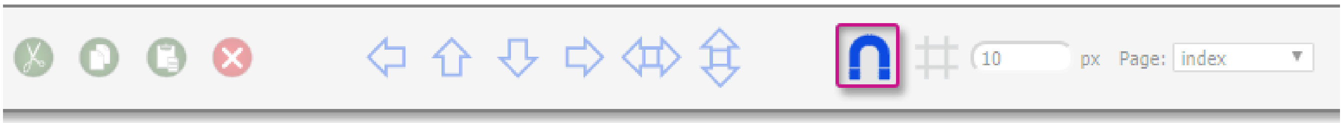


Clicking **Distribute horizontally** will spread the buttons evenly between the left and right buttons.

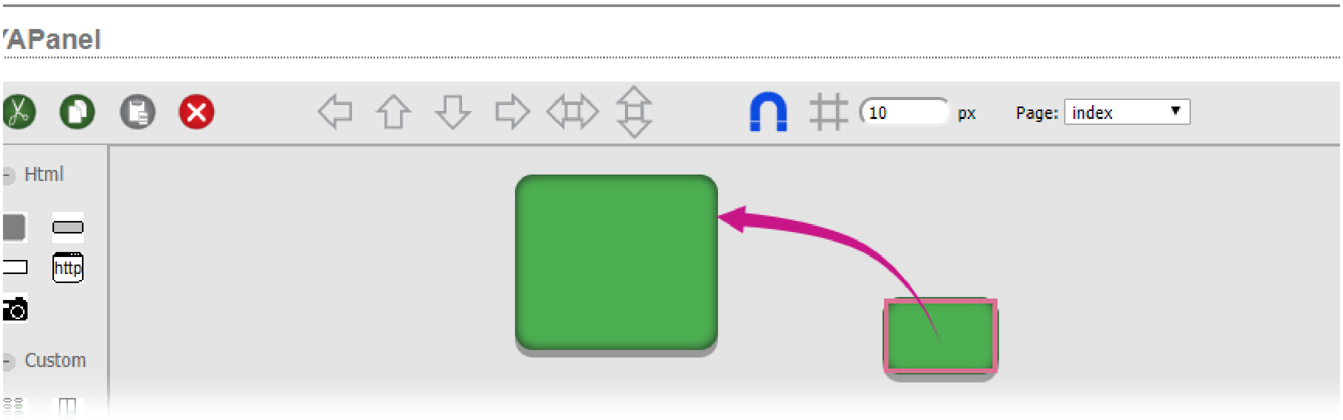




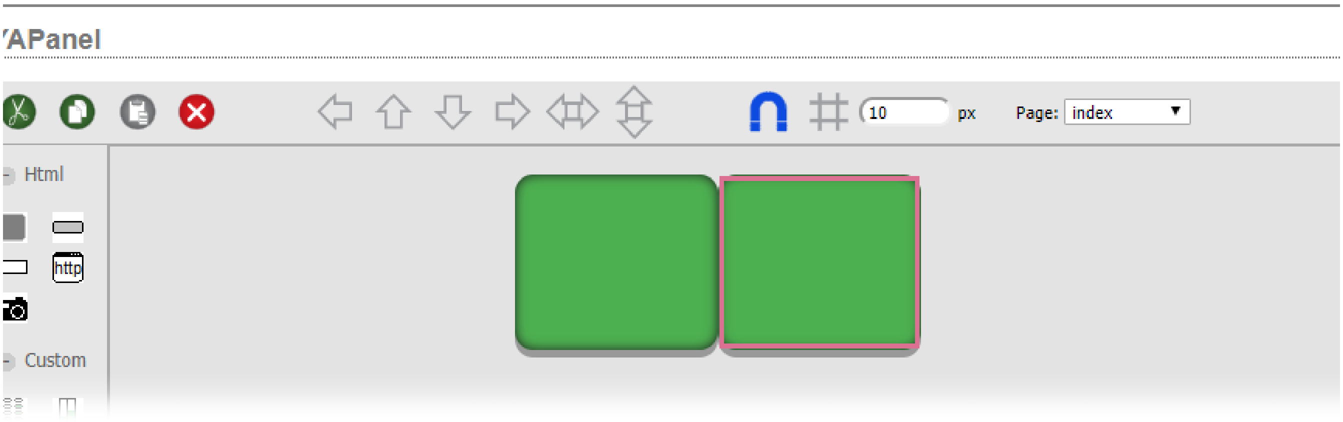
Magnet



The Magnet tool aligns and resizes items of like kind edge-to-edge. For example, create a new button and resize it. Then add a second button but leave it at the default size. If you then drag the small button so its edge meets the large button…

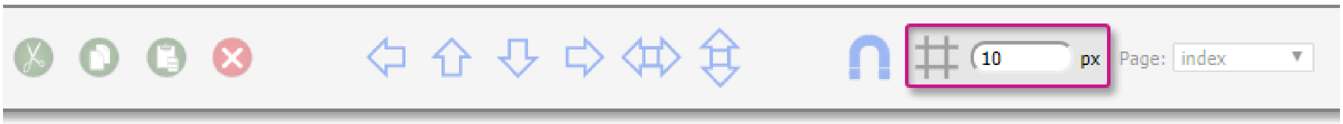


…the small button will immediately snap to the size of the large button and align itself to the large button's edge.



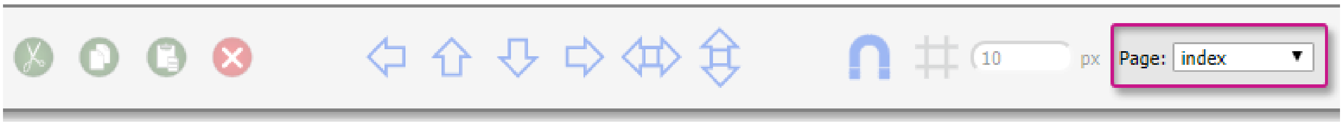
This is extremely useful when trying to build a Panel with many same sized components lined up edge to edge. However, it can also be disconcerting when you want the components to be different sizes and/or not to line up. The Magnet tool is enabled by default but may be disabled by clicking on the tool

Grid



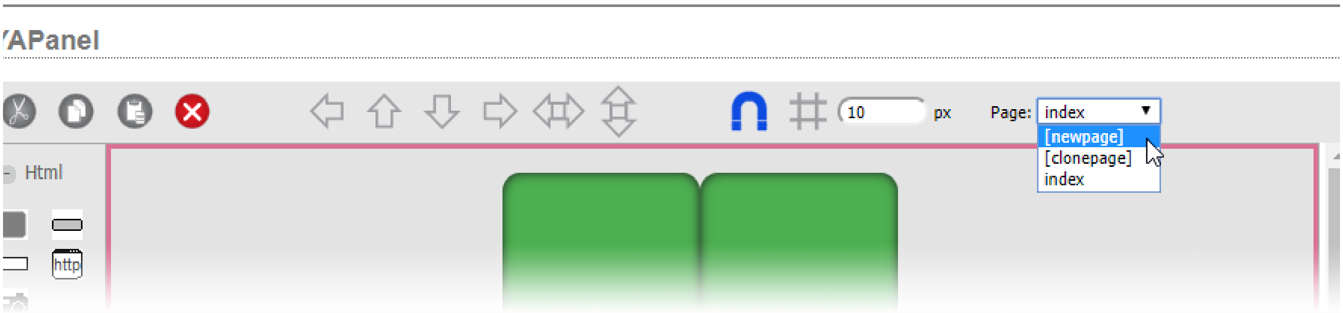
If enabled, the Grid tool aligns a component's location and size according to a grid of a specified pixel density. For example, if you enabled the grid with a pixel density of 10 pixels, you will notice that dragging components will jump by 10 pixels. This is useful when trying to evenly align components.

Page

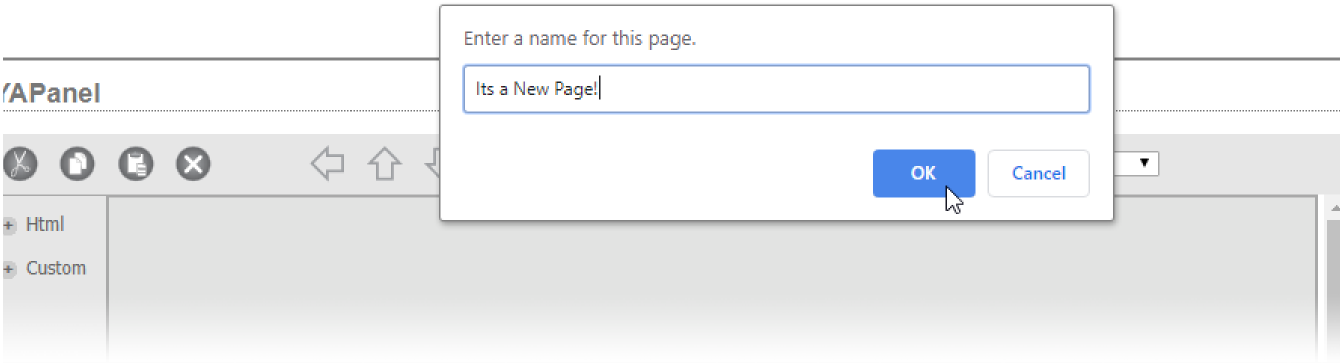


Each Panel has an index page which is the default page that will be loaded when the Panel is first displayed, but additional Pages may be created using the Page tool. Buttons or events may then be used to switch between the pages as described in the changing pages section of this chapter.

To design a new Page, click the down arrow and select the **[newpage]** option.



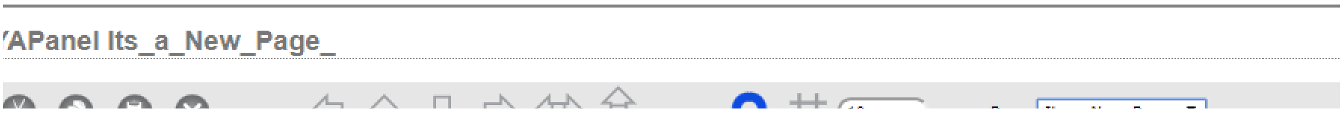
After clicking the [newpage] option, click **Save** and type a **name** for your new page. Then you can design a new page as if it was any other Panel.

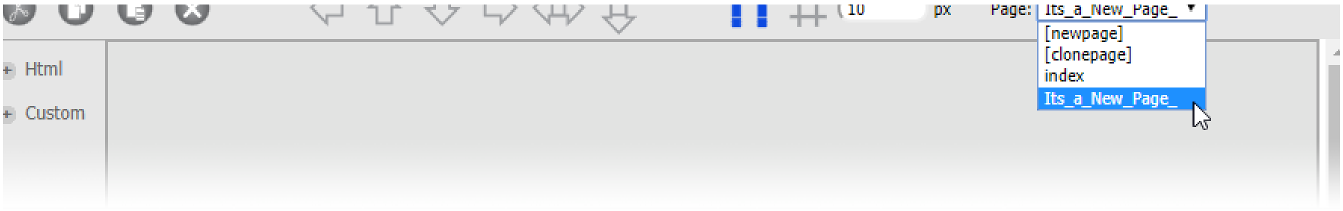


Enter a name for this page.

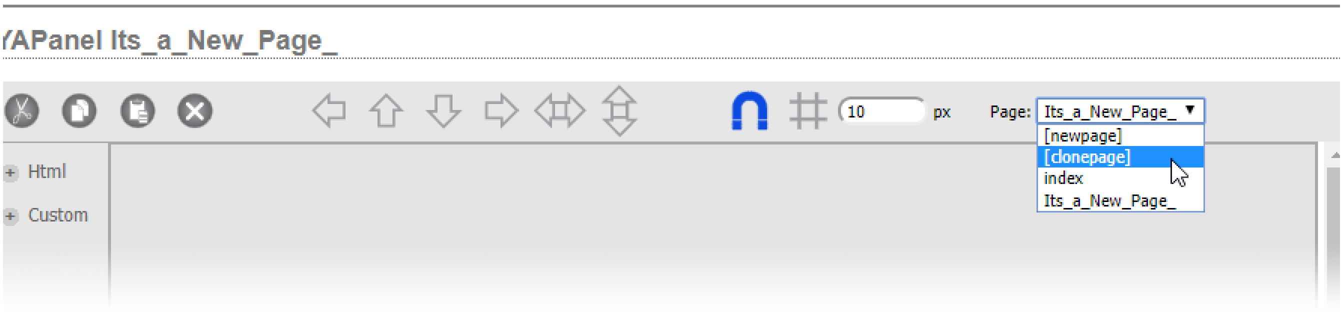
Its a New Page!

If a Panel has multiple pages, you can use this same dropdown to select the page to edit.

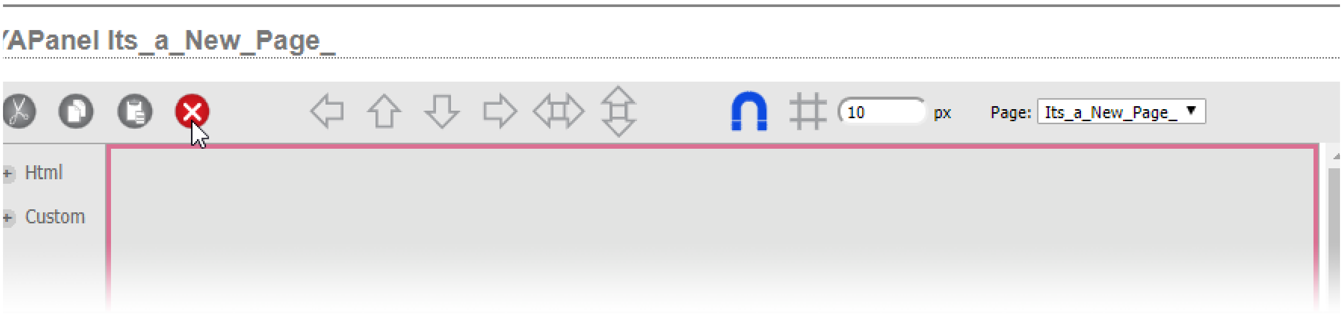




You can clone your work to a new page by selecting the page to be cloned and then selecting the **[clonepage]** drop-down item. The system will ask you for a new page name for the cloned page.



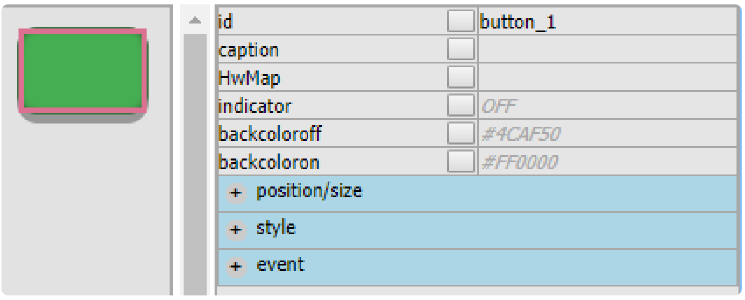
You can delete a page by selecting the page background and clicking the **delete** icon. The system will then ask you if you want to delete the page. You cannot delete the default index page.



Once multiple pages have been created you can use the change property along with buttons or property changes that will switch from one page to another. This is described in the *Changing Pages* section of this chapter.

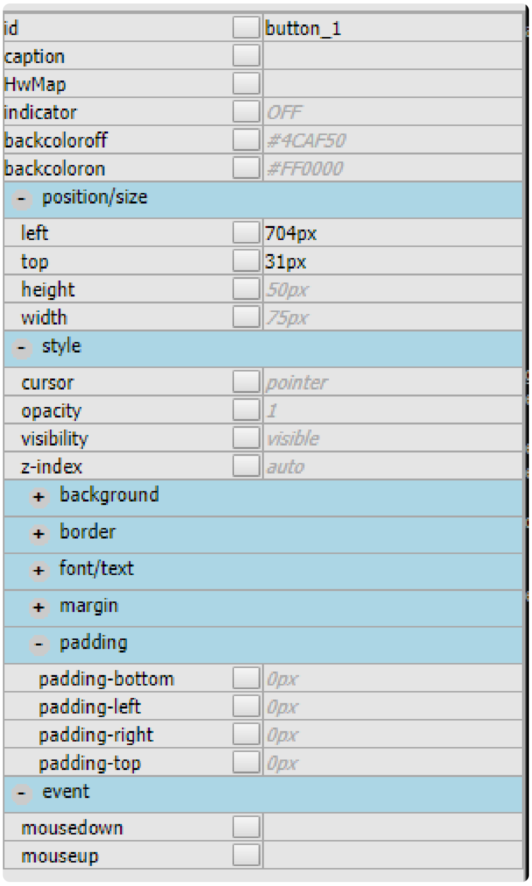
Property Grid

You were already introduced to the property grid in the section on adjusting properties above. This section will go into some additional detail. Drag a button onto your Panel and select it so that the property grid displays the available properties of the button.



id	□	button_1
caption	□	
HwMap	□	
indicator	□	OFF
backcoloroff	□	#4CAF50
backcoloron	□	#FF0000
+ position/size		
+ style		
+ event		

Different components may have different property sections and sub-sections as well as properties that are specific to that component, but this is an example of the property sections you will see. Expanding the sections will display additional sections and properties.



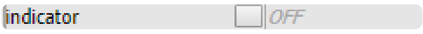
id	□	button_1
caption	□	
HwMap	□	
indicator	□	*OFF*
backcoloroff	□	*#4CAF50*
backcoloron	□	*#FF0000*
- position/size		
left	□	704px
top	□	31px
height	□	*50px*
width	□	*75px*
- style		
cursor	□	*pointer*
opacity	□	*1*
visibility	□	*visible*
z-index	□	*auto*
+ background		
+ border		
+ font/text		
+ margin		
- padding		
padding-bottom	□	*0px*
padding-left	□	*0px*
padding-right	□	*0px*
padding-top	□	*0px*
- event		
mousedown	□	
mouseup	□	

The majority of these properties are standard CSS style properties used by any web page designer. One of the best references we have found for CSS properties is w3schools.com: https://www.w3schools.com/CSSref/. This link will provide information about all of the CSS properties exposed in the property grid along with their meaning and usage information.

There are also some properties that you will not find in the CSS reference above because they are custom to our usage of that component. For example, in the case of the button component, *caption*, *HwMap*, and *indicator* are all properties that are not standard CSS properties. We will describe their usage more in the examples below.

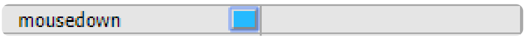
Bind Button

Each property in the property grid has a button between the property name and the property value at the end of the property name side of the grid. This is called a bind button.



indicator □ *OFF*

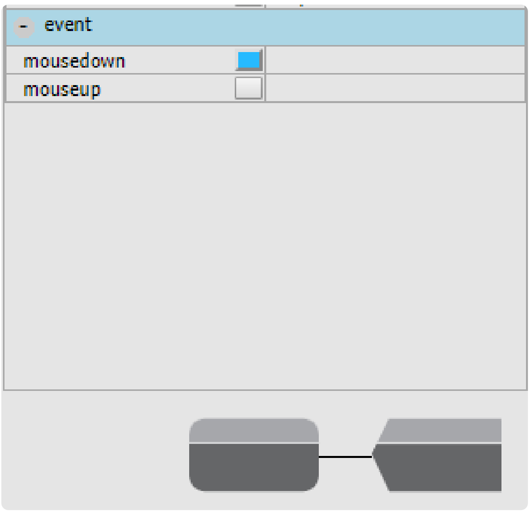
The bind button defines the properties that should be exposed to PathfinderCore PRO for use in Logic Flows. In some cases, there may be hundreds of properties for a given component, but there are only a few that you will want to dynamically change while the Panel is running. For example, once you position a button on a Panel and size it to the size you desire, it is unlikely that you will want that position to change while your end-user is using the Panel. Therefore, there is really no need for the left and top properties to be cluttering up the Logic Flow tree. Clicking the bind button for a given property will turn the button blue.



Saving the Panel will then identify to PathfinderCore PRO that this is a property that we expect to dynamically manipulate with Logic Flows and so should be tracked by PathfinderCore PRO and made available to Logic Flows.

Binding Flows

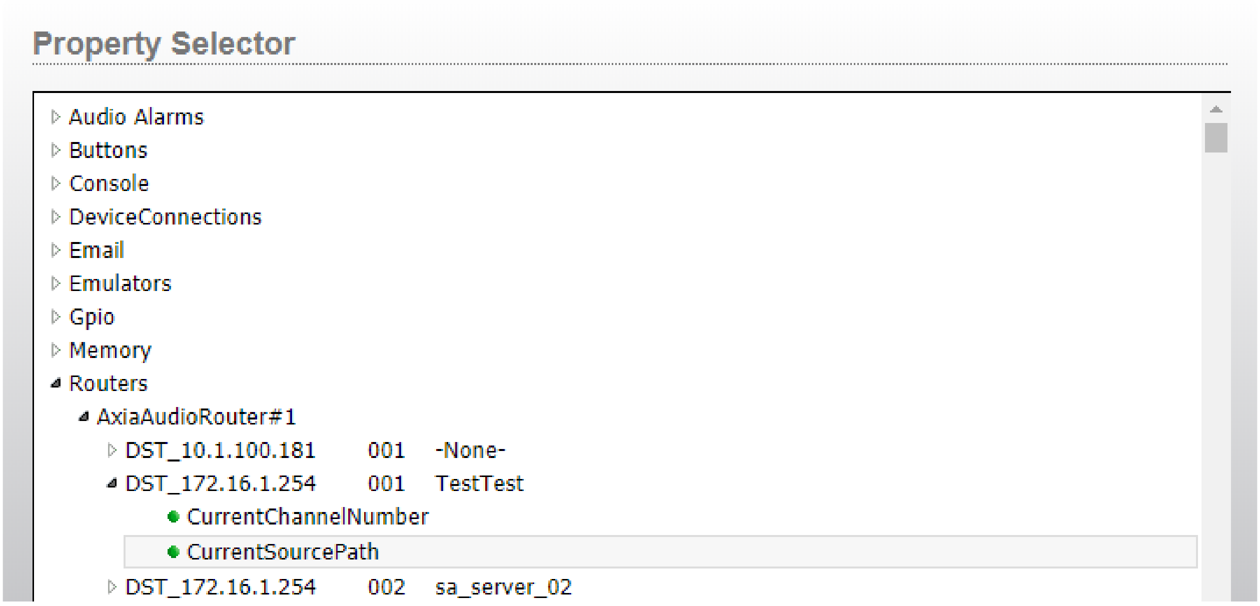
You may also notice that after enabling a property for binding that an image of a partial Logic Flow will appear at the bottom of the property grid:

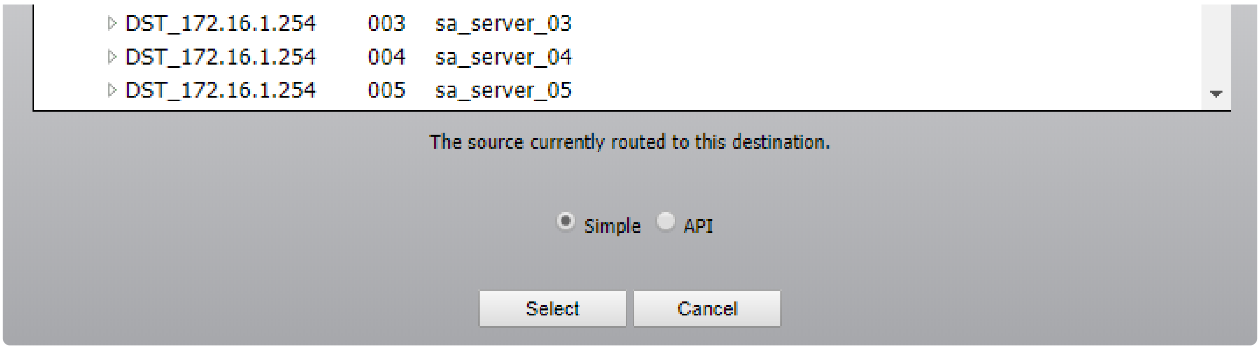


This is a simple shortcut that allows you to generate a simple flow to bind values to the property without having to switch over to the Logic Flow designer. In addition, since these flows simply bind system states to Panel properties, the flows generated by this method do not count against your license count. It is an easy way to quickly add simple functionality. But it will be easier to understand with an example.

Let’s say we want the button we have dragged onto the Panel to trigger a route change. Select the button and enable the binding button on the mousedown event. We are defining what we want to happen when the button is pressed. Then double-click on the endpoint in the flow image.

This will open the normal property selection dialog used in Logic Flows:





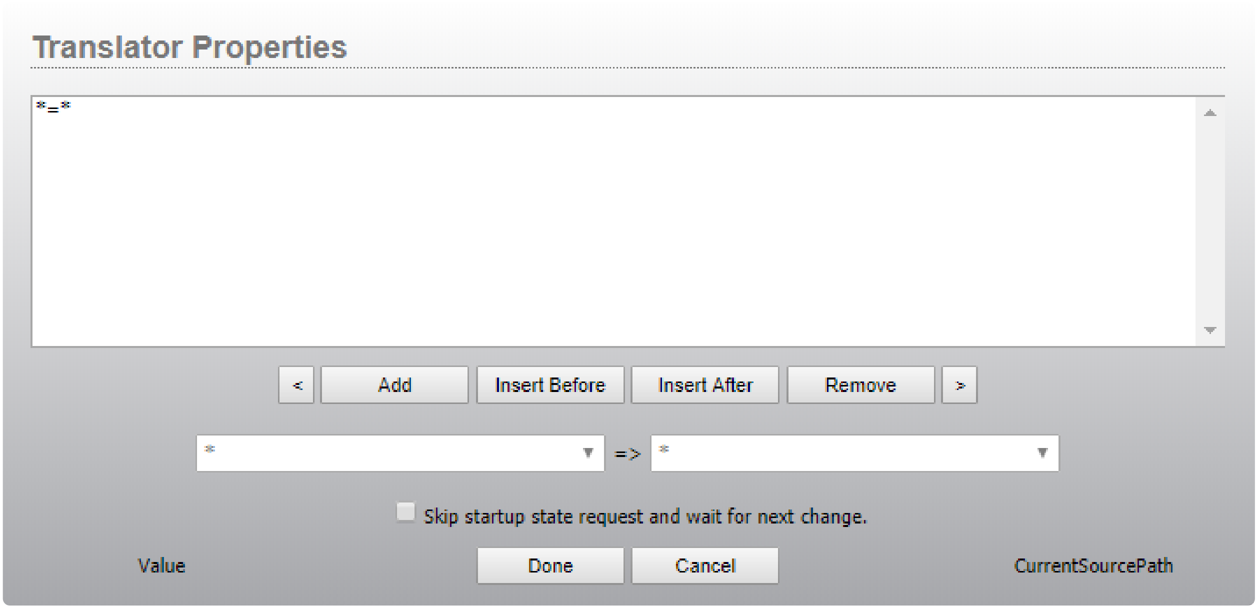
```
▷ DST_172.16.1.254    003   sa_server_03
▷ DST_172.16.1.254    004   sa_server_04
▷ DST_172.16.1.254    005   sa_server_05
```

The source currently routed to this destination.

● Simple ○ API

Expand the Routers section, expand a router, and expand the destination you want to change when the button is pushed, and then click on the CurrentSourcePath Property. Then click select.

The system will automatically move to the translation dialog.



Translator Properties

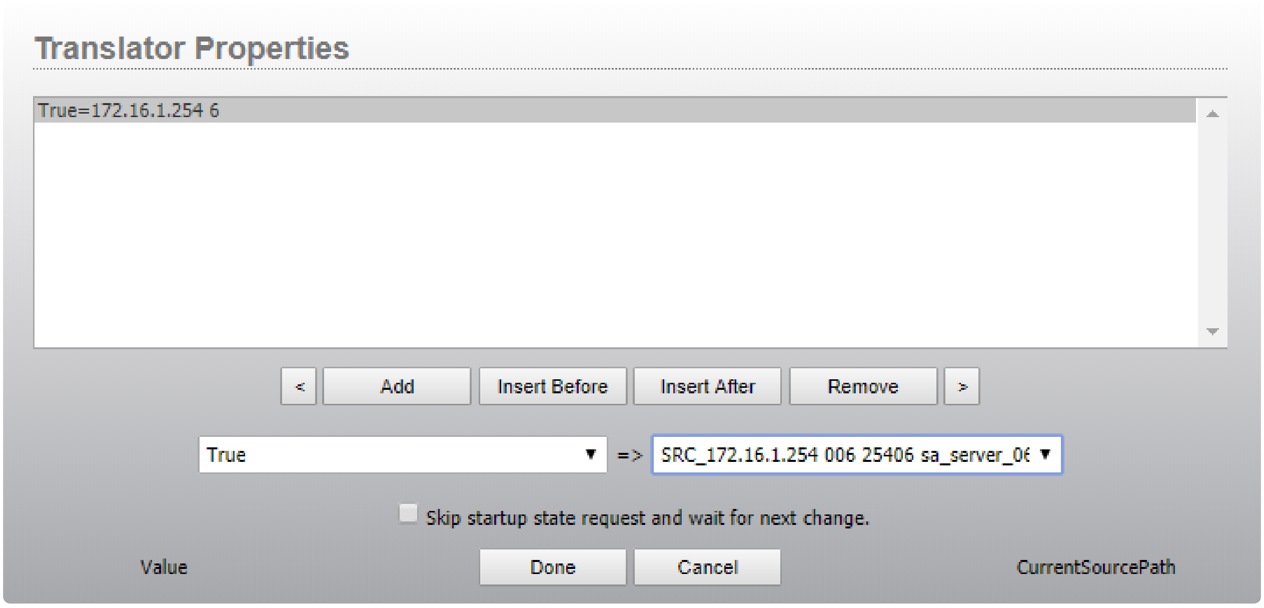
```
*=*
```

* => *

□ Skip startup state request and wait for next change.

Value   CurrentSourcePath

Click on the *=* item in the list and then select the True item in the left hand drop down and the source you want to route to the selected destination in the right hand drop down.



Translator Properties

```
True=172.16.1.254 6
```

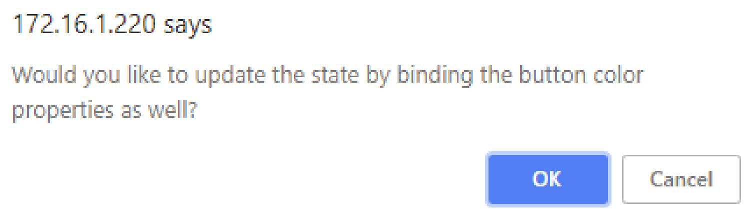
True => SRC_172.16.1.254 006 25406 sa_server_0€

□ Skip startup state request and wait for next change.

Value   CurrentSourcePath

We have just defined that if the mousedown event is true, the sa_server_06 source will get routed to the

TestTest destination. However, when you click Done you will get a pop-up message.

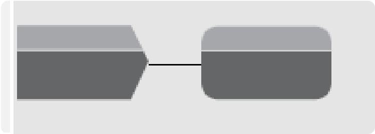


172.16.1.220 says

Would you like to update the state by binding the button color properties as well?

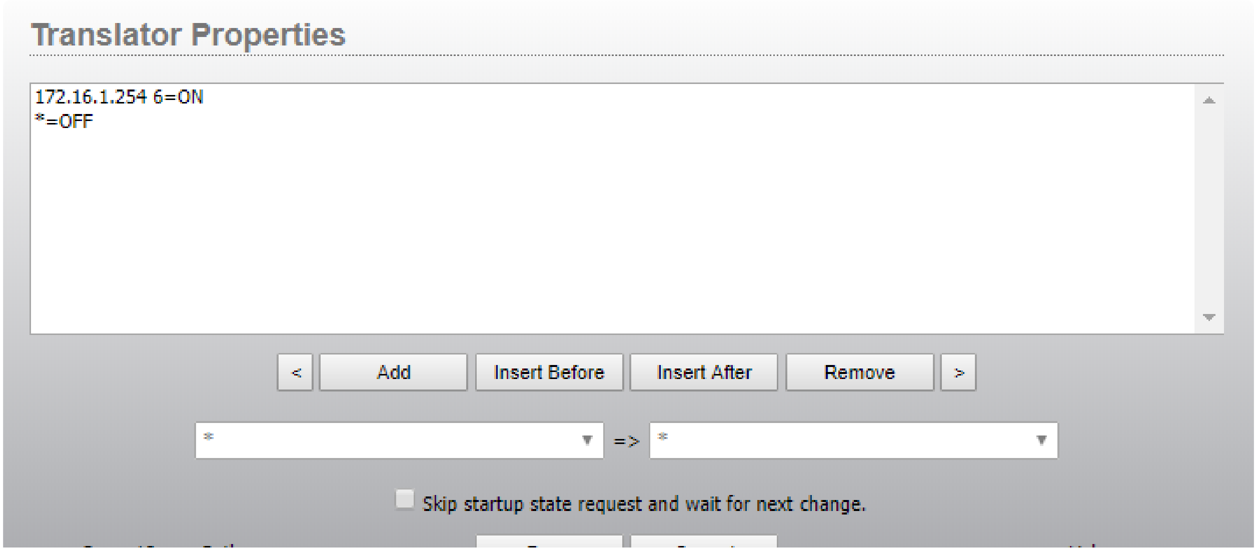
This message will only appear if you are generating flows on the mousedown or indicator properties of a button. In this case, it knows that since we are defining what we want the button to do, we probably also want some indication on the button that the requested action has been done. If we click OK, it will automatically turn the binding button on for the indicator property and open the flow definition for the indicator. In this case, it is smart enough to fill things in for us.

It is important to notice that the flow for the indicator property looks different than the one for the event.



The system also knows in which direction these flows should go. For example, with an event, the start point is not displayed in the flow because the event we have selected is the start point and the end point is what we are going to change. On the other hand, standard properties like the indicator are changed based on things that are changing in the system. So, you select what property in the system is causing the indicator to change. In that case, the partial flow shows the start point and the translation and the endpoint is the property of the Panel component we have selected. The rule of thumb is that events will display partial flows with an endpoint and other properties will display a flow with a start point. The missing part of the flow is the event or the property itself.

When we click OK in response to the message above you will notice the system will skip picking the start point. This is a special case for buttons where you are configuring the mouse down and indicator properties. Since we just defined what we want to change when mouse down is pressed, the pop-up message is asking whether we want the successful change of that route to be reflected in the indicator. So, if we click OK, the system automatically turns on the binding for indicator and fills in the start point with the destination selected, and then displays the translation settings.



Translator Properties

```
172.16.1.254 6=ON
*=OFF
```

* => *

□ Skip startup state request and wait for next change.

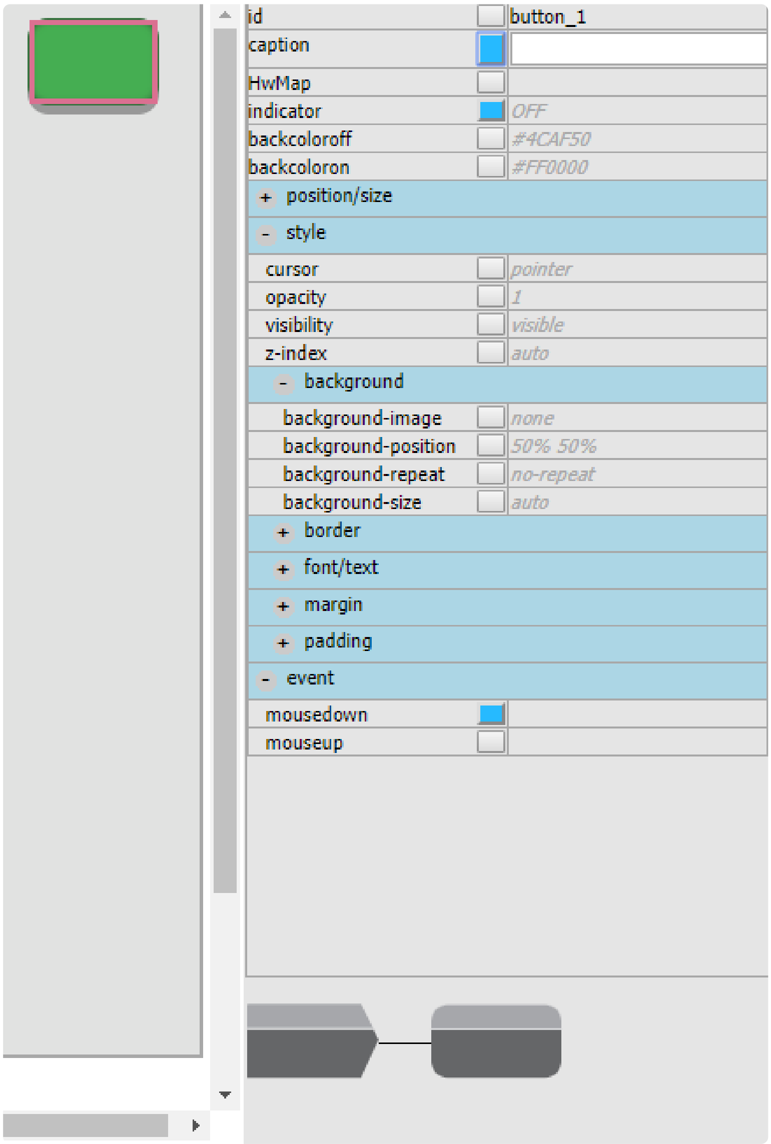
You will also notice that the system is assuming you will want the indicator to be on if the selected source is routed to the destination and off if it is not.

Click **Done**.

You will notice that the flows are no longer gray and have turned blue to indicate they have been defined. Saving the Panel will cause the flows to be created and start working in Logic Flows. Flows created in this manner will be generated in a special folder in Logic Flows called _Panels. The flows in this folder may be monitored for troubleshooting purposes but they cannot be changed from within Logic Flows. They are only edited through the Panel designer.

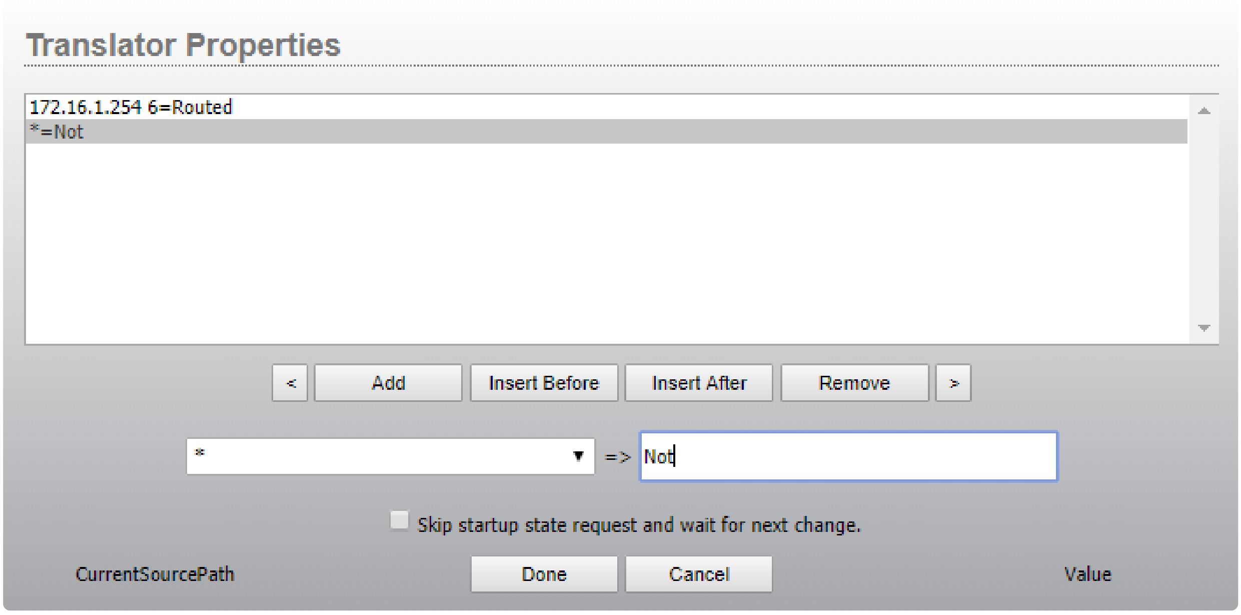
> Note: To see these flows working you need to go back to User Panels and open the Panel for usage by clicking on the Panel link rather than the edit link. It is helpful to have this open in a separate browser tab while you are working. Then, after saving changes in the designer, you can switch over to the tab with the running Panel, refresh the page, and see your changes in action.

To extend this example, turn on the binding for caption as well.



id	□	button_1
caption	☑	
HwMap	□	
indicator	☑	OFF
backcoloroff	□	#4CAF50
backcoloron	□	#FF0000
+ position/size		
- style		
cursor	□	pointer
opacity	□	1
visibility	□	visible
z-index	□	auto
- background		
background-image	□	none
background-position	□	50% 50%
background-repeat	□	no-repeat
background-size	□	auto
+ border		
+ font/text		
+ margin		
+ padding		
- event		
mousedown	☑	
mouseup	□	

Now double click on the start point and select the same CurrentSourcePath property of the same destination. Now in the translation select what you want the button to say when the source is routed and what you want it to say when it is not.



Translator Properties

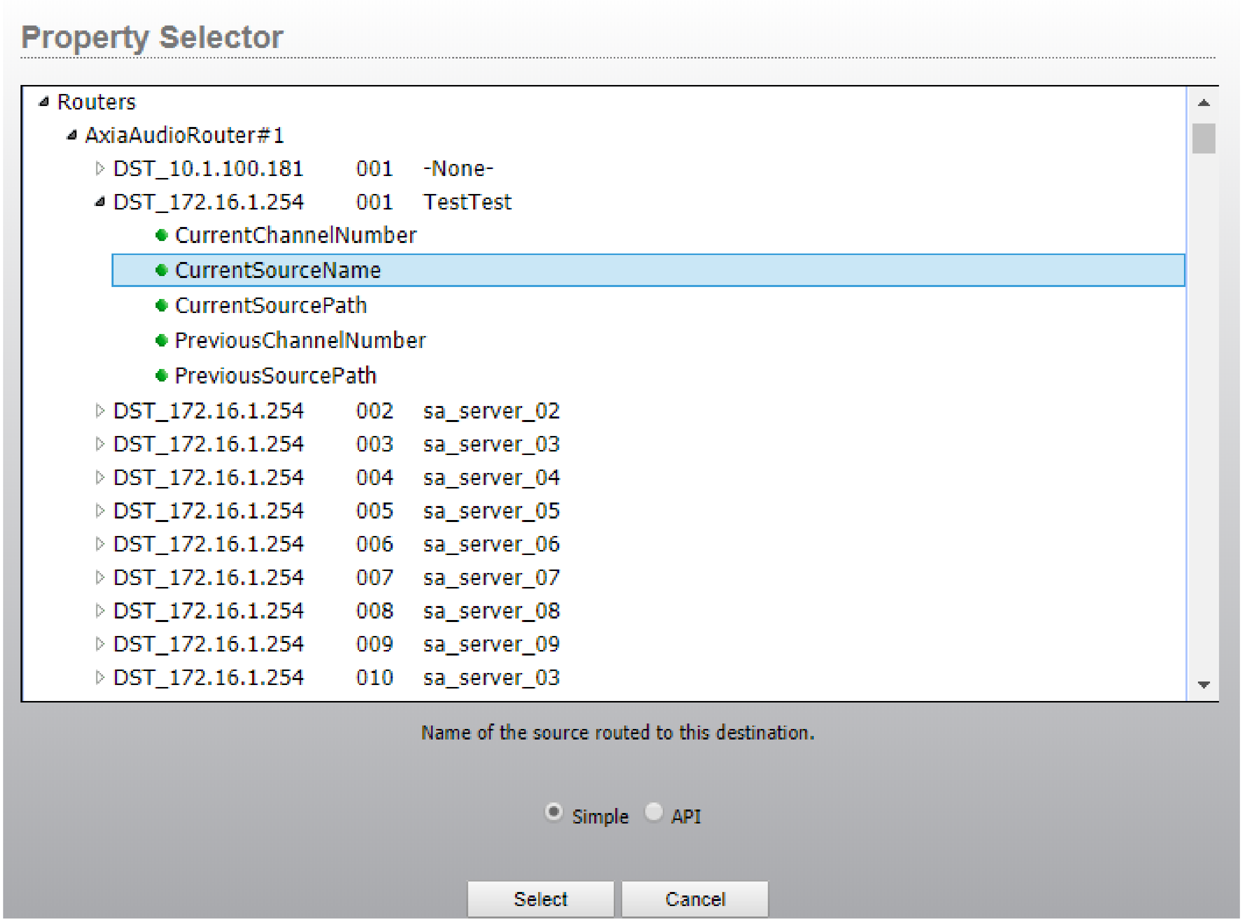
172.16.1.254 6=Routed
*=Not

* => Not

□ Skip startup state request and wait for next change.

Click **Done** and **Save** to save your changes. Executing the Panel should now display Routed or Not in the button's caption depending on whether the selected source is routed to the destination.

A more useful change you can make with the caption property is to select the CurrentSourceName property of the destination in the Logic Flow property selector and use a *=* translation. You can change this by double-clicking the start point while the caption property is selected.



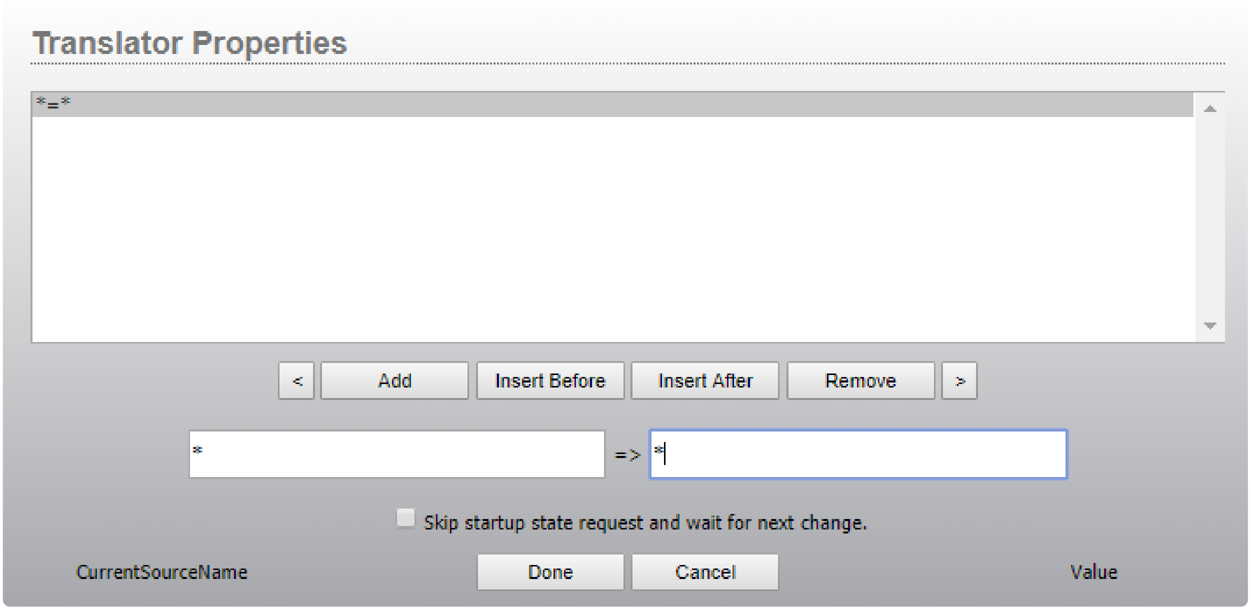
Property Selector

```
Routers
  AxiaAudioRouter#1
    DST_10.1.100.181     001   -None-
    DST_172.16.1.254     001   TestTest
        CurrentChannelNumber
        CurrentSourceName
        CurrentSourcePath
        PreviousChannelNumber
        PreviousSourcePath
    DST_172.16.1.254     002   sa_server_02
    DST_172.16.1.254     003   sa_server_03
    DST_172.16.1.254     004   sa_server_04
    DST_172.16.1.254     005   sa_server_05
    DST_172.16.1.254     006   sa_server_06
    DST_172.16.1.254     007   sa_server_07
    DST_172.16.1.254     008   sa_server_08
    DST_172.16.1.254     009   sa_server_09
    DST_172.16.1.254     010   sa_server_03
```

Name of the source routed to this destination.

● Simple ○ API

Now pick the currentSourceName Property instead of the CurrentSourcePath property and click select.


Translator Properties

=

* => *

□ Skip startup state request and wait for next change.

CurrentSourceName

Value

Change the translation to be *=*. Then click **Done**. Now the button's caption will be tied to the name of whatever source is currently routed to the destination.

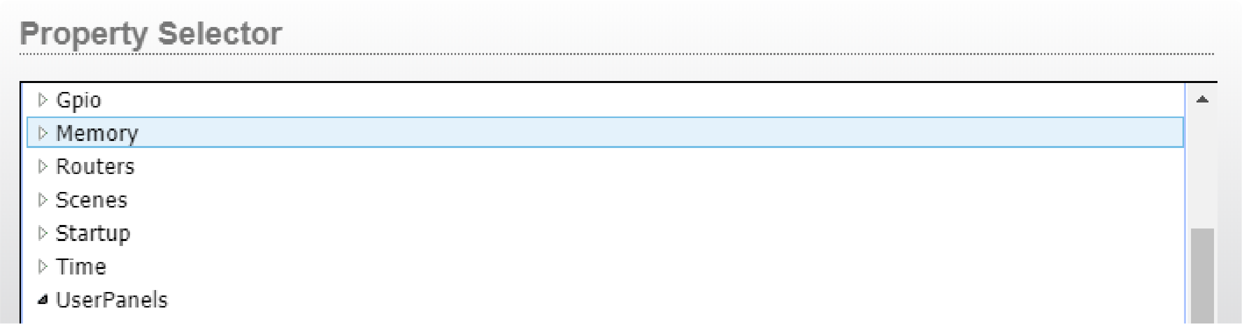
After saving the Panel and opening it up for use you should find that pressing the button will make the route change, the indicator will illuminate or go dark according to the back-color properties depending on whether the route is made, and the caption should display the name of the source that is routed to the destination.

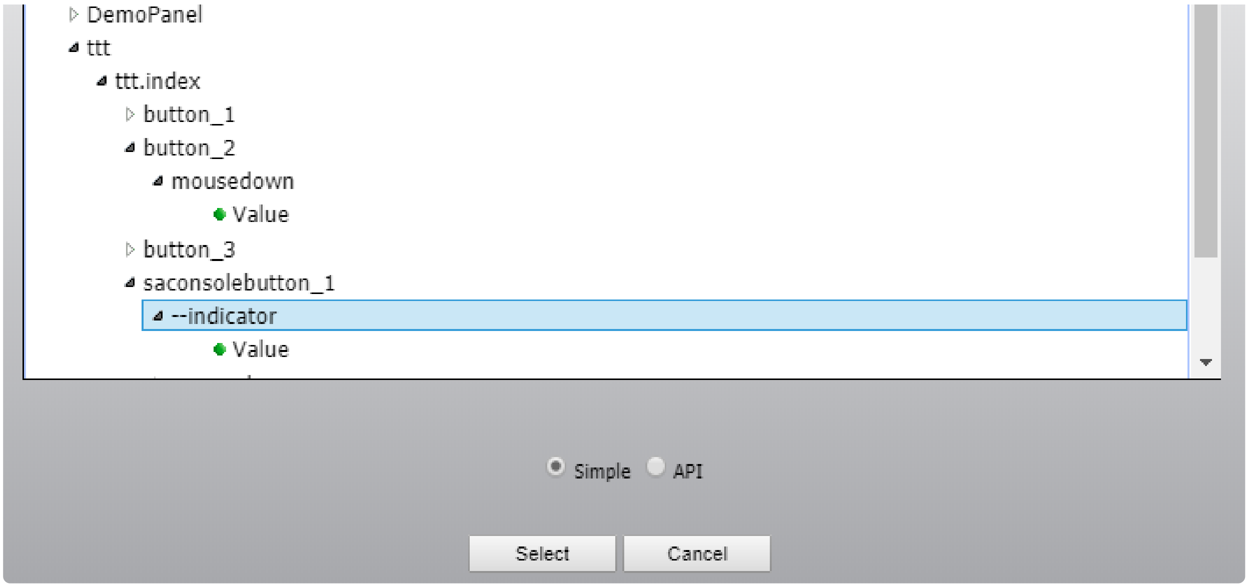
By using these techniques, you can edit functionality into the Panel components in very easy and extremely powerful ways.

Complex Panel Flows

In many cases you may wish to create more complex flows than described in the examples above. For example, you may want your indicator state on a button to be the product of numerous conditions in the system. These kinds of Flows can still easily be created but must be created within the Logic Flows designer.

Simply enable the binding button for the properties these Flows need to manipulate without generating a Flow in designer. Save the Panel and then from within the Logic Flows property selector, these properties will be available for use:





- DemoPanel
- ttt
  - ttt.index
    - button_1
    - button_2
      - mousedown
        - Value
    - button_3
    - saconsolebutton_1
      - --indicator
        - Value

Launching the Panel from a Desktop Icon (Windows)

If you are using Google Chrome, there are some command line options that will allow you to launch a Pathfinder Core PRO Panel as if it was an application. Copy and paste the following into a notepad or text editor:

```
"C:\Program Files (x86)\Google\Chrome\Application\chrome.exe" --app="data:text/HTML,<HTML>
```

When copied, this should be one line in the editor. The quotes at the beginning and end are part of the text so do not remove them. After the word “location” there is an http link.

- Change **[Username:Password]**to match the credentials used in your system.
- Change **[IPAddress]**to match the IP address of your Pathfinder Core PRO.
- Change **[ttt]** to the name of the Panel you want to launch.
- Change **[index]** if you want to target a page other than the default index page

For example, these values…

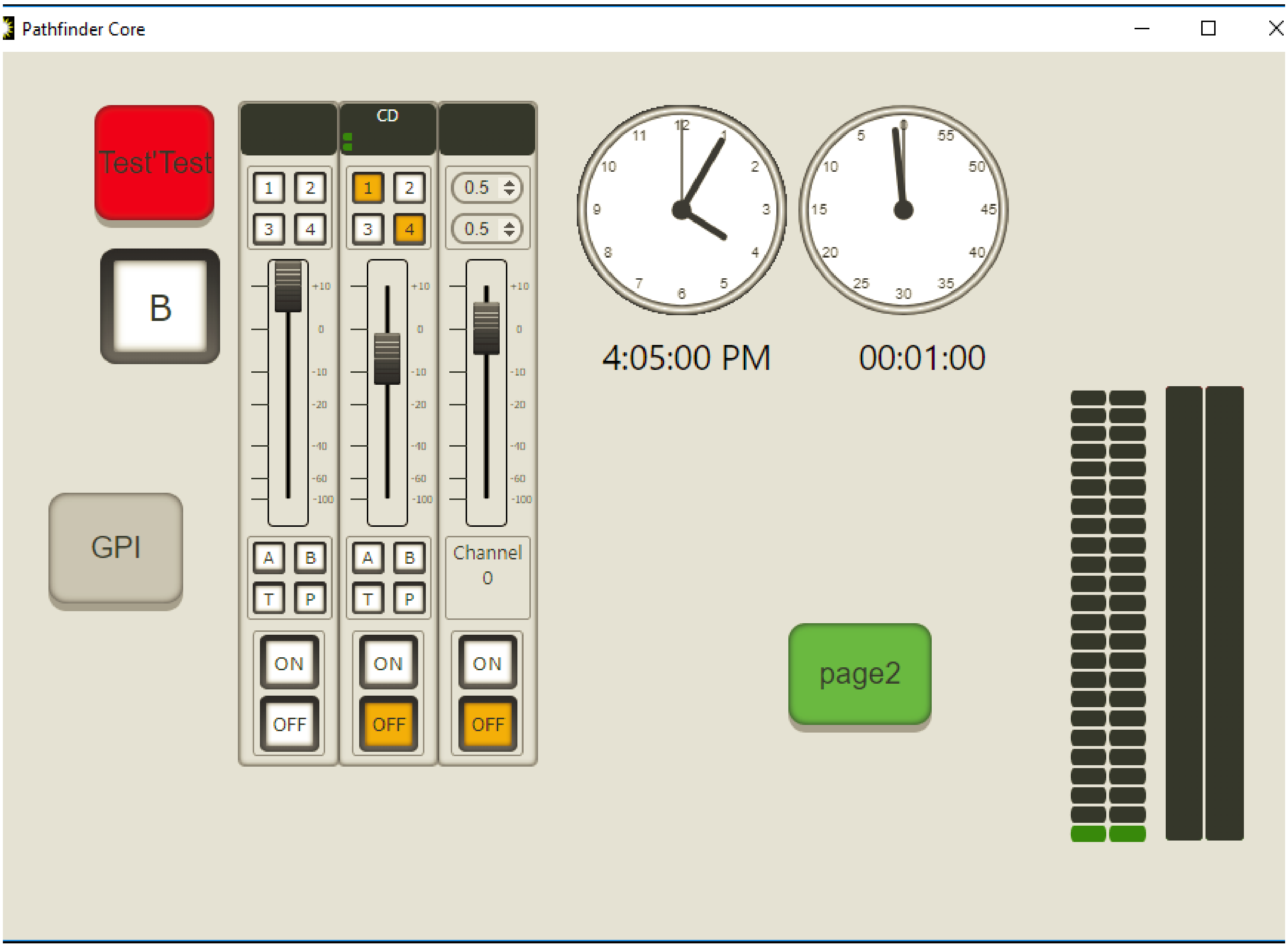
- **[Username:Password]**= Admin:Admin
- **[IPAddress]**= 172.16.1.56
- **[ttt]** =YAPanel
- **[index]** = Its_a_New_Page

…would result in a shortcut like this:

```
"C:\Program Files (x86)\Google\Chrome\Application\chrome.exe" --app="data:text/HTML,<HTML>
```

Once you have the edits made, select the entire text again and copy it to your clipboard.

Right-click on your desktop and create a new shortcut using your edited shortcut text. Double-clicking on the shortcut should now launch the Panel as its own application.



If you always want the Panel to launch in the same place on the screen you can add another option: window.moveTo(580,240). For example:

```
"C:\Program Files (x86)\Google\Chrome\Application\chrome.exe" --app="data:text/HTML,<HTML>
```

In the future, we will investigate a way to generate the shortcut (or at least the shortcut text) automatically.

Component Reference

Each component has many available CSS properties and component events. Many of these properties can be understood after some basic study that starts with a basic internet search for CSS property descriptions.

This section will describe each component, including custom properties beyond the standard CSS style properties.

We have included default images of the components below, but it is important to note that the CSS properties can be used to make the components look much different than the examples included below. Feel free to adjust border, color, and shadowing properties to achieve the design desired.

Important Note: Each component has an ID property allowing you to define a name for the component. This is useful in differentiating components in Logic Flows and debugging so it is a good habit to name components as you create them. IDs must be unique within the page.

HTML Components

Button

The **Button** component can control and indicate changes in the system. In addition to standard CSS style attributes, customizable properties include:

Property	Function
caption	Updates the inner HTML **text** of the button.
hwmap	Used to select a hardware LCD button. This hardware LCD button will then mirror the behavio of the software button. *See the* ***Hardware Mappin Buttons*** *topic below for more information.*
indicator	Used to set the button indication to *On*, *Off*, *Flash,* and *PULSE0* through *PULSE10000.* *See the* ***Button Indicator PULSE Values*** *topic below for more information*.
backcoloron	Sets the button’s color when the indicator state is **on**.
backcoloroff	Sets the button’s color when the indicator state is **off**.

Wherever this button can be used, a Console Button could also be used. The difference is that the Console Button has a slightly more elegant look.

Label

Labels can be used to generate textual label information in the system. In addition to standard CSS style attributes, customizable properties include:

Property	Function
textContent	Updates the inner HTML **text** of the label. Binding this property allows the label caption to be dynamically updated by Flows.

Input Box

Input boxes allow user input. In addition to standard CSS style attributes, customizable properties include:

Property	Function
value	Captured user input which can be applied to Logi Flows to make decisions and changes.
type	Allows you to switch the input type between several different possibilities including *text*, *number*, *color*, etc. For example, if you drag an Input box onto a Panel and change the type to *number* and then fill in the min, max, and step properties, you will obtain an input box with a set arrows for incrementing and decrementing the *value* property.

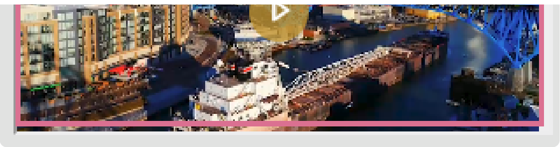
In addition to *mousedown* and *mouseup* events, Input Box events include:

Event	Function
change	The *change* event is raised when the user presse ENTER or moves out of the input field
input	The *input* event is raised as the user makes changes/types.

Both *change* and *input* events raise the current entered **value**.

To learn more about this feature visit: http://pathfinderpc.com/pfcorepro_downloads/panelinputboxes.mp4

Web Page



The **Web Page** component is an HTML iframe which allows you to embed a web page from another site into the Panel. This component can be used to display video streams and other web page content.

In addition to standard CSS style attributes, customizable properties include:

Property	Function
src	The **url** of the page to be displayed.

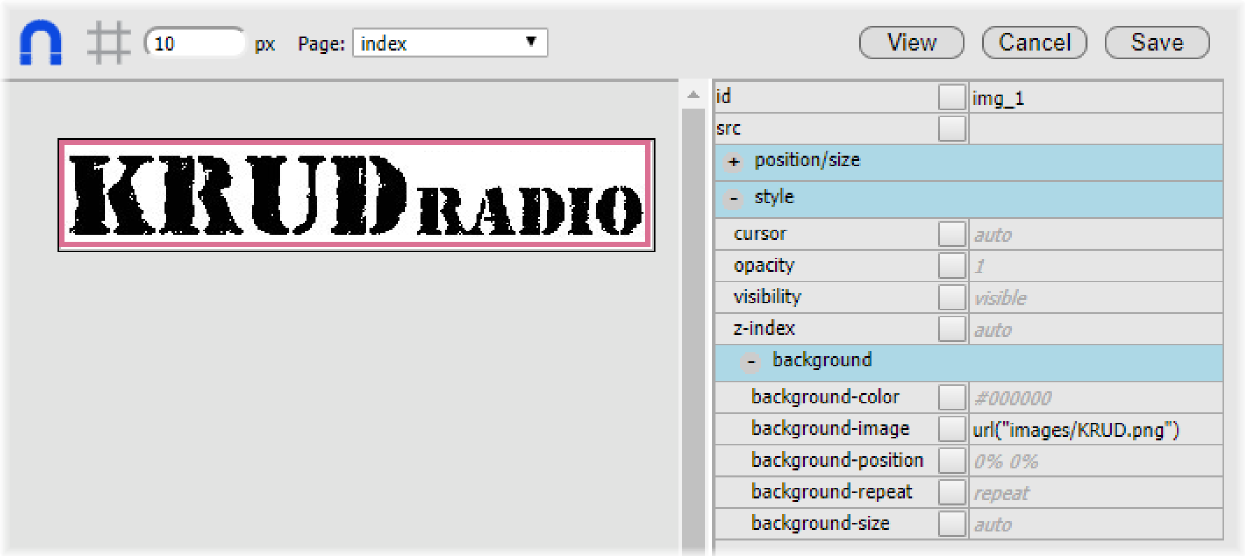
If you intend to use this as a background component with other components on top, you may need to manually adjust the z-index property of this component or the overlaid components to get them to display properly.

The Web Page component includes this event:

Event	Function
load	The **load** event cycles *False* when the frame start to load and *True* when the loading is complete.

Especially during testing, it is important to note that some sites (notably, Google) prevent their content from being displayed in an iframe.

Image



id	□	img_1
src	□	
+ position/size		
- style		
cursor	□	*auto*
opacity	□	*1*
visibility	□	*visible*
z-index	□	*auto*
- background		
background-color	□	*#000000*
background-image	□	url("images/KRUD.png")
background-position	□	*0% 0%*
background-repeat	□	*repeat*
background-size	□	*auto*

The **Image** component allows you to embed an image in a page. This component can be used to create background borders around a set of components.

It is recommended to use the *background-image* property rather than the *src* property to assign an image to this component

Before you use an **image** component, remember most elements also support the *background-image* property. For example, you can apply a *background-image* to the Panel itself or to buttons and labels.

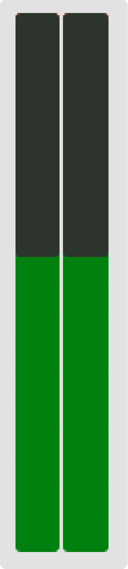
(In true radio fashion, the KRUD logo was lifted with no regard to the copyright owned by Jim Radcliffe and Brian Wilson. Their work is genius. Check it out at www.krud.com. Even better, embed it in a Panel using the Web Page component.)

Custom Components

Meters and Faders

HTML5 User Panels support two different kinds of **meters** (providing visualizations of Audio IOs that support metering) and two different kinds of **faders** (allowing gain manipulation of Audio IOs.)

Gradient Meter



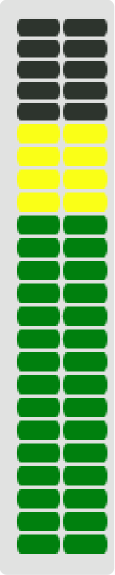
This component can be tied to Audio IOs that support metering in the system. *See the **Setting the IO Property** topic in this chapter.*

In addition to standard CSS style attributes, customizable properties include:

Property	Function
IO	Used to select the audio IO this meter will display.
orientation	Used to select whether the meter will display horizontally or vertically.
meterscale	Used to select the scale of the meter. Options include *standard*, *linear*, and *british*.

metrics	Used to define whether the numerical values for th meter are displayed next to the meter or not. Options include *none*, *lefttop*, *middle*, *rightbottom*.
optimumpercent	Defines the optimum meter percentage.
autosizefont	Used to define whether the metrics font will scale automatically as the size of the meter is adjusted.
led>color	Properties used to adjust the *on* and *off* colors use for each of three sections of the meter.

LED Meter



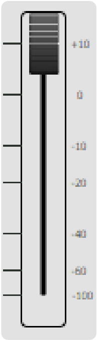
This component can be tied to Audio IOs that support metering in the system. *See the **Setting the IO Property** topic in this chapter.*

In addition to standard CSS style attributes, customizable properties include:

Property	Function
IO	Used to select the audio IO this meter will display.
orientation	Used to select whether the meter will display horizontally or vertically.
meterscale	Used to select the scale of the meter. Options include *standard*, *linear*, and *british*.
metrics	Used to define whether the numerical values for th meter are displayed next to the meter or not. Options include *none*, *lefttop*, *middle*, *rightbottom*.

autosizefont	Used to define whether the metrics font will scale automatically as the size of the meter is adjusted.
led>color	Properties used to adjust the *on* and *off* colors use for each of three sections of the meter.
led>border	Properties used to adjust the border settings of the individual LED blocks.

Fader



This component can be tied to Audio IOs that support gain manipulation. *See the **Setting the IO Property** topic in this chapter.* The Fader can also control non-audio control points. *See the **Setting the Control Property** topic in this chapter.*

The control displays the Fader value as you hover over the Fader. Fader values can be manipulated using the mouse wheel, dragging with the mouse, or touch and drag on a touch screen.

In addition to standard CSS style attributes, customizable properties include:

Property	Function
IO	Used to select the Audio IO this meter will display
control	Adds support for LwCH options and Qor faders. *See the **Setting the Control Property** topic in thi chapter.*
metrics	Determines how the numbering for the fader is displayed. Options include *none* and *lefttop*.
metricoffset	The percent of width used for the metrics.
metriclinecolor	The color used for the lines drawn for each metric line.
slider-height	The percentage of the overall component height used for the slider height. The greater the percentage, the taller the slider.
slider-width	The percentage of the overall component width used for the slider width. The greater the percentage, the wider the slider.

slider-opacity	Modifying this value will alter the opacity of the slider. Values can range between **0** (fully transparent) and **1** (fully opaque.)
slider-background	This property changes the slider color. The proper accepts gradient values: **linear-gradient(#770000, #ff0000)** or solid colors: **#770000**
slider-back-color	Primarily used for the simple and touch bar styles this will set the background color of the simple slider or of the touchbar. Clicking this field opens the standard color picker dialog. Additional color properties are available for these slider styles allowing for a great deal of granularit *See below for details*.
slider-margin-left	The margin offset for the slider.
slider-border-style	The slider border style.
slider-border-width	The slider border width.
slider-border-radius	The slider border radius.
slider-border-color	The slider border color.
slidebarwidth	The width in pixels of the bar on which the slider rides.
slidebarcolor	The color of the bar on which the fader rides.
slidebarradius	The radius in pixels of the bar on which the fader rides.
slidebardisplay	Determines whether the slider bar is displayed.

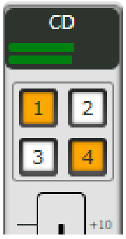
Additional **Default** Fader style properties:

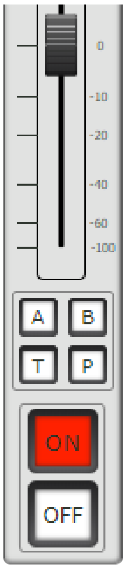
Property	Function
slider-top-color	Adjusts the color of the top edge of the default fad style. Due to opacity blending, this can be a subtl change.
slider-bottom-color	Adjusts the color of the bottom edge of the default fader style. Due to opacity blending, this can be a subtle change.
slider-default-line0-color	Adjusts the color of the first line on the default fad
slider-default-line1-color	Adjusts the color of the second line on the default fader.
slider-default-line2-color	Adjusts the color of the third line on the default fader.
slider-default-line3-color	Adjusts the color of the fourth line on the default fader.
slider-default-line4-color	Adjusts the color of the fifth line on the default fad
slider-default-line5-color	Adjusts the color of the sixth line on the default fader.
slider-default-line6-color	Adjusts the color of the seventh line on the defaul fader.

Additional Simple Fader style properties:

Property	Function
slider-simple-line-color	Adjusts the color of the line on the simple fader style.
slider-simple-line-display	Adjusts whether the line on the simple fader style displayed or not.

Console Fader





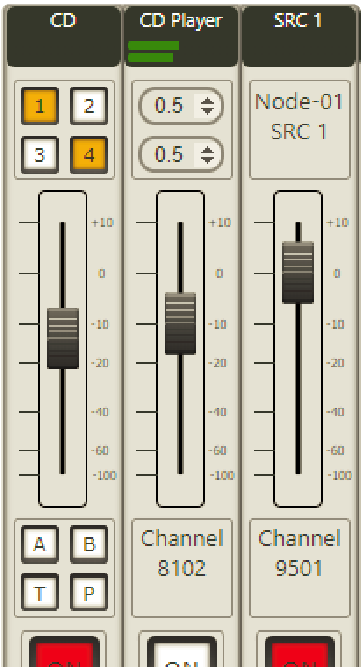
This component can be tied to Audio IOs that support gain manipulation. *See the **Setting the IO Property** topic in this chapter.*

The control displays the Fader value as you hover over the Fader. Fader values can be manipulated using the mouse wheel.

The **Console Fader** is a smart fader capable of displaying different properties depending on the type of IO assigned to it.

In the example above, this Console Fader has been assigned to a Fusion console input. When loaded, the component updates the controls according to the type of IO to which it has been assigned. In this case it shows the name of the **source profile**, a **meter** obtained from the input stage of the fader (*see note below*), the program **buss** assignments, a **fader** which maps to the Fusion fader, **A** and **B** user buttons, **T**alk and **P**review buttons, and **ON** and **OFF**.

This component will dynamically adjust what it displays depending on the type of IO it is associated with. Those changes will only be shown when the Panel is executed. That is when the capabilities of the assigned IO are analyzed. For example, after dragging a console fader onto the Panel and assigning IOs, when the Panel is executed you may see controls that look like:



In this case the first Fader is tied to the channel input of an Axia console. The second fader is tied to a VMIX channel input so the component changes to show the time up and down parameters. The third shows an XNode source.

> ⓘ Important Note: When the *control* property is used, metering will be deduced where possible unless the IO property is also filled in. If nothing is selected in the *control* property, the control point is inferred by the *IO* property where possible. For node Ios and Vmixers, the IO property is the recommended field to select. For console faders, the control property is recommended. Additionally, for iQx, Qor engine, and iQs style consoles, selecting a control point will use LwcpSs instead of Lwrp to obtain metering data.

Many of the properties discussed above in the **Fader** object have also been exposed in the **Console Fader** so that the color and style of the slider within the larger console fader object may be manipulated. These styles correspond directly to the styles.

In addition to standard CSS style attributes, customizable properties include:

Property	Function
IO	Used to select the Audio IO this meter will display
control	Adds support for LwCH options and Qor faders. *See the **Setting the Control Property** topic in thi chapter.*
fader-slider-background	This property changes the slider color. The proper accepts gradient values: **linear-gradient(#770000, #ff0000)** or solid colors: **#770000**
fader-slider-back-color	Primarily used for the simple and touch bar styles this will set the background color of the simple slider or of the touchbar. Clicking this field opens the standard color picker dialog. Additional color properties are available for these slider styles allowing for a great deal of granularity *See below for details*.
optimum	The gain level which is designed to be optimum o unity.
	Fader objects can have one of three different styles: **default**, **simple**, and **touchbar**.

faderstyle	- *Default* represents the same fader style as Pathfinder Core PRO has had previously - *Simple* turns the fader into a simple rectangle with rounded edges and an optional center lir - *Touchbar* is designed for touch interfaces. In this variation dragging with your finger or mouse anywhere in the fader rectangle will cause the bar level to go up or down
menu-enabled	For iQs, Qor, and iQx style consoles, this option defines whether the menu button will be enabled; the menu button may be assigned to the eq and dynamics section, but could also be used for your own custom purposes as well
meter-location	Whether the meter is located at the top of the fade in the label section or vertically along the right sid of the fader

Additional **Default** Fader style properties:

Property	Function
fader-slider-top-color	Adjusts the color of the top edge of the default fad style. Due to opacity blending, this can be a subtl change.
fader-slider-bottom-color	Adjusts the color of the bottom edge of the default fader style. Due to opacity blending, this can be a subtle change.
fader-slider-default-line0-color	Adjusts the color of the first line on the default fad
fader-slider-default-line1-color	Adjusts the color of the second line on the default fader.
fader-slider-default-line2-color	Adjusts the color of the third line on the default fader.
fader-slider-default-line3-color	Adjusts the color of the fourth line on the default fader.
fader-slider-default-line4-color	Adjusts the color of the fifth line on the default fad
fader-slider-default-line5-color	Adjusts the color of the sixth line on the default fader.
fader-slider-default-line6-color	Adjusts the color of the seventh line on the defaul fader.

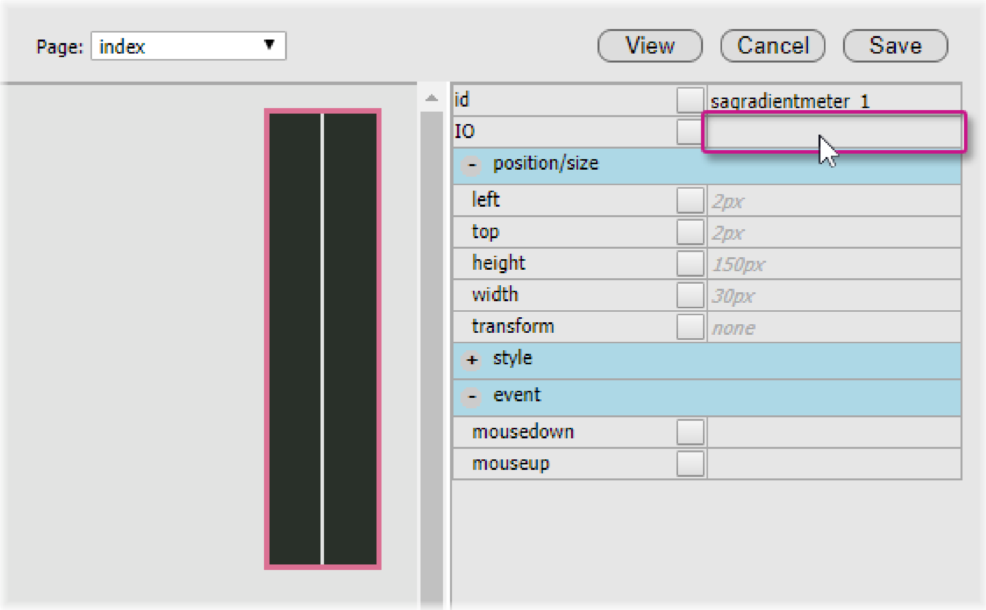
Additional Simple Fader style properties:

Property	Function
fader-slider-simple-line-color	Adjusts the color of the line on the simple fader style.
fader-slider-simple-line-display	Adjusts whether the line on the simple fader style displayed or not.

Setting the IO Property

Meters and Faders may be tied to an Audio IO using the component's *IO* property. For example:

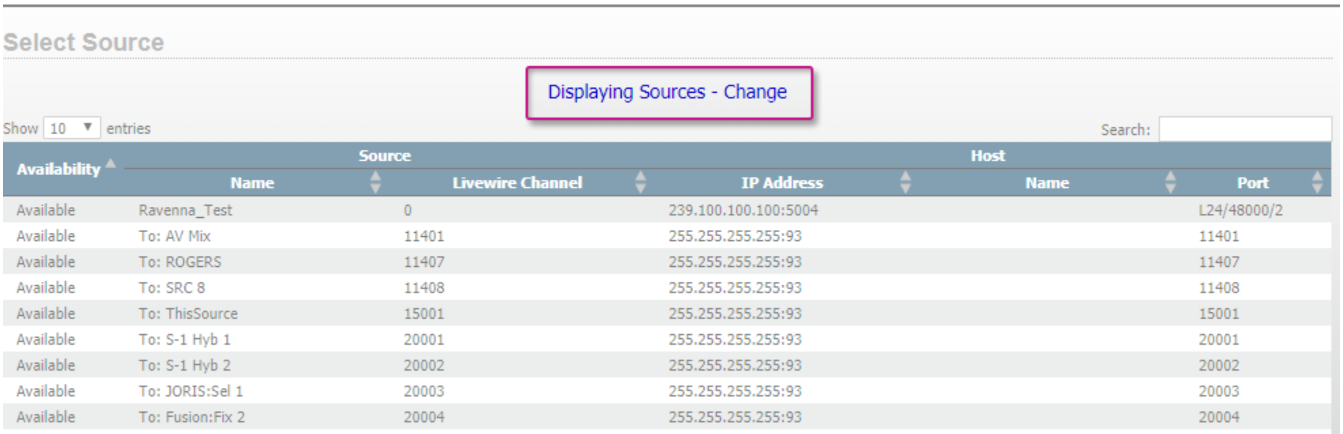
- Drag a **Gradient Meter** into a User Panel and resize it to an appropriate size
- Click the **IO** property to open the IO *Select Source* dialog box



Page: index

id	sagradientmeter_1
IO	
- position/size	
left	2px
top	2px
height	150px
width	30px
transform	none
+ style	
- event	
mousedown	
mouseup	

- The link at the top of the page will display whether *sources* or *destinations* are currently being displayed. Clicking the **link** will toggle between the two



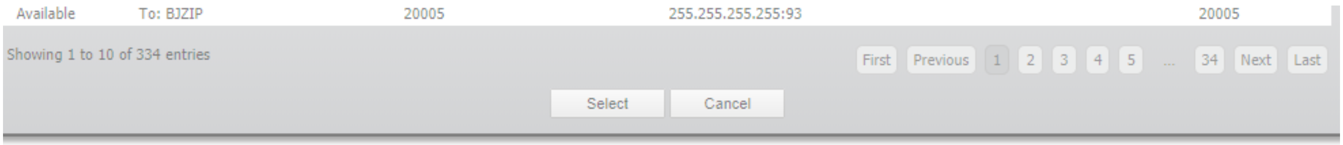
Select Source

Displaying Sources - Change

Show 10 entries

Search:

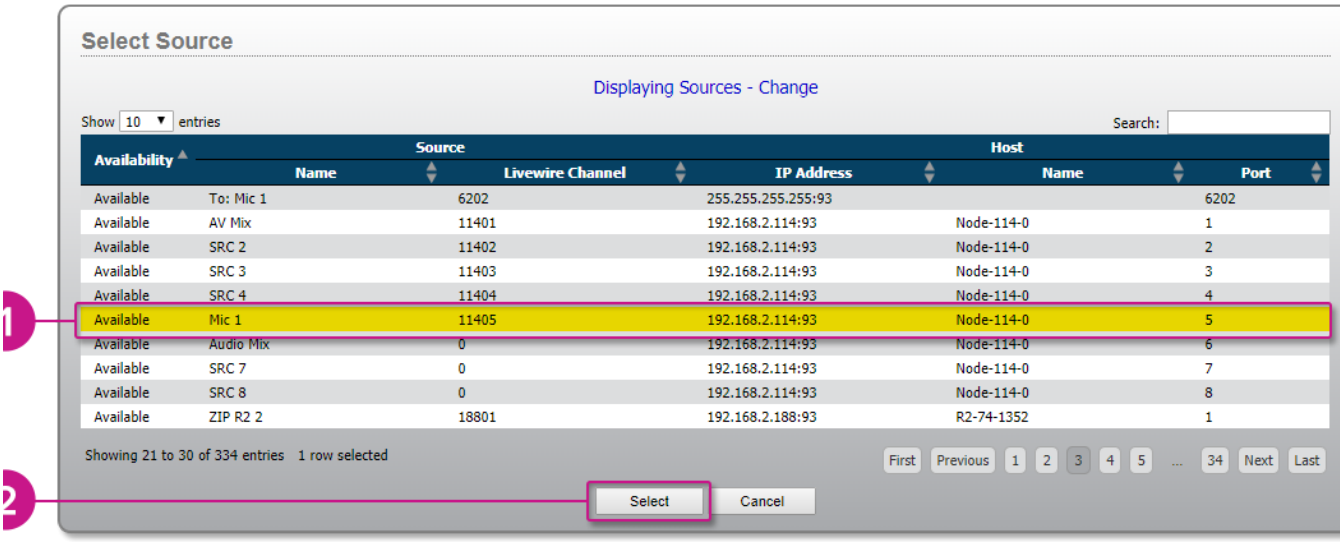
Availability	Source		Host		
	Name	Livewire Channel	IP Address	Name	Port
Available	Ravenna_Test	0	239.100.100.100:5004		L24/48000/2
Available	To: AV Mix	11401	255.255.255.255:93		11401
Available	To: ROGERS	11407	255.255.255.255:93		11407
Available	To: SRC 8	11408	255.255.255.255:93		11408
Available	To: ThisSource	15001	255.255.255.255:93		15001
Available	To: S-1 Hyb 1	20001	255.255.255.255:93		20001
Available	To: S-1 Hyb 2	20002	255.255.255.255:93		20002
Available	To: JORIS:Sel 1	20003	255.255.255.255:93		20003
Available	To: Fusion:Fix 2	20004	255.255.255.255:93		20004



Available	To: BJZIP	20005	255.255.255.255:93	20005

Showing 1 to 10 of 334 entries

- Click an **IO** whose metering you want to be tied to the meter and click **select**



Select Source

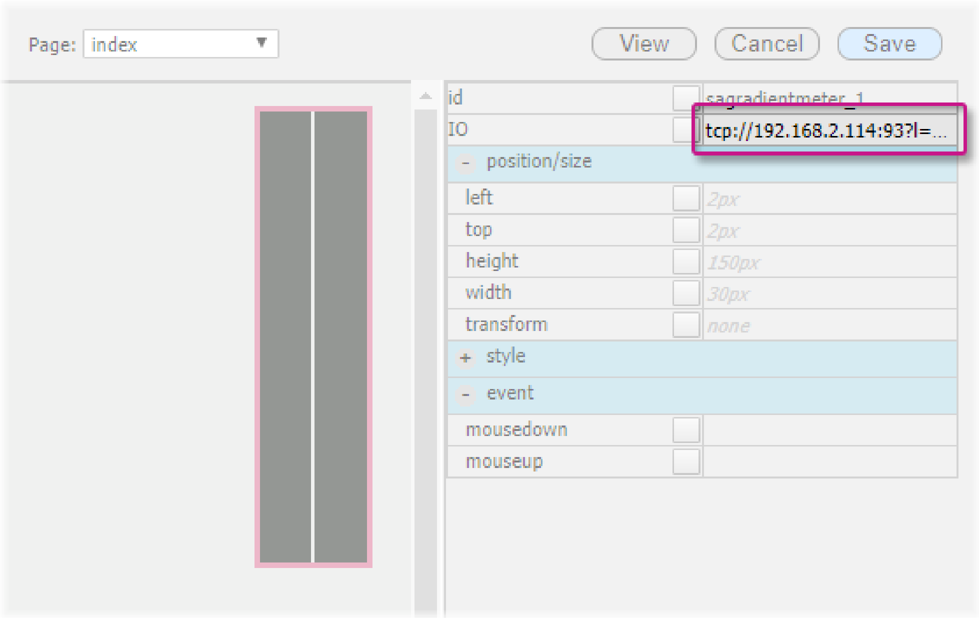
Displaying Sources - Change

Show 10 entries Search:

Availability	Source			Host	
	Name	Livewire Channel	IP Address	Name	Port
Available	To: Mic 1	6202	255.255.255.255:93		6202
Available	AV Mix	11401	192.168.2.114:93	Node-114-0	1
Available	SRC 2	11402	192.168.2.114:93	Node-114-0	2
Available	SRC 3	11403	192.168.2.114:93	Node-114-0	3
Available	SRC 4	11404	192.168.2.114:93	Node-114-0	4
Available	Mic 1	11405	192.168.2.114:93	Node-114-0	5
Available	Audio Mix	0	192.168.2.114:93	Node-114-0	6
Available	SRC 7	0	192.168.2.114:93	Node-114-0	7
Available	SRC 8	0	192.168.2.114:93	Node-114-0	8
Available	ZIP R2 2	18801	192.168.2.188:93	R2-74-1352	1

Showing 21 to 30 of 334 entries 1 row selected

- You can use the **Search** box to narrow down the list of available sources or destinations
- The IO field will fill with the path of the selected IO



Page: index

id: sagradientmeter_1
IO: tcp://192.168.2.114:93?l=...
- position/size
left: 2px
top: 2px
height: 150px
width: 30px
transform: none
+ style
- event
mousedown
mouseup

Since we are in the designer, no metering will be displayed. It will only display when the Panel is executed.

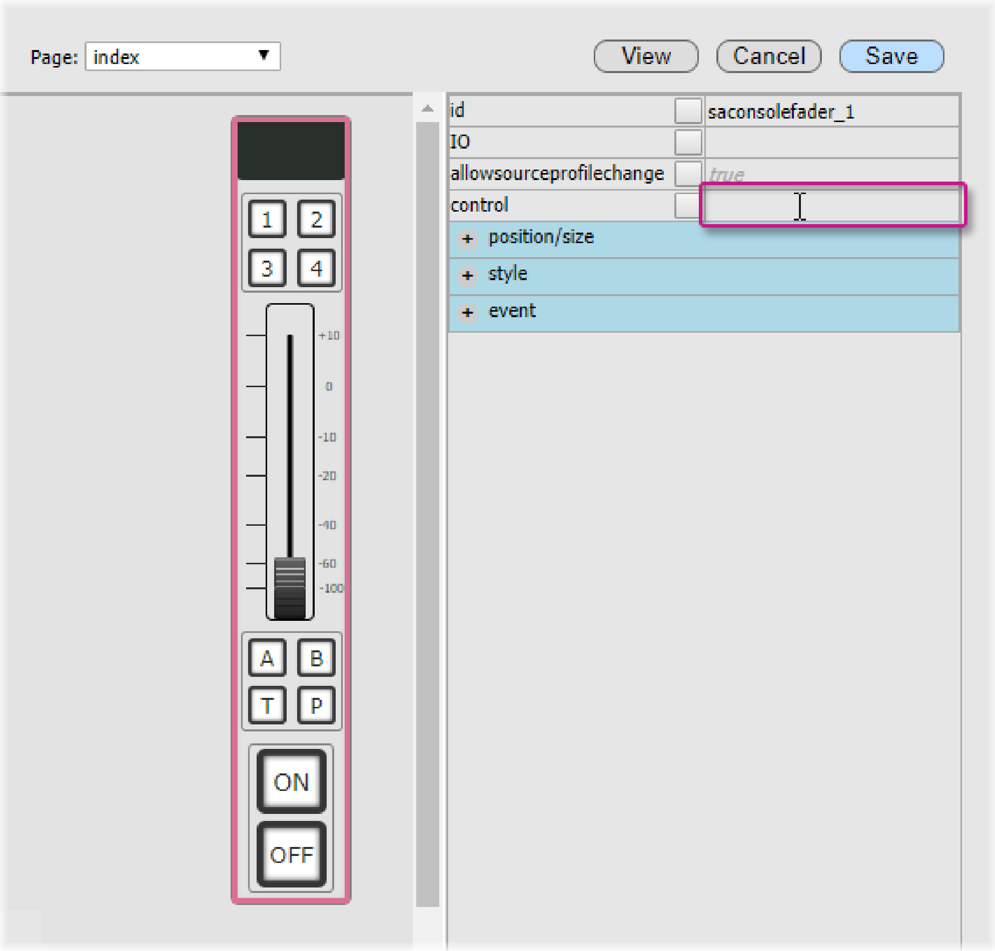
> ⓘ Important Note: Enabling the binding button for the IO parameter will make it available to Logic Flows, allowing you to dynamically change the IO assigned to the meter or fader.

Setting the Control Property

Console Faders may also be assigned to their control and metering points using the control property. This is recommended for control over physical mixer faders. It will intelligently deduce metering and control locations according to the console type and fader type selected.

To enable the *control* property of an existing Fader or Console Fader control:

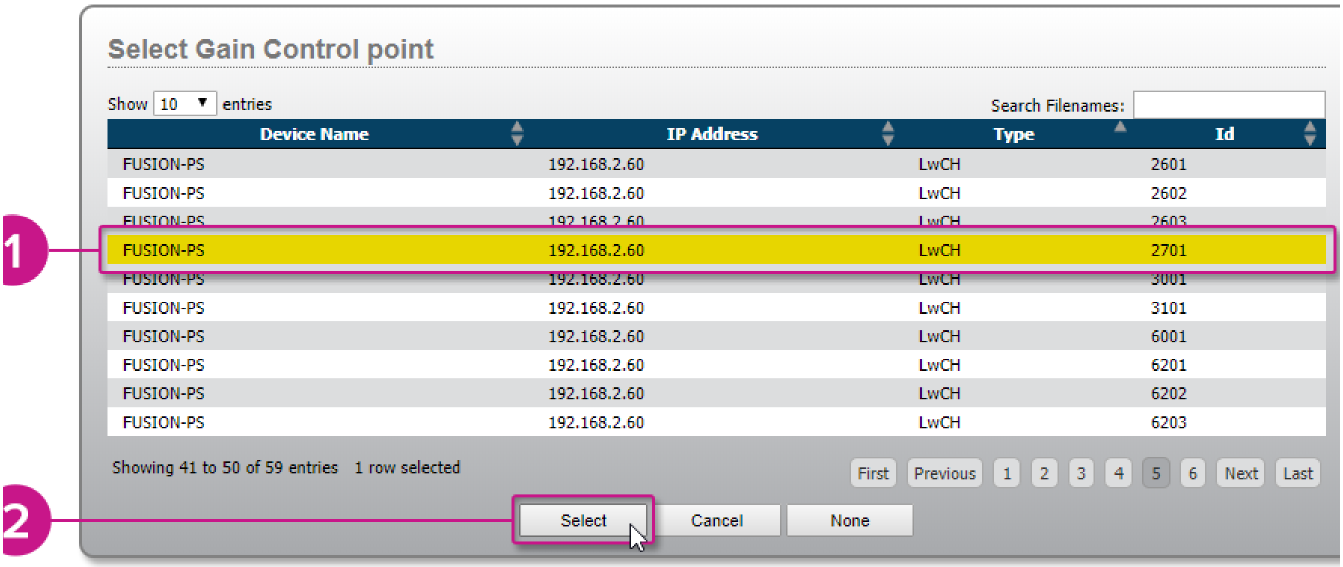
- With the Fader or Console Fader control selected, click the control field in the Property window



Page: index

id	saconsolefader_1
IO	
allowsourceprofilechange	true
control	
+ position/size	
+ style	
+ event	

- In the Select Gain Control Point dialog, click on the **device** to control and click **Select**



Select Gain Control point

Show 10 entries

Search Filenames:

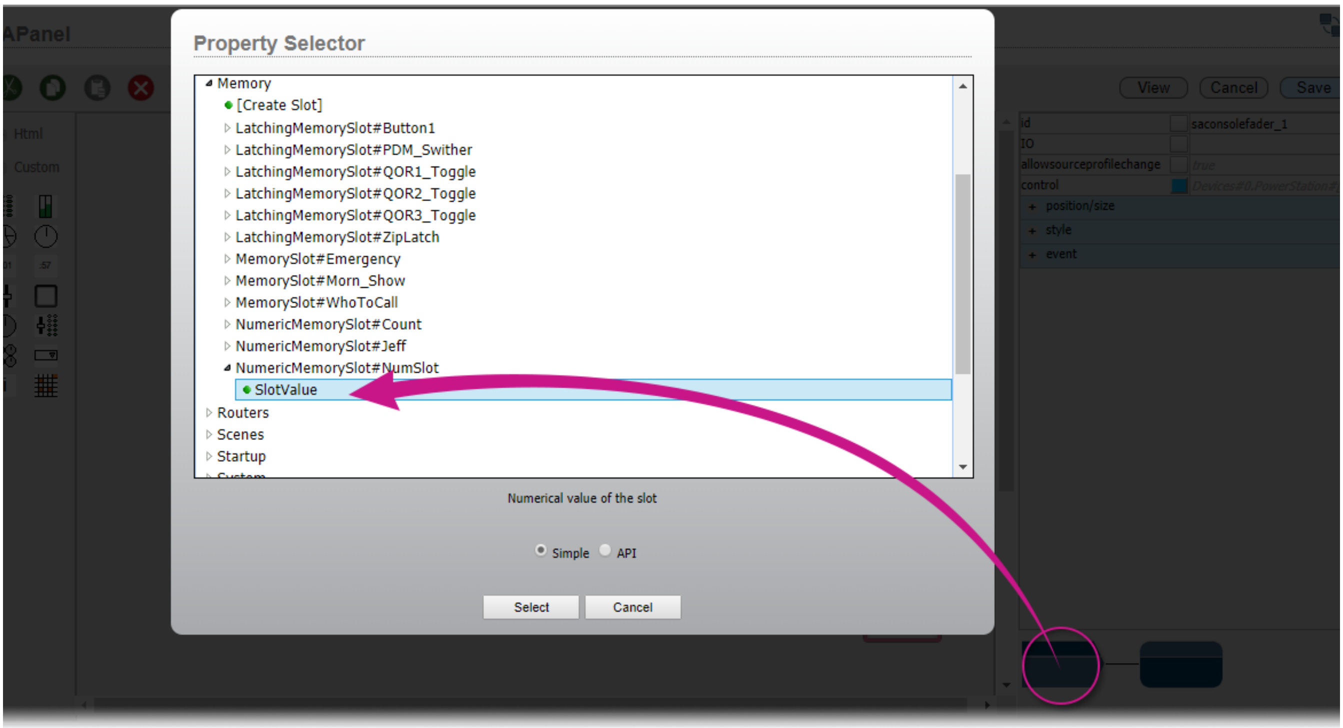
Device Name	IP Address	Type	Id
FUSION-PS	192.168.2.60	LwCH	2601
FUSION-PS	192.168.2.60	LwCH	2602
FUSION-PS	192.168.2.60	LwCH	2603
FUSION-PS	192.168.2.60	LwCH	2701
FUSION-PS	192.168.2.60	LwCH	3001
FUSION-PS	192.168.2.60	LwCH	3101
FUSION-PS	192.168.2.60	LwCH	6001
FUSION-PS	192.168.2.60	LwCH	6201
FUSION-PS	192.168.2.60	LwCH	6202
FUSION-PS	192.168.2.60	LwCH	6203

Showing 41 to 50 of 59 entries 1 row selected

Controlling Numeric Properties

If the fader is a standard fader rather than a console fader, you can alternatively set it to control any numeric property in the system.

- After selecting the fader and the control field, Click the Endpoints button to change the Select Gain control Point dialog to the Property selection tree. From there you can connect to any numeric property in the tree



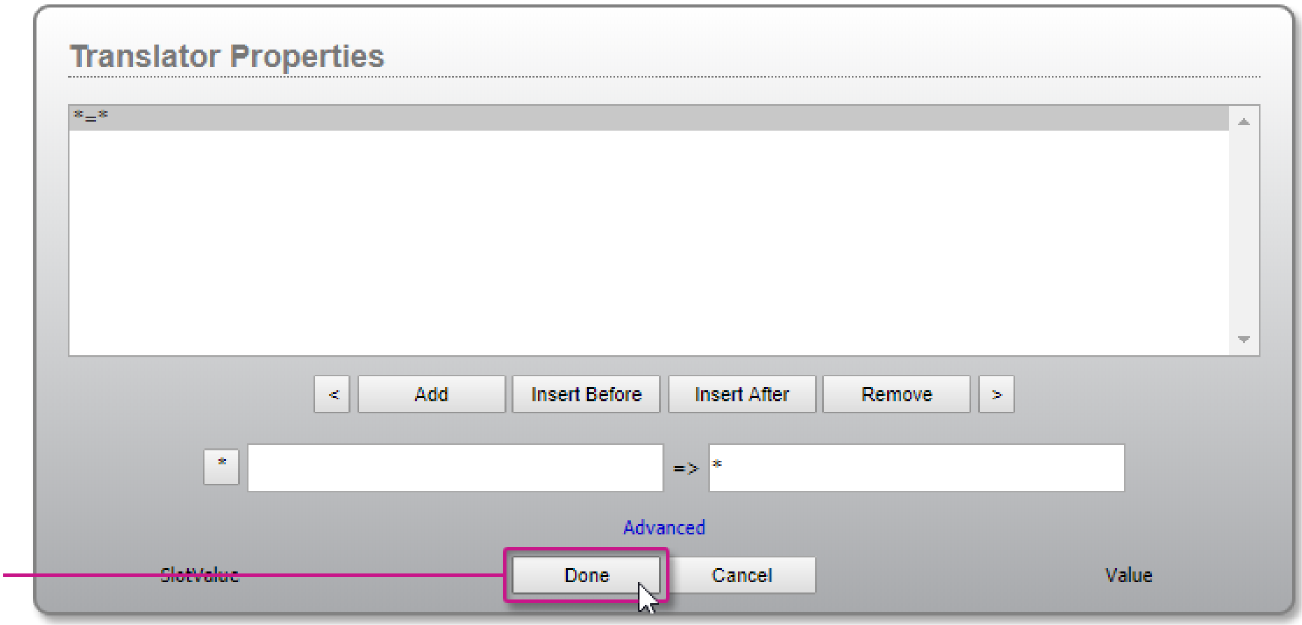
Property Selector

```
◢ Memory
  ● [Create Slot]
  ▷ LatchingMemorySlot#Button1
  ▷ LatchingMemorySlot#PDM_Swither
  ▷ LatchingMemorySlot#QOR1_Toggle
  ▷ LatchingMemorySlot#QOR2_Toggle
  ▷ LatchingMemorySlot#QOR3_Toggle
  ▷ LatchingMemorySlot#ZipLatch
  ▷ MemorySlot#Emergency
  ▷ MemorySlot#Morn_Show
  ▷ MemorySlot#WhoToCall
  ▷ NumericMemorySlot#Count
  ▷ NumericMemorySlot#Jeff
  ◢ NumericMemorySlot#NumSlot
    ● SlotValue
▷ Routers
▷ Scenes
▷ Startup
▷ System
```

Numerical value of the slot

○ Simple ○ API

- If you select a non-numeric property, the system will give you a warning. Avoid using non-numeric properties with the *control* property. *See **Controlling Non-Numeric Properties** below*
- Accept the default Translator properties



Translator Properties

```
*=*
```

* [ ] => [*]

- Save your Panel. The Fader will now control and update based on the value of the memory slot

For this to work properly there are several other properties that can be used:

Property	Notes
min	The minimum value allowable by the Fader.
max	The maximum value allowable by the Fader.
step	The steps that can be used for the change. A valu of 1 would mean the value has to be integers whereas a step value of .1 would allow for a singl decimal place in the values.
Type	This property has two options: Audio: the scale of the fader will follow a typical audio fader where larger moves in the optimal range of the fader rela to smaller decibel changes. Linear: the scale is a direct linear scale.

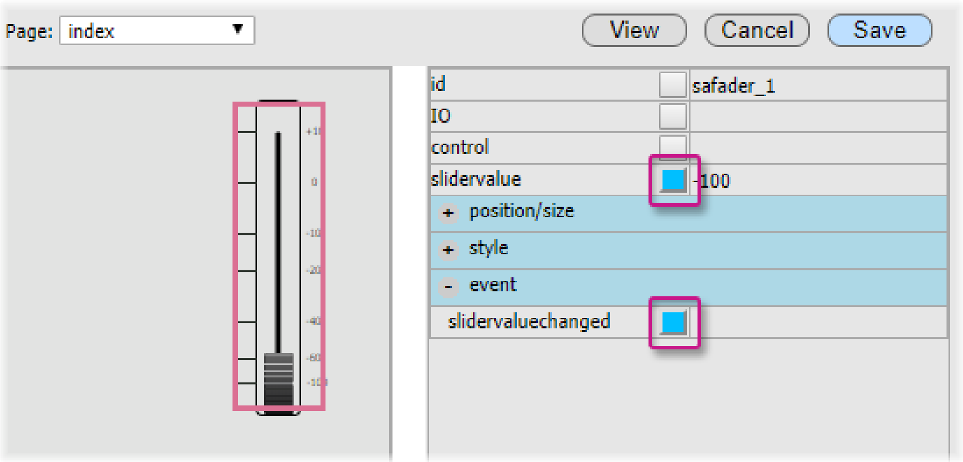
> Important Note: When using the *control* property to select the functionality for the fader, the system will try to automatically set the *min*, max, and *step* properties if the system knows what they should be for the given control point.

Controlling Non-Numeric Properties

Another option is not to use the *control* property to bind the functionality and instead to bind the *displayvalue* and *valuechange* properties directly to properties. This can be useful in situations where a translation is necessary.

For example, we could connect a Fader to a *true/false* or GPIO value.

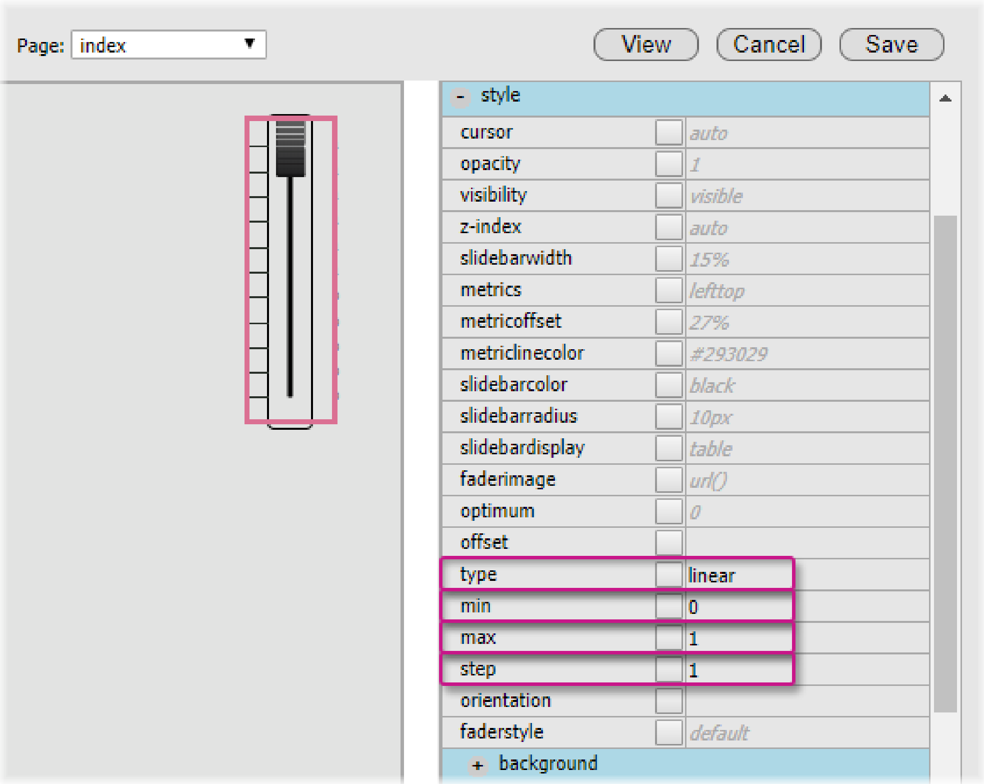
- In this case we do not use the *control* property but would instead enable the binding on the *slidervalue* property and the *slidervaluechanged* event.



Page: index

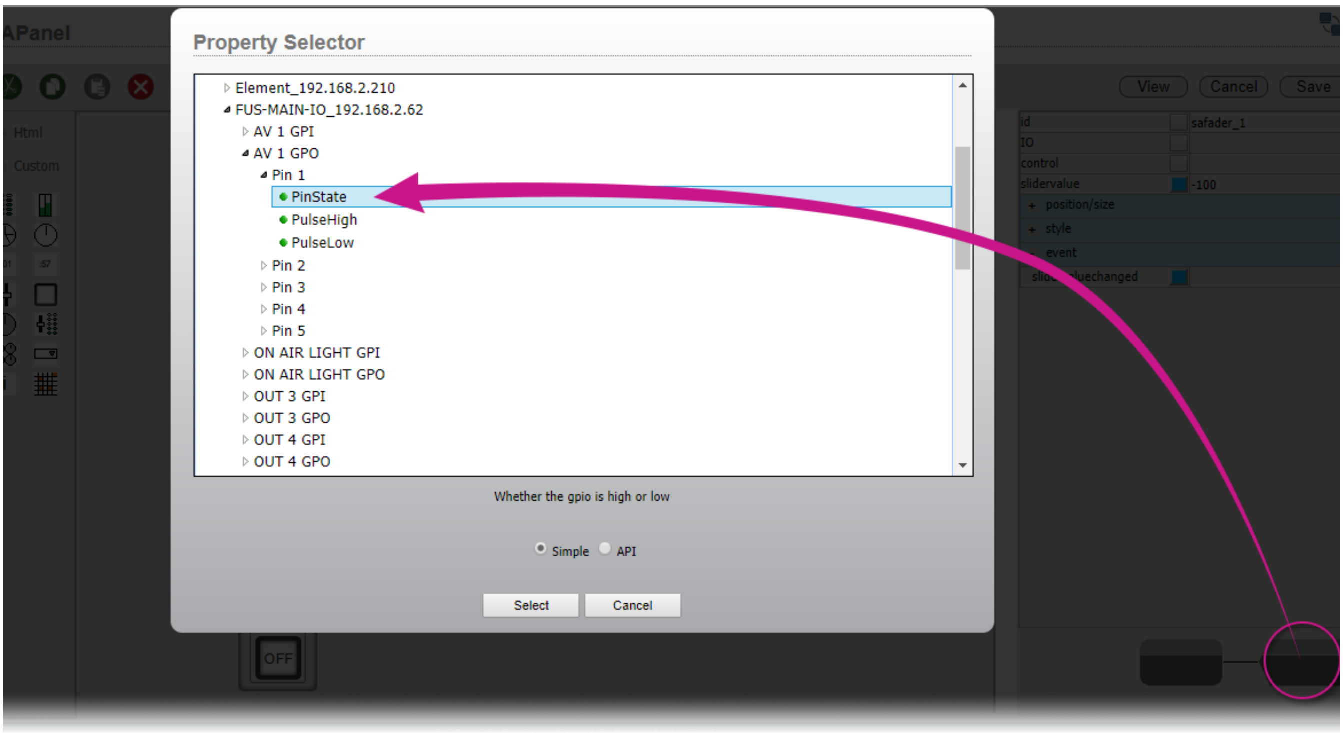
id	□	safader_1
IO	□	
control	□	
slidervalue	■	-100
+ position/size		
+ style		
- event		
slidervaluechanged	■	

- Set the *min* property to **0**, the *max* property to **1**, the *step* property to **1**, and *type* to **linear**



cursor	auto
opacity	1
visibility	visible
z-index	auto
slidebarwidth	15%
metrics	lefttop
metricoffset	27%
metriclinecolor	#293029
slidebarcolor	black
slidebarradius	10px
slidebardisplay	table
faderimage	url()
optimum	0
offset	
type	linear
min	0
max	1
step	1
orientation	
faderstyle	default

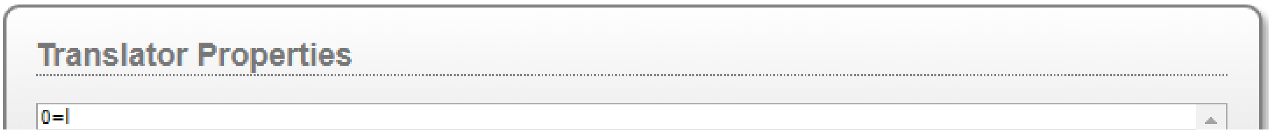
- Set the *slidervaluechanged* event to the GPO *pinstate* of a GPIO port



Whether the gpio is high or low

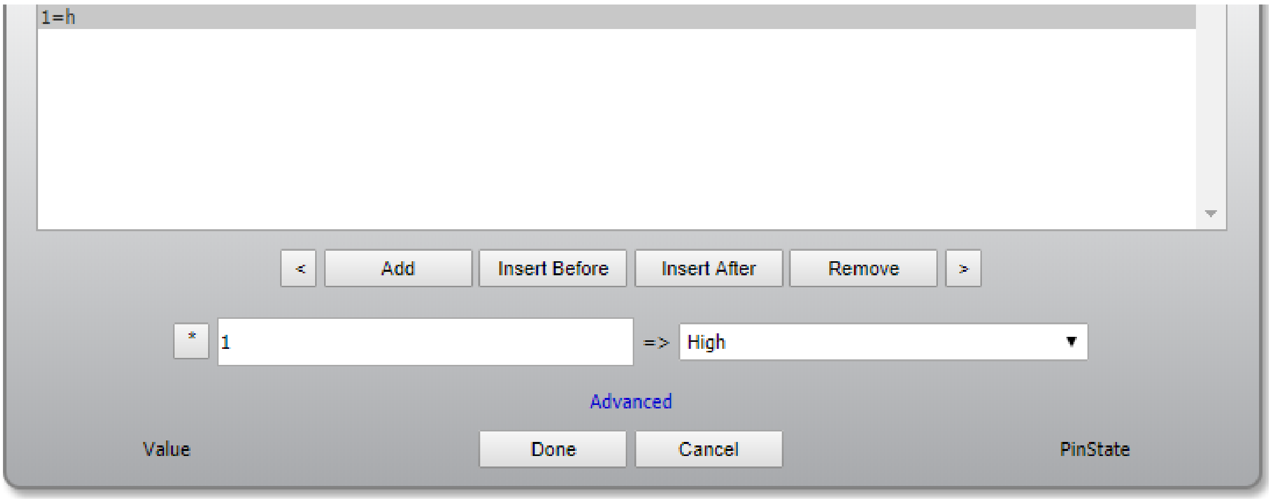
id	safader_1
IO	
control	
slidervalue	-100
slidervaluechanged	

- For the translation, set **0** to *Low* and **1** to *High*



Translator Properties

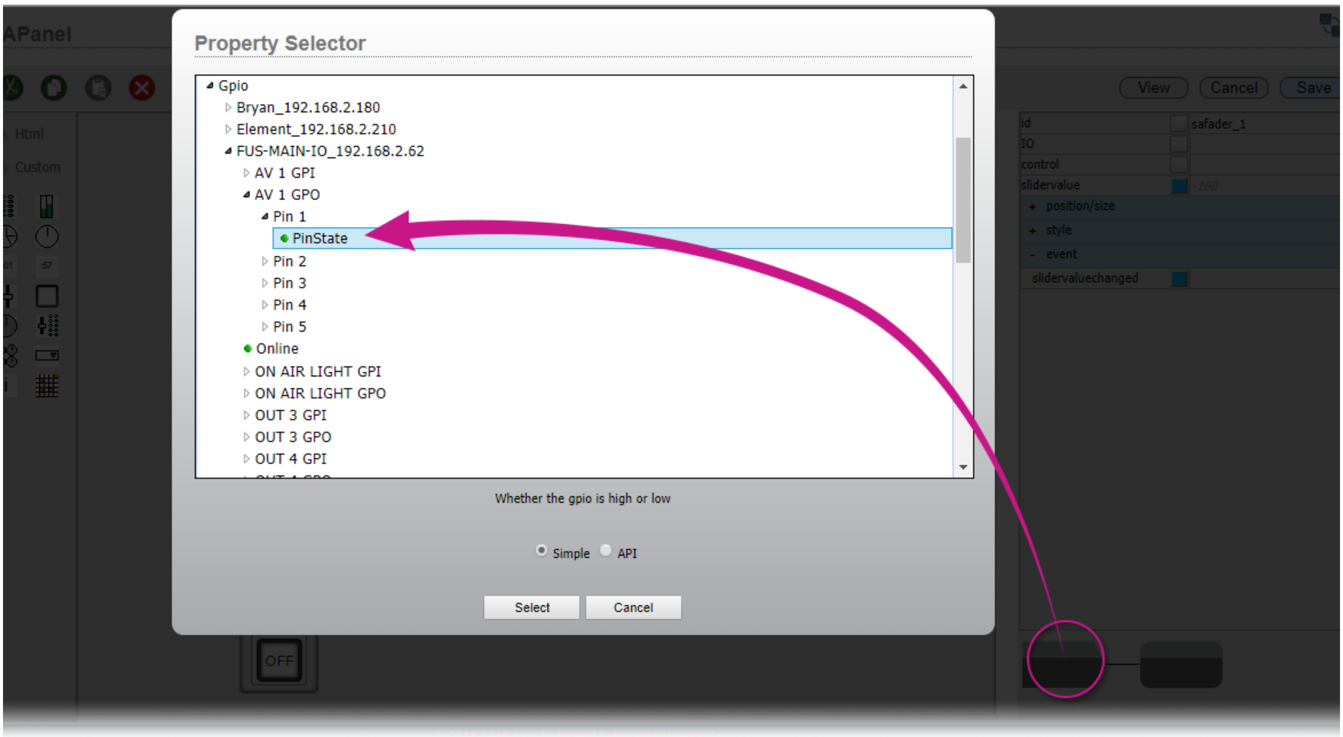
0=l



1=h

* 1 => High

- Set the *slidervalue* property to the same GPO *pinstate* as the *slidervaluechanged* event



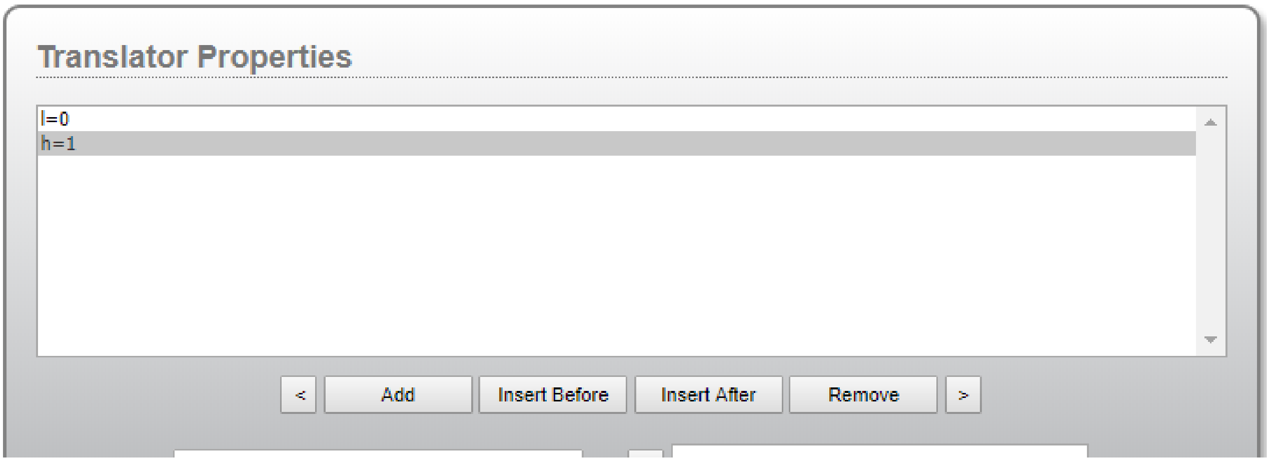
Property Selector

```
◢ Gpio
  ▷ Bryan_192.168.2.180
  ▷ Element_192.168.2.210
  ◢ FUS-MAIN-IO_192.168.2.62
    ▷ AV 1 GPI
    ◢ AV 1 GPO
      ◢ Pin 1
        ● PinState
      ▷ Pin 2
      ▷ Pin 3
      ▷ Pin 4
      ▷ Pin 5
    ● Online
    ▷ ON AIR LIGHT GPI
    ▷ ON AIR LIGHT GPO
    ▷ OUT 3 GPI
    ▷ OUT 3 GPO
    ▷ OUT 4 GPI
```

Whether the gpio is high or low

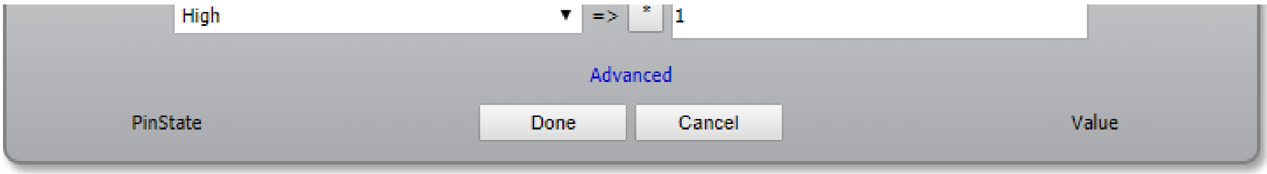
● Simple ○ API

- Reverse the translation. That will make sure that if something else changes the GPIO pin that it will be updated in the fader:



Translator Properties

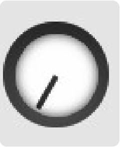
l=0
h=1



For this example, you may also want to turn the metrics off since there are only two valid states.

This is probably not the best use case for this functionality, but it shows how you can use the Fader control to manipulate any value in the system either directly using the control property if it is a numeric property or via translation and the *slidervalue* and *slidervaluechanged* property and event.

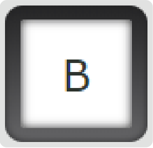
Console Knob



The rotary **Console Knob** may be used in the same way as the Fader above. The same properties for numeric and non-numeric control as well as min, max, step, etc. properties also apply to the *console knob* object.

This control can also have its *control* property used to select a console fader or other numeric property in the system to control in the same manner as described for the fader control above. In addition, this component also shares the indicator property with button objects so that the knob color may be changed based on some indication state.

Console Button



The **Console Button** component can control and indicate changes in the system. This button works the same as the HTML button but has a more interesting look.

In addition to standard CSS style attributes, customizable properties include:

Property	Function
saconsolebutton-caption	Updates the text displayed on the button.
hwmap	Used to select a hardware LCD button. This hardware LCD button will then mirror the behavio of the software button.
indicator	Used to set the button indication to On, Off, or Flash. The colors used for On, Off, and the two flash colors are the backcoloroff and on propertie

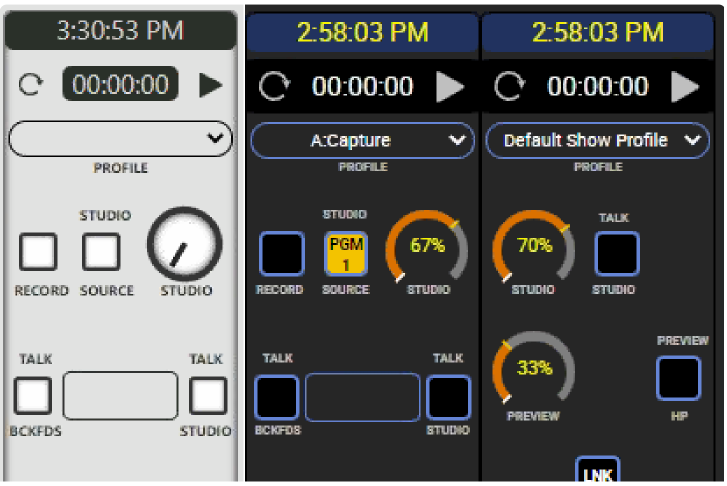
backccoloron	Color used when the indicator state is on.
backccoloroff	Color used when the indicator state is off.
saconsolebutton-image	Used instead of the standard CSS style background image to update an image inside the button. Because this component is built from several embedded HTML objects, this makes sure the correct inner component displays the image.
border-gradient	Options include *complex* and *simple*. A value of *simple* will remove the background top-to-bottom gradient. No value or a value of *complex* produces a button centered in a background. But as you resize this can cause the button to be off by a pixel one way the other. At smaller sizes this can become an objectionable artifact. The simple design is just a button with a border and therefore the border scales more perfectly. *Complex* gradients have a slightly softer feel while the *simple* gradients scale better.

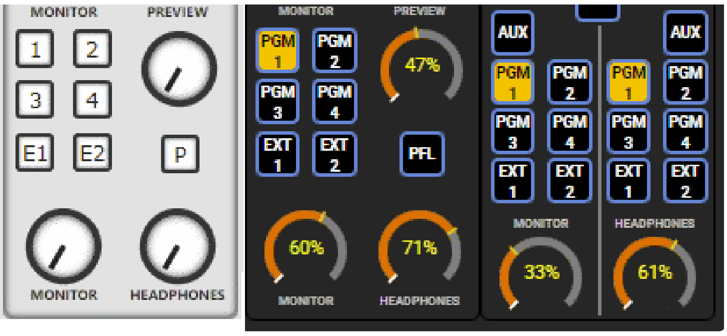
Console Monitor Section

> ⓘ Note: Fusion consoles require version 3.2.1.28 of the Fusion software in order to properly use the monitoring section with these consoles. Please click here for a document on obtaining and installing updated Fusion software.

The **Console Monitor Section** acts as a monitor section for consoles. It is important to note that the clock and timer functionality are controlled by Pathfinder Core PRO and not the console and so will not be in sync with the console timer and countdown clock. This is because those control points are not currently available in the control protocols.

Dragging this component onto a Panel will present a monitor section for a console:

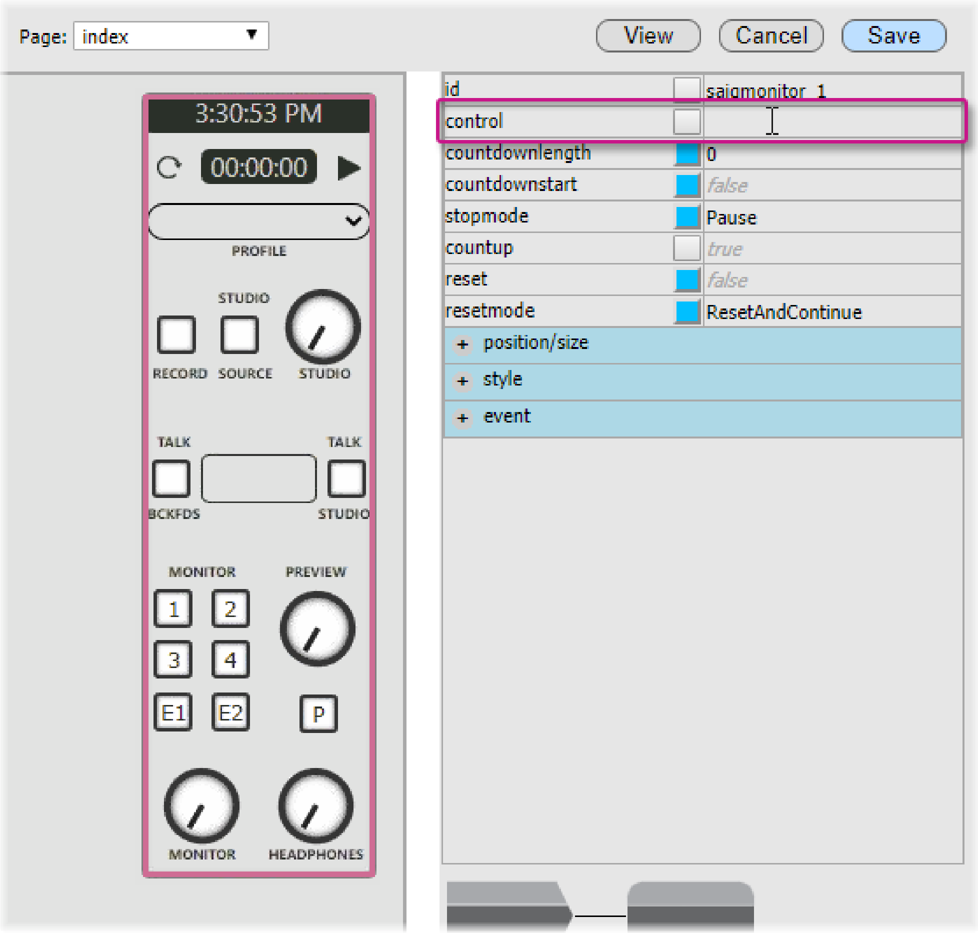




Examples of Monitor Sections

The picture above shows the monitor section in both default and dark themes as well as differences when attached to different types of consoles. The component will change its control layout when executing depending on the type of console it is attached to. For example, the monitor section shown in the center in the picture above is attached to an iQx, whereas the one on the right is attached to a Powerstation with Fusion console.

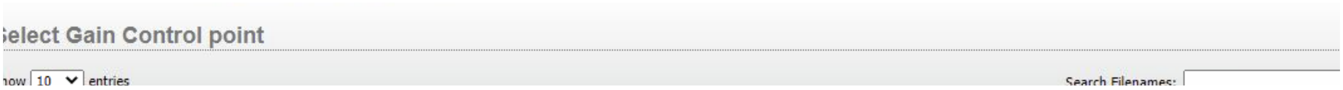
- To assign the component to a specific console, select the component in the Panel and click the *control* property



id	saigmonitor_1
control	
countdownlength	0
countdownstart	*false*
stopmode	Pause
countup	*true*
reset	*false*
resetmode	ResetAndContinue

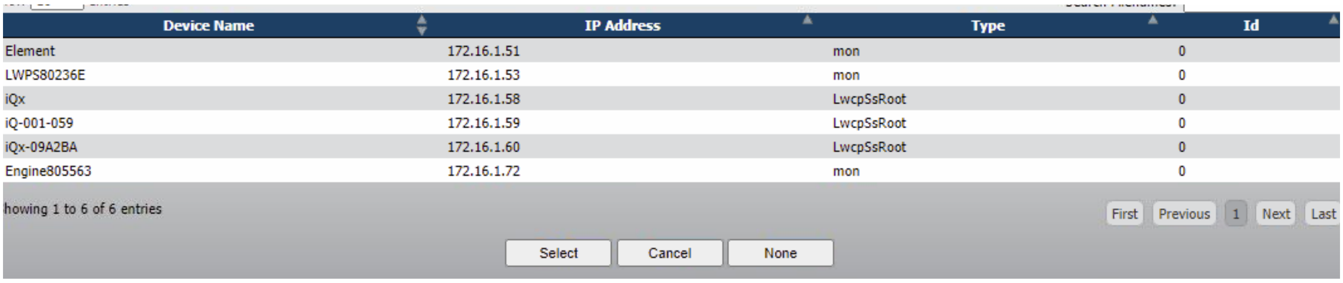
+ position/size
+ style
+ event

- In the list of available control points, you will find one for each Console; select the console you would like to associate with the monitor section


Select Gain Control point

how 10 entries

Search Filenames:

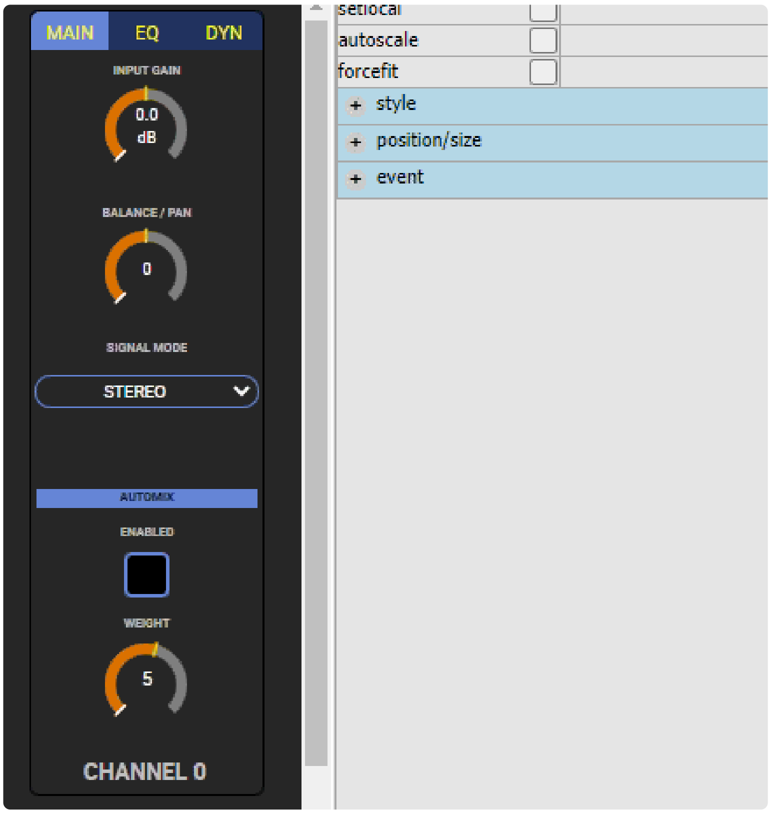


Device Name	IP Address	Type	Id
Element	172.16.1.51	mon	0
LWPS80236E	172.16.1.53	mon	0
iQx	172.16.1.58	LwcpSsRoot	0
iQ-001-059	172.16.1.59	LwcpSsRoot	0
iQx-09A2BA	172.16.1.60	LwcpSsRoot	0
Engine805563	172.16.1.72	mon	0

howing 1 to 6 of 6 entries

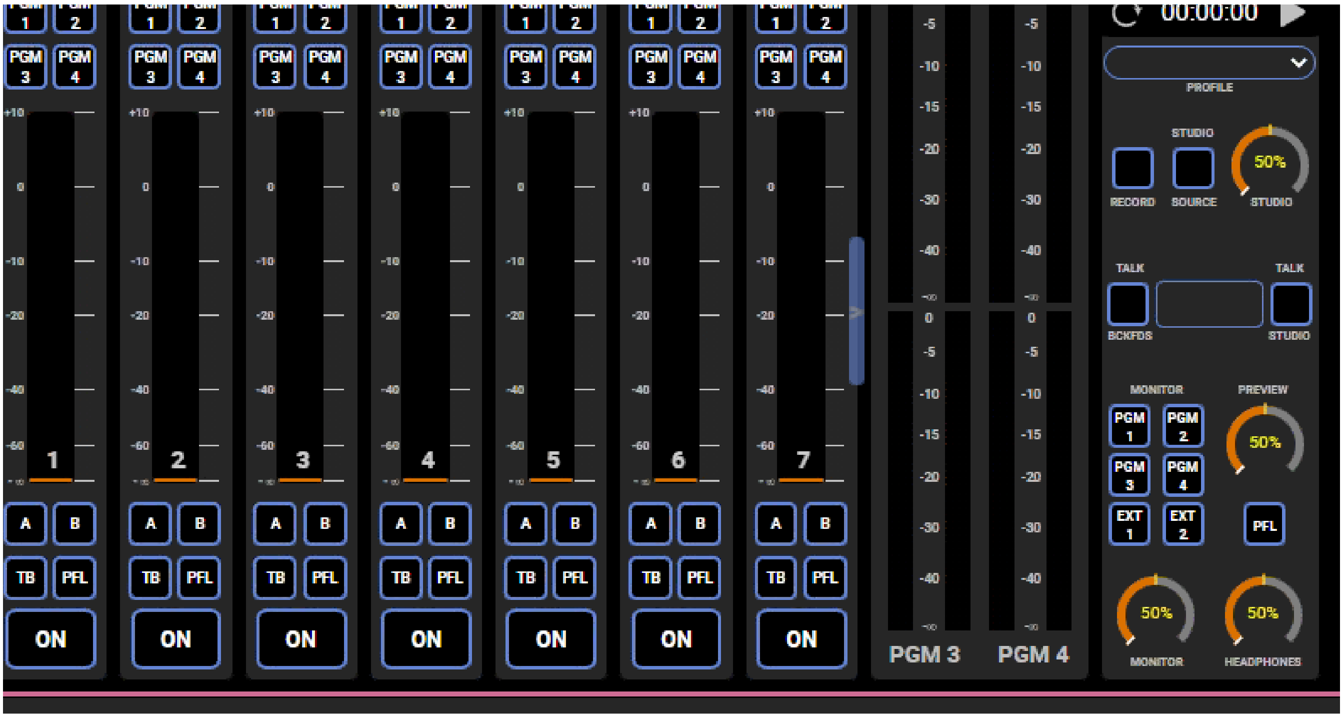
ⓘ Important Note: The Fusion and Powerstation require version 3.2.1.28 of the Fusion software in order to properly use the monitoring section with these consoles.

iQ Fader Processing



The iQ fader processing is a component that accesses the eq and dynamics section of Qor, iQx, and iQs faders. In order to use it, select the console from the control property and then assign a number 1 through 24 to the fader-number property to use it with a specific fader. This assignment can be automated using the menu button in the console fader object. Turn the binding of the fader-number property on. Then within each fader in the panel, use the menu assign and menu unassign event to assign the correct fader number to the component. A fader number of 0 can be used to make the component not have any effect. It is a good idea to hook the menu indicator to the eq/dynamics section fader-assign property as well so the menu button will only light if the current fader is selected in the component.

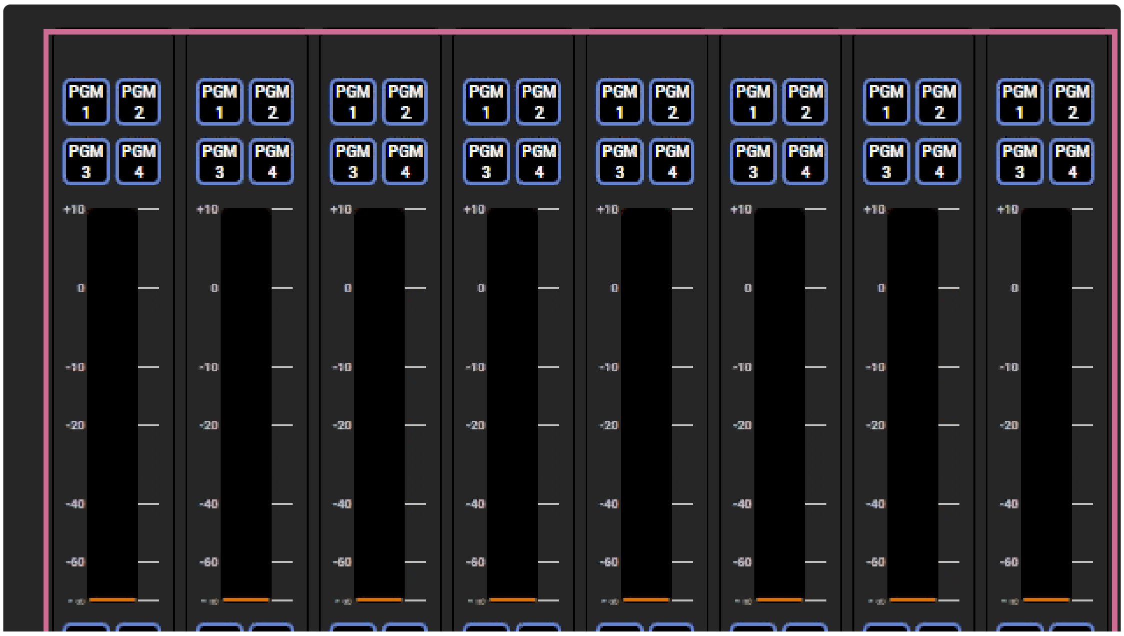
Console

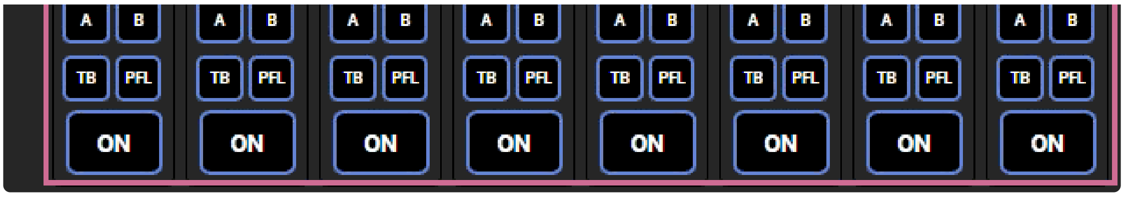


The console component can be used to create a control mechanism for a full console. After dropping this component onto a panel, use the control property to select from a list of consoles. When the panel with the component executes, it will automatically display and layout the controls according to the type of console it has been attached to. Increasing or decreasing the height of the component will adjust the height and size of the components within the console. Changing the width of the component will present or hide paging buttons for the fader list depending on how many faders the console has. The fader count is not correctly displayed in the designer and is fixed to 8. The fader list is only populated with the actual fader count when the console is initialized in a running panel.

> ⓘ Important Note: The Fusion and Powerstation require version 3.2.1.28 of the fusion software in order to properly use the monitoring section with this console component.

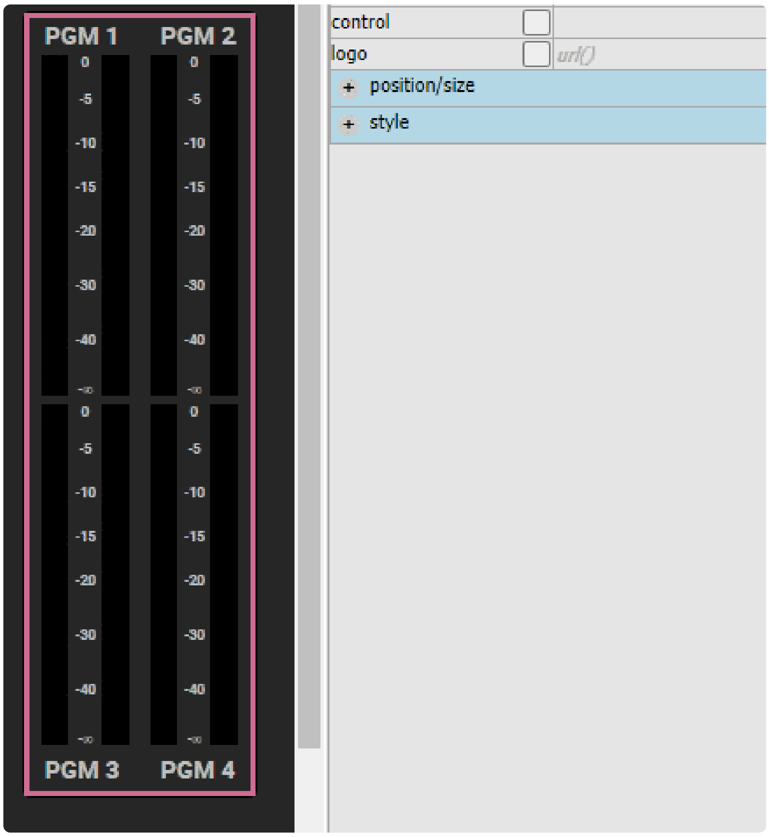
Console Fader Dock





This component will present a list of consoles in the control property in the designer. When executing the panel connected to a console, the dock will auto-fill with all of the faders the console has. It will present page left and right buttons automatically if there are more faders than can be displayed in width designed in the console component.

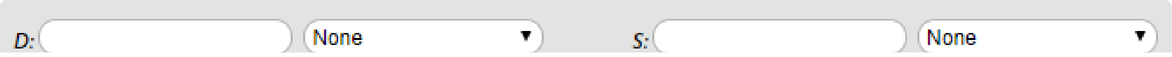
Console Meters

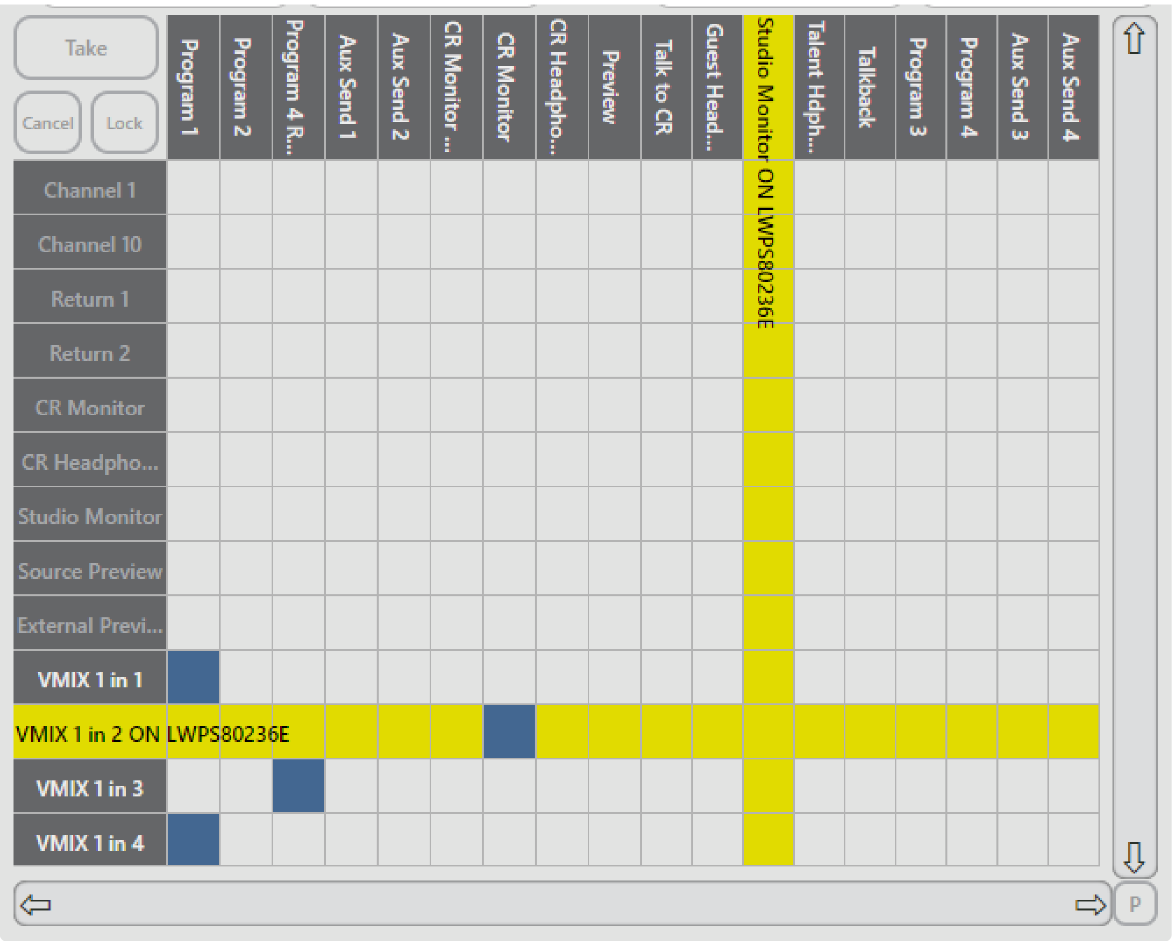


The console meters section presents metering for program busses 1 through 4. It will present a list of consoles through the control property in the designer that will allow you to map the control to a specific console. Like the updated console fader, it will automatically figure out the correct metering paths (lwrp/lwcpss) from the console it is attached to it. In addition this module has a logo property which will allow you to upload a logo image which will sit between the top and bottom meters. The size ratio of the logo must be designed in order to display correctly. The height should be approximately one fifth of the width. The example svg we have used had dimensions of 6774x1452 and is then scaled up and down accordingly. This may take some experimentation to get correct.

Router XY Matrix

This version adds a Router XY Matrix component both as a User Panel component and as an additional tab on the *Router-Details* page. It can be added to a Panel using the **Router XY Matrix** component.

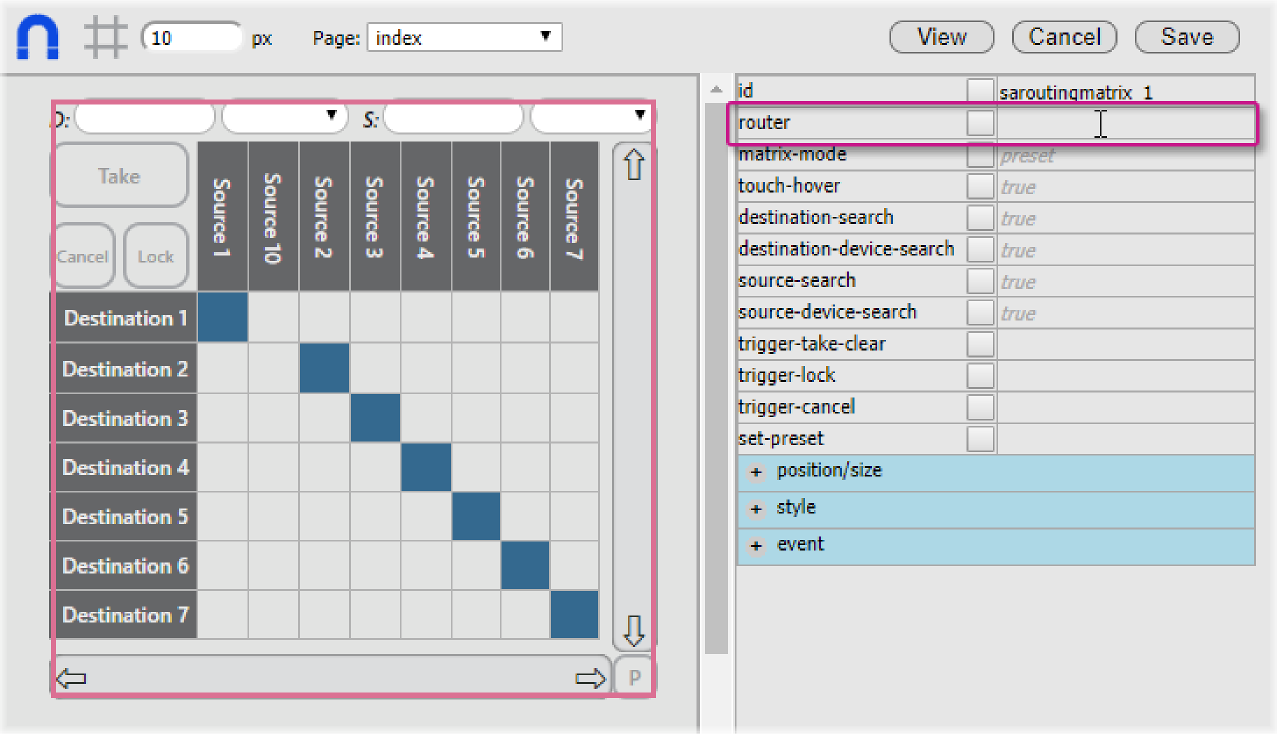




	Program 1	Program 2	Program 4 R...	Aux Send 1	Aux Send 2	CR Monitor ...	CR Monitor	CR Headpho...	Preview	Talk to CR	Guest Head...	Studio Monitor ON LWPS80236E	Talent Hdph...	Talkback	Program 3	Program 4	Aux Send 3	Aux Send 4
Channel 1																		
Channel 10																		
Return 1																		
Return 2																		
CR Monitor																		
CR Headpho...																		
Studio Monitor																		
Source Preview																		
External Previ...																		
VMIX 1 in 1	■																	
VMIX 1 in 2 ON LWPS80236E							■											
VMIX 1 in 3			■															
VMIX 1 in 4	■																	

After adding the component to the page, resize it and use the select a router to use with it.

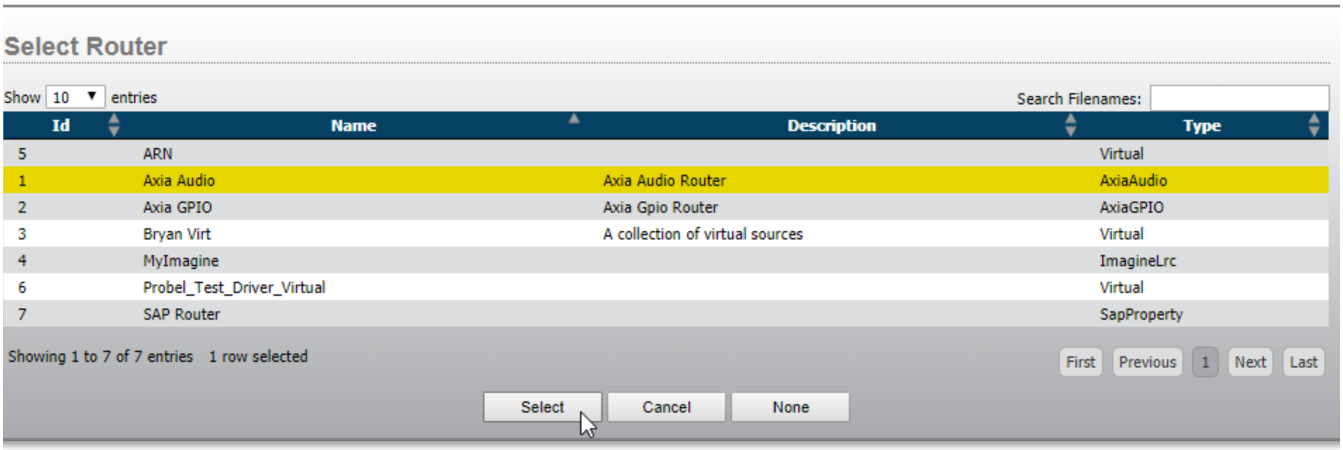
- With the component selected, click the **router** property



	Source 1	Source 10	Source 2	Source 3	Source 4	Source 5	Source 6	Source 7
Destination 1	■							
Destination 2			■					
Destination 3				■				
Destination 4					■			
Destination 5						■		
Destination 6							■	
Destination 7								■

id	□	saroutingmatrix_1
router	□	
matrix-mode	□	preset
touch-hover	□	true
destination-search	□	true
destination-device-search	□	true
source-search	□	true
source-device-search	□	true
trigger-take-clear	□	
trigger-lock	□	
trigger-cancel	□	
set-preset	□	
+ position/size		
+ style		
+ event		

- Click the desired router in the list and click Select



Id	Name	Description	Type
5	ARN		Virtual
1	Axia Audio	Axia Audio Router	AxiaAudio
2	Axia GPIO	Axia Gpio Router	AxiaGPIO
3	Bryan Virt	A collection of virtual sources	Virtual
4	MyImagine		ImagineLrc
6	Probel_Test_Driver_Virtual		Virtual
7	SAP Router		SapProperty

Showing 1 to 7 of 7 entries 1 row selected

The component will not populate with actual router data while viewed in the Panel Designer. This means that the actual column and row header size may be inflated since the default designer example population only has ten sources and destinations. It is useful to view it in the actual executing Panel to see how it will function.

Action Properties

Property	Function
trigger-take-clear	This property may be used by a Logic Flow and/o binding to remotely press the take/clear button. This will only have an effect if there is a cross poi preset for action.
trigger-lock	This property may be used by a Logic Flow and/o binding to remotely press the lock button. This wil only have an effect if there is a cross point preset for action.
trigger-cancel	This property may be used by a Logic Flow and/o binding to remotely press the cancel button. This will only have an effect if there is a cross point preset for action.
set-preset	This property allows a Logic Flow to preset a cros point. The syntax involves a string with the source and destination path with the X between the two. For example: tcp://172.16.1.97:93?l=SRC&d=src&i=2&t=aaudic X tcp://1 72.16.1.72:93? l=DST&d=dst&i=31&t=aaudio

Events

Property	Function
hover-source-name	Event raises the name of the source for the cross-point that the mouse is currently hovering over.
hover-source-description	Event raises the description of the source for the cross-point that the mouse is currently hovering over.
hover-source-path	Event raises the path of the source for the cross-point that the mouse is currently hovering over.
hover-destination-name	Event raises the name of the destination for the cross-point that the mouse is currently hovering over.
hover-destination-description	Event raises the description of the destination for the cross-point that the mouse is currently hoverin over.
hover-destination-path	Event raises the path of the destination for the cross-point that the mouse is currently hovering over.
preset-source-name	Event raises the name of the source for the cross-point that has been preset.
preset-source-description	Event raises the description of the source for the cross-point that has been preset.
preset-source-path	Event raises the path of the source for the cross-point that has been preset.
preset-destination-name	Event raises the name of the destination for the cross-point that has been preset.
preset-destination-description	Event raises the description of the destination for the cross-point that has been preset.
preset-destination-path	Event raises the path of the destination for the cross-point that has been preset.

One of the interesting things you can do with the *hover* and *preset* events is to bind the path to the IO field of meters. This allows you to create a Panel where hovering over a cross-point also shows metering for that cross-point. Note that metering does not yet support IO binding to virtual router IOs.

Style Properties

Property	Function

routerpath	Used to select the router which will be used with the matrix.
matrix-mode	There are three ways you can configure the XY matrix to work:
touch-hover	When this option is set to true and the matrix is being used with a touch screen, the first tap will generate the same hover lines as you see when hovering a mouse over the cross-point. The secor touch will preset. If this property is false, the first touch will both set the hover lines and set the preset.
destination-search, destination-device-search, source-search, source-device-search	Each of these corresponds to whether that particular search field appears and is available in the search bar. Setting all of them to false will hide the search bar. This is particularly useful for small matrices where searching is not necessary. Pleas note that even if the device selection drop-down fields are set to true, they will still not appear for virtual routers. They will only appear if used with a audio or GPIO router. Virtual routers will display th textual search fields for both source and destinatio if they are enabled in this property.
fixed-col-row-length	This defines the length of the row and column headers in pixels.
row-height	Defines the height of the rows and columns in the grid. It is important to note that the grid will always be filled. Therefore, if there are not enough entries to fill the grid, these parameters may be larger tha this height. Also, if you reduce this parameter you may find the grid extending off the bottom of the component. To correct this, reduce the font-size as well until it fits again. The sizing of items in the gri is a delicate balance so if you start altering these parameters from the defaults, you may have to tweak to get it to display correctly. The height is intentionally large enough for touch use.
route-engaged-color	This is the color (blue by default) of cross- points where a route exists.
route-preset-color	This is this the color (dark red by default) of cross- points that have been preset for an action.
route-hover-color	This is the color of the hover bars that show the ro and column over which the mouse is hovered.

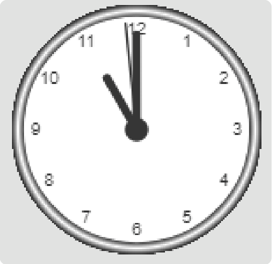
grid-line-color	This is the color of the lines in the grid.
disabled-button-text-color	This is the color of the text in disabled action buttons.
enabled-button-text-color	This is the color of the text in enabled action buttons. This property is no longer used as the tex color for enabled uses the specific button’s enable color below.
disabled-button-background-color	This is the background color of disabled buttons.
enabled-button-background-color	This is the background color of enabled buttons. This is overridden by hover effects.
locked-destination-background-color	This is the color of cross points that are locked.
locked-destination-text-color	This is the color of the text in the row headers for destinations that are locked.
matrix-scroll-bar-height	This is the height of the horizontal scroll bar and is also used as the width of the vertical scroll bar. Scroll bars will automatically disappear if all cross points can fit on the matrix.
matrix-search-bar-height	This is the height of the search bar. Please note that the correct way to hide the search bar is to disable the search field properties mentioned earlier in this property list.
scroll-bar-color	This is the color of the active part of the scroll bars

Using the XY Matrix

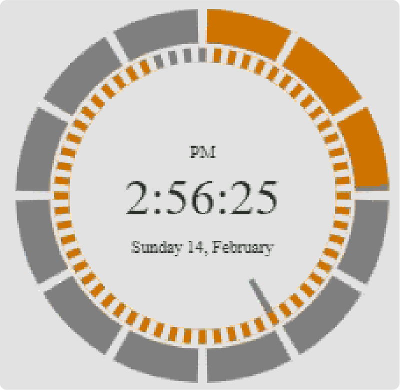
Once a matrix has been added to a Panel, a router assigned to it, and the Panel saved, view the Panel to see how it functions.

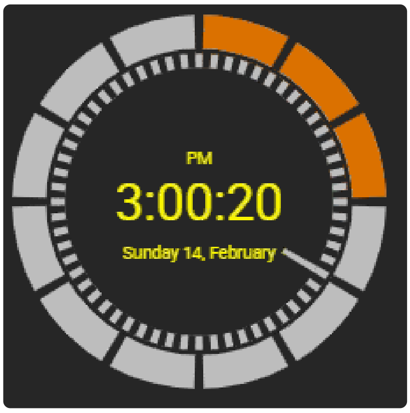
Since this component mirrors the functionality of the XY matrix presented on the XY tab in Routers, refer to the Routers chapter for details on how the end-user can use the routing matrix once it is deployed on an HTML5 User Panel.

Analog Clock



This component displays the current time using an analog clock style. The style may be changed using the clockstyle property between classic (shown above) and quasar (shown below):



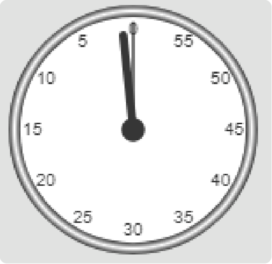


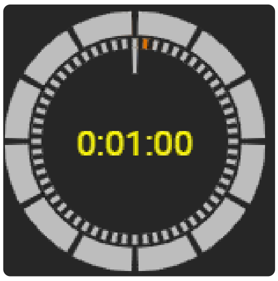
The color palette in the default theme is designed for the default theme. In addition, a variety of properties may be modified when the clock is in the quasar style in either default or dark theme so that you can adjust the colors to whatever you desire.

- Under background

    - quasar-clock-hours-color: changes the color of the background outside circle.
    - quasar-clock-elapsed-hours-color: changes the color of the elapsed hours in the outside circle.
    - quasar-clock-minutes-color: changes the color of the background inside circle.
    - quasar-clock-elapsed-minutes-color: changes the color of the elapsed minutes in the inside circle.
    - quasar-clock-marker-color: changes the color of the gaps between the mark points in the hours and minutes circle.
    - quasar-clock-second-hand-color: changes the color of the second hand.
- Under font/text
    - quasar-clock-ampm-color: changes the color of the am/pm text in the clock.
    - quasar-clock-time-color: changes the color of the time text in the clock.
    - quasar-clock-date-color: changes the color of the date text in the clock.

Analog Countdown





This component allows you to define and trigger a countdown. The clock will display the countdown value.

Custom properties include:

Property	Function
countdownlength	The time in seconds for the countdown.
countdownstart	Generally exposed via bindings for a Logic Flow t trigger the start of a countdown. Options are *true* c *false*.
Elapsed Event	This event can be raised when the countdown timer completes.
	This property defines what happens when *countdownstart* is set to false while the countdowr

StopMode	is progressing. The options are: - **StopAndReset**: stopping the timer by setting *countdownstart* to *false* will cause the timer to stop and be reset to the countdown value - **Pause**: stopping the timer by setting *countdownstart* to *false* will cause the timer to stop where it is in the countdown process and hold that value; setting *countdownstart* to *true* again will cause the countdown to continue from where it left off
CountUp	Changing this value to *True* instead of *False* will cause the timer to count up rather than down to th selected countdownlength. If countdownlength is zero it will count up indefinitely until stopped and/ reset.
Reset	This is an action property that can be used by Log Flows and/or bindings to reset the counter. Setting it to *true* will cause the timer to reset.
ResetMode	This property defines what happens when a reset is issued and a countdown is in progress. The options are: - **ResetAndStop**: Setting reset to true will caus the timer to reset and stop its countdown - **ResetAndContinue**: If the timer is running thi will cause it to reset and continue running. If th timer is stopped this will just reset the value
ClockStyle	Changes the analog clock style of the countdown clock between classic and quasar in the same wa as for the Analog Clock above

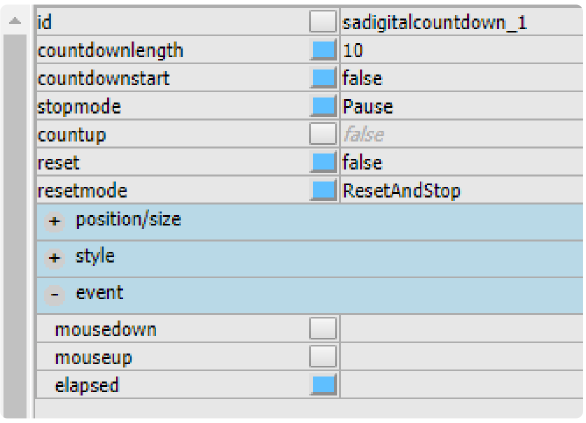
Additionally, this component also has the same color styling properties when used in quasar mode as listed in the analog clock above.

Currently, the timer does not automatically reset when the countdown completes. If that functionality is desired, it can be obtained by using the property bindings to hook the elapsed event to the reset state such that when the timer elapses, reset is triggered. This can be accomplished either by using the Flow in *elapsed* to change the reset or by using the Flow in *reset* to change based on the elapsed start point.

In both cases, the Panel must be saved with the binding buttons for these properties/events turned on to surface those options in the Flow property tree.

For example:

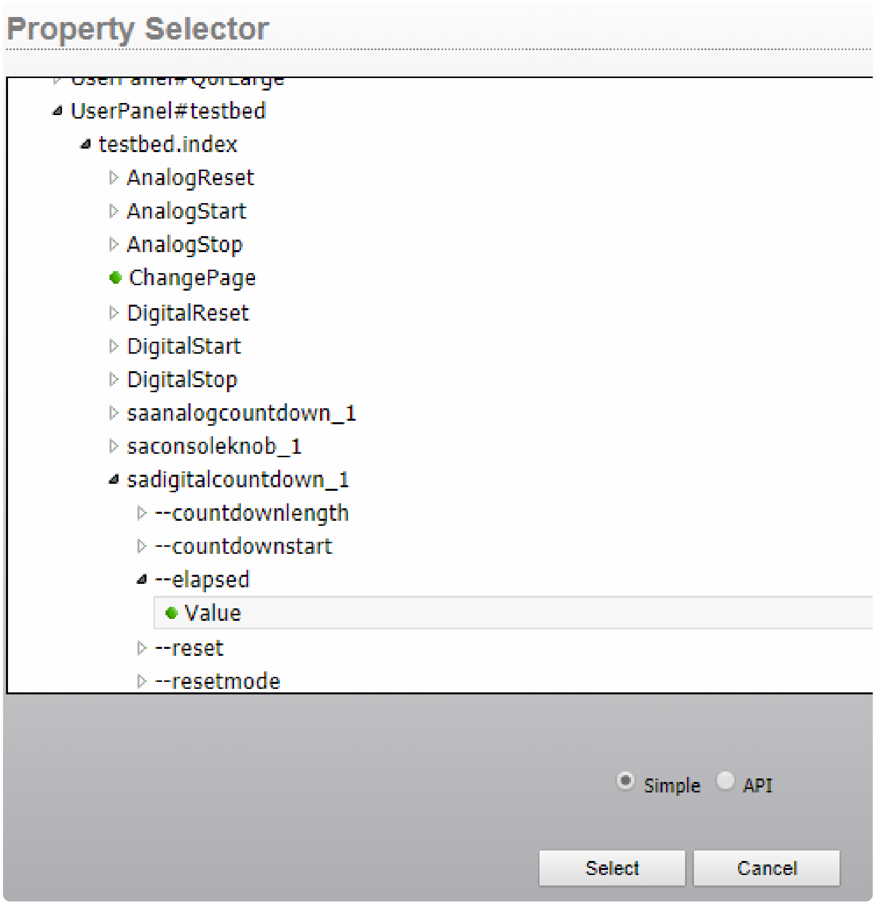
- Enable the binding buttons for both *elapsed* and *reset* have been enabled and save your Panel



id	□	sadigitalcountdown_1
countdownlength	■	10
countdownstart	■	false
stopmode	■	Pause
countup	□	*false*
reset	■	false
resetmode	■	ResetAndStop
+ position/size		
+ style		
- event		
mousedown	□	
mouseup	□	
elapsed	■	

- Select the *reset* property field and click the start point in the bottom corner

- Browse to the User Panels group, locate the digital countdown control in your specific Panel, and select the *elapsed* **Value** property

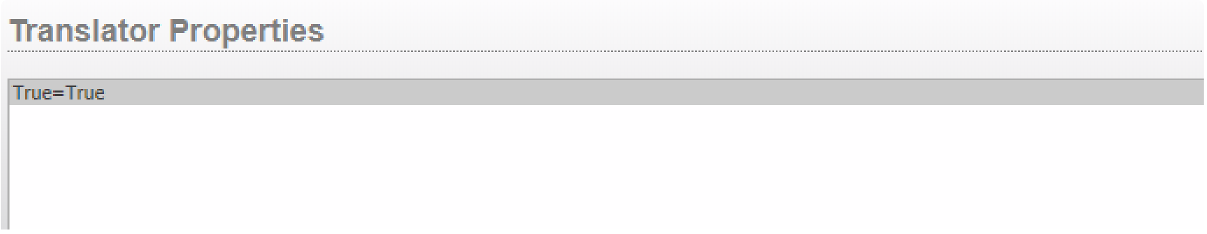


Property Selector

```
▷ UserPanel#QorLarge
◢ UserPanel#testbed
    ◢ testbed.index
        ▷ AnalogReset
        ▷ AnalogStart
        ▷ AnalogStop
        ● ChangePage
        ▷ DigitalReset
        ▷ DigitalStart
        ▷ DigitalStop
        ▷ saanalogcountdown_1
        ▷ saconsoleknob_1
        ◢ sadigitalcountdown_1
            ▷ --countdownlength
            ▷ --countdownstart
            ◢ --elapsed
                ● Value
            ▷ --reset
            ▷ --resetmode
```

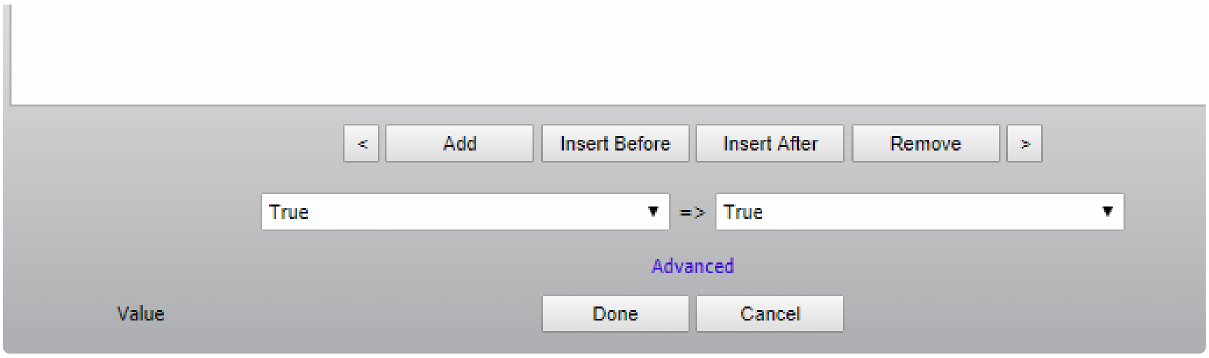
● Simple ○ API

- Select **True=True** for the translation



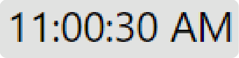
Translator Properties

True=True



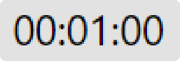
- Click **Done**. After saving the Panel if the countdown elapses the timer should automatically reset

Digital Clock



This component displays the current time using a digital clock style.

Digital Countdown



This component allows you to define and trigger a countdown. The clock will display the countdown value.

Custom properties include:

Property	Function
countdownlength	The time in seconds for the countdown.
countdownstart	Generally exposed via bindings for a Logic Flow t trigger the start of a countdown. Options are *true* c *false*.
Elapsed Event	This event can be raised when the countdown timer completes.
StopMode	This property defines what happens when *countdownstart* is set to false while the countdowr is progressing. The options are: • **StopAndReset**: stopping the timer by setting *countdownstart* to *false* will cause the timer to stop and be reset to the countdown value • **Pause**: stopping the timer by setting *countdownstart* to *false* will cause the timer to stop where it is in the countdown process and hold that value; etting *countdownstart* to *true*

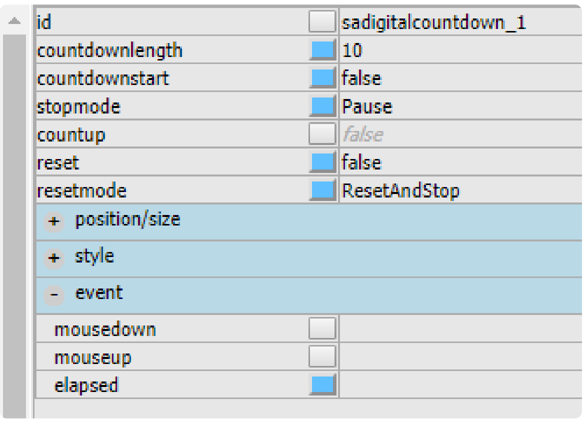
	again will cause the countdown to continue from where it left off
CountUp	Changing this value to *True* instead of *False* will cause the timer to count up rather than down to th selected countdownlength. If countdownlength is zero it will count up indefinitely until stopped and/ reset.
Reset	This is an action property that can be used by Log Flows and/or bindings to reset the counter. Setting it to *true* will cause the timer to reset.
ResetMode	This property defines what happens when a reset is issued and a countdown is in progress. The options are: • **ResetAndStop**: Setting reset to true will caus the timer to reset and stop its countdown. • **ResetAndContinue**: If the timer is running thi will cause it to reset and continue running. If tl timer is stopped this will just reset the value.

Currently, the timer does not automatically reset when the countdown completes. If that functionality is desired, it can be obtained by using the property bindings to hook the elapsed event to the reset state such that when the timer elapses reset is triggered. This can be accomplished either by using the Flow in *elapsed* to change the reset or by using the Flow in *reset* to change based on the elapsed start point.

In both cases, the Panel must be saved with the binding buttons for these properties/events turned on to surface those options in the Flow property tree.

For example:

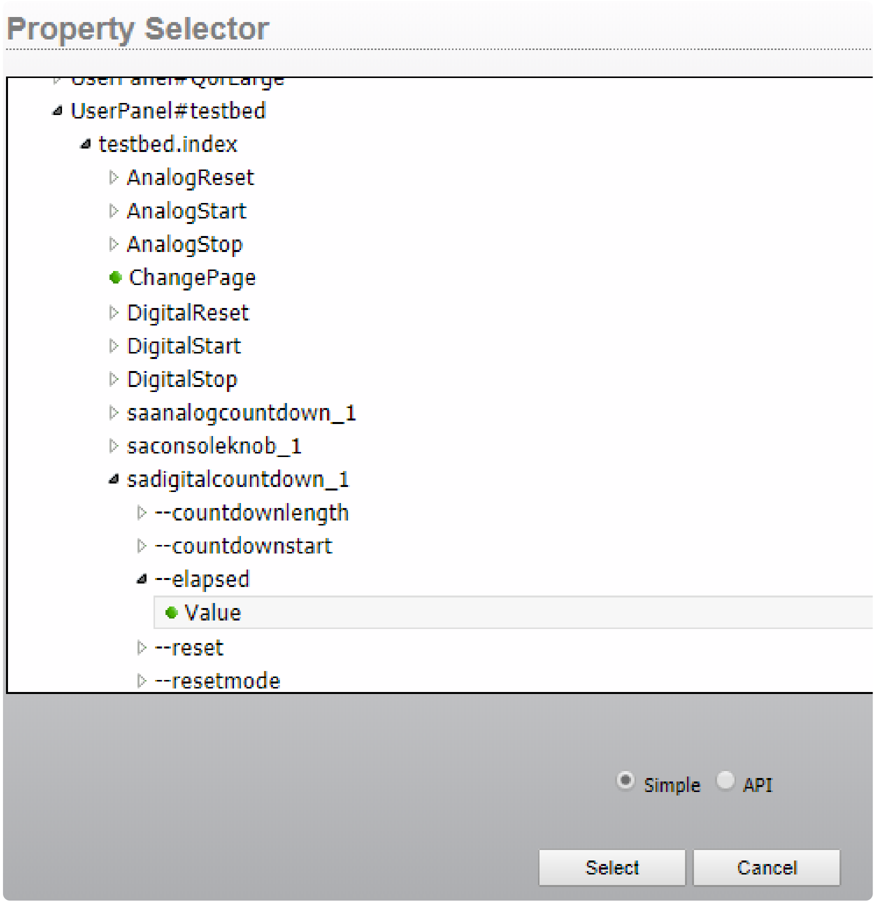
- Enable the binding buttons for both *elapsed* and *reset* have been enabled and save your Panel



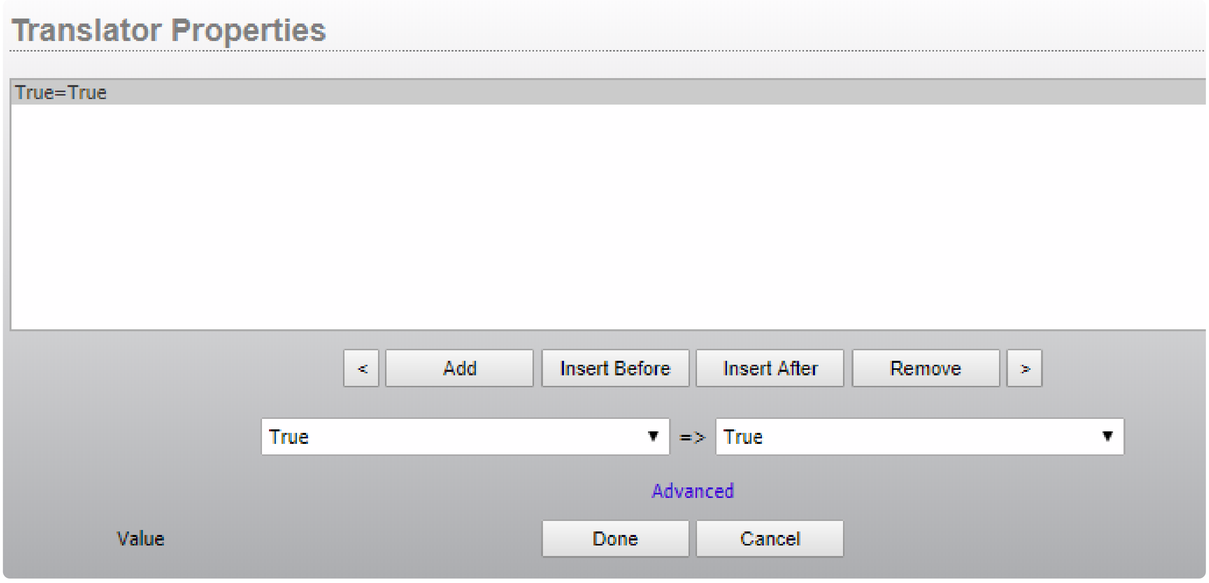
id	□	sadigitalcountdown_1
countdownlength	■	10
countdownstart	■	false
stopmode	■	Pause
countup	□	*false*
reset	■	false
resetmode	■	ResetAndStop
+ position/size		
+ style		
- event		
mousedown	□	
mouseup	□	
elapsed	■	

- Select the *reset* property field and click the start point in the bottom corner
- Browse to the User Panels group, locate the digital countdown control in your specific Panel, and select

the *elapsed* **Value** property

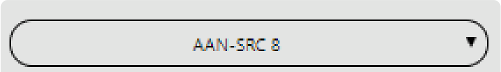


- Select **True=True** for the translation



- Click **Done**. After saving the Panel if the countdown elapses the timer should automatically reset

List Selector



At this point in time, the list selector component should be used primarily by advanced users that understand the API. In a future version, we will add options for simplifying the selection of list content and data. This component provides a drop-down list for selecting elements in the system. For example, it could be used as a source selector for a virtual router or a show profile selector for a console. Unfortunately, until we finish a more intuitive configuration user interface, some knowledge of the API and inner working of PathfinderCore PRO are required to configure this component. A thorough understanding of SapV2 (Appendix A) will help in the configuration of this component. Also, feel free to reach out to support for guidance.

There are 4 primary properties that are used together to describe the list options and another two for event and state:

- Listsearchpath: This holds a SapV2 object path and optional property value for the root from which all objects in the list will be obtained; for example:
    - Routers#0.VirtualRouter#4 SapObjectType=VirtualSource
        - This specifies that we will be filling the list with data from virtual sources in virtual router number 4
    - Devices#0.Qor#[tcp://172.16.1.59:93].LwcpInterpreter#0.LwcpRoot#0.AppControl#0 ObjectName=ShowProfile
        - This specified that we will be filling the list with data from show profiles on the Qor at 172.16.1.59
- Listsearchdepth: The number of branches below the listsearch path root to look for elements; many times, this will be 1 for one level deeper than the root selection, but it could be higher or -1 for infinite For Example:
    - Devices#0.Qor#[tcp://172.16.1.59:93].LwcpInterpreter#0.LwcpRoot#0.AppControl#0.ShowProfile#0
        - This is one branch deeper than the search path in the last example above
- ItemDisplayProperty: This is the property that will be used as the display data for each object; this is what the user sees; in both of the examples above this would likely be the name property
- Itemselectvalue: This is the property whose value will be used in the change and current value events. In the examples above it would be the Id and/or ObjectId properties
- CurrentValue: this can be used by logic flows to select the displayed value
- Change event: this can be used to make a change when a different item in the drop down is selected; the value that will be available in the translation is defined by the itemdisplayproperty

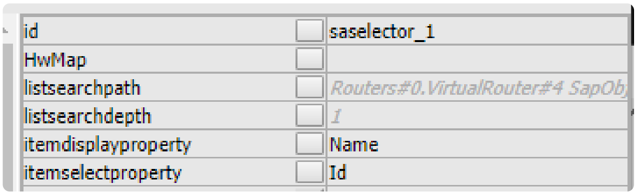
Let’s work through two examples.

List Selector Example 1

For example 1 we will use the selector to present a list of sources from a virtual router. When the user selects a different source, it will change a specific destination on that router. Make sure you have a virtual router in the system. In this case, we will use VirtualRouter 4. Set the following property values:

- Listsearchpath: Routers#0.VirtualRouter#4 SapObjectType=VirtualSource
- Listsearchdepth: 1
- Itemdisplayproperty: Name

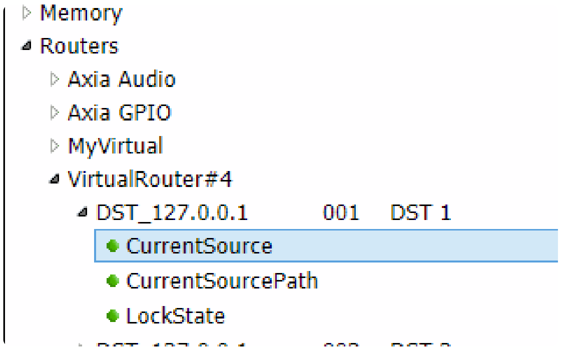
- Itemdisplayvalue: Id



id	□	saselector_1
HwMap	□	
listsearchpath	□	Routers#0.VirtualRouter#4 SapObj
listsearchdepth	□	1
itemdisplayproperty	□	Name
itemselectproperty	□	Id

This means that we are looking for objects up to 1 level deep below the Routers#0.VirtualRouter#4 path whose SapObjectType=VirtualSource. The last part of this is important so that the selector does not also show destinations. For each source that is found the selector will create an item in the select list whose display value is taken from the name property and whose select value is taken from the id property. This will display a list of sources by name from virtual router 4 and the value used when a selection is made will be the source's id.

Next, enable the binding on the currentvalue and change properties. For the change property select Destination 1's current source property in the virtual router.

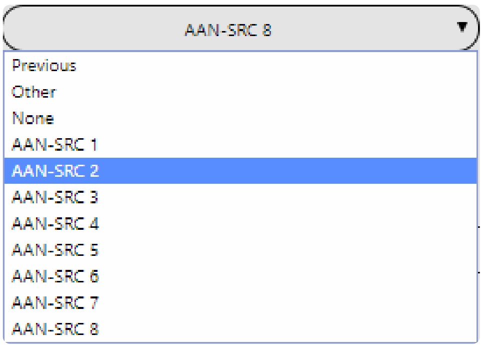


```
▷ Memory
◢ Routers
   ▷ Axia Audio
   ▷ Axia GPIO
   ▷ MyVirtual
   ◢ VirtualRouter#4
      ◢ DST_127.0.0.1        001    DST 1
         ● CurrentSource
         ● CurrentSourcePath
         ● LockState
```

And use *=* for the translation. This will make sure that any time a new selection is made in the list, the number of that source will be passed to the destination's currentsource property thereby affecting a route change.

For the currentvalue property, select destination 1's current source again as the start point. Translation would again be *=*. This will ensure that if the destination route changes even by some other means that the drop down list will change to show the correct source.

In this way, we have created a route selector from a drop-down list.



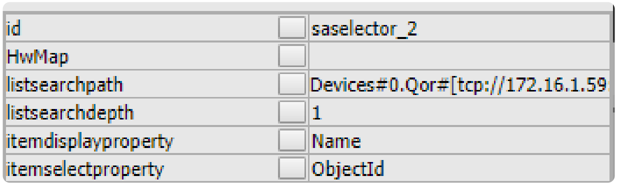
AAN-SRC 8 ▼

Previous
Other
None
AAN-SRC 1
AAN-SRC 2
AAN-SRC 3
AAN-SRC 4
AAN-SRC 5
AAN-SRC 6
AAN-SRC 7
AAN-SRC 8

List Selector Example 2

In example 2 we will use the selector to present a list of show profiles for a qor/iq. When the user selects a different show profile for the qor/iq, it will change the show profile. Set the following property values (modifying for your own device):

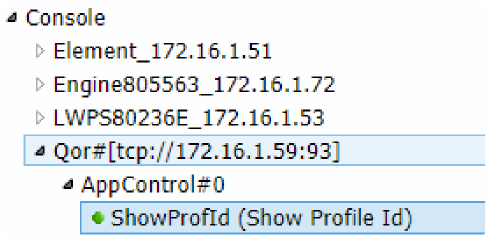
- Listsearchpath: Devices#0.Qor#[tcp://172.16.1.59:93].LwcpInterpreter#0.LwcpRoot#0.AppControl#0 ObjectName=ShowProfile
- Listsearchdepth: 1
- Itemdisplayproperty: Name
- Itemdisplayvalue: ObjectId



id	☐	saselector_2
HwMap	☐	
listsearchpath	☐	Devices#0.Qor#[tcp://172.16.1.59
listsearchdepth	☐	1
itemdisplayproperty	☐	Name
itemselectproperty	☐	ObjectId

This means that we are looking for objects up to 1 level deep below the Devices#0.Qor#[tcp://172.16.1.59:93].LwcpInterpreter#0.LwcpRoot#0.AppControl#0 path whose SapObjectType=ShowProfile. For each show profile that is found the selector will create an item in the select list whose display value is taken from the name property and whose select value is taken from the ObjectId property. This will display a list of sources by name from Qor/Iq in question and the value used when a selection is made will be the show profile's id.

Next enable the binding on the currentvalue and change properties. For the change property select the console's appcontrol ShowProfId property.

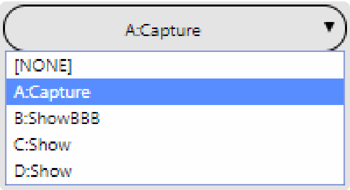


Console
Element_172.16.1.51
Engine805563_172.16.1.72
LWPS80236E_172.16.1.53
Qor#[tcp://172.16.1.59:93]
AppControl#0
● ShowProfId (Show Profile Id)

And use *=* for the translation. This will make sure that any time a new selection is made in the list, the number of that show profile will be passed to the console's showprofid property thereby affecting a show profile change.

For the currentvalue property, select showprofid property again as the start point. Again use *=* for the translation. This will ensure that if the show profile changes even by some other means, that the drop-down list will change to show the correct show profile.

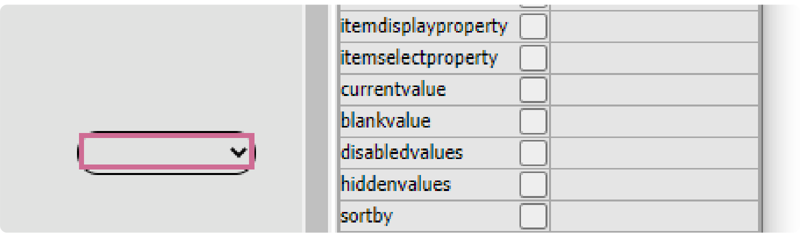
In this way we have created a drop-down list for selecting a new show profile.



A:Capture
[NONE]
A:Capture
B:ShowBBB
C:Show
D:Show

We realize that until more intuitive user configuration user interfaces are created, this component may be challenging. If you need to understand how to use it for a specific task, please reach out to support and realize that currently this is an advanced feature and may require some time for them to obtain the correct parameters for your task.

In addition to the primary properties discussed above which are required to make the list selector work properly, there are several additional properties that require explanation:



blankvalue

Each item in the list has a display value obtained from the display property and an actual value that gets used when the item is selected obtained from the itemselectproperty. If the current actual value is blank, there may not be an item in the list which matches a blank value. This allows you to define a value to use in the list selection if the current value is blank. For example, when using a virtual destination list, the none route source may return a currentsource value of blank rather than zero. The id value for the none source is zero. By entering 0 in this field, the 0 value will be used when selecting the current item in the list whenever a blank value is returned from the system. Without this setting, the list might display the last known selection rather than none.

disabledvalues

This field can accept a comma delineated list of values which if they exist in the list will display in the drop down list but be disabled and unselectable. Use the value from the itemselect property in this list not the display value. For example if the drop down list was showing the virtual sources of a virtual router, using an entry of:

-3,-1

Would make the Other and Previous sources disabled.

hiddenvalues

This field works exactly the same as disabledvalues except it will hide the entries presented in the comma delineated list.

sortby

This option allows you to sort the items in the drop-down list by the original inbound order, alphabetically by value, or alphabetically by the display name.

--option-background-color

This allows you to define the background color of the drop-down menu that pops up when you try to select a new item.

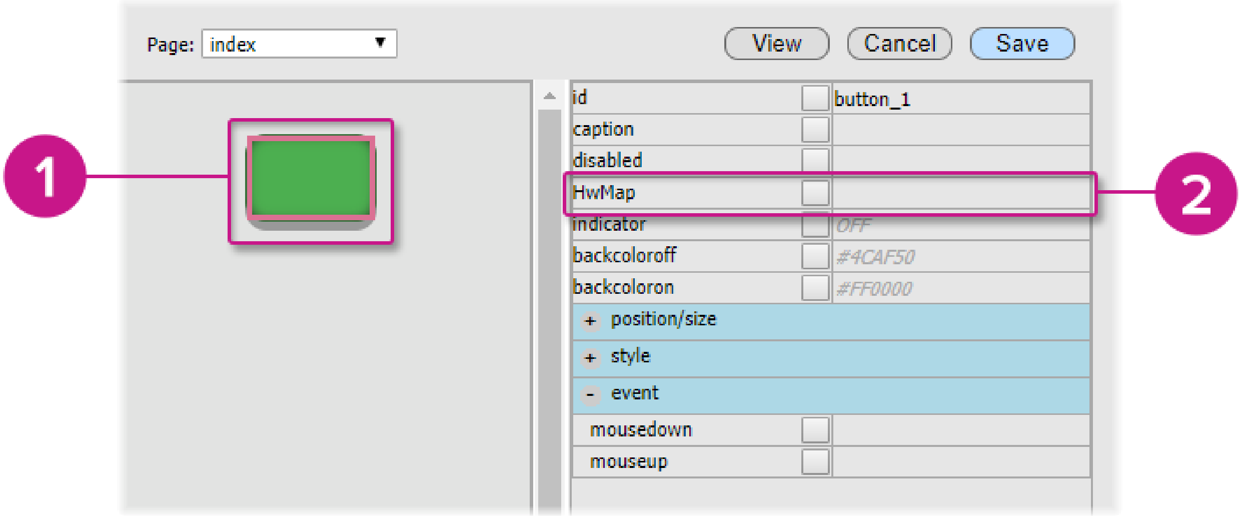
--option-color

This allows you to define the font color of the items in the drop-down menu that pops up when you try to select a new item.

Hardware Mapping Buttons

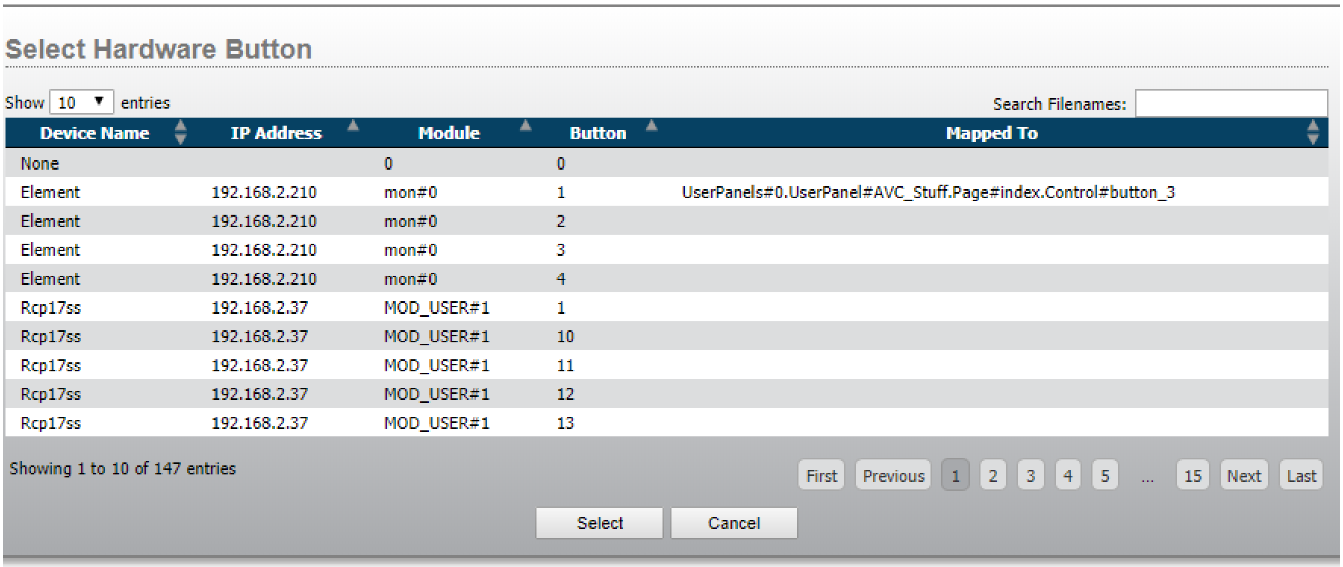
HTML or Console Buttons may be mapped to physical LCD buttons in the console or rack mount button panels. Hardware mapping makes the physical button mirror the behavior of the software button. To define this:

- Click on a User Panel **Button** and click the **HwMap fiel**



id	□	button_1
caption	□	
disabled	□	
HwMap	□	
indicator	□	OFF
backcoloroff	□	#4CAF50
backcoloron	□	#FF0000
+ position/size		
+ style		
- event		
mousedown	□	
mouseup	□	

- A dialog will appear listing the known hardware buttons in the system



Select Hardware Button

Show 10 entries Search Filenames:

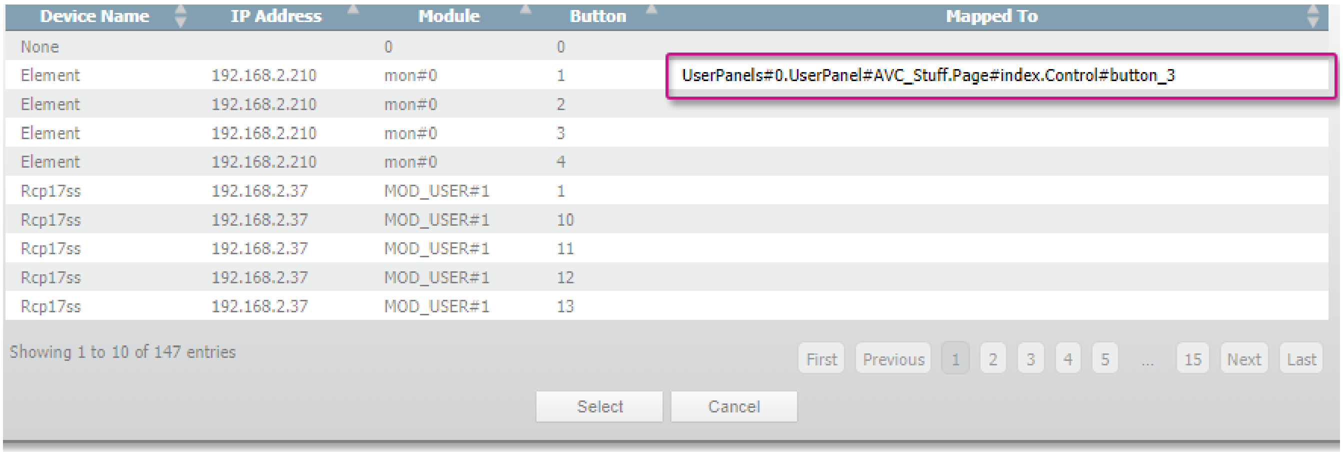
Device Name	IP Address	Module	Button	Mapped To
None		0	0	
Element	192.168.2.210	mon#0	1	UserPanels#0.UserPanel#AVC_Stuff.Page#index.Control#button_3
Element	192.168.2.210	mon#0	2	
Element	192.168.2.210	mon#0	3	
Element	192.168.2.210	mon#0	4	
Rcp17ss	192.168.2.37	MOD_USER#1	1	
Rcp17ss	192.168.2.37	MOD_USER#1	10	
Rcp17ss	192.168.2.37	MOD_USER#1	11	
Rcp17ss	192.168.2.37	MOD_USER#1	12	
Rcp17ss	192.168.2.37	MOD_USER#1	13	

Showing 1 to 10 of 147 entries

- The **Mapped To** field will display a value if the button has already been hardware mapped as each hardware button can only be mapped to a single software button. Physical buttons do not have to be hardware mapped. They can also just be used directly with Logic Flows

Select Hardware Button

Show 10 entries Search Filenames:



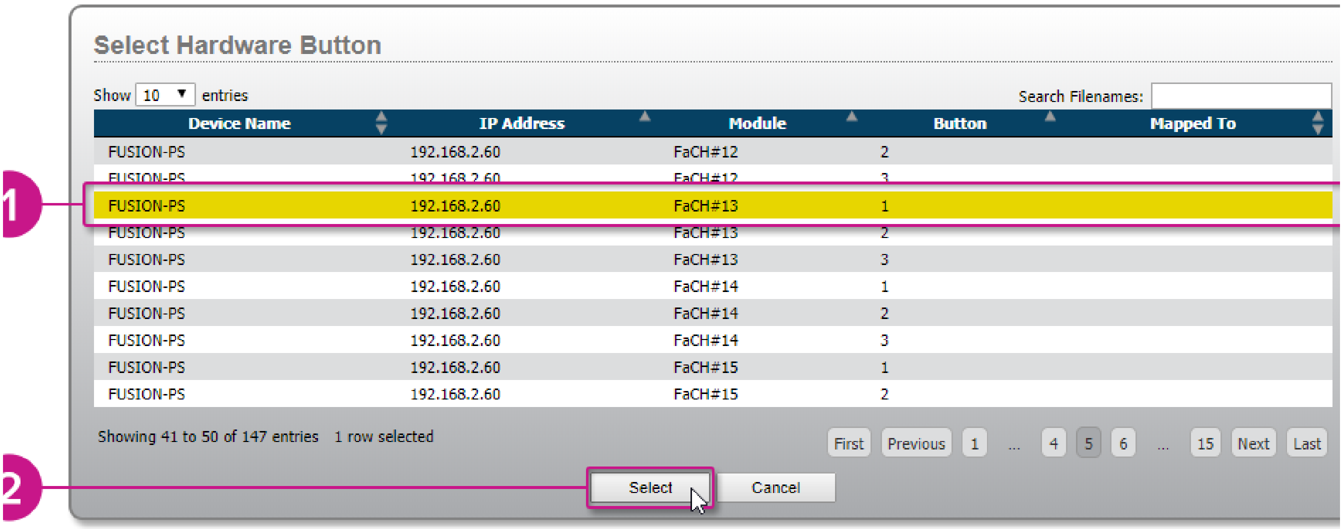
Device Name	IP Address	Module	Button	Mapped To
None		0	0	
Element	192.168.2.210	mon#0	1	UserPanels#0.UserPanel#AVC_Stuff.Page#index.Control#button_3
Element	192.168.2.210	mon#0	2	
Element	192.168.2.210	mon#0	3	
Element	192.168.2.210	mon#0	4	
Rcp17ss	192.168.2.37	MOD_USER#1	1	
Rcp17ss	192.168.2.37	MOD_USER#1	10	
Rcp17ss	192.168.2.37	MOD_USER#1	11	
Rcp17ss	192.168.2.37	MOD_USER#1	12	
Rcp17ss	192.168.2.37	MOD_USER#1	13	

Showing 1 to 10 of 147 entries

If the button you want is not shown, make sure the LCD Panel or console is in the devices list. If not, it needs to be discovered. *See the **Discovery** topic in Chapter 1 for more details on discovering devices.*

LCD Panels must be manually discovered in devices using the add and Lwcp discovery. If a console device exists in the system but is not showing the LCD buttons, try pressing a few of the LCD buttons and then refreshing the web page.

- Select the **button** you want to hardware map and click **Select**



Select Hardware Button

Show 10 entries

Search Filenames:

Device Name	IP Address	Module	Button	Mapped To
FUSION-PS	192.168.2.60	FaCH#12	2	
FUSION-PS	192.168.2.60	FaCH#12	3	
FUSION-PS	192.168.2.60	FaCH#13	1	
FUSION-PS	192.168.2.60	FaCH#13	2	
FUSION-PS	192.168.2.60	FaCH#13	3	
FUSION-PS	192.168.2.60	FaCH#14	1	
FUSION-PS	192.168.2.60	FaCH#14	2	
FUSION-PS	192.168.2.60	FaCH#14	3	
FUSION-PS	192.168.2.60	FaCH#15	1	
FUSION-PS	192.168.2.60	FaCH#15	2	

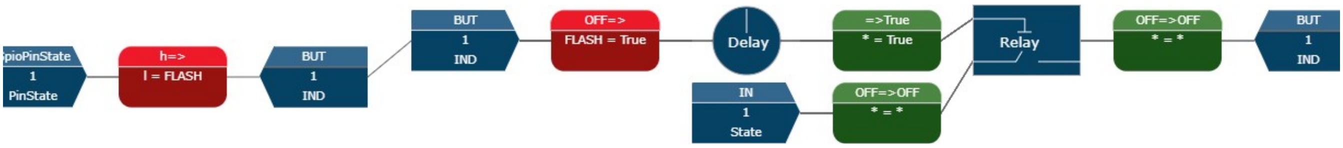
Showing 41 to 50 of 147 entries 1 row selected

Important Note: Once the Panel is saved, hardware mapping in HTML Panels takes place natively in the application and does not require hardware map Logic Flows as the legacy Panels did.

Button Indicator PULSE Values

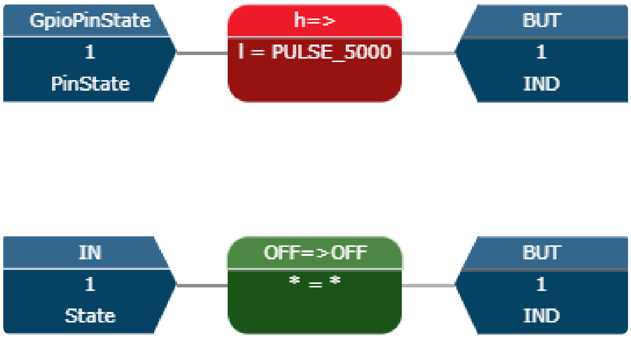
PULSE0 through *PULSE10000* will flash an HTML or Console Button for the specified number of milliseconds (*PULSE**[ms]***)and then return to the last requested state. If the indicator state is changed while the pulse flash is in progress the new state will be returned once the flash time is complete. *PULSE0* may be used to cancel an in-progress pulse. This allows the easy binding of state (ON/OFF) conditions while allowing an additional Flow to control flashing.

For example, if we want a GPI to cause a button to flash for 5 seconds and then turn on or off depending on the state of VMIX 1, we could build a Flow like this:



This type of Flow is useful for situations where you need to flash a button as an alert but retain a known state after the flashing is complete.

With the addition of *PULSE,* we can achieve the same functionality with a much simpler Flow:

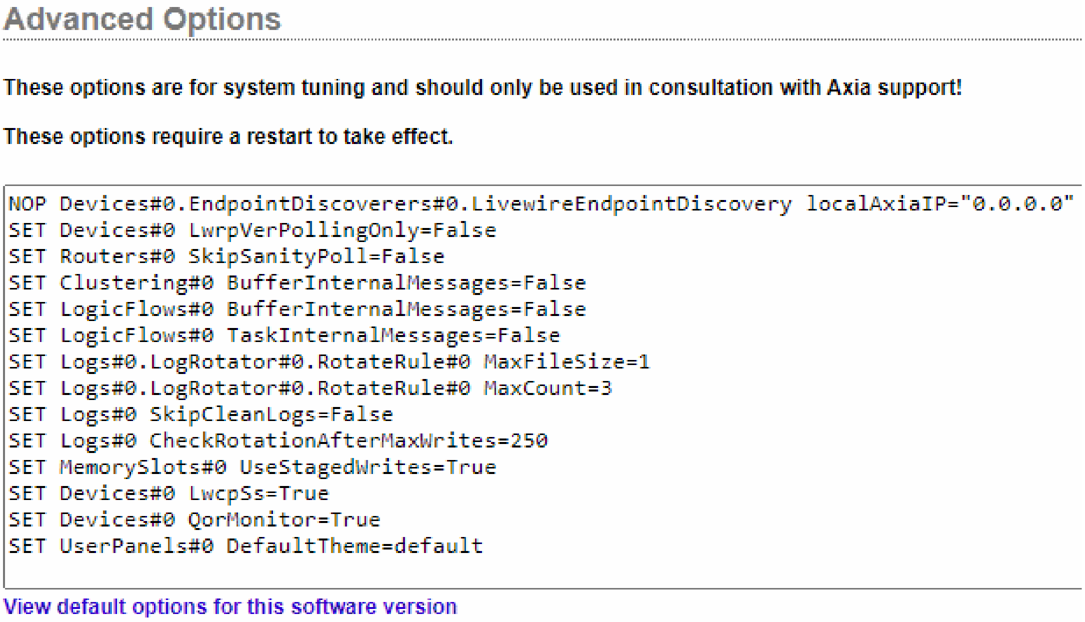


In this case, the button indication is tied to the VMIX state but setting Pin 1 to *Low* will cause a 5000ms (5-second) flash. It will then return to whatever state the VMIX is currently in even if that state has changed during the duration of the pulse.

If the button is an HTML5 button this can be made even simpler by binding the indicator to the VMIX state within the User Panel and only implementing the *PULSE* (flash) flow in Logic Flows.

Theme Advanced Options

If you decide you prefer one theme to the other, you can set an Advanced option to make it the default theme for new panels.



Advanced Options

These options are for system tuning and should only be used in consultation with Axia support!

These options require a restart to take effect.

```
NOP Devices#0.EndpointDiscoverers#0.LivewireEndpointDiscovery localAxiaIP="0.0.0.0"
SET Devices#0 LwrpVerPollingOnly=False
SET Routers#0 SkipSanityPoll=False
SET Clustering#0 BufferInternalMessages=False
SET LogicFlows#0 BufferInternalMessages=False
SET LogicFlows#0 TaskInternalMessages=False
SET Logs#0.LogRotator#0.RotateRule#0 MaxFileSize=1
SET Logs#0.LogRotator#0.RotateRule#0 MaxCount=3
SET Logs#0 SkipCleanLogs=False
SET Logs#0 CheckRotationAfterMaxWrites=250
SET MemorySlots#0 UseStagedWrites=True
SET Devices#0 LwcpSs=True
SET Devices#0 QorMonitor=True
SET UserPanels#0 DefaultTheme=default
```

View default options for this software version

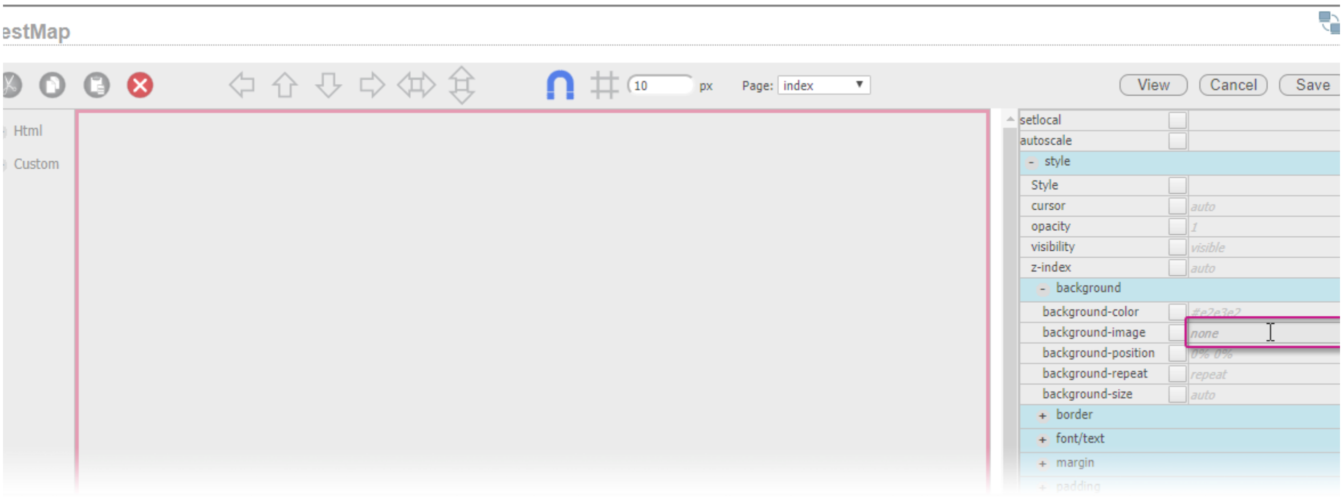
The only two options that can be used for the DefaultTheme advanced setting are default and dark. Setting this option to any other setting may prevent you from creating new panels until it is fixed. A reboot is required before any advanced setting takes effect.

It is also important to note that the theme just presents you with the default styling of the components. All properties for customizing that style to your own liking and design are still available in both themes.

Changing Pages

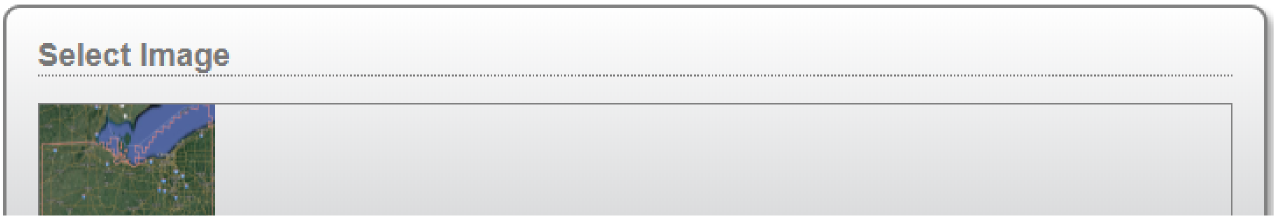
Creating additional pages is described in the page section of the toolbar documentation earlier in this chapter. In this section, we will explore how to switch between panels and pages. Switching between multiple Pages can be accomplished by using the ChangePage property. Let's build an example set of linked pages. This example will create a map with an alarm state that can be clicked to obtain more details on the alarm from a second page.

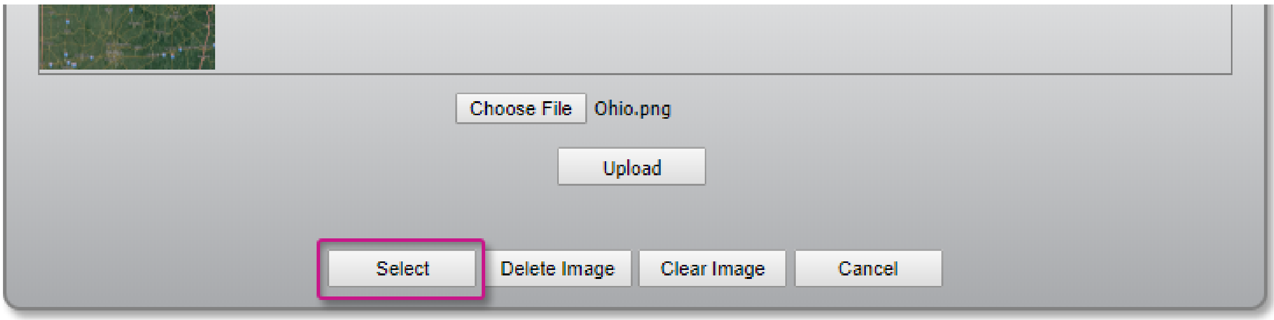
- Create a new Panel and click **Save**. Name the Panel **TestMap**
- Click on your new **Panel** to select it. In the property grid, expand the **style>background** section and click the **background-image** property. A dialog will appear for selecting and uploading images



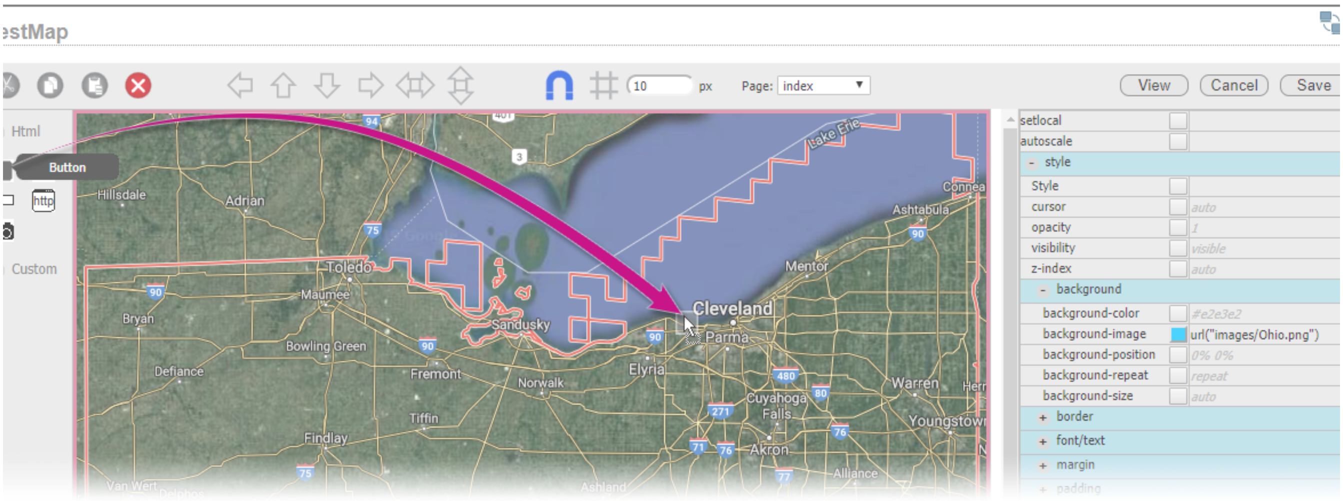
setlocal	
autoscale	
- style	
Style	
cursor	auto
opacity	1
visibility	visible
z-index	auto
- background	
background-color	#e2e3e2
background-image	none
background-position	0% 0%
background-repeat	repeat
background-size	auto
+ border	
+ font/text	
+ margin	
+ padding	

- For this example, we will upload an image of a map of Ohio; click **Choose File**, select your stored image, and click **Upload;** the image will appear in the selection list. Click on the **image** and click **Select; t**he background of the Panel will now display the image of Ohio





- Drag a **button** into the Panel and position it next to Cleveland

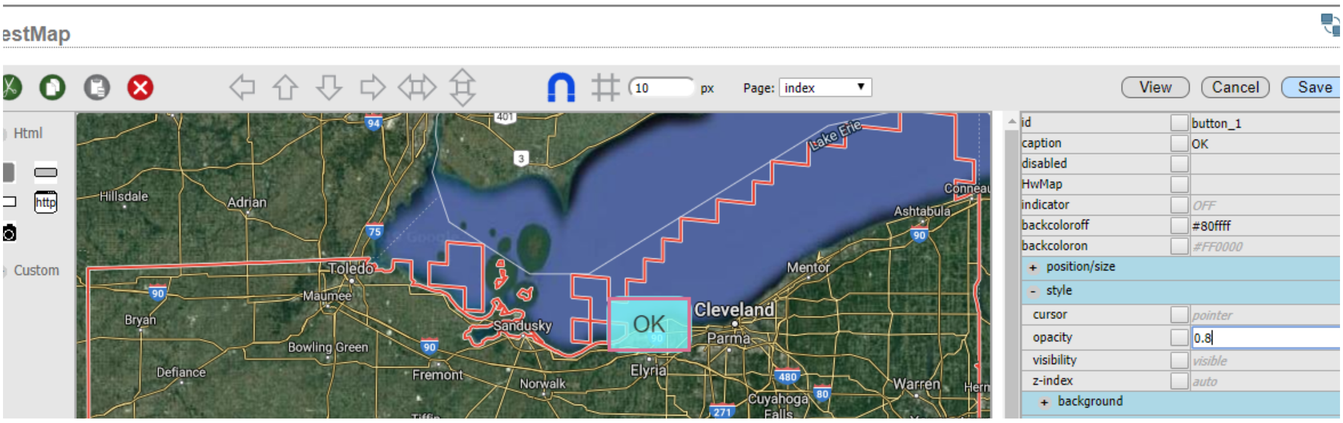


setlocal	
autoscale	
- style	
Style	
cursor	auto
opacity	1
visibility	visible
z-index	auto
- background	
background-color	#e2e3e2
background-image	url("images/Ohio.png")
background-position	0% 0%
background-repeat	repeat
background-size	auto
+ border	
+ font/text	
+ margin	
+ padding	

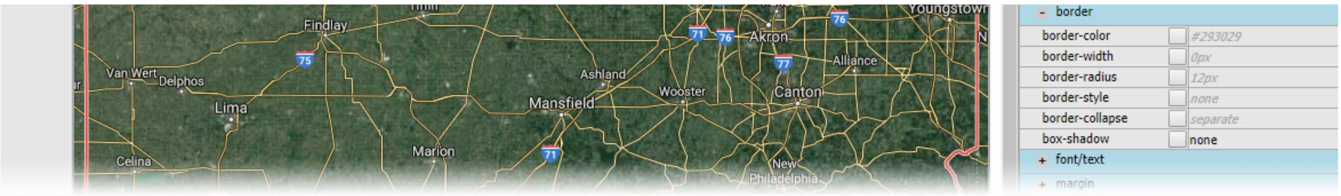
- Click on your new **button** to select it. In the Properties grid, modify the button's Properties:

Property	Value
caption	OK
backcoloroff	cyan/#80FFFF
style>opacity	.8
style>border>box-shadow	None

Your Panel should now look something like this:

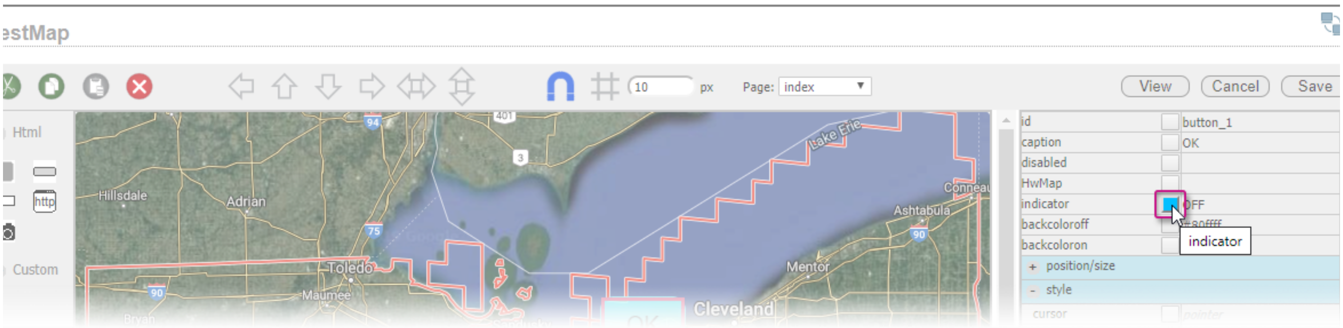


id	button_1
caption	OK
disabled	
HwMap	
indicator	OFF
backcoloroff	#80ffff
backcoloron	#FF0000
+ position/size	
- style	
cursor	pointer
opacity	0.8
visibility	visible
z-index	auto
+ background	



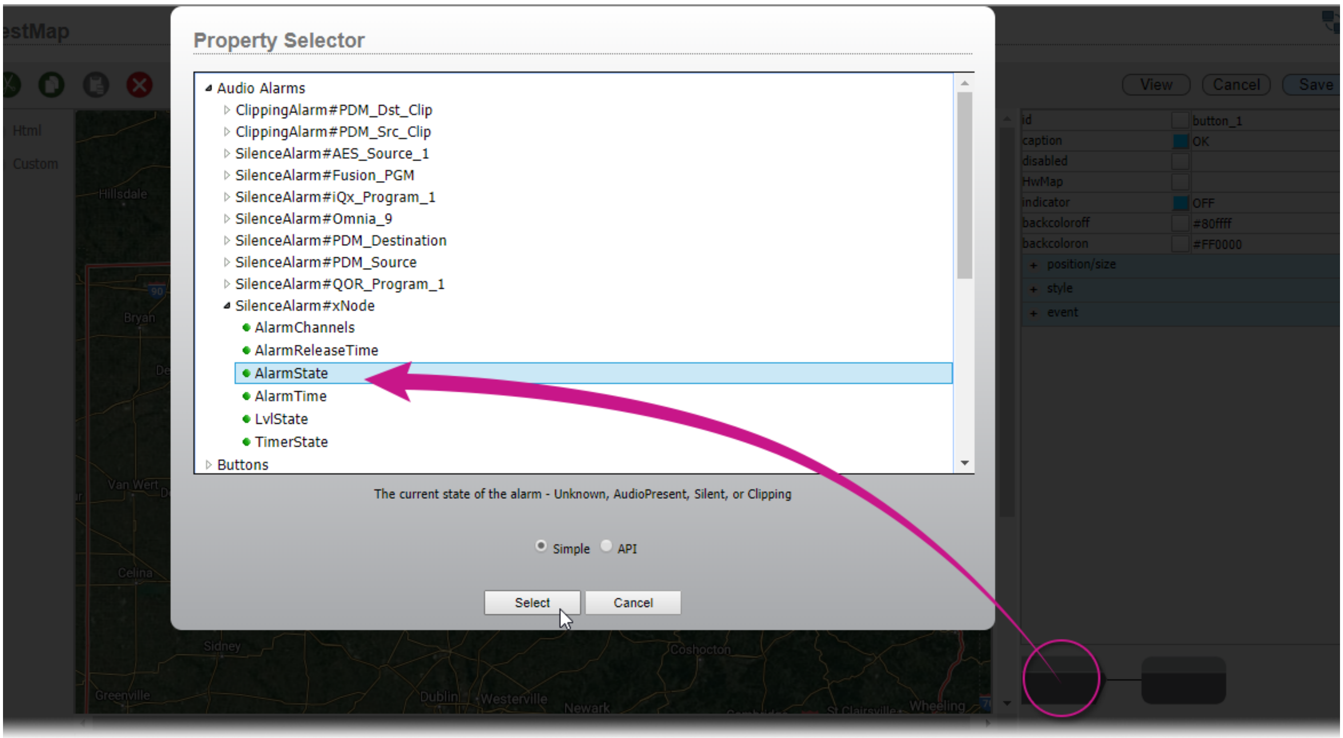
border-color	#293029
border-width	0px
border-radius	12px
border-style	none
border-collapse	separate
box-shadow	none

- Enable the binding for the **indicator** state of the button



id	button_1
caption	OK
disabled	
HwMap	
indicator	OFF
backcoloroff	#80ffff
backcoloron	

- Click on the **Start Point** and select a *SilenceAlarm* **AlarmState**



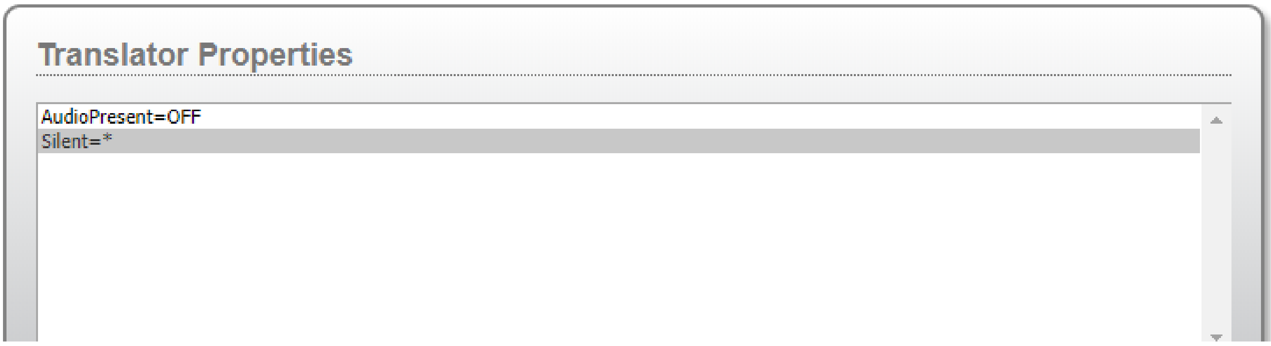
Property Selector

- Audio Alarms
  - ClippingAlarm#PDM_Dst_Clip
  - ClippingAlarm#PDM_Src_Clip
  - SilenceAlarm#AES_Source_1
  - SilenceAlarm#Fusion_PGM
  - SilenceAlarm#iQx_Program_1
  - SilenceAlarm#Omnia_9
  - SilenceAlarm#PDM_Destination
  - SilenceAlarm#PDM_Source
  - SilenceAlarm#QOR_Program_1
  - SilenceAlarm#xNode
    - AlarmChannels
    - AlarmReleaseTime
    - AlarmState
    - AlarmTime
    - LvlState
    - TimerState
- Buttons

The current state of the alarm - Unknown, AudioPresent, Silent, or Clipping

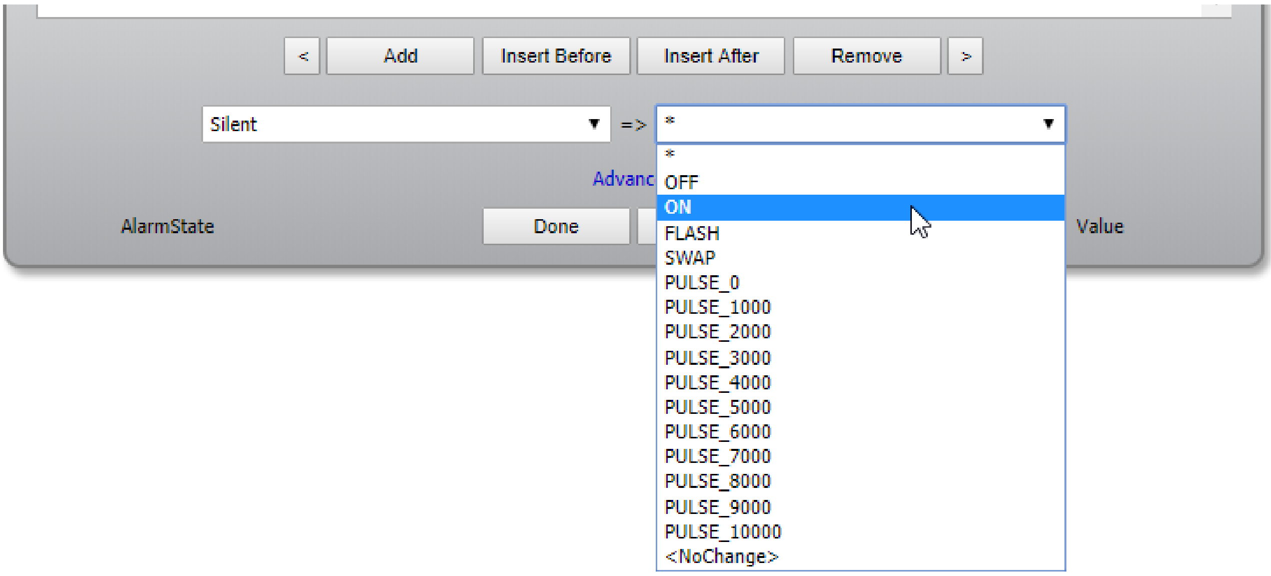
○ Simple ○ API

- For the Translation, select *AudioPresent* as indicator **OFF** and *Silent* as **ON**

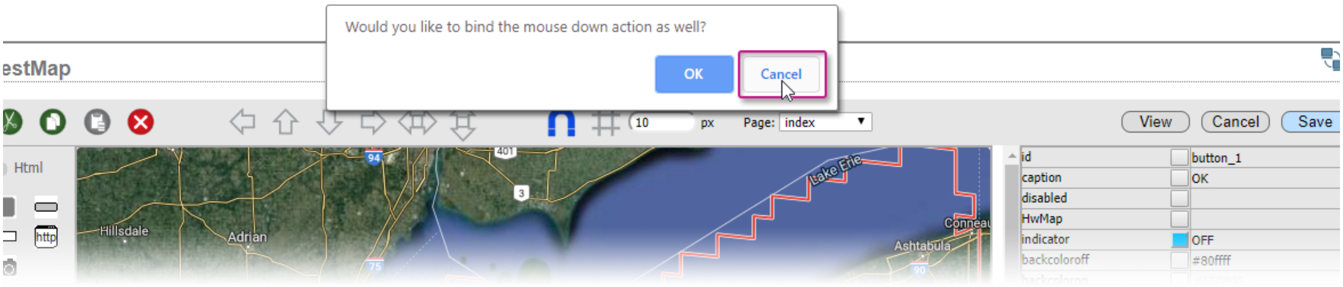


Translator Properties

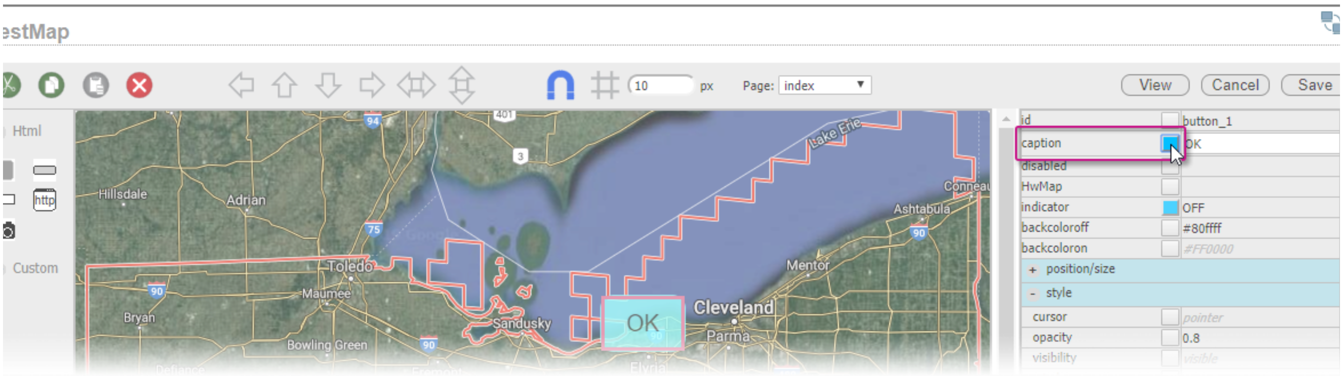
```
AudioPresent=OFF
Silent=*
```



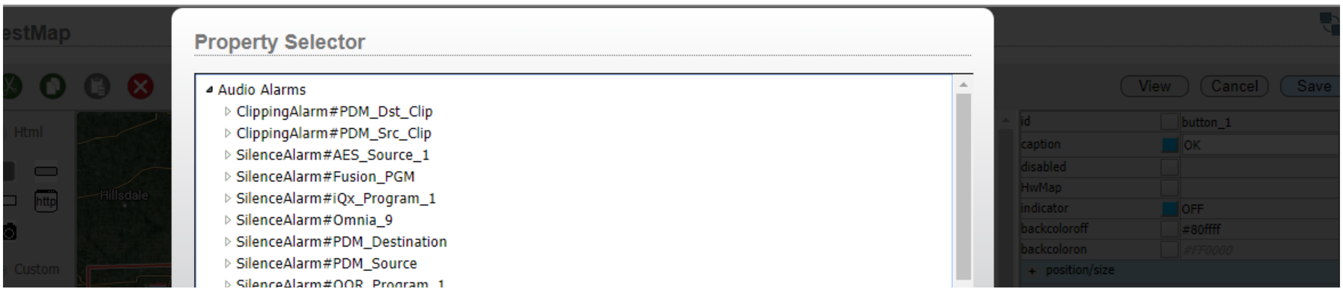
- Click **Done;** when the system asks about the *mouse down* action, click **Cancel;** in this case, the *mouse down* is going to change pages and has nothing to do with the property we are picking for the indicator so we will set it separately

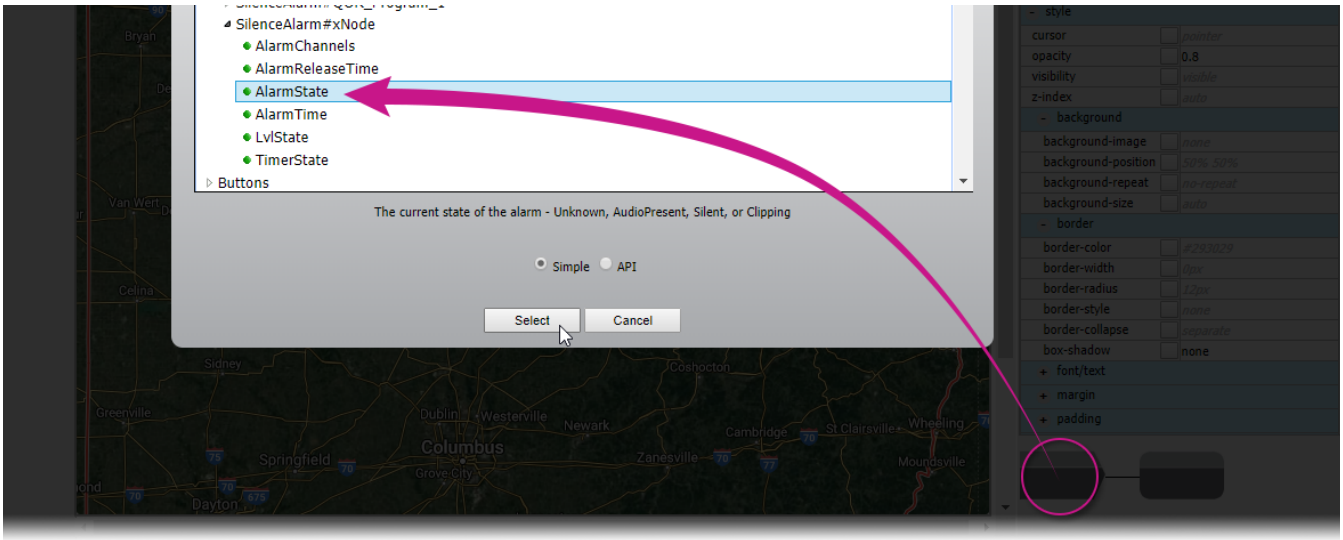

Would you like to bind the mouse down action as well?

- Turn on the binding for the *Caption* property



- Click the Flow's **Start Point;** select the same *AlarmState* property and click **Select**





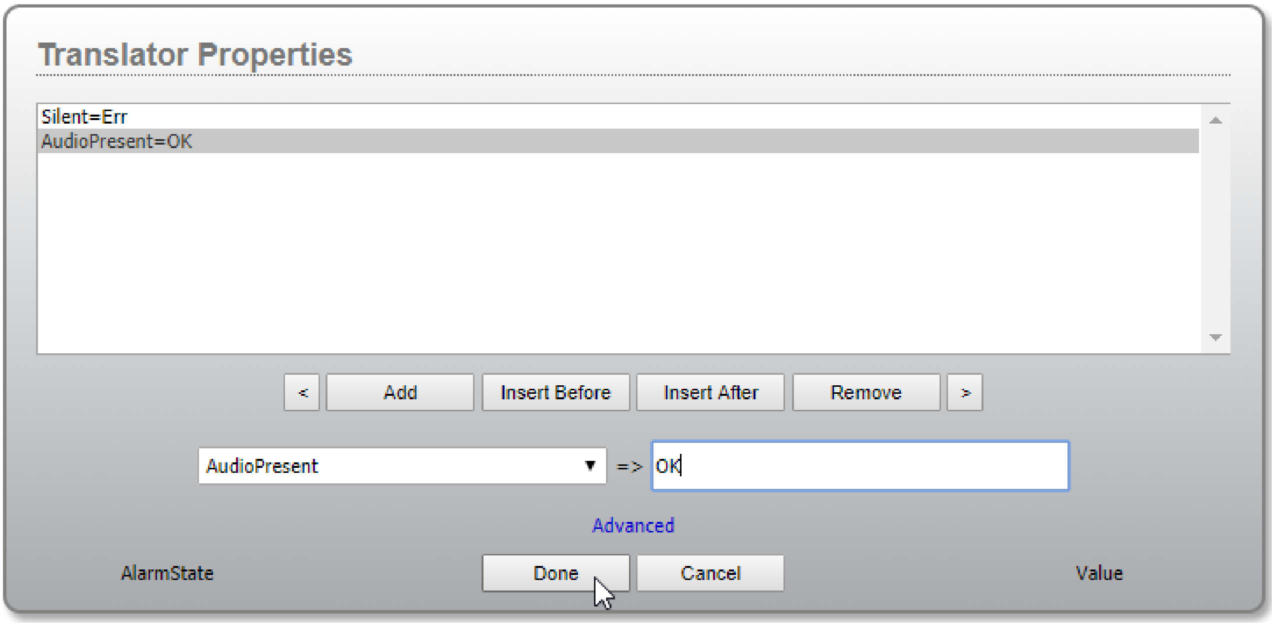
SilenceAlarm#xNode
- AlarmChannels
- AlarmReleaseTime
- AlarmState
- AlarmTime
- LvlState
- TimerState

Buttons

The current state of the alarm - Unknown, AudioPresent, Silent, or Clipping

● Simple ○ API

- For the Translation make *Silent* convert to **Err** and *AudioPresent* to **OK**. Click **Done**

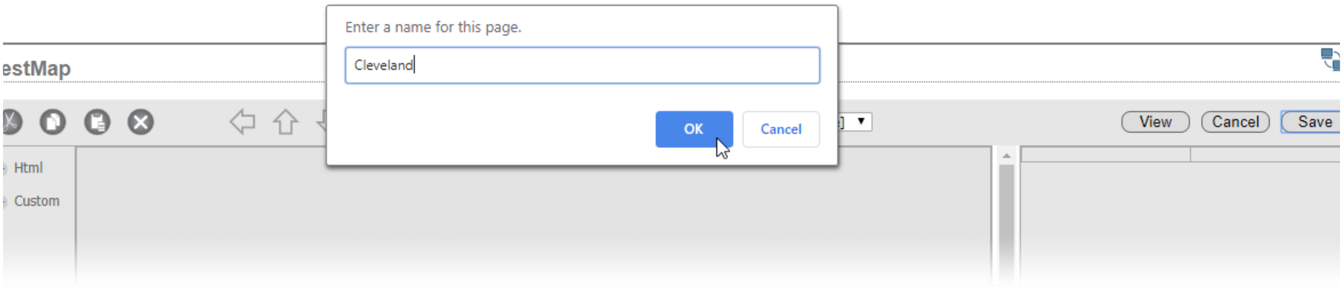


Translator Properties

Silent=Err
AudioPresent=OK

AudioPresent => OK

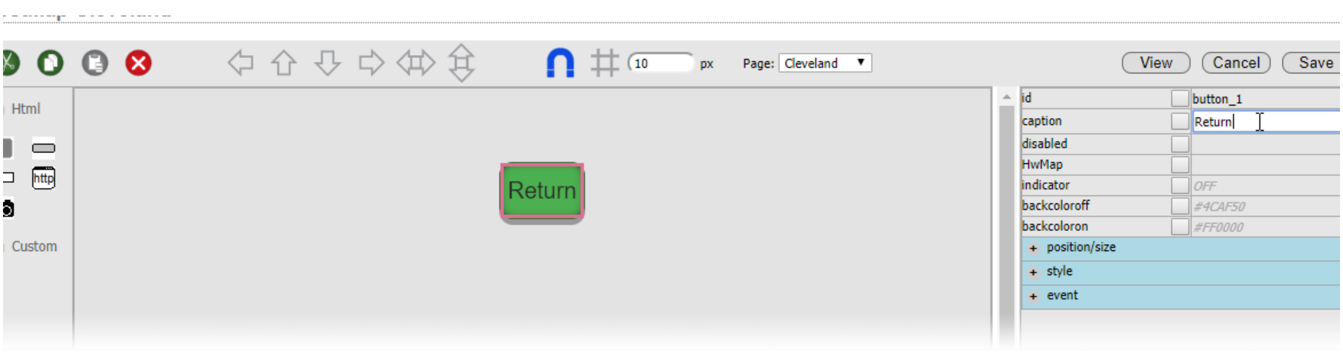
- Save the Panel; at this point, if we were to execute this Panel, the button next to Cleveland should be **cyan** and **OK** if the silence alarm has audio presence, and **Err** and **Red** if it is silent
- From the Page drop down, select **[NewPage];** save the new page with the name **Cleveland**



Enter a name for this page.

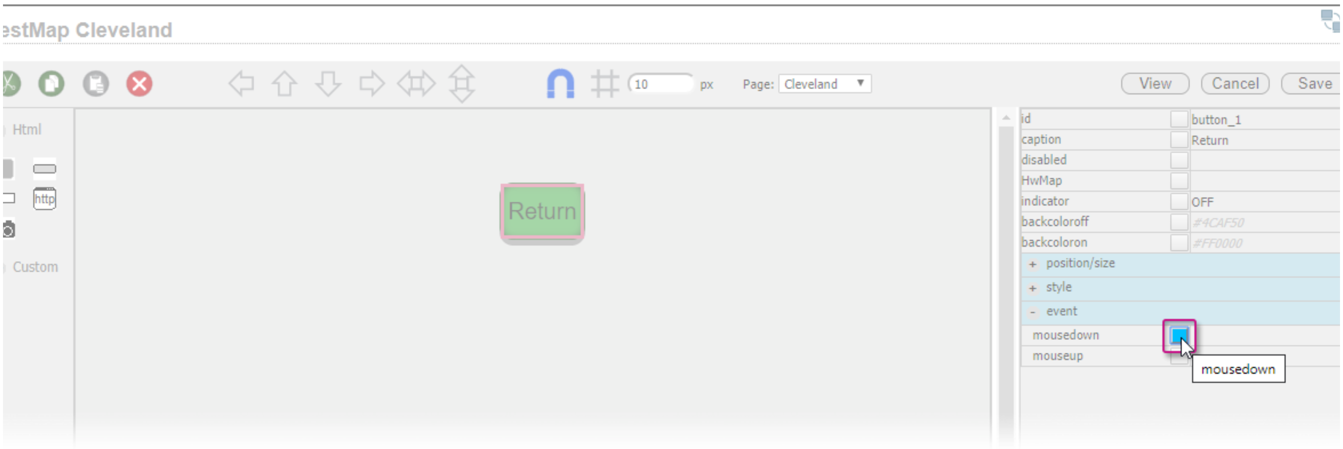
Cleveland

- Drag a new button onto the new page and set its caption to **Return**



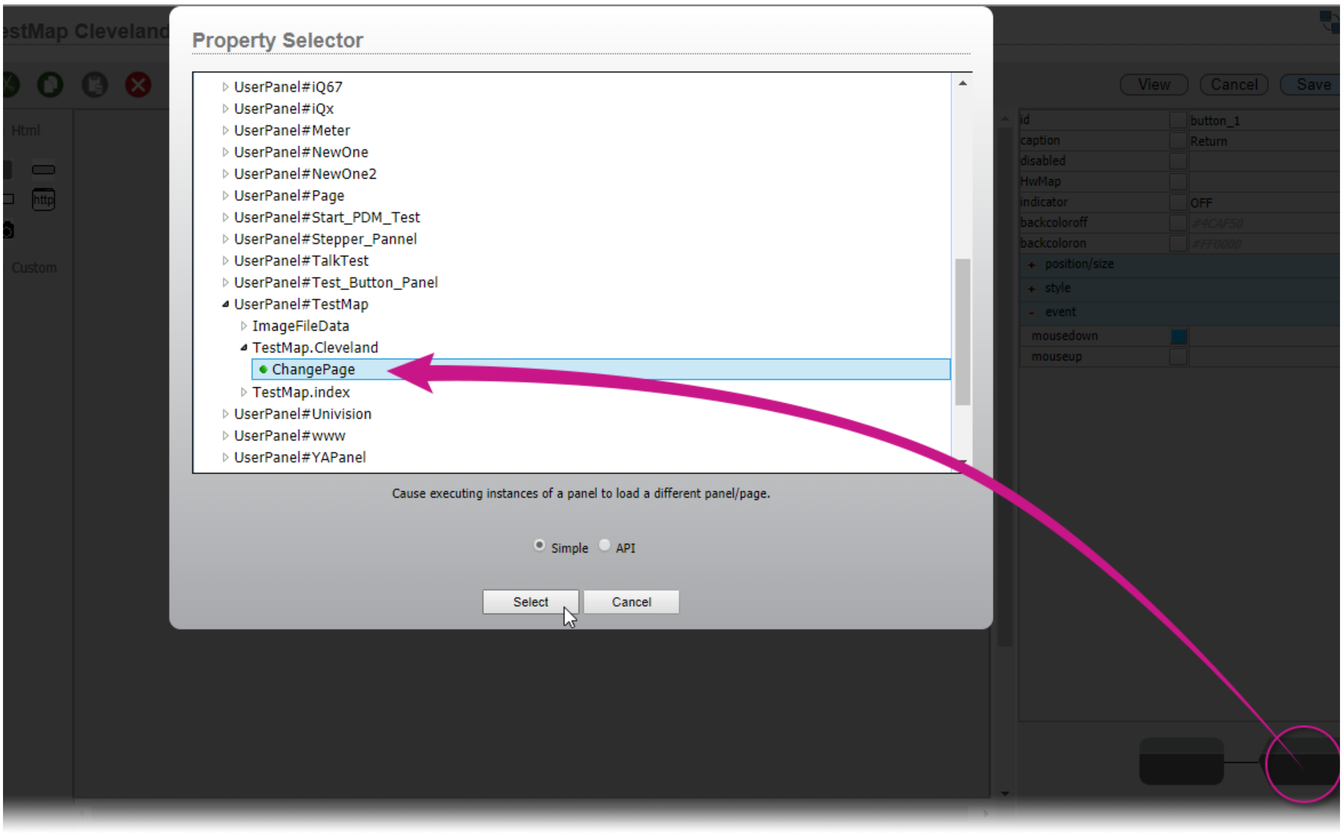
id	button_1
caption	Return
disabled	
HwMap	
indicator	OFF
backcoloroff	#4CAF50
backcoloron	#FF0000
+ position/size	
+ style	
+ event	

- With the button still selected, enable the mouse down binding



id	button_1
caption	Return
disabled	
HwMap	
indicator	OFF
backcoloroff	#4CAF50
backcoloron	#FF0000
+ position/size	
+ style	
- event	
mousedown	☑
mouseup	□

- Click the **End Point**. Expand the *UserPanels* section of the Logic Flow property selector. Expand *UserPanel#TestMap>TestMap.Cleveland* section and select the **ChangePage** property. Click **Select**



Property Selector

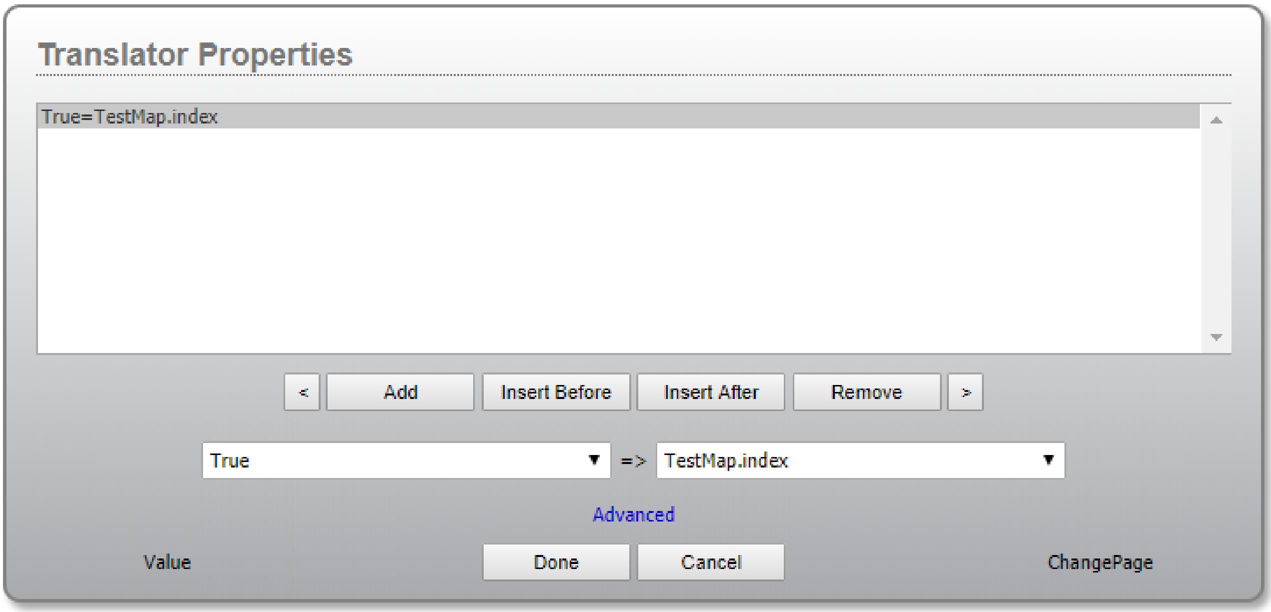
- UserPanel#iQ67
- UserPanel#iQx
- UserPanel#Meter
- UserPanel#NewOne
- UserPanel#NewOne2
- UserPanel#Page
- UserPanel#Start_PDM_Test
- UserPanel#Stepper_Pannel
- UserPanel#TalkTest
- UserPanel#Test_Button_Panel
- UserPanel#TestMap
  - ImageFileData
  - TestMap.Cleveland
    - ChangePage
  - TestMap.index
- UserPanel#Univision
- UserPanel#www
- UserPanel#YAPanel

Cause executing instances of a panel to load a different panel/page.

● Simple ○ API

> ⓘ Note: If you do not see TestMap.Cleveland in the Logic Flow property selector, it means you have not yet saved the new page.

- In the translation set **True=TestMap.index** and click **Done**

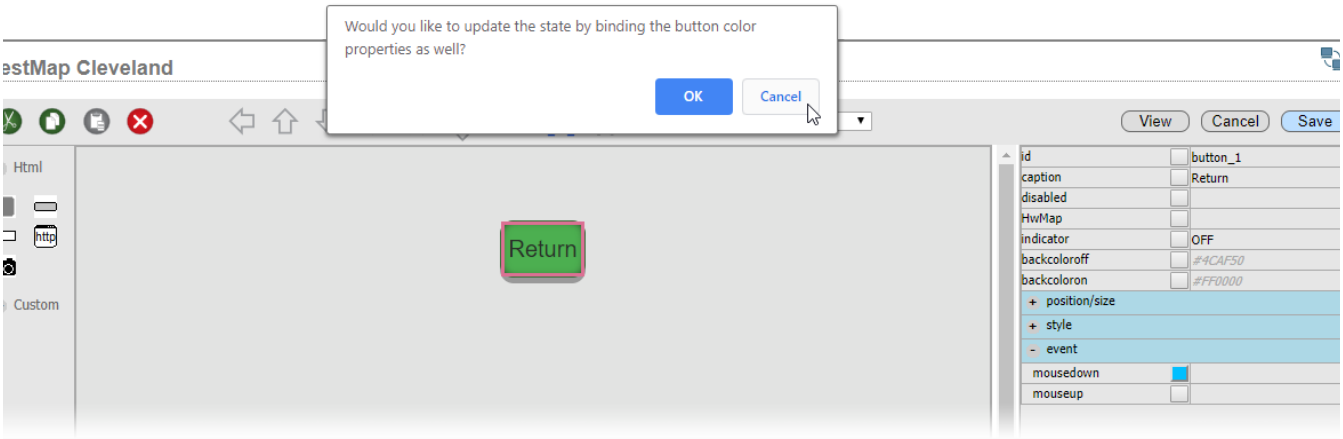


Translator Properties

True=TestMap.index

True => TestMap.index

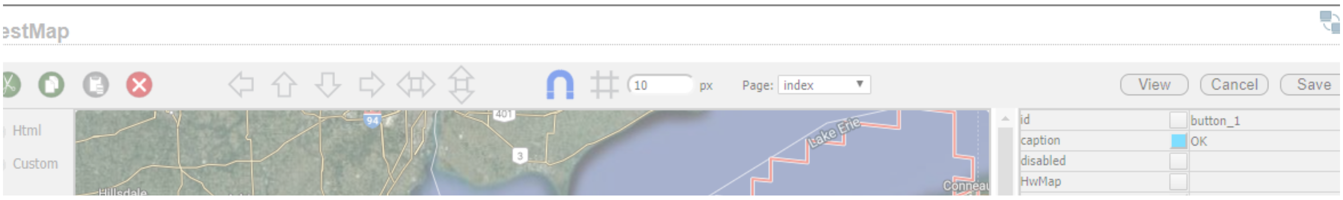
- Click **Cancel** when asked about updating the button color properties



Would you like to update the state by binding the button color properties as well?

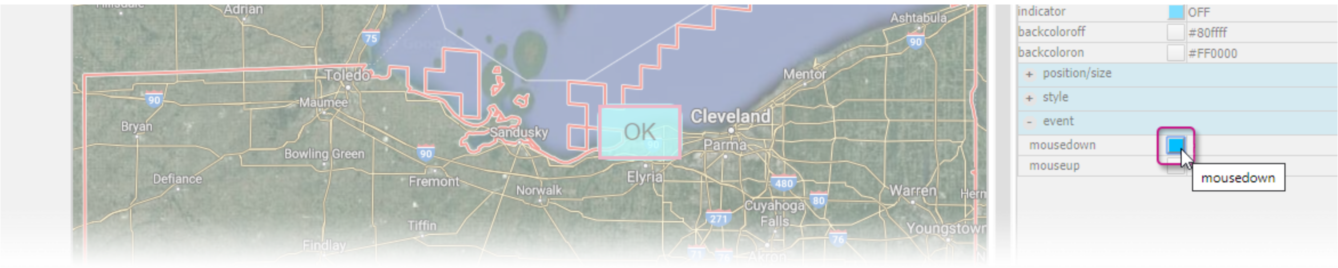
id	button_1
caption	Return
disabled	
HwMap	
indicator	OFF
backcoloroff	#4CAF50
backcoloron	#FF0000
+ position/size	
+ style	
- event	
mousedown	
mouseup	

- Save the Page. We have just defined that when we click the return button on the Cleveland page it will set the ChangePage property of the Cleveland page to **TestMap.index**, effectively returning to the index page
- Use the Page drop down to select the **index** page again
- Click the **OK** button to select it. Turn on the binding for the mouse down event



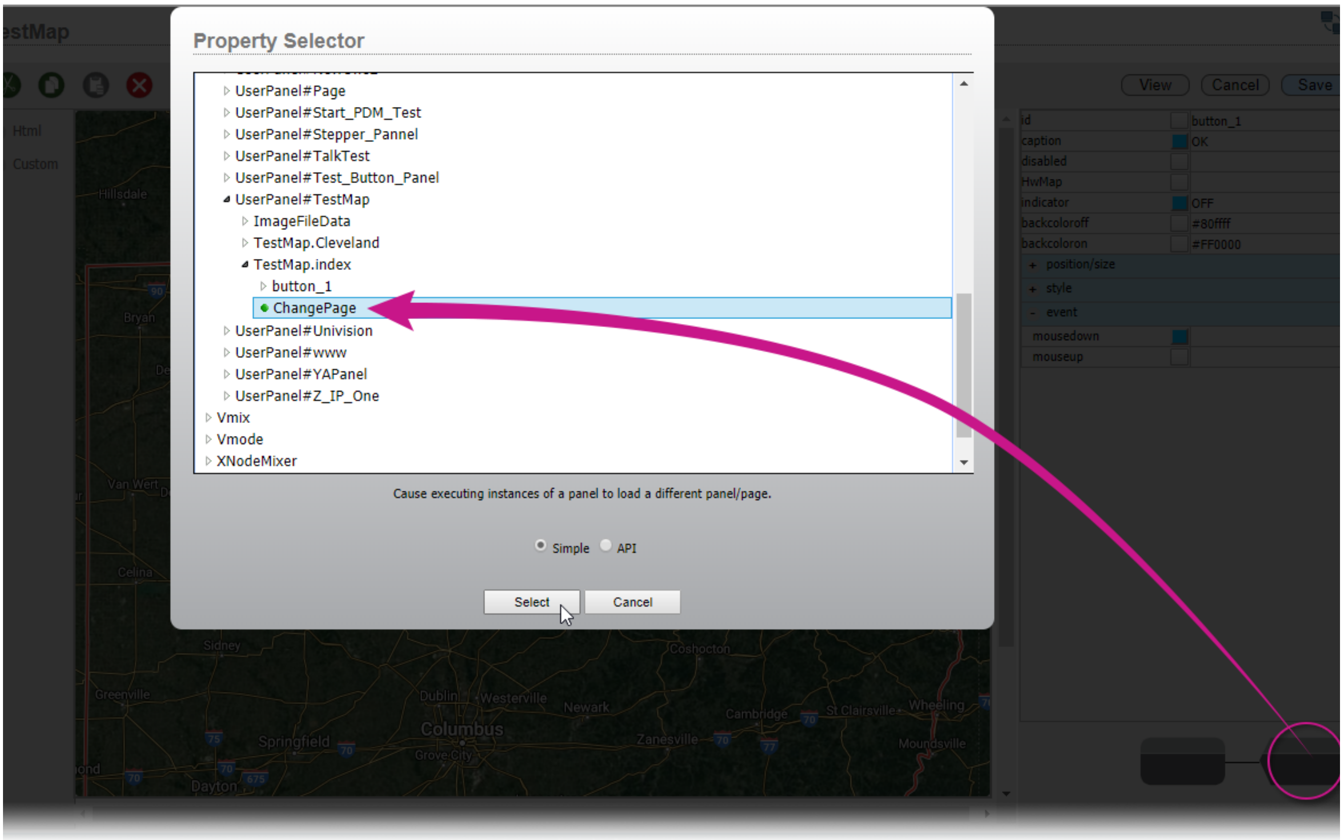
Page: index

id	button_1
caption	OK
disabled	
HwMap	



indicator	🟦	OFF
backcoloroff		#80ffff
backcoloron		#FF0000
+ position/size		
+ style		
- event		
mousedown	🟦	
mouseup		

- Click the **End Point**, select the *UserPanels>UserPanel#TestMap>TestMap.index* **ChangePage** property



Property Selector

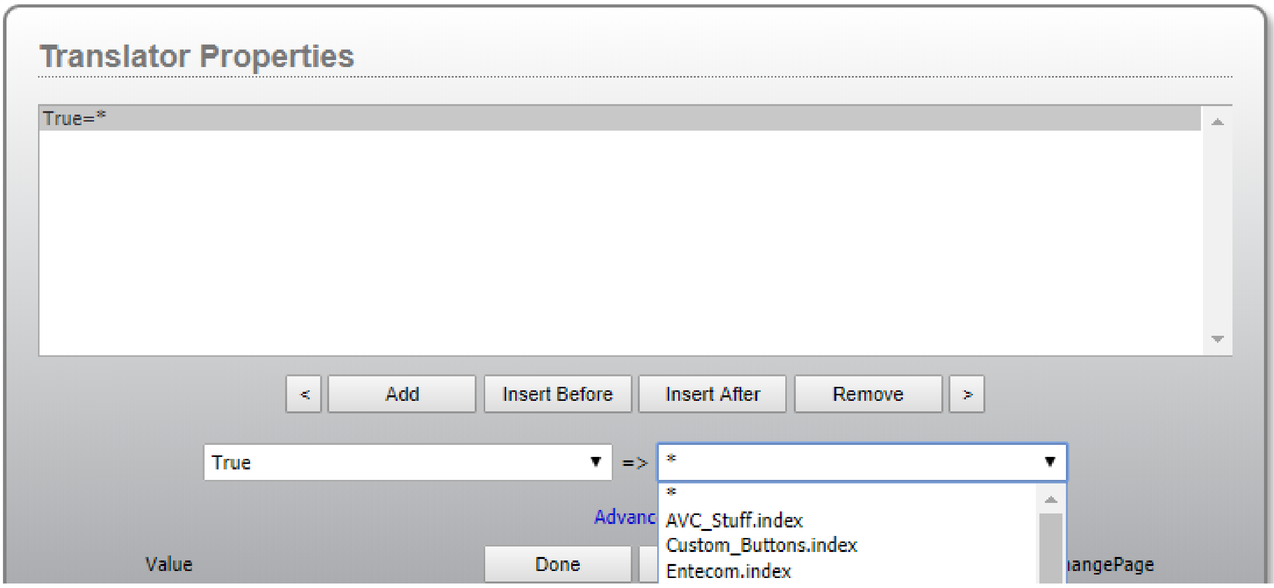
```
▹ UserPanel#Page
▹ UserPanel#Start_PDM_Test
▹ UserPanel#Stepper_Pannel
▹ UserPanel#TalkTest
▹ UserPanel#Test_Button_Panel
◢ UserPanel#TestMap
    ▹ ImageFileData
    ▹ TestMap.Cleveland
    ◢ TestMap.index
        ▹ button_1
        ● ChangePage
▹ UserPanel#Univision
▹ UserPanel#www
▹ UserPanel#YAPanel
▹ UserPanel#Z_IP_One
▹ Vmix
▹ Vmode
▹ XNodeMixer
```

Cause executing instances of a panel to load a different panel/page.

● Simple ○ API

id		button_1
caption	🟦	OK
disabled		
HwMap		
indicator	🟦	OFF
backcoloroff		#80ffff
backcoloron		#FF0000
+ position/size		
+ style		
- event		
mousedown	🟦	
mouseup		

- In the Translation set **True=TestMap.Cleveland** then click Done

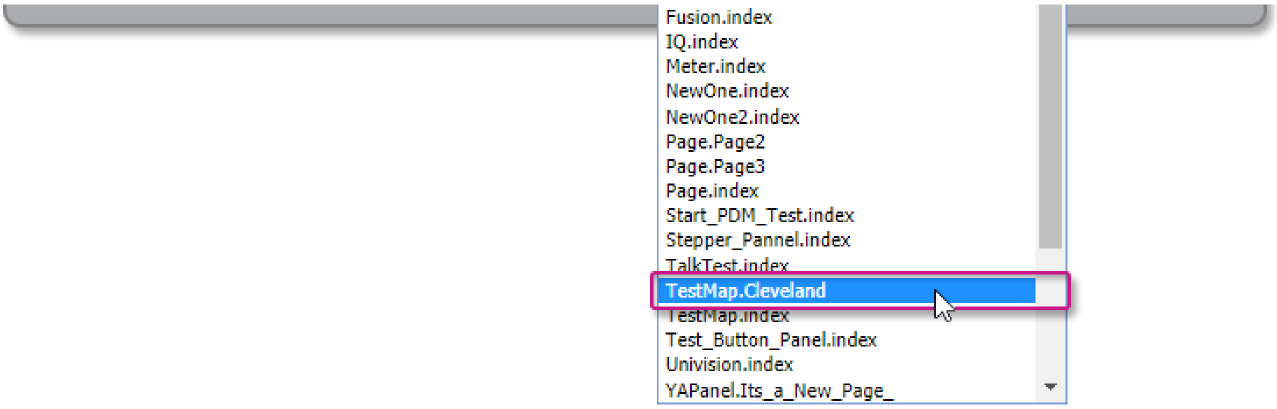


Translator Properties

True=*

True ▼ => * ▼

*
AVC_Stuff.index
Custom_Buttons.index
Entecom.index



- Click **Cancel** for the binding button color question and save the **Panel**
- Launch the Panel; the button on the main map should switch caption and color based on the silence alarm state; clicking the OK button should take you to a new Page called Cleveland, while clicking the return button should take you back to the main map index page

You are always selecting the *changepage* property of the current page and defining the Page you want to move to in the button's translation dialog.

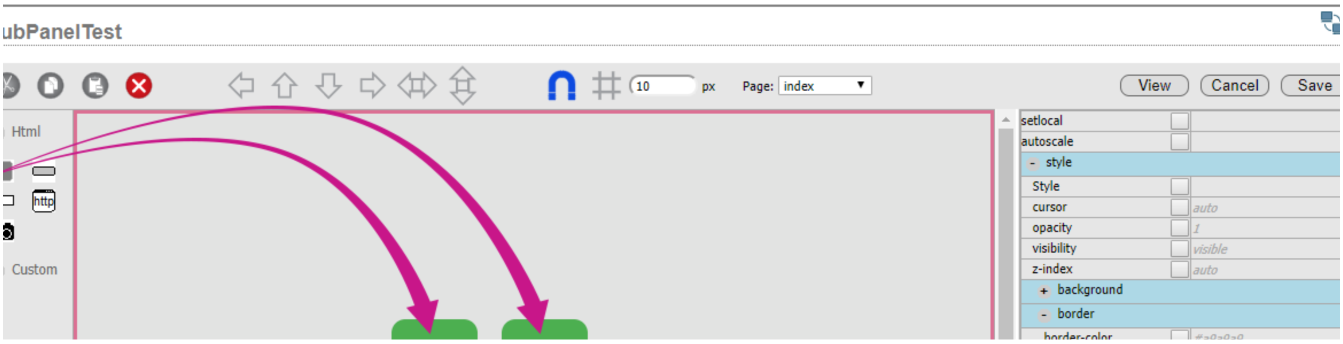
Another interesting point is that the *changepage* property is also available to normal Logic Flows. For example, rather than binding this to the *mousedown* event, we could create a flow in Logic Flows that automatically switches the Panel page to the *Cleveland* Panel whenever a silence alarm occurs and back again when audio presence is restored. Additionally, the *changepage* property is not limited to pages within the same Panel. You could switch to a completely different Panel.

In a real use case, we would fill the Cleveland page with more information than just a return button. For example, we might create a system flow chart page with meters and buttons and silence alarm states of various parts of the chain to more easily determine where the failure occurred.

SubPanels

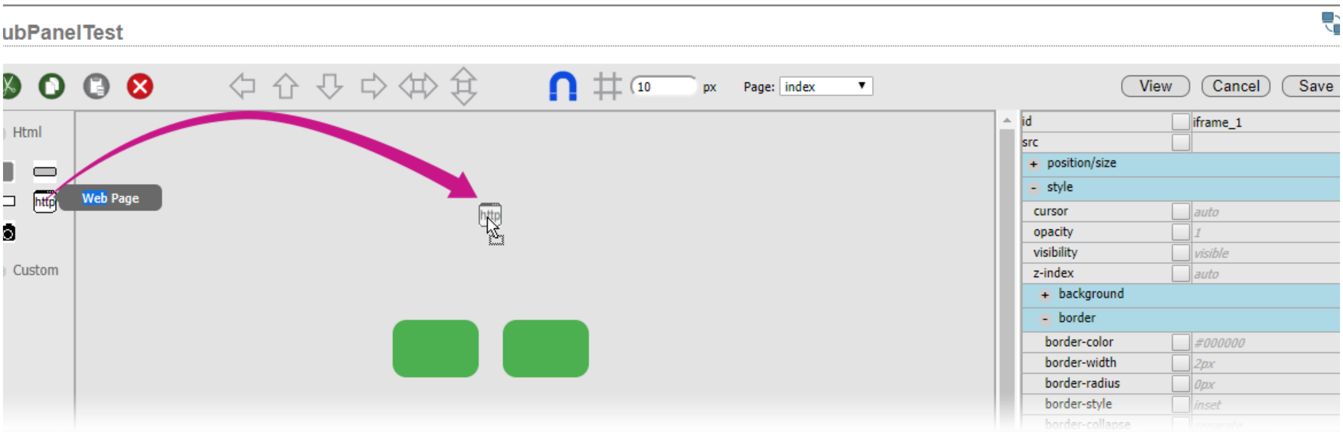
HTML5 Panels include a web page component which is essentially an HTML5 iframe component that can be used to embed other web pages and components (for example from a corporate page or video link) into a Panel. It can also be used to embed a different Panel into a parent Panel. For example:

- Create a new Panel. Save the **Panel** and name it **SubPanelTest**
- Drag two buttons into your new Panel



border-width 2px
border-radius 0px
border-style solid
border-collapse separate

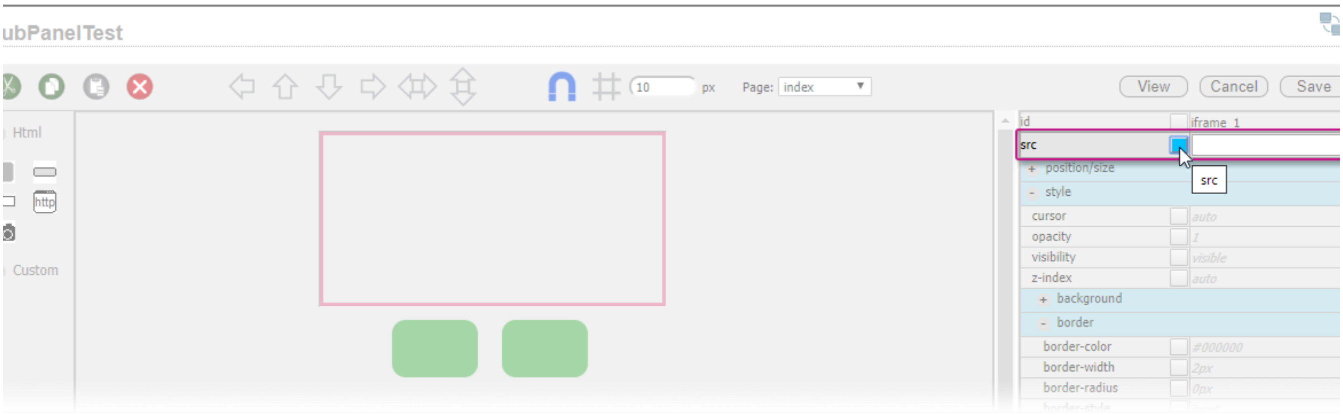
- Drag in a **web page** component



Page: index

id	iframe_1
src	
+ position/size	
- style	
cursor	auto
opacity	1
visibility	visible
z-index	auto
+ background	
- border	
border-color	#000000
border-width	2px
border-radius	0px
border-style	inset
border-collapse	separate

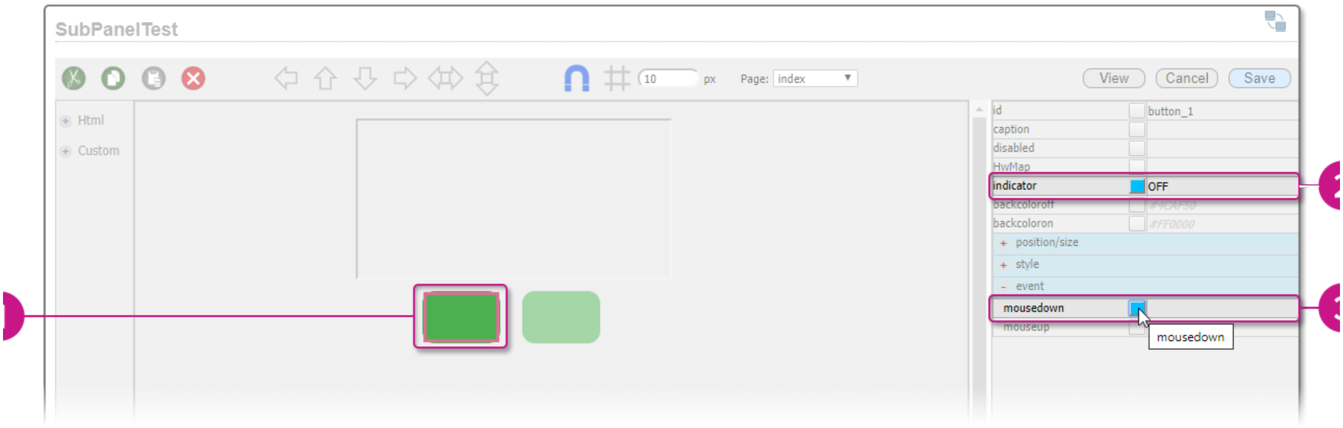
- Enable the binding on the **src** property of the web page component.



Page: index

id	iframe_1
src	
+ position/size	
- style	
cursor	auto
opacity	1
visibility	visible
z-index	auto
+ background	
- border	
border-color	#000000
border-width	2px
border-radius	0px

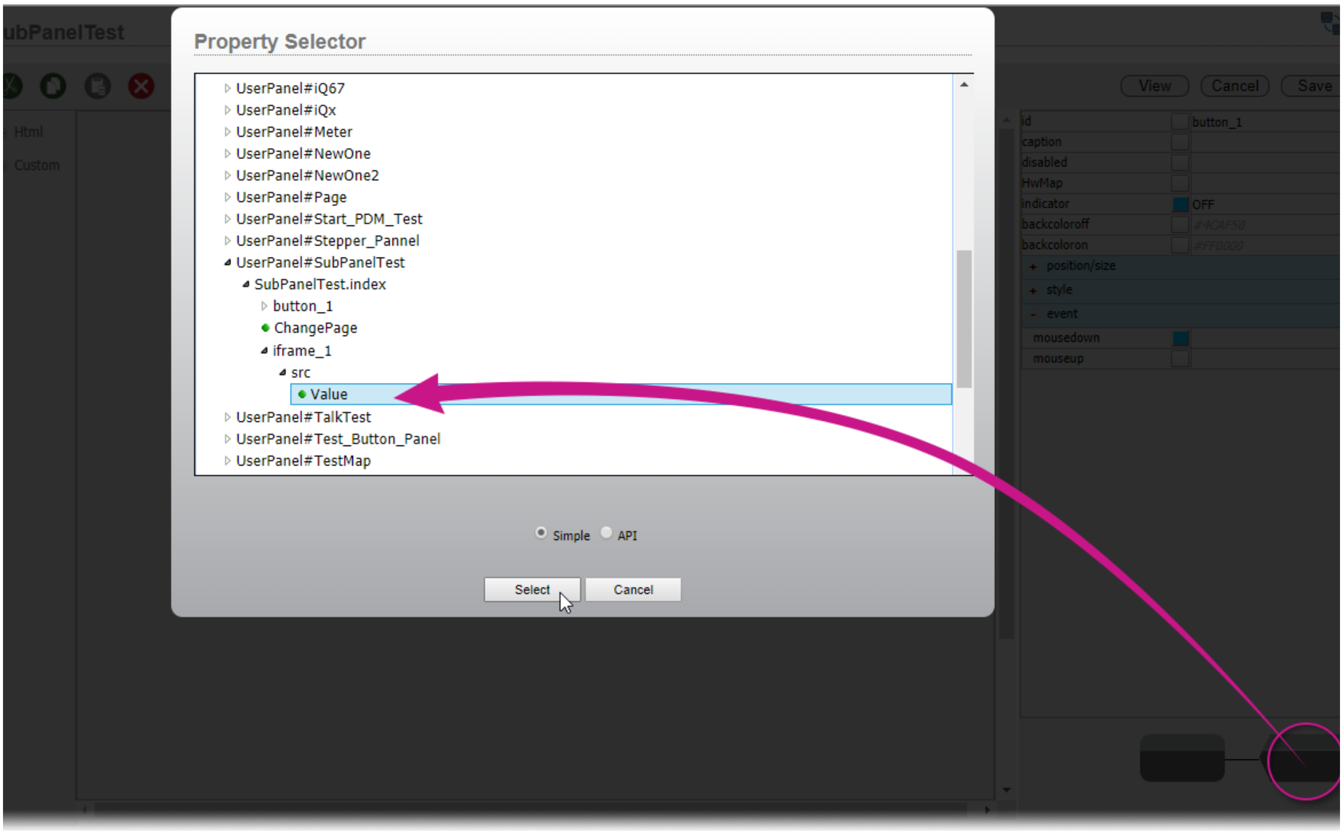
- Click the left-hand **button** to select it, enable the binding for both **indicator** and **mouse down**, and save the **Panel**.



Page: index

id	button_1
caption	
disabled	
HwMap	
indicator	OFF
backcoloroff	#4CAF50
backcoloron	#FF0000
+ position/size	
+ style	
- event	
mousedown	
mouseup	

- Select the mouse down property Flow diagram endpoint and select the iframe src property as the endpoint.



Property Selector

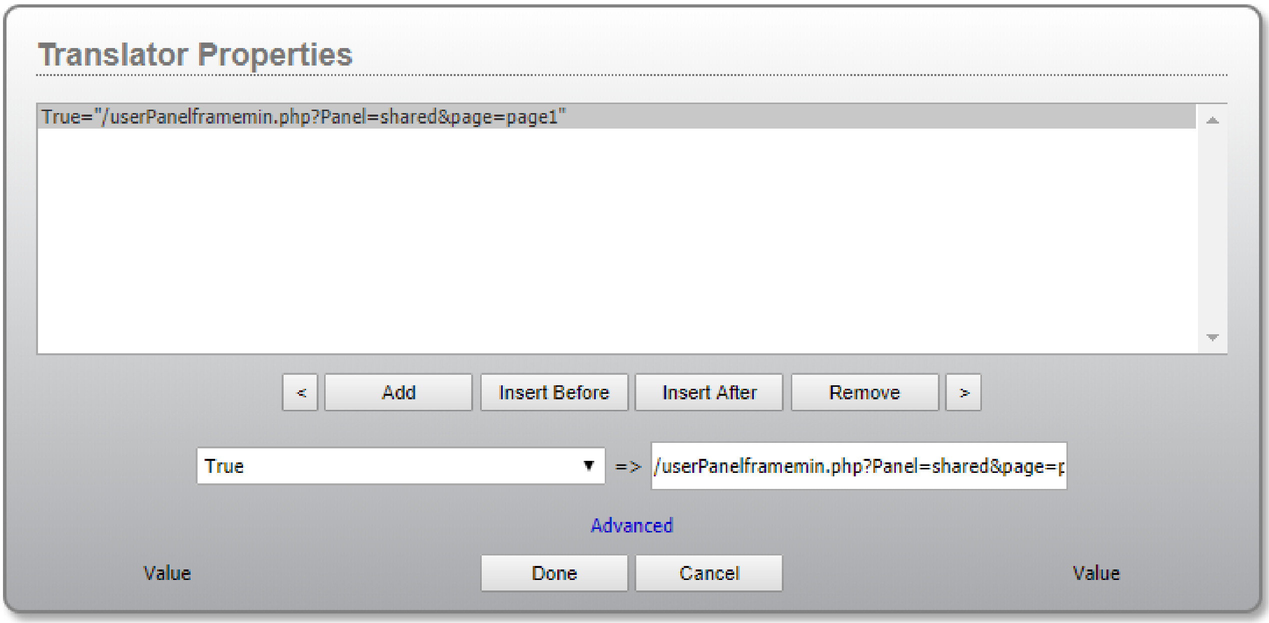
- UserPanel#iQ67
- UserPanel#iQx
- UserPanel#Meter
- UserPanel#NewOne
- UserPanel#NewOne2
- UserPanel#Page
- UserPanel#Start_PDM_Test
- UserPanel#Stepper_Pannel
- UserPanel#SubPanelTest
  - SubPanelTest.index
    - button_1
    - ChangePage
    - iframe_1
      - src
        - Value
- UserPanel#TalkTest
- UserPanel#Test_Button_Panel
- UserPanel#TestMap

○ Simple ○ API

- For the translation make the mousedown true equal to:

```
/userPanelframemin.php?Panel=shared&page=page1
```

Replacing the Panel name (shared) and page name (page1) with the name of the Panel and page you wish to display as a subPanel.



Translator Properties

True="/userPanelframemin.php?Panel=shared&page=page1"

True => /userPanelframemin.php?Panel=shared&page=p

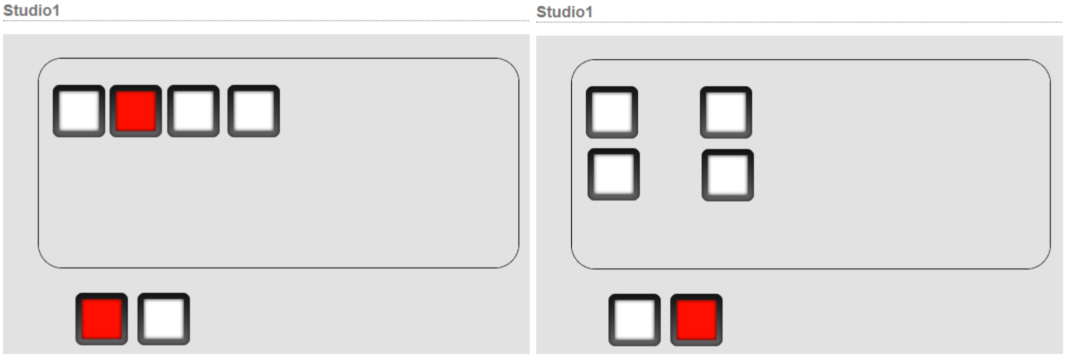
It is important to use the **framemin.php** page rather than the **frame.php** page so that the full Pathfinder Core PRO header and menu system do not appear in the subPanel.

-

Click **Done** and allow the reverse binding to be set up so the indicator will light when the selected page is loaded in the iframe.

- Repeat the procedure for the right-hand button, referencing a different page in your Translator statement.
- Save your Panel.

When the main Panel is executed, the web page component will load with the Panels specified by your buttons as subPanels. In this way we can create shared content that can be loaded in multiple Panels as shown below:



SubPanel and shared content problems

The example above works great for many kinds of shared content. For example, we could create a Panel for each studio where one of the selection buttons loads a clock and meter subPanel, and another loads an airchain display subPanel.

The challenge comes when the shared subPanel needs to target different things depending on the parent Panel.

For example, what if the subPanel buttons in the examples above were source selectors but needed to target a different destination depending whether they were loaded in Studio 1’s parent Panel or Studio 2’s? The subPanel could be a specific and even paged set of selections common across all studios, but would need to target the fader on which ever studio the subPanel was loaded in.

The temptation to solve this problem would be to create some sort of Logic Flow that changes what happens depending on which Panel last loaded the subPanel (Studio 1, 2, etc.) However, this approach does not allow for the subPanel to be loaded in both at the same time. Also, the subPanel itself has no knowledge that it is loaded as a subcomponent of another Panel.

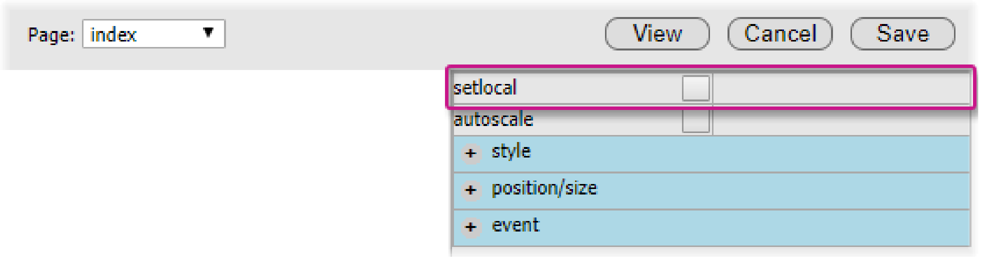
The only way to solve this problem is to introduce messaging internal to a given instance of a browser, allowing the subPanel to communicate with the parent Panel instead of with a specific destination and vice versa. That kind of messaging is the purpose of the **Panel SetLocal** and **PanelMemorySlot**.

Panel SetLocal

> ⓘ Important Note: This is considered an advanced feature and is often used in tandem with the

> PanelMemorySlot feature.

Each HTML5 Panel includes a write only property called **SetLocal**. Selecting the overall Panel will display this property in the property tree:



This property sends messages to set another property on the running instance of the Panel, its parent, or a subPanel. Like the *PanelMemorySlots* property, this property operates inside a single, specific instance of a Panel rather than upon all running instances of a given Panel.

In general, this property's value will not be set in the Panel designer. Instead, the binding will be enabled and then can be used or bound to other items in the Panel via the Panel's internal flows. The syntax that makes this possible is:

<target>|<elementId>|<propertyName>=<PropertyValue>

For example, you could enable the binding of this property and then when a button is pressed you can set the value of this property to:

iframe_1|MyButton|Indicator=ON

The target portion can either be:

- parent = assumes this Panel is loaded as a webbrowser (iframe) subPanel of another Panel and we wish to send a property change to the parent Panel.
- local = tries to set a property on the local instance of the Panel
- name of a web browser (iframe) element in the form = tries to set a property on an element in a Panel running in a web browser element in the existing Panel. This variation looks for an iframe in the current Panel of the correct name and then passes the property message to the web page running in that iframe.

Examples:

- parent|MyButton|Indicator=ON
  - Attempts to turn the Indicator property On for a Button named MyButton that is expected to reside on a Panel in which this Panel is running as a subPanel.
- local|MyButton|Indicator=ON
  - Attempts to turn the indicator property On for a button named My Button that is expected to reside on this instance of the Panel.
- Iframe_1|MyButton|Indicator=ON
  - Attempts to turn the indicator property On for a button named MyButton that is expected to reside on

a Panel loaded in a web browser (Iframe) element called Iframe 1.

It is important to understand that **SetLocal** sets properties only in this instance of the browser.

If you have the Studio 1 Panel loaded on two different computers and you perform an action that uses the **SetLocal** property to change an element's property, that change will only happen on that computer's instance of the Panel unless other external bindings are also in use.

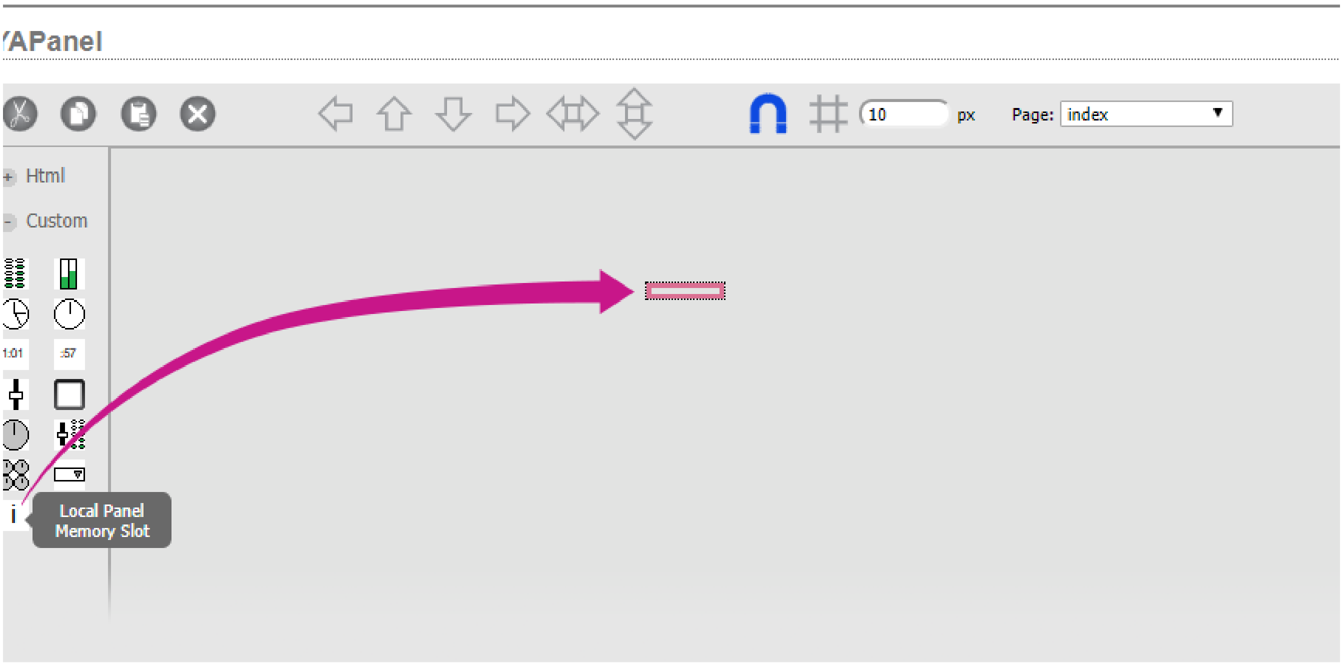
If the change is bound to a component inside the Panel it will remain inside that instance of the Panel, but if it is bound to something like a route change that is outside of the Panel it will likely happen on all. This can be seen because using this property as a binding to a component inside the Panel will not generate a Panel based Logic Flow and binding it to something outside the Panel will. The nuances of this are a bit subtle but should become clear as we work through the example below.

Panel Memory Slots

> ⓘ Important Note: This is considered an advanced feature and is often used in tandem with the Panel *SetLocal* feature.

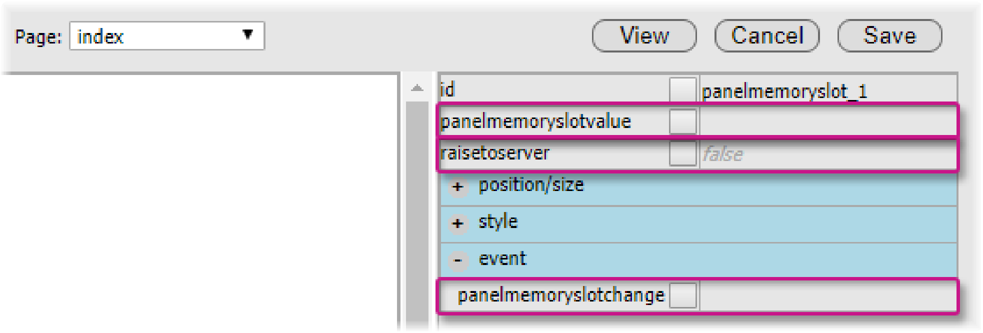
Panel Memory Slots are like normal memory slots except that they only live inside a browser instance of a Panel. They can be thought of as a javascript variable that also raises a change event inside the running instance of the Panel.

The item can be found in the custom tool section. Dragging it onto a Panel will create a dotted component. This component will be invisible when the Panel is actually executing.



Like the *SetLocal* property above, if this is bound using a Panel Flow to an item inside the Panel (button, iframe, etc), no flow in the larger pathfinder will be created. Instead the value will just be changed or affect change within the running instance of the Panel only. If the Flow is bound to an item outside the Panel (for example a route change of console fader state), then a Flow will be created.

The Panel Memory slot has three important properties/events in the property grid:



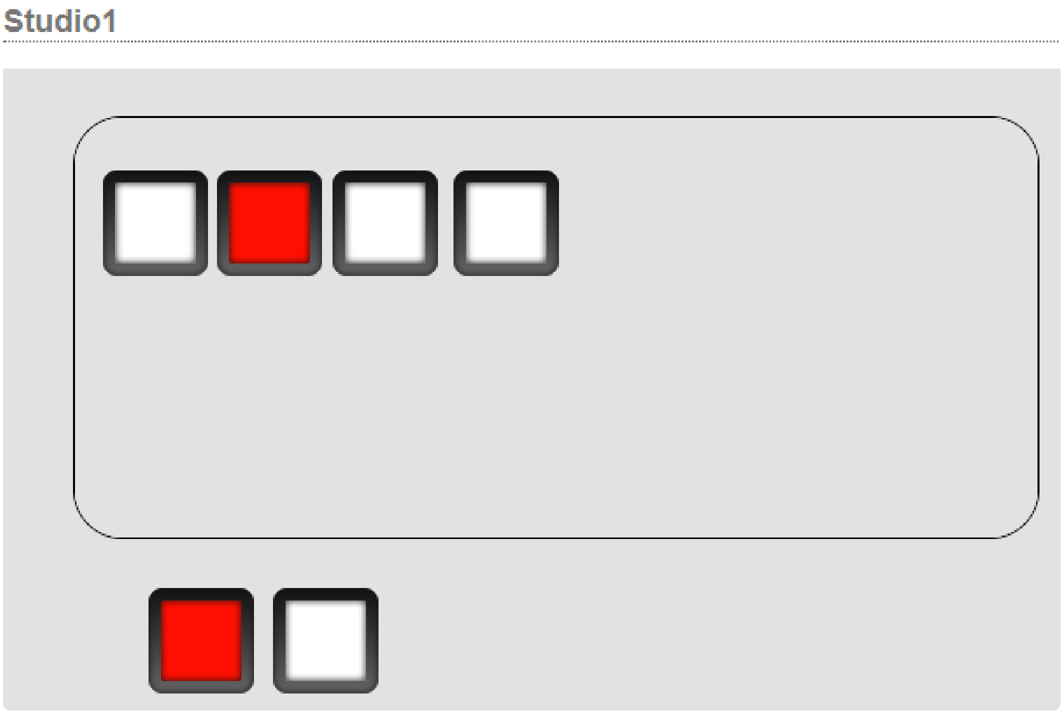
Page: index

id	panelmemoryslot_1
panelmemoryslotvalue	
raisetoserver	false
+ position/size	
+ style	
- event	
panelmemoryslotchange	

Property	Notes
panelmemoryslotvalue	The value of the Panel memory slot.
raisetoserver	Used to determine whether change events are raised only inside the local instance of the Panel also to the server.
panelmemoryslotchange	An event that fires when the value changes. This event carries the new value rather than just a true or false as its data.

Shared SubPanel Example

Returning to our example above:



In this example, what if the four buttons in the SubPanel are tied to sources on destinations dependent on which studio's parent Panel the subPanel is loaded in. To accomplish this, we are going to use three Panel Memory Slots: two in the Studio1 parent Panel, and one in the SubPanel. Our configuration might look like this:

Studio1 Panel

In addition to general Panel settings, we need to create two Panel Memory Slots: *currentsource* and *selectsource*.

Memory slot: currentsource

This memory slot will pass the *currentsource* value of a virtual router destination to a specific subpanel instance. This is accomplished using these bindings:

- ***panelmemoryslotvalue*** will be bound to the *currentsource* property of a virtual router destination using a Panel flow and a translation like this: *=*

This means whenever the *currentsource* property of that virtual router destination changes, the *currentsource* value will be assigned to this memory slot.

- ***panelmemoryslotchange*** will be bound to the *setlocal* property of the Studio1 Panel using a translation like this: *=iframe_1|currentsource|Panelmemoryslotvalue=*

This means whenever the *currentsource* value changes, it will attempt to also set the same value on whatever Panel is running in the subPanel loaded to iframe_1.

Memory slot: selectsource

This memory slot will change the route on the destination whenever its value changes.

- ***panelmemoryslotchange*** will be tied to the same virtual router destination's current source property.
- ***raisetoserver*** needs to be set to **true**.

General Settings

- ***setlocal*** needs to be enabled.

Shared SubPanel – page 1

In this shared SubPanel, in addition to general Panel settings, we need to create one Panel Memory Slots: *currentsource*.

Memory slot: currentsource

This memory slot will receive and match the current source from the parent's *currentsource panelmemoryslotchange* change event. This is accomplished using these bindings:

- ***panelmemoryslotvalue*** must be enabled.
- ***panelmemoryslotchange*** must be enabled.

***** General Settings**

-

- ***setlocal*** needs to be enabled.
- We will bind each of the button indicators to the value of currentsource with a translation like:
  - 1=On
  - *=Off
  - Where 1 is the source whose button indicator we want to light
- When the Panel is running, this will cause the subPanel to light the correct button depending on which source number has been fed to it from the parent.
- Last we will bind the button press to the Shared Panel's setlocal property with a translation that looks like:
  - True=parent|selectsource|panelmemoryslotvalue=1
  - Where 1 is the source whose button has been pressed.

- This will allow the button press to pass the source selected to the parent Panel's selectsource memory slot. Which if we remember is bound via its change event to the actual destination's currentsource.

The interesting part of the steps above is that we can now clone Panel 1 to Panel 2. In Panel 2 we change the bindings on currentsource and selectsource to a different destination for Studio2. And we make sure that the Panel selection buttons point to studio 2's iframe rather than studio 1's. But the shared Panel does not need any changes. It will just start working for studio 2 and studio 2's destination. More importantly if we need to add functionality to the shared component it will work for both studios.

The key thought process that needs to be utilized is that if the shared content manipulates something specific then none of these steps are necessary. But if the shared content needs to manipulate something different depending on the parent Panel it is loaded in, then we need to pass messaging between the internal Panel and the parent to accomplish that.

This is one example of how these features may be used. It could be expanded greatly. For example, the src property of the iframe could be assigned using a setlocal so that it does not need to be updated for each studio page. Or we could add additional Panel memory slots for manipulating other destinations or even dynamically selecting the destination at the parent level.

Additionally, there are future features planned for embedded routing components that may make this specific example less necessary. However, even if we add such components there will always be situations that are custom and do not work with our preconceived components.

It should also be pointed out that in many situations it is probably simpler, easier, and more understandable to just clone. Because this is a more advanced feature it is also more advanced to understand. However, for the times when the embedded content is duplicated in too many places and needs to be periodically changed, this allows for that kind of reuse.

> ⓘ Important Note: to see a video presentation of the functionality described below visit this link: http://pathfinderpc.com/pfcorepro_downloads/reusablesubpanels.mp4

Prior to Pathfinder Core PRO’s HTML 5 Panels, User Panels could be created using a variation of PanelDesigner from the original Pathfinder PRO product. As this functionality is deprecated, we would urge all users to migrate to the much more powerful HTML5 Panels.

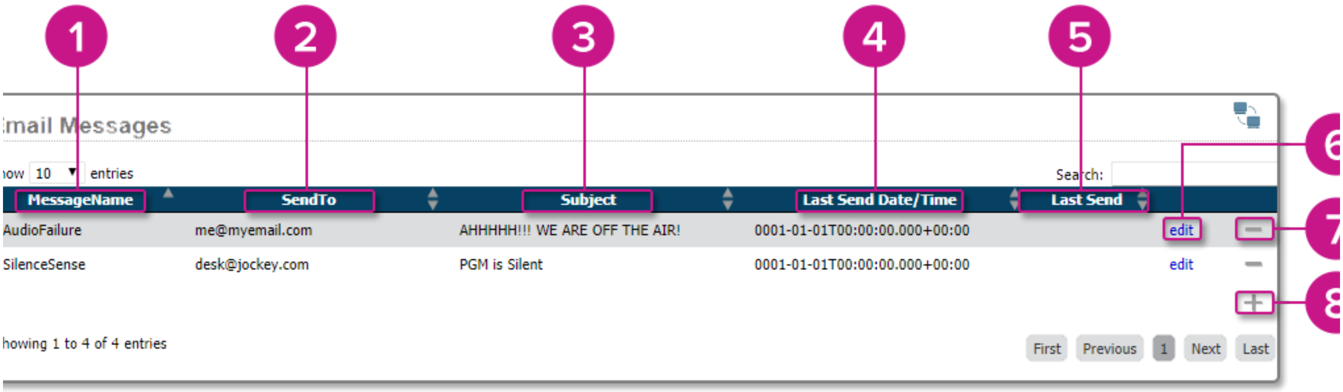
The Panels created in the original Pathfinder PRO are called Legacy User Panels and are not interoperable with the new HTML 5 Panels. For example, you cannot open an HTML5 Panel in PathfinderPC/Mini and you cannot open a Legacy User Panel in a web browser.

E-mail Messages

Logic Flows can be used to send e-mails when conditions exist in your system that require attention. For example, it is common to send e-mails to the station’s engineers when silence is detected at a transmitter site.

To send e-mails, you must first configure the e-mail host settings under the System navigation section. If you have not configured these settings, please review the Email Settings section of the System chapter.

Once the e-mail host has been configured and test e-mails can be sent successfully, click the **Email Messages** link in the Navigation bar. This is where you can create e-mail messages that can be sent when specific actions take place in the system.



MessageName	SendTo	Subject	Last Send Date/Time	Last Send		
AudioFailure	me@myemail.com	AHHHHH!!! WE ARE OFF THE AIR!	0001-01-01T00:00:00.000+00:00		edit	—
SilenceSense	desk@jockey.com	PGM is Silent	0001-01-01T00:00:00.000+00:00		edit	—

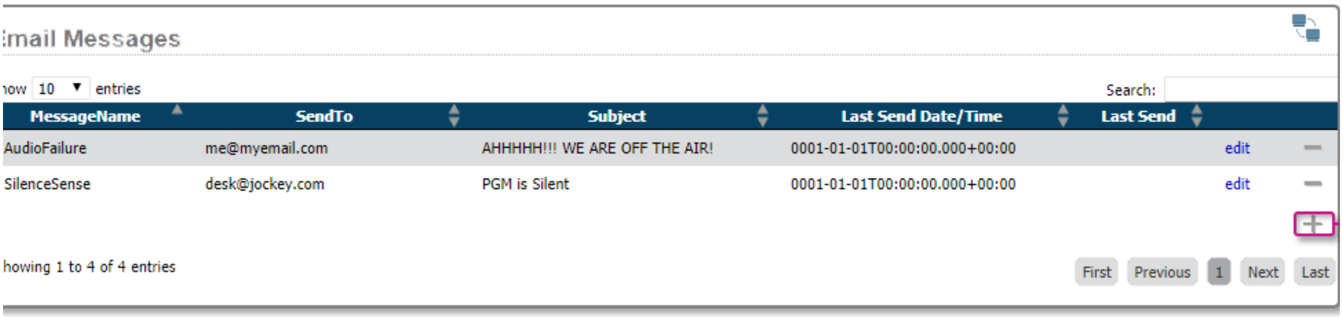
howing 1 to 4 of 4 entries

Reference	Notes
1	The unique **name** of the e-mail message.
2	The SendTo **e-mail addresses** to which the message will be sent.
3	The **Subject** line to be used by the e-mail message.
4	Displays the last **time** the message was sent.
5	Displays information about whether the **last send** attempt was successful or failed.
	Click **edit** to modify the associated e-mail

6	message’s properties.
7	Click the **minus** icon to remove the associated e-mail message.
8	Click the **plus** icon to add a new e-mail message.

Adding a New Email Message

- Click the **plus** icon to create a new email message



mail Messages

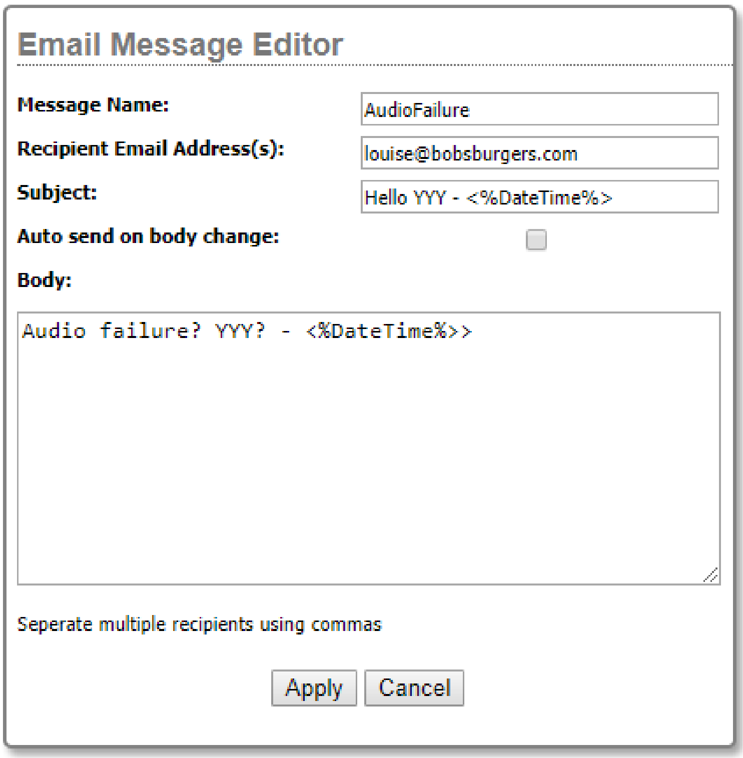
ow 10 entries

Search:

MessageName	SendTo	Subject	Last Send Date/Time	Last Send		
AudioFailure	me@myemail.com	AHHHHH!!! WE ARE OFF THE AIR!	0001-01-01T00:00:00.000+00:00		edit	−
SilenceSense	desk@jockey.com	PGM is Silent	0001-01-01T00:00:00.000+00:00		edit	−

howing 1 to 4 of 4 entries

- Complete the fields on the Email Message Editor



Email Message Editor

Message Name: AudioFailure

Recipient Email Address(s): louise@bobsburgers.com

Subject: Hello YYY - <%DateTime%>

Auto send on body change: □

Body:

```
Audio failure? YYY? - <%DateTime%>>
```

Seperate multiple recipients using commas

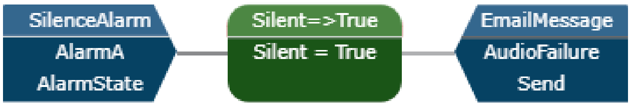
Field	Notes
Message Name	Type a **Name** for this email message. This name will be used when this message is accessed via

	Logic Flows or the API.
Recipient Email Address(s)	Type a comma-delineated list of **e-mail addresse** to which this message will be sent.
Subject	Type a **Subject** line for generated emails.
Auto send on body change	When **checked**, you have the option to define ne body content in your e-mail via a Logic Flow, and have it send automatically without having to set th **Send** property.
Body	Type the **text** of the e-mail message. You can embed the date and time the message was sent into the subject or body of the email using the variable **<%DateTime%>**. In the example above, when the email is created, the Subject like would include the current date and time like this: Hello YYY – 2017-12-14T13:51:28.033-05:00 While the Body of the message would look like th Audio failure? YYY? - 2017-12-14T13:51:28.033-05:00

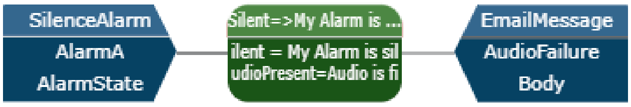
- When all fields are complete, click **Apply** to save your changes

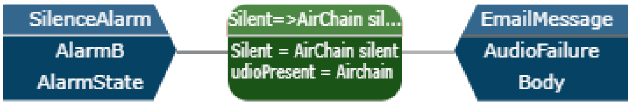
Using E-mail Messages in Logic Flows

Once created, e-mail messages become available for use in Logic Flows. In the following example, we will send the *AudioFailure* email message whenever MyAlarm's *AlarmState* becomes *Silent*. We accomplish this by setting the write-only property *Send* to the value *True*.



You can also change the body of the e-mail message as a Logic Flow end point. This becomes especially useful if you have several conditions that could cause an e-mail alert. For example, we could create a generic e-mail message making sure that the *AutoSend on body change* checkbox is engaged. Then we could create multiple Logic Flows, each of which alters the body of the message depending on the start point. For example:





Here we use two Logic Flows to send four different messages to the same e-mail recipients.

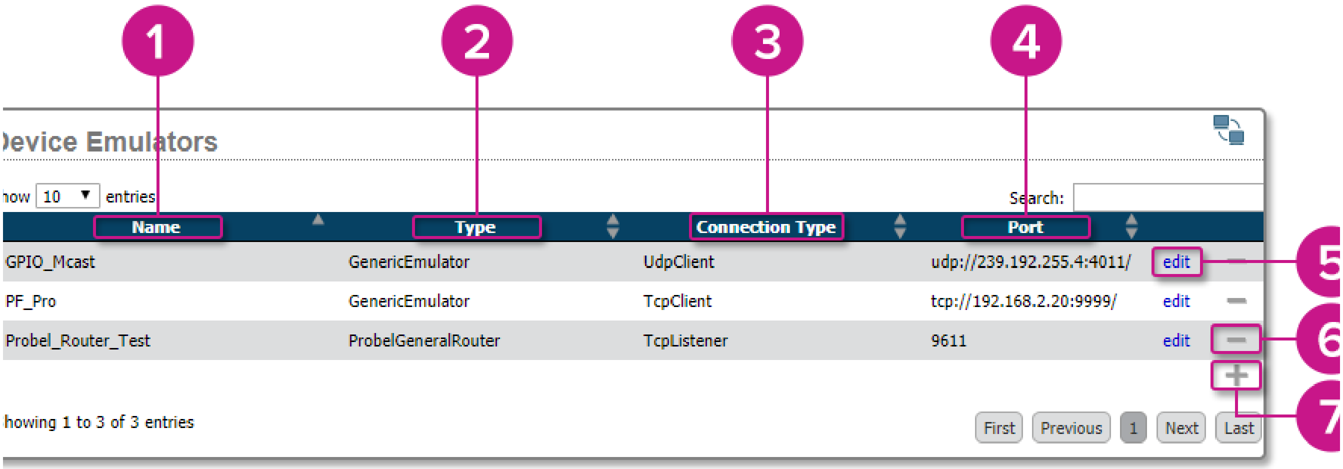
- First Flow
  - If “AlarmA” is Silent -> Set the body of the “AudioFailure” message to be “My Alarm is Silent.”
  - If “AlarmA” has audio -> Set the body of the “AudioFailure” message to be “My Alarm is fine.”
- Second Flow
  - If “AlarmB” is Silent -> Set the body of the “AudioFailure” message to be “AirChain Silent.”
  - If “AlarmB” has audio -> Set the body of the “AudioFailure” message to be “AirChain OK.”

Since we engaged the option to send the e-mail each time the body property of the e-mail changes, this is all that we need to do to handle four alarm conditions using one defined e-mail message.

Device Emulators

Pathfinder Core PRO can emulate other routing systems. This can be particularly useful if you need to interface with an automation system that has not implemented any of the protocols that drive Axia routing systems.

Click the **Device Emulators** link in the navigation bar to display the list of device emulators currently defined in the system.



evice Emulators

how 10 entries

Search:

Name	Type	Connection Type	Port		
GPIO_Mcast	GenericEmulator	UdpClient	udp://239.192.255.4:4011/	edit	−
PF_Pro	GenericEmulator	TcpClient	tcp://192.168.2.20:9999/	edit	−
Probel_Router_Test	ProbelGeneralRouter	TcpListener	9611	edit	−

+

howing 1 to 3 of 3 entries

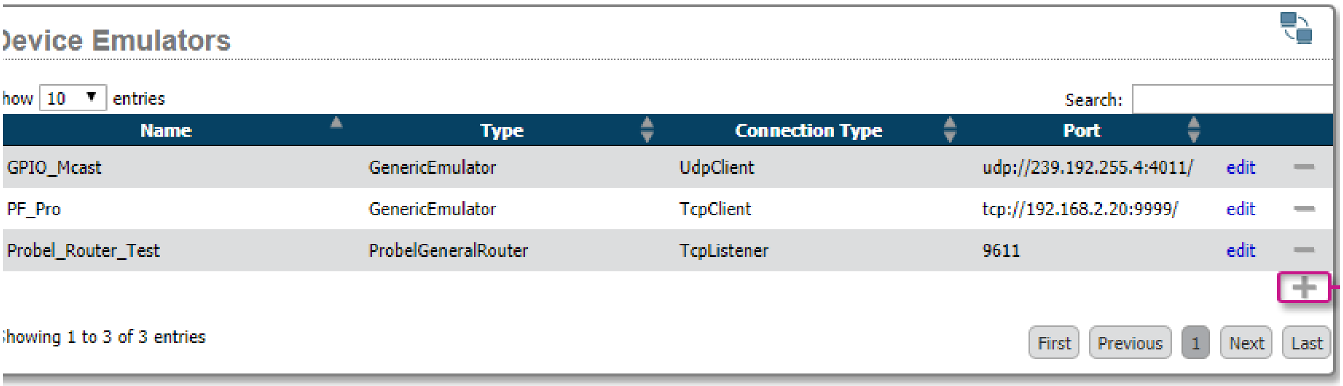
First Previous 1 Next Last

Reference	Notes
1	The **name** of the emulator.
2	The **type** of the emulator (*Generic Emulator*, *Prob General Router*, *Probel General Switcher*).
3	The **connection type** to be used with the emulatc

4	The **Port** used for the connection
5	Click **edit** to modify the associated emulator's properties.
6	Click the **minus** icon to remove the associated emulator.
7	Click the **plus** icon to add a new emulator. Pathfinder Core PRO currently supports three typ of device emulation, each presenting slightly different configuration options depending on the emulator type.

Creating a Generic Emulator

- Click the **plus** icon in the Device Emulators page to create a new emulator



evice Emulators

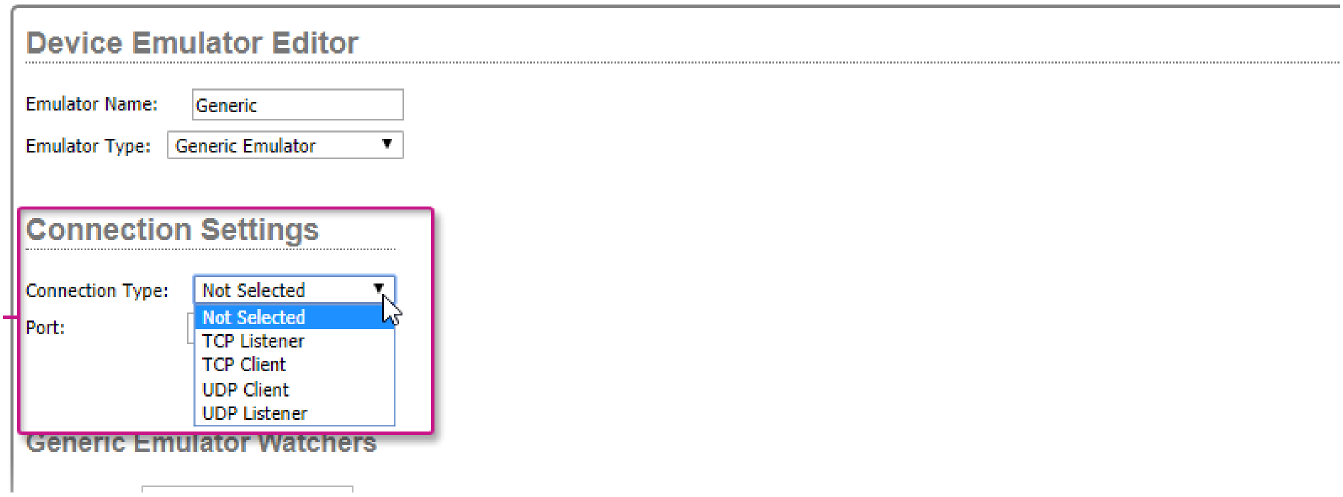
how 10 entries

Search:

Name	Type	Connection Type	Port		
GPIO_Mcast	GenericEmulator	UdpClient	udp://239.192.255.4:4011/	edit	—
PF_Pro	GenericEmulator	TcpClient	tcp://192.168.2.20:9999/	edit	—
Probel_Router_Test	ProbelGeneralRouter	TcpListener	9611	edit	—

howing 1 to 3 of 3 entries

- Type a unique **Name** for this emulator and select **Generic Emulator** from the *Emulator Type* drop-down list
- Complete the **Connection Settings;** the Generic Device Emulator is just a TCP/UDP connection that can be used to send and receive custom messages from an automation system or some other third-party system



Device Emulator Editor

Emulator Name: Generic

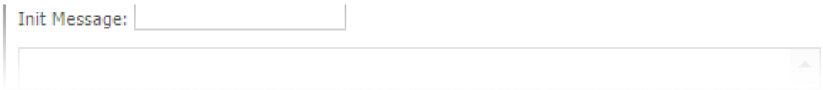
Emulator Type: Generic Emulator

Connection Settings

Connection Type: Not Selected

Port:

Generic Emulator Watchers



Init Message:

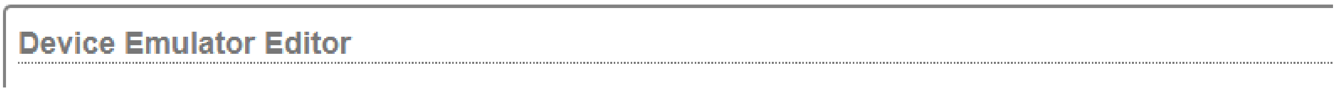
Field	Description
Connection Type	From the drop-down list, select either **TCP/UDP Listener** or **TCP/UDP Client**, depending on whether: • The system using this emulator will establish the connection to Pathfinder Core PRO (an *inbound* connection—select **TCP/UDP Listener**) or, • Pathfinder Core PRO should initiate the connection to the other device (an *outbound* connection—select **TCP/UDP Client**.)
Port	If the other device is initiating the connection to Pathfinder Core PRO (Pathfinder Core PRO is the **TCP/UDP Listener**) type the **port** on which Pathfinder Core PRO will listen for the connection If Pathfinder Core PRO needs to initiate and maintain the connection (Pathfinder Core PRO is the **TCP/UDP Client**) type the **IP address** and **po** of the device to which Pathfinder Core PRO shou connect.

> ⓘ **Important Note:** Pathfinder Core Pro does not currently support serial ports. If you need serial port control, investigate serial-to-TCP converter solutions, or use Pathfinder’s Port Router application to bridge the serial data to TCP data.

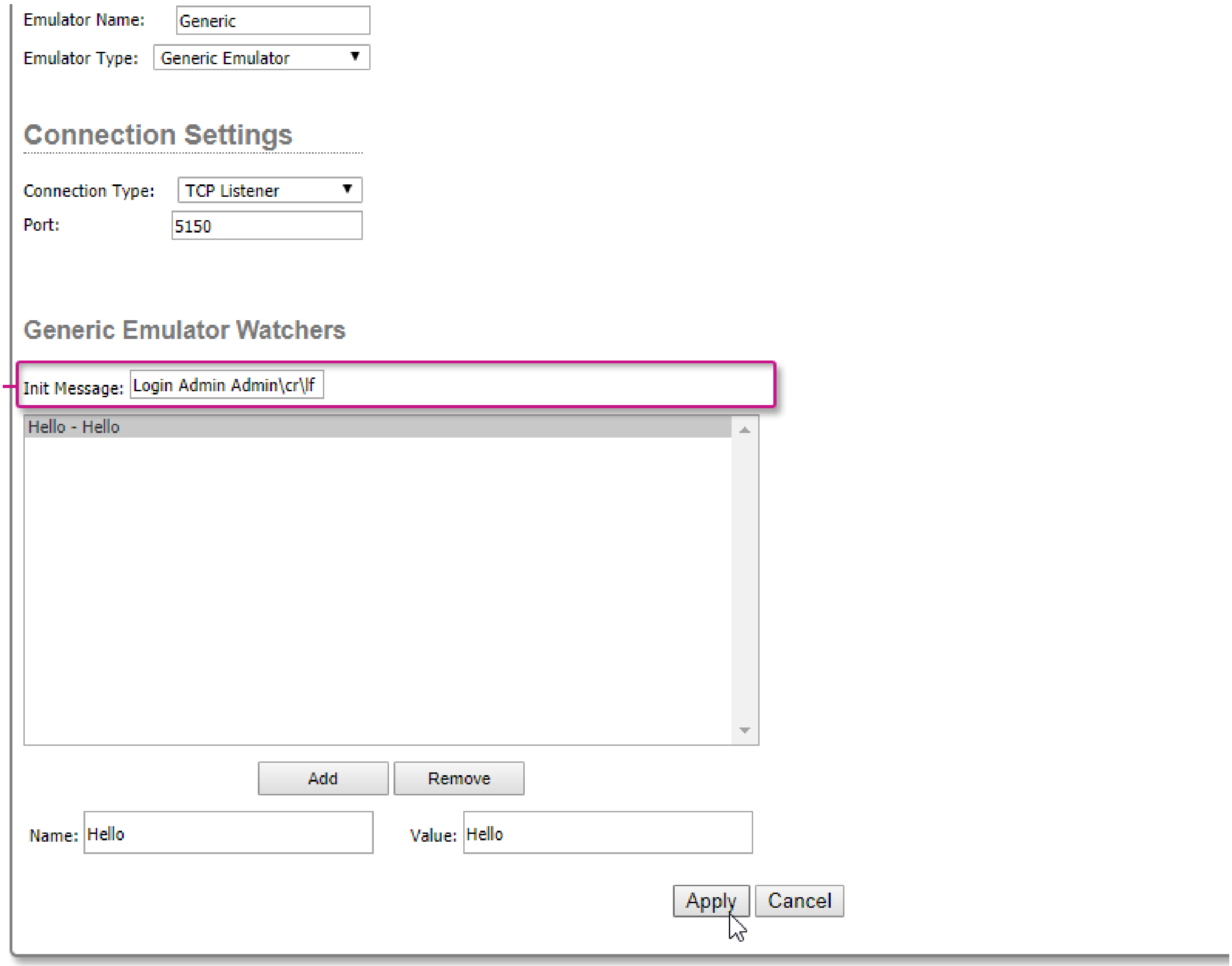
- Type your initialization message in the **Init Message** field; this message will be sent when this Generic Emulator connects allowing the emulator to automatically send login or other initialization messages, including special escape characters (see the *Escape Characters* section later in this chapter)

  In the case of *Listener* types, this message will be sent to each client that connects to the listener. In the case of *Client* connection types, this message will be sent each time the client connects.

  Generic Device Emulators have a write-only property that can be used by Logic Flows called **ToSend**. This property allows you to create a flow to send any kind of data out of the Generic Emulator you wish.



Device Emulator Editor


Emulator Name: Generic

Emulator Type: Generic Emulator

Connection Settings

Connection Type: TCP Listener

Port: 5150

Generic Emulator Watchers

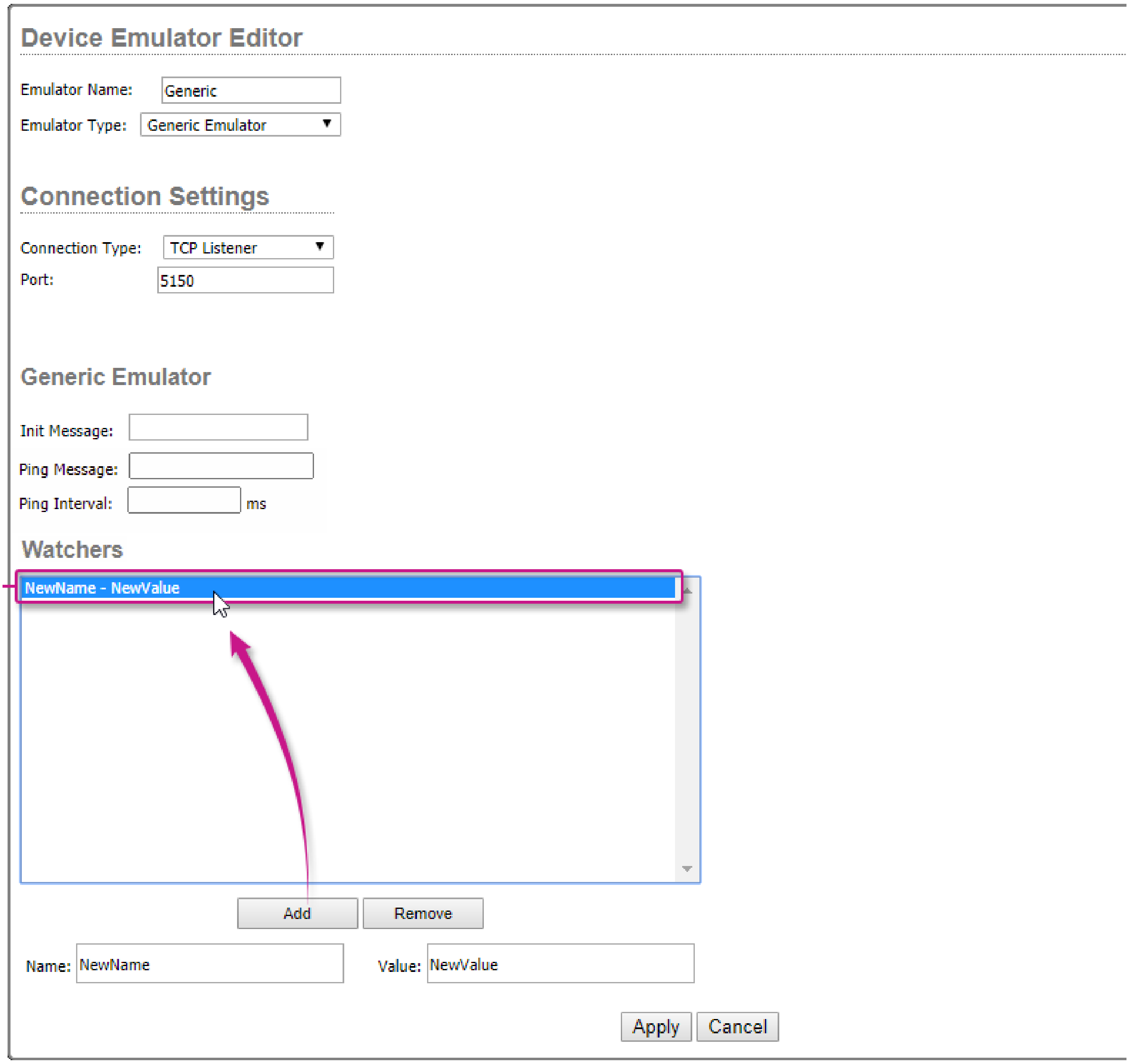
Init Message: Login Admin Admin\cr\lf

Hello - Hello

Name: Hello

Value: Hello

- Generic emulators also have a **Ping Message** field and **Ping Interval** field which are optional.
  - For situations where you are using a TCP client and you want to check to make sure the connection is still alive, sending a periodic ping can force the network to validate that. Leaving the **PingMessage** blank will mean this feature is not used.
  - When a remote device closes its TCP connection normally, there is a handshake that takes place that tells us the connection has been closed. The emulator will then try to start re-establishing the connection. However, if the remote device loses power or the connection dies in a way that does not allow the handshake to take place, then the DeviceEmulator will not know the connection is dead until either a failed operating system keepalive packet failure (which might take up to an hour or two) or until we attempt to send a message to the remote device.
  - The ping message and ping interval allow you to define a message that gets sent every so often to make sure the connection is still working. These messages should be something that makes sense to the application on the remote end.
  - It also allows us to test not only that the connection is still intact but that the application at the remote end is still responding. Note that it still may take longer than the interval time to detect a failure (often 15 to 60 seconds). Therefore, it does not make sense to set the interval too low.
- Generic Device Emulator can trigger actions based on incoming data; to define these actions, we create **Watchers.** Each Watcher looks for information coming to the emulator port and when it sees that data, it briefly sets the Watcher's *Triggered* property to *True.*



Device Emulator Editor

Emulator Name: Generic

Emulator Type: Generic Emulator

Connection Settings

Connection Type: TCP Listener

Port: 5150

Generic Emulator

Init Message:

Ping Message:

Ping Interval: ms

Watchers

NewName - NewValue

Name: NewName

Value: NewValue

With the new Watcher selected, type a **Name** for this Watcher and the **Value** being watched for in the *Name* and *Value* fields directly below the Watcher list.

You can edit an existing Watcher by clicking on the Watcher entry in the list and updating the values in these *Name* and *Value* fields.

- When all fields are complete, click Apply to save your changes and create the Emulator.

Escape Characters

Both the **ToSend** properties and the Watcher values support several special character strings to represent special characters:

Escape Sequence	Character
\cr	Carriage Return

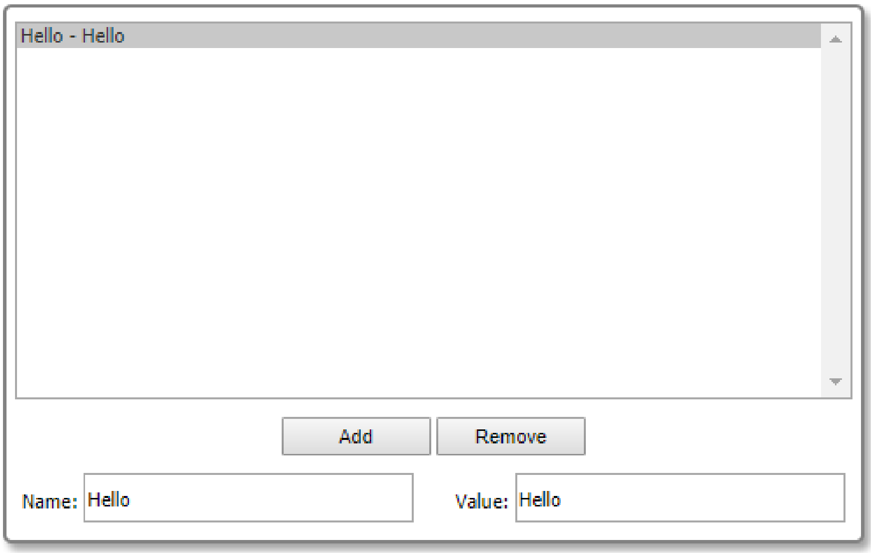
\lf	Line Feed
\t	Tab
\	used to un-escape an escape
\%XX	Hex value where XX is a two-character hex value for the character to send or receive.

Therefore, a Watcher that is watching for "MyName\cr\lf" is looking for the string “MyName” followed by a carriage return and line feed.

In the case of double slashes, such as "MyName\\cr\\lf", the double slashes are converted to a single literal slash for the case where you actually want to send the value “\cr\lf.”

Using Watchers in Logic Flows

The **Name** of the Watcher is used to identify the Watcher in the system and will appear as an object underneath the Emulator in the Logic Flows property selection tree.



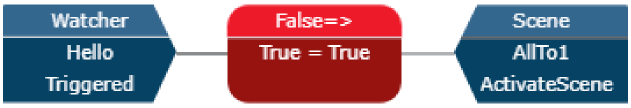
Hello - Hello

Name: Hello

Value: Hello

In the example above, we have created a single Watcher on this device emulator named **Hello**, and it is watching for the value **Hello**. When we see the word “Hello” come into this emulator, then the Watcher’s momentary *Triggered* property will be set to *True* and then reset to *False* until the next time the value “Hello” is registered.

We can create a Logic Flow where the Start Point is the *Triggered* property of the Watcher and do anything we like in the system when we see the string “Hello” come in through the Device Emulator port.



In this Logic Flow we have defined that every time we see the string “Hello” on the Watcher in question, we will activate the scene **AllTo1**.

Additional Emulator Properties for Logic Flows

Additional properties related to Generic Emulators that may be used in Flows include:

- **Connected** – This property will be *True* if there are any clients connected or if the TCP client is successfully connected and *False* otherwise
- **ConnectedCount** – This property will retain the number of currently connected clients to the emulator
- **ConnectionLost** – This property will switch to *True* and then back to *False* when a client disconnects
- **ConnectionObtained** – This property will switch to *True* and then back to *False* each time a client connects
- **TriggeredValue** - When the Watcher input is discovered, that value will be assigned to the *TriggeredValue* property momentarily and then the value will be set back to blank. It is momentary in order to raise changes when the same value enters the emulator repeatedly

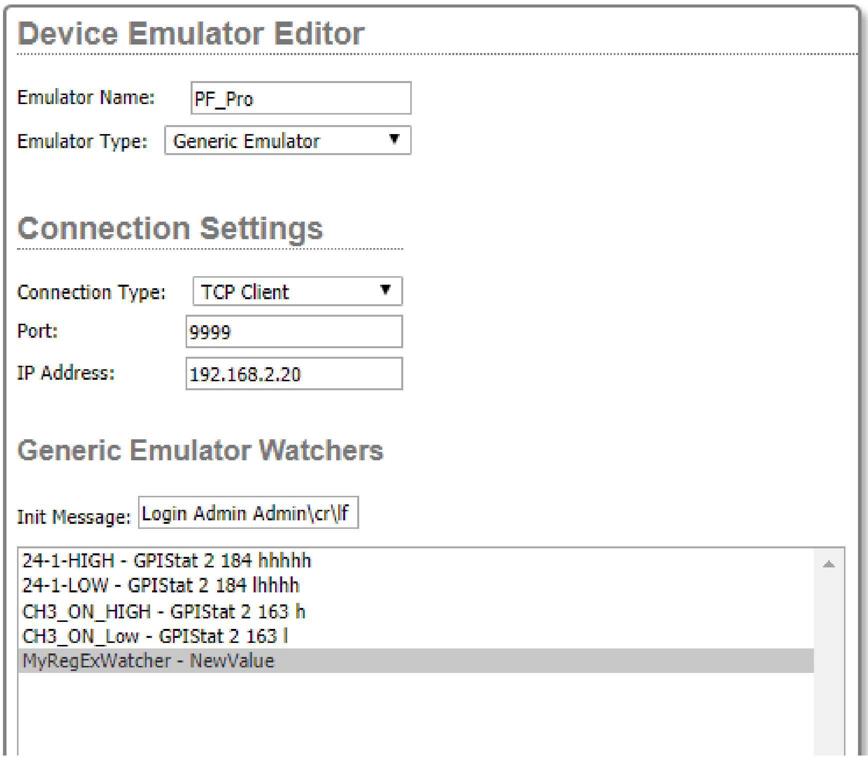
Watchers and Regular Expressions

Generic Device Emulator Watchers also support the use of regular expressions or **Regexes.** Pathfinder Code PRO uses the Microsoft .NET variation of regular expressions—an advanced language used for pattern matching in textual information.

Regular expressions parse the Watcher's input relative to the regular expression pattern, tripping the Watcher's *Triggered* property to *True* in case of a match. Pathfinder Code PRO Watchers make use of two flavors of regular expressions:

1. **Regex.Match(<expression_pattern>)**
2. **Regex.IsMatch(<expression_pattern>)**

To add a regular expression to a Watcher, type the **RegEx statement** in the Watcher *Value* field:



Device Emulator Editor

Emulator Name: PF_Pro

Emulator Type: Generic Emulator

Connection Settings

Connection Type: TCP Client

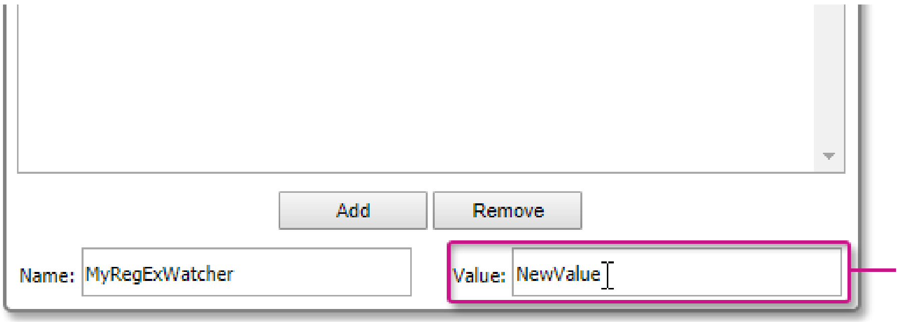
Port: 9999

IP Address: 192.168.2.20

Generic Emulator Watchers

Init Message: Login Admin Admin\cr\lf

```
24-1-HIGH - GPIStat 2 184 hhhhh
24-1-LOW - GPIStat 2 184 lhhhh
CH3_ON_HIGH - GPIStat 2 163 h
CH3_ON_Low - GPIStat 2 163 l
MyRegExWatcher - NewValue
```



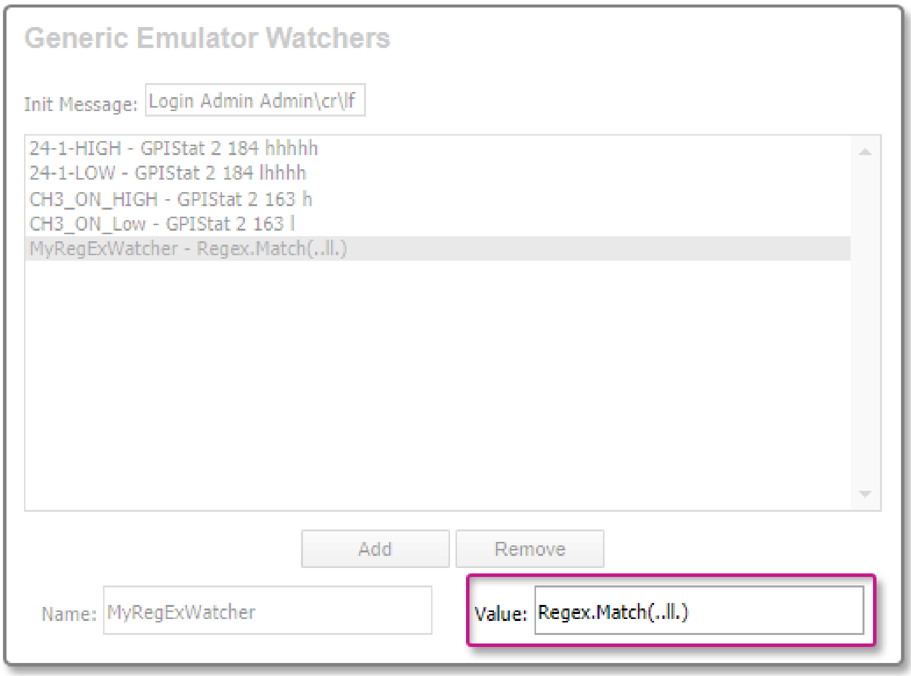
Name: MyRegExWatcher

Value: NewValue

Regex.Match Method

Regex.Match evaluates Watcher input against the expression, with valid matches resulting in the *Triggered* property being set to *True*. However, because a Regex can be used to match multiple different chunks of text, it is often also useful to have the exact chunk of text that was matched available to a Logic Flow. That is where the *TriggeredValue* property comes in.

Typing a Value like this **Regex.Match(..ll.)**will match any series of five characters where the third and fourth characters are **l**.



Generic Emulator Watchers

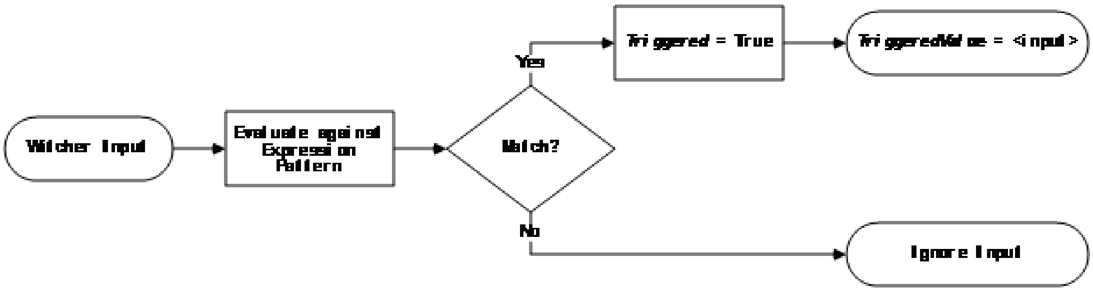
Init Message: Login Admin Admin\cr\lf

```
24-1-HIGH - GPIStat 2 184 hhhhh
24-1-LOW - GPIStat 2 184 lhhhh
CH3_ON_HIGH - GPIStat 2 163 h
CH3_ON_Low - GPIStat 2 163 l
MyRegExWatcher - Regex.Match(..ll.)
```

Name: MyRegExWatcher

Value: Regex.Match(..ll.)

Watcher inputs like **Yello**, **Hello**, **Mello**, **hhlll**, **hhllh**, will trip the Watcher’s *Triggered* property to *True* and allow the *TriggeredValue* property to pass the input value that resulted in the match.



As another example, to extract a duration from a satellite message such as "**CBS050",** we could use this regular expression in the Watcher’s *Value* field:

Regex.Match((?<=CBS).{3})

This Regex would set the Watcher's *Triggered* property to *True* any time it sees six characters where the first three are **CBS** and allow the *TriggeredValue* property to pass only three characters that fall after the **CBS**. In this example, the *TriggeredValue* property would equal **050**. This expression makes use of the look-behind assertion of regular expressions.

A different Watcher could then be created for a different show from the same Emulator.

Regex.Match((?<=NBC).{3})

Regex.IsMatch Method

Regex.IsMatch() performs a similar function except rather than passing the value of the match, the *TriggeredValue* will output *True* if a match exists. This makes this method largely redundant since a match will also set the *Triggered* property to *True*, but there are times when this method can be useful in Logic Flows as discussed below.

Special Characters

Carriage returns and line feeds must be considered when the incoming data is a line-based protocol. Failing to properly account for these special characters when parsing incoming data will cause incorrect results.

If your Watcher input will include carriage returns and line feeds at the end of each message, you could use something like **(.*\r\n)** or **((?s).*)**. The latter uses single-line mode.

Escape Sequence	Character
\r	Carriage Return
\n	Line Feed (new line)
\t	Tab
\e	Escape

Modifying Default Behavior

There are several options that can be used within the expression to modify the function of the pattern matcher. For example, by default, the dot or period character (**.**) matches any character except the new line. If you want it to match the new line character as well, you can enable the single line option to modify the function of the wildcard within the pattern matcher:

Regex((?s).*).

The inline **?s** modifies the regular expression engine to use the single line option when analyzing. These modifying options are documented here:

https://docs.microsoft.com/en-us/dotnet/standard/base-types/regular- expression-options

Regular expressions are extremely advanced and can appear to be very complicated. One joke about regular expressions is that once you solve a problem with a regular expression you now have two problems. However, they can also be awesomely powerful for the situations where they are needed.

They are part of one of the most commonly used and advanced text pattern matching and manipulation languages across all programming languages used today. However, they will also be slower and more CPU intensive than the normal matches Pathfinder Core PRO performs because they have not been optimized at compile time in the same way as the native code. This is only one reason why they should be used with caution and sparingly. Use them only when you need them and not when Pathfinder Core PRO's inherent pattern matching is a better option.

Also, test thoroughly. It is quite easy to design a regular expression that you think is correct but misses certain edge cases. The regex tester listed in the links below can be very useful in this case.

Again, this is an advanced feature and incorrect use can cause unpredictable results. But it is also a very powerful tool for certain situations.

Learn more about Regular expressions using these links:

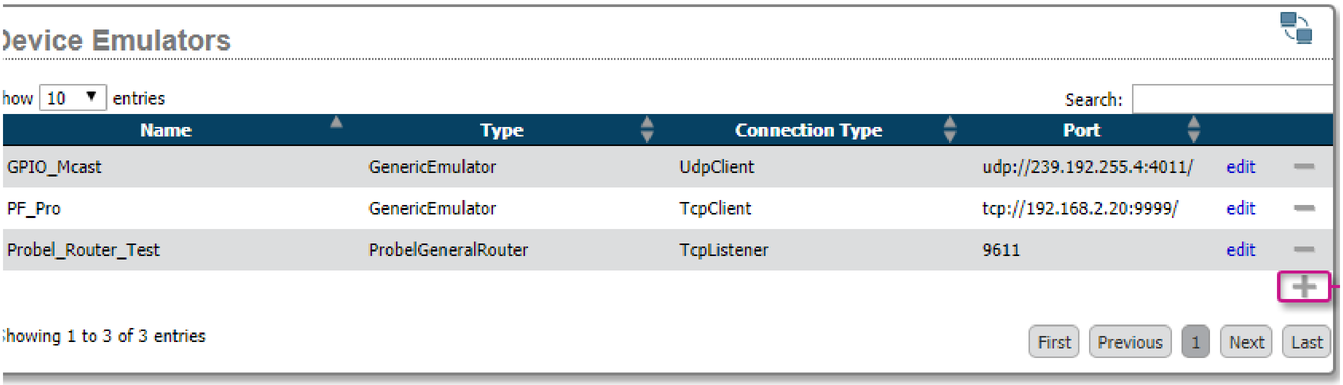
- https://www.regular-expressions.info/
- https://docs.microsoft.com/en-us/dotnet/standard/base-types/regular- expression-language-quick-reference
- http://regexstorm.net/tester

Probel General Router and General Switcher

There are two Probel routing protocols supported by Pathfinder Core PRO: *Probel General Router* and *Probel General Switcher*. The syntax of these protocols is different so it is important to consult the automation system's manual regarding which protocol to use.

In both cases, the protocol provides routing control. As a result, after selecting this protocol type, the web page will present a drop-down list with the available routers so that you can choose which router the Emulator will control.

- Click the **plus** icon in the Device Emulators page to create a new emulator.



evice Emulators

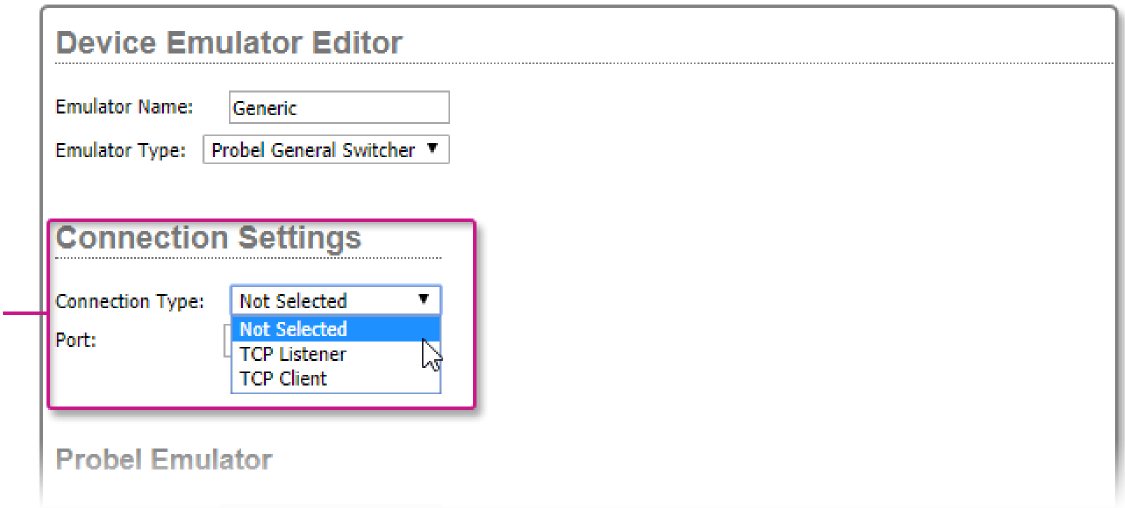
how 10 ▼ entries

Search:

Name	Type	Connection Type	Port		
GPIO_Mcast	GenericEmulator	UdpClient	udp://239.192.255.4:4011/	edit	−
PF_Pro	GenericEmulator	TcpClient	tcp://192.168.2.20:9999/	edit	−
Probel_Router_Test	ProbelGeneralRouter	TcpListener	9611	edit	−

howing 1 to 3 of 3 entries

- Type a unique **Name** for this emulator and select either **Probel General Switcher** or **Probel General Router** from the *Emulator Type* drop-down list.
- Complete the **Connection Settings**.



Device Emulator Editor

Emulator Name: Generic

Emulator Type: Probel General Switcher ▼

Connection Settings

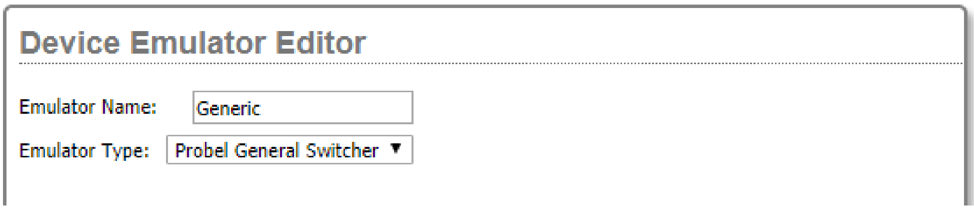
Connection Type: Not Selected ▼

Port:

Probel Emulator

Field	Description
Connection Type	From the drop-down list, select either **TCP Listener** or **TCP Client**, depending on whether: • The system using this emulator will establish the connection to Pathfinder Core PRO (an *inbound* connection—select **TCP Listener**) o • Pathfinder Core PRO should initiate the connection to the other device (an *outbound* connection—select **TCP Client**.)
Port	If the other device is initiating the connection to Pathfinder Core PRO (Pathfinder Core PRO is the **TCP Listener**) type the **port** on which Pathfinder Core PRO will listen for the connection. If Pathfinder Core PRO needs to initiate and maintain the connection (Pathfinder Core PRO is the **TCP Client**) type the **IP address** and **port** of the device to which Pathfinder Core PRO should connect.

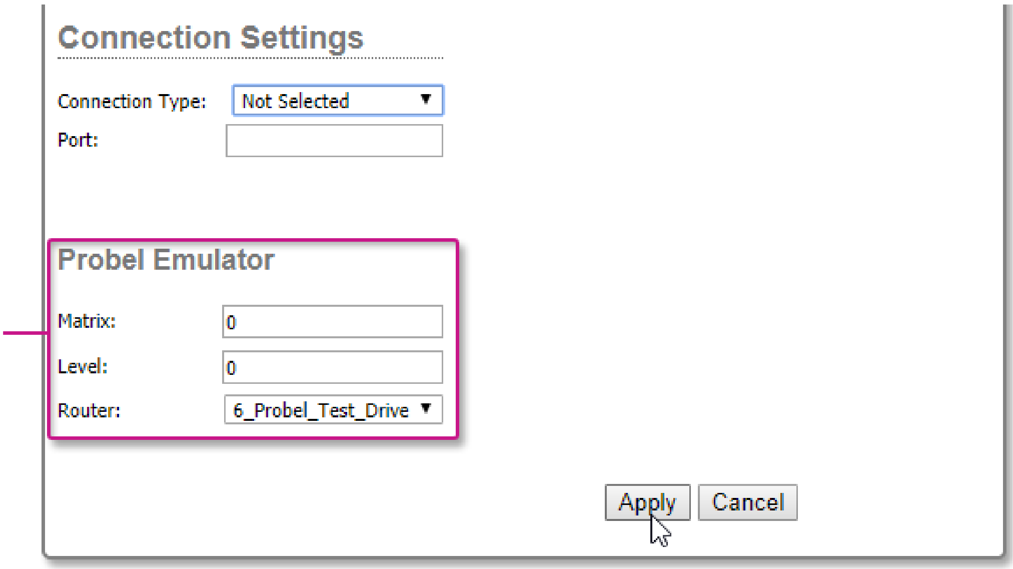
- Type Matrix and Level values, and select a Router from the drop-down list. When all fields are complete, click Apply to save your changes and create the Emulator.



Device Emulator Editor

Emulator Name: Generic

Emulator Type: Probel General Switcher ▼


Connection Settings

Connection Type: Not Selected

Port:

Probel Emulator

Matrix: 0

Level: 0

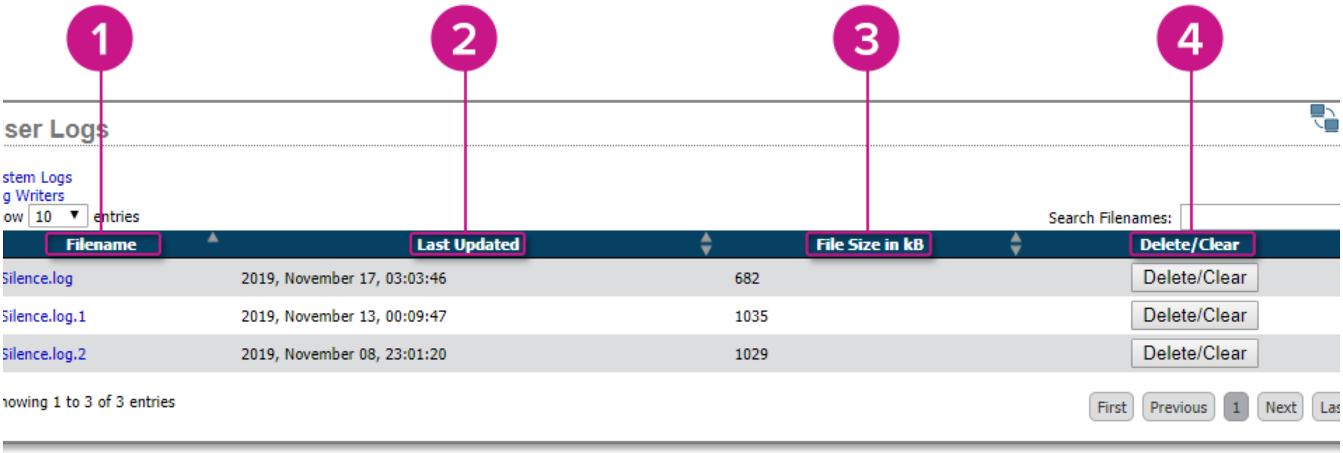
Router: 6_Probel_Test_Drive

Logs

Pathfinder Core PRO provides a variety of different logging mechanisms as well as maintaining its own internal system logs. There are two kinds of log files created by the system: *System Logs* and *User Logs*.

- **System Logs** are created by the system’s internal services and primarily define information for use by the developers.
- **User Logs** are where you can define changes on your Axia network that you want to be entered into log files for later review.

Click on the **Logs** link in the Navigation bar to view the current logs in the system as well as to define logging options.



ser Logs

stem Logs
g Writers

ow 10 entries

Search Filenames:

Filename	Last Updated	File Size in kB	Delete/Clear
Silence.log	2019, November 17, 03:03:46	682	Delete/Clear
Silence.log.1	2019, November 13, 00:09:47	1035	Delete/Clear
Silence.log.2	2019, November 08, 23:01:20	1029	Delete/Clear

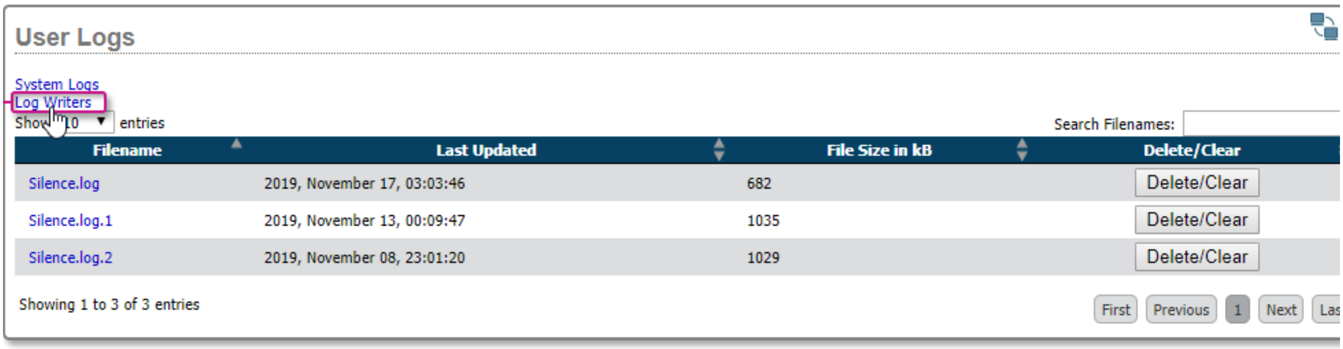
nowing 1 to 3 of 3 entries

Reference	Notes
1	The **Filenames** of all available logs. *See “Log Fil Maintenance” section later in this chapter*.
2	The date and time the log was **Last Updated**.

3	The **File Size in kB** of the log.
4	Options to either **Delete** or **Clear** the log.

Creating New Log Writers

To see the list of Log Writers that have been created in the system, click the **Log Writers** link on the User Log window.

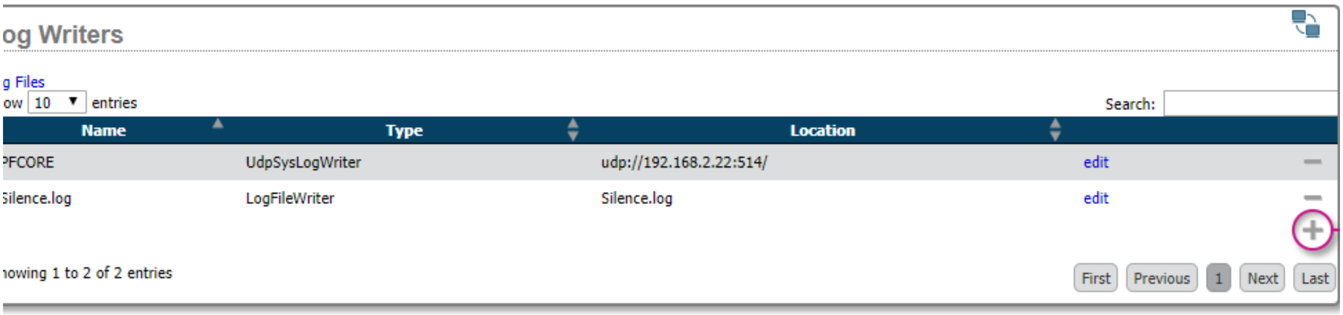


Filename	Last Updated	File Size in kB	Delete/Clear
Silence.log	2019, November 17, 03:03:46	682	Delete/Clear
Silence.log.1	2019, November 13, 00:09:47	1035	Delete/Clear
Silence.log.2	2019, November 08, 23:01:20	1029	Delete/Clear

Showing 1 to 3 of 3 entries

Each Log Writer in the system will be displayed on this page by *Name* along with its type and location.

- To create a new Log Writer, click on the **plus icon** on the *Log Writers* screen.



Name	Type	Location	
PFCORE	UdpSysLogWriter	udp://192.168.2.22:514/	edit
Silence.log	LogFileWriter	Silence.log	edit

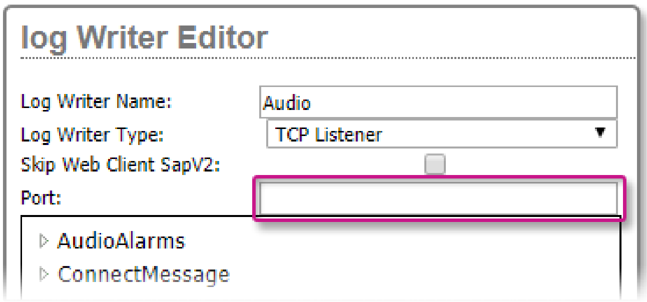
howing 1 to 2 of 2 entries

- Type a **Name** for this Log Writer in the *Log Writer Name* field.
- From the *Log Writer Type* drop-down list, select this Log Writer's **Type**. There are four types of Log Writer that can be created:

Log File Type	Description
Log File	Stored locally on the Pathfinder Core PRO systen these logs are subject to regular rotation, deletion and cleanup depending on space requirements. This Log File Type is useful for logging recent events but should not be relied upon for long term storage.
	This type of Log Writer will listen on a TCP port ar

TCP Listener	send the log entries as plain text messages to any application that connects. For example, you could use a putty (telnet) session to connect to the syste and monitor log changes dynamically.
TCP Client	This type of Log Writer will attempt to connect to a IP address and port, and if it can make the connection, will send the log messages as plain text messages to the listening application.
Syslog	This is the recommended method for long term log storage. Syslog is a highly utilized standard protocol for log collection, viewing, and storage used by many systems and IT scenarios. A numb of syslog collection applications exist, both freeware and commercial. A quick internet search should provide you with a variety of options.

- If you are configuring a *TCP Listener* or *Client* Log Writer, type the **TCP Port** associated with the TCP application.



log Writer Editor

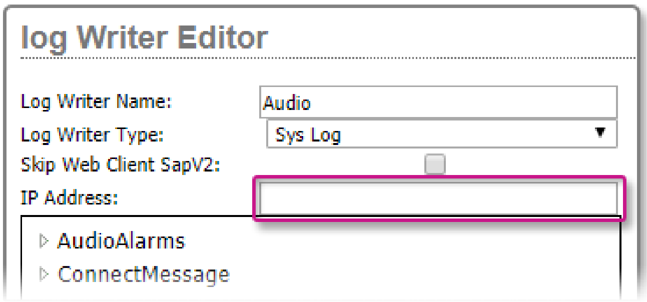
Log Writer Name: Audio

Log Writer Type: TCP Listener

Skip Web Client SapV2: □

Port:

- If you are configuring a *Sys Log* Log Writer, type the **IP Address** of the collector application



log Writer Editor

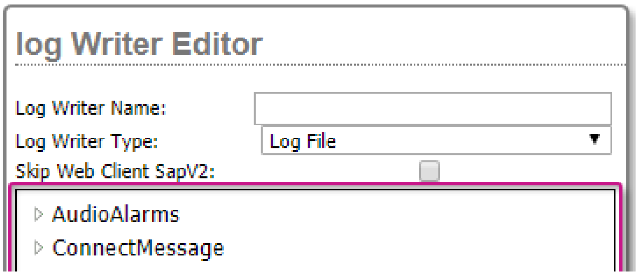
Log Writer Name: Audio

Log Writer Type: Sys Log

Skip Web Client SapV2: □

IP Address:

- Use the selection window at the bottom of this dialog to select what log options you wish to log.

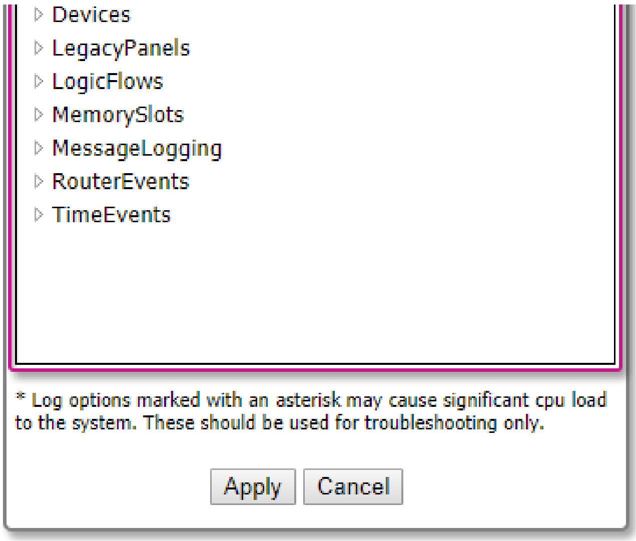


log Writer Editor

Log Writer Name:

Log Writer Type: Log File

Skip Web Client SapV2: □



* Log options marked with an asterisk may cause significant cpu load to the system. These should be used for troubleshooting only.

Expand the tree and check the items you wish to log. The available items are:

AudioAlarms

- AlarmState: Generates a log message whenever an audio alarm state changes.
- LvlState: Generates a log message whenever the audio threshold for silence is passed for longer than 250 ms. This occurs before the countdown to an alarm state begins. It can be used to log the actual transitions being reported by the equipment but may also generate a large amount of log content depending on the type of audio content being monitored.

ConnectMessage

- Connected: Logs connections and disconnections from equipment. Use this log message to find equipment with which Pathfinder Core PRO is struggling to maintain a connection.
- Online: Logs changes to the online/offline state of the device. For a device to be considered online, proper communications must be occurring on all identified ports.

Devices

- VMIXGain: Logs changes to VMIXer gain settings.
- VMIXState: Logs changes to VMIXer On/Off settings.

LegacyPanels

- PanelPropertyChanged: Logs changes to properties of any of the controls in user panels.

LogicFlows

- CombinerOutputChanged: Logs changes whenever a logic flow translator changes its output.
- TranslatorOutputChanged: Logs changes whenever a logic flow translator changes its output.

MemorySlots

-

MemorySlotChanged: Logs whenever a memory slot changes its value.

MessageLogging (*Note: use with support or when trying to find a problem as these can increase CPU load. These are for troubleshooting only*.)

- AccessViolations
- LoginFailures
- LoginSuccesses
- LwcpIncoming: Lwcp incoming messages.
- LwcpOutgoing: Lwcp outgoing messages.
- LwrpIncoming: Lwrp incoming messages.
- LwrpOutgoing: Lwrp outgoing messages.
- SapV2ExternalIncoming: SapV2 messages coming from outside the system.
- SapV2ExternalOutgoing: SapV2 messages being sent outside of the system.
- SapV2InternalIncoming: Incoming sapV2 messages between services.
- SapV2InternalOutgoing: Outgoing SapV2 messages between services.

RouterEvents

- GPIO State: Logs changes to GPIOs within the system.
- RouteState: Logs any route changes that take place within the system.

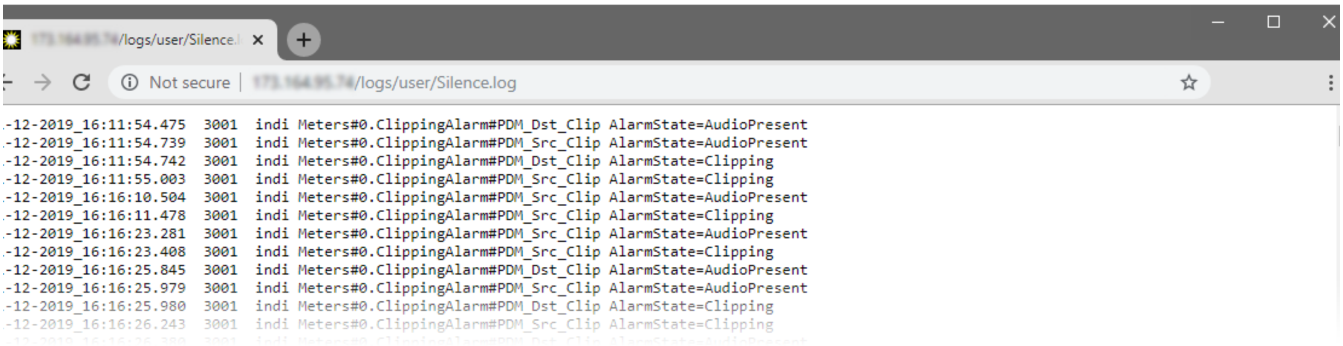
TimeEvents

- Elapsed: Logs whenever timers elapse.
- Enabled: Logs changes to a timer’s enabled state.

When all fields are complete, click **Apply** to save your changes and create the Log Writer.

Viewing Existing Logs

Clicking a log **Filename** in the *User Logs* window opens the log in a browser.



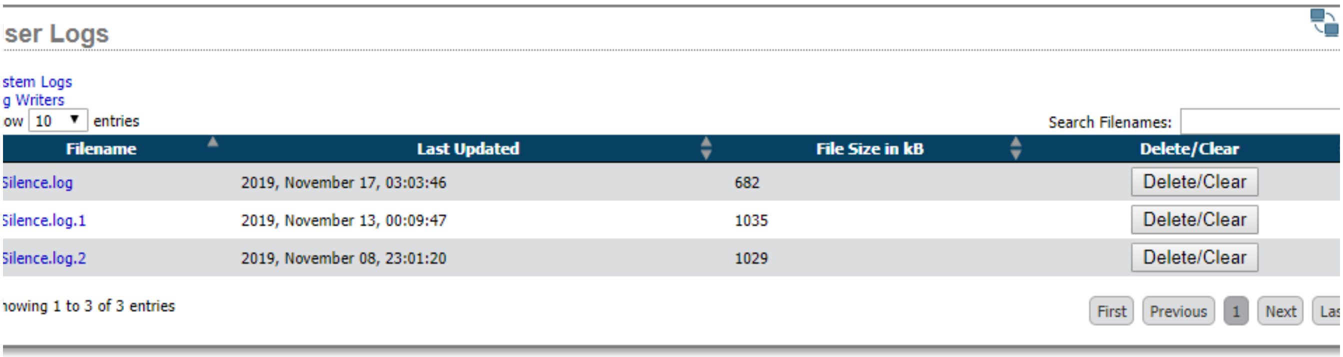
[redacted]/logs/user/Silence.l
Not secure
[redacted]/logs/user/Silence.log

```
.-12-2019_16:11:54.475  3001  indi Meters#0.ClippingAlarm#PDM_Dst_Clip AlarmState=AudioPresent
.-12-2019_16:11:54.739  3001  indi Meters#0.ClippingAlarm#PDM_Src_Clip AlarmState=AudioPresent
.-12-2019_16:11:54.742  3001  indi Meters#0.ClippingAlarm#PDM_Dst_Clip AlarmState=Clipping
.-12-2019_16:11:55.003  3001  indi Meters#0.ClippingAlarm#PDM_Src_Clip AlarmState=Clipping
.-12-2019_16:16:10.504  3001  indi Meters#0.ClippingAlarm#PDM_Src_Clip AlarmState=AudioPresent
.-12-2019_16:16:11.478  3001  indi Meters#0.ClippingAlarm#PDM_Src_Clip AlarmState=Clipping
.-12-2019_16:16:23.281  3001  indi Meters#0.ClippingAlarm#PDM_Src_Clip AlarmState=AudioPresent
.-12-2019_16:16:23.408  3001  indi Meters#0.ClippingAlarm#PDM_Src_Clip AlarmState=Clipping
.-12-2019_16:16:25.845  3001  indi Meters#0.ClippingAlarm#PDM_Dst_Clip AlarmState=AudioPresent
.-12-2019_16:16:25.979  3001  indi Meters#0.ClippingAlarm#PDM_Src_Clip AlarmState=AudioPresent
.-12-2019_16:16:25.980  3001  indi Meters#0.ClippingAlarm#PDM_Dst_Clip AlarmState=Clipping
.-12-2019_16:16:26.243  3001  indi Meters#0.ClippingAlarm#PDM_Src_Clip AlarmState=Clipping
.-12-2019_16:16:26.380  3001  indi Meters#0.ClippingAlarm#PDM_Dst_Clip AlarmState=AudioPresent
```

Log entries can be viewed in the browser window directly or copied to a text file for more robust searches.

Log File Maintenance

You may notice that some logs (such as the *Silence* logs in the screen shot below) have multiple copies with a period and a number at the end.



ser Logs

stem Logs
g Writers

ow 10 entries

Search Filenames:

Filename	Last Updated	File Size in kB	Delete/Clear
Silence.log	2019, November 17, 03:03:46	682	Delete/Clear
Silence.log.1	2019, November 13, 00:09:47	1035	Delete/Clear
Silence.log.2	2019, November 08, 23:01:20	1029	Delete/Clear

howing 1 to 3 of 3 entries

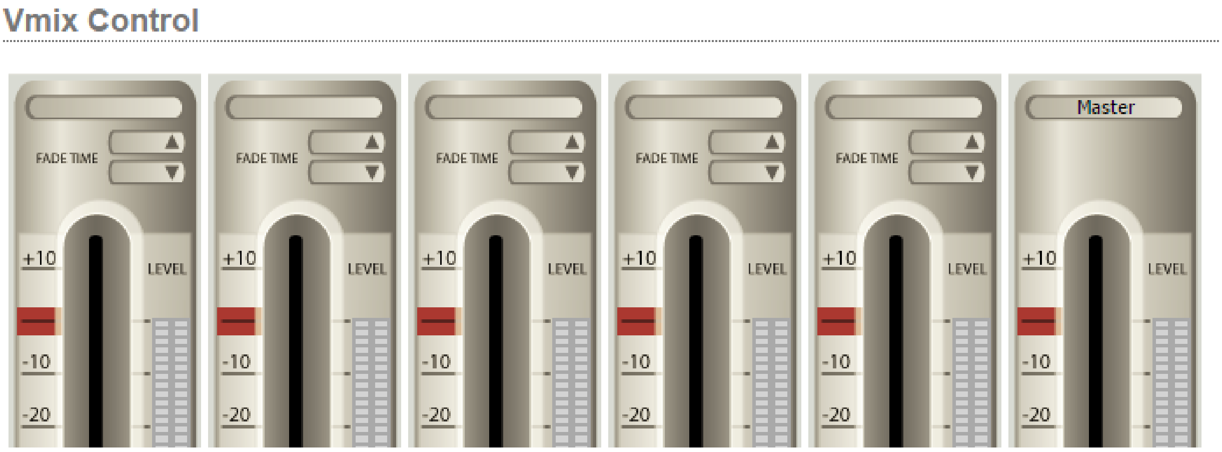
First Previous 1 Next Las

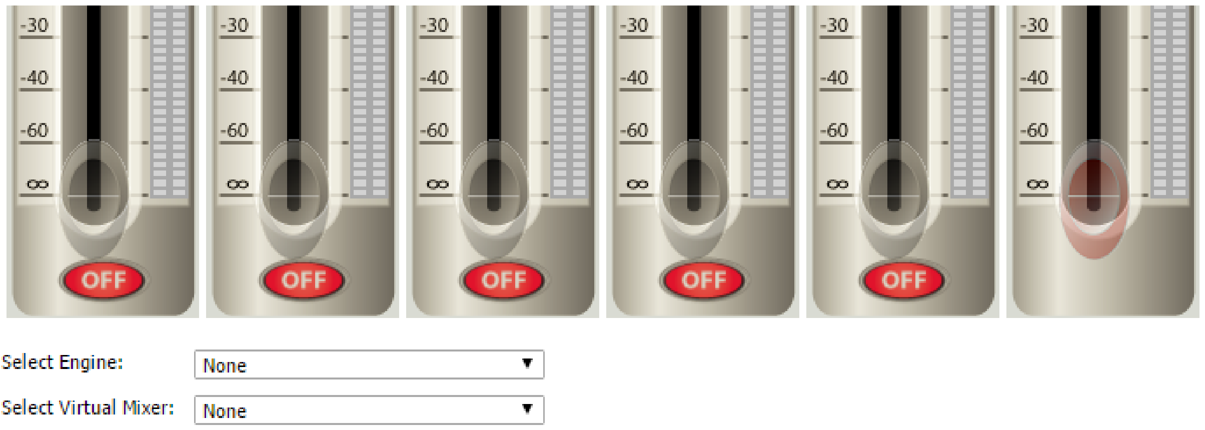
Due to space restrictions, Pathfinder Core PRO maintains an intelligent log rotation service which will rotate the log files as they become too large. Generally, log files are rotated four times with a new file created when the current log reaches approximately 4MB. When the fourth log gets too large, the oldest log is discarded and the filename reused.

There are some other rules which this service uses that might generate more aggressive log manipulation and/or deletion if space is becoming limited. For this reason, it is recommended that you use an outside syslog-based logging service to capture logs that you wish to store for longer periods of time. Review the *Creating New Log Writers* section for additional details.

Vmix Control

One of the powerful features of Axia Audio mixing engines are their virtual mixers. The Element and Fusion mix engines include sixteen virtual mixers which can be used to create mixes of various audio streams. These virtual mixers can be controlled using Logic Flows to dynamically create mixes of audio content based on changes in the system. Additionally, there is a web page for managing these virtual mixers, in effect creating on-screen mixers. To access this user interface, click the **Vmix Control** link on the navigation bar.

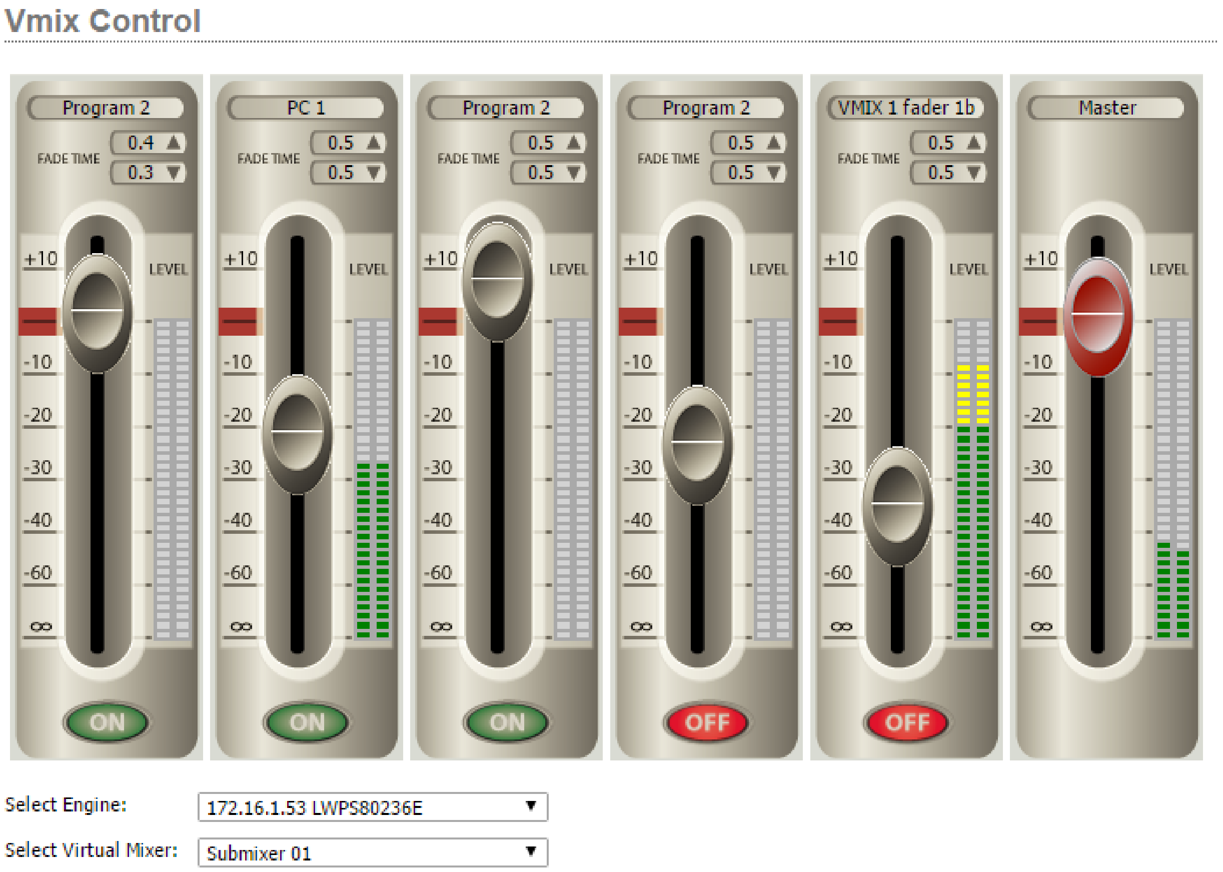




Select Engine: None

Select Virtual Mixer: None

When you first open this web page, the faders will be dimmed out and none of the buttons will function as you first need to define which engine and which Vmixer to control. Select the mix engine from the *Select Engine* drop down, and then select one of the 16 virtual mixers from the *Select Virtual Mixer* drop down.



Vmix Control

Select Engine: 172.16.1.53 LWPS80236E

Select Virtual Mixer: Submixer 01

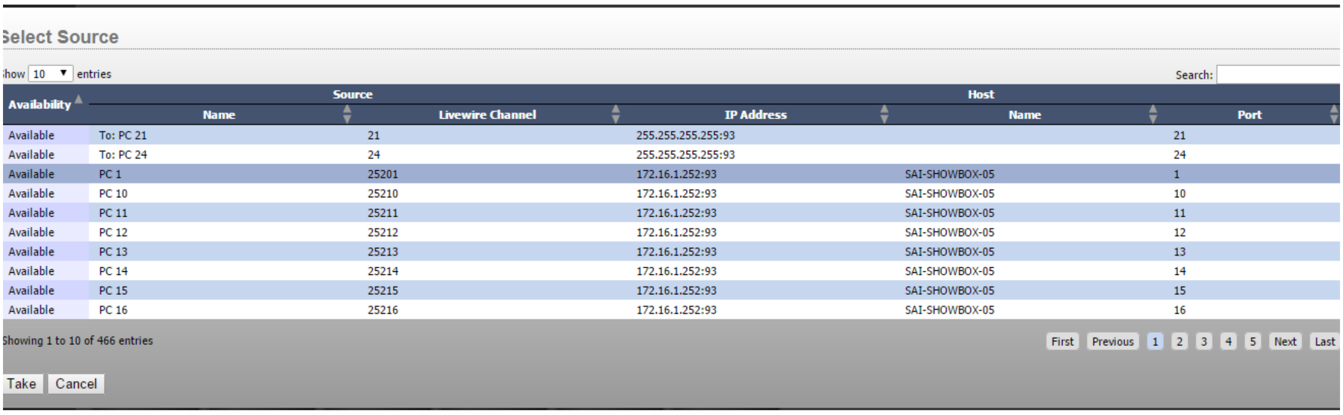
The interface will update to display the state of the selected virtual mixer. You can use the drop-down lists to switch between different Vmixers.

Each Vmixer has five input channels and a master fader. The master fader is labeled as *Master* and has the red slider. The input channel meters display the real-time metering of the source assigned to each fader. This metering is pre-fader, showing the level of incoming audio. The meter next to the Master fader displays the level at the output of the virtual mixer and is therefore post-Master fader.

Dragging a fader’s on-screen slider up or down will adjust the VMixer channel’s gain.

Clicking the On/Off buttons will turn a fader on and off. Turning a virtual mixer fader on or off is subject to the *fade up* and *fade down time* parameters. These can be changed by clicking on the fields next to the up

arrow or down arrow in the Fade Time section. Times are entered in seconds. So if you have a fade up time of 1.2 seconds, when you click the On button the audio will fade in to the level defined by the fader over the course of 1.2 seconds. Then the fader control will dynamically allow you to control the level. The same is true when turning the fader off. Clicking the off button will fade the audio out dependent on the fade down time.
Finally, clicking on the source name at the top of each fader will allow you to select a different audio source for the fader. A list of available sources will appear.



Availability	Source		Host		
	Name	Livewire Channel	IP Address	Name	Port
Available	To: PC 21	21	255.255.255.255:93		21
Available	To: PC 24	24	255.255.255.255:93		24
Available	PC 1	25201	172.16.1.252:93	SAI-SHOWBOX-05	1
Available	PC 10	25210	172.16.1.252:93	SAI-SHOWBOX-05	10
Available	PC 11	25211	172.16.1.252:93	SAI-SHOWBOX-05	11
Available	PC 12	25212	172.16.1.252:93	SAI-SHOWBOX-05	12
Available	PC 13	25213	172.16.1.252:93	SAI-SHOWBOX-05	13
Available	PC 14	25214	172.16.1.252:93	SAI-SHOWBOX-05	14
Available	PC 15	25215	172.16.1.252:93	SAI-SHOWBOX-05	15
Available	PC 16	25216	172.16.1.252:93	SAI-SHOWBOX-05	16

Showing 1 to 10 of 466 entries

Highlight the new source and click take to select the new source for the fader. Or, click cancel to return to the mixer web page.

Vmixer Shortcuts

It is also possible to generate direct links to a particular Vmixer. This can be useful if you wish to create a shortcut on a user’s desktop that will present a specific Vmixer. In order to create such a shortcut in Windows, right click on the desktop and select new shortcut. For the location type:

http://172.16.1.221/vmixcontrol.php?engine=172.16.1.63&vmixer=1

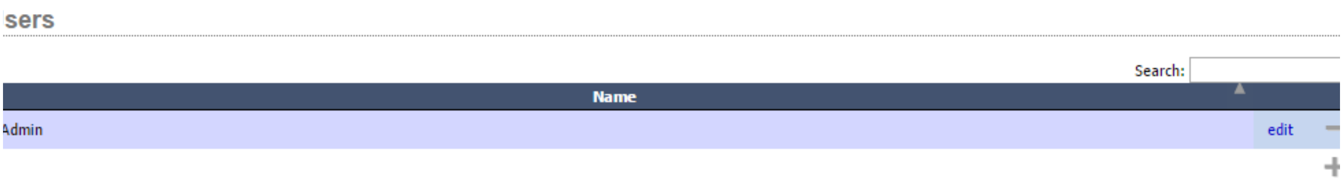
Replace the 172.16.1.221 with the IP address of your Pathfinder Core PRO. Replace the 172.16.1.63 address with the address of the mix engine you wish to control. And replace the 1 at the end of the link with the virtual mixer number within the engine you wish to control. Then click next and provide a name to create the shortcut. And finally, click finish. Or, just enter the address into the address bar of the user’s browser and then add it to favorites.

When you create the link this way, the user will not see Pathfinder Core PRO’s navigation bar or the drop downs to select a different engine or Vmixer. The web page will only present the specified virtual mixer.

Users

When you first log in to Pathfinder Core PRO, the username and password you use is:
Username: Admin
Password: Admin

However, this can be changed and controlled using the Users link on the navigation bar.

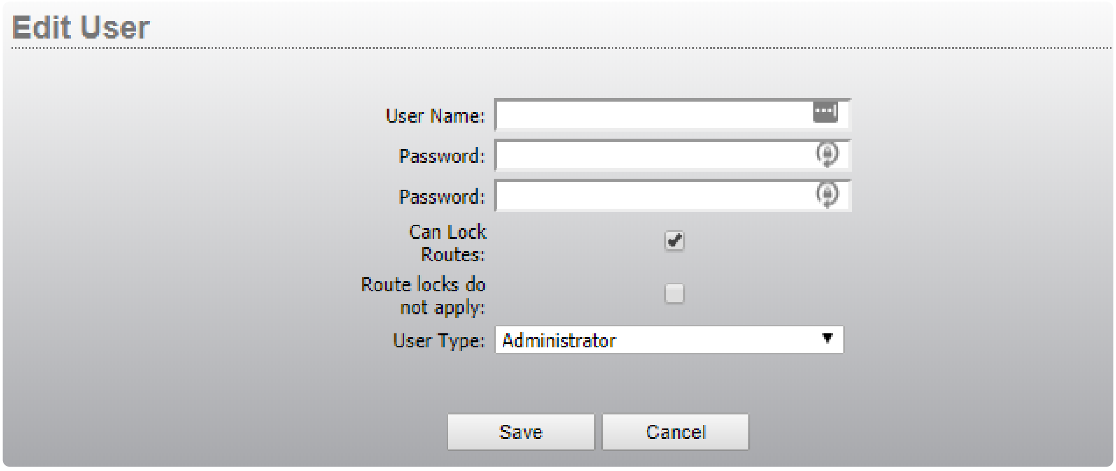


sers

Search:

Name	
Admin	edit

Click edit to change a user’s password or click the plus icon to add users to the system. Use the minus icon to delete users from the system. The system will not allow you to delete all users. At least one user must always exist in the system or there is no way to use it. Therefore, the user interface will show an error message if there is only one user left and you try to delete it.



Edit User

User Name:

Password:

Password:

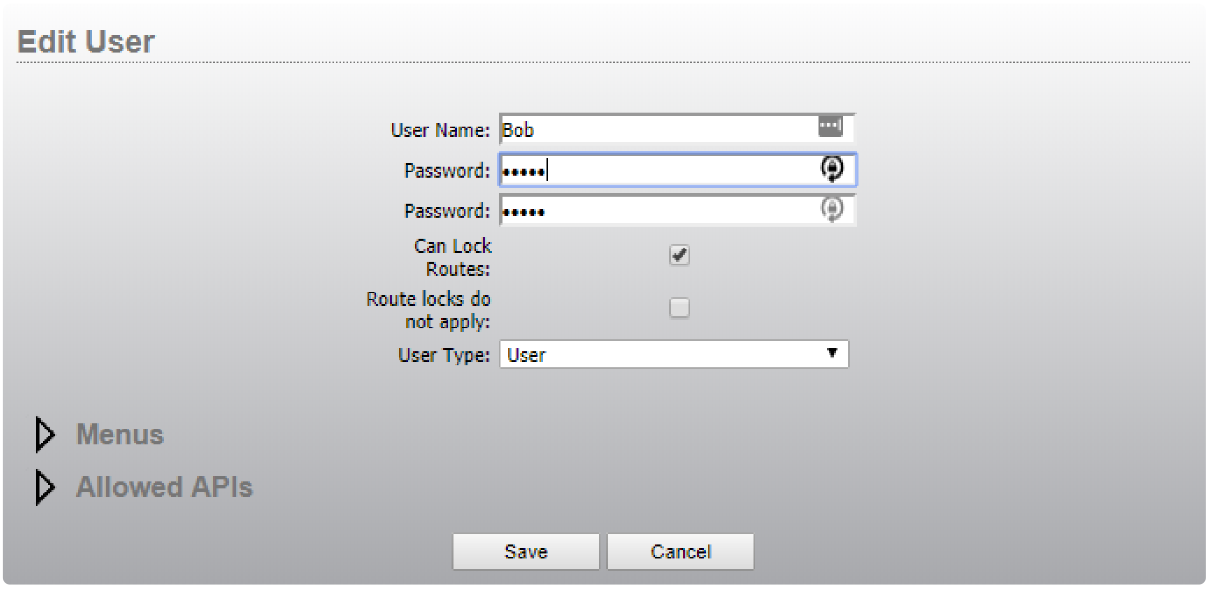
Can Lock Routes: ☑

Route locks do not apply: □

User Type: Administrator

After creating a new user, that user’s credentials may be used to log into the Pathfinder Core PRO’s web pages or using PathfinderClient, PathfinderMini, or PanelDesigner.

Users may be Administrative users who have access to everything or standard users. It is very important never to delete all of the Administrative users or you may lose access to the system requiring support to get access to the operating system to reset things. If you select a standard user, additional options will appear for defining what rights the user has. The *CanLockRoutes* defines whether the user has the ability to lock and unlock routes. The *Route locks do not apply* option is for future use.



Edit User

User Name: Bob

Password: •••••

Password: •••••

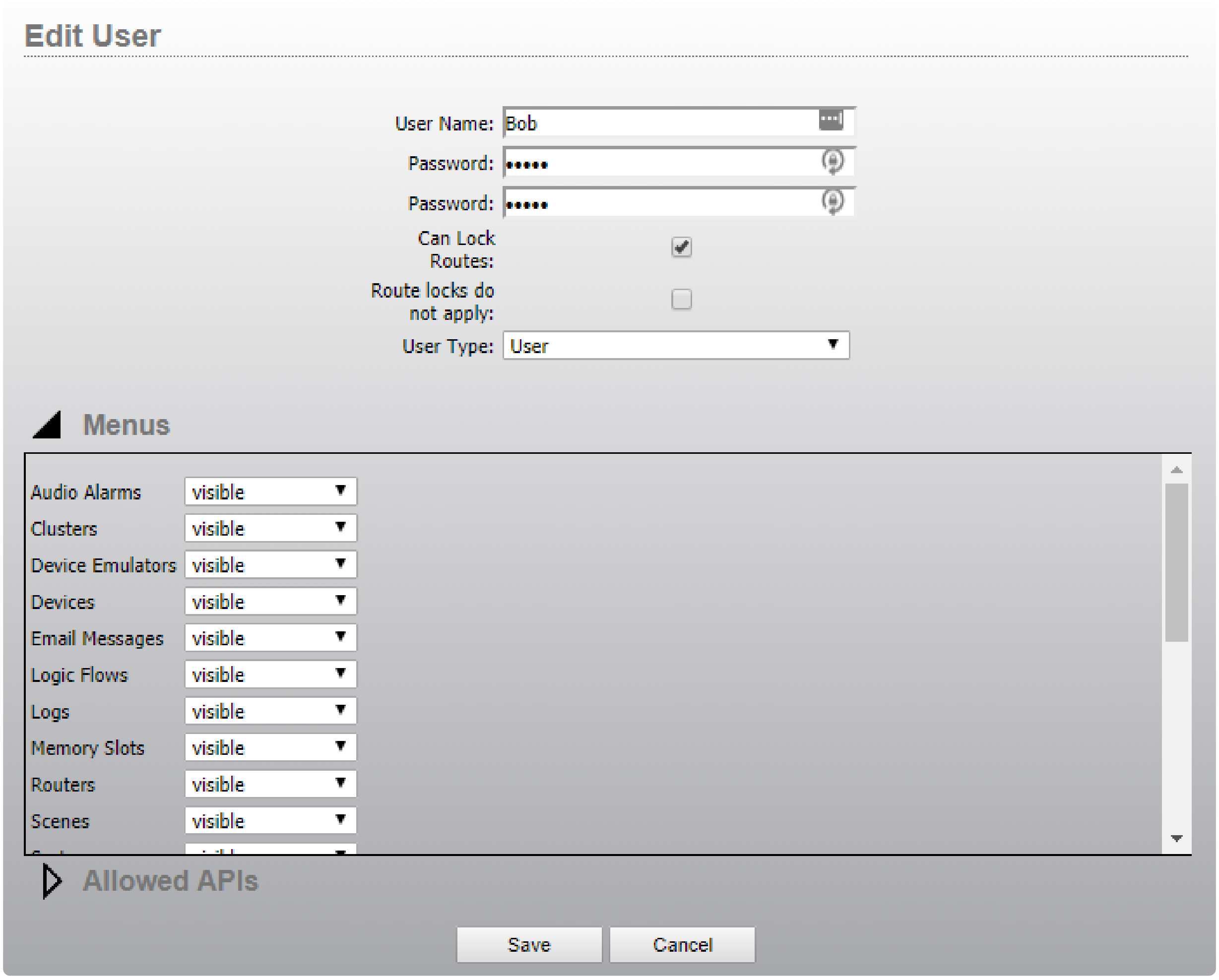
Can Lock Routes: ☑

Route locks do not apply: □

User Type: User

The arrows next to each section may be expanded to specify the user rights. By default a standard user will see all navigation menu items but will have no rights in the API so most or all of the menu items will not populate with any data.

The menus section allows you to define which navigation bar menu items are available to the user.



Edit User

User Name: Bob

Password: •••••

Password: •••••

Can Lock Routes: ☑

Route locks do not apply: □

User Type: User

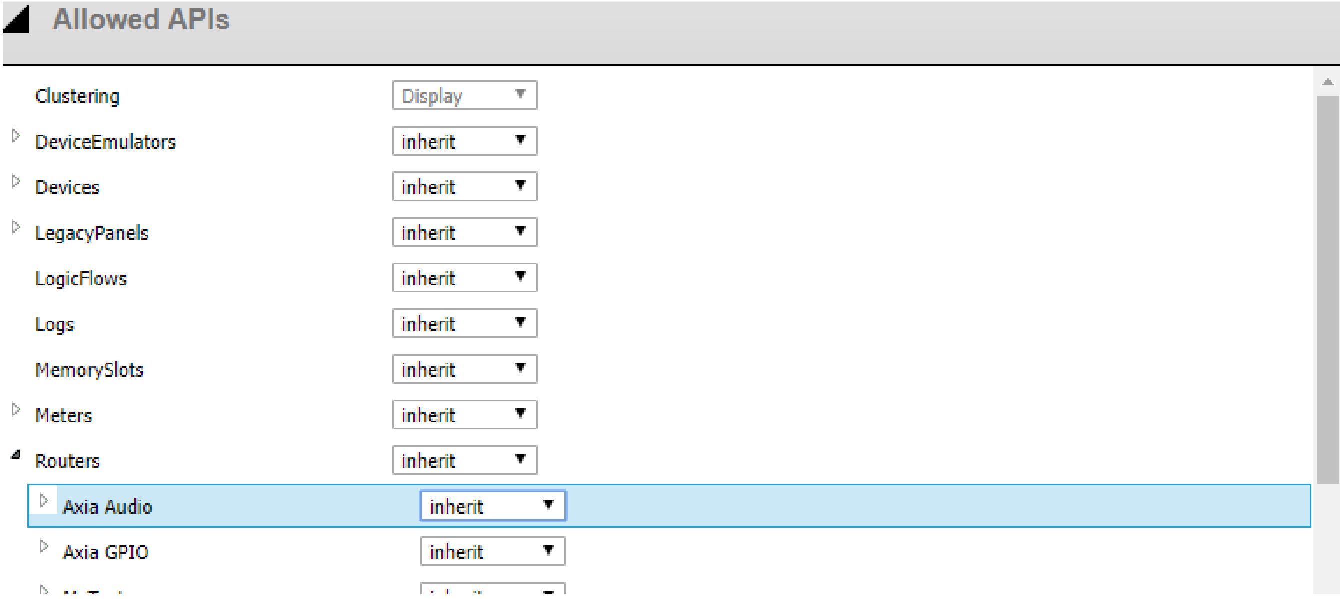
Menus

Audio Alarms	visible
Clusters	visible
Device Emulators	visible
Devices	visible
Email Messages	visible
Logic Flows	visible
Logs	visible
Memory Slots	visible
Routers	visible
Scenes	visible

Allowed APIs

The menus listed are the same as in the navigation bar on the web page. By setting items to hidden they will no longer appear in the user's navigation bar.

The allowed APIs section defines in much more detail what the user may access via the web pages, port 9600 login, and the client applications:



Allowed APIs

Clustering	Display
DeviceEmulators	inherit
Devices	inherit
LegacyPanels	inherit
LogicFlows	inherit
Logs	inherit
MemorySlots	inherit
Meters	inherit
Routers	inherit
Axia Audio	inherit
Axia GPIO	inherit

MyTest	inherit
MyVR	inherit

Each item in the displayed list has several options:

- *Inherit* means that branch in the tree inherits the rights of the parent branch. In the case of root level objects such as Logic Flows the inherited state is *No Access*. *No Access* explicitly denies access to that branch.
- *Display* allows the user to display the resource but not change or use it. For example, if you wanted the user to be able to view the route states of a router but not make any route changes, then selecting display for the router access would be a good idea.
- *Change/Use* allows the user to do normal operations with the resource but not change the definition of the resource. For example, in the case of routers, selecting change/use would allow the user to make route changes on the router but would not allow them to add sources or destinations to the router.
- *Full access* allows the user to have complete control over the resource including the ability to add to or delete the resource. This is the equivalent of Administrative rights to the resource.

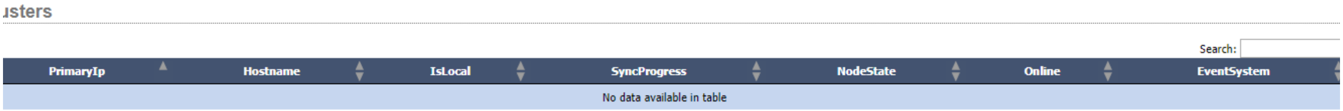
It is important to note that by default, branches below a specific branch inherit the parent's properties. So, if you set the Routers branch to display, all routers will be displayed. You can then set no access on routers you do not want the user to see and/or Change/Use access to routers you want them to be able to use.

Clusters

Pathfinder Core PRO clustering is designed to address concerns over hardware failures. Clustering allows you to connect multiple networked Pathfinder Core PRO systems, so that they automatically and dynamically share configuration data and states. They also monitor each other, taking over processing events if one system fails.

As you prepare to configure your cluster:

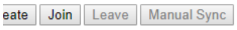
1. Begin with a single Pathfinder Core PRO
2. Create the routers and add other required events and components
3. Take a backup of your configuration and download it to your local computer. *See the section on Backup/Restore for more information.* This will guarantee you can get back to your current operational state if you make a mistake when setting up your cluster.
4. Click the **Clusters** link in the navigation bar



usters

Search:

PrimaryIp	Hostname	IsLocal	SyncProgress	NodeState	Online	EventSystem
No data available in table						

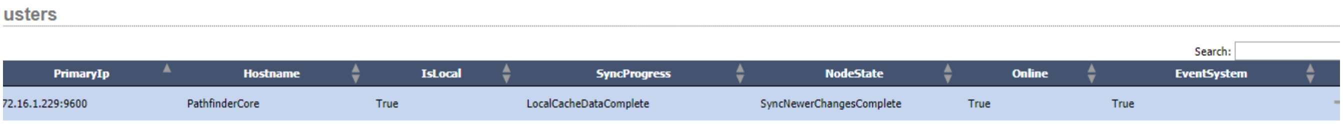


Four buttons are available at the bottom of the Clusters page; **Create**, **Join**, **Leave**, and **Manual Sync**.

> ⓘ Note: **Leave** and **Manual Sync** are disabled until the system is part of a cluster.

Creating a Cluster

First, click Create to create the cluster. The system will ask you to create a cluster administration password. This password is important because it will be used throughout the cluster so that the Pathfinder Core PRO units can log in to each other and synchronize data. When you provide this password, the system will create a special ClusterAdmin user for this purpose. You will be able to see this user when you view the Users navigation link and a severe warning will be issued if you try to change the password of that user since it must be identical across all systems. Once the cluster has been created, the Clusters link should look something like this:


usters

Search:

PrimaryIp	Hostname	IsLocal	SyncProgress	NodeState	Online	EventSystem
72.16.1.229:9600	PathfinderCore	True	LocalCacheDataComplete	SyncNewerChangesComplete	True	True

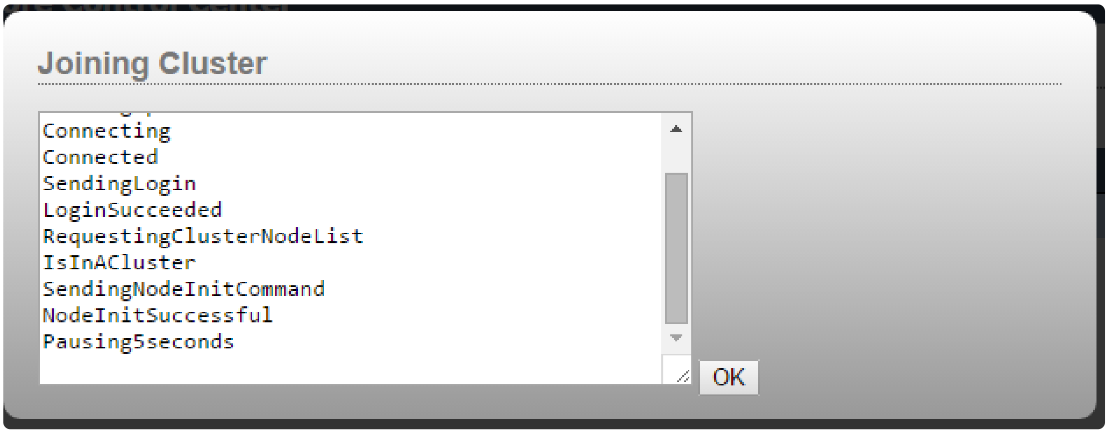
The system is now part of a one node cluster. This display will show the hostname and IP address for each node in the cluster. When there is more than one systems in the cluster, the IsLocal field will show which of the systems you are currently logged into. It will also display the state of each Pathfinder Core PRO system in the cluster. The minus button should only be used in the rare situation where you need to forcefully remove a non-functioning node from the cluster. The preferred method is to go to the node to be removed and click the Leave button.

Joining a Cluster

Now it's time to add the second system into the cluster. It is important to note that the second system’s configuration will be overwritten during this process. If the system is new, make sure you first assign an IP address using the front panel. The IP must have network access to the first unit in the cluster, so pay attention to the network configuration. You do not need to enable Livewire discovery on this second system.

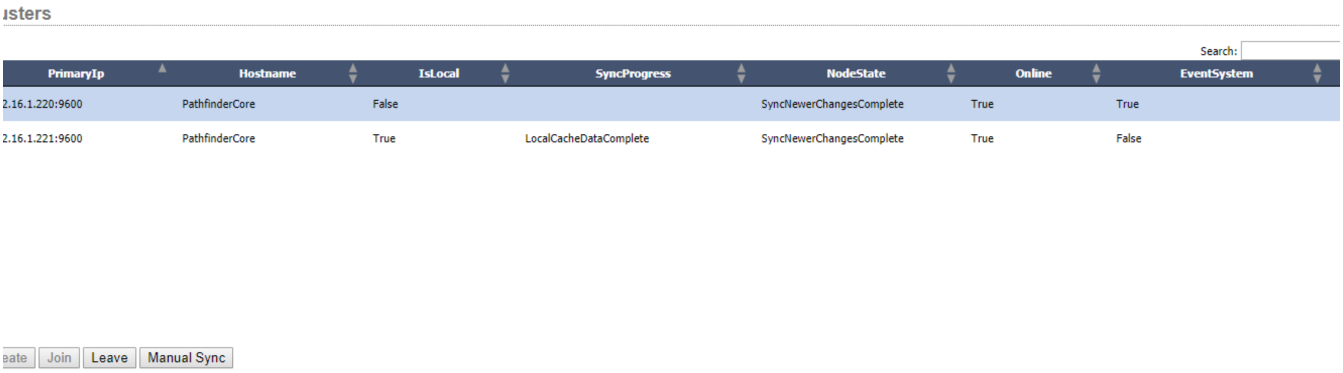
On the second system, browse to the clusters link on the navigation bar, and then click the Join button. The system will ask for the IP address of any Pathfinder Core PRO system that is already a part of the cluster. It

The system will now pop up a progress message box as the system joins the cluster.



```
Connecting
Connected
SendingLogin
LoginSucceeded
RequestingClusterNodeList
IsInACluster
SendingNodeInitCommand
NodeInitSuccessful
Pausing5seconds
```

The join process involves telling the first system that the new node is joining the cluster. It will then ask the first system to generate a special backup. The second system will download that backup and restore and then reboot. At that point, the cluster should be created, and you should be able to see both systems on the cluster page.



PrimaryIp	Hostname	IsLocal	SyncProgress	NodeState	Online	EventSystem
2.16.1.220:9600	PathfinderCore	False		SyncNewerChangesComplete	True	True
2.16.1.221:9600	PathfinderCore	True	LocalCacheDataComplete	SyncNewerChangesComplete	True	False

A successfully synchronized cluster should show all node states on the web page of all systems as SyncNewerChangesComplete. If that is not the case in your cluster and it does not resolve to that state within a few minutes after a restart, then contact Axia support to help determine why the cluster is not synchronizing properly.

Follow this procedure with subsequent Pathfinder Core PRO systems you wish to have join the cluster.

Once the cluster is created, changes to the configuration and state information are automatically synchronized across the Pathfinder Core PRO systems in the cluster. Additionally, in the case of a failure of one of the Pathfinder Core PRO systems, one of the other systems will begin processing the events. See the section on Events and Timers for more information.

Manual Sync

Sometimes it may be desirable to force the cluster into synchronicity if there is some doubt that it is properly synchronized. In order to do this, browse to the system that is incorrect (out of sync) select the cluster menu item and select the manual sync button. This will ask for confirmation and then will request a backup from the other (known good) system, download and restore that backup and then reboot. This is usually a

Leaving a Cluster

If you wish to leave a cluster, make sure that the system is connected to other nodes in the cluster. Then click on the Clusters link and click the Leave button. The system will revert back to a stand-alone system. It is important to note that after leaving a cluster, the configuration will still be the same as it was prior to leaving the system. It is recommended to quickly remove the system from the network and/or factory default the system. Otherwise, the system may start trying to process events that the rest of the cluster is also trying to process.

Events and Timers

There are a number of elements of a Pathfinder Core PRO system that should only execute on a single system at a time so that a cluster is not requesting duplicate changes from a device. Examples of this are Logic Flows and Timers. Those will always execute on the system that is currently active and has the lowest IP address. Heartbeats are constantly sent and monitored between the systems so that if a system stops responding the next lowest IP address system will take over those responsibilities.

Clustering and SapV2

SapV2 is the protocol that is used both for cluster communications and for internal communications. This protocol is available on port 9600 and is described in more detail in the appendix of this manual. If the cluster does not seem to be working properly there is a great deal of information that can be collected about the clustering process using this protocol to try and solve the problem.

GPIO Node Clustering

The Pathfinder Core PRO GPIO Node in a cluster is synchronized as well. Port additions, removals, and configurations applied to either Pathfinder Core PRO of the cluster will be replicated to the other Pathfinder Core PRO in the cluster.

Pin closure states will also be replicated in most cases. If the port has been configured with a multicast GPIO channel or snake routing assignment (IP/port), replication becomes a bit more nuanced. Closures that are directly tripped will still be replicated. However, closures that are sensed from the mcast or unicast snake (GPIs from the other snake port) will not be replicated. This is because it is assumed that both cluster nodes are monitoring those changes directly from the third piece of equipment and will pick up their states directly from the other end of the snake. Therefore, cluster replication of that state would in fact be redundant and could lead to race conditions.

When dealing with unicast snake mode (IP/port assignments), there are some additional clustering nuances to be aware of. These are most easily discussed via an example.

When you route a GPIO node to the Pathfinder Core PRO GPIO port, the address field in the GPIO port on both cluster nodes will get filled by IP Address/port. For example:

- Pathfinder Core PRO A = 172.16.1.241
- Pathfinder Core PRO B = 172.16.1.242
- GPIO XNode = 172.16.1.85

If you route (using the Pathfinder Core PRO GPIO Router) the XNode GPIO source 1 to the Pathfinder Core PRO A or B destination 1, the destination port on both Pathfinder Core PROs will look like:

- CFG GPO 1 SRCA:”172.16.1.85/1”

In this example both Pathfinder Core PROs are monitoring the GPI changes on port 1 of XNode at 172.16.1.85 and making matching changes on their GPO pins. This is straightforward.

However, if we route either of the Pathfinder Core PRO servers to the XNode GPIO things get a bit more interesting. We can only apply one IP Address to the SRCA field on the XNode destination. Therefore, Pathfinder Core PRO will use the Axia livewire address of whichever Pathfinder Core PRO in the cluster currently has its event system active. In the example above Pathfinder Core PRO A would typically be the active server in the cluster. When you make the route change on either Pathfinder Core PRO, the system will use the active server IP address on the XNode destination. The XNode destination would look like:

- CFG GPO 1 SRCA:”172.16.1.241/1”

If Pathfinder Core PRO A Server were to fail or get shut down, the B Server will loop through its routes and find any destination that Pathfinder Core PRO A’s GPIO ports were routed to and switch them to Pathfinder Core PRO B’s external livewire ip address.

- A Fails
- XNode port gets reassigned:
    - CFG GPO 1 SRCA:”172.16.1.242/1”

If A comes back online the port will switch back to use the A server when the event system comes online again.

One important note about this is that the XNode GPIO clears its pin states to be all high each time a route change happens and then reassigns them to the new values based on the new GPIs. This means that if a snake route is held low on the XNode based on a low GPI on Pathfinder Core PRO, during a cluster failover those pins might flicker to high and then back to low. This is usually a virtually instantaneous flip.

This should only cause an issue if you have a Pathfinder Core PRO GPI that is unicast snake routed to an XNode (or other Axia GPIO device) GPO and that closure is both normally held low and you have an activity (such as an automation advance) that happens on the flip from high to low and you fail over to the secondary node in the cluster. But this is also similar to what would happen if a device supplying that same closure were to be restarted. Note that this is not an issue for multicast snake routing.

In the future we may address this issue by using floating IPs where the ip route would not change but rather the ip address used for the external routes would float/move between the Pathfinder Core PRO devices.

We highly recommend that users do some testing of this functionality to fully understand how it works. For the vast majority of use cases it should be seamless.

Appendix A SapV2

SapV2 stands for Software Authority Protocol Version 2. It is the glue that provides all the communication between the services insider Pathfinder Core PRO. It is also available for control and troubleshooting purposes on port 9600 for those who wish to take the seriously deep tech dive into Pathfinder Core PRO.

> Important Note: Since SapV2 makes up the guts of inter-process communication within Pathfinder Core PRO, it is both incredibly powerful but also incredibly dangerous in the wrong hands. Backups are highly recommended before working with this protocol as it can be used not only to monitor objects but to create and destroy them. For instance, all of the web page user interfaces manipulate the Pathfinder Core PRO configuration using this API; therefore, you can do anything the configuration web pages can do using this protocol. BE CAREFUL!

In order to interact with SapV2, open a telnet session to port 9600 on the Pathfinder Core PRO.

In order to work with the system in any way using the protocol, you first must log in:

```
Login <username> <password>
```

Replace the bracketed username and password with a valid user and password.

For example:

```
Login Admin Admin
```

Message Overview

SapV2 syntax is based on Axia’s Lwcp specification for syntax, but with some additional features that allow for object and property discovery and listing. As such, the Lwcp documentation can be a useful precursor to reading this appendix.

All messages in SapV2 are case insensitive and end with a carriage return and line feed. Each SapV2 message may consist of an operator, object path, list of properties, and list of system items. The operator and object path are required. For example:

```
get Devices#0.AnalogNode#[tcp://172.16.1.81:93] Name, Description $TXID=555
```

In the message above:

- Operator = get
- ObjectPath = Devices#0.AnalogNode#[tcp://172.16.1.81:93]
- Properties = Name, Description
- System Items = $TXID=555

The list of possible operators is listed and described later in this document.

The object path represents branches of a tree like a file system path where the separator between each branch is a period. Each branch consists of the branch type followed by the pound sign and the unique identifier for the branch. In the example above, there are two branches:

- Devices#0
- AnalogNode#[tcp://172.16.1.81:93]

The branch type of the first branch is Devices and the identifier is 0. The second branch type is AnalogNode and the identifier is [tcp://172.16.1.81:93]. The identifier for the second branch is wrapped in square brackets because the identifier includes symbols that are reserved for object path segregation – namely the period. If a branch identifier includes periods or the pound sign, the identifier must be wrapped in square brackets. Spaces are not allowed within branch identifiers and therefore are not allowed within the object path.

Commas between properties are optional. Commas should not be used between system items.

System items always come at the end of a message and system items all begin with the dollar sign. Therefore, when parsing a SapV2 message, the system item section begins when you encounter an un-escaped space followed by a dollar sign. Since any property value that includes a space must be enclosed in quotes, dollar signs after a space may exist in a property value if they are enclosed in quotes. If a double quote needs to be embedded within a property value, it needs to be escaped with the backslash - \”. If a carriage return and linefeed is to be embedded in the value of a property use the %BeginEncap% %EndEncap% as defined in the Lwcp specification.

For example:

```
indi Devices#0.AnalogNode#[tcp://172.16.1.81:93] Name="AAND-001-081 $Hello" $TXID=555
```

In the example above the property value of Name is:

AAND-001-081 $Hello

The parser does not start parsing the system items when it reaches the space and dollar sign because it is enclosed in quotes as part of the property value. Instead, the system item list begins at $TXID.

In the following example, we needed to include a carriage return and linefeed as part of the value of a property, so we use encapsulation:

```
set MemorySlots#0.MemorySlot#Hello Slotvalue=%BeginEncap%My Name Is
Freddy%EndEncap%
```

In this example, the value of the memory slot will be set to “My Name is Freddy” with a carriage return and linefeed between Is and Freddy. Some properties in the system also accept escape values to represent carriage return and line feed. For example, the generic protocol translator accepts \cr\lf for its ToSend property.

Discovery

Unlike the current version of Lwcp, SapV2 has been designed to extend Lwcp to allow for the discovery for all objects and properties in the system. As in Lwcp, by using the get operator, you can specify an object path and a property and retrieve the property’s value:

```
get Devices#0.Element#[tcp://172.16.1.51:93] Name
indi Devices#0.Element#[tcp://172.16.1.51:93] Name="Element"
```

However, in SapV2 if you do not specify a property, that means you are requesting a list of all the properties and values that the object has.

```
get Devices#0.Element#[tcp://172.16.1.51:93]
indi Devices#0.Element#[tcp://172.16.1.51:93] ExpectLwcp="True", DeviceType="element", Has
```

Additionally, if you place the object path separator at the end of the object path (the period), it means you are requesting the list of sub objects of the requested object.

```
get Devices#0.Element#[tcp://172.16.1.51:93].
indi Devices#0.Element#[tcp://172.16.1.51:93].LwcpInterpreter#0
indi Devices#0.Element#[tcp://172.16.1.51:93].LwrpInterpreter#0
```

In this way, it is possible to walk the tree and discover all objects and properties in the system.

If you wish to know when a list has been completely returned, you can add the $DONE system item.

```
get Devices#0.Element#[tcp://172.16.1.51:93]. $DONE
indi Devices#0.Element#[tcp://172.16.1.51:93].LwcpInterpreter#0
indi Devices#0.Element#[tcp://172.16.1.51:93].LwrpInterpreter#0 $DONE
```

By using the $DONE system item, the last item in the list carries the $DONE item.

If you only specify the period as the object path then you will get all of the root level objects.

```
get .
indi LogicFlows#0
indi Users#0
indi Devices#0
indi Routers#0
indi Meters#0
indi TimeEvents#0
indi Requests#0
indi PFCEmulator#0
indi DeviceEmulators#0
indi Logs#0
indi LegacyPanels#0
indi UpdateModerators#0
indi MemorySlots#0
```

$MAX_DEPTH System item examples

Rather than walking the tree to discover all of the objects, it is also possible to use the $MAX_DEPTH system item to define multiple layers of returned branches at once.

```
get Devices#0. $MAX_DEPTH=2
indi Devices#0.EndpointInvestigator#0 $MAX_DEPTH=2
indi Devices#0.EndpointDiscoverers#0 $MAX_DEPTH=2
indi Devices#0.DeviceCreator#0 $MAX_DEPTH=2
indi Devices#0.MicrophoneNode#[tcp://172.16.1.99:93] $MAX_DEPTH=2
indi Devices#0.AnalogNode#[tcp://172.16.1.81:93] $MAX_DEPTH=2
indi Devices#0.AnalogNode#[tcp://172.16.1.97:93].LwrpInterpreter#0 $MAX_DEPTH=2
indi Devices#0.MicrophoneNode#[tcp://172.16.1.99:93].LwrpInterpreter#0 $MAX_DEPTH=2
```

Every branch that is up to two levels beyond the specified object path is returned in this case. If you leave off the ending period, it will return all of the same object paths above but it will also include all of the properties for each object path.

```
get Devices#0 $MAX_DEPTH=2
indi Devices#0.EndpointInvestigator#0 Abort="False", OpenInvestigatorCount="0", FriendlyNa
indi Devices#0.EndpointDiscoverers#0 FriendlyName="EndpointDiscoverers" $MAX_DEPTH=2
indi Devices#0.DeviceCreator#0 Abort="False", OpenInvestigatorCount="0", FriendlyName="Dev
indi Devices#0.MicrophoneNode#[tcp://172.16.1.99:93] DeviceType="livemic", HasGpios="False
indi Devices#0.AnalogNode#[tcp://172.16.1.81:93] DeviceType="liveio", HasGpios="False", Ha
indi Devices#0.MicrophoneNode#[tcp://172.16.1.99:93].LwrpInterpreter#0 FriendlyName="LwrpI
indi Devices#0.AnalogNode#[tcp://172.16.1.97:93].LwrpInterpreter#0 FriendlyName="LwrpInter
```

Using the $DONE in addition will return $DONE on the last message item in the list.

```
get Devices#0 $MAX_DEPTH=2 $DONE
indi Devices#0.EndpointInvestigator#0 Abort="False", OpenInvestigatorCount="0", FriendlyNa
indi Devices#0.EndpointDiscoverers#0 FriendlyName="EndpointDiscoverers" $MAX_DEPTH=2
indi Devices#0.DeviceCreator#0 Abort="False", OpenInvestigatorCount="0", FriendlyName="Dev
indi Devices#0.MicrophoneNode#[tcp://172.16.1.99:93] DeviceType="livemic", HasGpios="False
indi Devices#0.AnalogNode#[tcp://172.16.1.81:93] DeviceType="liveio", HasGpios="False", Ha
indi Devices#0.MicrophoneNode#[tcp://172.16.1.99:93].LwrpInterpreter#0 FriendlyName="LwrpI
indi Devices#0.AnalogNode#[tcp://172.16.1.97:93].LwrpInterpreter#0 FriendlyName="LwrpInter
```

The value -1 is special to the Max_Depth system item as it represents infinite. Therefore, using a Max_Depth value of -1 returns everything below the current branch. For example, the following command would return all of the objects at any branch level underneath Devices#0:

```
get Devices#0. $MAX_DEPTH=-1
```

This command would return all of the objects and their properties at any branch level underneath Devices#0:

```
get Devices#0 $MAX_DEPTH=-1
```

Use the $MAX_DEPTH with some caution as it can take a bit of CPU power to generate some of these large lists. It is better to subscribe to changes of a certain branch and then cache the results of MAX_DEPTH rather than polling large lists.

It is also possible to use a property name with the MAX_DEPTH system item. For example:

```
get Devices#0.WinDriver#[tcp://172.16.1.254:93] Pinstate $MAX_DEPTH=-1 $DONE
```

Would list every instance of the PinState property that exists below Devices#0.WinDriver#[tcp://172.16.1.254:93] with the $DONE at the end. The response (abbreviated) below might look like:

```
indi Devices#0.WinDriver#[tcp://172.16.1.254:93].LwrpInterpreter#0.LwrpRoot#0.Gpi#10.GpioP
indi Devices#0.WinDriver#[tcp://172.16.1.254:93].LwrpInterpreter#0.LwrpRoot#0.Gpi#10.GpioP
indi Devices#0.WinDriver#[tcp://172.16.1.254:93].LwrpInterpreter#0.LwrpRoot#0.Gpi#10.GpioP
indi Devices#0.WinDriver#[tcp://172.16.1.254:93].LwrpInterpreter#0.LwrpRoot#0.Gpi#10.GpioP
indi Devices#0.WinDriver#[tcp://172.16.1.254:93].LwrpInterpreter#0.LwrpRoot#0.Gpi#14.GpioP
indi Devices#0.WinDriver#[tcp://172.16.1.254:93].LwrpInterpreter#0.LwrpRoot#0.Gpi#14.GpioP
indi Devices#0.WinDriver#[tcp://172.16.1.254:93].LwrpInterpreter#0.LwrpRoot#0.Gpi#14.GpioP
```

> Note: The example above is abbreviated. This would actually return 120 INDIs (24*5 pins).

Multiple properties may also be returned in which case only objects that contain both properties will be returned. For example:

```
get Devices#0.WinDriver#[tcp://172.16.1.254:93] Pinstate, FriendlyName $MAX_DEPTH=-1 $DONE
indi Devices#0.WinDriver#[tcp://172.16.1.254:93].LwrpInterpreter#0.LwrpRoot#0.Gpo#18.GpioP
indi Devices#0.WinDriver#[tcp://172.16.1.254:93].LwrpInterpreter#0.LwrpRoot#0.Gpo#18.GpioP
indi Devices#0.WinDriver#[tcp://172.16.1.254:93].LwrpInterpreter#0.LwrpRoot#0.Gpo#18.GpioP
indi Devices#0.WinDriver#[tcp://172.16.1.254:93].LwrpInterpreter#0.LwrpRoot#0.Gpo#13.GpioP
indi Devices#0.WinDriver#[tcp://172.16.1.254:93].LwrpInterpreter#0.LwrpRoot#0.Gpo#13.GpioP
```

You can also specify a value in a property and then only items where the property matches that value will be returned.

```
get Devices#0.WinDriver#[tcp://172.16.1.254:93] Pinstate="l" $MAX_DEPTH=-1 $DONE
indi Devices#0.WinDriver#[tcp://172.16.1.254:93].LwrpInterpreter#0.LwrpRoot#0.Gpo#18.GpioP
indi Devices#0.WinDriver#[tcp://172.16.1.254:93].LwrpInterpreter#0.LwrpRoot#0.Gpo#18.GpioP
```

Subscription Examples

All of the examples above also work with subscriptions. For example:

```
SUB Devices#0 $MAX_DEPTH=-1
```

This will not return anything immediately. Any time any property beneath Devices#0 changes, it will get returned.

If you wanted to subscribe to all GPIO changes you could send:

```
SUB Devices#0 PinState $MAX_DEPTH=-1
```

Init Examples

Most object may also be created and destroyed using the INIT and DEL operators. The arguments required to create an object can be obtained by sending a get command to the object path of an existing object with the special hidden property called Constructor. See Hidden Properties for details.

```
INIT MemorySlots#0.MemorySlot SlotName="Hello" SlotValue="Hello" Persistent=True
```

In this case, the final object in the path does not have an ID associated with it because the constructing parameters will define the ID as part of the construction process. If we were subscribed to MemorySlot changes, we should see the results of this creation returned as a NEW message.

Delete Examples

Objects may be deleted from the system using the DEL operator and an object path. Properties are not valid in a Delete message.

```
DEL MemorySlots#0.MemorySlot#Hello
```

If the connection has subscribed to changes that match this deletion, there will be a response:

```
LED MemorySlots#0.MemorySlot#Hello
```

The LED operator is the response for a DEL operation but will only be returned if a subscription has been made that matches the deletion object path.

Request for syntax

The RFS operator stands for Request For Syntax and the server will respond with SFR which stands for Syntax For Request. The RFS operator allows you to discover what a specific property expects for its value.

```
rfs Devices#0.WinDriver#[tcp://172.16.1.254:93].LwrpInterpreter#0.LwrpRoot#0.Gpo#6.GpioPin
sfr Devices#0.WinDriver#[tcp://172.16.1.254:93].LwrpInterpreter#0.LwrpRoot#0.Gpo#6.GpioPin
```

We get a variety of information about the property from this command. We know that the property is read/write, that it expects an option and the valid options are l or h. We know that it is a stable property which means that its syntax should not be expected to change in future versions of PFCore unless carefully noted and/or with backward compatibility handling. We also know that the l or h may be displayed in a more user-friendly fashion in a UI as Low or High. And we know that this property is one that should be displayed in the simple tree of logic flows (IsSimpleUi).

The list of syntax types that will be returned is documented in the RFS/SFR notes later in this document.

Routing Paths

It should be noted that in some places in the system (specifically underneath the Routers#0 root object), object identifiers use a URI type syntax. For example:

Routers#0.AxiaAudioRouter#1.AxiaAudioSource#[tcp://172.16.1.71:93?l=SRC&d=src&i=66&t=aaudio]

This identifier (tcp://172.16.1.71:93?l=SRC&d=src&i=66&t=aaudio) is called a PathIo and uniquely identifies an IO in the system. It may be broken up into its relative pieces as follows:

- tcp://172.16.1.71:93: This is the uri of the ip address and port of the device on which the IO was discovered.
- l=SRC: This stands for Level and could contain SRC, DST, GPI, GPO
- d=src: This stands for Direction. This seems redundant to level but it is not. This defines whether the IO is a source or destination irrespective of whether it is a GPIO
- i=66: This is the IO port identifier as reported by the equipment
- t=aaudio: This is the type of IO which could include
    - aaudio: AxiaAudio
    - agpio: AxiaGpio
    - virt: Virtual

Operators

Each operator is comprised of up to 8 characters and are always terminated by a trailing whitespace.

- SET: Request to update information on the system.
- GET: Request for information from the system.
- INDI: Indication that information on the system has changed.
- SUB: Request to receive indications from an object path on the system when changes occur.
- UNSUB: Request to stop receiving indications from an object path on the system.
- RFS: Request for property formats of an object on the system.
- SFR: Response to property format requests.
- INIT: Request to create a new object.
- NEW: Response message for a successful init if subscribed.
- DEL: Deletes an object from the system.
- LED: Response message operator for a delete message if subscribed.
- NOP: No operation.
- SYNC: Used to pass a message for cluster synchronization purposes.
- LOGIN: Login to the system.
- ACK: Acknowledgement from the system to a request tagged with the $ACK system item.
- PUB*: Request to anchor the object path or paths.
- UNPUB*: Request to un-anchor the object path, or paths.
- ATCH*: Attaches to an object (macro for SUB then PUB on the object).
- ATTACH*: Weak alias for ATCH.
- HELP*: Request for helpful information about an object or property.
- LINK*: Request to create a link between an object and a host object.
- UNLINK*: Request to remove a link between an object and a host object.

* For Future use. Not currently implemented.

Untargetted Operators

There are a couple of operators that break the rules about requiring an object path. These operators stand on their own as complete commands:

- LOGOUT: Logout of the system.
-

- EXIT: Request that the system breaks the current connection.
- QUIT: Alias for Exit.

Operator Examples

LOGIN

LOGIN {username} {password}

Example:

```
login Admin Admin
```

Returns:

```
login successful
login failed
```

GET

GET {Object}.{Object} {Property}

Example:

```
get Devices#0.WinDriver#[tcp://172.16.1.252:93] HasGPIO
```

Object may be a specific object. Object may be terminated with a full stop (.), which indicates that a listing of all descendent Objects is being requested.

Object may be terminated with a pound sign (#), which indicates that a listing of all objects matching that type is being requested. *
* For Future use. Not currently implemented.

Property may not be specified, in which case it is a request for all of the properties for that object.

> Important Note: Objects are often returned in alphabetical order to make them easier to read and/or use in a user interface. However, it takes extra CPU power to organize the return list alphabetically. For faster response times and lower CPU impact, the $UNSORTED system item may be used.

SET

SET {Object}.{Object} {Property}={Value},{Property}={Value}

Sets a property value in the system. Use RFS to determine what the property expects and whether it is writable.

Example:

```
SET Devices#0.WinDriver#[tcp://172.16.1.254:93].LwrpInterpreter#0.LwrpRoot#0.Gpo#6.GpioPin
```

Note: A SET will only respond with a change message if there is a subscription active that meets the bounds of the change.

INDI

INDI {Object}.{Object} {Property}={Value}

This is a response from the server for a get command or for a change that is within the bounds of a subscription request.

Example:

```
INDI Devices#0.WinDriver#[tcp://172.16.1.254:93].LwrpInterpreter#0.LwrpRoot#0.Gpo#6.GpioPi
```

SUB

SUB {Object}.{Object} {Property}

Subscriptions can be to an Object Path, an Object, or a Property. $MAX_DEPTH may also be used to subscribe to property changes at different depths.

Example:

```
SUB Devices#0.WinDriver#[tcp://172.16.1.254:93].LwrpInterpreter#0.LwrpRoot#0.Gpo#6.GpioPin
SUB Devices#0 PinState $MAX_DEPTH=-1
SUB Devices#0.WinDriver#[tcp://172.16.1.254:93].LwrpInterpreter#0.LwrpRoot#0.Gpo#6
```

INIT

INIT {Object}.{Object} {ConstructorProperty}={Value} {ConstructorProperty}={Value} ...

Creates an object which is defined as initable within the system.

Example:

```
INIT MemorySlots#0.MemorySlot SlotName="Hello" SlotValue="Hello" Persistent=True
```

DEL

DEL {Object}.{Object}

Deletes an object which may be removed from the system.

Example:

```
DEL MemorySlots#0.MemorySlot#Hello
```

NOP

NOP {Object}.{Object}

This operator stands for No Operation. The rest of this message should be ignored, but it should still be structured as a valid SAPv2 Message. This can be used in configuration files that store SapMessages and essentially will act like remarking out the message. This allows you to leave the message in the file for future use. The system will read and parse the message, but will not act upon it.

Example:

```
NOP Devices#0.WinDriver#[tcp://172.16.1.254:93].LwrpInterpreter#0.LwrpRoot#0.Gpo#6.GpioPin
```

SYNC

Sync {Object}.{Object} Property

This operator stands for Synchronize. It is used exclusively for internal clustering synchronization messages.

Example:

```
Sync MemorySlots#0.MemorySlot#MySlot SlotValue="hello"
```

RFS and SFR

RFS {Object}.{Object} {Property}

SFR {Object}.{Object} {Property}=[{Read/Write},{Syntax},{DirectSubscriptionOnly},{IsStable},{SyntaxUsage}],{Property}=[...]

RFS is a request for syntax request and SFR is the server’s response.

Example:

```
rfs Devices#0.WinDriver#[tcp://172.16.1.254:93].LwrpInterpreter#0.LwrpRoot#0.Gpo#6.GpioPin
sfr Devices#0.WinDriver#[tcp://172.16.1.254:93].LwrpInterpreter#0.LwrpRoot#0.Gpo#6.GpioPin
```

The SFR response may carry multiple pieces of information about the property including:

- Read/Write Values
  - RO: Read Only
  - WO: Write Only
  - RW: Read and Write
- Syntax Values
  - BOL[[true,TRUE,t,1],[false,FALSE,f,0]]: Boolean value with True and False specifics
  - ENU[Value,Value,Value(Display)]: Enumeration
  - LST[Separator,Syntax Type]: List
  - TBL[Header,Header,Header,Header,Header]: Table
  - NUM[Minimum,Maximum,Increments]: Numeric value
  - TXT[Restrictions]: Text value with Restrictions on what type of text
  - BIN: Binary value
  - HEX: Hexadecimal value
  - BMP: Bitmap value
  - PNG: Portable Network Graphics value
  - URI: Uniform Resource Indicator value
- IsSimpleUi: True or False
- IsStable: True or False

Notes on RFS implementation.

- Plain Text: SyntaxType=TXT
- Boolean: SyntaxType=BOL
- Number: SyntaxType=NUM
- Number range
  - SyntaxType=NUM[<Miniumum>,<Maximum>,<increment>]
  - SyntaxType=NUM(1,10,1)
- Enum
  - SyntaxType=ENU[value(display),value(display),value(display)] *
- IP Address: SyntaxType=TXT[Ip]
- Netmask: SyntaxType=TXT[IpNetmask]
- Mulicast IP: SyntaxType=TXT[IpMcast]
- StaticOptionList
  - SyntaxType=OPT[Single:optvalue(display),optvalue(display)] *
- MultiSelectStaticOptionList
  - SyntaxType=OPT[Multi:optvalue(display),optvalue(display)] *
  - StaticSapOptionList   SyntaxType=OPT[SapSingle:(sapPathForList),valueProperty,displayproperty] *

- StaticSapOptionList
  - SyntaxType=OPT[SapMulti:(sapPathForList),valueProperty,displayproperty] *
- DateTime: SyntaxType=TIM
- Color: SyntaxType=COL
- Vb6 OLE Style Color: SyntaxType=OLE
- URL: SyntaxType=URI

*display is optional - if not present uses the optvalue

> Note1: Syntax types with options have the option data (shown in the outer []) expressed in the syntaxUsage field of SapProperty Attribute.
>
> Note2: ENU and StaticOptionList are really the same except the values in ENU must be numbers.
>
> Note3: If the SapSingle is used with an object path of .. , the .. refers to the object path 1 level up from the current object level (example if the property is on an object and the path to that object is devices#0.powerstation#0.lwcpinterpreter#0, .. means devices#0.powerstation#0.
>
> Note4: If the SapSingle is used with an object path of @ , the @ refers to the property object's current object path.

Object Path

{ObjectType}#{ObjectId}.{ObjectType}#{ObjectId}.{ObjectType}#{ObjectId} . . .

An Object path is comprised of a list of branches similar to a folder structure where the period is used as the separator. Each Branch is comprised of a type and an Id separated by the pound sign.

Example:

Devices#0.WinDriver#[tcp://172.16.1.254:93].LwrpInterpreter#0.LwrpRoot#0.Gpo#6.GpioPinState#4

If the id contains any characters that are reserved for the path architecture (. or #), the id must be enclosed in square brackets. Spaces are not allowed in object paths.

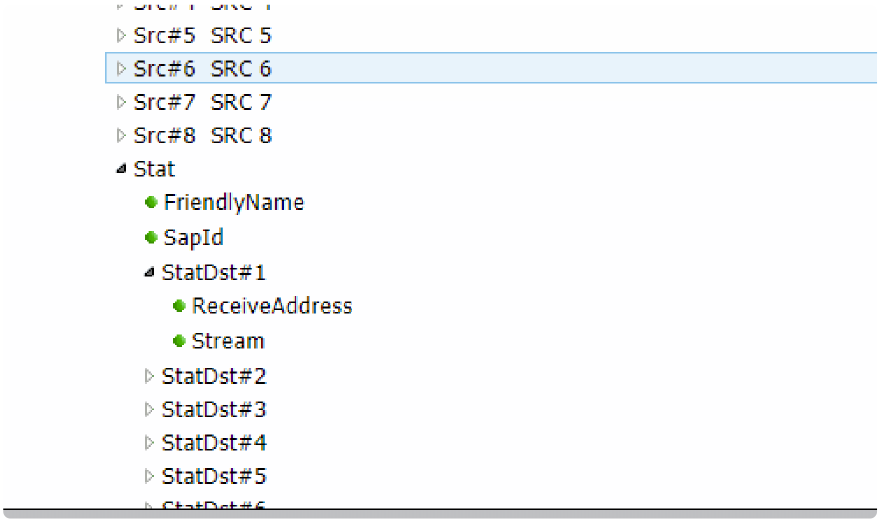
Properties

Property Names may not include spaces and must not begin with the dollar sign. The equals sign with no spaces is used between the property and its value where a value is relevant to the command usage.

```
set MemorySlots#0.MemorySlot#Hello SlotValue="Howdy"
```

Lwrp STAT Properties

This version includes data obtained from the STAT command in Lwrp for devices that support it. These read-only properties may be found in the API tree in logic flows under the device branch that supports it.



- Src#5 SRC 5
- Src#6 SRC 6
- Src#7 SRC 7
- Src#8 SRC 8
- Stat
    - FriendlyName
    - SapId
    - StatDst#1
        - ReceiveAddress
        - Stream
    - StatDst#2
    - StatDst#3
    - StatDst#4
    - StatDst#5

There are four types of STAT objects described below:

Stat Object	Description
STAT SRC	StatSourceAddress = Source Address data.
STAT DST	Stream = Whether the stream assigned to this destination is up or down. Receive Address = Receive address data often used with Aes67 indications.
STAT ICH	AesSync = Only appears on AES inputs and displays whether AES sync is in error or ok.
STAT SYNC	Master – master synchronization data. Slave – slave synchronization data. Sync – Synchonization source. FrequencyAdjust – adjustment value in ppm.

> ⓘ Important Note: Stat Sync data collection is not enabled by default as this process generates a lot of data, especially in systems with large numbers of xNodes. To enable stat sync data collection for a given device, use the API on port 9600 and send the command:

```
SET Devices#0.XNodeCombo#[tcp://172.16.1.95:93].LwrpInterpreter#0 SubscribeToSync=True
```

Also note that while turning the SubscribeToSync option on for a device will be saved between restarts, the state is not currently cluster synchronized

Because the API matches the object layout of the device’s Lwrp protocol some of the new settings can be found in a variety of places. The new paths include:

- LwrpInterpreter#0.LwrpRoot#0.Decoder# LwrpInterpreter#0.LwrpRoot#0.Encoder#
- LwrpInterpreter#0.LwrpRoot#0.Cfg#0.Enc# LwrpInterpreter#0.LwrpRoot#0.Cfg#0.Dec#

For example, the destination properties for a decoder or encoder are available via the LwrpRoot#0.Cfg#0.Enc and LwrpRoot#0.Cfg#0.Dec paths whereas the encoder algorithm and bitrate settings are available via the LwrpRoot#0.Encoder and LwrpRoot#0.Decoder paths.

IPort Control

The *LwrpRoot#0.Cfg#0.GPIO* path controls whether the GPIO ports are enabled or disabled in the system. The actual GPIO port may or may not be present depending on whether the encoder or decoder are active. All of these paths are available for use in Logic Flows via the API tree, however, some are read-only and so will only appear when using a start point.

To figure out which properties match config options in the Iport UI, open a TCP connection to PathfinderCore PRO using Putty to port 9600 and subscribe to changes on the device in question.

```
sub Devices#0.ZephyrIPort#[tcp://172.16.1.71:93] $MAX_DEPTH=-1
```

Then you can make changes in the IPort’s UI and see the changed properties in Pathfinder Core PRO. This can be useful to see where certain properties are located in the API tree.

Hidden Properties

The Constructor property will return Init messages required to recreate an object that supports that property. However, it will not show up in the general list of properties for an object. It must be queried directly.

```
get MemorySlots#0.MemorySlot#Whatever Constructor
```

Property Values

If the property value is to include a space, it must be surrounded by quotes.

```
set MemorySlots#0.MemorySlot#Hello SlotValue="Howdy There"
```

If the property value is to include a carriage return and/or line feed, it must be encapsulated using %BeginEncap%%EndEncap%.

```
set MemorySlots#0.MemorySlot#Hello SlotValue=%BeginEncap%Howdy There
All
You
Beautiful
```

```
People
%EndEncap%
```

If a double quote needs to be included in the value, it needs to be escaped using the backslash.

```
set MemorySlots#0.MemorySlot#Hello SlotValue="\"Howdy There\""
```

System Items

- $ACK: Request Acknowledgement to Operation.
- $TRXID/$TRXI/$TXID: Transaction ID to track response to Operation.
- $IND/$INDI: Request Indication from SET without needing a subscription.
- $DONE: Requests that the DONE system item be applied to the last message returned from a request.
- $PROPATTR: Specifies that the properties in the GET/RFS messages are Syntax Format properties - such as getting all properties that are Readwrite=RO.
- $OP: Operation. Used in ACK Operation responses to denote the Operation that was sent with the $ACK System Item. Also used in Subscription operations where you want to specify only messages with a specific operator. For example DEL or LED messages. (eg. sub MemorySlots#0 $MAX_DEPTH=-1 $OP=LED).
- $MAX_DEPTH: Used to specify the Depth of the subscription or return. -1 equals all object paths below the request.
- $UNSORTED: Return messages do not need to be sorted for easier readability. Used when querying large lists where order does not matter.
- $CONTAINS_PROP: Indicates a request for objects that contain the specific property. If message ends with object separator (.) then it will just return objects. Otherwise, it will return all properties of the objects that include the requested property. This is not valid for subscription messages. It is only valid for Get. Please note that this can cause heavy cpu load as it requires walking the object tree and iterating all properties in each object. Use sparingly and cautiously.
- $EXCLUDE_PROPS: Exclude certain properties from the return.
- $SHOW_CLUSTER: Certain objects related to clustering are hidden and not returned in subscription and get responses unless this system item is included
- $CLUSTER_CHANGES: Subscribe to only changes that are related to cluster synchronization. Returns will carry this as well.
- $CLUSTER_QUIET: This has special meaning to Set and Init messages. It means the message is the result of a Change on another server and is being set to keep in sync. And therefore the change should not be broadcast to other servers that have subscribed to cluster changes to prevent numerous duplicate messages.
- $LAST_UPDATE: Used in conjunction with $CLUSTER_CHANGES to show the date/time stamp of the most recent update.
- $INCLUDE_MOUNTS: This will also follow and include mount points from an object to a mounted object. For example, from a route point to the mounted device IO. It applies to Get, Set, and Sub

operator. Mount points are currently used sparingly within PathfinderCore PRO so this option is only useful in come situations.

- $STATUS*: Response status code
- $COUNT*: Request for number of objects below the current object path
- $DEPTH*: Request for an operation to be iterated to a specified depth. Default is 1, and a non-positive integer depth will result in infinite depth.
- $TSTAMP*: Time stamp.

* For Future use. Not currently implemented.

Additional System Item Notes

$PROPATTR is the system item for Property Attribute. It can equal AND or OR.

$PROPATTR=AND <Default>
$PROPATTR=OR

If this attribute is set it means that the properties defined in the property list of the Get/RFS message are not literal properties. Rather, they are requesting objects that have properties that have formats described in the property list.

For example:

```
GET Devices#0 ReadWrite="RO|RW" SyntaxType="TXT" $MAX_DEPTH=2 $PROPATTR=OR
```

The pipe allows listing multiple options for a format value. The And/OR in the PropAttr defines whether a match is made if either of them matches or only if all match when multiple properties exist in the message.

Release Notes

For release software update packages, visit http://pathfinderpc.com/releaseDownloadsCorePro.htm

Click on the subsection for the appropriate major/minor version to view release notes.

1.8.xx.xx

Version 1.8.11.01 02/21/2022

> Warning: This version is a special build. Please review the release notes below to determine if 1.8.11.00 or 1.8.11.01 is more appropriate to your environment.

- This version should be functionally the same as 1.8.11.00. However in order to try to address an issue that has been only rarely encountered, this version pulls in the operating system and framework updates from the beta branch. If you have an issue with PathfinderCore PRO periodically (weeks or months apart) restarting and support reviews the backup and determines that there is a specific segfault error message in the messages log, this version may be recommended. Otherwise most customers who wish to stay in the release rather than beta track can continue to use 1.8.11.00.

- Fixed an issue with LWRP route changes on mix engines where an algorithm designed to prevent an overflow of the command buffer in the engine was introducing more delay than necessary when making large numbers of route changes at once.

Version 1.8.11.00   02/21/2022

- Fixed an issue with LWRP route changes on mix engines where an algorithm designed to prevent an overflow of the command buffer in the engine was introducing more delay than necessary when making large numbers of route changes at once.

Version 1.8.10.01   02/10/2022

> ⓘ Warning: This version is a special build. Please review the release notes below to determine if 1.8.10.00 or 1.8.10.01 is more appropriate to your environment.

- This version should be functionally the same as 1.8.10.00. However in order to try to address an issue that has been only rarely encountered, this version pulls in the operating system and framework updates from the beta branch. If you have an issue with PathfinderCore PRO periodically (weeks or months apart) restarting and support reviews the backup and determines that there is a specific segfault error message in the messages log, this version may be recommended. Otherwise most customers who wish to stay in the release rather than beta track can continue to use 1.8.10.00.

Version 1.8.10.00   02/09/2022

- Fixed an incorrect calculation of the memory percentage on the system web page.
    - This calculation was including disk buffer cache which is not actually consumed or unavailable memory.
- Fixed a bug that was leaking a small amount of ram each time a tcp connection was connected and then disconnected from port 93 of the internal GPIO node.
- Added code to restart the internal GPIO node if the process dies.
- Fixed a bug with the list that kept track of port 93 connections to the internal GPIO node that in rare situations might cause the GPIO node application to crash due to the list not properly supporting multi-threading.

Version 1.8.9.00   01/25/2022

- Fixed an issue with cluster synchronization of licenses when certain time zones were in use.
    -

This is in response to an issue where a backup license was not obtaining the correct license count from the primary server. Log messages showed UTC offset errors. We discovered one place in the code where we were using an incorrect DateTime comparison in relation to licenses which particularly was affected by certain time zone offsets. This should be fixed in this version.

Version 1.8.8.00 12/07/2021

- Fixed an issue where the AUDIO_MODE property of VMODEs was set to read only when it should be read/write. The property should now be available as end points in logic flows.
- Fixed an issue where the FpStatPollRate property could not actually be changed.
  - Additionally made a change where assigning an FpStatPollRate value of 0 (or any value under 100) would stop the polling of FpStat data.
  - To retain the setting between restarts, use the Advanced Options.
  - Note that advanced option changes require a restart to take effect.

Version 1.8.7.00 10/14/2021

> ⓘ Note: This version also contains the fixes from the unreleased version 1.8.6.00. Please review those release notes as well to understand all changes in this version.

> ⚠ Warning: Changes had to be made to the build container used to compile the release branch of PathfinderCore PRO. While these changes should have no effect, there is always a chance something will compile and behave slightly differently. Please report any issues you may encounter.

- Fixed issues with cluster synchronization of the Internal GPIO node's pin states.
  - In some situations double pin entries would be expressed in some of the nodes. While this should be benign, in some external use cases, it was causing unexpected behavior. Work was done in this version to try and address that issue.
- Updated the build container used to build the product.

Version 1.8.6.00 10/07/2021

> ⚠ Warning: This version was not released to the public due to testing errors. The fixes listed below are also present in 1.8.7.00.

- Fixed a bug with the internal GPIO node where clearing a route on a GPIO would not reset the pins back to high.
- Fixed a bug where the internal GPIO node was responding with a change message in Lwrp even if no pin state changed.

- Fixed bug when changing properties in the user panel properties list by clicking on the name column.
    - Clicking on the name column rather than the value property would sometimes not cause the metadata in the translation properties to update properly. This would mean clicking on the active translation afterwards would present the previous item's drop down items.

Version 1.8.5.00 07/23/2021

- Fixed a bug with subscriptions in logic flows where the need to subscribe/unsubscribe was being counted per object flow and property, but the actual sub and unsub was only using the object.
    - When multiple flows used different properties on the same object, the removal of one of them could cause the whole object to get unsubscribed and the remaining flows using start points on that object to stop functioning until a reboot occurred.
    - This occurs during flow editing and not spontaneously on a running system.
    - This bug has probably existed almost since the beginning of PathfinderCore PRO and may explain a number of anomalies systems integrators have reported when first setting up flows at a site.
- Added code to handle input and select box change events when the change is the same as the last requested change but might not actually be the current value.
    - Previously this condition would not issue a change command.
- Fixed a bug with the gpio node where if multiple ports had the same multicast channel number only one would be returned for processing.
    - This could lead to only one port firing or none if the one returned was in the wrong mode (Node/Console) to fire.
- Removed a logging message regarding udp buffers that was an unnecessary holdover from debug testing and was displaying in the supervisor log.
- Fixed an issue where when a SapProperty router source or destination was tied to a different route point, a lock recursion error could occur in the logs during loading.
    - This was because the sub object also was trying to be mounted.
    - This fix addresses that problem and prevents the write lock recursion log messages.
    - We do not think this additional mounting attempt was causing any other issues other than the additional log messages.
- Fixed an issue where SapProperty sources and destinations were trying to be added to a lookup table where they were not needed.
    - This was causing log messages about trying to add duplicate sources and destinations
    - We do not think this was causing any issues other than the erroneous log messages
- Fixed a spot in the code where logic flow folder creation messages might get created in the file system with a message that was missing the trailing LogicFlowFolder in the object path
    - This appeared to be benign as the folders still loaded.
- Added code to correct for init messages in logic flow folder init files if they were missing the trailing LogicFlowFolder in the object path.
    - In cases where it was a first level folder, this was causing a constructor parameter list error in the logs trying to create root level logic flows.
    -

The error appears to be benign but this should fix it.

- Added an exception for the device emulator log message about redirecting the initial id message as it is expected message.
- Downgraded the "Setup and Startup from connection function complete" log message to debug state as it is not informative in normal logs.
- Cleaned up the "deviceid branch not found" log messages as they are expected and not informative.

Version 1.8.4.00 06/18/2021

- Fixed a bug which was introduced in 1.8.3.00 with the internal GPIO node where pulse messages (DURATION) would cause additional pins on the port to flip state.

Version 1.8.3.00 06/04/2021

- Fixed an issue where SapProperty memory slots might not subscribe to changes and get their initial state from their underlying properties after a reboot.
  - The subscription message was being sent before the internal connection to the SapMessage engine was being set up. This is now fixed.
- Fixed an issue with the internal gpio node where it would accept pin state letters other than l, h, and x.
- Fixed an issue with the CMD option in the internal gpio node when used in a cluster.
  - Using the CMD command on a gpio pin would cause a message to bounce back and forth between the Pathfinder internal gpio nodes in the cluster. This is now fixed.

Version 1.8.2.00 04/28/2021

- Updated the build process to work better with the Telos Alliance continuous integration pipeline.
- Added code to prevent a double entry which sometimes occurred after dropping a new event onto a calendar and then switching views.
- Fixed an issue where recurring events set to Saturday were not displaying on Saturday in the calendar.
- Fixed a bug with the timer list display where it was not always showing the correct state of timers without a browser refresh.

Version 1.8.1.00 04/14/2021

- Fixed an issue with timers where the new end date/time in date time and day of week timers was not being updated to the new offset after a daylight savings time shift.
- Fixed an issue where when time jumps from before a dayofweek timer event to a new day, that could cause the event to raise last raised repeatedly when it executes.
- Fixed issues with generic emulators that use UDP client and UDP listener.
  - In some cases the UDP client would stop sending data after a failed attempt.
  - Additionally this version adds a local port to UDP client for bi-directional communication.
  -

This change is cherry-picked from 1.9.0.03.

- Reworked syslog log writers to use the new UDP client changes from generic emulators
  - This fixes an issue where the UDP log writer might stop sending data to the syslog server after it fails to send a few messages.
  - There is a possibility once that stoppage happened that the UDP client might also leak a small amount ram.
  - These issues with UDP syslogs should be fixed in this version, but the fixes did require overhauling the UDP client technology so report any issues you may encounter.
- Fixed an issue where None sources as base points pointed at a virtual router base point might not clear properly due to an incorrect ordering of the destination base point return.
- Fixed an issue where some LWCP messages were bypassing logging options when Lwcp message logging was enabled.
- Added code to skip source list polling on Qor versions earlier than 2.4.0.2 in order to prevent triggering the source list query memory leak in Qor software earlier than that version.
  - Updating Qor to 2.4.0.3 or later is recommended to regain the best functionality with PathfinderCore PRO.
- Modified the compilation and build procedure to be more inline with the continuous integration process of other Telos Alliance projects.
- Added some code to streamline event data changes in a calendar in the scheduling system.

Version 1.8.0.00 02/20/2021

> ⓘ Note: This version is identical to 1.7.13.23 but updated to the release version number. Patches to any bugs will be released as patch update and described here. New features will move to a 1.9 beta version.

- Changed the documentation links in the software to point to the 1.8.x.xx version of the manual and remove links to beta documentation.

1.7.xx.xx

Version 1.7.13.23 02/19/2021

> ⓘ Note: This version is the last in the 1.7 beta branch before 1.8 release. Please use the selector in the upper left corner of this documentation to switch to the desired software version to see additional release notes and features.

> ⓘ Warning: This version introduces a change which applies only to newly factory built Pathfinder Core PRO units on the AE-2000 (2021) platform which ship from the factory with 1.8 software on them. In an effort to bring all products that use this platform into alignment, the left and right nic will be switched as to which is the AoIP/Livewire port and which is the Office port. This will not affect any units in the field or previously shipped units and updating units to this version of the software will not change your ports. Newly built units will have a flag generated during the production process that indicates this change. Systems integrators who are used to having the nics the way they previously have been should take note when new units start arriving.

- Added code to the NIC analysis script on startup that looks for the presence of a file generated during production that indicates that the Right NIC looking at the back of the unit should be the AoIP/Livewire NIC and the left one should be the Office NIC.
  - This only affects newly built units on the AES2000 (2021) platform shipping with the 1.8 software pre installed.
- Fixed an issue with an incorrect factory default DNS addresses.
  - By default PathfinderCore PRO ships with Google DNS addresses
  - It is recommended to change that as your IT department recommends
- Added a ReconnectLwcpVx write only property to the VXEngine device type for triggering a reconnect of the LwcpVx connections and used that in the device recycle icon in the devices list.
- Fixed an issue that was causing some LwcpVx connections not to recycle after a loss of connection.

Version 1.7.13.22   02/14/2021

- Fixed a bug with the console monitor object when pointed at Qor, iQx, and iQs consoles that would cause the studio selection option to only display PGM1 as an option.
- Fixed a bug where the time clock was not advancing on the full console widget.
- Fixed a bug with the countdown clock on the full console widget not working.
  - See Important note below to enable this functionality.
- Fixed a bug with the countdown clock not displaying the run button if the automatic mode was enabled on a Qor, iQs, or iQx console.
- Fixed a bug with the console fader component where if you are dragging the touchbar and release the mouse while over a disabled button, the touchbar would remain captured.

> ⓘ Important note: This note only applies to the full console component as opposed to the individual section components. The fixes for the countdown clock on the full console component required the changing of object types and the addition of clock parameters which must be enabled for it to work. New console widgets dragged onto a panel will work by default, but in order to get an existing console's countdown parameters to work, remove the console component from the panel, save the panel, and then drag a new one on and save again. You have to save without the original console component before adding the new console component and saving again.

Version 1.7.13.21 02/08/2021

- Added heartbeat and fader source list polling to lwcpss connection to detect a failed connection sooner.
- Changed property decoration on the lwcpss source object's id, nextid, and name properties to display a list of source profiles in the logic flow property value helpers.
- Added lwcpss into the connections that get recycled through the device list recycle icon.

Version 1.7.13.20 02/05/2021

- This build represents a merge of the beta branch into the master branch in preparation for a new release.
- Fixed some device handling around iQs.
- Completed task 15330.
- Fixed an issue with numeric memory slots where negative values cannot be used in the range (i.e. -1-10 or -10--1)
- Fixed an issue where fusion and quasar engine did not have console_type, console_ip, and audio_mix_mode properties available.
- Fixed an issue where fader lwcp sanity polling was not always working properly
- Added src_list to the fader poll in order to pick up changes in the source profile list
  - Fader polls go through each fader in approximate 30 second intervals, so it might take a while before new profiles appear
  - If they need to appear quicker, recycling the device connection can force a re-query.
- Added a timer for consoles which 60 seconds after reconnect will requery the fader src profile list for each fader at 500ms intervals.
  - This handles the situation where the console allows lwcp to connect and be queried before the source profile lists are loaded.
- Update saconsolefader when used with iQs to match iQx.
- Fix a bug with instantiating numeric memory slots when including both a value and a range.
  - This was causing the creation not to take place without creating it without the range and then editing to add the range.
- Fix an issue with various console components which use metering not being able to derive the correct metering point when the control device and engine device are separate such as with quasar.
  - Use the control point (as opposed to the io point) for these situations.

Version 1.7.13.19 12/07/2020

> ⓘ Warnings:
> Please review the release notes and Beta Documentation for 1.7.13.15 and 1.7.13.18 as this version has all of those new features.

-

Fixed places in the web user interface where ShowProfID case sensitivity was incorrect which could potentially cause additional items to show up in the logic flows tree.

- Added code to force any instances of ShowProfId to ShowProfID.
- Fixed an issue in virtual routers that was preventing the source packages with more than one None from presenting properly in the web UI.
- Added the iQs device type to devices.
- Exposed the livewire channel and a few other properties in Axia Audio Sources to the API view.

Version 1.7.13.18   11/22/2020

> ⓘ Warnings:
> Please review the release notes and Beta Documentation for 1.7.13.15 as this version has all of those new features.
>
> The changes below have had minimal testing at this point in time.

- This version makes some modifications to aes67 source routing when applied to mix engine vmix fader inputs.
  - Proper AES67 Lwrp syntax in the Mix Engine and Power Station is only applicable to VMODE. However if the stream in question matches the normal livewire UDP settings of 24 bit, 2 channel, 48Khz and is within the livewire multicast ip range on UDP port 5004, it may be used on vmixers. This version alters the Lwrp messages to support this:
- ===Sending===
- If route attempt requests an AES67 source.
  - If the AES67source is fmt type (instead of sip) and the destination is of type VMIXFaderIn and the fmt type is ";fmt=L24/48000/2":
    - If the multicast range starts with 239.192. or 239.193. and the port is 5004, then send multicast address only to the ADDR field.
    - If the multicast range does not start with 239.192. or 239.193. or the port is not 5004 then use mcastaddress:port.
      - This may not be accepted by the engine at this point in time.
  - If the AES67source is not the ";fmt=L24/48000/2" type or the destination is not of type VMIXFaderIn then send the full AES67 format style message.
    - This will be accepted by the engine for VMODE but likely not for VMIX at this point in time.
- ===Receiving===
- Attempt a normal lookup in the source address hash table based on what is in the ADDR field first.
- If that fails:
  - If ADDR has fmt and/or sip, then do nothing else as the correct lookup has already occurred.
  - If we receive address:port in the ADDR field with no fmt and no sip, then upconvert the message format to AES67 full format by appending the ";fmt=L24/48000/2" and try the lookup again.
  - If we receive address with no port and we have not found anything, then upconvert to the full path by

appending the ":5004;fmt=L24/48000/2" and try the lookup again.

- Note that this introduce additional lookup attempts if the address field contents are not known in the first lookup which could theoretically have some minimal performance impact.

Version 1.7.13.17 11/20/2020

> ⓘ Warning:
> Please review the release notes and Beta Documentation for 1.7.13.15 as this version has all of those new features.
>
> This version fixes an important bug introduced in 1.7.13.15 which prevented the editing of virtual routers.

- Fixed issue created by the none source routing work in 1.7.13.15 that was breaking the ability to add virtual Ios into a virtual router.
- Fixed an issue where adding None source base points into a virtual source package would not actually clear the route since the None source path has no IO type.
  - This fix moves none sources into the correct IO type package in the correct order when getting the ordered source list.
  - It accomplishes this by looking at the iotype of equivalent orderindex in the destination package.
  - This means using None sources in a virtual source package requires matching the orderindex between source and destination.
- Added active_layer property to the Quasar Appcontrol object which accepts values of 1 to 4.
  - This allows monitoring and changing the active layer in Quasar.

Version 1.7.13.16 11/04/2020

> ⓘ Warnings:
> Please review the release notes and Beta Documentation for 1.7.13.15 as this version has all of those new features.
>
> This version is primarily a bug fix version to fix a few things found with the new features in 1.7.13.15.
>
> Panels that have been edited using 1.7.13.15 should be edited again and resaved due to the bug described below.

- Fixed a bug introduced in 1.7.13.15 that was causing the link to the theme css file for user panels to be incorrectly stored in the panel as an absolute address with the ip address of the saving host.
  - This would break panel displays in a cluster if the machine that stores the panel was not active.
  - This may also break panel displays if DNS or NAT was used.
  -

    To fix this edit any panel you edited under 1.7.13.15 and click save again to reset the link to a relative link.

- Fixed a bug introduced in 1.7.13.15 that was causing the panel title in the browser tab to show a fixed panel name of iQs rather than the correct panel title.
- Fixed a bug introduced in 1.7.13.15 that was causing the None source selection for virtual routers in the route table take list to report no source selected.
- Fixed an issue where scheduling system templates were not synchronizing across the cluster correctly if the Assigned By Resource checkboxes were enabled.

Version 1.7.13.15 10/30/2020

> Warning:
> This version represents a feature freeze of the beta branch in advance of a 1.8 release.
>
> Bugs will continue to be addressed during the final beta period in the 1.7 branch until we release as 1.8. New features will be created in a new 1.9 beta branch.

- From 1.6.13.00: Fixed an issue that would prevent user panels from working if the security settings denied access to the main Devices object since their ability to work relies on being able to check the service load state and the ping property.
    - Get operator only is granted access to that object's few properties for all valid user logins so that these properties may be queried.
- From 1.6.13.00: Added some error trapping to catch a rare situation where an ip client in the process of being disposed and sending at the same time might cause a crash.
- From 1.6.13.00: Removed an error message from the logs regarding the failure to remove meters from a lookup collection that is not actually an error.
- From 1.6.13.00: Clarified an info log message to make it less error-like regarding Qor Lwcpss message handling for amix
    - "amix" parameter support currently requires the beta version.
- From 1.6.13.00: Fixed an issue with the userpanels xy grid and locks with virtual routers.
- From 1.6.13.00: Added code to decode ampersands properly in userspanel properties that use url as their property type in the property grid.
- From 1.6.13.00: Fixed an issue where the user panels list web page still displays notes and warnings about using legacy panel designer.
    - These notes should have been removed in version 1.4.

> Warning:
> This version has major additions to user panels as outlined below and in the beta documentation.
> Please review the Beta Documentation on these changes.
> Please report any bugs or feedback you encounter with the new features to support.
> Styles and colors may still change in this theme during the beta period before release.

- Added a dark theme to user panels.
- Modified the user panel's monitor section control to dynamically adjust its control layout when used with Fusion or Quasar in addition to its current ability to be used with Qor/iQx.
  - For Fusion this requires Fusion software version 3.2.1.28.
- Added a Quasar style to user panel analog clocks and countdown timers.
  - This style is the default when the dark theme is chosen.
- Added an advanced option to specify the theme to use for new panel creations.
  - SET UserPanels#0 DefaultTheme=default
  - Requires a restart to become active.
- Added a full console control to user panels.
- Made console fader controls dynamically use the Lwcpss metering when an iQs control point is chosen.
- Added a fader dock control to user panels.
- Added a program buss metering module control to user panels.
- Added a Qor/iQx fader processing module to user panels.
- Added swap to mon talk_stud and mon cr_pvw properties.
- Added Lwcpss meters as an option in the metering service.
  - sm LwcpSs 172.16.1.89/ch#1
  - sm LwcpSs 172.16.1.89/pgm#1
- Added some code to set meters to -1000 if we get a DST ADDR:255.255.255.255 or ADDR: message in order to clear a meter when an engine resource gets unrouted and then disappears from the engine.
- Added a pollingpaused property to the meter data which stops sending messages to the mix engine on a DST that has an ADDR of 255.255.255.255 since the engine removes such a DST from the DST list and will start responding with error1000 messages.
- Fixed an issue where engine meters were getting hung when the destination changed to a none source.
- Added consistency to how virtual routers and sap property routers report CurrentSource and CurrentSourcePath when a none source is routed.
  - CurrentSource now reports 0 rather than no entry and CurrentSourcePath always reports :// and uses those as the entries for that IO in logic flows for correct matching.

> ⓘ Warning:
> This could be a breaking change if your flow is specifically looking for empty string for CurrentSource for virtual routers.

- Fixed an issue where sap property routers were not presenting none and previous in the api source options.
- Fixed other places related to the virtual router where source path for none was being replaced with a full path.
- Fixed a bug where SapProperty sources and destinations which have another IO as their target might generate recursion mount error messages in the logs
  -

    Changed MountComplete to be true only if the underlying mapping either does not require mounting to be mount complete or has achieved mount completion.

- Update javascript packages.
- Added a logout link to all web pages.
- Fixed a bug with calendar viewer not being able to display recurring events.
    - This was broken as of 1.7.6.08.
- Fixed a bug where it was not possible to change a template type from one time to recurring.
- Fixed an issue with bank updating that prevented bank updating from working over NAT or DNS.
- Enforced route tables showing blank source io if the source io is 0.
- Cleaned up presentation code for sap property routers for none routing.
- Added code to allow None to show up in the available options for base io imports to virtual routers.
    - This is useful when you want to create a virtual source that only clears part of the underlying multiple base points.

Version 1.7.12.14 09/25/2020

- From 1.6.12.00: Fixed an issue with user panel countdown clocks where the change of countdownlength was not being raised to sapV2 and therefore would not update data in the countdown clock without refreshing the panel web page.
- From 1.6.12.00: Fixed an issue with user panel countdown clocks where reloading the page or changing the countdown time on a page where the timer had elapsed but not been reset would show the count down duration instead of 0.
- From 1.6.12.00: Added code for user panel countdown timers in countdown mode to display 0 if countdownlength minus elapsed milliseconds is less than zero.
- From 1.6.12.00: In the User Panel Designer, added an alert and block to the countdown timer -- countdownlength, stopmode, and resetmode properties to prevent them from being unbound as binding must be left on for them to function properly.

Version 1.7.11.14 09/22/2020

- Improved the Delay combiner performance when it is tripped multiple times within the delay time.
    - The multiple instances of the delayed task were being created in a limited pool which could cause laggy behavior in rare situations.
    - This has now been changed to use a more optimized task delay mechanism.
    - Similar changes were also made to timers in the UserPanel property, panel, and Countdown Clock as well as in a memory slot write function.

Version 1.7.11.13 09/11/2020

- From 1.6.11.00: Fixed an issue with UserPanels where it is possible to start receiving state messages before the bindings file is loaded leaving those states with no place to go.
    -

    This could cause some panel objects to have an incorrect starting state when the panel gets loaded.
  - To address this problem, this version does not connect the web socket until the bindings file has been loaded.

Version 1.7.10.13 09/10/2020

- From 1.6.10.00: Fixed an issue where phase 1 of the Lwcp state initialization was sending GET SUPV STATE whereas Quasar would only accept GET supv state.
  - The result was that with Quasar we would often not get to phase 2 and therefore not get initial states for a variety of parameters until those parameters changed.
  - For example, a user panel label bound to the show profile name would likely be blank on system startup until the show profile was changed or we happened to poll that parameter.

Version 1.7.9.13 09/01/2020

- Fixed an issue where prior to this version the \%XX pattern for escaping hex binary bytes in the generic emulator client was limited to the ascii range (bytes 0 to 127).
  - While this version now supports \%00 to \%FF, a true binary client is planned for the future.

Version 1.7.9.12 08/26/2020

- From 1.6.9.00: Fixed an issue where generic emulators set to TcpClient mode were not raising connected change events.
- From 1.6.9.00: Fixed Connected property for Device Emulator listeners so that it will be true if at least one client is still connected and False otherwise.
- From 1.6.9.00: Fixed an issue where subscribing to a GenericEmulator would yield a ToSend event with an empty string for each ToSend attempt.
  - ToSend is a write only property and should not raise events.
- From 1.6.9.00: Fixed Generic Emulator ConnectedCount not displaying the correct count and not changing when tcpclient is selected as the connection type.
- In Html5 User Panels, made the box-sizing css style available in all elements.
- In Html5 User Panels, made the text-align and line-height css styles available in labels.
  - Note that centering text vertically in a label can be a bit finicky in html labels.
  - Using the line-height style can help with this by setting the line height to the same height as the label or half that height if two lines will be displayed.
- Added a PingMessage and PingInterval field to Generic Emulators.
  - When using a tcp client connection, connection drops are detected automatically if the remote device properly closes the connection.
  - However in the case where the device loses power or closes the socket unexpectedly, we may not know that until the next time we try to send a message.
  - These properties allow you to define a message and interval to send that will make sense to the remote device and will allow you to be sure the connection is still good.

  - Please review the documentation on this feature: Beta Documentation
- Added an hp202 object with a ConsoleIp property to the hp202 device and hooked up the initial sync command as well.
  - Note that because we do not have an hp202 for testing, we cannot be certain this feature will work and need hp202 users to test it and provide feedback.

Version 1.7.8.11 08/15/2020

- From 1.6.8.00: Fixed an bug introduced in version 1.6.6.00 where user logs were no longer accessible from the web ui and were instead showing a forbidden access error message.
- From 1.6.8.00: Fixed a bug where the disabled property on Html5 user panel buttons was not being respected by touch actions on touch screens.
- From 1.6.8.00: Added a blankvalue property to the saselector drop down control in html5 panels.
  - The blankvalue property allows you to define a value to use if the value returned is blank.
  - For example, the manual example refers to using the currentsource property of a virtual destination and filling the list with virtual sources.
  - The None source has a value of "0" but often a cleared destination route is returned as a blank field instead of 0.
  - Using 0 in the blankvalue property will force the empty currentsource to be interpreted as 0 which matches the none source causing it to be displayed.
  - Without this property being configured, a destination with nothing returned as the current source will simply not make a change to the current item in the list causing it to retain a route where none actually exists.
  - This has been implemented as a separate field because each list might or might not require different values to be used or not used for a blank field.
- Made mon object's hp_vol, cr_vol, pv_vol, and st_vol read/write instead of write only now that the fusion/quasar code supports it.
- Added the button-panel-background-color property to the Html5 user panel saroutingmatrix control.
  - This can be used to change the color behind the clear/take/etc buttons.
  - In order to make this background transparent, shift click to manually enter a value in the field and type the word transparent.
- Added --option-background-color and --option-color properties to the Html5 user panel saselector control.
  - These can be used to configure the background color and font color of option items in the drop down box.
  - Note that the selected item background and font color will still be defined by the OS as this is an OS setting.
  - Offering control over the selected item colors would require using a custom rather than standard drop down component which will be considered for the future.
- Made the Other source greyed out and unselectable in the routing source take list as it cannot actually be sent as a change value.

- Added the hiddenvalues and disabledvalues property to the html5 saselector control.
  - These properties can be filled with a comma delineated list of values which if they appear in the option list will be disabled or hidden.
  - Note that values and not the displayed names should be used in these fields.
  - For example, if the source list is filled with virtual sources, you could hide or make unselectable the previous and other source by using the value:
    - -1,-3
  - in this field.
- Added a sortby property to the html5 saselector control which accepts values of None, Value, or Name.
  - This will cause the options displayed in the list to be sorted by the order in which they were received, the order of the options value, or the order of the text used to display the item.

Version 1.7.7.10 07/17/2020

- From 1.6.7.00: Change the factory default file to alter the timezone localtime symlink rather than overwrite the file.
  - In some cases this was overwriting the underlying symlinked file causing a specific timezone to show Eastern offset rather than the correct offset for the timezone after a factory default.
  - Bank updating would fix the problem as well until the next factory default.
- From 1.6.7.00: Added a question to the r2 production install script to ask if it is being installed on the mbx platform.
  - If the answer is yes, the crystal fontz display application is disabled and we use the new default ip addresses in the ip address questions.
- Added a subscription message for the mon object during connection to Pwerstation, Fusion, and Quasar as some new properties require a scubscription.
- Added the Read/Write hp_lnk (Headphone Link) property to the mon object in PowerStation, Fusion, and Quasar.
- Added the Write Only hp_vol (Headphone Volume) property to the mon object in PowerStation, Fusion, and Quasar.
- Added the Write Only cr_vol (Control Room Volume) property to the mon object in PowerStation, Fusion, and Quasar.
- Added the Write Only pv_vol (Preview Volume) property to the mon object in PowerStation, Fusion, and Quasar.
- Added the Write Only st_vol (Studio Volume) property to the mon object in PowerStation, Fusion, and Quasar.
- Added the Read/Write cr_sel (Control Room Select) property to the mon object in PowerStation, Fusion, and Quasar.
- Added the Read/Write hp_sel (Headphone Select) property to the mon object in PowerStation, Fusion, and Quasar.
- Added the Read/Write st_sel (Studio Select) property to the mon object in Quasar only.
- Added the Read/Write cr_pvw (Control Room Preview) property to the mon object in Quasar only.
-

Added the Read/Write cr_pvw (Headphone Preview) property to the mon object in PowerStation, Fusion, and Quasar.

- Added the Read/Write talk_stud (Talk to Studio Switch) property to the mon object in PowerStation, Fusion, and Quasar.
  - Does not support initial state get but does support events so changes should be reported but not initial startup state.
- Added the Read/Write Asg_AUX(1-8) (Aux buss assignment) properties to the FaCH and LwCH objects in Quasar only.
- Added an aux object to Quasar FaCH and LwCH objects.
- Added a Write Only pre_post_on property to the new Aux buss object of LwCH and FaCH in Quasar.
- Added a Write Only pre_post_faser property to the new Aux buss object of LwCH and FaCH in Quasar.
- Added a Write Only gain property to the new Aux buss object of LwCH and FaCH in Quasar.

Version 1.7.6.09 06/13/2020

- This version includes minor fixes over 1.7.6.08.
  - 1.7.6.08 involved major changes, please review the release notes and Beta Documentation for 1.7.6.08 and report any bugs you encounter.
- Changed the slot label interval on calendars to be hourly only to clean up the look of the calendar grid times.
- Enabled an event limit in the month view of calendars to clean up that view when large numbers of events are present.
- Fixed a bug where the new QSCONN object support added in 1.7.6.08 was not present on a newly discovered fader panel without a restart.
- Fixed an issue with the description data for FPSTAT Master property which should have more values than just 0 or 1 and added the descriptive definitions of those values to the drop downs.

Version 1.7.6.08 06/12/2020

- This version includes major changes in how PathfinderCore PRO is compiled and built in order to bring it more inline with other Telos Alliance products and Continuous Integration policies.
  - As this is a major change, please report any bugs you encounter.
  - Patch version is bumped to 6 as these build changes also exist in the release branch for any future release builds.
- Fixed bugs introduced by the new scheduling system that would not allow newly created scene items or timer events to work properly when being created with virtual, SapProperty, or virtual mixing destinations as the start or end property.
- Fixed an issue where the enabled property on new timers would default to blank rather than to True.
- Fixed performance issues with loading the timers list in the web ui with large numbers of timers.
- Fixed performance issues with displaying calendars with large numbers of events on a single calendar.

- Added a drop down list when editing scenes or timer start and end properties and selecting a virtual mixing destination.

  - The drop down allows you define whether the change will be directed at the AddSource or RemoveSource property.
  - Please review the documentation on this feature: Beta Documentation
- Switch scheduling calendars to display 15 minute time blocks instead of 30.
- Optimized the loading of time based events and memory slots during system startup.
- Added a Range and LoopAction property to Numeric Memory Slots.
  - Please review the documentation on this feature: Beta Documentation
- Modified the API code for Virtual Mixing destination's AddSource and RemoveSource properties to accept PathIos of sources in the virtual mixing routers in addition to the IO number.
- Added the QSCONN Lwrp object to Quasar fader panel devices.
  - This object allows for the ability to change the master module to which a fader module is attached.
  - Please review the documentation on this feature: Beta Documentation
- Added the FPSTAT lwrp object to Xnode devices.
  - This object allows for the ability to poll and monitor states such as power supply status, NIC up down, main board temperature, etc.
  - Please review the documentation on this feature: Beta Documentation

Version 1.7.5.07 04/22/2020

- From 1.6.5.00: Fixed an issue where an image assigned to an SaConsole button would not display and would disappear after saving the panel.
- From 1.6.5.00: Fixed an issue changing the name of a logic flow view (folder) would cause flows within that view and/or its sub views that have combiners in the flow not to function past the combiner.
  - Previously the workaround was to either reboot or rebuild the flow.
- Decorated the Path property of virtual base io objects so that it appears in the api tree of logic flows.
- Decorated the Ordinal position property of virtual base io objects so that it appears in the api tree of logic flows.
- This version adds an optional calendar based scheduling module which requires an additional license.
  - If you would like to help with beta testing this feature please contact us for a 90 day demo license.
  - Please review all documentation on this feature in the beta manual before updating: Beta Documentation

Version 1.7.4.06 04/06/2020

- This version adds Vx Control Protocol support.
  - Note: This feature should be considered beta.
  - Please review all documentation on this feature in the beta manual before updating: Beta Documentation
- From 1.6.4.00: Fixed an issue where the default friendlyname if no other was specified for an object was the object type without the id which led to duplicate entries in the logic flows tree.

- From 1.6.4.00: Fixed issues with image properties in the user panel designer not displaying a list of panel images as choices in the translation dialog.
- From 1.6.4.00: Added the overflow property to label controls which when used in conjunction with text-overflow can prevent the text from overflowing the label.
- From 1.6.4.00: Fixed an issue in the user panel designer where the image selection dialog appeared when pressing the bind button for background-image and img src properties when it should not.
- From 1.6.4.00: Fixed an issue in the user panel designer where the image selection dialog appeared when clicking the property header for background-image and img src properties when it should not.
  - These two issues made it difficult to get to the flow editing of those properties.
- From 1.6.4.00: Fixed an issue with image fields in the user panel designer property list where the shift-click option for manual editing would not work.
- From 1.6.4.00: Fixed issues with images switching from relative url to full url when saving a user panel design.

Version 1.7.3.05 03/21/2020

- Fixed a bug with the property selection UI with the new start and end properties in TimeEvents which was not switching the io selection list back to destinations after selecting a source value.
- Fixed some other UI property and value selection bugs with the new TimeEvents options.

Version 1.7.3.04 03/19/2020

- From 1.6.3.00: Fixed a bug introduced in 1.6.2.00 which only affects a small number of customers who have been provided with information about using auto insert limits in Axia Audio Routers to split Ios from different sites into different routers via the API.
  - 1.6.2.00 changed when backfeed sources where tested against the autoinsert limits which broke backfeed discovery when auto insert limits were used.
  - That should be fixed in this version.

> ⓘ Warning:
> From 1.6.3.00: If migrating from a patch version earlier than 1.6.2.xx or 1.7.2.xx review the notes for 1.6.2.00 and 1.7.2.04 regarding a manual sync.

Version 1.7.2.04 03/18/2020

- From 1.6.2.00: Fixed additional bugs related to discovering backfeeds introduced in 1.5.20.43.
  - New backfeeds were often not being discovered into the router or on clusters they might only be discovered on one of the systems or neither of them.
- From 1.6.2.00: Fixed an issue where newly discovered Ios (especially backfeed ios) might not make it into the io lookup table on the secondary server in a cluster.
  - This might cause certain parameters on those ios to not update until/unless the secondary server

took over.

- From 1.6.2.00: Fixed issues with the cluster synchronization of newly discovered backfeeds which was not working correctly in all situations.
- From 1.6.2.00: Fixed an issue where the secondary server in a cluster might not show the current owner changes for backfeeds until/unless it took over as primary.
- From 1.6.2.00: Fixed an issue with Gpio routing to iport destinations which was not allowing the route to take place.
  - The command was sending invalid name information in the SRCA property which other devices allowed but Iport did not.

> ⓘ Warnings:
> From 1.6.2.00:
> Because of the discovery issues above it is possible that clusters might have backfeed Ios on only one server or the other. Therefore, after updating, a manual sync from primary to backup is recommended. Review the manual for details on executing a manual sync. To sync from Primary to Secondary, open the clustering tab on the secondary server and click the manual sync button. This will request a special clustering backup from primary and restore that backup into the secondary PathfinderCore PRO which will subsequently trigger a reboot of the secondary PathfinderCore PRO.
>
> From 1.7.1.04:
> This build also contains the timer changes documented in 1.7.1.04. Please review the release notes below and Beta Documentation for 1.7.1.04 as these timer changes are beta changes.

Version 1.7.1.04 03/12/2020

- This version adds end times to DateTime and DayOfWeek timers so that a time range may be expressed.
- This version adds start and end properties to DateTime and DayOfWeek timers that may directly be executed at the start and end time of the range without the need for a logic flow.
- This version adds a cleanup option that can be used to define when or if datetime timers should be deleted after they execute.
  - Note: The feature changes listed above should be considered beta.
  - Please review all documentation on these features in the beta manual before updating: Beta Documentation
- Added a ClearElapsed write only property to DateTime and DayOfWeek Timers to be used in certain testing procedures.
- Modified DateTime and DayOfWeek timer editing to display with the programmed time rather than a conversion to the browser's time in the case where the browser is in a different timezone.
- Modified DateTime and DayOfWeek timer editing to default to the PathfinderCore PRO's utc offset for clock based times rather than the browser's offset.
- Added comments to clock and fixed time radio button options in the time editor to show more specifically that clock time will adjust with DST.

- Added UTC field for clock based timers which is disabled for changing but shows the UTC offset for the event from PathfinderCore PRO.

Version 1.7.1.03 02/28/2020

- Fixes a bug with restApi devices where the post command would carry the operator and url in the content field in addition to the content instead of just the content.
- Adds escape character possibilities to email message bodies for things like carriage returns and linefeeds.
    - See the documentation for generic emulator ToSend escape sequences for details as both use the same sequences.
- Adds code to allow the swap value on Html5 Panel disabled properties.
- Fixes the disabled property for Html5 panel knobs and gauges.
- Adds a --display-channels property to Html5 meters under the led section with values of stereo, left, or right in order to allow support for mono meters.

Version 1.7.1.02 02/27/2020

- From 1.6.1.00: Fixed bugs introduced in 1.5.20.43 related to how backfeeds are handled differently with iQx/Qor.
    - The changes required to handle differences in how iQx/Qor handle backfeeds caused backfeeds to sometimes be created without certain information in the database.
    - In some situations this could cause the backfeed to display as a normal source rather than a backfeed source.
    - The backfeed rtp stream address was also appearing in some cases in the device's normal port causing multiple source targets for a route due to the duplicated multicast address.
    - This version fixes the missing data in the database on ingest of the io so that it properly gets created as a backfeed object.
    - This version fixes the missing data when new backfeeds are discovered on an iQx/Qor and are added to the database.
    - This version sets the device port to no address, disabled, and a default name in the source table when being used as an autobackfeed style source.
- From 1.6.1.00: Updated copyright footer on web pages.

Version 1.7.0.02 01/02/2020

- Added a name field to the rest api device from 1.7.0.01 for a better user experience.
- Added an enabled property to virtual router source and destination base points.
    - Please review the Beta Documentation for details.
- Added a PushRouteOnBaseEnable property to virtual routers that works alongside the enable property of base points.
-

- Added a Virtual Mixing router for dynamically adding and removing sources from a set of mixable destinations.
    - Please review the Beta Documentation for details.

Version 1.7.0.01 12/11/2019

- This is the first beta version with new beta features since the 1.6 release.
- Added a rest api device.
    - Please review the Beta Documentation for details.
- Added a CalculateTime property to timestamp memory slots to allow conversion from other time formats to the pattern specified.
    - Please review the Beta Documentation for details.

1.6.xx.xx

Version 1.6.13.00 10/30/2020

- Fixed an issue that would prevent user panels from working if the security settings denied access to the main Devices object since their ability to work relies on being able to check the service load state and the ping property.
    - Get operator only is granted access to that object's few properties for all valid user logins so that these properties may be queried.
- Added some error trapping to catch a rare situation where an ip client in the process of being disposed and sending at the same time might cause a crash.
- Removed an error message from the logs regarding the failure to remove meters from a lookup collection that is not actually an error.
- Clarified an info log message to make it less error-like regarding Qor Lwcpss message handling for amix.
    - "amix" parameter support currently requires the beta version.
- Fixed an issue with the userpanels xy grid and locks with virtual routers.
- Added code to decode ampersands properly in uerspanel properties that use url as their property type in the property grid.
- Fixed an issue where the user panels list web page still displays notes and warnings about using legacy panel designer.
    - These notes should have been removed in version 1.4.

Version 1.6.12.00 09/25/2020

- Fixed an issue with user panel countdown clocks where the change of countdownlength was not being raised to sapV2 and therefore would not update data in the countdown clock without refreshing the panel web page.

- Fixed an issue with user panel countdown clocks where reloading the page or changing the countdown time on a page where the timer had elapsed but not been reset would show the count down duration instead of 0.
- Added code for user panel countdown timers in countdown mode to display 0 if countdownlength minus elapsed milliseconds is less than zero.
- In the User Panel Designer, added an alert and block to the countdown timer --countdownlength, stopmode, and resetmode properties to prevent them from being unbound as binding must be left on for them to function properly.

Version 1.6.11.00 09/11/2020

- Fixed an issue with UserPanels where it is possible to start receiving state messages before the bindings file is loaded leaving those states with no place to go.
  - This could cause some panel objects to have an incorrect starting state when the panel gets loaded.
  - To address this problem, this version does not connect the web socket until the bindings file has been loaded.

Version 1.6.10.00 09/10/2020

- Fixed an issue where phase 1 of the Lwcp state initialization was sending GET SUPV STATE whereas Quasar would only accept GET supv state.
  - The result was that with Quasar we would often not get to phase 2 and therefore not get initial states for a variety of parameters until those parameters changed.
  - For example, a user panel label bound to the show profile name would likely be blank on system startup until the show profile was changed or we happened to poll that parameter.

Version 1.6.9.00 08/25/2020

- Fixed an issue where generic emulators set to TcpClient mode were not raising connected change events.
- Fixed Connected property for Device Emulator listeners so that it will be true if at least one client is still connected and False otherwise.
- Fixed an issue where subscribing to a GenericEmulator would yield a ToSend event with an empty string for each ToSend attempt.
  - ToSend is a write only property and should not raise events.
- Fixed Generic Emulator ConnectedCount not displaying the correct count and not changing when tcpclient is selected as the connection type.

Version 1.6.8.00 08/15/2020

- Fixed an bug introduced in version 1.6.6.00 where user logs were no longer accessible from the web ui and were instead showing a forbidden access error message.

- Fixed a bug where the disabled property on Html5 user panel buttons was not being respected by touch actions on touch screens.
- Added a blankvalue property to the saselector drop down control in html5 panels.
    - The blankvalue property allows you to define a value to use if the value returned is blank.
    - For example, the manual example refers to using the currentsource property of a virtual destination and filling the list with virtual sources.
    - The None source has a value of "0" but often a cleared destination route is returned as a blank field instead of 0.
    - Using 0 in the blankvalue property will force the empty currentsource to be interpreted as 0 which matches the none source causing it to be displayed.
    - Without this property being configured, a destination with nothing returned as the current source will simply not make a change to the current item in the list causing it to retain a route where none actually exists.
    - This has been implemented as a separate field because each list might or might not require different values to be used or not used for a blank field.

Version 1.6.7.00 07/17/2020

- Please review the release notes for the non-public 1.6.6.00 version as those changes exist in this version as well.
- Change the factory default file to alter the timezone localtime symlink rather than overwrite the file.
    - In some cases this was overwriting the underlying symlinked file causing a specific timezone to show Eastern offset rather than the correct offset for the timezone after a factory default.
    - Bank updating would fix the problem as well until the next factory default.
- Added a question to the r2 production install script to ask if it is being installed on the mbx platform.
    - If the answer is yes, the crystal fontz display application is disabled and we use the new default ip addresses in the ip address questions.

Version 1.6.6.00 06/12/2020

- This version includes major changes in how PathfinderCore PRO is compiled and built in order to bring it more inline with other Telos Alliance products and Continuous Integration policies.
    - As this is a major change, please report any bugs you encounter.
- This version was not publicly released.

Version 1.6.5.00 04/22/2020

- Fixed an issue where an image assigned to an SaConsole button would not display and would disappear after saving the panel.
- Fixed an issue changing the name of a logic flow view (folder) would cause flows within that view and/or its sub views that have combiners in the flow not to function past the combiner.
    -

Previously the workaround was to either reboot or rebuild the flow.

Version 1.6.4.00 04/06/2020

- Fixed an issue where the default friendlyname if no other was specified for an object was the object type without the id which led to duplicate entries in the logic flows tree.
- Fixed issues with image properties in the user panel designer not displaying a list of panel images as choices in the translation dialog.
- Added the overflow property to label controls which when used in conjunction with text-overflow can prevent the text from overflowing the label.
- Fixed an issue in the user panel designer where the image selection dialog appeared when pressing the bind button for background-image and img src properties when it should not.
- Fixed an issue in the user panel designer where the image selection dialog appeared when clicking the property header for background-image and img src properties when it should not.
    - These two issues made it difficult to get to the flow editing of those properties.
- Fixed an issue with image fields in the user panel designer property list where the shift-click option for manual editing would not work.
- Fixed issues with images switching from relative url to full url when saving a user panel design.

Version 1.6.3.00 03/19/2020

- Fixed a bug introduced in 1.6.2.00 which only affects a small number of customers who have been provided with information about using auto insert limits in Axia Audio Routers to split Ios from different sites into different routers.
    - 1.6.2.00 changed when backfeed sources where tested against the autoinsert limits which broke backfeed discovery when auto insert limits were used.
    - That should be fixed in this version.

> ⓘ Warning:
> If migrating from a patch version earlier than 1.6.2.xx or 1.7.2.xx review the notes for 1.6.2.00 and 1.7.2.04 regarding a manual sync.

Version 1.6.2.00 03/18/2020

- Fixed additional bugs related to discovering backfeeds introduced in 1.5.20.43.
    - New backfeeds were often not being discovered into the router or on clusters they might only be discovered on one of the systems or neither of them.
- Fixed an issue where newly discovered Ios (especially backfeed ios) might not make it into the io lookup table on the secondary server in a cluster.
    - This might cause certain parameters on those ios to not update until/unless the secondary server took over.
-

Fixed issues with the cluster synchronization of newly discovered backfeeds which was not working correctly in all situations.

- Fixed an issue where the secondary server in a cluster might not show the current owner changes for backfeeds until/unless it took over as primary.
- Fixed an issue with Gpio routing to iport destinations which was not allowing the route to take place.
  - The command was sending invalid name information in the SRCA property which other devices allowed but Iport did not.

> Warning:
> Because of the discovery issues above it is possible that clusters might have backfeed Ios on only one server or the other. After updating, a manual sync from primary to backup is recommended. Review the manual for details on executing a manual sync. To sync from Primary to Secondary, open the clustering tab on the secondary server and click the manual sync button. This will request a special clustering backup from primary and restore that backup into the secondary PathfinderCore PRO which will subsequently trigger a reboot of the secondary PathfinderCore PRO.

Version 1.6.1.00 02/27/2020

- Fixed bugs introduced in 1.5.20.43 related to how backfeeds are handled differently with iQx/Qor.
  - The changes required to handle differences in how iQx/Qor handle backfeeds caused backfeeds to sometimes be created without certain information in the database.
  - In some situations this could cause the backfeed to display as a normal source rather than a backfeed source.
  - The backfeed rtp stream address was also appearing in some cases in the device's normal port causing multiple source targets for a route due to the duplicated multicast address.
  - This version fixes the missing data in the database on ingest of the io so that it properly gets created as a backfeed object.
  - This version fixes the missing data when new backfeeds are discovered on an iQx/Qor and are added to the database.
  - This version sets the device port to no address, disabled, and a default name in the source table when being used as an autobackfeed style source.
- Updated copyright footer on web pages.

Version 1.6.0.00 12/05/2019

- 1.6.0.00 release build for fanless engine, r2, and vm platforms.
- This build is the same as 1.5.20.43 but moved to the master release branch and versioned as 1.6.0.00.
- All future bugs related to features already in this version will become patches to 1.6 versioned as 1.6.x.00 where x is the patch number.
- All new features will be created in the next 1.7 beta branch.
- See Version Numbering for details.

- Updated the manual link to version 1.6 of the manual.

1.5.xx.xx

Version 1.5.20.43 11/26/2019

> ⓘ Important Note:
> This version is still feature frozen in the 1.5 beta branch in preparation for a 1.6 release.
>
> Bugs will continue to be fixed in the 1.5 branch during the remainder of the beta testing but any new features will go into a 1.7 beta branch.

- Fixed an issue with the current owner property of backfeeds when used with iQx and Qor.
  - Backfeed messaging in iQx and Qor differs in how it presents backfeed sourcing compared with Fusion.
  - This required adding some custom code to handle backfeed current owner tracking when Qor and iQx are involved.

Version 1.5.20.42 11/15/2019

> ⓘ Important Note:
> This version is still feature frozen in the 1.5 beta branch in preparation for a 1.6 release.
>
> Bugs will continue to be fixed in the 1.5 branch during the remainder of the beta testing but any new features will go into a 1.7 beta branch.

> ⓘ Warning:
> Critical bug fix with Logic Flow's ability to send Emails introduced in 1.5.20.33.
>
> As a part of the licensing changes in 1.5.20.33 we broke the ability for stored email message objects to be accessed directly by their object path in the API. In most cases this also broke the ability for logic flows to access the message object to send email.
>
> Any users of version 1.5.20.33 (Oct 4, 2019) through 1.5.20.41 (Nov 11, 2019) should update to 1.5.20.42 as soon as possible to regain email messaging capability.

Version 1.5.20.41 11/13/2019

> ⓘ Important Note:
> This version is still feature frozen in the 1.5 beta branch in preparation for a 1.6 release.

> Bugs will continue to be fixed in the 1.5 branch during the remainder of the beta testing but any new features will go into a 1.7 beta branch.

- Fixed an issue where the Logic Flows simple tree was showing a ToSend property on both the Generic Emulator and on the emulator's tcp/udp client.
  - The latter should have been API only as you should use ToSend on the emulator itself.
- Fixed an issue where the friendly name property was incorrect on Device Emulators causing them to sometimes lose their name representation in the logic flow tree.
- Did some minor optimization in the name lookup for logic blocks with io names for large routing systems.
  - Originally a name lookup queried the entire Router Manager for the pathio to name lookup.
  - This version more intelligently queries the information accessing the object path hash table.
  - This improves performance when switching between flow folders where start and end points use audio and gpio route points.

Version 1.5.20.40 11/11/2019

> ⓘ Important Note:
> This version is still feature frozen in the 1.5 beta branch in preparation for a 1.6 release.
>
> Bugs will continue to be fixed in the 1.5 branch during the remainder of the beta testing but any new features will go into a 1.7 beta branch.

- Fixed a bug with inner panels relying on panelmemoryslots as described in 1.5.8.07 that was causing indicators that should have been displaying the value from the internal panel memory slot to display blank instead.
  - This was broken in 1.5.9.09.
  - There was an assumption that the first message return with a done set system item was a response from the initial state request of the panel and that was not the case.
  - Now it looks for done set with a specific transaction id to make sure the initial get state is complete before applying the internal panelmemory states.
- Fixed a problem with the xy matrix where it would require a browser refresh in order to pick up newly added or removed IOs.
- Fixed an issue where the drop down selectors would not work in the user panel property editor on Firefox.
- Fixed a issue in Firefox where the bind button height was inordinately short.
- Fixed an issue with the audio alarm editor where clicking on the ellipsis for a source or destination that is defined in the audio alarm but no longer exists as an io in the router would throw an error silently and fail to open the io selection dialog.
- Fixed an issue with the lines between flow items in the user panel editor not displaying properly in edge and Firefox.
- Added Firefox and ms user-select none options to other css places where we had selected user-

select:none in the css.

Version 1.5.20.39 11/08/2019

> ⓘ Important Note:
> This version is still feature frozen in the 1.5 beta branch in preparation for a 1.6 release.
>
> Bugs will continue to be fixed in the 1.5 branch during the remainder of the beta testing but any new features will go into a 1.7 beta branch.

- Fixed an issue with executing user panels in Firefox where dragging a fader or knob would cause items to be selected.
- Fixed a bug with gradient meters not displaying in Firefox.
- Fixed some button sizing issues when simple mode was used on console buttons in Microsoft Edge.
- Fixed some text centering issues with buttons when used in Microsoft Edge.

> ⓘ Important Note:
> Despite the above fixed items, Chrome is still the recommended browser for the best user experience.

- Added a 1500 ms delay to the UI for router deletion to make sure the UI has a chance to send the command to the equipment before reloading the routers list.
- Fixed a bug where virtual routers with base points referencing other virtual routers would cause a problem in the base point display which if resubmitted during editing could cause additional IOs in funky ways.
  - Essentially the UI was requesting and presenting an infinite depth list rather than just the base IOs under the current IO.
  - This has been fixed.

Version 1.5.20.38 11/04/2019

> ⓘ Important Note:
> This version is still feature frozen in the 1.5 beta branch in preparation for a 1.6 release.
>
> Bugs will continue to be fixed in the 1.5 branch during the remainder of the beta testing but any new features will go into a 1.7 beta branch.

- Fixed a bug with auto resizing when the panel is running in the core pro browser with menu system rather than a pop up or standalone panel.
  - This bug was introduced approximately 2 months ago when we changed the name of one of the parent divs.

Version 1.5.20.37 11/01/2019

> ⓘ Important Note:
> This version is still feature frozen in the 1.5 beta branch in preparation for a 1.6 release.
>
> Bugs will continue to be fixed in the 1.5 branch during the remainder of the beta testing but any new features will go into a 1.7 beta branch.

- This build should theoretically be identical functionally to 1.5.20.36.

  - It is the result of merging the beta branch changes into the master branch in preparation for a new release.

Version 1.5.20.36 11/01/2019

> ⓘ Important Note:
> This version is still feature frozen in the 1.5 beta branch in preparation for a 1.6 release.
>
> Bugs will continue to be fixed in the 1.5 branch during the remainder of the beta testing but any new features will go into a 1.7 beta branch.

- Fixed a bug with set time from pc and hardware clock management.
  - Pathfinder Core PRO should be set to sync to an NTP clock for accurate time services.
  - This issue is not a problem for PathfinderCore PRO instances which are properly set to sync to NTP.
  - In the case where NTP is not configured properly or is unreachable, setting the time using the set time from pc would cause the clock to look correct until a reboot.
  - After a reboot though the clock might be off by the time zone offset amount.
  - Ntp if configured would correct this automatically.
  - The problem was caused by not properly informing the hardware clock that the time being set was a local time and should be adjusted accordingly to utc so that the hardware clock is utc and the os clock is then derived from that based on the utc offset.

Version 1.5.20.35 11/01/2019

> ⓘ Important Note:
> This version is still feature frozen in the 1.5 beta branch in preparation for a 1.6 release.
>
> Bugs will continue to be fixed in the 1.5 branch during the remainder of the beta testing but any new features will go into a 1.7 beta branch.

- Fixed a bug where with Logic flow folder renaming.
  -

  - While you could not create a logic flow folder with a forward or back slash in the name, after creating a folder using a legitimate name you could rename it to one with a bad name.
  - The result could be funky behavior where certain flows would be nested incorrectly after a reboot and might not work properly.
  - Slashes are used in os file system and are therefore illegal in the name.
  - This version tightens the validation to prevent folders from being created or renamed using these illegal characters.

Version 1.5.20.34 10/24/2019

> ⓘ Important Note:
> This version is still feature frozen in the 1.5 beta branch in preparation for a 1.6 release.
>
> Bugs will continue to be fixed in the 1.5 branch during the remainder of the beta testing but any new features will go into a 1.7 beta branch.

- Fixed a potential memory leak with continuing device reconnection attempts.
  - Also improved socket cleanup and disposal code.
- Made the metering device reuse its client socket rather than destroying and creating a new one.
- Fixed an issue where hardware mapped buttons would not work with the mouse up event properly.
- Fixed a bug where enabled state changes were not hooking to licensing properly sometimes requiring a restart to update license usage counts.
- User panel element ids must start with a letter to be fully browser compatible.
  - If you enter an element id that starts with a number, that first number will be converted to the equivalent numerical word followed by an underscore when the id is stored.
  - Previously trying to enter an id that started with a numeral would cause the item to jump to the top corner of the panel and become inaccessible.
- Removed the links to the legacy client applications as they are deprecated.
- Updated the Date/Time stamp for manual links in the systemstatus page.
- Updated email web page to Say SSL/TLS instead of just SSL as it figures out the request to upgrade to TLS so this label was confusing.
- Fixed a bug when deleting a pass thru combiner that left a residual half complete flow that would also have to be deleted.
- Reworked licensing names to better match marketing materials.

Version 1.5.20.33 10/04/2019

> ⓘ Important Note:
> This version represents a feature freeze for the 1.5 beta branch in preparation for a 1.6 release.
>
> Bugs will continue to be fixed in the 1.5 branch during the remainder of the beta testing but any new features will go into a 1.7 beta branch.

- Updated the logic flows simple tree to present friendly names for Quasar buttons.
- Fixed a display bug with time on the system time web page when utc offset was positive rather than negative.
- Added a localtime property to the System#0.Time#0 object.
- Added a local time field to the system time web page and made the original time printout display using iso format.
- In Virtual Router IO editing, added the paging and entry count selection fields to the BaseIo package table for use with larger base ios.

- Fixed a bug where if you delete a device that had hardware mapped buttons, you then would not be able to change the hardware mapping of the button due to an error in the web page when trying to find the missing device path.
- Added a cell-autoscale property to the saroutingmatrix user panel component which is true by default on panel route matrixes and false in the routes table version.
  - When false the cells stay roughly the same height and width rather than expand to fix the matrix.
- Fixed a bug in saroutingmatrix that was causing the listed rows to run over the end of the component and not allow scrolling to the last row in situations where no search was in place and the horizontal scroll bar was hidden.
- Modified the saroutingmatrix styles a bit for better header display.
- Changed the code in saroutingmatrix such that filtering (searching) with small numbers of returns will change cell size if autoscaling is on but not if it is off.
- Fixed a bug where a blank "from" value in a logic flow translation conversion item would cause the UI to hang in an endless loop.
- Added Startup state to the memory slot grid view.
- Changed the Quasar monitor section friendly name to be Master_TS_Monitor as that is more intuitive.
- Switched off the use of begin/end on metering commands as the devices were not respecting that optimization anyway and the iports/engines were actually throwing error messages.
- Added Lwrp CMD support into PCP gpio node.
- Added the command (Lwrp CMD) item to gpi and gpo and exposed it in the API logic flow tree.
  - It is important to note that while CMD on GPO ports works across all devices, CMD on GPI ports only works on software nodes such as the windows driver and the Pathfinder Core PRO gpio node.
- Made the Command item of gpios clear after 100ms inside the Pathfinder Core PRO data structures and SapV2.
  - Note that if your logic requires this clear state to not hit the logic endpoints, it can be skipped by making a conversion item where the from is bank and the to is <NoChange>
- Fixed a bug where restricted by license translators were not displaying with the dark color as they should.
- Added a note to the hover balloon for translators that have been restricted by license.
- Changed the license counting such that you no longer need to specify whether an extension license applies to logic or audio and all licenses including the base licenses function as a shared pool.

    - Starting with this version the total license count is a pool that is shared between audio and logic points.
    - Therefore the base license which used to be 500 audio and 500 logic is now just a shared 1000 point pool used for audio and logic.
- Updated the licensing display on the system page and license editing page to reflect the license counts and pool utilization.
- Internal Task 572
- Internal Task 573
- Internal Task 574

Version 1.5.20.32 9/23/2019

- Added an option to silence and clipping alarms where you can select Left, Right, All, or Any in order to specify which channels need to be silent or clipping in order to trip the alarm.
    - By default existing alarms will function in the Any state to match previous functionality.
- Fixed a bug where editing and changing a silence or clipping alarm could leave it in an unknown state until the next message.
    - Now if the subscription state already exists it will adopt the correct change which may trigger a count.
- Set Quasar monitor button count to 32.
- Added an AllInternalServicesLoaded property for use by the web pages in case a device emulator has problems starting.
    - Previously some web pages would recycle their websocket connection until all services were loaded.
    - This change excludes Device Emulators from that restriction.
- Added some 30 second crosschecking to discover any device emulators that did not hook into the main application properly during startup due to a bound cpu and need to be relaunched.
- Added a ClearRoute value to SapPropertyRouter available in the router translation button.
    - When a property route is cleared, by default the value of the destination property is not changed.
    - This field can be used to specify a value to be sent to destination property when the source assigned to it in the Property Router is cleared.
    - Note that currently this field requires knowledge of the correct value to send. In the future we may add additional helpers for this field.
    - The default value is <NoChange> in order to match functionality from the previous build.
- Updated some Quasar monitor section button naming with more friendly names in the logic flow tree.
- Fixed an issue in User Panel editing where internal sub panels using the IFrame might re-grab the focus while trying to edit the src link of the IFrame.
- Added a function to strip out Kaspersky injected header code before saving a panel if it exists.
    - Some versions of Kaspersky have been known to inject code into every web page that is accessed.
    - In previous versions, this injected code was sometimes being saved out to the panel definition.
- Made the switch of fader and meter orientation more intuitive by swapping the height and width at the

same time.

- Adding some titling in the SapProperty Router translation dialog for hover pop up explanations.

Version 1.5.20.31 9/16/2019

- Added a SapProperty Router for using any properties PathfinderCore PRO knows about as sources and destinations in a router.
  - Please review the Beta Manual for details.
- Added an advanced field in the logic flow translation dialog for manually editing the selected conversion list item if necessary.
  - This field allows you to bypass the helper fields such as the color selector and directly enter data into the conversion.
  - Please review the Beta Manual for details.
- Added Regular Expression capabilities to Generic Emulation watchers and translator from-conversion list items.
  - Also added a TriggeredValue property.
  - Please review the Beta Manual for details.
- Added a NoChange option to logic flow translation to-conversion list items.
  - Please review the Beta Manual for details.
- Fixed a bug with hardware mapped buttons that make use of the disabled option.
- Fixed a bug with the ToSend option of Generic Emulators that might incorrectly strip off carriage returns and line feeds at the end or beginning of a message that should be passed through.
- Fixed a bug that has existed since the ui overhaul in 1.5.19.25 where the advanced link in logic flow translation was not working.
- Fixed a bug that has existed since the ui overhaul in 1.5.19.25 where the clear route would only work after selecting a source in the take list.
- Altered the take and cancel button style in XYMatrix slightly to prevent overlap with low io counts.
- Fixed a bug in the XYMatrix where after doing a search and resize of the xy grid, the vertical scroll bar and the preset rotation controls would overlap.
- Fixed a delay when opening a router with no ios in the web ui.
- Reworked the advanced options translation display code to be more intuitive across user panels and SapProperty router.

Version 1.5.20.30 9/03/2019

- Fixed some bugs in scenes related to cluster synchronization in the 1.5.20.29 build.
  - These primarily caused issues in the case where a scene design was modified while one of the nodes was offline and then re-synced.
- Fixed a bug with scenes in 1.5.20.29 where the correct version stamp was not being written into the storage file.

Version 1.5.20.29 9/02/2019

- Completed an editor webpage for creating and editing scenes.
  - Please review the Beta Manual for details.
- Initial feedback on the xy matrix from 1.5.20.28 determined that the cell resizing when using search filters was distracting.
  - This version maintains cell size during search filtering.
  - This also addresses an issue where when filtering, the display list could extend beyond the bottom of the component without the scroll bar appearing.
- Quasar button support.
- Added fader user buttons and monitor module user buttons to the button list available when hardware mapping in user panels.
- Improved some performance issues when initially entering the panel designer related to populating the data selection tables.

Version 1.5.20.28 8/20/2019

- Added additional styles (simple and touchbar) to fader and consolefader user panel objects.
- Added an orientation property for horizontal and vertical display of fader user panel objects.
- Added a number of additional styling properties to the fader and consolefader user panel objects.
- Added a router xy matrix both as a user panel object and on the route details web page for routers.
- Please review the Beta Manual for details on these new features.
- Added some minor optimizations to io state changes in the router details tables.
- Added hover titles for properties in the user panel property list in order to better display long property names.

> ⓘ Important Note:
> The changes above and detailed in the Beta Manual required a number of changes to the user panel default skin css file. If the components discussed do not appear correctly in the panel designer web page, your executing panel, or in the router-details web page, try forcing the browser to refresh its cache.
>
> In Chrome this can be accomplished by holding Ctrl and Shift while clicking the reload this page icon while on the web page having difficulties.
>
> You should only have to do this once on each of these three pages and then the browser will fetch and cache the revised stylesheet file.

Version 1.5.20.27 8/1/2019

- From: 1.4.20.00: Added some changes to the metering device object to correct the possibility of a crash during a lost connection disabling of the timers.
  -

  - We have seen two sites that have exhibited crashes with a logged stack trace on the same line of code.
  - This line of code deals with the disabling/enabling of certain timers during a loss of connection to the equipment of a metering device.
  - While we have not been able reproduce in the lab we believe the changes in this version should correct the problem.
- Made some minor performance improvements in fader manipulation.

Version 1.5.19.26 7/22/2019

- Please review the release notes and beta documentation for 1.5.19.25 as well.
- Please review the Beta Manual for details on all of the features listed below.
- Added support for the LwcpSs control protocol within Iqx and Qor based consoles.
  - This adds many new control points in logic flows for Iq/Qor based consoles.
  - This option may be disabled via the configuration advanced options page by adding the following line:
  - SET Devices#0 LwcpSs=False
- Added a monitor section Html5 userpanel component for use with Iqx and Qor based consoles.
  - This option may be disabled via the configuration advanced options page by adding the following line:
  - SET Devices#0 QorMonitor=False
- Added a border gradient property with values of simple and complex to saconsole buttons.
- Added an autoscale property to html5 user panels.
  - Known Issues: there is a known issue with the height_and_width value and consolefaders.
  - This issue is discussed in the beta documentations as well.

Version 1.5.19.25 7/17/2019

- This build reflects a major refactoring of the web ui underlying code.
  - Also includes updates to the web ui data tables to a newer api and deferred rendering option.
  - Almost all of the files in the web ui were manipulated in some fashion during this rework.
  - Please report any issues you encounter so that we may fix them as soon as possible.
  - Please review the Beta Manual for details on these changes.
- Added a new sort icon at the beginning of the routes table in routers that sorts based on the destination path io for a more natural ordering.
- Removed some unnecessary links to jquery in panel definition files.
- Added code to clean up unnecessary jquery links when an existing panel is saved.
- Fixed a bug with the Pathfinder gpio and multicast which broke receive multicast communications in situations where both livewire and office nics were active and the gateway was on the office nic.
- Fixed an issue with the stop button for livewire endpoint discovery not updating after discovery was

stopped.

- Fixed a bug where save was not available when editing an existing user that was not an administrator.
  - This bug was introduced in 1.5.10.10.
  - The problem occurred because we moved to a progressive tree where not all data is loaded up front but rather it is loaded as the branches were being expanded.
  - Unlocking the save required a full load of the data which would not occur.
  - In this version we still do not fully load but we do a cross check on save and assume that originally selected settings that do not exist in the current instance of the tree were never expanded and changed and should therefore retain the value from the original setting.

Version 1.5.19.24 6/13/2019

- Fixed a bug in a custom meter handler when unassigned meters are on the panel.
- Fixed a bug where the property dialog in user panels was showing endpoint options as opposed to start point options in some situations.
- Added a timestamp memory slot type.
  - Please review the Beta Manual for details on this feature.
- Fixed a bug where passing false to the Zip One drop property would also drop the call when only True should cause a drop.
- Cleaned up some unused files in the Web UI folders.
- Clean build on OS.

Version 1.5.19.23 6/10/2019

- The changes in this version relate to html5 user panel components.
  - Please review the Beta Manual for details on these features.
- Updated the Fader control to be able to manipulate numeric and non-numeric properties in addition to the normal fader mapping.
- Added a rotary knob control.
- Updated the user panel analog and digital countdown timers with additional properties for stop, reset, pause, elapsed complete, and count up functionality.
- Faders and knobs now will display their value when hovering with the mouse.
- Faders and knobs will now work with the mouse wheel.
- Added a drop down component.
  - Configuration is complex at this point in time and currently should be for advanced users only.
  - More intuitive user interfaces for configuration will be coming in the future.
  - Please review the beta manual documentation for details.
- Added source profile selection to console faders.
  - This feature may also be disabled using the allowsourceprofilechange property.
- Exposed the transform css property to user panels.

- Holding the shift key while clicking into the property fields for user panel component will now allow direct text editing bypassing the normal helper dialogs.
- Qor LwcpSs support - hidden for internal testing only at this point in time. Will be released in a later build. (552 notes)
- Qor Monitor section panel component support - hidden for internal testing only at this point in time. Will be released in a later build. (553 notes)

Version 1.5.19.22 6/09/2019

- Added a write only append property to memory slots that allows you to append data into a memory slot as the endpoint of a logic flow.

Version 1.5.19.21 5/17/2019

- From 1.4.19.00: Included the VMware paravirtualized scsi driver in the vm and vm installer images.
  - This version only makes a change in the vm version.
  - The driver has been included for testing with VMware sites where ISCSI is being used for the host storage.

Version 1.5.18.21 5/16/2019

- Added support for Zip One Control including phonebook dialing and management.
  - Please review the Beta Manual for details on this feature.
- Added support for Iport specific Lwrp commands.
  - Please review the Beta Manual for details on this feature.
- Both of these features should be considered beta features.
- Please report any issues you experience and be prepared to roll back to a previous version/bank if you encounter any difficulties.

Version 1.5.18.20 5/13/2019

- From 1.4.18.00: Reworked day of week timer threading to prevent day of week timers from getting stuck in an elapsed state when they are supposed to flip back to not elapsed 10 seconds after execution.
  - Added an hourly safety catch to catch any that are in the wrong state.
  - Improved performance of timer lookups and moved some of the dayofweek elapsed resetting into each day of week object.
- From 1.4.18.00: Fixed a bug in object translators where if the input was a specific property rather than an object but the output was an object, the system would process other properties on the input side than just the one selected.
- From 1.4.18.00: Cleaned up some css around meters.
- From 1.4.18.00: Fixed a problem with html5 gradient meter off colors shifting the transition points if they

were not all the same color.

- From 1.4.18.00: Fixed a bug that was causing the schedule column for DayOfWeek events to switch to a number rather than the correct value until the web page was refreshed.

Version 1.5.17.20 4/30/2019

- Please review the notes on 1.5.16.18 and 1.6.16.19 regarding the gpio that now exists in the beta branch.
- From 1.4.17.00: Fixed a bug with user log rotation that was causing rotated files to have large blocks of NUL characters at the beginning in certain situations.
- From 1.4.17.00: Fixed an issue where activate scene did not submit the scene messages to the system in the order specified.
  - It is important to understand that even though the scene will now submit the change messages in order that does not guarantee they will be completed in order.
  - Each Scene item is sent to the system without waiting for completion.
- From 1.4.17.00: Added code to the silence alarms to more quickly detect device failures.
  - Currently the default Linux settings will only raise an error on tcp send failures after approximately 15 minutes of failed retry attempts.
  - This is much longer than the similar windows defaults.
  - Device management relies on application level ping messages to detect failures quicker but this was not implemented in silence alarm device connections.
  - In this version silence alarm device connections will also drop the tcp connection and will start counting down the failure after approximately 45 seconds of failed ping (VER) messages.
- From 1.4.17.00: Switched the SRC Rtpp property to be read/write.
- From 1.4.17.00: Fixed a bug where opening an object translator created with an API property while the Simple tree is selected and then selecting the item in the translation convert list might leave the property name field blank which after committing would wipe out that api property name in the conversion.
- From 1.4.17.00: Disallowed most punctuation in panel names.
  - This fixes some panel incorrect functionality when certain characters such as the slash are used in the panel or page name.
- From 1.4.17.00: Moved the meters and scenes to startup prior to logic flows in the service startup list.
- From 1.4.17.00: Added some thread locking around device online/offline state in router IO DeviceOnline property setting to prevent a potential race condition where offline to online messages in quick succession especially during initial load could theoretically interfere with each other.
- From 1.4.17.00: Fixed an issue with gpi dynamic additions not getting an index in the device io collection.
  - This can cause the ios not to show online/offline correctly.
- Added the ability to insert a pause as a scene item for situations where a pause in the scene execution is necessary.
  - This feature can only be used via the API at this point in time until we complete the Scene editor UI.
  - When initing the scene item use a changemessage property of:
    nop . Pause=5000

  - Where 5000 is the pause timeout in milliseconds.
  - Example: init Scenes#0.Scene#Off.SceneItem order=4, changeMessage="nop . Pause=5000"
- Added an offset property to Html5 faders which can be used to offset the fader gain range similar to the custom property in the fusion and Qor consoles.
  - Since the offset property in the console is not exposed to Pathfinder for reading we cannot detect the console setting, so this parameter must be set for each fader in the Html5 panel designer if you want it to be anything other than the default 0 offset.
  - For example setting the offset value to -10 will cause the top of the fader to be 0 rather than +10.
  - You can also use the optimum property to change what value the fader double click will move to.

Version 1.5.16.19 4/24/2019

- Please review the notes on 1.5.16.18 as this version builds upon the internal gpio node added in that version.
- This version adds cluster synchronization to the internal Gpio node in Pathfinder Core PRO.
  - Please review the Beta Manual for details on the nuances related to synchronization of this feature.
  - The internal Gpio node is still a new feature to PathfinderCore PRO so report any issues you encounter as you start using it.
- Fixed a bug where loopback (127.0.0.1) snake gpio routing that was in external gpio devices was either not matched or showing up as pcpgpio sources in the gpio router.
- Fixed an issue in the internal gpio node where adding and removing ports would be picked up by the gpio router, but not on subsequent adds because some data was not being cleaned up in the devices tree.
- Fixed some issues with connection recycling in the internal gpio node.
- Added some code to prevent looping Lwrp ERROR 1000 messages in the internal gpio node between clustered Pathfinder Core PROs.
- Fixed an issue with added ports in the internal gpio node when a global ADD GPI or ADD GPO had been requested.
  - In the previous build the new ports would not be subscribed to.
- Fixed some issues with Pathfinder Core PRO routing and object updating when internal gpio port counts are changed.
- Added code to clear internal snake subscriptions when an internal gpio port is removed from the system.
- Fixed a bug in the web page with sensing whether the internal Pathfinder Core PRO gpio node was in the devices list or not
- Fixed an issue where the connection to the internal Pathifnder Core PRO gpio node was occasionally recycling based on a poll lack of response timer.
  - This was a fault in the polling when the device was local and therefore could respond threading wise before the poll send date/time stamp was set.

Version 1.5.16.18 4/15/2019

-

This version adds a virtual gpio node into PathfinderCore PRO that can interact with both snake routing and multicast gpio.

> ⓘ Important Note:
> This is a preview build of this feature and cluster synchronization of this feature is not complete yet. Please review the Beta Manual for details on this feature.
>
> There is a known bug where adding new ports do not show up in the routers. Removing the device from the devices list and then clicking on the gpio link to prompt rediscovery will fix this until we have a patch.

Version 1.5.16.17 4/02/2019

- From 1.4.16.00: Fixed a bug with case sensitivity of On/ON and Off/OFF with html5 buttons.
- From 1.4.16.00: Fixed an issue where changing backcolor on or off on an html5 button while a flash was in progress would not change the flashing color without turning flash off and back on again.
- From 1.4.16.00: Added a None option to the hwmap list to be used to unmap a mapped button.
- From 1.4.16.00: Added the ability to hwmap buttons to the over bridge display on fusion button modules as this was missing and should have been there.
- Added hardware map binding to label and input box. Still needs testing.

Version 1.5.15.16 3/22/2019

- From 1.4.15.00: Fixed a bug where removing a virtual route point's base ios would cause the underlying audio point to become unmounted from the device DST/SRC object.
  - This could break routing for that IO until a reboot takes place.
  - Also fixed some issues with mount point meta data after a removal which only came to light after this change which made remounting problematic.
  - Mounting and umounting primarily occurs in routers where fast direct access (mounting) of the underlying DST/SRC objects from the devices tree is necessary.
- From 1.4.15.00: Fixed an issue with rediscovery of deleted audio ios where the DST/SRC object still exists.
- From 1.4.15.00: Added some code to re-query the resulting DST/SRC after a manual reconnect request of the device.
- From 1.4.15.00: Fixed a bug with removing ios from the devicepath lookup collection probably introduced in 1.4.9.00.
- From 1.4.15.00: Fixed a bug where removing an IO (or other object that was mounting other objects) in some cases could cause the base device object (mounted object) to get removed from the lookup collection.
  - This would mean the object would show up in the lists when walking the tree but would return INDI NONE when accessed directly.
- From 1.4.15.00: Added a chunk of code to make sure Html5 panels clean up any residual unused bindings on save.

- From 1.4.15.00: Fixed some outdated unit tests.
- Fixed an issue with API trees and branch selection.
- Fixed issues with api tree auto selection.
- Added support for the Omnia Volt device type.
- Added hp202 as a device type.
  - We do not have one of these so this is completely untested and may or may not work with the equipment.
- Stored the DEVN from the device into the devices database so that if the device is an unknown device and is offline on startup the sapobject type would be correct rather than generic.

  - For existing unknown devices this field will be updated in the database when the device is connected to and the SapObjectType changed based on the DEVN discovery.
- Added an additional cross check when loading from the device to change an unknown device type to a known one at load time if the DEVN can be interpreted based on the new coded devices.
- Fixed an issue where devices list web page was not always updating background page data without a browser refresh.

Version 1.5.14.15 3/13/2019

- From 1.4.14.00:Fixed a memory leak in database record updating.
  - Checks to determine if we needed to insert or update were spawning a datareader object which was never being closed and therefore never released from ram.
  - This was causing more and more of these to exist in ram as data writes occurred.
  - This was especially prevalent when writing regularly changing memory slots that were set to the LastKnown state.
  - If this is your situation please also review the notes on LastKnown state for memory slots.
  - The LastKnown option should only be used when necessary as it causes additional disk I/O.
  - Use the LastKnown option only when the memory slot state cannot and will not be able to be determined via flows by the existing state of the system on startup.

Version 1.5.13.15 3/07/2019

- From 1.4.13.00: Fixed a bug introduced in 1.4.9.00 that was only removing ios from ram on either gpio or audio when a device was removed rather than both.
  - The database was being cleaned properly so a restart was necessary to fully remove those ios prior to this version.
  - This could also lead to clustering anomalies if the device was added back into the system before a restart occurred.
  - A restart of both could also be used to clear those anomalies.

Version 1.5.12.15 3/06/2019

- Fixed an issue from 1.5.10.11/12 where the API tree in logic flows was not displaying.
  - This is a quick fix to get the API tree working again for those that need it.
  - Known Issue: There is still an issue with double clicking start and endpoints that are already assigned when the API tree is selected where it may not find and select the property in the tree.
  - Known Issue: We will work on a fix for this issue in the next build.
- Fixed an issue with pulse on html5 buttons and lcd buttons where the flash would not stop if the indicator was in a blank (unset) state when the pulse started.
  - Now that is considered to be an off state.

- Fixed an issue where the disable property in Html5 buttons added in 1.5.9.10 did not work properly with mapped hardware buttons.
- Added a folder enable and disable option in logic flows that allows the enabling or disabling of all flows in the folder.
  - A recurse option also exists to change the enable/disable state on flows in the sub folders as well.
  - Please review the Beta Manual for details on this feature.

Version 1.5.12.14 3/04/2019

- Known Issue: This version has a broken API tree in logic flows. Simple tree is working with the optimizations from 1.5.10.11/12. Targeting a fix for version 1.5.12.15.
- From 1.4.12.00: Fixed an issue where the omnia one would still show as offline if it was in the database prior to the update to 1.4.11.00.
  - Since the Lwcp connection was already in the database it would still get loaded.
  - Now it gets skipped.

Version 1.5.11.14 3/01/2019

- Known Issue: This version has a broken API tree in logic flows. Simple tree is working with the optimizations from 1.5.10.11/12. Targeting a fix for version 1.5.12.15.
- From 1.4.11.00: The Omnia-one device type incorrectly had LWCP options enabled. This is fixed.
  - This was causing the device to appear as offline in some cases.
- Added the pulse option to html5 panel button indicators and LCD button states.
  - Please review the Beta Manual for details on this feature.
  - Known Issue: If the button state is blank flashing may not stop. Working on a fix for this in 1.5.12.15.
- Please review notes for 1.5.10.11 and 12 below as well.

Version 1.5.10.13 2/24/2019

> ⓘ Important Notes:
> Known Issue: This version has a broken API tree in logic flows. Simple tree is working with the

> optimizations from 1.5.10.11/12. Targeting a fix for version 1.5.12.15.
> This version (in addition to 1.5.10.11 and 1.5.10.12) have major changes to the web page data loading architectures. Please report any bugs and be prepared to roll back to 1.5.10.10 if issues are uncovered.

- Fixed an issue where the logic flows simpletree json file did not have versioning in the ajax request and so would require a Ctrl-Refresh of the browser to obtain the new changes.
- Fixed an issue from the rework of the json file in 1.5.10.11/12 which caused some devices and routers to be presented in the property tree by their id rather than their friendly name.

Version 1.5.10.12 2/20/2019

> Important Notes:
> Known Issue: This version has a broken API tree in logic flows. Simple tree is working with the optimizations from 1.5.10.11/12. Targeting a fix for version 1.5.12.15.
>
> This version (in addition to 1.5.10.11) has major changes to the web page data loading architectures. Please report any bugs and be prepared to roll back to 1.5.10.10 if issues are uncovered.

- Reworked the branch loading in logic flows property tree to allow large groups to be loaded at once without frequent rendering.
  - This improves load performance of some branches such as the main audio router branch.
- Applied the same changes as above to the logic flows folder tree.
- Added Xnode mix points back into the simple tree now that the property tree is filled on demand rather than being pre-filled.

Version 1.5.10.11 2/15/2019

> Important Notes:
> Known Issue: This version is reported to having broken the API tree in logic flows. Simple tree is working. Targeting a fix for version 1.5.12.15.
>
> This version has major changes to the web page data loading architectures. Please report any bugs and be prepared to roll back to 1.5.10.10 if issues are uncovered.

- Made some minor improvements to router web page loading for large systems.
  - Work is continuing on this optimization and should improve more in future versions.
- Changed the logic flows loading page to not pre load router resources.
- Changed the logic flows simple tree to load branches on demand rather than pre-loading data.
  - This improves a cpu hang during the initial logic flows page load on larger systems.
-

- Changed all pages that were using json pulls to use SapV2 streamed messages instead.
  - This improves cpu and memory utilization in Core PRO during large list loading as it removes the need to build large json blobs in ram.
- Changed the message counter during router loads to only change on every hundred rather than every message to reduce rendering load.
- Fixed an issue where DeviceEmulators were not appearing in the property grid when accessed from html5 panel designer.
- Added a UseStagedWrites option to MemorySlotManager to control the frequency of updates to the cf card to no faster than every minute on vm and r2 and no faster than every 15 minutes for fanless.
  - This is to save the cf card if the customer uses a fast changing memory slot with the lastknown option enabled despite the warnings against this in the manual.
  - The option can be disabled via the options file if necessary.
- Fixed a bug that was preventing object translation properties to appear in the to field of the translation dialog in certain cases.
  - It looks like this has been there for a while but may have been masked in previous versions by the lower level rfs pulls which are now staged in this version.

Version 1.5.10.10 2/13/2019

- From 1.4.10.00: Fixed a bug introduced in 1.3.13.31 with time zones that was not setting one of the operating system shortcuts to the correct time zone symlink.
  - If you suspect the need to reset the time zone first update to this version and then switch the time zone to something else and then back to the desired time zone, set the time using ntp or the set time from pc link and then reboot.
- From 1.4.10.00: Fixed a bug introduced in 1.3.13.31 where the set time from pc web ui link was not working.
- Fixed an issue introduced in 1.5.9.09 where the web page during the reboot after a bank update might navigate away from the correct system web page.

Version 1.5.9.10 2/11/2019

- Fixed a bug introduced in 1.5.9.09 that was causing cluster data to show up as routers on the routers web page.
- Added a new property to allow the enabling and disabling of html5 user buttons.

> ⓘ Important Note:
> Sometimes when rebooting the web page will try to roll to the other nic or to the other node in the cluster. This will be fixed in the next build. In the mean time you may have to type the correct url back into the browser to see the results of the update/reboot.

Version 1.5.9.09 2/8/2019

> ⓘ Warning:
> This version pulled from the site due to a bug. All changes in this version are also in 1.5.9.10 but without the bug.

- Made the browser page title reflect the panel name when opening an html user panel.
- Added a ping/pong message every 10 seconds on html user panels to act as a keepalive if network proxies or security try to close the websocket due to inactivity.
- Added code to reconnect user panels automatically if the websocket loses connection.
  - This will attempt to reconnect to the Pathfinder Core PRO currently in use first.
  - If that connection fails it will cycle through the other livewire and office tcp connections in the cluster.

> ⓘ Important Note:
>
> If the panel rolls to a different Pathfinder Core PRO in the cluster a new login may be required. We intend to address this second login request during a cluster redirect through the use of floating ips in the future.

Version 1.5.9.08 12/31/2018

- From 1.4.9.00: Improved device removal performance.
  - Certain recursive, redundant, and non-optimized calls were causing high cpu load in larger systems during a device removal while cleaning up the device's route points.
  - Added some code to prevent a device removal from making redundant database calls to the router database.
  - Made optimizations in the router io removals for a device removal by including hashed lookups of ios by device.
  - Made io removal messages when initiated by a device removal cluster quiet as they should be handled by the device removal cluster message.
  - This reduces unnecessary cluster messaging.
  - Fixed the sub cache descendent branch removals by querying the object for the additional branches to be removed rather than looping through all items repeatedly.
- From 1.4.9.00: Improved the io lookup collection for theoretically better performance.
- From 1.4.9.00: Very minor performance improvement to router load times.
- From 1.4.9.00: Added an exception to not cache messages for response for Device Emulator ToSend when forwarding them to the emulator.
- From 1.4.9.00: Made a couple of other minor optimizations which may improve performance slightly.
- Demo option (Notes 517).
- Added control option to html5 panel fader and console fader controls to support Lwch in addition to Fach as well as QOR based consoles.
  - Please review the Beta Manual for details on this feature.

- Made Lwch objects for a specific console follow the fader gain value of the console's fach object that currently has that channel loaded.
  - The subscribe command in LWCP for fader gain states in the consoles does not support a similar variation for Lwch objects so this keeps them in sync within Pathfinder Core Pro's objects.
- Hid the A/B buttons in html5 console fader controls when the control property is directed at a Qor Fach object since Radius and Iq do not have the A/B buttons.

Version 1.5.8.07 12/21/2018

- From 1.4.8.00: Extended the disk device timeout in the OS to 180 for the vm build for better compatibility with network (iscsi) backing storage.
- Added support for Imagine Logical Router Control Routers.
  - Please note: This feature is very much a beta feature at this point in time and is still undergoing testing.
  - Please review the Beta Manual for details how how to add an Imagine router to your Pathfinder Core PRO system.
- Added a load event to the web browser Html5 panel control (iframe) which cycles false when the frame starts to load and true when the loading is complete.
- Hid the network connection icon during initial panel loading in frame min to prevent its momentary appearance during launch and subpanel loading.
- Added support for the lwrp STAT DST, STAT SRC, STAT ICH, and STAT SYNC commands to Axia audio devices.
  - These features are currently only in the API tree of logic flows under the device.
  - Note: Because STAT SYNC will be infrequently used and can generate lots of traffic in large systems, it is disabled by default.
  - Enable STAT SYNC with an API command like: SET Devices#0.XNodeCombo#[tcp://172.16.1.95:93].LwrpInterpreter#0 SubscribeToSync=True
  - Please review the Beta Manual for details on properties and functionality exposed via the STAT command.
- Exposed route destination lock states in the logic flows simple tree.
- Added an html5 panel input box for capturing user input.
  - This input box can be displayed in several different types for capturing different kinds of data.
  - Please review the Beta Manual for details on this new Html5 panel control.
- Added a setlocal property to html5 main panels that can be used to set properties on components in the panel, its parent, or subpanels.
  - Syntax is target|elementid|propertyname=propertyvalue.
  - Target can either be local for the local panel, parent for the parent panel, or the name of an iframe.
  - For details on using the feature please review the Beta Manual.
- Added a panelmemoryslot control to html5 panels.
  - This control is similar to a normal memory slot except that its scope is internal to a specific instance of the panel.

- Please review the Beta Manual for details on this new Html5 panel control.

Version 1.5.7.06 12/06/2018

- From 1.4.7.00: Added some code to swap a device path in the legacy Panel device cache if the object path changes relative to the ip:port.
  - If a device type changes (engineacl to fusion), it must be removed and readded to PathfinderCore PRO to generate the new object paths.
  - If the device type changes (for example engineacl to fusion) and the device is removed and re-added, legacy hardware maps are not valid without a restart.
  - This version allows a save of the legacy panel to regenerate the hardware maps with the new device type object path after the device has been removed and re-added without a restart.
  - Io mapping on legacy panels may have to be recreated in this case due to io number changes after removing and adding the device.
  - This situation should be virtually non existent as device types do not change unless you replace the device with one of a different type.
- From 1.4.7.00: Added an exception to prevent log messages for duplicate inits when the objects are the default none, previous, and other ios.
- From 1.4.7.00: Added a DataIndex property to better track the internal database index for ios.
- From 1.4.7.00: Fixed an issue where adding audio ios via a cluster message was not incrementing the max index which meant a virtual io addition after that might try to reuse an index causing it to get created in ram but not stored to the database with a database error in the log.
- From 1.4.7.00: Added cluster synchronization of the routers MaxDataIndex.

> ⓘ Warning:
> CRITICAL NOTE: The items above fix a critical bug in clustered scenarios. Prior to this version if you added an Axia Audio device to an active cluster and then created a virtual router after that prior to any restart, the ios for that virtual router would appear on the secondary node but might not get written to the database. This means they could be missing on the secondary node after a restart.
>
> If you think your system may be in this state it is highly recommended that you backup both systems and then do a manual sync from primary to secondary.
>
> To do a manual sync from primary to secondary, go to the secondary node and from the clustering tab on the secondary node click manual sync in order to take and restore a backup from the primary node.
>
> It is recommended that you do this and then upgrade to the new version.

Version 1.5.6.06 11/20/2018

- From 1.4.6.00: Fixed a bug where Ios returned from the equipment with no rtpaddress or enabled fields (vx and intercom) could replace the none source in some routes causing those sources rather than none to display and causing corresponding virtual destinations to report an other state rather than a none route.

Version 1.5.5.06 11/07/2018

- Please review all release notes from 1.5.5.05 and 1.4.5.00 as this is just a quick patch on top of those changes.
- Fixed a bug introduced in 1.5.5.05 that was preventing flow editing within html5 panels because the link to the advanced options was not present when viewing the flows from user panels.

Version 1.5.5.05 11/06/2018

- From Patch 1.4.5.00 => Please review all release notes for version 1.4.5.00 as well as they apply to this version also.
- Worked on an analysis tool for missing mount points.
- Fixed an issue where the execution limit property could be incorrectly affecting logic flow license count.
- Added an option in the logicflows root to change the default settings for recursion.
  - Changing this will also update all translators that are currently at the default settings.
  - Property is DefaultRecursionSettings and is in the format iterations/ms.
  - For example 50/1000 means recursion will disable if the translator executes more than 50 times in 1000 ms.
- Added execution limit settings to the translator dialog to allow adjusting the recursion detection settings.
- Moved skip initial startup and recursion settings into an advanced settings in the translator ui.

Version 1.5.4.04 10/15/2018

- From 1.4.4 Patch: Fixed an issue with rtpstream address swapping such as happens with vxengine that can leave source to destination routing unresolved after rtp addresses (livewire channel number) has been changed.
- From 1.4.4 Patch: Fixed a bug where legacy panels with IO mapping were not resolving the map path when very large routers were involved because the routers had not finished loading yet.
  - This corrects an issue where the subscription to those router loading states was not passing the correct source or destination state on newly added ios during the load process.
- Fixed a bug in Logic Flows introduced in 1.5.0.01 where the ExecutionLimit propety change was being raised by EvalSuccess causing poor performance due to that property raise generating repeated saves to the cf card.
- Fixed a bug with Logic Flow license counting introduced in 1.5.3.03 or earlier.

Version 1.5.3.03 09/23/2018

- From 1.4.3 Patch: Fixed bugs with backfeed switching. Names and owners were sometimes getting confused when backfeed sources moved between faders.

- From 1.4.3 Patch: Fixed a bug with memory slots where changing the startup state when editing an existing slot was not always getting stored to the backing storage and so could revert after a restart.
- From 1.4.3 Patch: Removed an unnecessary lookup and debug message.
- From 1.4.3 Patch: Added code into the logic flows ui to issue a re-query of the show profile or source profile list when the corresponding object/property is selected.
    - The consoles do not announce additions or removals of source and show profiles but they also happen too infrequently to warrant polling so this change allows the ui to request updates when configuring those parameters.
- From 1.4.3 Patch: Fixed an issue where None (-1) was not an option in FaCH source profiles to be used for unloading a profile from a fader and indicating that unloaded state.

- Updated the vm iso installers, update packages, and images to support /dev/hda and /dev/vda as root OS disks in addition to the normal /dev/sda.
    - Required for some vm hypervisors such as Stratus.
- Added device objects for infinity devices so they would appear as inifinity objects rather than generic livewire devices.
- Added a write only property called SendToRestApi to Inifinity objects which accepts strings in the format: OPERATOR PATH VALUE (i.e. PUT /api/s/main/audio/speakerMute true) to sent http commands.
    - Responses are not functioning yet but PUTS should work.
    - It is important to know if the value needs to be wrapped in quotes and to use those quotes if necessary.
- Added a write only property to MemorySlotsManager to change the value of all slots that have a certain value.
    - Example: SET MemorySlots#0 ChangeAllByValue=oldvalue-->newvalue
    - Use --> to separate the old from new value.
- Added a CopyValue property to memory slots that allows you to copy a value from one memory slot to another via a command.
    - Example: SET MemorySlots#0 CopyTo=slot1-->slot2
    - Use --> to separate the old from new value.
    - Useful for dynamically altering the copy source and/or target via a logic flow translator.

Version 1.5.2.02 09/14/2018

- From 1.4.2 Patch: Fixed a bug where the bindings file for html5 panels was sometimes being cached by the browser causing settings to appear to not stick or revert when making changes.
    - This file is now pulled with a no cache option and with a version stamp of current time to prevent the browser from using a cached value for the json data.

Version 1.5.1.02 09/14/2018

- From 1.4.1 Patch: Saving panels was sometimes generating css in both the html and css file and it was possible at times for them to conflict - especially with hardware mapping.
    -

  - The save now strips out any inline styles in the html leaving all styles in the css file only.
- From 1.4.1 Patch: Fixed an issue where changing a color on one button and then moving to the next and clicking the color again would sometimes update immediately to the color of the previously changed button.
- Fixed an issue where the log message generated by translator recursion detection was not including the path to the translator being disabled.
- Changed the link to the new beta features manual for 1.5 and set display to visible again for it.
- Added an init messages field to generic emulators in order to send a login or other init message on connect.
- Added a Connected Property to Device Emulators for logic flows to determine whether the emulator has any successfully connected clients.
- Added a ConnectedCount Property to Device Emulators for logic flows to start based on the number of connected clients.
- Added a ConnectionLost Property to Device Emulators for logic flows to trigger off of when connections get lost.
  - This property pulses true then false when a connection gets lost.
- Added a ConnectionObtained Property to Device Emulators for logic flows to trigger off of when connections get obtained.
  - This property pulses true then false when a connection gets obtained.

Version 1.5.0.01 09/11/2018

- First beta build after 1.4 release.
- Added swap as an option in html5 panel and control's visible property.
  - Switches between visible and hidden values.
- Added a SendCriticalMessage Property to the email host object that can be used to send messages to the critical event email address.
  - Property should be set with a subject and message where the subject will be split from the message at the first carriage return and line feed.
  - May not be usable via logic flows yet, only via the API.
- Added recursion detection to translators to prevent a badly formed recursive or looping logic flow from killing the cpu or causing a stack overflow.
  - The default functionality is to disable the translator if the translator is being analyzed for execution more than 50 times per 1000 milliseconds.
  - These parameters may be changed via the api by setting the ExecutionLimit property on the translator.
  - For example, to make the flow get disabled if it is executed more than 20 times per 1000 ms, use the command:
  - set LogicFlows#0.LogicFlowFolder#Test.ListTranslator#6ef05ca8-e530-4326-8a35-85e9e41495af ExecutionLimit="20/1000"
  - In the case that a translator gets disabled by recursion detection, a log message will be generated and an attempt will be made to send an email to the critical event email address.

  - There is no UI yet for editing the recursion detection settings to anything other than the default.
  - It currently must be changed to anything other than the default via the API.

1.4.xx.xx

Version 1.4.20.00 8/1/2019

- Added some changes to the metering device object to correct the possibility of a crash during a lost connection disabling of the timers.
  - We have seen two sites that have exhibited crashes with a logged stack trace on the same line of code.
  - This line of code deals with the disabling/enabling of certain timers during a loss of connection to the equipment of a metering device.
  - While we have not been able reproduce in the lab we believe the changes in this version should correct the problem.

Version 1.4.19.00 5/17/2019

- Included the VMware paravirtualized scsi driver in the vm and installer images.
  - This version only makes a change in the vm version.
  - The driver has been included for testing with VMware sites where ISCSI is being used for the host storage.

Version 1.4.18.00 5/13/2019

- Reworked day of week timer threading to prevent day of week timers from getting stuck in an elapsed state when they are supposed to flip back to not elapsed 10 seconds after execution.
  - Added an hourly safety catch to catch any that are in the wrong state.
  - Improved performance of timer lookups and moved some of the dayofweek elapsed resetting into each day of week object.
- Fixed a bug in object translators where if the input was a specific property rather than an object but the output was an object, the system would process other properties on the input side than just the one selected.
- Cleaned up some css around meters.
- Fixed a problem with html5 gradient meter off colors shifting the transition points if they were not all the same color.
- Fixed a bug that was causing the schedule column for DayOfWeek events to switch to a number rather than the correct value until the web page was refreshed.

Version 1.4.17.00 4/30/2019

- Fixed a bug with user log rotation that was causing rotated files to have large blocks of NUL characters at the beginning in certain situations.
- Fixed an issue where activate scene did not submit the scene messages to the system in the order specified.
  - It is important to understand that even though the scene will now submit the change messages in order that does not guarantee they will be completed in order.
  - Each Scene item is sent to the system without waiting for completion.
- Added code to the silence alarms to more quickly detect device failures.
  - Currently the default Linux settings will only raise an error on tcp send failures after approximately 15 minutes of failed retry attempts.
  - This is much longer than the similar windows defaults.
  - Device management relies on application level ping messages to detect failures quicker but this was not implemented in silence alarm device connections.
  - In this version silence alarm device connections will also drop the tcp connection and will start counting down the failure after approximately 45 seconds of failed ping (VER) messages.
- Switched the SRC Rtpp property to be read/write.
- Fixed a bug where opening an object translator created with an API property while the Simple tree is selected and then selecting the item in the translation convert list might leave the property name field blank which after committing would wipe out that api property name in the conversion.
- Disallowed most punctuation in panel names.
  - This fixes some panel incorrect functionality when certain characters such as the slash are used in the panel or page name.
- Moved the meters and scenes to startup prior to logic flows in the service startup list.
- Added some threadlocking around device online/offline state in router IO DeviceOnline property setting to prevent a potential race condition where offline to online messages in quick succession especially during initial load could theoretically interfere with each other.
- Fixed an issue with gpi dynamic additions not getting an index in the device io collection.
  - This can cause the ios not to show online/offline correctly.

Version 1.4.16.00 4/02/2019

- Fixed a bug with case sensitivity of On/ON and Off/OFF with html5 buttons.
- Fixed an issue where changing backcolor on or off on an html5 button while a flash was in progress would not change the flashing color without turning flash off and back on again.
- Added a None option to the hwmap list to be used to unmap a mapped button.
- Added the ability to hwmap buttons to the overbridge display on fusion button modules as this was missing and should have been there.

Version 1.4.15.00 3/22/2019

- Fixed a bug where removing a virtual route point's base ios would cause the underlying audio point to become unmounted from the device DST/SRC object.
  - This could break routing for that IO until a reboot takes place.
  - Also fixed some issues with mount point meta data after a removal which only came to light after this change which made remounting problematic.
  - Mounting and umounting primarily occurs in routers where fast direct access (mounting) of the underlying DST/SRC objects from the devices tree is necessary.
- Fixed an issue with rediscovery of deleted audio ios where the DST/SRC object still exists.
- Added some code to requery the resulting DST/SRC after a manual reconnect request of the device.
- Fixed a bug with removing ios from the devicepath lookup collection probably introduced in 1.4.9.00.
- Fixed a bug where removing an IO (or other object that was mounting other objects) in some cases could cause the base device object (mounted object) to get removed from the lookup collection.
  - This would mean the object would show up in the lists when walking the tree but would return INDI NONE when accessed directly.
- Added a chunk of code to make sure Html5 panels clean up any residual unused bindings on save.
- Fixed some outdated unit tests.

Version 1.4.14.00 3/13/2019

- Fixed a memory leak in database record updating.
  - Checks to determine if we needed to insert or update were spawning a datareader object which was never being closed and therefore never released from ram.
  - This was causing more and more of these to exist in ram as data writes occurred.
  - This was especially prevelant when writing regularly changing memory slots that were set to the LastKnown state.
  - If this is your situation please also review the notes on LastKnown state for memory slots.
  - The LastKnown option should only be used when necessary as it causes additional disk I/O.
  - Use the LastKnown option only when the memory slot state cannot and will not be able to be determined via flows by the existing state of the system on startup.

Version 1.4.13.00 3/07/2019

- Fixed a bug introduced in 1.4.9.00 that was only removing ios from ram on either gpio or audio when a device was removed rather than both.
  - The database was being cleaned properly so a restart was necessary to fully remove those ios prior to this version.
  - This could also lead to clustering anomolies if the device was added back into the system before a restart occurred.
  - A restart of both could also be used to clear those anomolies.

Version 1.4.12.00 3/04/2019

- Fixed an issue where the omnia one would still show as offline if it was in the database prior to the update to 1.4.11.00.
    - Since the Lwcp connection was already in the database it would still get loaded.
    - Now it gets skipped.

Version 1.4.11.00 3/01/2019

- The Omnia-one device type incorrectly had LWCP options enabled. This is fixed.
    - This was causing the device to appear as offline.

Version 1.4.10.00 2/13/2019

- Fixed a bug introduced in 1.3.13.31 with time zones that was not setting one of the operating system shortcuts to the correct time zone symlink.
    - If you suspect the need to reset the time zone first update to this version and then switch the time zone to something else and then back to the desired time zone, set the time using ntp or the set time from pc link and then reboot.
- Fixed a bug introduced in 1.3.13.31 where the set time from pc web ui link was not working.

Version 1.4.9.00 12/31/2018

- Improved device removal performance.
    - Certain recursive, redundant, and non-optimized calls were causing high cpu load in larger systems during a device removal while cleaning up the device's route points.
    - Added some code to prevent a device removal from making redundant database calls to the router database.
    - Made optimizations in the router io removals for a device removal by including hashed lookups of ios by device.
    - Made io removal messages when initiated by a device removal cluster quiet as they should be handled by the device removal cluster message.
    - This reduces unnecessary cluster messaging.
    - Fixed the sub cache descendent branch removals by querying the object for the additional branches to be removed rather than looping through all items repeatedly.
- Improved the io lookup collection for theoretically better performance.
- Very minor performance improvement to router load times.
- Added an exception to not cache messages for response for Device Emulator ToSend when forwarding them to the emulator.
- Made a couple of other minor optimizations which may improve performance slightly.

Version 1.4.8.00 12/21/2018

- Extended the disk device timeout in the OS to 180 for the vm build for better compatibility with network (iscsi) backing storage.

Version 1.4.7.00 12/06/2018

- Added some code to swap a device path in the legacy Panel device cache if the object path changes relative to the ip:port.
  - If a device type changes (engineacl to fusion), it must be removed and re-added to PathfinderCore PRO to generate the new object paths.
  - If the device type changes (for example engineacl to fusion) and the device is removed and re-added, legacy hardware maps are not valid without a restart.
  - This version allows a save of the legacy panel to regenerate the hardware maps with the new device type object path after the device has been removed and re-added without a restart.
  - Io mapping on legacy panels may have to be recreated in this case due to io number changes after removing and adding the device.
  - This situation should be virtually non existent as device types do not change unless you replace the device with one of a different type.
- Added an exception to prevent log messages for duplicate inits when the objects are the default none, previous, and other ios.
- Added a DataIndex property to better track the internal database index for ios.
- Fixed an issue where adding audio ios via a cluster message was not incrementing the max index which meant a virtual io addition after that might try to reuse an index causing it to get created in ram but not stored to the database with a database error in the log.
- Added cluster synchronization of the routers MaxDataIndex.
  - CRITICAL NOTE: The items above fix a critical bug in clustered scenarios.
    - Prior to this version if you added an Axia Audio device to an active cluster and then created a virtual router after that prior to any restart, the ios for that virtual router would appear on the secondary node but might not get written to the database.
    - This means they could be missing on the secondary node after a restart.
    - If you think your system may be in this state it is highly recommended that you backup both systems and then do a manual sync from primary to secondary.
    - To do a manual sync from primary to secondary, go to the secondary node and from the clustering tab on the secondary node click manual sync in order to take and restore a backup from the primary node.
    - It is recommended that you do this and then upgrade to the new version.

Version 1.4.6.00 11/20/2018

-

Fixed a bug where Ios returned from the equipment with no rtpaddress or enabled fields (vx and intercom) could replace the none source in some routes causing those sources rather than none to display and causing corresponding virtual destinations to report an other state rather than a none route.

Version 1.4.5.00 11/06/2018

- Locked down device addition and loading to license validation.
- Fixed an issue where enabling two identical channel numbers and then disabling one of them could leave routes in a state where the route takes but does not display properly in PCP.
  - This fix involved recrafting the rtp stream address lookup table and how it functions relative to address changes and enable/disable of properties.
- Fixed an issue with non standard engine version numbers so that an error is not thrown and caught.
- Recrafted database writes for devices and routers databases so make sure deleted and creates are done atomically.
  - Previously there was a possibility of deletes not being completed before create messages got added into the queue.
  - This would affect and be primarily seen when thirdy party apps delete an entire router and then immediately recreate the router and ios.
- Fixed a benign log message about duplicate ios being added for aes67, none, and previous sources.
- Fixed a bug where deleting a device with an ip that is the beginning of another ip (172.16.25 and 172.16.251) could cause deletion of both devices IOs from the back end database.
- Modified some database code in the routers database to make it more transactional.
- Added code to clean up SystemIos, AxiaAudioIos, and AxiaGpios if the router that gets deleted is and AxiaAudio or AxiaGpio router.
- Fixed a bug where router manager was not always successfully subscribing to device deletions meaning that a device deletion might not clear up the ios for that device from the audio and gpio routers.
- Refactored device status discovery code during start up of routers to enforce all subscriptions before all getters.
- Added a mount complete property to ios and virtual base ios to detect whether the backing object has been successfully mounted.
  - For example: get Routers#0.AxiaAudioRouter#1 MountComplete=False $MAX_DEPTH=-1
  - The above will return mount points that have not been established between the route point and its base IO from the devices branch.
  - It is normal for some engine resources to not be mounted as engines sources and destinations are not returned if they are inactive.
- The router database had a stream enabled field for Axia sources but it was not being read from and written to.
  - Changed it so that this is stored to the database.
  - This means the first start after upgrading will do a lot of writing to the database as this field gets set to its normal state.
  - Also sources will be loaded with rtp not enabled until it detects from the device that it is unless a state (non blank) is stored in the database.
  -

-
    - This also allows for better troubleshooting when backups are sent to support since the rtp enabled state will reside correctly in the database.
- Moved the thread locking to include the initial search when attempting to fill a mount point.
    - This helps prevent a potential race condition where the object is not found but then added before ending up the pending collection leaving it permanently unmounted.
- Added some code to double check mount points after they have been added to the pending collection to handle a potential race condition case where the object gets added between searching for it and adding its absence to the pending collection.
    - This helps prevent mounts from becoming stuck unmounted.
- Fixed a bug in the UI where virtual base points would not show if their mapped io did not exist.
- Fixed a bug where the removal of devices and routes would not cause the mapped virtual ios to be added back into the mountpending collection.
    - Therefore if the device was added back into the system the virtual points would not re-acquire the map.
- Fixed a bug where deleting a device and its Audio sources and destinations would cause virtual points that map to those points to drop their base io existence in the database only.
    - This would make things look like the base ios still exist but after a restart they would get removed.
- Fixed an issue where move and push would not cause a relook up of the modified virtual io route state causing them sometimes to appear as blank routes until the next change.
- Fixed an issue in the UI where removing an IO might not remove from the routes table without a refresh of the web page.
    - This was also causing extra IOs to appear after moving or pushing until a refresh of the web page was done.
- Eth4Can had HasAudio set to true. It is now false.
- Made buttons in html 5 panels work properly with touch by emulating mouse down and up with touch and release rather than click with touch.
- Adding proper touch/multi touch control to the console fader buttons.
- Fixed a bug where Previous states in Axia Audio Destinations could change when there was no actual change of the multicast address.
    - For example changing the address field from 239.192.0.101 <> to 239.192.0.101
- Fixed a bug related to changing an html5 panel Control id whose properties are mapped to other items on the same page that could leave the buttons flows non working and the old button id orphaned in the logic flows.
- Fixed an issue where labels and images could not be moved on the html5 panel using the arrow keys because they shifted their focus.
    - Used drag helpers for these objects so that now they are movable with the keyboard keys.
- Fixed an issue in html5 designer where after a dropping a control onto the panel the keyboard move buttons would not work without a second click on the control.
- Fixed a bug with collections that was allowing the OnCollectionChanged event to be raised before items were added to the index.
    - This was causing scenes to not set their initial state properly.
- Hid the include logs option on the restore dialog.

    - There are security issues with overwriting all logs.
    - This option fails if turned to true.
    - For now hiding the option until such time as we decide to handle such issues.
- Fixed an issue in the logic flow editor where the views scroll bar was disappearing when the canvas size changed.
- Fixed an issue where the src property of image html5 elements was not hooked up to the image selector dialog.
- Changed the file date time format on the backups page for more accurate sorting when the date/time column is used for sorting.
- Added a write only Requery property to virtual routers that would double check any missing mount points and then recheck the route state of each io in the router.
    - This is not for normal use but could be useful to force a recheck of all route points in the virtual router.
    - Set Router#0.VirtualRouter#3 Requery=True
- Added an icon to trigger the new requery property on the routers page on each Virtual Router line.
    - Upon clicking, it will ask for confirmation before triggering the requery.
    - Upon clicking, it will ask for confirmation before triggering the requery.
    - This can be used to recheck the route status amd mount state of each IO in the virtual router.
    - Should not be normally used (hence the warning confirmation) and can cause cpu load on larger virtual routers while the analysis loop takes place.
- Fixed an issue where renaming a logic flow view/folder with logic flows that were disabled in it where the flows also had combiners could leave the folder with scrambled flows.

Version 1.4.4.00 10/08/2018

- Fixed an issue with rtpstream address swapping such as happens with vxengine that can leave source to destination routing unresolved after rtp addresses (livewire channel number) has been changed.
- Fixed a bug where legacy panels with IO mapping were not resolving the map path when very large routers were involved because the routers had not finished loading yet.
    - This corrects an issue where the subscription to those router loading states was not passing the correct source or destination state on newly added ios during the load process.

Version 1.4.3.00 09/23/2018

- Fixed bugs with backfeed switching. Names and owners were sometimes getting confused when backfeed sources moved between faders.
- Fixed a bug with memory slots where changing the startup state when editing an existing slot was not always getting stored to the backing storage and so could revert after a restart.
- Removed an unnecessary lookup and debug message.
- Added code into the logic flows ui to issue a requery of the show profile or source profile list when the corresponding object/property is selected.

  - The consoles do not announce additions or removals of source and show profiles but they also happen too infrequently to warrant polling so this change allows the ui to request updates when configuring those parameters.
- Fixed an issue where None (-1) was not an option in FaCH source profiles to be used for unloading a profile from a fader and indicating that unloaded state.

Version 1.4.2.00 09/14/2018

- 1.4.2.00 Patches.
- Fixed a bug where the bindings file for html5 panels was sometimes being cached by the browser causing settings to appear to not stick or revert when making changes.
  - This file is now pulled with a no cache option and with a version stamp of current time to prevent the browser from using a cached value for the json data.

Version 1.4.1.00 09/14/2018

- 1.4.1.00 Patches.
- Saving panels was sometimes generating css in both the html and css file and it was possible at times for them to conflict - especially with hardware mapping.
  - The save now strips out any inline styles in the html leaving all styles in the css file only.
- Fixed an issue where changing a color on one button and then moving to the next and clicking the color again would sometimes update immediately to the color of the previously changed button.

Version 1.4.0.00 09/10/2018

- 1.4.0.00 release build for fanless engine, r2, and vm platforms.
- This build is the same as 1.3.13.35 but moved to the master release branch and versioned as 1.4.0.00.
- All future bugs related to features already in this version will become patches to 1.4 versioned as 1.4.x.00 where x is the patch number.
- All new features will be created in the next 1.5 beta branch.
- See Version Numbering for details.

1.3.xx.xx

Version 1.3.13.35 09/10/2018

- Removed the service restart buttons from the services page as they are no longer effective now that the services are launched internally to the supervisor process.
- Event system state on the logic flows page was superimposed over the nic state icon. Adjusted css to fix.

- Fixed an issue where saconsole buttons in html5 panels were leaving the inner button visible when the button was set to hidden.
- Fixed a bug where if you saved a panel using legacy PanelDesigner and entered the name during the save instead of before the save and the name has a space, it would not get replaced with underscores and the panel would be undeletable due to an incorrect filename.

Version 1.3.13.34 09/07/2018

- Fixed a bug in html5 panels where if editing two different panels on two browser instances, clicking save on one would cause a reload on both.

Version 1.3.13.33 09/06/2018

- Fixed a bug with pasting copied custom user controls.
  - For example custom console buttons would appear with double inner buttons after saving pasted components.
- Added some fixes for ampersand encoding and decoding in links where the links are exposed as values in the html5 panel property grids.
- Fixed a bug where if you created a panel with bindings, then deleted the panel, then created the panel again with the same name and bindings, the binding logic flows would not get created the second time without a restart.
- Updated production installer to ask about initial ip addresses for multiple production builds in the same network.
- Bumped the font size in crystal fonts display up one size in Linux to look better under dotnetcore.
- Added a bit of code to handle engine aes67 responses without the 5004 port.
- Removed an unnecessary force of the nic name to Eth0/1 during boot.
- Fixed a benign issue with setting network address with the cfontz display and console menu related to storing broadcast address and prefix values.
- Fixed a bug where object translators could not be used in panel bindings.
- Added a view button to the panel designer page which opens a running instance of the panel in a window.
- Fixed a bug with the IFB changes that was not sending the correct syntax for ptt on the up state.
- Fixed an issue with the flash property and legacy panel hardware mapping.

Version 1.3.13.32 08/28/2018

> Important Note:
> Please review release notes for version 1.3.13.31 relative to changes in the operating system as these also apply to this version.

- Cleaned up presentation of lwcp objects and properties for Qor.

  - Some properties and objects were being presented based on similar objects in Fusion but do not apply in Qor.
  - They have been removed in Qor branches of the logic flows property tree and API.
- Fixed an issue where monitor section properties were not being displayed in the Logic Flows simple tree.
- Fixed an issue where monitor section objects and properties were sometimes not appearing after a restart until a button was pushed.
  - Note that many of the monitor section properties are read only and so will only be presented in relation to logic flow start points.
- Fixed an issue with html panel faders outputting an incorrect value to xnode gain states.
- Added correct numerical range for INGN and OUGN in attributes for SRC and DST.
- Fixed the logic flow tree css so that properties line up under their parents more intuitively.
- Added support for Element/Fusion fader IFB. Please review the beta documentation for important details on using these parameters.

Version 1.3.13.31 08/22/2018

- This version is functionally equivalent to 1.3.13.30 but with a new operating system and target framework.
- Starting with 1.3.13.21 we have had parallel development branches with a new operating system and target framework. This alternate version (referred to as the buildroot version) has been going through internal testing as well as testing with select customers for several months. The difference in this version is that the operating system has been completely reworked to be a purpose built version of Linux rather than being based on a stripped-down version of Debian. This gives us much more control over what exists in the operating system. Additionally any of the code based on dot net technologies has been reworked and recompiled to target the dotnetcore framework as opposed to the mono framework. We have done this to obtain the following advantages:
  - Better control over the operating system and an optimized build process.
  - Better control over the codebase.
  - Much smaller footprint on the disk (~300MB as opposed to ~600MB). This leaves more space for image files with panels, etc.
  - Smaller update packages (~100MB as opposed to ~200MB)
  - Significant performance improvements and lower cpu utilization under dotnetcore.
  - Improvements in memory consumption and management.
- Subsequent releases will use this new operating system and codebase.

> ⓘ Warnings:
> Users of PathfinderCore PRO on the fanless engine platform (as opposed to vm beta testers) will see an initial increase after startup in memory consumption. This is due to the fact that this build moves from a 32bit OS to a 64 bit OS on the Fanless Engine platform. However we see much better memory management as the system runs.

> The update package will make certain changes to the boot sector. Upgrading and downgrading of software will continue to work properly in all but one situation. If you upgrade a bank to .31 or later and then switch back to a pre .31 version and try to overwrite the .31 bank with a pre .31 software image, the write will work but booting into that downgraded bank will hang. Power cycling will then boot bank into the previous bank. In this case you can run a special update package that only fixes the boot sector. Note this only happens when running on a pre .31 bank and trying to downgrade a .31 bank to a pre .31 software image. Most customers will never experience this. We only expect to see this with systems integrators and support engineers who may need to be switching their software versions regularly to match that of their customers.
>
> The repair update package can be obtained at Boot Sector Fix Fanless Engine or Boot Sector Fix VM

Version 1.3.13.30 08/22/2018

- Fixed an issue with logic flows startup where external start point states were being requested numerous times if the same startpoint existed in multiple flows when it only needed to be requested once.
  - In some cases this was causing heavy analysis and long startup times for complex flows where certain start points were reused regularly.
  - Added a hash table to make sure requests only happen once after the initial flow objects are loaded.
- Changed subscriptions in sapclients to use indexing for non-max depth subscriptions for better lookup performance.
- Stopped the last evaluated property from being raised in logic flows to reduce subscription analysis.
- Removed some unnecessary and noisy sap messages related to logic flows for faster response times and cleaner analysis.
- Reworked LogicFlow Manager to handle logic flow internal changes directly rather than subscribing and passing them through the normal subscription matching process for better performance.
- Added an option to combiners called RaiseOutput that takes two values: RaiseOutputOnSet(Default) and RaiseOutputOnChange.
  - This defines whether an analysis due to a change on a combiner input causes the rest of the flow to execute if no change to the combiner's output occurs.
  - When set to RaiseOutputOnSet, any change entering the combiner will cause blocks beyond the combiner to also be analyzed and set.
  - When set to RaiseOutputOnChange, blocks beyond trhe combiner will only be analyzed if the output state of the combiner changes.
  - This option can be very useful in certain situations to reduce analysis load.
  - See beta manual for more details on this feature.
- Also added a write only property on Logic Flows folders called ChangeAllCombinerRaiseOutput which can be used via the API to set all combiners under that folder and its subfolders to one state or the other.
  - Note that if you change a combiner from the default RaiseOutputOnSet state to the new RaiseOutputOnChange state this will alter the stored init messages in a way where that flow will not

be loadable by older versions of the software.

- Added UI changes for the raiseoutput property in combiners.
- Changed the watchdog ping startup delay to 10 minutes from 2 minutes for very large systems that have a great deal to load on startup.
- Fixed the ChangePage functionality in html5 panels.
  - This was broken in 1.3.13.28 and 29.
- Fixed the occasional double click necessary to move between pages in Html5 Panels.
- Removed Analog micro logging from php scripts.
  - This logger was accidentally left in place for debugging and was generating logs in the /tmp folder which is not managed by log rotation.
  - In the case of buildroot preview versions this was leaking ram since /tmp is a ramdisk in that version.
  - In the case of normal versions this was causing unnecessary space to be used on the cf card.
- Fixed an issue where bindings on html5 panel controls were not being copied when a copy and paste took place.
- Fixed a bug where changing the name of an html5 panel control with bindings would cause the bindings to get lost.
- Fixed a bug in html5 panels where removing all characters from a control name would cause it to jump to the far left corner and if you then try to select it without first giving it a valid name (leave the name blank) it would become unselectable.
- Fixed a bug with the new route locking where changing pages in the router was causing locked ios to show up with an unlocked icon even though they were still locked.
- Fixed an issue where when editing a virtual router, the edit page would not return to the points tab.
- Added additional version stamps to html5 user panel links to reduce browser cache problems.
- Made the html5 panel designer save button wait for confirmation of successful save from the server before refreshing the page to allow for slow writes of large panels on cf cards.
- Added the swap option to html5 panel button indicators.

Version 1.3.13.29 08/09/2018

- Fixed a bug with users and new object messages where if a standard user with no rights to the created object was logged in, it could use an indi NONE message for all user socket responses instead of the correct message depending on the user's rights.
  - This was causing problems with clustering as well in some cases as an new NONE was being generated instead of the new object to the other node.
  - This only occurs randomly when a standard user with limited rights is logged in
  - Though it can occur consistently for a while depending on the order of the client returns when iterating through the clients where the first in the iteration does not have rights to the object.
  - Fixed in this version.
- Because of the bugs addressed by this version, a manual sync may be a good idea after updating both nodes in the cluster just to ensure nodes are properly synchronized.

Version 1.3.13.28 08/02/2018

- Fixed a problem with new user creation with the route locking options.
  - They were not properly exposed in the constructors.
- Fixed a bug in html panel console channels where double digit fader numbers (10,11) were being confused with 1 due to an incorrect startswith line of code.
- Fixed some other panel bugs related to object path matching.
- Added user panels into user api security links.
  - User security for html panels is still in progress and not complete in this version.
- Fixed a bug where any login was updating clustering lastupdate on all users during the validation process.
- Fixed a bug where no access was updating a noaccess security profile in the database for each failed login.
- Fixed issues with user clustering.
  - Removal objects were not being created when users were removed and cluster was not in sync meaning the users could reappear when synchronization was reestablished.
- Improved the network connectivity and route locking icons.
- Fixed a bug with cloning panels where the folder security was not being set properly so new pictures could not be uploaded into the cloned directory.
- Added an option to fix panel directory security for panels cloned under previous versions.
  - Login to port 9600 with the appropriate user name and password and send SET UserPanels#0 FixPanelSecurity=True
  - This command should only be necessary if you have cloned html panels in versions prior to this version and can't upload pictures to the cloned panels.
  - Contact support if you need help.

Version 1.3.13.27 (Internal Only) 07/31/2018

- Fixed a race condition that could happen with virtual routers where the sources/destinations have multiple base points.
  - It is expected that these will pass through a -3 other state as route changes happen, but prior to this fix it was possible that while the route state would be correct in the data structure, the -3 might get sent out the subscribed ports after the full change.
- Added route locking in routers. See beta documentation for details.
- Added CanLock and LocksDoNotApply options to users.
  - Note: Locks do not apply option is working for the API but not for the web page UI at this point in time.
  - When locking a virtual route the underlying base point will not be locked.
  - When locking a base point, the virtual destination will become system locked and the lock can only be cleared by clearing the base point lock.
  - See Beta documentation for more details.
-

- Added testing support for in development equipment.

Version 1.3.13.26 (Internal Only)   07/23/2018

- Fixed a clustering bug for memory slot and html panel states.
  - With memory slots whose startup state are not set to last known, the current state is not written but the last sync value on a restart thinks sync is up to date so it was not pulling the latest value from the other node.
  - Similar issue with many html property states.
  - Clustering now pulls all sync dates for these objects to make sure sync state is updated on these objects.
- Fixed an issue with image syncing after one of the nodes was offline.
- Decreased ping time for clustering for more frequent pings.
- Added a warning that images may not be uploaded if the panel has not been saved - i.e. still in New Panel state.

Version 1.3.13.25 (Internal Only)   07/20/2018

- First version with Panel clustering for internal testing.
- Fixed a bug with SapProperty generation in messages when there are embedded encapsulations.
- Fixed <br> in a label so that it converts to \n when sent to Lcd buttons.
- Added a system item and used it for the initial get request to exclude NOSYNC and non cluster items - $CLUSTER_SYNCABLE
- Fixed a clustering issue where during initial synchronization one system would ask for changes since the time of the last known in sync and the return would include times equal to rather than just later than.
  - This could have unexpected results for objects changes right before one of the servers wen offline.
  - Example create a panel when both in sync. Shut down one and delete the panel on the other.
  - During a restart, the panel might get recreated.
- Fixed the css refernces to include version data to help with browser caching problems with the css when working on a panel design.

Version 1.3.13.24   07/13/2018

- Fixed an issue where the parameters of the disk space added to the system status page in 1.3.13.23 were reversed.
- Fixed the message displayed in logic flows regarding disk space to be more descriptive.

Version 1.3.13.23   07/11/2018

- Added the ability to send ascii as hex in translators and Lwrp/Lwcp tosend using \%XX where XX is the hex numeric value to send.
  -

    Escape values are now: \cr, \lf, \t, \%XX.
    - Double slash to escape the escape slash and send the literal value.
    - So for example \%41 sends A but \\%41 sends \%41 and %41 sends %41.
- Fixed an issue with the disk space alerts added in 1.3.13.22 that was sending erroneous -1 values.
- Fixed an issue where the reported diskspace alert values would not match the alert parameter but rather the current value which made it hard to match in flows.
- Added the Disk Space alert to simple flows.
- Added Disk Space to the system status web page.
- Made the routes tab the default tab when you open a router rather than the points tab.

Version 1.3.13.22 07/04/2018

- Added a checkbox to enable/disable push and move on virtual router editing.
- Added a confirmation message on push changes in virtual router editing.

Version 1.3.13.21 06/30/2018

- Fixed a bug that was causing a loop looking up all ordinal base points in a virtual route over and over again times the number of ios in the router when opening the edit page for the virtual router.
- Fixed a bug where if a source was added to the virtual io base package of a destination, the application could crash during startup.
- Added an advanced option to check whether logs need to be rotated every x number of message writes.
    - Enable this option in advanced options by adding: SET Logs#0 CheckRotationAfterMaxWrites=250
- Added a diskspace object to the API including an alerts property that can be set via advanced options.
    - By default this is enabled for 95%
    - This feature is still a work in progress and is only available for monitoring in logic flows via the api tree.
    - This feature will be fine tuned and added to system level email alerts soon.
- Increased the open file limit from 1024 to 4096 in order to increase the number of available open sockets, etc.
- Added the IO number in the routes page for virtual routers.
- Fixed an issue where the sort arrows for some routers for the destination name would instead be tied to the pointer arrow.
- Added a confirmation message if you attempt to delete a base io on a virtual source or destination.
- Fixed an issue where newly discovered devices would sometimes show as offline even though they were online.
- Added a manual reconnect api property to lwrp and lwcp interpreters.
- Fixed a bug that was sometimes causing the connection recycling on a failure to stop attempting to connect.
- Fixed an issue for better manual recycling of connections.

- Added an icon to the devices web page to allow recycling of device connections manually.
- Added livewire channel number column to the import lists. This column is only visible if importing sources from an Axia Audio Router.
- Added an alert error message if the user tries to change a virtual io id to one that is already in use in the router.
  - Previously it would just overwrite the io.
- Hid certain router editing fields when editing is disabled due to user rights.
- Added the ability to import starting with a specific number in the virtual router.
  - The default value of -1 will import at the end of the router.
- Added cross checking to make sure importing at a certain id would not overwrite existing ids.
- Got moveup, movedown, pushup, and pushdown in virtual routers working.
  - Move up and move down will increase or decrease the id of a selected IO swapping with the one it is moving to if it is already in use.
  - Push up and push down will increase or decrease the id of a selected IO pushing all later/earlier ones up or down up to the next hole in the ios.
  - Push down will fail if there are no available open id numbers.
  - Down or up refers to the Virtual IO numbering which may or may not be the current sort order in the grid.
  - It is highly recommended to sort the router by the IO number when using this feature to prevent confusion.
  - These features can be used to specify specific numerical ordering in a virtual router to match controller system numbering.
  - We may still revisit this in the future to migrate this to a in browser change that is applied rather than actually changing the router on the fly for each change.
- Fixed a bug that was causing gpios to not mount backing device data correctly in some cases.
- Fixed an issue where incorrect pending import ios could wind up in the import list if cancel was selected and a new router or import list selected before the first ones were complete.

Version 1.3.13.20 05/10/2018

ⓘ Important Notes:
First public beta of html5 panels. Please review the beta documentation for version 1.3.13.20 in detail: Beta Feature Documentaion 1.3.x

This version is a combination of several internally released versions.

Please review the notes below for versions 1.3.13.14 through 1.3.13.20 for the full scope of the changes.

There are a number of changes and fixes in addition to the html5 user panel changes.

Html5 panels will not synchronize in a cluster in this version. That is coming soon.

Html5 panel faders will not work with Qor in this version. That is coming soon.

- Html5Panel changes since internal release 1.3.13.19
  - Added versioning to the html panel sub panel iframe loading to improve browser caching issues.
  - Fixed some of the magnet snapping that was acting a bit funky to be more intuitive.
  - Fixed bugs with the spread functionality.
  - Stored the snapto and magnet states in browser local storage so the state is remembered on page reloads.
  - Fixed an issue with multiselect dragging that was requiring the shift or ctrl key to be held while dragging when custom type elements were selected.
  - Fixed an issue where clicking property row names were not always updating to the correct flow display.
  - Fixed a bug where not all name data for button devices was being filled into the data grid.
  - The flow block was not being hidden when changing selected objects.
  - Fixed a bug with changing bindings that would sometimes add duplicate binding properties into the backing storage causing unpredictable binding behavior.
  - Added some code so that selecting a hardware map button will automatically enable hardware map bindings.
  - Fixed a panel de-serialize issue with bound IO fields.
  - Got changing of io on faders and meters working.
  - Got proper io selection on AudioIO based binding.
  - Fixed an issue with apostrophes in user panel names.
  - Fixed box shadow property in grid to show actual box-shadow value. Previously it was trying to do an RGB conversion.
  - Wrapped up beta documentation for html5 user panels.

Version 1.3.13.19 05/04/2018

- Moved OS to use mono 5.8 as there are reports of a memory leak surrounding file objects in 5.10.
- Optimized a bit of code with panel hardware map loading
- Added user names to the access violation log message.
- Excluded none as a possible access violation.
- Fixed logging of login attempts using the alternate login . username=xxx password=xxx method
- Fixed an issue introduced in 1.3.12.12 that affected io discovery in certain situations.
  - Occasionally it would not add new ios discovered during startup.
- Made individual Router IO deletions cause a re-query of the device data which causes the io to get added back if auto insert is on.
  - 10 second delay on the re-add.
- Fixed an incorrect value in default advanced options for SkipCleanLogs.
  - SET Logs#0.SkipCleanLogs=False should have been SET Logs#0 SkipCleanLogs=False
- Hid the discovery autostart checkbox on the admin-system page and removed the persistent property on

the livewire discovery.

  - Moved that functionality to an advanced option since it is not a recommended setting.
  - To enable the autostart of discovery change the nop to init for discovery in the advanced options.
- Fixed an issue with the Online property that was causing the false state to get raised repeatedly during reconnect attempts.

Version 1.3.13.18 04/30/2018

- Fixed an issue with users where save could be clicked before the API tree was filled and populated.
- Fixed an issue with users where the API tree did not automatically display for general users if only root level objects were enabled.
- Removed the LogAccessViolations advanced option briefly added in 1.3.13.17.
- Moved log access violations, login successes, and login failures to log tree options under the messaging branch.
- Made some small performance improvements.

Version 1.3.13.17 04/24/2018

- Updated one log message due to a structure that is marked obsolete in nlog4.
- Removed NLog 2 from the project library now that we have moved to nlog 4.
- Added code to retain the original date/time stamps when rotating log files so that they retain original data rather than the time of the rotation.
- Fixed an issue where the log list was not showing the correct date time stamp. They should have been showing last modified not last changed.
- Fixed bugs with user editing.
  - Editing and then saving a general user without expanding the sub branches of the api tree that had custom options set would cause the custom options to be lost.
  - Fixed an issue where canceling an edit user and then moving to a different user might not display the correct API data.
  - Canceling an edit now causes a page reload.
  - This version also expands all sub branches that have been set to anything other than inherit when you edit the user.

Version 1.3.13.16 04/23/2018

- Non Public - Internal Release Only
- Preview of Html Panels for internal review only.
- Added a way to get data with a similar callback as json but in multiple messages instead of json to core connector.
- Changed the routers loading code for better performance.
  -

  Removed a bug that was causing unneeded lookups inside a loop.
  - May need to refresh the page code with Ctrl-Shift plus refresh on the display page for the router to make sure the new javascript is loaded.
- Moved router callbacks back to messages rather than a single json blob in order to prevent possible out of memory issues with large routers.
- Fixed issues where dns.gethostname might not return a response.
- Optimized local ip address lookup code.
- Fixed a bug where editing a virtual IO that had no base points would present an empty base point in the javascript UI.
- Fixed a bug in LwcpClient and SapClient regarding selecting incoming only for message logging.
- Made an optimization such that if Sapv2 messaging is enabled the message is only built if enabled on at least one logger and then is remembered throughout the rest of that loop.
- Added the Skip Web Client SapV2 option to log writers.
  - If checked and SapV2 external messages are enabled, this will not log messages from the web clients.
- Added an option for logging user rights access violations to the sai-supervisor.nlog log.
  - This is in the advanced options and requires the value SET Users#0 LogAccessViolations=True.
  - Also added LogRotator settings to the options default file.
  - Note that turning on LogAccessViolations has the possibility of generating large logs at will log any branch that is requested but rejected due to access restrictions.
  - In max_depth message situations this could generate lots of log messages.
  - This is recommended to turn on for debugging and back off otherwise.
  - The parameter may be adjusted via the Users#0 Sap path as well for temporary setting and unsetting.
- Added information to the log message about the source of the access voilation - what method sourced it.
- Modified the cleanlogsforsize script to not execute the clean if the /srv/pathfindercore/skipcleanlogs.txt file exists and contains skipcleanlogs=true.
  - Modified the factory default script to delete /srv/pathfindercore/skipcleanlogs.txt if it exists.
- Fixed an issue with subscription accessviolation logs that was generating access violations on any message the user did not have access to whether or not a subscription existed.
- Added a network icon to show connection state.
  - Hooked that to the socket connect and close to show whether the web page socket is connected to Core PRO.
  - Added an error alert message if sending a change message (init, set, del) when the socket is closed.
- Updated the baseinstall script to include python-numpy which is supposed to have some performance benefits with websockify.

- Updated the OS to use mono 5.10.
- Changed nlog from version 2.2 to version 4.5. Still needs testing. 2.2 was broken with mono 5.10.
- Failed login attempts will now get logged to sai-supervisor.nlog.
-

- New Advanced options (may not display in an upgrade but may be added):
  - SET Users#0 LogAccessViolations=False
  - Ignore option above - removed in the next build
  - SET Logs#0.LogRotator#0.RotateRule#0 MaxFileSize=1
  - SET Logs#0.LogRotator#0.RotateRule#0 MaxCount=3
  - SET Logs#0 SkipCleanLogs=False
- LogAccessViolations options will log user access violations to sai-supervisor.nlog.
  - Note this can generate a lot of data as it will log any message from a sub, get, etc. that is restricted due to user rights.
  - It is recommended this gets turned on for debugging and turned off again for production.
- MaxFileSize will set the maximum file size in MB for each log before it gets rotated.
  - Please do not alter log rotation and max size settings unless you are working in a vm with plenty of disk space.
  - There is some slop on either side of this as it is analyzed periodically rather than on each write.
  - If SkipCleanLogs is false there will still be a 100MB restriction on the entire logs folder and forceful cleaning to enforce that.
- MaxCount will set the maximum number of rotatable logs for each log type.
- SkipCleanLogs will allow you to disable the cron job that clens up log files by forceful deletion to attempt to keep the log folder < 100MB.
  - This option should never be set to True for cf cards or installs with limited log space.

Version 1.3.13.15 04/04/2018

- Non Public - Internal Release Only
- Preview of Html Panels for internal review only.
- Fixed a bug with the htmlpanels button pull for hardware mapping.
- NAB 2018 show version.

Version 1.3.13.14 04/03/2018

- Non Public - Internal Release Only
- Fixed a bug with the getversion script that was always returning 1.0.0.00 rather than the correct version.
  - This was affecting previous attempts to reduce the need for clearing the browser cache.
  - Forcefully reloading pages after the update may still be necessary as browser cache invalidation is still a work in progress.
- Preview of Html Panels for internal review only.

Version 1.3.13.13 03/26/2018

- (From 1.2.13.00) Fixed an issue with DayOfWeek timers during the Daylight Savings Time Shift.
  -

  - In some cases it was causing an endless loop for an hour after each DayOfWeek timer scheduled for the day the time springs forward.
  - This was also causing the watchdog to reboot the system periodically until the hour after the timer should have fired had passed.
- (From 1.2.13.00) Fixed a bug that was causing the cluster backup generator to sometimes hang and then fail in the backup generation.
- (From 1.2.13.00) Fixed a bug where cluster backups in a plus GMT time zone could not be deleted due to the plus in the filename.
  - In this version plus is replaced with underscore in the generated name.

Version 1.3.12.13 03/04/2018

- Made Object Translators able to have multiple matching inputs so that a single input could trigger multiple property changes on an output object.
  - Review the beta documentation for details.

Version 1.3.12.12 02/26/2018

- (From 1.2.12.00) Fixes an issue with passwords with certain characters.
  - Due to a malformed regex some passwords would not get written properly from the web page password editing to the back end.
  - When setting the Admin password this could cause a lockout of admin capabilities forcing a call to support.
  - Characters that could cause problems prior to this update include: \^$.|?*.
  - While this version fixes the problem, you may have to forcefully reload the web page to make sure the new code versus cached code is used.
  - In chrome while on the user page, hold Ctrl and Shift while clicking refresh to reload the new JavaScript files.
  - It is also not a bad idea to have a second Admin user that works before changing the password of the primary admin user. That will allow access if the password change is problematic.
- (From 1.2.12.00) Fixed an issue with Classic Ids being incorrect between two nodes of a cluster when a device was added after the cluster was formed.
  - This could cause issues with user panel controls which would work when connected to one server and not the other.
  - While this fixes the bug that was causing the classic id mismatches, it is recommended to do a manual sync to force synchronicity if you find this to be a problem in your system.
  - To do a manual sync go to the secondary server and click the manual sync button.
  - This will request a special backup from the primary, pull it over to the secondary, and execute a restore on the secondary.
- (From 1.2.12.00) Added warning messages if the file selected for a bank update does not appear to be named correctly for the deployment target.

- Some of the logic flow changes in 1.3.11.11 were not working completely as expected.

  - Some change messages that did not execute a change were not being passed through as intended.
  - Logic Inputs were not being cleared of their init state during disable/enable.
  - Enabling a disabled flow was sometimes causing things to execute after a relay combiner that should not have.
  - Please note that Logic flow changes from 1.3.11.11 and 1.3.12.12 could cause changes to how logic flows function. If problems arise please revert and report the flow to support.
  - Review beta documentation for details on the logic flow functionality changes in 1.3.11.11.
- Added a shutdown menu item to the console menu and R2 crystalfontz display.
- Added code for proper licensing links with vm.

Version 1.3.11.11 02/20/2018

- Fixed an issue where in some cases extending a logic flow would not pick up the initial state immediately until sourcing translators changed.
- Fixed a bug that was not including the id in the set message for translators when the SkipInitialGet option is enabled to the backing storage.
- Worked on an improved implementation of the SkipInitialGet option.
- Fixed an issue where the SkipInitialGet request did not work if there was another flow that had the same start point which did not have this option engaged.
- Updated copyright on the web page to 2018

Version 1.3.11.10 02/16/2018

- (from 1.2.11.10) Fixed an issue with log writers that were tcp listeners.
  - Changing the log parameters was sometimes causing an application crash due to a socket exception.
- (from 1.2.11.10) Updated Client/Mini links to 5.81.
  - Made meterfaders send -3276.8 to console fader when below -79.0 to force the Fader_State=DOWN value.
  - Fixes an issue where meterfaders were not populating controls properly if pointed at fusion vmixers or console channels.
  - Fixes an issue where meterfaders were not controlling console gain faders correctly.
- Fixed an issue where Lwcp Mon commands were being sent to the equipment as mon#0 instead of mon.
  - This was not causing any issues because the equipment does not currently allow get or set with the mon object.
- Changed logic flow translators and combiners to execute on inbound messages whether or not a change takes place.
  - This is potentially a breaking change but makes certain kinds of flows work more intuitively.
  - See beta documentation on this change.

  - If you have problems with certain flows after moving to this version please revert and then report the flows to us so we can evaluate.
- Consoles now support a subscribe message for fader gain states reducing the ability to poll.
  - This new feature has been implemented in this version so that console fader gain changes are now subscribed to.
  - This allows for faster realization of these changes when they happen on the physical console rather than having to wait for a periodic poll response.

Version 1.3.10.09 02/12/2018

- Added the swap value to properties that are binary (two state) in nature.
  - This allows for much simpler latching logic flows.
  - For details see the section on 1.3.10.09 in the Beta feature documentation.

Version 1.3.10.08 02/08/2018

- Fixed potential issues with item deletes and the possibility of the web page refreshing before the message was sent.
- Fixed a problem where the base io editing on routers would not work correctly if numbering was off.
- Fixed an issue where the page could be reset after editing base ios before all messages got sent to the server.

Version 1.3.9.07 02/06/2018

- Fixed an issue where legacy panels could endlessly causing excessive cpu load trying to rectify classic ids if the classic id does not exist.
- Fixed a bug with virtual IO base point editing that was preventing the removal or reordering of base points in a virtual io.
- Fixed a bug that was causing virtual base IO reordering and sometimes deleting of base io points not to work correctly.
  - In some cases the final rectification was incorrectly ordering the base ios.
  - The divergence point in the base point analysis was generating a delete message based on the path including the mount point instead of just the virtual base io path.
  - Note: Base IO reordering still may not be entirely correct. Still testing and tweaking the code.
- Changed client link to 5.78 which disables/hides the Scheduling button which is non-functional in CorePro.
- Changed client application links to point to PathfinderPc.com instead of internally to save some space on the cf card.

Version 1.3.7.05 01/19/2018

- Updated the operating system and kernel to newer versions including mono from 4.2 to 5.4.

Version 1.3.7.04 01/19/2018

- Fixed an issue which could cause memory leaks when running on a processor other than the fanless engine I5 processor.

> ⓘ Important Note:
> The first startup after bank updating beginning with this version may take 10 to 15 seconds longer as a new file is compiled and generated. Subsequent startups will return to normal.

Version 1.3.7.03 12/14/2017

- Based on 1.3.6.02 but with the patches from 1.2.7.00 plus a few new features.
- From 1.2.7.00 patches: Added scrolling on logic flow folders for long lists of folders in the web UI.
- From 1.2.7.00 patches: Added some code to force ownership and securities on network config files during a restore operation including an update restore.
  - Attempts to fix a reported issue where occasionally the office network cannot be changed from the web page.
- From 1.2.7.00 patches: Fixed an issue where double clicking on an endpoint would select the correct property but sometimes would display the incorrect property description.
- From 1.2.7.00 patches: Fixed a bug where deleting combiner inputs was not always causing the combiner to get rewritten to backing storage.
  - As a result the deleted passiveinput ids would load after a restart even though no translator was hooked to them.
  - This could cause unassigned states on those combiner inputs and prevent the combiner from analyzing properly after the reboot.
  - This fix will prevent this from happening with future flows.
  - If you have a flow in this state it is easiest just to recreate it.
  - If the flow is particularly complex to recreate, contact support as there are ways using the API to remove the extra combiner inputs.
- From 1.2.7.00 patches: Added a hover balloon over combiners to show the passive input count.
  - Used to determine a discrepancy in the input count.
  - This will allow you to determine if a combiner is in the state mentioned above where it has more inputs than translators tied to its inputs.
- Fixed a typo in the web page with the clear options for the delay combiner.
- Added an option to use <%DateTime%> in email messages and subject lines to insert Date and Time into the message.

Version 1.3.6.02 12/06/2017

- Based on 1.3.5.01 but with the patches from 1.2.6.00 plus a few new features.
- From 1.2.6.00 patches: Fixed a bug that was preventing the creation of String Builder memory slots. This was broken as of 1.0.0.36 when startup states were introduced.
- From 1.2.6.00 patches: Fixed a bug that was causing Virtual Routers not to show the Other source state in many situations.
- Changed email host code such that a blank user name will cause the email to be sent without any credentials. The host would need to support non-credentialed emails.
- Added a pair of properties to DelayCombiners to allow for a momentary output value:
    - Clear output after countdown completes: When the countdown completes the correct value is passed to the output and then it is set to the clear value afterwards.
    - Clear Value: The value to reset the output to after passing the delayed value through.Version 1.3.7.03 12/14/2017

Version 1.3.5.01 Beta 11/29/2017

- First beta version with new features since 1.2.0.00.
- Based on 1.2.5.00 patch status.
- Added a link on the system page to documentation on the new beta features until they are folded into a new revision of the manual when beta moves to release.
    - Use this link to gain more information about using the new beta features.
- Added a write only property to interval timers called reset.
    - When a timer is running this will restart the countdown.
    - If it is not running this will set elapsed to false.
- Added a write only property to interval timers called ResetStart.
    - When a timer is running this will restart the countdown.
    - If it is not running this start it.
- Added a write only property to interval timers called ResetStop.
    - When a timer is running this will stop the timer.
    - If it is not running this will set elapsed to false.
- Added copy, cut, and paste to the Logic Flow views for copying, cutting, and pasting entire views.
    - Select a view to copy and click the copy icon.
    - Select a new parent and click the past icon.
    - The system will ask for a new name for the view to be pasted.
    - Paste will always paste flows in a disabled state assuming you will want to modify the settings after the copy and paste.
    - If a view has sub views those will also be copied.
    - You cannot paste a view into itself.
    - As this is a beta feature please take regular backups while beta testing until this feature is proven.

- Double clicking a combiner will no longer rotate through combiner options. Instead it will open a combiner editing dialog.
  - This allows you to select the desired combiner type directly.
  - Additionally new variations of combiners require configuration parameters and so this dialog is also necessary for setting those parameters.
- Added an Equality combiner.
  - This combiner accepts multiple inputs and yields True or False depending on whether the inputs are the same.
  - A case sensitive or insensitive option is also provided in the dialog when an Equality combiner is selected.
- Added a Delay combiner.
  - This combiner allows you to introduce delay into a flow without needing to use an interval timer.
  - Delay Combiners only accept a single input.
  - Interval timers should still be used when multiple different flows need to interact with the interval timer.
  - The Delay Combiner includes several configuration parameters:
    - Delay Time: The time of the delay in milliseconds.
    - Reset delay if input changes: If checked any time the input to the combiner changes the delay is reset rather than continuing its countdown.
    - Output Value: Select whether the delay combiner should output the input value from the start of the delay countdown or the input value at the end of the countdown once the countdown is complete.
    - Cancel Value: A value that if seen at the Delay input will cancel the delay countdown.
- More detailed documentation of the changes may be found at: betamanualchanges1-3.pdf

1.2.xx.xx

Version 1.2.13.00 03/26/2018

- Fixed an issue with DayOfWeek timers during the Daylight Savings Time Shift.
  - In some cases it was causing an endless loop for an hour after each DayOfWeek timer scheduled for the day the time springs forward.
  - This was also causing the watchdog to reboot the system periodically until the hour after the timer should have fired had passed.
- Fixed a bug that was causing the cluster backup generator to sometimes hang and then fail in the backup generation.
- Fixed a bug where cluster backups in a plus GMT timezone could not be deleted due to the plus in the filename.
  - In this version plus is replaced with underscore in the generated name.

Version 1.2.12.00 02/26/2018

- Fixes an issue with passwords with certain characters.
  - Due to a malformed regex some passwords would not get written properly from the web page password editing to the back end.
  - When setting the Admin password this could cause a lockout of admin capabilities forcing a call to support.
  - Characters that could cause problems prior to this update include: \^$.|?*.
  - While this version fixes the problem, you may have to forcefully reload the web page to make sure the new code versus cached code is used.
  - In chrome while on the user page, hold Ctrl and Shift while clicking refresh to reload the new JavaScript files.
  - It is also not a bad idea to have a second Admin user that works before changing the password of the primary admin user. That will allow access if the password change is problematic.
- Fixed an issue with Classic Ids being incorrect between two nodes of a cluster when a device was added after the cluster was formed.
  - This could cause issues with user panel controls which would work when connected to one server and not the other.
  - While this fixes the bug that was causing the classic id mismatches, it is recommended to do a manual sync to force synchronicity if you find this to be a problem in your system.

  - To do a manual sync go to the secondary server and click the manual sync button.
  - This will request a special backup from the primary, pull it over to the secondary, and exeucte a restore on the secondary.
- Added warning messages if the file selected for a bank update does not appear to be named correctly for the deployment target.

Version 1.2.11.00 02/16/2018

- Fixed an issue with log writers that were tcp listeners.
  - Changing the log parameters was sometimes causing an application crash due to a socket exception.
- Updated Client/Mini links to 5.81.
  - Made meterfaders send -3276.8 to console fader when below -79.0 to force the Fader_State=DOWN value.
  - Fixes an issue where meterfaders were not populating controls properly if pointed at fusion vmixers or console channels.
  - Fixes an issue where meterfaders were not controlling console gain faders correctly.

Client/Mini 5-81 02/16/2018

ⓘ Important Notes:
Updates to the Pathfinder Client and Mini for Core only.
This Client update can be used with Core PRO software 1.2.2.00 or later

- Made meterfaders send -3276.8 to console fader when below -79.0 to force the Fader_State=DOWN value.

Client/Mini 5-80 02/15/2018

ⓘ Important Notes:
Updates to the Pathfinder Client and Mini for Core only.

This Client update can be used with Core PRO software 1.2.2.00 or later

- Fixes an issue where meterfaders were not populating controls properly if pointed at fusion vmixers or console channels.
- Fixes an issue where meterfaders were not controlling console gain faders correctly.

Version 1.2.10.00 02/08/2018

- Fixed potential issues with item deletes and the possibility of the web page refreshing before the message was sent.
- Fixed a problem where the base io editing on routers would not work correctly if numbering was off.
- Fixed an issue where the page could be reset after editing base ios before all messages got sent to the server.

Version 1.2.9.00 02/06/2018

- Fixed an issue where legacy panels could endlessly causing excessive cpu load trying to rectify classic ids if the classic id does not exist.
- Incorporated scripting changes related to the dynamic construction of a core dll from 1.3.7.04 as they apply to 1.2.xx as well.
- Fixed a bug with virtual IO base point editing that was preventing the removal or reordering of base points in a virtual io.
- Fixed a bug that was causing virtual base IO reordering and sometimes deleting of base io points not to work correctly.
  - In some cases the final rectification was incorrectly ordering the base ios.
  - The divergence point in the base point analysis was generating a delete message based on the path including the mount point instead of just the virtual base io path.
  - Note: Base IO reordering still may not be entirely correct. Still testing and tweaking the code.
- Changed client link to 5.78 which disables/hides the Scheduling button which is non-functional in

CorePro.

- Changed client application links to point to PathfinderPc.com instead of internally to save some space on the cf card.

Version 1.2.7.00 12/14/2017

- Based on 1.2.6.00
- Added scrolling on logic flow folders for long lists of folders in the web UI.
- Added some code to force ownership and securities on network config files during a restore operation including an update restore.
  - Attempts to fix a reported issue where occasionally the office network cannot be changed from the web page.
- Fixed an issue where double clicking on an endpoint would select the correct property but sometimes would display the incorrect property description.
- Fixed a bug where deleting combiner inputs was not always causing the combiner to get rewritten to backing storage.
  - As a result the deleted passiveinput ids would load after a restart even though no translator was hooked to them.
  - This could cause unassigned states on those combiner inputs and prevent the combiner from analyzing properly after the reboot.
  - This fix will prevent this from happening with future flows.
  - If you have a flow in this state it is easiest just to recreate it.
  - If the flow is particularly complex to recreate, contact support as there are ways using the API to remove the extra combiner inputs.
- Added a hover balloon over combiners to show the passive input count.
  - Used to determine a discrepancy in the input count.
  - This will allow you to determine if a combiner is in the state mentioned above where it has more inputs than translators tied to its inputs.

Version 1.2.6.00 12/06/2017

- Based on 1.2.5.00
- Fixed a bug that was preventing the creation of String Builder memory slots. This was broken as of 1.0.0.36 when startup states were introduced.
- Fixed a bug that was causing Virtual Routers not to show the Other source state in many situations.

Version 1.2.5.00 11/28/2017

- Adjusted the settime parameter when setting the time from local pc to enforce more complete and proper formatting between the JavaScript browser and PathfinderCore PRO.
-

Modified the backup script to include the DNS settings file.

  - Previous to this change DNS settings were being lost during firmware updates.
  - Note: the update uses the existing backup script to do the update so DNS will have to be reset after this update and should then be correct for future updates.
- Refactored some code in LogicFlowManager which theoretically might leave objects in ram that had been deleted when a complete folder was deleted.
- Fixed a bug in Client and Mini that was causing frozen meters when audio cut out in some situations where decay rate was used.
  - This fix requires installing the 5.77 version of mini and client.

Version 1.2.4.00 11/13/2017

- Fixed a bug introduced in 1.2.1.00 that in certain rare cases could cause logging to create a deadlock on the SapV2 connection.

Version 1.2.3.00 11/09/2017

- Fixed a bug that was causing Logic Flow views to leave residual data in the backing storage after more than 1 rename of a view.
- Changed the audio alarms web page data table to make the save state option enabled.
- Changed the scenes web page data table to have filtering enabled.

Version 1.2.2.00 11/03/2017

> Important Notes:
> You may need to hold cntrl-shift and click refresh to refresh the logic flows web page the first time after updating to ensure you are not using cached javascript.
>
> Be sure to update client applications (client, mini, panel designer) to the versions in this software package.

- Fixed an issue with day of week timers in a cluster that was causing them not to fire on subsequent days in some situations.
  - The elapsed property on a Day of week event when in a cluster was not resetting back to false such that the next day event would not happen.
  - The elapsed property should now flip back to false 10 seconds after it flips to true.

Version 1.2.1.00 11/02/2017

> Important Notes:
> You may need to hold cntrl-shift and click refresh to refresh the logic flows web page the first time after updating to ensure you are not using cached javascript.

> Be sure to update client applications (client, mini, panel designer) to the versions in this software package.

- Fixed inconsistencies between lists on the web pages as far as pagination controls and count fields.
- Limited send buffer size of network loggers.
- Fixed a file access problem with log file writers after changing the write fields.
- Fixed a bug that was sometimes causing day/time and day of week timers not to execute when in clustering after a restart.
- Fixed a bug where a collection in log files was growing with repeat instances of file information.
- Removed some DateTimeOffset calls that were unnecessary.
- Cleaned up some string processing for potentially faster results.
- Added code to not send poll messages if send buffer already has things waiting in the queue.
    - This prevents the queue from increasing and causing a memory leak.
- Discovered that meters were defaulting to an incorrect polling rate and not respecting the request of the clients.
    - This in some cases was causing the memory to leak.
- Removed a LastContacted datetimeoffset that was getting set with every set of bytes received as it was degrading performance.
- Fixed an issue with send buffers that theoretically could allow dequeued items to still consume memory.

Version 1.2.0.00 10/20/2017

> Important Notes:
> You may need to hold cntrl-shift and click refresh to refresh the logic flows web page the first time after updating to ensure you are not using cached javascript.
>
> Be sure to update client applications (client, mini, panel designer) to the versions in this software package.

- Updated the manual to version 1.2.
- Fixed a bug that could cause client silence bubbles to appear on a node's alarm web page after a restart when in a cluster.
- Updated the bank description.
- Fixed an issue where if both nodes have the same office ip address, cluster reconnects could end up causing a node to connect to itself.
    - Now the system will not try to connect to a remote cluster node on a local ip address.
- Fixed issues where event system state changes were not being raised to subscriptions in all cases.
    - This meant sometimes you had to refresh the clustering web page to see the actual event system state.
- Fixed an issue where leaving a cluster would delete the ClusterAdmin user from all other nodes in the

cluster as well.

1.0.xx.xx

Version 1.0.0.42 Beta 10/11/2017

> ⓘ Important Note: You may need to hold cntrl-shift and click refresh to refresh the logic flows web page the first time after updating to ensure you are not using cached JavaScript.

- Fixed a bug where object translators were sending an incorrect message for obtaining the initial state of the object.
  - This was affecting the initial state setting of color and text on hardware maps.

Version 1.0.0.41 Beta 10/10/2017

> ⓘ Important Note: You may need to hold cntrl-shift and click refresh to refresh the logic flows web page the first time after updating to ensure you are not using cached JavaScript.

- Fixed a bug with logic flow folder/view renaming that was causing garbage folders with identical object ids to be left behind - since 1.0.0.40.
- Fixed a bug where factory default was not properly wiping out the new LogicFlows folder - since 1.0.0.40.
- Fixed a bug in the UI code that was creating a garbage combiner with an undefined operator when deleting a combiner.
- Fixed a bug in the logic flows color selections where when selecting translations that had colors, the color selectors were not updating with the color.
- Fixed an issue with the Alarm web page where editing an existing alarm would not show the name of the io in the io field.
  - It was showing the address instead. Now it will switch to the name once the name is resolved.

Version 1.0.0.40 Beta 09/22/2017

> ⓘ Warnings:
> This version is a rollup of unreleased changes from builds 1.0.0.36 to 1.0.0.39.
>
> Please review the release notes for 1.0.0.36 through 1.0.0.40 to fully understand the changes.
>
> 1.0.0.36 through 1.0.0.39 have significant changes in Client, Mini, and Panel Designer as far as connection recycling and connection cpu load.
>
> Client, Mini, and PanelDesinger must be re-installed as well to recognize these benefits.

- Fixed an issue where Day Of Week time events were not shifting or not shifting properly over daylight savings time shifts even after the changes in 1.0.0.36.

Version 1.0.0.39 Beta 09/21/2017

> Warnings:
> Version 1.0.0.36 and later have major changes as well as some important bug fixes. Please read the release notes for more details. Please report any issues as soon as possible.
>
> 1.0.0.36 through 1.0.0.39 have significant changes in Client, Mini, and Panel Designer as far as connection recycling and connection cpu load.
>
> Client, Mini, and PanelDesinger must be re-installed as well to recognize these benefits.

- Fixed a bug where a blank logicflows.txt file would cause the application to crash trying to update to the new logic flows storage structure introduced in 1.0.0.36.
- Fixed an issue introduced in 1.0.0.36 where DateTime events could not be inited for the future.
- Fixed an issue where the new clock/fixed options introduced in 1.0.0.36 for day of week and date time events was not being loaded from the database properly after a restart.
- Fixed a bug where time manager date/times were being forced back to within 100 seconds of now on load from database. This bug introduced in 1.0.0.36.
- Hid the event system message on the logic flows page until a valid response is returned. This prevents stand alone instances from showing an unknown state.
- Changed max time differential to be 10 minutes in future instead of 100 seconds.
  - Cluster nodes need to be within 10 minutes of each other and ideally much closer.

Version 1.0.0.38 Beta 09/18/2017

> Warnings:
> Version 1.0.0.36 and later have major changes as well as some important bug fixes. Please read the release notes for more details. Please report any issues as soon as possible.
>
> 1.0.0.36 through 1.0.0.38 have significant changes in Client, Mini, and Panel Designer as far as connection recycling and connection cpu load.
>
> Client, Mini, and PanelDesinger must be re-installed as well to recognize these benefits.

- Fixed an issue with cluster joining. The ClusterAdmin password was not getting set properly.
- Added an optional file for PathfinderPC Client at C:\ProgramData\PathfinderPC\fixedips.txt.
  - If this file exists PathfinderPC client will use it for the ip address list to attempt instead of what is reported by clustering.
  -

  - This allows you to lock client to specific PathfinderCore PRO instance if you want.

- Fixed a bug with Panels that causes panels not to display or become confused in Panel Designer, mini, and client if the panel names are continuations of other panel names.
  - For example Blah and Blah2. Often one of these would not display in designer and editing could overwrite the wrong one.
- Fixed an issue with the new logic flow backing storage changes in 1.0.0.36/37 that might cause flow objects not to load due to differences in linux line endings in the file.
- Compiled PathfinderPC Client as 5.76.
- Fixed an issue where the new internal check in clustering was not clearing on a loss and reconnect sometimes leaving the cluster resynchronization process in a hung state.
- Fixed an issue with the new memory slot startup state option. It was not changeable from the API or UI except at creation.

Version 1.0.0.37 Beta 09/15/2017

> Warnings:
> Version 1.0.0.36 and later have major changes as well as some important bug fixes. Please read the release notes for more details. Please report any issues as soon as possible.
>
> 1.0.0.36 through 1.0.0.37 have significant changes in Client, Mini, and Panel Designer as far as connection recycling and connection cpu load.
>
> Client, Mini, and PanelDesinger must be re-installed as well to recognize these benefits.

- Fixed a bug with incorrect logic gate results for inputs greater than two for NOR, NAND, XOR, XNOR gates.

Version 1.0.0.36 Beta 09/14/2017

> Warnings:
> This version has some major changes as well as some important bug fixes. Please read the release notes for more details. Please report any issues as soon as possible.
>
> This version has significant changes in Client, Mini, and Panel Designer as far as connection recycling and connection cpu load.
>
> Client, Mini, and PanelDesinger must be re-installed as well to recognize these benefits.

- Fixed a critical bug with date/time and day of week timer. Events were stored with the UTC offset.
  - This means that after a daylight savings time update, the event would execute at the requested time under the previous UTC offset which in many cases would be wrong.
  - There is now an option to select whether a date/time or day of week timer is tied to clock time or fixed time.
  - By default all old timers will be tied to clock time.

  - Clock time means that an event will continue to execute at a specific time according to the clock before and after a time zone offset from utc shift.
  - Fixed time is useful in situations like satellite feeds where the sourcing program's time zone does not shift.
  - In that case after a daylight savings time shift the event should happen at a different clock time.
  - This is a critical fix for anyone using this feature and needs to be updated prior to the Daylight savings time shift.
- Completely changed how logic flows are stored in the backing storage.
  - Originally logic flows were stored in a single file of SapV2 init messages used to recreate all of the flows on startup.
  - There is the potential if corruption, errors, or a crash were to happen while the file was being written for the loss of many flows.
  - This version stores flows to a folder architecture with a single one or two line sapV2 init file for each combiner or translator.
  - This means that edits and/or changes will only affect the translator and combiner files being edited.
  - After migrating to this version on startup the old flow file will be read and rewritten to the new format.
  - The file will then be marked as already converted but left in place for safety.
  - This has important ramifications for rolling back to a previous version.
  - New flows created after the conversion will not appear after a software rollback.
  - This is a pretty major change so report any issues to Axia support.
- Updated factory default and restore scripts to delete the new LogicFlows folder during the factory default or as part of the restore process.
- Fixed a bug introduced in 1.0.0.34 with obtaining the initial Vmixer states.
  - Adding IN_SELECT to vmixers meant that the initial state query message we were sending included the IN_SELECT property.
  - In Engine versions prior to 3.1.5, IN_SELECT does not return a response.
  - This meant the entire initial VMIX IN state query was failing and we were not obtaining the VMIX state on startup.
  - Only subsequent changes would be seen by the flows.
  - This version does not request the IN_SElECT state if ENGINE/PowerStation version is prior to 3.1.5.
  - Currently we do not query the initial IN_SELECT state of Iports as the query state has not been ported to an Iport version to our knowledge yet.
- Fixed bugs with Logic Flows that were allowing the flows to run on the secondary server once local sync had happened but before sync was complete with the remote device.
  - Now attempts must be completed to all devices before logic flow states will update and initial gets are sent.
  - This was causing flows to occasionally run on secondary during startup when primary was already executing them.
- Added an option for the state of a memory slot on startup. You can now select whether the memory slot

will start blank, with a fixed value, or with the last known state.

  - It is important to understand that selecting the last known state will mean writes to the backing storage for each change of the slot.
  - For cf card based PathfinderCore PRO systems this can be intensive and slow so select the memory slots you need to retain state between restarts carefully.
- Added an additional clustering cross check step to double check date/time stamps on internal objects with no remote references such as memory slots, timers, and panel items.
- Changed cluster web ui page to submit additional properties in the ClusterAdmin init message to force Admin state for the ClusterAdmin user when it is created. Broken as of .35.
  - This was causing clustering join problems in 1.0.0.35.
- Changed the last update properties related to clustering to force them to be no later than 100 seconds from now.
- Forced all DateTimeOffset to string conversions to go through the same method to maintain formatting consistency.
- Fixed a bug with panel designer as it was not outputting version stamps in the correct format. Added code to force versions to the correct format upon ingest.
- Worked on database auto fixing of dates more than 100seconds in the future other than Timer based DateTime events.
- Fixed a bug in logicflows UI. Display was showing partial flow elements from folders 1 to 2 levels down from the selected folder even if those lower folders were not selected.
  - This was causing odd partial looking flows to be displayed.
- Rewrote lwcp/sap message parsing to greatly reduce string copying when dealing with large messages such as the contents of a large panel file embedded in the message.
  - Huge performance difference for messages like this.
- Migrated the new message parsing changes to the javascript code as well.
- Added the $EXCLUDE_PROPS system item. Using this item means the properties in the get message should be the ones excluded from the response messages.
- Reduced the panel xml returned during mini startup.
- Used a background task to obtain panel xml data.
- Removed the currentpanelxml from get requests in Client and Mini. This was causing unnecessary noise and processor load.
- Added code to only send differences in properties between initial xml load and current on startup for panels.
  - This should vastly reduce initial panel property push.
- Changed the ContainsPath messages to a full router sync for io sync with panels and metering to prevent multiple loops through the router on the server side hunting for classic ids.
- Removed the syncio recursive contains_prop lookup with client and mini.
- Revised the client and mini reconnect code significantly.
- Spread out the load of uninitialized combiner input timeout requests over a period of time to reduce the initial processor spike.
- Changed the dispose code for pcp objects (client and mini) to improve client shutdowns.

- Adding debug messages for debugging connection recycling
- Greatly improved client and mini connection recycling issues when the client loses connection with the server.
  - Previously in certain situations the client/mini might endlessly recycle connection attempts sometimes with large data dumps from Core PRO on each attempt.
- Randomized the client connection order a bit by randomly selecting whether to go through the connection list in ip address order or reverse ipaddress order.
- Added a force option for manual selection of ipaddress.
- Compiled the new clients as 5.74.

- Added a ping to PCP to test responsiveness and recycle connection if unresponsive.
- Fixed a bug where Routers and UserPanels menu items could not be disabled in the user editing.
- Also attempted to add some css coloring to hidden versus visible but most browsers are not displaying it.

Version 1.0.0.35 Beta 08/08/2017

> ⓘ Important Notes:
> User security changes caused large amounts of code to be modified in 1.0.0.34 and this version. Please report any problems.
>
> This version fixes bugs with the new user security features found during in-house testing of 1.0.0.34.
>
> 1.0.0.34 was never publicly released but the feature set is described and applicable to this version so please review the 1.0.0.34 release note carefully.

- Fixed a variety of bugs with cluster synchronization of the new user features from 1.0.0.34.
- Fixed a bug with initial tree display with new users creates.
- Fixed a bug with changing api settings in the tree where previously selected but unchanged settings were not getting written out again.
- Made the password property in Sapv2 changeable with the set operator.
  - Previously it was read only and required recreating the user to change it.
  - If the password is not structured as an Apache hash, the property setter will detect that and hash it.
- Reworked the user editing web page to make it work more cleanly now that passwords and properties are more easily editable.

Version 1.0.0.34 Beta 08/03/2017

> ⓘ Important Note:
> User security changes caused large amounts of code to be modified in this version. Please report any problems.

- Fixed a bug where IN_SElECT was not appearing in the property tree for VMIX.
- Fixed a bug where IN_SELECT was read only for both VMIX and VMODE when it should have been Read/write.
- Fixed bugs with sap memory slot selection that got introduced when we moved to json blobs.
- Added some code to property trees to swap the loading to No Items Available if response comes back with None or there is no query message in the node.
- Removed icon for loading/no items in property tree.
- Added code to insert version number into JavaScript and css links to help make browsers recognize a software update and reload JavaScript rather than relying on the cached version.
- Made cfontz use the watchdog so if we do not ping the display every 5 seconds or so, it will reset.
- Added user level security.

> Warnings:
> It is critical that at least one Administrative user is maintained at all times in the system.
>
> Deleting all Administrative users may prevent you from accessing the system properly and will require a call to support and possibly a deletion of the user files to gain access again.

- You can now define users as type User in addition to type Admin.
- After selecting the user type (as opposed to the Admin type) two trees will appear for selecting user rights.
  - These trees are not present for Admin users because Admin users have access to everything.
- The Menus tree affects which menus will be available to the user in the web page's navigation bar.
  - When these options are modified the system will create a custom navigation bar for the user.
- The allowed APIs list presents a list of common objects to which the user may be given one of 5 levels of access.
  - Inherit - The security is inherited from the parent branches of the tree. This is the default for most branches.
  - No Access - The user cannot see or use this object.
  - Display - The user can see the object and view the changes to properties on the object but cannot make any changes.
  - Change/Use - General properties of the object may have their values changed by the user, but creation and deletion are not allowed.
  - Full Access - User has full access rights to the object.
- By default sub objects inherit security from their parents.
- A good example to discuss is routers.
  - No Access - The router will not be visible to the user in either the web pages or Pathfinder Client.
  - Display - the user can open and view the router in both the client and web page, but they will not be

    able to make any route changes.

    - Change/Use - the user can use the router normally but will not be able to add or remove route points.
    - Full Access - the user has the rights to completely change the router including deletion and creation rights.
- The rights selected apply to the API (port 9600) and since both client and the web pages use the API they are affected accordingly.
- The system can actually allow security changes at any branch level within the API branches, but only the normally used ones are present in the UI tree.
- Currently there is no API view of the UI tree. This may be added in the future if it is deemed necessary.
- VmixControl access is currently a little tricky as it requires both display access to the engine device, full control of the VMIX Sub object, and change access to the audio router.
    - In the future we may simplify this and also make virtual routers work properly with the VMIX control application.
    - If routing is not necessary, display access to the router can be used.
- It is important to note that some web pages require Admin rights.
    - For example backup/restore, bank changes, update uploading, and log manipulation are only allowed for Admin users.
    - The licensing section of the System page is also hidden from non-admin users.
    - All configuration options on the System Configuration page are also disabled if you are not an Admin user.
    - Cluster web page buttons are also disabled if you are not an Admin user.
    - Remember these web pages can be hidden in the navigation bar using the menu tree of the user definition.
- Users and their securities should sync across the cluster is this version.
- Revised PathfinderClient_Core as version 5.72 to use new security settings.

Version 1.0.0.33 Beta 07/04/2017

- Fixed a bug with the Constructor property of Aes67 sources.
    - It was returning blank and therefore clustering was unable to reproduce new Aes67 ios on secondary nodes.
    - Please Note: A one time manual sync may be necessary to get Aes67 sources that existed prior to the upgrade from primary to secondary.
- Fixed a bug where the web page was not showing the updated host name after a route change with Aes67 sources.
    - Instead the host would get cleared until the web page was reloaded. This should be fixed.
- Fixed an issue where altering a sip based aes67 source would cause the livewire channel field to be populated with the description instead of 0 until the web page is refreshed.

Version 1.0.0.32 Beta 07/03/2017

- Fixed a bug where the generic client (tcp client) in device emulators was not automatically reconnecting.
- Fixed a bug where the probel general client (tcp client) in device emulators was not automatically reconnecting.
- Fixed a bug where the probel switcher client (tcp client) in device emulators was not automatically reconnecting.
- Got Aes67 editing and deletion working in the web pages.
- Made Aes67 sources show the corresponding livewire channel in the ui if it is multicast and is in the livewire multicast address range.
- Made Aes67 sources available for routing in the PathfinderClient application (version 5.71).
- Updated the system page download links for the new version of PathfinderPCClient_Core_5.71.
- If the description field of an Aes67 source ends with " ON devicename", then devicename will also be displayed in the hostname field.
  - For example: MySource ON MyDevice. The " ON " is case sensitive.
- Fixed a label in Aes67 source addition/editing window. In multicast selection option, host IP should have been labeled multicast ip.
- Added automatic backlight dimming to optional crystalfontz display.

Version 1.0.0.31 Beta 06/28/2017

- Added support for the CrystalFontz front panel display.
- Added a menu at the Linux console that provides equivalent functionality to the front panel. This is useful if the front panel is unavailable for any reason.
- Did build installer work for easier deployment in the production department.
- Added options to the mnt/conf/displays directory for appropriate display binary launching and configuration.
- Fixed a bug that reset messages were not pulling from ModAccess for modaccess in PowerStation.
- Fixed a bug that the autoreconnect property for devices was being presented as read only instead of read/write.
- Fixed an issue where IN_SELECT in VMODE was not fully populated with all viable options.
- Fixed an issue where IN_SELECT did not exist in VMIX and it should.

Version 1.0.0.30 Beta 06/08/2017

> ⓘ Important Note:
> This build has some significant changes to logic flows. Please report any problems.

- LwrpSendMessage and LwcpSendMessage were not appearing in the API logic flows tree. This is fixed.
- Added the ability to use escape character sequences in LwrpSendMessage and LwcpSendMessage

properties.

- Added code to clear LastOutput state of translators that should not execute on backup cluster node.
- In the logic flows web page there is now a message in the top right hand corner of the screen indicating whether the event system is active on the particular node.
- In the logic flows web page end points will display with the disabled color if the event system is not active.
- Fixed a number of places in the API tree where friendly names were not displaying properly and so lists of objects were indistinguishable from each other.
- Fixed a bug created by the changes in 1.0.0.29 which in some cases was causing long logic flow load times especially if start point objects did not exist in the system.
  - If the load time got too long it could cause the watchdog to kill and restart the supervisor.
  - Modified the changes to remove this long load time.
- Modified the code for logic flow initial get requests depending on the cluster system state.
- Removed xnode mixer branches from the simple tree for now.
  - Large numbers of xnodes was causing significant delay in the initial loading of the property list.
  - Working on a better solution to this in a later version.
  - In the mean time these branches are still available in the API tree under Devices/xnodedevice/lwrpinterpreter/lwrproot.

Version 1.0.0.29 Beta 06/05/2017

> ⓘ Important Note:
> This build has some significant changes to logic flows. Please report any problems.

- Fixed a bug. Updating a generic translator watcher value was not triggering a write to the back end storage.
  - This means changing a watcher value after it had been created might revert to the old value after a restart in previous versions.
- In Logic Flows stripped values from combiner inputs and translator start and endpoints when writing to and reading from back end storage.
- Added some code to prevent combiners from being analyzed until valid values from an initial get request to each input have been received and/or a timeout waiting for those inputs occurs.
- Modified the code to tune the initial get requests of logic flow states when a new flow gets loaded and/or the system starts.
- Fixed some bugs that were causing extra and incorrect messaging with relay combiners were tied to object translators.
- Fixed a bug where object translator property changes matching the root level of the object translator's object path were not being sent properly.
- Made internal translations work on the non-active cluster node. Only ones whose activeoutput is pointed externally will be blocked until the node becomes active.

  - This helps internal transitions to be in the correct state already during a switchover.
- Many of the above Logic Flow changes target some inconsistencies with logic flow states during system startup.
- In the translator dialog added the "Skip startup state request and wait for next change option."
  - If this is checked changes for the start point will be subscribed to, but an initial get will not be sent and changes will wait for the first property change thereafter.
- Made some of the logic flows modal dialogs unselectable to prevent drag selection from highlighting arbitrary sections of the dialog.
- Fixed a bug with loading previously discovered user modules.
  - It would only reload the first discovered module per ip address on a restart. This is fixed.
  - This was reported in relation to IP Tablet.
- Added options for Sound4 processors.
  - Note: Voco will still not be discovered properly due to an issue with the Voco Lwrp messaging.
  - This has been reported to the Sound4 team to address.
- Fixed a bug that was causing a recursive get request on the selected item in the API tree.
  - This was causing the UI to keep moving to the selected item when attempting to scroll and was also causing unnecessary messaging.

Version 1.0.0.28 Beta 05/26/2017

- Fixed a bug with licensing after a factory default.
  - Reading the unlicensed state could cause a crash.

Version 1.0.0.27 Beta 05/26/2017

> ⓘ Important Notes:
>
> Significant changes have been made to the web page messaging in logic flows and route lists.
>
> Please clear the browser cache and/or use the browser method to forcefully reload JavaScript files.
>
> For Chrome hold Ctrl+Shift and click the circular arrow refresh button next to the address bar on each relevant page.
>
> Failure to do this may cause the browser to attempt to use cached JavaScript which will not use the new changes and may or may not work properly.

- Fixed a bug with Gpio routing.
  - Gpio routes were not displaying as routes in the router if the name was embedded in the SRCA field from the equipment.
  - Since we are now sending the name as a part of the command this was breaking the display of gpio routes in many cases.
  -

  - Not sure in which version this bug was introduced.
- Fixed a bug introduced with the changes in 1.0.0.26 with logic flows and device emulators.
  - Device Emulators were not appearing in the logicflows tree because they were external and not being handled by the bulk json command.
- Fixed a bug that was preventing a double quote and a comma from coexisting in a flow title.
  - If you had both in a flow title, the quote was not being escaped properly and the constructor message was being dropped.

Version 1.0.0.26 Beta 05/22/2017

> Important Notes:
> Significant changes have been made to the web page messaging in logic flows and route lists.
>
> Please clear the browser cache and/or use the browser method to forcefully reload JavaScript files.
>
> For Chrome hold Ctrl+Shift and click the circular arrow refresh button next to the address bar on each relevant page.
>
> Failure to do this may cause the browser to attempt to use cached JavaScript which will not use the new changes and may or may not work properly.

- Revised logic flows web page extensively to use more optimized messaging for better performance.
- Added a timeout of 750ms when selecting flow folders so that you can select/deselect multiple folders without a redraw between each one.
- Revised router web pages to also optimize messaging for better performance.
- Fixed a bug from .24 or .25 which might break the io list loading in alarm web page and possibly fully break vmixcontrol web page.
- Added an error trap to handle io import failures.
- Fixed issues with lag when loading the initial large list. Recursive parsing as the large json list came in was causing issues
- Fixed an issue with parsing aes67 pathios.
- Finished the Aes67 io addition web pages.
- Refactored the Sap Message parsing in web pages for better performance and to make it closer to back end code.
- Fixed a bug with the asynchronous handling of inbound data in the web page sap message parsing.
- Fixed a bug in the logic flows tree that was causing sort to happen on branches over and over again as items were added when it only needed to be sorted once after the additions.
  - Modified the sorting algorithm to further reduce extra sort operations.
- Added button 3 to Lwch and fach faders for element/fusion.
  - These will only do anything on the revised European modules that have this button but there is currently no way to detect which is present.
  - In the future perhaps use the press of button 3 to detect and populate.

    - But for now it will be on all of these objects.
- Fixed a bug where object translators would continue to add available properties as you switched between different object translators instead of resetting the list to the new object.
- Fixed a bug where sometimes object translators would only present one property in the list until you closed and reopened the dialog.
- Completed the Virtual Router editing web pages.
- Modified the css with more apparent row selection colors for all lists.
- Removed the color alteration for sorting on all lists as it was unclear.
- Added unselectable css to most of the lists to prevent weird click and drags display issues.
- Added unselectable css to logic flows to prevent weird click and drags display issues.

Version 1.0.0.25 Beta 05/01/2017

> Warnings:
> While this version fixes a couple of bugs it also has some new features that are incomplete and therefore not entirely working.
>
> We are releasing it anyway to provide the bug fixes to customers that need those fixes.
>
> Please review these release notes for details.

- Fixed an issue that was preventing the addition of combiners when picking start and endpoints.
    - This prevented certain types of valid flows from being created/extended.
- Fixed an issue that could cause blocks in a flow to overlay when a translator with an external start point is joined to a combiner in the middle of a flow.
- Fixed a bug that was causing tcp listeners in emulators not to be able to send data to connected clients when in a cluster.
- Work in progress: Added the ability to add AES67 streams to the Axia Audio Router manually.
    - There is still some work we need to do to make the addition UI more user friendly.
- Work in progress: Started the ability to edit virtual routers from the web UI.
    - Please Note: This work is not complete and the web pages are not fully functional yet.
- Work in progress: Added the JSON system item to retrieve blocks of responses in a json blob.
- Expect another version very soon with the work in progress complete.

Version 1.0.0.24 Beta 04/15/2017

- This is the first version since the new build process has been implemented in 1.0.0.23.
    - Please report any issues as that may indicate permissions or other file settings that have been missed in the new build scripts.
- Added a pair of scripts to monitor the ifconfig network files for gateway changes and clone the gateway change to the opposing file.
-

Made extension licenses cluster aware.
- Added round robin connection support for clustering between Axia and Office Lan for better clustering redundancy.
- Added the ability to change the default level for silence and clipping on the Alarms web page.
- Fixed an issue introduced by the Qor changes in 1.0.0.18 where Lwch channels were not presenting the But#1 and But#2 objects on Element and Fusion anymore.
- Updated Pathfinder Client to 5.70.
  - Switched the default state of meters obtained by right click on route points to use Standard scale and 3000 decay.
  - Increased the default width of source and destination name columns especially in situations where the left hand route panel is closed.
- Added the MIX and INP objects to Xnode devices.
- Exposed the MIX capabilities in the logic flows simple tree.
  - Additional work is planned in the UI surrounding this capability, but this exposes the XNode Mixer functionality to be used in Logic Flows.

Version 1.0.0.23 Beta 04/06/2017

- No real functional changes in this version.
- This version was about automating the OS and update package build process with this version as the result.
  - Report anything that does not work as this would indicate problems with the new build script that need to be resolved.

Version 1.0.0.21 Beta 03/29/2017

> ⓘ Important Notes:
> This version includes a major rework of clustering started in 1.0.0.14. See notes from 1.0.0.14
>
> We recommend doing a manual sync on the secondary server after both are upgraded to force synchronicity.

- Fixed an issue with Fach and Lwch mute state sending the incorrect case sensitivity.
  - We were sending MUTE_State=MUTED instead of Mute_State=MUTED.
- Updated licensing to support vm version of license.
- Fixed a bug that was sometimes causing the new generic emulator udplistener object to fail to be created.
  - SapId was requesting the local endpoint uri before it was available.

Version 1.0.0.20 Beta 03/27/2017

> ⓘ Important Notes:
> This version includes a major rework of clustering started in 1.0.0.14. See notes from 1.0.0.14
> We recommend doing a manual sync on the secondary server after both are upgraded to force synchronicity.

- Fixed an issue where Mute_State was being displayed in logic flows as on and off instead of Muted and normal.
- Fixed an issue where changing the name and/or description of a router was not always getting written to the backing storage and so could revert after a restart.
- Added a confirmation dialog to device emulator removal.
- Updated PanelDesignerCore to 5.68
  - This version adds an experimental menu item called Open Local.
  - This will allow you to open a panel designer file from PFPro that is on your local hard drive and then resave to CorePro.

> ⓘ Warnings:
> Use SaveAs rather than save to save it to CorePro for the first time.
>
> The panel may need some changes in order to get mappings correct.
>
> This is a new feature and not completely proven or tested.
>
> If the panel is not synchronizing correctly across the cluster or being retained after a restart after uploading it, try deleting it and resaving it from Panel Designer.
>
> There are differences between PFPro and CorePro in panels so please report any panels that do not work properly so we can continue to make this feature better.

Version 1.0.0.19 Beta 03/25/2017

> ⓘ Important Notes:
> This version includes a major rework of clustering started in 1.0.0.14. See notes from 1.0.0.14
>
> We recommend doing a manual sync on the secondary server after both are upgraded to force synchronicity.

- Added a confirmation to timer deletion.
- Added a Generic Udp Client and Listener to Generic Emulators.
- Removed Timestamp from MessageReceived as it was not being used and therefore was just superfluous memory utilization and calculations.
- Fixed a bug with LegacyPanel renaming.
  - The new code that made sure removals were created with SapIdChanges when necessary had a bug that was not expressing the object path in the removal correctly.

  - This was causing problems with Legacy Panel synchronization.
  - If node B was offline and a panel was renamed, when B came back online and synchronized it would end up with two copies of the panel under both names.
  - This has been fixed.
- Working on fixing an issue with renaming LogicFlowFolders.
  - If node B was offline and a Flow folder was renamed, when B came back online and synchronized it would end up with an empty folder on the B server.
  - This requires querying the sub objects in that case when a new object is constructed based on a Sync Message.
  - Also added some optimization code to prevent numerous unneeded messages for branches where this is not an issue.
- Fixed multiple issues with Device Emulator cluster synchronization.
- Fixed an issue with Generic Emulator Watcher removals and cluster synchronization.
- Fixed issues with "get _" and Last_Update with redirected services such as DeviceEmulators.
- Fixed some bugs that were returning an extra NONE from the update moderators when the underscore was used as the object path with last update values.
  - This is because updatemoderators have not been updated to support these features.
  - So excluded them to prevent the extra None message.
- In PathfinderPCClient_Core5.68:
  - Added buttons under the virtual router editing grids that will ask for a new id number and allow you to assign one as long as it is not already in use.
  - Added a button to the virtual router IO editing window that reset the virtual io name and description to the selected item in the list.
  - Fixed some issues regarding description fields in the virtual router io editing window.
    - They were showing ip addresses instead of descriptions.
    - In some cases this could still occur on the first attempt to edit a virtual IO.
    - Closing and reopening the edit window will display the correct description.
  - Widened the Id column in virtual router editing for longer numbers.
  - Compiled as 5.68_Core

Version 1.0.0.18 Beta 03/21/2017

> ⓘ Important Notes:
> This version includes a major rework of clustering started in 1.0.0.14. See notes from 1.0.0.14
>
> We recommend doing a manual sync on the secondary server after both are upgraded to force synchronicity.

- Added support for Qor console changes.
  - Requires Qor version 2.2.0.125 or later
  -

Lwcp access will not be attempted if the Qor software version is earlier than 2.2.0.125.

- Added some code to prevent increment/decrement from being processed on numerical memory slots during sync operations to prevent the value creeping when a node comes back online.
- Some work done on logic flow folder renaming when one server is offline in a cluster. There is still an issue here we are working on fixing. See Known issues below.
- Changing the name of a panel could cause it to be unusable until the system was restarted. This should be fixed.
- Worked on name changing of objects and making sure that synced properly in a cluster.
- Added stacktrace to the generated log message in ordinalmessagequeue failures for easier debugging.
- Added some better error trapping to cluster message handling to prevent a group of messages from not being processed when one in the middle fails.
- Fixed an issue where changes to a translator's conversion list would not sync when the secondary was offline and came back online.
- Fixed and issue where you cannot change the pattern of a StringBuilder memory slot because the set accessor did not exist.
  - This affected changing it from the web page and some cluster sync scenarios.
- Fixed a bug that was preventing name and description properties from being updated on VirtualDestinations via SapV2.
  - This also affected cluster synchronization of changes on Virtual Destination name and description changes.
- Fixed a bug where the BaseRouter class was not exposing the Subversion property.
- Fixed an issue where cluster sync messages would get processed by the code for sending to remote devices.
  - This in some cases prevented certain updates. For example this was causing virtual destination name changes to not sync properly when secondary comes back online.
- Known Issues:
- No confirmation message on the deletion of timer on the timers web page.
- Changing the name of a logic flow folder on one server when the other server is offline can leave the changed folder empty on the server that is being brought online after it syncs.
  - This will be fixed in the next build.
  - This can be fixed by a manual sync.
- Changing the name of a Legacy Panel on one server when the other server of offline can leave both the old panel and the new one on the server that is being brought online after it syncs.
  - This will be fixed in the next build.
  - This can be fixed by just deleting the extra panel on the newly recovered server.

Version 1.0.0.17 Beta 03/16/2017

> ⓘ Important Notes:
> This version includes a major rework of clustering started in 1.0.0.14. See notes from 1.0.0.14

- Fixed an issue that was causing IE to lock up on the systemstatus web page.
  - Please note that IE is still not updating the cpu usage as frequently as other browsers, but at least it is no longer locking up the web page in this version.
- Sped up the update interval on cpu usage on the system status web page.
- Fixing an issue with processing removal items on the Email branch. Since they are a level up, the system#0 was not being passed through.
- Updated client/Mini/PanelDesigner application to 5.67.
  - The only change in this version of clients is an update of the dlls to support nested encapsulation messages.
- Added the SynAck operator for acknowledging cluster synchronization changes and updating the node version stamp.
  - This makes update version tracking more accurate requiring fewer sync messages during startup synchronization.

Version 1.0.0.16 Beta 03/14/2017

> ⓘ Important Notes:
> This version includes a major rework of clustering started in 1.0.0.14. See notes from 1.0.0.14

- Fixed a bug with nested encapsulation that was introduced in 1.0.0.15.
  - This bug was causing a parsing error that was preventing legacy panels from being initialized properly.
- Fixed a bug with the synchronization of device emulators. The constructor was being synced as a single message rather than multiple as intended.
- Fixed a bug introduced in 1.0.0.15 that was preventing multiple device emulators of the same type from being loaded during startup.
- Fixed a bug where the redirector for deviceemulators was not obtaining the correct version stamp.
  - This prevented emulators from being deleted when an offline node came back online and one of the emulators had been deleted while it was offline.
- Fixed an issue with calculating memory like TextBuilder slots where they where clustering LastUpdate was being updated when it should not.
  - This was preventing proper removal in the case of an offline server coming back online.
  - The state change as it came back online was overriding the remove from the online server.

Version 1.0.0.15 Beta 03/08/2017

> ⓘ Important Notes:
> This version includes a major rework of clustering started in 1.0.0.14. See notes from 1.0.0.14

- Added a confirmation message to the leave button of Clustering.
- Hid the itemsloadeded and removed on the clustering tab as these are meaningless after the clustering revisions in 1.0.0.14.
- Fixed a bug with properties that SetLastUpdate but are existence only synchronization objects.
  - Those were not being transitioned across the clustering.
  - This could be seen with logic flow objects where the enabled property state would not replicate.
- Changed the log level for initObject when the object already exists to Info for normal inits and Debug for clustering inits.
- Combiners were set to full synchronize instead of existence only. Fixed.
- Object translators were passing additional property values if their input was a relay or pasthru combiner. Should be fixed.
  - This was occasionally causing cyclic clustering messages.
- Added escaping of nested BeginEncap and EndEncap escape them by wrapping with <> - for example: <%BeginEncap%> - multiple levels would then become <<%BeginEncap%>>.
- Decorated the device password property to set last update so that it synchronizes properly across the cluster.
- Updated the name and description property changes to be decorated with clustering setslastupdate.
- Added a manual sync option to the clustering web page.
  - Clicking this in an active cluster will raise a confirmation message.
  - If confirmed the system will request a backup from one of the other nodes in the cluster, download and restore that backup, and reboot.
  - This feature may still be a bit rough. Need to add better status information to show possible failure states if it is unsuccessful.
  - Also this button is currently active whenever the cluster exists but in the future should only be active if one of the other nodes in the cluster is currently online.

Version 1.0.0.14 Beta 03/04/2017

> ⓘ Warnings:
> This version includes a major rework of clustering.
>
> Take a backup before upgrading to this version.
>
> It is recommended to have all nodes leave the cluster, upgrade them them to the new version, and then recreate the cluster.
>
> Please report any issues. We are still actively testing this version internally as well for any clustering issues.

- Added error catching around log files in case they are not obtainable.
  - In a destroy/create scenario of a logger this also allows the previous instance to release its hold on the file.

- Optimized the SapMessaging to remove some unneeded storage structures except when SapMessages are being stored as SapObjects.
- Fixed an issue where Email Host was getting added with NOW as the lastversion and then getting the lastversion reset with the file version.
- Fixed a bug where it is possible for clustering to think a node is done starting all services when services are still loading.
- Fixed a database storage error that affected storing persistent changes to memory slots.
- Fixed a bug with the parameterization of update queries in certain cases. This bug has existed since the database changes in 1.0.0.01.
- Fixed a bug where ACK was not being returned for diverted property changes such as route changes.
  - Ack means the message has been received interpreted and the target obtained - not necessarily that the change has been completed.
- Optimized DeviceEmulator removals.
- Added some code to prevent recreating DeviceEmulator request objects if the device emulator already exists.
  - This prevents a clustering redundant device emulator creation situation.
- Fixed a bug specific to the powerstation and Fusion that could cause a stack overflow crash during certain subobject removals.
- Fixed a bug with virtual io deletions from the database. There was an incorrect query syntax.
  - Probably started after the database revisions in 1.0.0.01 but could have been earlier.
- Fixed an issue where Synchronization messages could cause changes to be sent to equipment when they should not be.
  - In the case of route points this was occasionally causing some destination routes to be cleared during certain cluster synchronization steps.
- Fixed an issue where certain changes were causing an extra Last update cluster message.
- Fixed an issue where removing an object was causing a recursive analysis of the path to object lookup list to remove entries.
  - This caused huge cpu load when doing things like deleting a router with a large number of ios.
- Added the ability to query using the Last_Update system item and a greater than operator to return changes later than a certain date/time.
- Added the ability to return Constructor with other properties by using the $CONTAINS_PROP system item along with the Constructor property.
- Added the ability to request objects with properties using get _
  - This can produce lots of data and it not intended for general use without additional parameters.
  - It is used in the revised clustering synchronization in addition to LastUpdate to return changes later than certain time stamps.
- Completely redesigned and rewrote clustering
  - This version uses a fraction of the messaging between systems during synchronization of previous versions.
  - This version removes some potentially large and now unnecessary data caches.

Version 1.0.0.13 Beta 02/03/2017

- Fixed a bug in the time creation window with date time events. The default value was incorrectly formatted.
- Fixed a bug with DayOfWeek events. They were firing repeatedly for about 15 seconds at the requested time. This should be fixed.
- Fixed a bug where NextRaiseTime was not always being raised to SapV2 when it changed.
- Since mount points the internal address of Axia Audio sources were not getting updated when the RtpStream Address (Livewire Channel) was being changed. This is fixed.
- Since mount points the internal address and livewire channel changes on Axia Audio sources were not being raised as change messages on route points when the RtpStream address changed.
- Fixed an issue where source channel number changes were not being updated in the io grids on the web page.

- Updated the client, mini, and panel designer to 5.66.
  - Added a CenterPad property to the Meter control in panels.
    - Increasing the center pad will open space between the left and right meters, and once large enough scale markings will appear.
  - Added a MeterStyle property to both the meter control and meterfader control in panels.
    - This property will allow you to switch the meters from Led to gradient on either control.
  - Added a MeterScale property to both the meter control and meterfader control in panels.
  - This property will allow you to switch between linear (same as old meters), standard which has different decibel ranges accented, and British which uses a more BBC like scale.
  - Added a LevelLineColor property for adjusting the color of the lines in the scale markings on meter controls.
  - Added a DecayRate property to meter and meterfader controls.
  - This property allows you to define a time range in ms over which the meter will fade to nothing and yields smoother looking meters.
  - The time range defines the falloff time if the level drops from 0 to -100.
  - For shorter drops, the falloff is computed as an equivalent rate over the length of the drop.
  - We are finding that around 3000ms on the decay rate yields much more natural looking meters.
  - Please note that setting the decay rate to anything other than zero will increase cpu load on the client displaying the meters as the falloff is computed and drawn at 10ms intervals. This will be most noticeable on larger meter panels.
  - Please note that for now in this version the default meter style is the same as the meters have always been. You must edit the panels to make use of these new features.
  - Updated the gain range of meterfader controls to reflect xnode gain ranges.
  - Added a FaderNominal property to MeterFader. This property allows you to set the red nominal gain box at a value other than the default 0.
  - Added the property HideGainValue to MeterFaders. This property hides the box displaying the gain value.
    -

This is designed to be used if you are altering the FaderNominal level and do not want to confuse the user by displaying the actual gain value from the device.

Version 1.0.0.12 Beta 01/25/2017

- Fixed a bug that was sending VMIX#0.SUB#1. . . instead of VMIX.SUB#1 to the equipment.
  - This worked on some versions of Element/Engine/Fusion software and not on others but the latter is more correct.
- Fixed a bug that was introduced when we moved the services internal during the past few beta versions.
  - Changing properties on mounted objects was not always working. For example trying to change the gpio on a virtual router that had the gpio mounted was broken.
- Fixed a bug with setting pinstates property instead of the individual pin object pinstat property.
- Added new versions of PanelDesigner, Mini, and Client (5.64).
  - Added a CenterPad property to the Meter control in panels.
    - Increasing the center pad will open space between the left and right meters, and once large enough scale markings will appear.
  - Added a MeterStyle property to both the meter control and meterfader control in panels.
    - This property will allow you to switch the meters from Led to gradient on either control.
  - Added a MeterScale property to both the meter control and meterfader control in panels.
  - This property will allow you to switch between linear (same as old meters), standard which has different decibel ranges accented, and British which uses a more BBC like scale.
  - Added a LevelLineColor property for adjusting the color of the lines in the scale markings on meter controls.
  - Added a DecayRate property to meter and meterfader controls.
    - This property allows you to define a time range in ms over which the meter will fade to nothing and yields smoother looking meters.
    - The time range defines the falloff time if the level drops from 0 to -100.
    - For shorter drops, the falloff is computed as an equivalent rate over the length of the drop.
    - We are finding that around 3000ms on the decay rate yields much more natural looking meters.
    - Please note that setting the decay rate to anything other than zero will increase cpu load on the client displaying the meters as the falloff is computed and drawn at 10ms intervals. This will be most noticeable on larger meter panels.
- Added code to LegacyPanelManager to handle the new properties in meters and meterfaders in panels.

Version 1.0.0.11 Beta 01/09/2017

- Fixed an issue where DeviceDicoveryItems were not being cleaned up from the path lookup table after they were done being used.
  - This was preventing an item from being submitted for discovery a second time if it had been removed or failed the discovery the first time without a restart of the software first.
- Fixed an issue where newly discovered devices were often recycling their connections once after being added into the system when they did not need to be.

- Fixed an issue where pending device investigations would not be queryable with a SapV2 get request.
- Fixed an issue where device investigations were sometimes getting stuck in the open investigation list.
  - This could prevent new investigations without a restart since the concurrent investigations have been limited to a count of 3 in order to prevent overloading the processor during large numbers of device discovery requeue.
- Marked the enabled property on Combiner and Translator for cluster synchronization.
- Improved detection of property changes on items that are set for existence only synchronization but have specific properties that require full synchronization.
  - This could be a breaking change. Please report any issues with clustering if the arise.
- Implemented ordering when pulling ip addresses from the investigation todo list.
  - Please note that items may not be added in order because of response time differences with parallel investigation, but it should be closer and less random when investigating a specific ip range.
- Added a property to modify the number of allowed parallel device investigations. By default it is 3. This is not saved between restarts.
  - Use the Advanced options to add this to the startup config options if it needs to be retained.
- Fixed a bug that could cause the device list to be analyzed repeatedly trying to convert unknown lwrp devices to known types.
- Fixed a bug with re-investigating happening with emulators with certain device type misnames.
  - This should not happen in real life but should be fixed.
- Fixed bugs with the license counting on routers. It was counting active sources more than once in some cases and was also including the None and Previous sources.
- Added code to exclude translators that point to other objects within the flow (for example combiners) so that only actionable endpoints are counted against the licensing.
- Also made route licensing get counted after all route points are loaded during startup instead of potentially causing cpu load while loading is taking place.
- Fixed a cluster sync speed issue with disabling/enabling logic flows.
- Fixed an issue with exception logging in Clustering.
- Reworked the PathIo object to only require SapObject meta data in the rare cases it is needed rather than for all instances.
  - This potentially has a significant impact on reducing memory consumption on large systems.
- Channel Numbers was not being raised as an INDI when Axia Audio Destination routes were changing. This is fixed.
  - This was causing the issue where livewire channel numbers were not being updated in the web page routing list without a refresh of the page.

Version 1.0.0.10 Beta 12/30/2016

> Warnings:
> This version along with 1.0.0.09 have significant internal changes with SapV2 and parallel message processing.

> If these changes cause problems, please report and revert to an earlier version.

- Fixed issues with the IPClient that was causing Linux to retain excessive numbers of open file handles when sockets were recycling.
-
    - In some situations this was causing too many open files errors in the logs and connectivity problems as well as some additional cpu load. This was typically seen only in large systems where a number of devices where offline.
- Fixed an issue introduced in 1.0.0.09 with the local service changes. Mount points were not being established properly if the mount target was not available on the first initial mount request.
- Fixed a benign issue where PFCorePro was requesting show profile states from Iports which do not have show profiles.
- Improved the device discovery process under Linux.
- Added a property to the device investigator object to display in progress investigations.
- Updated the notes in the advanced options web page to indicate that the options require a restart to take effect.

Version 1.0.0.09 Beta 12/21/2016

> ⓘ Important Notes:
> This version has significant internal changes with SapV2 and parallel message processing.
>
> If these changes cause problems, please report and revert to an earlier version.

- Fixed a bug with the SapV2 revisions in 1.0.0.08 that was not doing case insensitive comparisons on sub and unsub.
- Added a property option to switch Lwrp message polling to be VER only. This appears in Device#0 as LwrpVerPollingOnly.
- Got rid of the timer being used for VirtualMount mapping and instead depend on analyzing based on additions and removals.
    - Greatly reduces cpu load when there are many outstanding virtual basi ios trying to be mapped when the ios they are being mapped to do not exist.
- Added an option to skip router sanity polling - Set Routers#0 SkipSanityPoll=True/False
- Recrafted the receive part of IpClient to share a buffer rather than recreating and to use the ReceiveAsync methodology instead.
- Property changes that were not supposed to raise cluster messages were raising them in some cases. This should be fixed in this version.
- Moved all services to local startup without the extraneous redirects.
    - Added startup config file options to define this.
- Reworked the threading, task switching, ordered messaging to improve performance. The system was spawning more tasks than necessary.

- This is a shift in how parallel processing is handled in the application.

- Fixed a bug that was in certain cases leaving the clustering stuck in a GatheringLocalCacheData state.
- Added a property to Clustering to define whether to use message buffering for internal messages. False by default.
- Added a property to LogicFlows to define whether to use message buffering for internal messages. False by default.
- Added a property to LogicFlows to define whether to use message tasks for internal messages. False by default.
- Added a property to allow logic flows to spawn tasks on inbound messages if desired.
- Revised the options file so that it could contain the settings for the optional queueing, tasks, sanity polling and VER polling.
- Reworked Supervisor to write the options file with default values if it does not exist on startup and to set the Linux permissions such that nginx can manipulate the file.
- Reworked Supervisor to write a default options version file with default values for the software revision if it does not exist on startup.
  - This can be used to reset to the software version's defaults.
- Added a web page for the advanced configuring of the options file.
  - Included warnings that the options are for tuning a system in collaboration with Axia support only.
- Fixed an issue with editing virtual ios in the client where changing a base io might spawn a second base io or not be picked up in the Pcp cache requiring a client restart to see the changes.
- Fixed an issue with changing the name or description of a Virtual IO via the client application.
  - It could cause the base ios to be removed from the virtual IO.
- Fixed an issue where removing base ios from a virtual io using the client was not always working properly.
- Fixed an issue where after changing a base io, the description field in the editing window grid was not updating.
- In the client, moved the Create and Edit menu items to the top of the list so that with large router lists they are not below the list.
- Compiled Client, Mini, and PanelDesigner to 5.63 to make sure all are using most recent dlls.
- Known Issues:
  - If a Virtual Router is left open when you shut down the Client, we have occasionally seen the virtual router open up when the client restarts with no route states. Closing and reopening the router in the client fixes this. Still investigating if/when this happens.
  - There has been a report of the web page for routing not updating the source channel number for audio routers.

Version 1.0.0.08 Beta 12/08/2016

> ⓘ Warning:
> This version has significant internal changes with SapV2. If they cause problems, please report and revert to an earlier version.

- Reworked the code such that redirects if using an Internal sap client bypass the message queuing and reworking of transaction ids, etc and instead just pass the originating client directly to the redirected object. This should improve performance especially in parallel client get requests.
- Reworked a number of other things with internal Sapv2 messaging for better performance and parallel processing.
- Made internal clients by default not use a message queue.
- Enabled the queue on the receive side of service connection home clients only. So sends go from the service to device manager with no queue but responses may get queued.
- Fixed a bug that was introduced between 1.0.0.05 and 1.0.0.06 which broke Lwrp discovery.
- Fixed a bug with the log writer editor that on each save after connection was adding _log to the name of the file.

- Finished fixing LWRP so that DST no longer sends a blank Name field when sending route changes.
- Lwrp Change messages are now generated by a separate method than was being used for emulation messages as they do need to be slightly different in many cases.
- In Axia Audio route changes included the Io name in the Address message when making standard route changes.

Version 1.0.0.06 Beta 12/02/2016

- Fixed a few places where event handlers may not have been cleaned up properly after object deletion causing the potential for memory leaks.
- Fixed an issue where event handlers might get removed during mount point removal where the primary object still exists.
- Made discovery device connections stop discovering during disposal to make sure event handlers are removed.
- Fixed an issue where audio routers might not reduce the used source count on removed source ios.
- Fixed an issue where gpio routers might not remove a lookup when the corresponding io was removed.
- Added a hook for specific removal options to be processed in a route during io removals. This is similar to what already existed for io additions.
- VXEngine does not have Lwrp addressable Gpios. Switched the HasGpios property for this device from True to False.
- Added some code for generic devices that present Gpios in VER to flip the generic device from HasGpios=False to HasGpios=True.
- Modified LogManager to work as an internal process.
- Added the ability to enable logging of Lwrp, Lwcp, SapV2Internal, and SapV2External messages.

> ⓘ Warnings:
> Logging such messages can cause increased load and stress on the cpu. These options are

> designed for troubleshooting.
> The SapV2Internal item is especially verbose and should only be used in consultation with support and/or development.

- Also hooked mount requests to the SapV2 internal message logging.
- Added a warning asterisk in the log writer configuration web page regarding message logging.
- Fixed an issue where virtual sources and destinations were trying to mount empty device paths.
    - Virtual sources and destinations should not mount anything. Only their BaseIos in the packages should mount.
- Fixed an issue where none and previous routes were requesting mounts with empty device paths.
- Prevented mount messaging if the requested path was empty.

Version 1.0.0.05 Beta 11/28/2016

- Fixed a bug introduced in 1.0.0.02 with sending lwcp changes to the equipment. That change started sending an erroneous object in the path as part of the lwcp message object path which means lwcp commands would not always work as of 1.0.0.02.
- Removed the attempts from panel hwmap generation. It will now continue trying to regenerate its flows every 5 seconds until it is successful rather than stopping after 4 attempts.

> Important Note:
> It is a known issue that hardware map flows will not be generated for a panel if the panel has a control mapped to a device ip that does not exist in the devices list.

- Made hardware map devices list pull unsorted for better cpu utilization.
- Fixed an issue where GPI names were not being picked up all the time.
- This version adds the CFG GPO mount point under GPI sources in addition to where it was previously under GPO Destinations.
- Fixed an issue where gpio name changes were not always being written to backing storage.
- Fixed a bug where sometimes if a device was offline when PCP started, audio alarms associated with that device would not be reset and start working when the device comes online again without a restart of PCP.

Version 1.0.0.04 Beta 11/28/2016

- Not released to public. Bad compile.
- Skipped this version number.

Version 1.0.0.03 Beta 11/23/2016

- Not released to public do to problems with the SapV2 changes in this version.
- Found a problem with the new SapV2 changes introduced in 1.0.0.02 that was causing items to get a path with no root object. This was affecting some object's property change reporting.

Version 1.0.0.02 Beta 11/21/2016

- Not released to public do to problems with the SapV2 changes in this version.
- Fixed a memory leak that could happen if device data points were not present after a restart that were utilized by routers. This could happen after a restart if the device in question was not online.
- Optimized some of the code around SapV2 object paths.
- Fixed an issue where services might not get shut down properly during a restore operation.

Version 1.0.0.01 Beta 11/17/2016

- Fixed an issue that was sometimes causing router deletes not to get stored to the backing storage. Previous was causing duplicate removals in a dictionary and erroring out. That is fixed.
- Fixed an issue that was causing removals to sometimes get written to the backing storage even if the cluster was synchronized or there were no nodes in the cluster.
- Made one more fix potentially related to the audio router not posting routes when multiple Axia Audio routers are present.
- Added a timeout in waiting for the router to sync in pcp on virtual routers in order to account for missing base io resources.
- Fixed virtual router editing when Base Ios are pointed at devices that do not exist. This condition was not allowing the virtual router to be edited previously.
- Fixed a bug where routes might not work when there were multiple Axia audio routers in the system (only currently possible using the API).
- Fixed a bug with clients where devices and/or the underlying base ios cannot be discovered when resaving the router.
- Ordered routers by Id in the client application.
- Added the id field to the routers web page.
- Fixed a bug where gpio names were not getting picked up into the Gpio Router.
- Fixed an issue where the mount points for gpio destinations was incorrect. Pins were appearing under cfg.gpo mountpoint instead of under the gpo mount point which did not exist. And cfg.gpo was connected to the wrong gpo object.
- Ordered the virtual router list by router id.
- Ordered panels by name in Panel Editor.
- Fixed an issue where using a password when initializing a DiscoveryDeviceData object via the API in order to investigate an object was not passing the password object through to the investigation.
- Fixed an issue where set messages in the API with acks were not always returning an ack.
- Fixed an issue where del messages in the API with acks were not always returning an ack.

- Fixed an issue where double led messages were often being returned when a del message was sent in the API.
- Reworked the database queries for better security and to properly allow characters like the single quote in IO names.
- Modified the code so that logic flows that live under the hardware maps folder as well as the hardware maps folder itself do not get written to backing storage as those flows get recreated anyway by the legacy panels load function.
- Fixed an issue where hardware maps were going through a double create and delete during startup. This should fix an issue where occasionally hardware map flows were not coming back after a restart requiring opening and resaving the panel to get them back.
- Fixed some pcp errors with associating the correct router to scenes in the client application.
- Fixed an issue where the writepanel property of panels was being written to the xml when it should not have been.

Version 1.0.0.00 Release 11/03/2016

- Version 1.0 release.
- Edge browser and Internet Explorer both lock up on several of the web pages. Chrome is the recommended browser to use though Firefox also works.
- Silence and audio presence thresholds are currently fixed at -80.0 and -1.0 and cannot be changed. That will be addressed in a later version.
- The gateway needs to be assigned in both Lan and Wan to the same Gateway address or it may pick randomly which one to use.
- Clustering currently only uses the Axia Nic. In the near future it will use both for redundancy.
- Clustering will eventually pick up extension licenses from connected clustered units but in version 1.0 extension licenses must be added manually to all units in a cluster.
- A virtual router may not be editable in PathfinderPC Client if the device the virtual router is referencing has been deleted from the system. Fix will be posted to the beta page shortly.