Pathfinder Core PRO

Installation & User's Guide

Manual Rev 1.4 - October 2018

Notices and Cautions

CAUTION:

The installation and service instructions in this manual are for use by qualified personnel only. 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 autoranging line voltage input. Ensure the power voltage is within the specified range of 100-240v. The ~ symbol, if used, indicates an alternating current supply.

This symbol, wherever 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.

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

CAUTION: DOUBLE POLE/NEUTRAL FUSING

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 fuse only with same type and value for continued protection against fire.

WARNING:

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 equipment rack, an easily accessible safety disconnect device 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 in the top 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.

WARNUNG:

Die Installations-und Serviceanleitung in diesem Handbuch ist für die Benutzung durch qualifiziertes Fachpersonal. Um Stromschläge zu vermeiden führen Sie keine andere Wartung durch als in dieser Betriebsanleitung aufgeführt, es sei denn Sie sind dafür qualifiziert. Überlassen Sie alle Reparaturarbeiten qualifiziertem Fachpersonal.

Dieses Gerät hat eine automatische Bereichseinstellung der Netzspannung.
Stellen sie sicher, dass die verwendete Netzspannung im Bereich von 100-240V liegt.
Das Symbol ~, falls verwendet, bezeichnet eine Wechselstromversorgung.

Dieses Symbol, wo immer es auftaucht, macht Sie auf nicht isolierte, gefährliche elektrische Spannung (ausreichend um einen Stromschlag hervorzurufen) innerhalb des Gehäuses aufmerksam. Spannungen.

Dieses Symbol, wo immer es auftaucht, weist Sie auf wichtige Bedienungs-und Wartungsanleitung hin. Lesen Sie die Bedienungsanleitung.

ACHTUNG: ZWEIPOLIGE ABSICHERUNG / NULLEITER ABSICHERUNG

Das Netzteil des Gerätes hat eine interne Sicherung eingebaut. Auch wenn die Sicherung durchgebrannt ist, können auf einigen primären Bauteilen noch gefährliche Spannungen vorhanden sein. Wenn ein Austausch der Sicherung erforderlich ist, ersetzen Sie die Sicherung nur mit gleicher Art und Wert für den kontinuierlichen Schutz gegen Feuer.

WARNUNG:

Das Gerätenetzkabel ist die Haupttrennvorrichtung. Die Steckdose sollte sich in der Nähe des Gerätes befinden und leicht zugänglich sein. Das Gerät sollte nicht so angeordnet sein, dass der Zugang zum Netzkabel beeinträchtigt ist. Wird das Gerät in ein Rack eingebaut, sollte eine leicht zugängliche Sicherheitstrennvorrichtung in den Rack-Aufbau mit einbezogen werden.

Um die Gefahr von Stromschlägen zu verringern, darf dieses Produkt nicht Regen oder Feuchtigkeit ausgesetzt werden. Dieses Gerät ist nur für die Benützung im Innenbereich. Dieses Gerät erfordert freie Luftzirkulation für eine ausreichende Kühlung. Blockieren sie nicht die Lüftungsschlitze auf der Geräteoberseite und den Seiten des Gerätes. Unzureichende Belüftung kann das Gerät beschädigen oder Brandgefahr verursachen. Platzieren Sie das Gerät nicht auf einem Teppich, Poster oder andere Materialien welche die Lüftungsöffnungen beeinträchtigen könnten.

Wird das Gerät anders als in der, vom Hersteller angegebenen Weise verwendet, kann der, durch das Gerät gegebene Schutz beeinträchtigt werden.

USA CLASS A COMPUTING DEVICE INFORMATION TO USER. WARNING:

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 interference 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.

CANADA 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.

Pathfinder Core PRO Manual

TRADEMARKS

Axia Audio and the Axia logo are trademarks of TLS Corp. All other trademarks are the property of their respective holders.

NOTICE

All versions, claims of compatibility, trademarks, etc. of hardware and software products not made by Axia Audio which are mentioned in this manual or accompanying material are informational only. Axia 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.

WARRANTY

For the latest Telos Alliance warranty, visit: telosalliance.com/warranty

UPDATES

The operation of Pathfinder Core PRO is determined largely by software. We routinely release new versions to add features and fix bugs. Check the Axia Audio web site for the latest. We encourage you to sign-up for the email notification service offered on the site.

FEEDBACK

We welcome feedback on any aspect of Pathfinder Core PRO, 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.

SERVICE

You must contact Axia before returning any equipment for factory service. We will need your unit's serial number, located on the back of the unit. Axia 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 Axia Technical Support at +1-216-622-0247. All other customers should contact local representative to make arrangements for service.

We Support You...

BY PHONE / FAX:

- You may reach our 24/7 Support team anytime around the clock by calling +1-216-622-0247.
- For billing questions or other non-emergency technical questions, call +1-216-241-7225 between 9:30 am to 6:00 PM, USA Eastern time, Monday through Friday.
- Our Fax number is +1-216-241-4103.

BY E-MAIL:

- Technical support is available at **support@telosalliance.com**.
- All other questions, please email **inquiry@telosalliance.com**.

VIA WORLD WIDE WEB:

The Axia Audio web site has a variety of information which may be useful for product selection and support. The url is **telosalliance.com**.

REGISTER YOUR PRODUCT

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

TELOS ALLIANCE

1241 Superior Avenue E.
Cleveland, OH., 44114 USA

+1-216-241-7225 (phone)
+1-216-241-4103 (fax)
+1-216-622-0247 (24/7 technical support)

support@telosalliance.com
inquiry@telosalliance.com

Table of Contents

War of the Waves

Dear Valued Customer,

It's with great pride and a tip of the hat to an incredible team that I congratulate you on your new Telos Alliance product. Everything we do here at the radio division of the Telos Alliance is with one end goal in mind: To help broadcasters declare victory in extremely competitive environments. By purchasing this product from us, in essence, you have declared war on your competition.

After all, the majority of Telos Alliance employees were broadcasters themselves once, and the products we've developed over the years have been designed as solutions to specific issues faced on the front lines of our industry. We're right there in the trenches with you and have the weapons you need in your arsenal.

Telos Systems is a catalyst to out-of-this-world sound, with the most powerful and popular broadcast telephone systems in the industry; IP/ISDN codecs and transceivers; plus processing/encoding for streaming audio. We built an industry on the back of these amazing telephony systems, and they are still going strong.

While we at the Telos Alliance never forget our roots, we are also blazing trails in terms of new technologies like stream-encoding and AoIP, so that all types of broadcasters can excel in this ever-evolving digital world.

Omnia Audio not only lets you stand out on the dial with your unique signature sound via legendary audio processors, audio codecs, and microphone processing, it lets you give your listeners a better streaming experience across devices with innovative stream encoding / processing software and hardware.

Axia Audio is a driving force behind the AES67 AoIP standard, and its networked AoIP radio consoles, audio interfaces, networked intercom, and software products continue to move AoIP adoption forward and help broadcasters streamline operations with cohesive, smart, and feature-rich AoIP ecosystems. Last, but certainly not least, 25-Seven has traditionally been known for its audio delays, but its Voltair watermark monitor/processor has made a name for itself more recently as the disruptive product that helped broadcasters take back their ratings and harness the true power of their listening audiences.

You work so hard on your programming day-in and day-out, it deserves technology that will optimize sound and performance at every point in the airchain and online. Armed with Telos products, you have what you need to set your competition squarely in your crosshairs.

With that, I'll leave you to prep your armaments. I hope that you will enjoy your Telos Alliance products for many years to come!

Sincerely,
Frank Foti
Executive Chairman of the Board and Founder, Omnia Audio

Chapter 1

Quickstart

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 make a few basic assumptions:

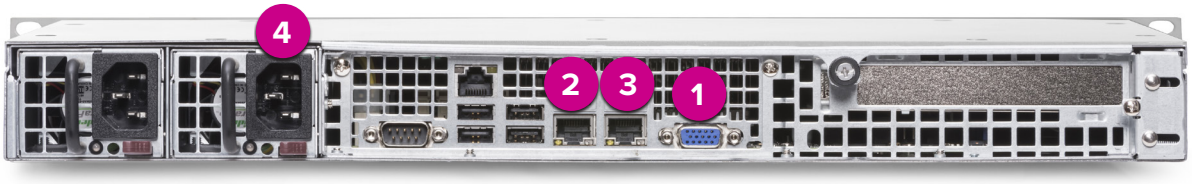
- That you have some knowledge of network basics and network terminology,
- That you are familiar with other Axia Livewire products,
- That you have a correctly configured network switch

Once you're all set up, please peruse the remainder of this manual to gain an in-depth knowledge of Core PRO's more advanced options. Please refer to the quick start guide below that matches the platform you purchased (R2, Virtual Machine, or Fanless Engine). Once completed with the product specific quick start section, finish with the Discovery section of this quick start guide.

Platform Specific Configuration

R2 Platform Configuration

Connection



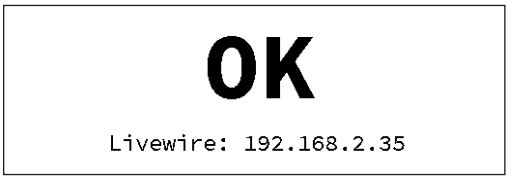
1. The VGA connector on the back of the Pathfinder Core PRO studio engine is not necessary for normal operations. All configuration can be completed through the front panel and the online web pages. Support may ask you to connect a monitor to this port in the course of a support session but otherwise you may leave this port disconnected.

2. Connect the left NIC (when looking at the back of the Pathfinder Core PRO) to your AoIP network using a Cat-6 Ethernet cable.

3. Connect the right NIC to the office network if you want employees from the office network to be able to use panels and other capabilities provided by Pathfinder Core PRO.

4. Using the IEC cables supplied, connect the Pathfinder Core PRO's power supply input to AC mains.

IP Address Configuration



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

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

 = Up

 = Down

 = Left

 = Right

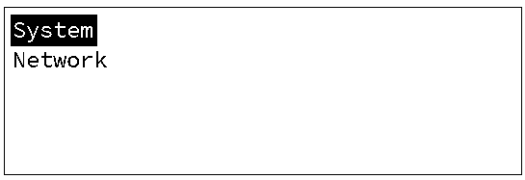
 = Enter

 = Escape

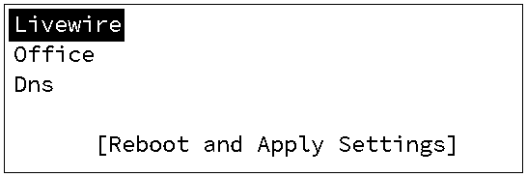
Enter the IP address, netmask, and gateway for the system. To do this:

1. Tap the **Enter** key in the center of the keypad once. A menu view will appear.

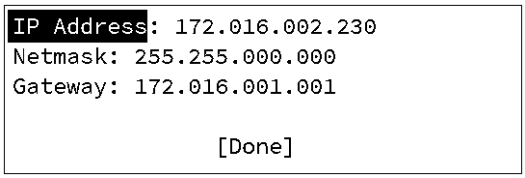
2. Tap the down arrow to highlight the network menu and then the enter key to get into the network menu.



3. Tap the enter key again to enter the Livewire network settings.

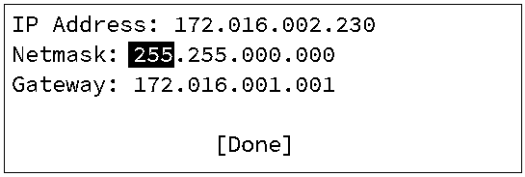


4. Tap the right arrow key to enter the IP Address, then use the up and down arrow keys to increment or decrement each number tapping the right arrow key to advance to the next number.



5. When you reach the end of the IP Address field, tap the right arrow again to return to the main item selection.

6. Arrow down to move to the Netmask. Tap the right arrow to move into the netmask assignment section. Use the up and down arrow to select valid netmask entries and the right arrow to advance to the next section. The system will only allow you to enter valid netmask addresses.



7. Once the netmask is entered properly arrow to the right again to get back to the main sections and then arrow down to the Gateway field.

8. Repeat the procedure to set the gateway. Please note that 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.

9. Arrow down to click Done. This will return to the network menu. Repeat the procedure for the office network and DNS settings.

10. Once all addresses have been assigned, tap the arrow down to select Reboot and Apply Settings.

11. Tap enter to reboot the system.

Move to the discovery section of this quick start chapter.

Virtual Machine Platform Configuration

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

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

Minimum VM instance requirements

The virtual instance created should have the following 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)
- Recommended to use Intel PRO variations for network interface emulation.

Note that larger amounts of ram, processors, and hard disk space can be used for large systems to improve performance and capacity or to make more space available for logs.

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 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 install will not continue. If found, a second confirmation will be requested. 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 install. After boot, 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 by the use of the arrow keys and the ENTER/ESC key. Use the UP/DOWN arrow keys to move the highlight to the NETWORK menu option. When highlighted, press the ENTER key. Select the Livewire network interface and press ENTER. Configure the desired IP settings, again using the ENTER/ESC key to edit or exit and the ARROW keys (UP/DOWN) to increment/decrement the values. You can also type numerals instead of using the arrow keys. When setting the IP, NETMASK, and GATEWAY are complete, select the DONE options. Do the same for the Office interface and the DNS configuration (DNS may be required for email

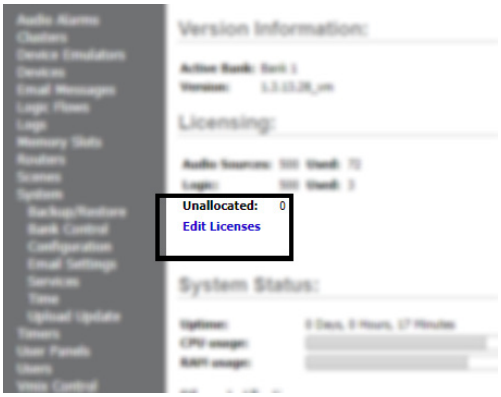
configuration). Once all configurations are complete, select the “Reboot and Apply Settings”. Confirm any request. The system will reboot to the network configuration. The next step is to access the web interfaces of the product. The default authentication is UserName *Admin* with Password *Admin*.

First time licensing access and setup

Access the web interface by typing in the IP address configured earlier in a web browser. Authentication will be requested and the default is

UserName: *Admin*

Password: *Admin*

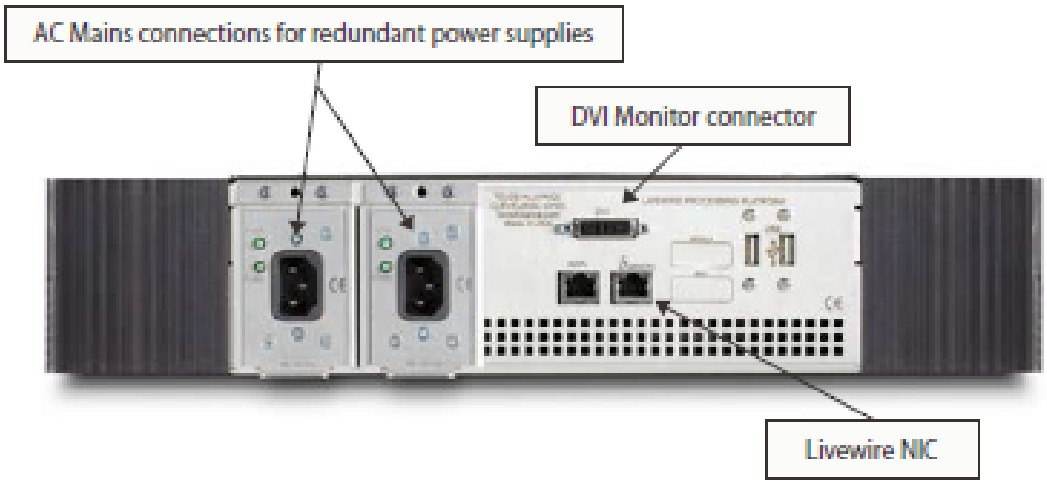


The first step after logging in will be to enter a License. Select **Edit Licenses** link on the System page. A non valid license will appear. Take note of the Request code. Select the NotValid license and copy the Request code so it is in memory followed by the **Generate License** link next to the empty *License Key* field. A new window will come up to the Pathfinder licensing web site. Create a new account unless you have an existing Pathfinder account.

In setting up a new account, make sure the *Activation Type* is set to “Pathfinder Core PRO vm”. The Access Code is provided through the purchase of the VM product. The Request Code is the value from the invalid license which was noted earlier and if you copied it, can paste in the open field. The remaining fields are customer information for registration of the product. It is mportant to fill this information so that it reflects the end customer (maybe not the contracting integrator or installer). Once submitted the account will be generated and License information will be available which should be copied to the License Key field from earlier to validate the license. Once a valid license is in place, reboot to make sure the license is active. Once the licensing is complete, and the network of devices are online, the next logical step is to discover the system. Move to the discovery section of this quick start chapter.

Fanless Engine Configuration

Connection



The DVI monitor connector on the back of the Pathfinder Core PRO studio engine is not necessary for normal operations. All configuration can be completed through the front panel and the online web pages. Support may ask you to connect a monitor to this port in the course of a support session but otherwise, you may leave this port disconnected.

Connect the RJ-45 port marked with the Livewire logo to a port on your AoIP network's local switch using a CAT-6 Ethernet cable.

Connect the RJ-45 port marked with the WAN caption to your office network if you wish to run Pathfinder Client applications in the business environment so that your users may control Axia equipment without providing direct access to the Axia network.

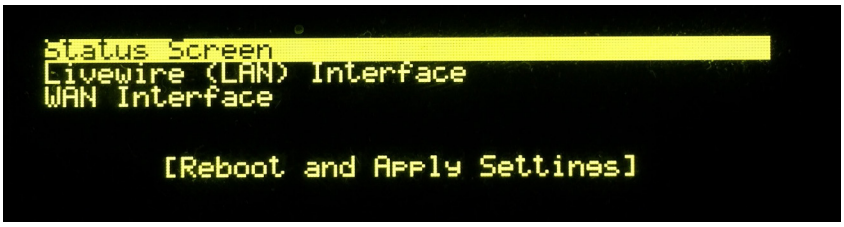
Using the IEC cables supplied, connect the Pathfinder Core PRO's power supply input to AC mains.

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

IP Address Configuration

You will need to enter the IP address, netmask, and gateway you wish to use for the system. To do this:

1. Tap the rotary encoder on the front panel, once. A menu view will appear.



2. Rotate the encoder to highlight "Livewire (Lan) Interface", and tap the encoder again to select it. A new menu will appear.

3. Tap the encoder to enter the "IP Address" setting.

4. Rotate the encoder to move the onscreen cursor. A tap on any of the cursor positions allows you to edit its numeric value; rotate the encoder to increment or decrement the value and tap to accept the change. Repeat this until the IP address you desire is entered. When you scroll beyond the last digit of the IP address the system will return to highlighting the main menu items.

5. Rotate the encoder to highlight "Netmask", and tap the encoder to select it. Rotate the encoder to scroll through the possible Netmask settings. Tap the encoder again to return to the main menu items.

6. Rotate the encoder to highlight "Gateway", and tap the encoder to begin changing the gateway settings. Repeat the steps you used to set the IP address to set the Gateway address.

7. Once the gateway has been set, rotate the encoder again to highlight Save, and tap the encoder to save the "Livewire (LAN) Interface settings.

8. Rotate the encoder to "WAN Interface" and repeat the procedures above to set the WAN IP address.

Important Note: Core PRO only allows a single gateway and will use whichever gateway address is most recently set on LAN or WAN. It is recommended to use the same gateway for both. Having multiple gateways could cause the system to arbitrarily select which one to use with unpredictable results so this is disallowed.

---

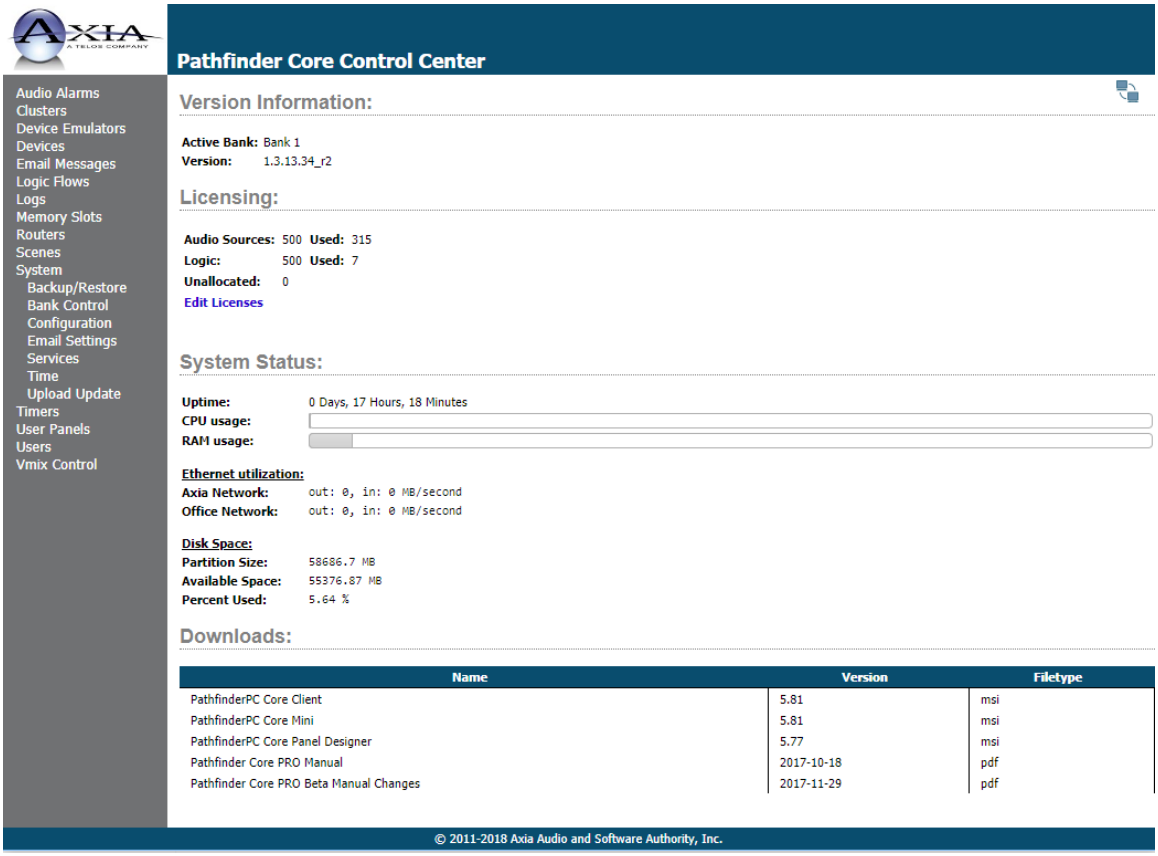
9. Once the WAN IP address settings have been assigned, rotate the encoder to select Reboot and Apply Settings.

Discovery

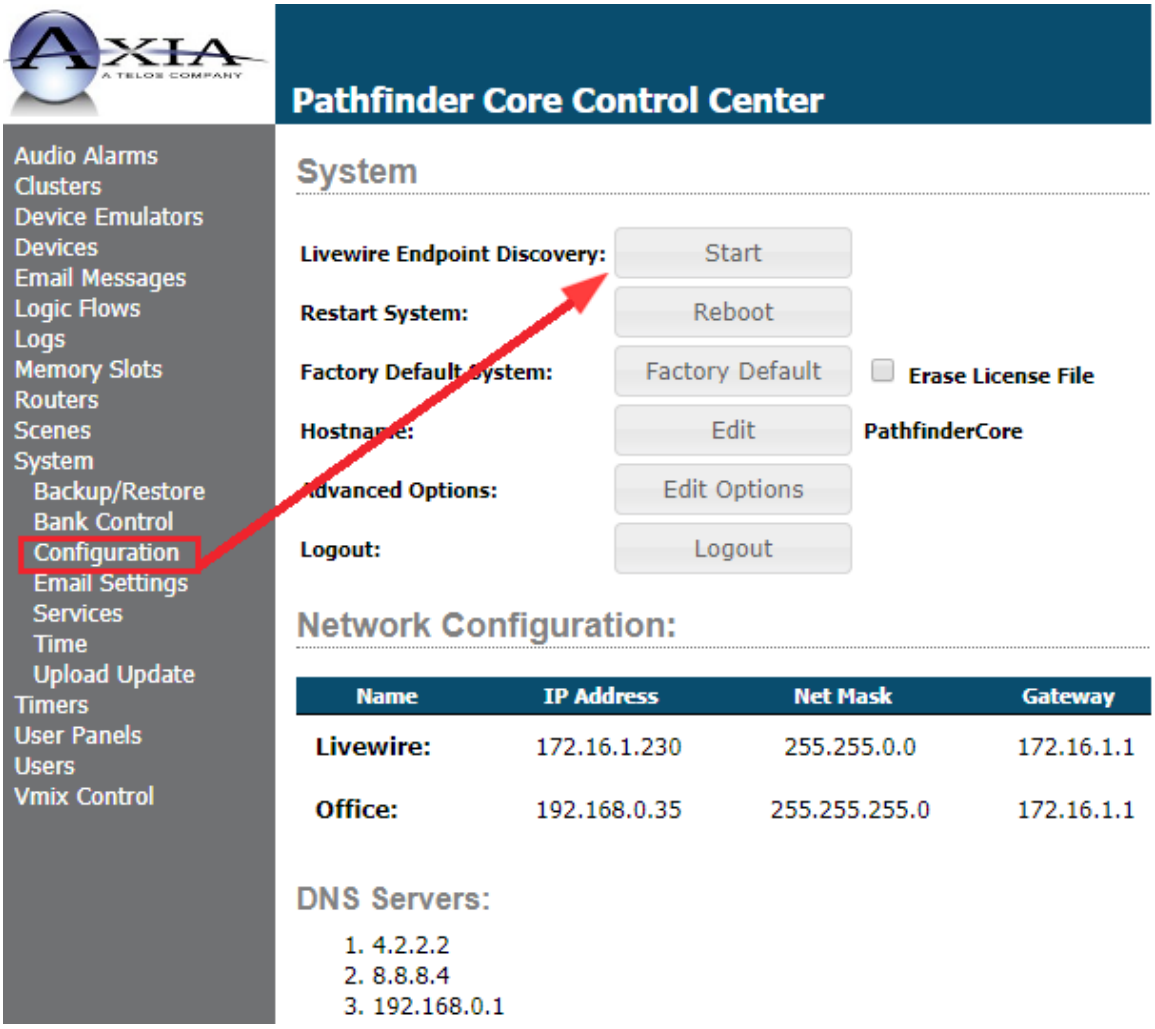
The next step requires a Web browser on a PC connected to either the office or Livewire network. If your computer is connected to the office network, then open your browser and enter the IP you assigned to the WAN port into the URL bar. If your PC is connected to the Axia network, then open your browser and enter the IP you assigned to the livewire port into the URL bar. An authentication window will appear. Enter the following values:

- Username = Admin
- Password = Admin

Select “OK” and your Pathfinder Core PRO control panel will be loaded.



From the links in the left hand navigation bar select Configuration under the System heading. A new page will load.



Next to “Livewire Endpoint Discovery”, click on the Start button.

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.

As long as the Discovery is running, the system will continue to look for new devices. There is no complete message, but if you click on the Devices link in the navigation bar, you should start to see devices being populated into that page.

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

Chapter 2

Overview

Pathfinder Core PRO is designed to offer you the ultimate in 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 all of the 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.

But 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 allow your users to have the precise degree of control and information over the system you wish them to have. 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 a number of 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. That browser should provide the most consistent and best user experience.

If data does not seem to display properly, or if data is missing in whatever browser you are using, it is sometimes useful to try refreshing the web page using the refresh button. Many browsers also have a key combination that will force a refresh from the serving device rather than perhaps relying on cached data. For example with chrome you can hold the Ctrl and shift keys while clicking the refresh button to force the browser to reload from the server rather than relying on cached data. If you find things that do not work or behave properly in a specific browser or are not updating 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.

Navigation Bar

All of the 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. 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 indicates that there is a problem with the communications between your browser and Pathfinder Core PRO. Check the network connectivity between your browser and Pathfinder Core PRO and that Pathfinder Core PRO is booted and functioning properly. An additional symptom of this problem may be that configurations look as if they were blank. If that happens before panicking that your PathfinderCore 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

Search:

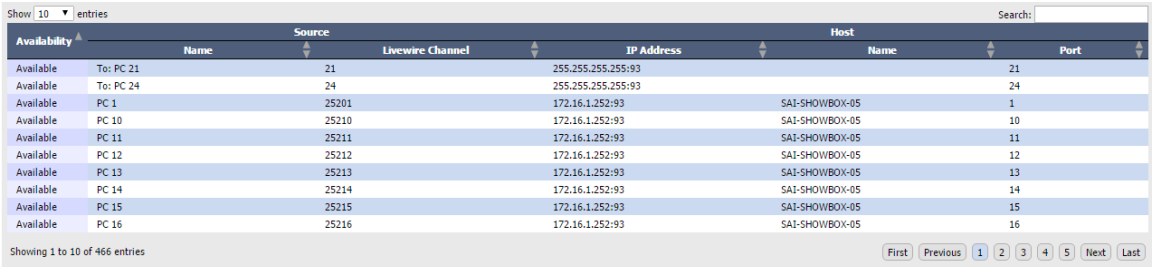
Name	Online	Description	Type	Address	Webpage
AAMD-001-99	true		MicrophoneNode	tcp://172.16.1.99:93	http://172.16.1.99
AAND-001-081	true		AnalogNode	tcp://172.16.1.81:93	http://172.16.1.81
AAND-001-97	true		AnalogNode	tcp://172.16.1.97:93	http://172.16.1.97
AGND-001-085	true		GpioNode	tcp://172.16.1.85:93	http://172.16.1.85
AMND-001-083	true		MicrophoneNode	tcp://172.16.1.83:93	http://172.16.1.83
Element	true		Element	tcp://172.16.1.51:93	http://172.16.1.51
Engine9AD67A	true		Engine	tcp://172.16.1.61:93	http://172.16.1.61
EngineE40FA8	true		Engine	tcp://172.16.1.63:93	http://172.16.1.63
LWPS80236E	true		PowerStation	tcp://172.16.1.53:93	http://172.16.1.53
Node-01	true		XNodeCombo	tcp://172.16.1.95:93	http://172.16.1.95
NodeImitator_100	true		AnalogNode	tcp://172.16.250.1:93	http://172.16.250.1
Omnia8x-001-069	true		Omnia8x	tcp://172.16.1.69:93	http://172.16.1.69
PSIO-001-055	true		PowerStationIo	tcp://172.16.1.55:93	http://172.16.1.55
Qor	false		Qor	tcp://172.16.1.59:93	http://172.16.1.59
Rcp17ss	true		Rcp17ss	tcp://172.16.1.87:4010	http://172.16.1.87
SAI-SHOWBOX-05	true		WinDriver	tcp://172.16.1.252:93	http://172.16.1.252
satestserver	true		WinDriver	tcp://172.16.1.254:93	http://172.16.1.254
ZephyriPort	true		Zephyr1Port	tcp://172.16.1.71:93	http://172.16.1.71

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. 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 data in the list that you are expecting to see, 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.

Many of the lists will have the hyphen sign in the last column of the list. This indicates that the particular item in the list may be removed from the system by clicking that icon. This is the delete icon. 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 plus sign will usually open another dialog box with specifics for how to add a new item into the system.

As lists expand beyond what will fit on a page there may also be navigation buttons. For example:


Show 10 entries Search:

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 First Previous 1 2 3 4 5 Next Last

In this example, there is a drop down in the upper left corner that allows you to define how many entries you wish to see on a page and a set of buttons in the bottom right corner that allows you to navigate through the pages of the list. The bottom left corner may also display the total quantity of items which are being shown. It is important to understand that these navigation tools are subject to the search field. In the case above if we were to enter PC 14 into the search field, then we may get only a single entry in the list and the page buttons would only show a single page.

Event System Catch 22

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

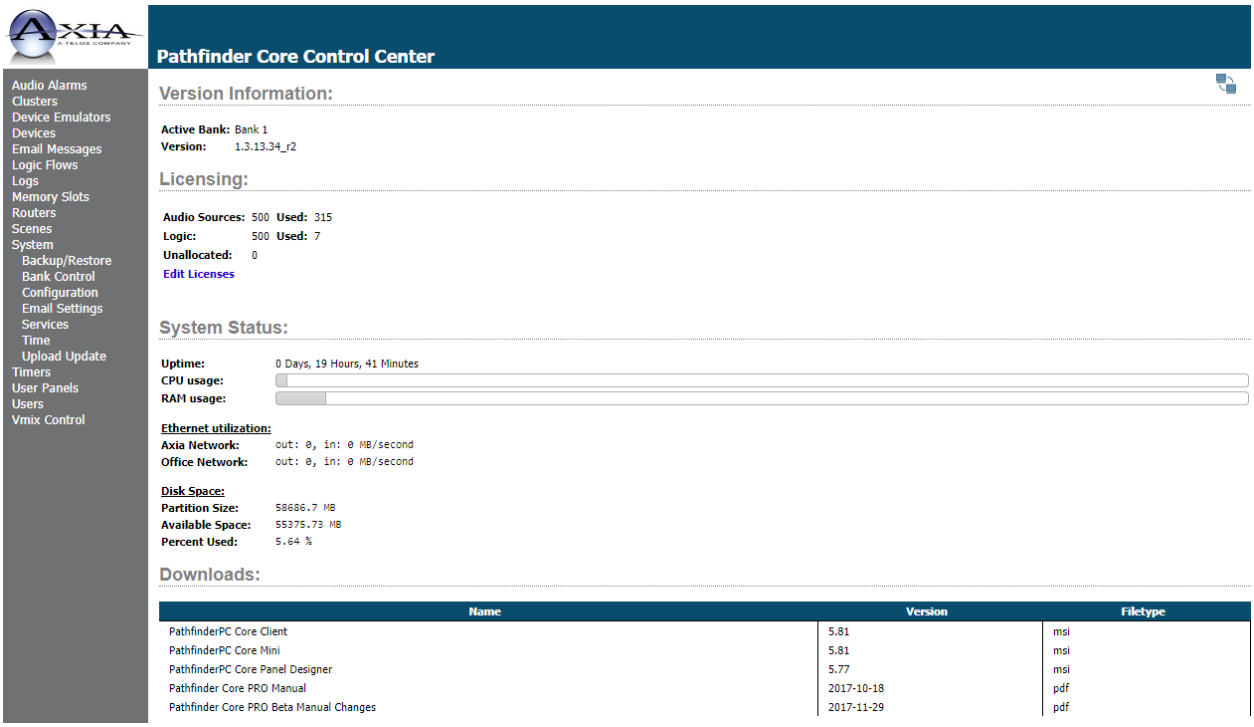
There are certain subjects such as Memory slots that are primarily useful in the context of Logic Flows. So you have to understand Logic Flows in order to make the best use of memory slots. However, the examples and discussions within the Logic Flows section of the manual also will refer to memory slots in order to present some of its examples. Wherever possible, we 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.

Chapter 3

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 that will display when clicking the System link in the navigation bar.



This screen is informational only and does not provide anything that you can actually configure other than the licensing. There are three sections.

Version

This shows which of the two firmware banks is currently active and the version of software that is 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.

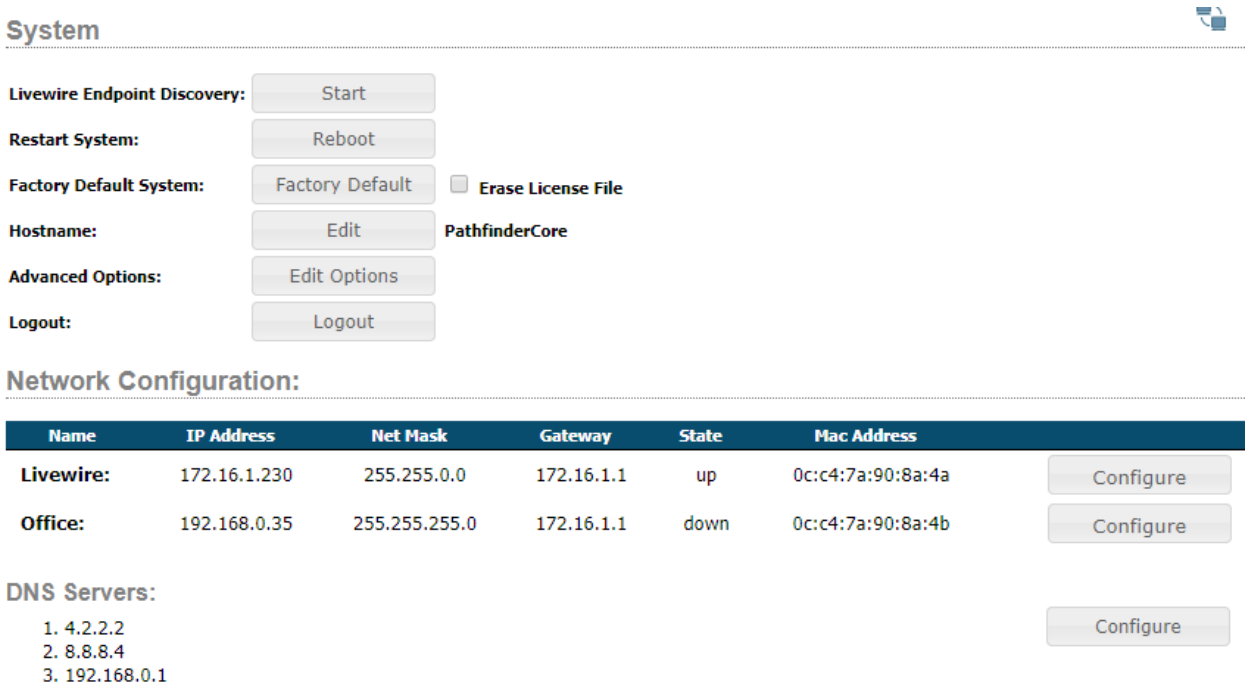
Downloads

This section provides download links for client applications that may be used with the Pathfinder Core PRO system. These client applications may be installed on any windows systems in your network. It is important to note that these links may change when you upgrade the firmware in the system. After upgrading the firmware, it is always a good idea to check this page and see if there are upgrades to the client applications. In most cases, older clients will continue to work with new firmware versions, some functionality might be restricted as new features are added. More information about these client applications can be found later in this manual. Additionally, it should be noted that we are actively working to move all functionality that currently exists in these applications into the browser at which point these applications will most likely be deprecated. So wherever possible use the web browser functionality over the clients in these download links.

These links also provide access to this online manual.

Configuration

Under the System heading in the navigation bar is a configuration link. If you progressed through the Quickstart section of this manual, you will already have been introduced to this page.



Livewire Endpoint Discovery

Clicking the start button on Livewire endpoint discovery will enable Pathfinder Core PRO's ability to detect new devices on the network. 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 the discovery is enabled the button will say Stop instead of Start indicating that clicking the button will Stop additional automatic discovery.

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 also why we recommend setting up your basic Axia infrastructure first, before introducing Pathfinder Core PRO to the mix.

Restart System

If you click the "Reboot" button, the system will ask for confirmation to make sure the button was not accidentally clicked, and it will then cause the system to reboot.

Factory Default System

If you click the "Factory Default" button, the system will ask for you to confirm the decision twice as this option will erase all configuration data and return the system to the way it was shipped from the factory. It is highly recommended that you make a backup of your system and download it to your local computer before using this 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 may require a call to support to relicense 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 to the system. Hostname changes require a restart to take effect.

Advanced Options

This section if for use in consultation with Axia support only. It provides methods for changing certain system functionality variables.

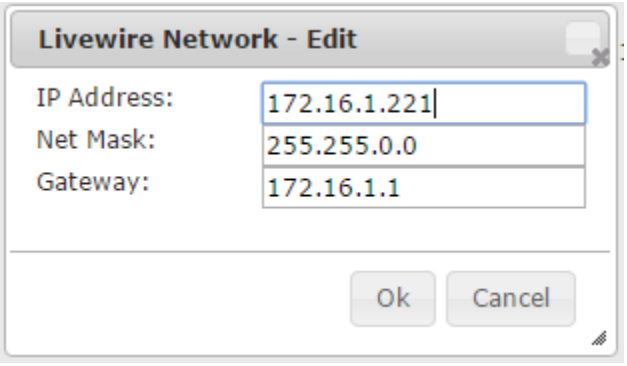
Logout

Logs the current user out of the system and requests new login credentials.

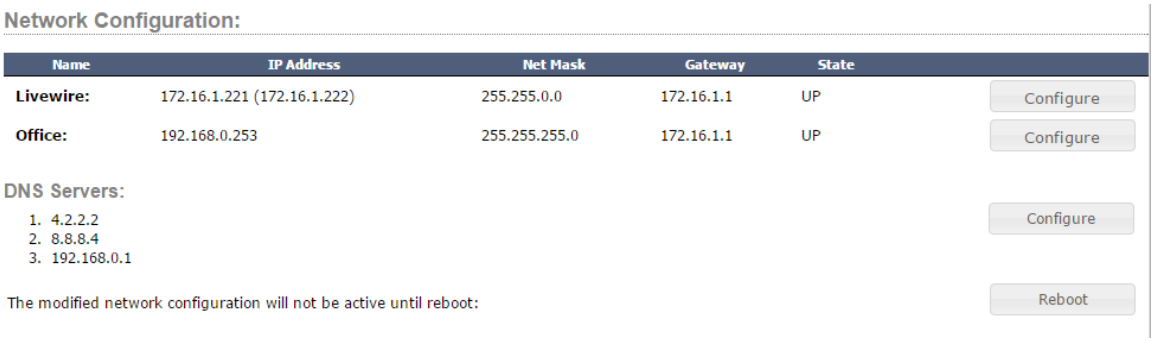
Network Configuration

Livewire and Office

In addition to configuring the IP addresses of the system via the front panel display, they can also be configured using this web page. This section displays the current IP address information for each of the network interfaces in the system. Clicking on the Configure button will present a dialog for editing the appropriate network interface information.



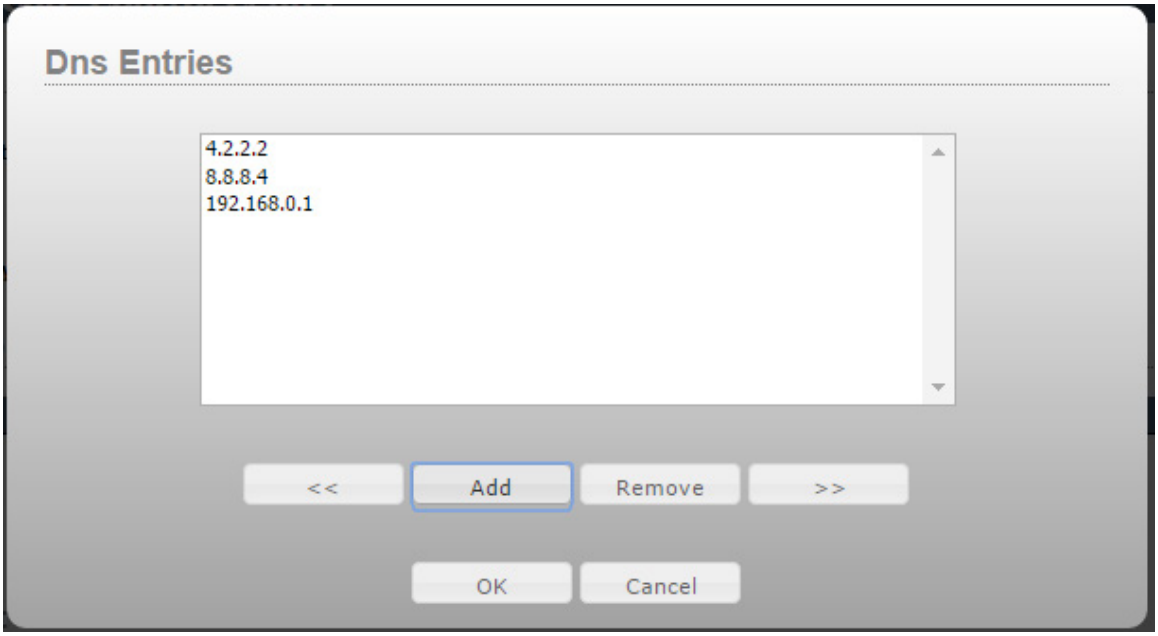
Enter the appropriate IP address, netmask, and gateway and click OK. Or, click cancel to leave this dialog without making any changes. IP Address changes will not take effect until you reboot the system. Therefore, after changing the IP address you will notice that the configuration screen will now show the IP address changes in parenthesis and a new reboot button will appear.



DNS

The DNS section lists the DNS servers defined in the system. DNS is important because the system can send emails to alert administrators and employees of changes in your network that require attention. For example, you can send emails when critical sources or destinations such as air chains become silent. In order 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 what DNS servers to use.

If you click on the Configure button you will be presented with a dialog that will allow you to add and remove DNS servers and also change the order in which the system will attempt to use them.



Click the Add button to add a new DNS server to the list. Select a DNS server in the list and click Remove to remove a DNS server. Select a DNS server in the list and click the << or >> buttons to move it up and down in the priority list. Click cancel to leave the dialog without making any changes, or OK to save your changes to the system. DNS changes do not require a reboot to become active.

Licenses

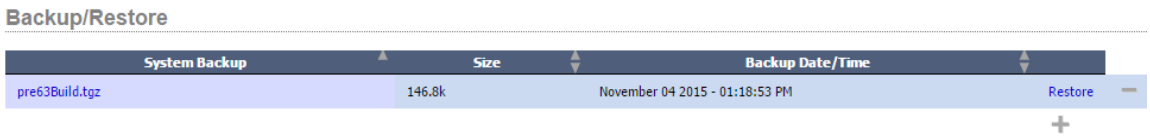
Each Pathfinder Core PRO system should come from the factory with a base license that allows 500 Axia audio sources and 500 logic flows. Additional licenses to add functionality may be purchased through your Axia distributer. Each additional license provides an additional 500 Axia audio sources or 500 logic flows. Each hardware Pathfinder Core PRO requires a base license. But the additional licenses are shared between the Pathfinder Core PRO units that are participating in a given clustered system. This means that if you need a redundant Pathfinder Core PRO system that supports 1000 Axia audio sources, you will need to purchase two Pathfinder Core PRO systems (each of which comes with a base license), and then 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 a router with 1000 sources. Please note that virtual sources and gpio sources do not count against the license. Only Axia audio sources whose streams are enabled and logic flow endpoints are counted in the licensing. Additionally, 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.

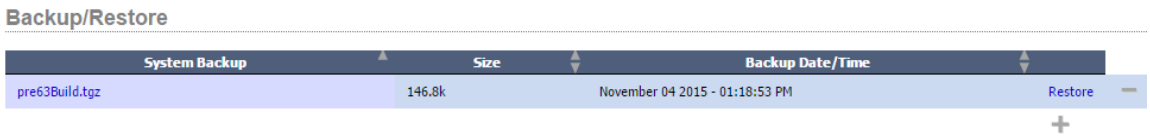
In order to view or add additional licenses, click on the System heading on the navigation bar and then click the "Edit Licenses" link.

The licensing page will display your base license information. It will also provide a list of your additional licenses. Click the New License line and enter the new Request Code and License Key to add a new license. Then select whether the new extended license should be applied to routes (sources) or logic flows. Finally click Submit Changes to add the license to the system. License changes will require a reboot to become effective. Click the Configuration link under System to find the reboot button.

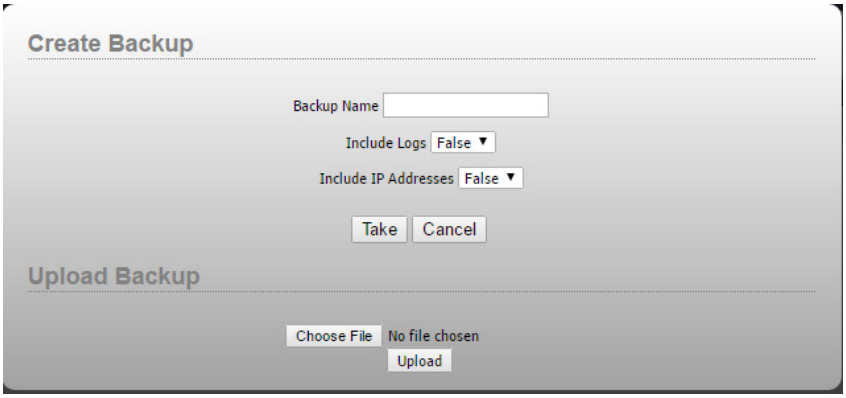
Backup/Restore

The backup and restore page presents a list of all backups in the system.



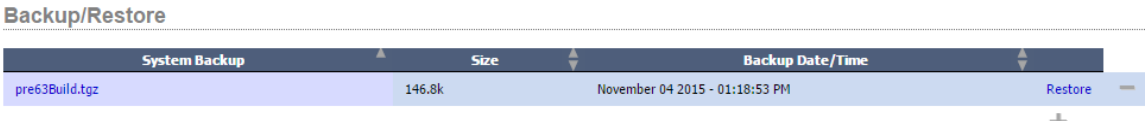


If this is the first time using this page, there will probably not be any backups present in the system. To make a new backup, or to upload a backup from your local computer back into the system, click the plus icon.



In order to create a new backup, enter 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. If you are intending to restore the backup on a different Pathfinder Core PRO, you may not wish to include the IP address so that the restore process does not overwrite the IP address of the other unit. Clicking Cancel will exit the dialog without making any changes. It is important to note that the backup will include any configuration files needed to return your system to its current state. If logs are included it will also include all system and application logs that are currently on the system. The log information can be very useful to tech support if they are trying to assist in troubleshooting an issue. It is likely they will ask you to take a backup including log files and send it to them.

In the case you are trying 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 the choose file and select a Pathfinder Core PRO backup file on your local system. Then click upload. After doing this the backup will be available in the list of backups on the preceding screen.



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.

The minus icon will delete a backup from the system. Since there is limited space on the storage medium it is only recommended to keep a few backups on the system at a time.

Clicking the restore icon will request confirmation that you really wish to restore the system to that state and then it will shut down the Pathfinder Core PRO services, restore the configuration files to the state of that backup, and reboot the system.

It is very important to understand this process. While the restore function is usually pretty quick, the system will not be functioning during the restore and reboot operation.

It is also important to note that backups live in the software bank. Therefore, moving to a new software bank will not display the backups from the previous bank. This is another reason to copy the backups to your local computer. You can then upload them back into a new bank if desired.

Upload Update

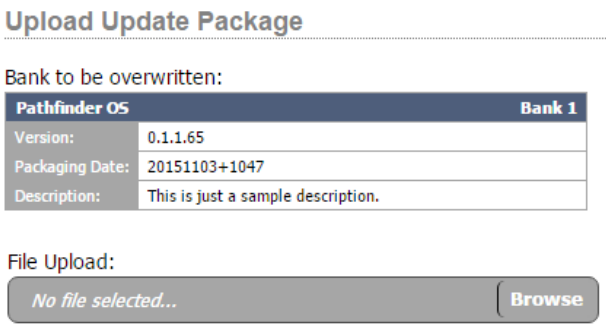
One of the most important tasks the Administrator of Pathfinder Core PRO will have to perform is periodic firmware updates. This project is undergoing a very active development lifecycle with many new features planned for future versions. That, in addition to fixes for any bugs that are reported to us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

The system has two banks for software versions so that if there is a need to return to an earlier version, it will live in the preceding bank. The process for upgrading your system is as follows.

- Take a backup of your system as explained in the previous section of this manual. We always recommend backing up your system before any major change just to be safe. The backup process will actually generate its own backup as well, but it never hurts to have a safety stored on your local computer.

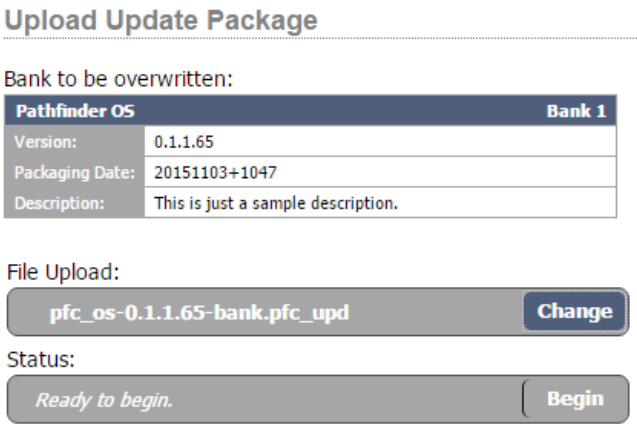
- Download the new firmware version from the Axia Audio web site to your local computer. The file name extension will be .pfc_upd which stands for Pathfinder Core PRO update package. It is also a good idea to take a moment to scan the release notes to understand the changes between the version you are 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 executing. The update will always update whichever bank is not currently executing. So if you are currently running on bank 0, the update process will update bank 1.

- Click the Browse button and select the file you downloaded from the Axia web site.



- The File Upload bar will change to the name of the file which you have selected to upload to the system 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 installing vm software to a Fanless Engine platform) a warning will appear.

- Click Begin to begin the updating process.

- A new page will appear with a progress bar that presents stages and some additional information regarding what is taking place at each stage of the upgrade process so that you can monitor the progress of the update. The update may take several minutes to complete.

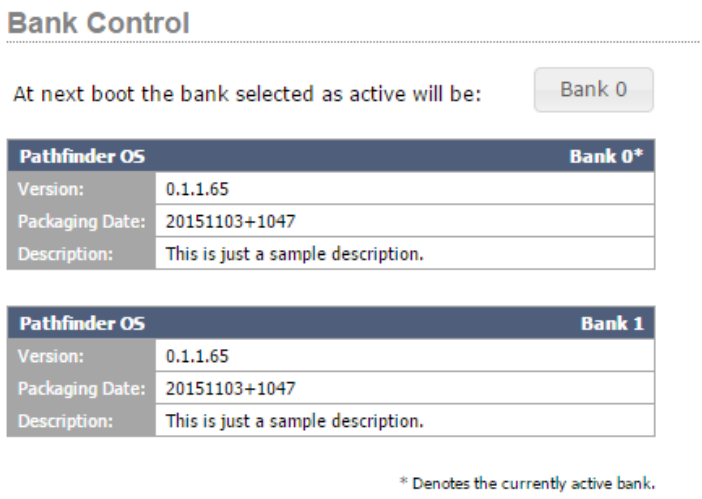
Important Note: It is important to understand the status of the configuration files during an upgrade process. The configuration files for the system reside within the currently executing bank. During the upgrade process, a backup of the configuration from the currently running bank is automatically 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. If you receive an error message instead of processing succeeded, please contact support. Because you are updating the bank which is currently not executing, the system should continue to run even if there was an error with the update.

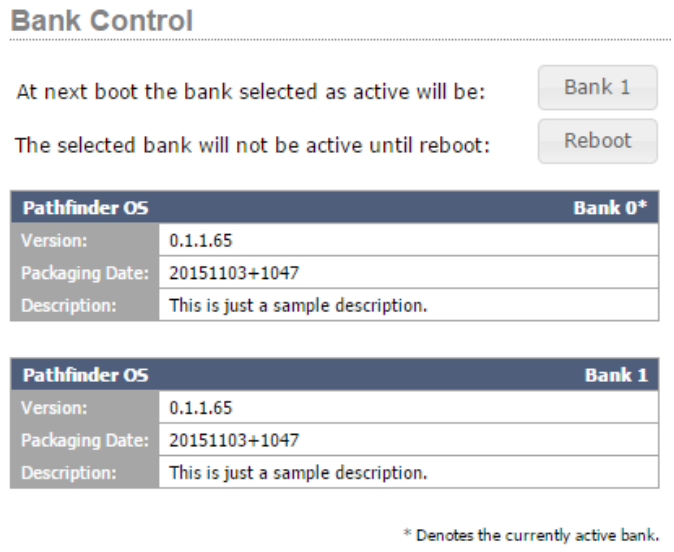
- In order to actually boot into the updated bank and start using the new software, click the Bank Control link in the navigation bar.

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.



The bank button can be used to switch which bank will be used on the next reboot. It also displays the bank that is currently selected as the next bank that will be booted at reboot. After changing the bank that will be used on next reboot, a reboot button will appear on the page.



Clicking on the reboot button 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.

It is important to understand some features of the bank selection process. 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. The reason this is important is that 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.

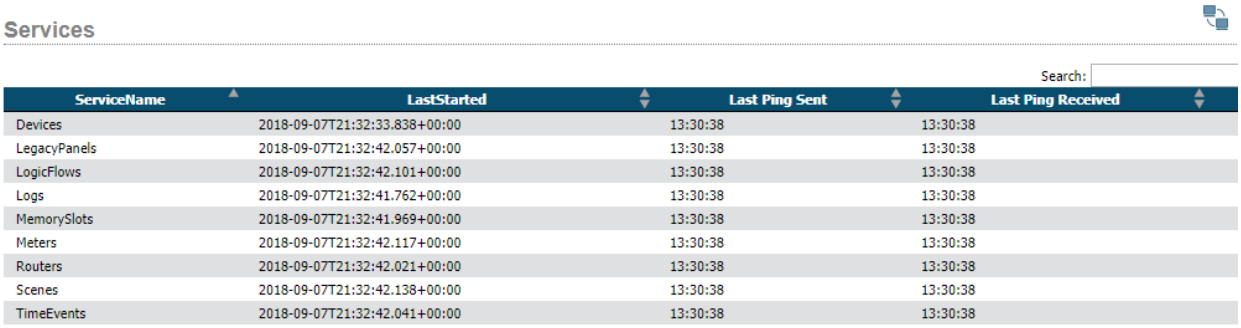
This also means that after rebooting into a new bank you should return to the bank control web page to make sure the boot was successful and you are executing on the desired bank. If the boot fails, it could fail back to the previous bank and you will only know that by double checking this page.

Important Note: We have occasionally seen some browsers try to cache the state of this page even though we have requested in the web page code that the browser does not cache this data. So 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 and Cntrl while clicking the refresh icon to forcefully reload the page.

It is also important to understand the state of configuration files between bank changes. Configuration files reside within the executing bank. Therefore, if you are switching to a different bank the configuration may be different. So it is 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 has 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 that 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.



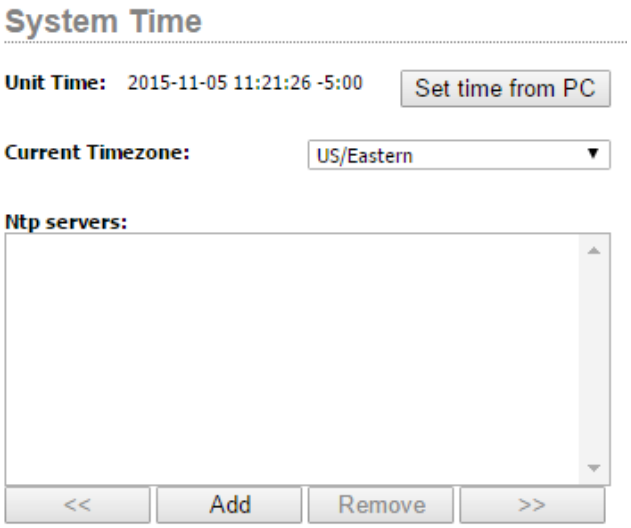
This page will only be used in conjunction with a session with Axia tech support, but it is worthwhile discussing what this page represents. In an effort to make Pathfinder Core PRO as robust as possible, the system has a sophisticated watchdog process. This process monitors the state of each of the services in the system. It 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 after enough subsequent failures, reboot the system.

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.



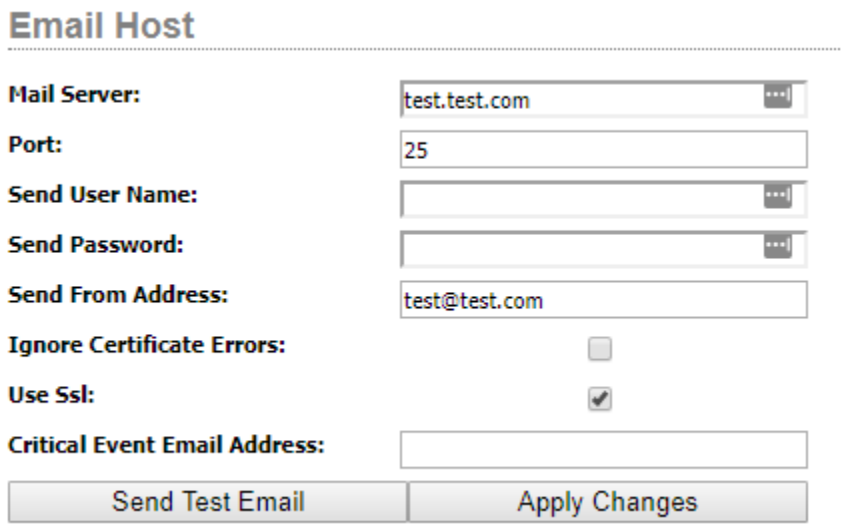
The System Time configuration page allows you to define NTP servers, set the system's time zone, and sync the current time to that of your local PC. Use the time zone drop down to define the correct time zone for the system. Click the Set time from PC button to update Pathfinder Core PRO's current time to that of your local PC. Or preferably enter NTP servers into the list so that Pathfinder Core PRO can update its time automatically and therefore 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 to 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 log when changes happen. And finally, proper cluster synchronization relies on date time settings to determine whether a specific piece of information is more up to date on one system or another and therefore whether a synchronization

of that piece of information needs to take place. Therefore it is highly recommended that you use NTP servers to guarantee accurate time in your Pathfinder Core PRO system. 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 on the Email Host link.



These settings are similar to those used by any other email client application. It is important to note that 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 PathfinderCore PRO to skip including any credentials in the email sends.

> **Important Note:** After applying a blank password, the field will most likely show a series of stars. This does not mean the password is not blank. 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.

The ignore Certificate option also requires some explanation. If you are using SSL encryption and the SSL certificate chain provided by the host is not in Pathfinder Core PRO's trusted list, then the message will not be sent and an error will be generated. Enabling this option will cause the system to use SSL encryption but ignore the SSL certificate error and send the message anyway.

Once you have the parameters configured correctly, click "Apply Changes" to store the parameters to the system. Then use the "Send test Email" button to generate a test email. The page will ask for an email address to send the test message to and will then attempt to send the message. Apply must be clicked before the Send Test Email will use the new settings. 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.

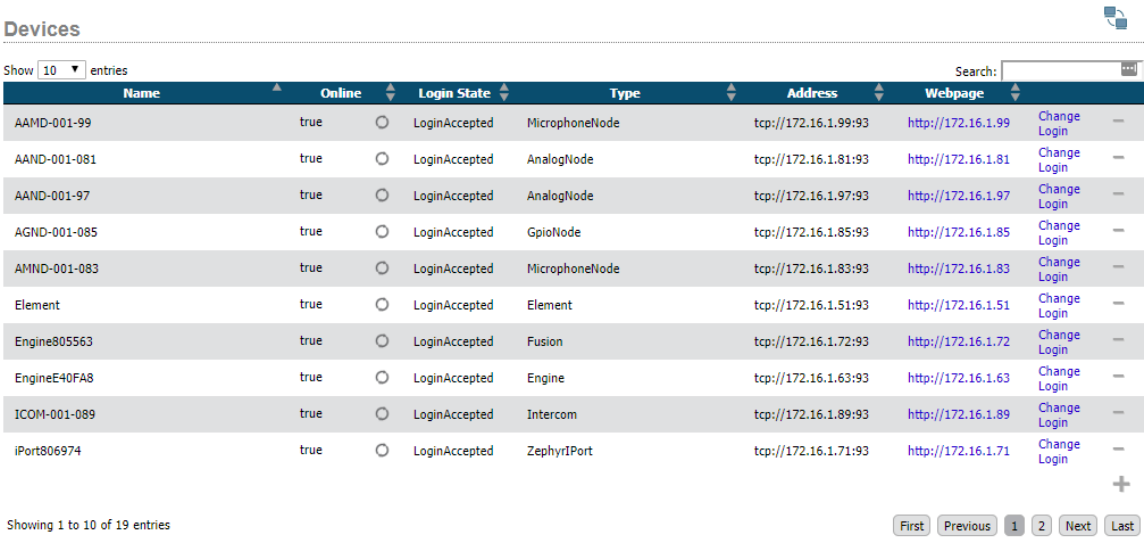
Go to the Email Messages section of this manual for more details about creating email messages which the system can send and how to determine when those messages are sent.

The Critical Event Email Address field may be used to alert an email address of critical issues that take place such as a system restart or a service failure.

Chapter 4

Devices

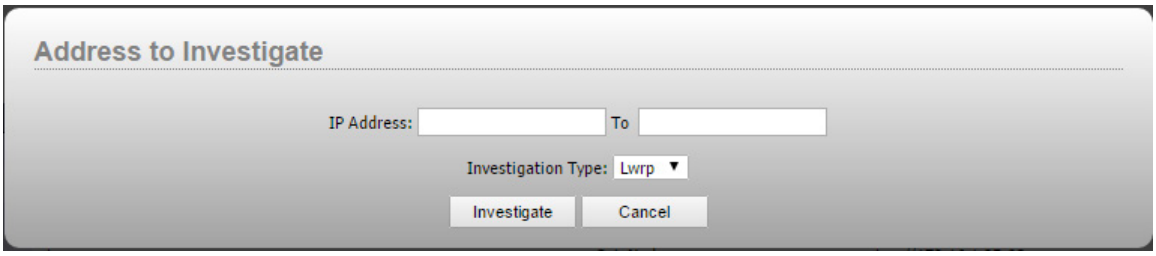
The Devices link in the navigation bar displays a list of all of the devices which Pathfinder Core PRO has discovered.



Each device entry in the list also includes a link that will take you directly to the device's configuration web page. There's another link that allows you to define a different login password required by the device. You can use the minus icon to remove a device from the system. 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 rediscovered and added back in to the list if it is still on the network.

Each device also has a connection recycle icon in the Online column.  This icon should rarely if ever be used. Clicking this icon will cause Pathfinder to drop and reconnect its connections to the device.

There are some devices that are not discoverable automatically. These include the 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.



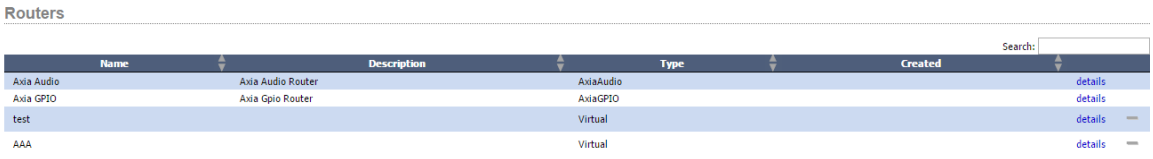
A dialog box will appear that allows you to define a range of IP addresses to scan and the investigation method to use. If you are adding a single device enter the same IP address in both boxes. The “To” box should auto-populate when you enter an IP address into the IP address box. If you are adding an Axia Audio or GPIO device, select Lwrp as the investigation type. If you are adding a rackmount OLED or LCD panel into the system, use the Lwcp investigation type.

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 can be contacted and data can be discovered from the device, then it will be added into the devices list. Its routing resources will also be added into the requisite routers.

Chapter 5

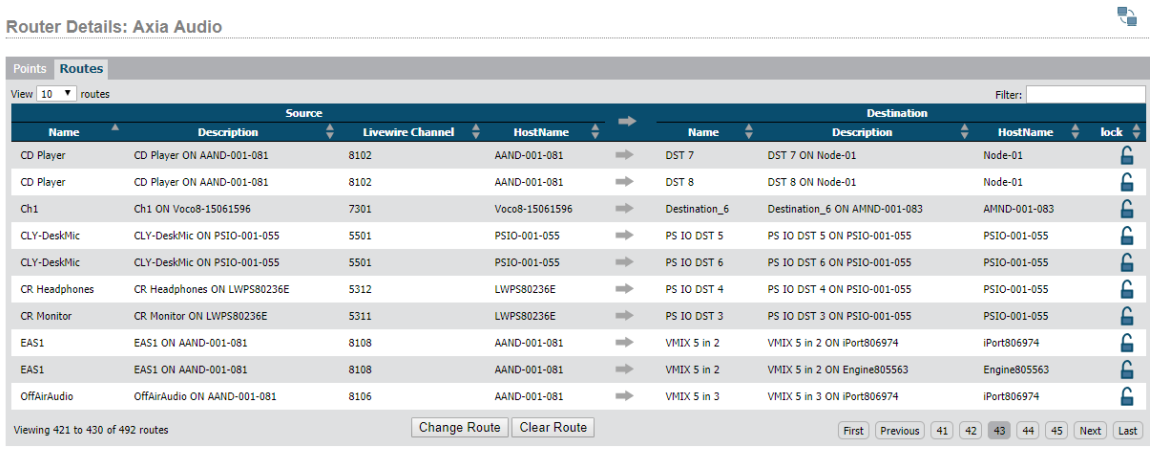
Routers

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



Pathfinder Core PRO currently supports Axia Audio Routers, Axia GPIO Routers, and Virtual Routers. The Axia Audio Router and Axia GPIO Router are automatically generated based on the devices that are discovered in the system. These two routers will always be in the system and may not be removed. Virtual Routers may also be created and removed from this page. More information regarding the different types of routers is discussed later in this section.

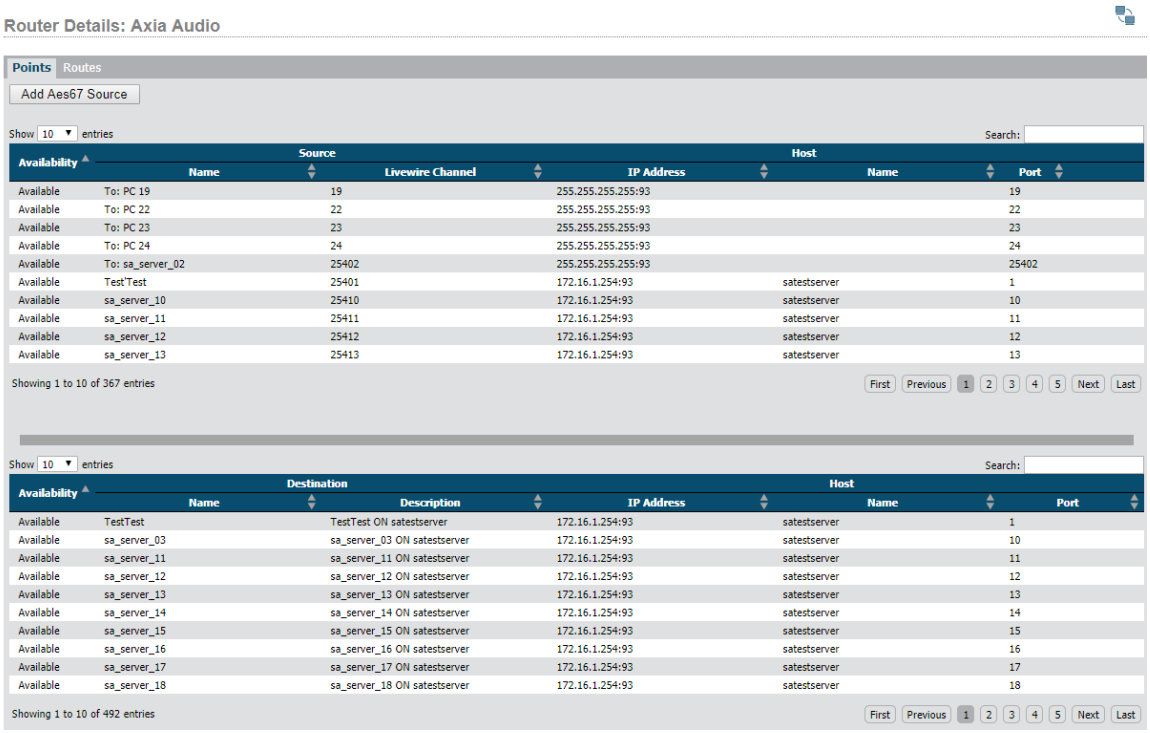
Clicking on the details link for any router will present a new screen with the routing status of the router.



This screen has two tabs on the top - points and routes. Selecting one tab or the other will display the data relevant to that tab.

Points

The points tab simply shows a list of all of the sources and destinations in the router along with relevant information for each source or destination.



The columns in each list are broken into two major sections – Source/Destination and Host. The left half of the table shows columns related directly to the source or destination such as the source's name, Livewire channel, and availability. The right half of the table shows information about the host device where that source or destination resides.

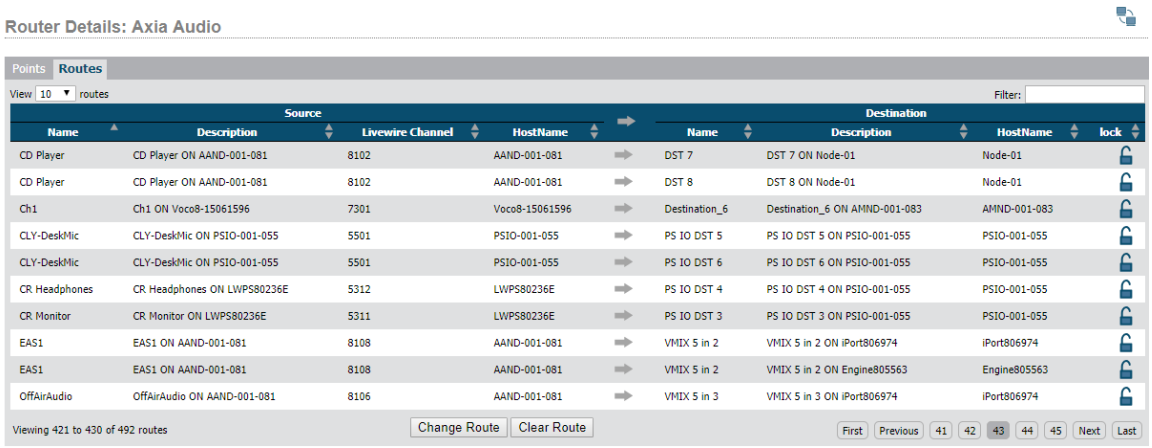
- Availability – This means whether the source or destination is currently available for use. If a device is offline or unreachable, this column may show offline. It may also show "SystemLocked" for destinations whose source cannot be changed by the router. For example, mixer faders cannot be changed using standard routing commands. They must be changed using the console or by issuing special source profile load commands. Therefore, they will display as System Locked.
- Name (Source/Destination) – The Name of the source or destination.
- Livewire Channel (Axia Audio Sources Only) – The Livewire channel number assigned to the source.
- IP Address – The IP Address of the host device on which the source or destination is located.
- Name (Host) – The name of the device on which the source or destination is located.
- Port – The physical (or virtual) port number that identifies the source or destination on the device.

Since these lists can contain a large number of 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.

Routes

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

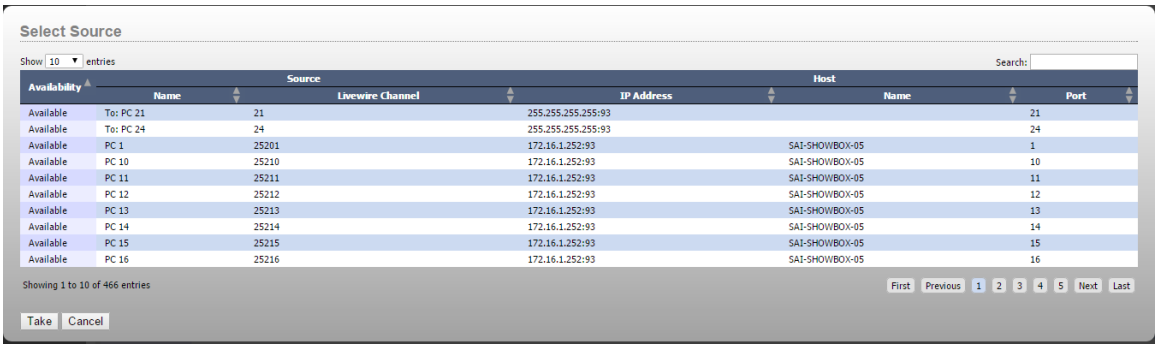


Again, this screen is divided into two sides. The source information for each active route is on the left, and the destination information is on the right. The columns display the information about each active route.

- Source
  - » 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
  - » 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 also be made from this routes tab screen by clicking a line in the list and then clicking the "Change Route" button, or by double clicking on any line 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.

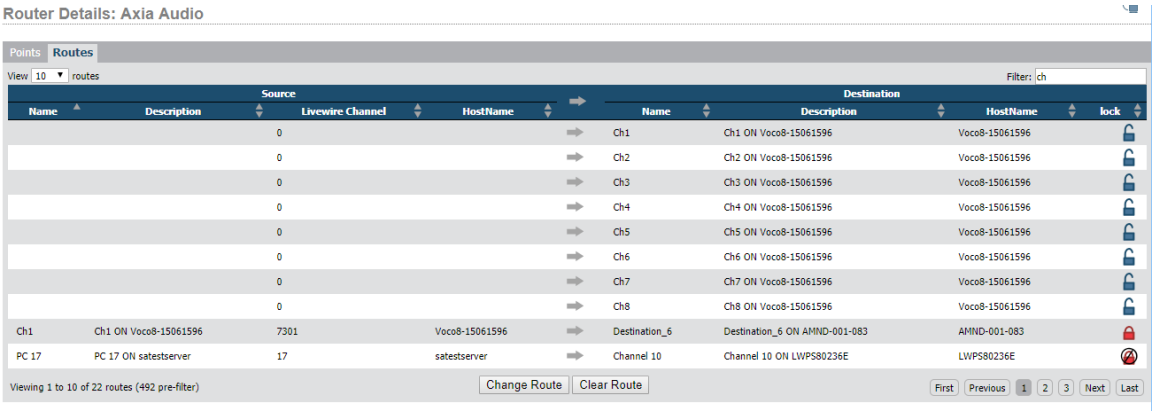


Select the source in the list and select "Take" to change the route. Or simply double click on any of the sources in the list. Clicking cancel will exit the route selection dialog without making any changes. Again, these lists can be lengthy so use the search function to narrow down the selections. The search filter can be remembered 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. It is also important to notice that a source will not be found 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.

Routers can also be controlled by the Pathfinder PC_Core Client application. See the chapter on Pathfinder PC_Core Client.

Locking

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

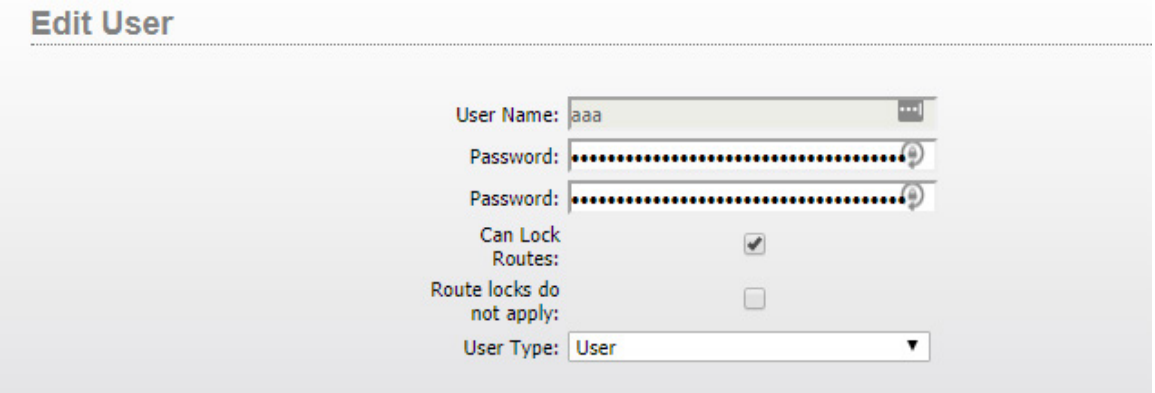


Each lock state is represented by an icon in the Routes table of the router:

Locked:  Unlocked:  System Locked: 

System locked (with the exception of 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.

It is important to note that other than system locks, locking and unlocking is a Pathfinder state and is not actually changing 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. Routes that are locked must be unlocked before they can be changed. In addition it is possible to define whether a particular user has the rights to change route locks. When defining a user, a field now exists to configure this:



Note: The Route locks do not apply option is for future use and not fully functional at this point in time.

Enabling the Can Lock Routes checkbox can be used to define if the user has the rights to lock and unlock routes. This allows an Administrator to lock routes and make them unchangeable by normal users.

Some special notes need to be made in relation to virtual routers. A virtual destination is comprised of one or several 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.

Note: The fact that locking a virtual destination does not lock its underlying base point is different from how previous versions of Pathfinder worked but provides better functionality as described above.

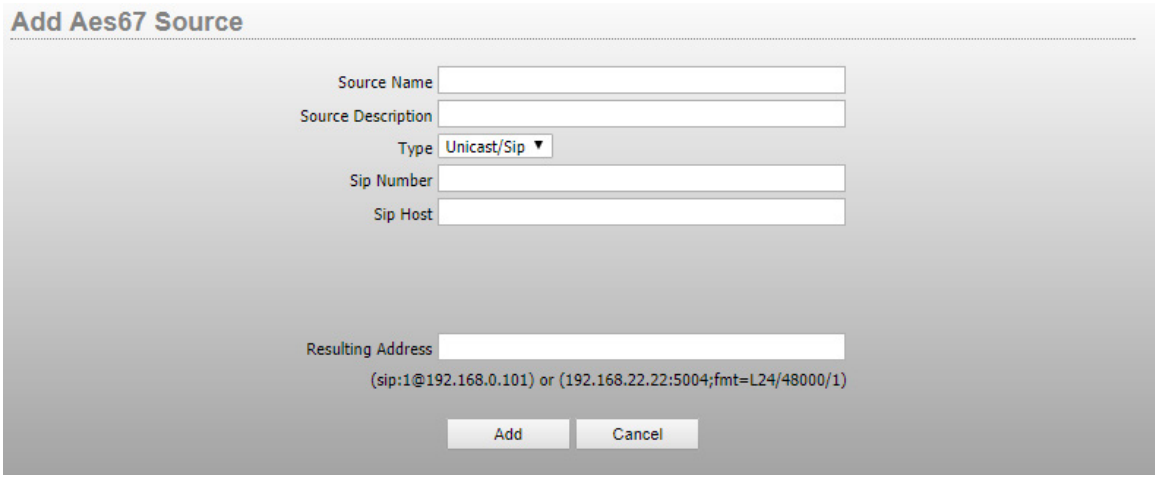
Router Types

Axia Audio Router

The Axia Audio Router is automatically created by the system. As devices get discovered or manually added into 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 router 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.

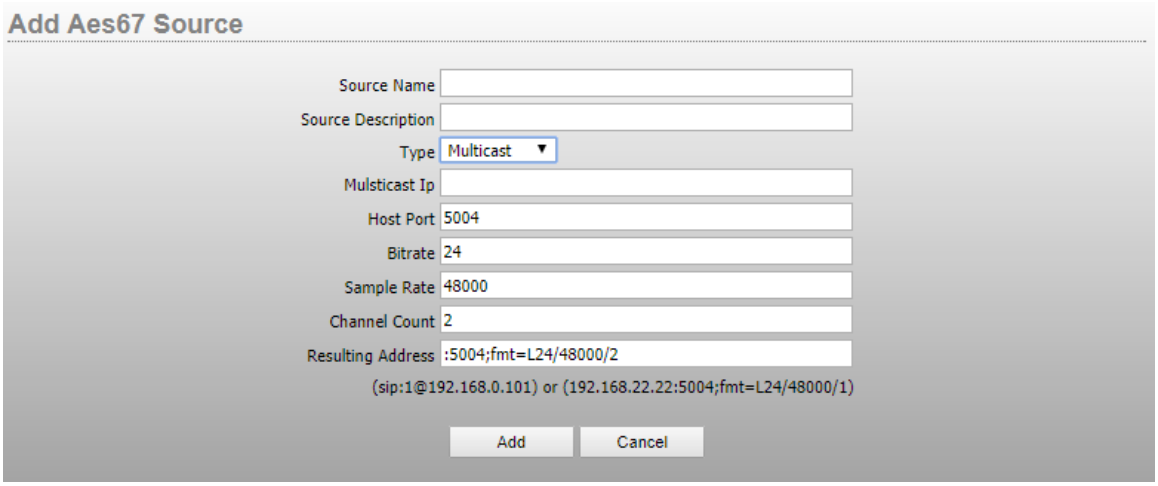
You may notice in the points tab that some sources will carry an IP Address of 255.255.255.255:93 and will have nothing in the host name column. These sources will also have "To:" at the start of the Name. These sources are Axia Backfeeds generated by the mixing engines. They are not assigned a specific 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.



Use the Source Name and Description fields to define the name and description by which the source will be identified in the system. 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 to look like:



In this case fill in the multicast IP address and adjust the other parameters to reflect the Aes67 source settings and 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, PathfinderCore 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 important to understand all of the methods because if you try to use conflicting methods at the same time on a GPIO port, that 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. And these route changes can 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 and then directly trip the closures using Pathfinder's (or some other automation system's) event system by sending closure commands to the device.

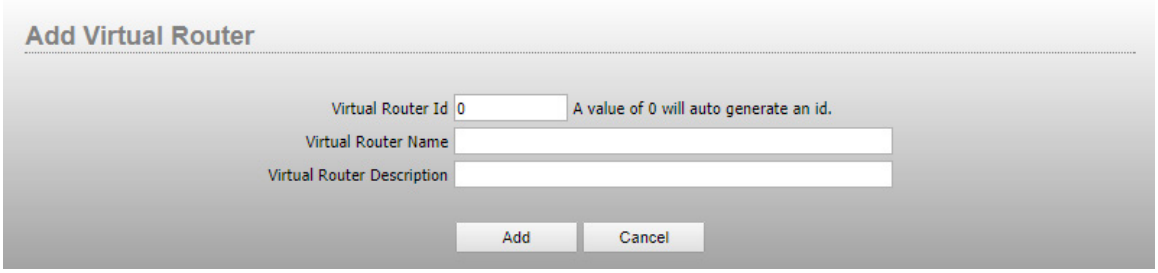
The reason it is important to understand these three methods of GPIO closures is that it is possible to do multiples of them at the same time which is occasionally useful but more often, just a mistake. For example, we often see users who think that the GPIO ports should have a Livewire channel number and so just arbitrarily assign them. Then you can get into a situation where they have 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 calls to Axia Support. It is important to understand how you intend to use a given GPIO port, and configure it correctly for that task.

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 entire network.

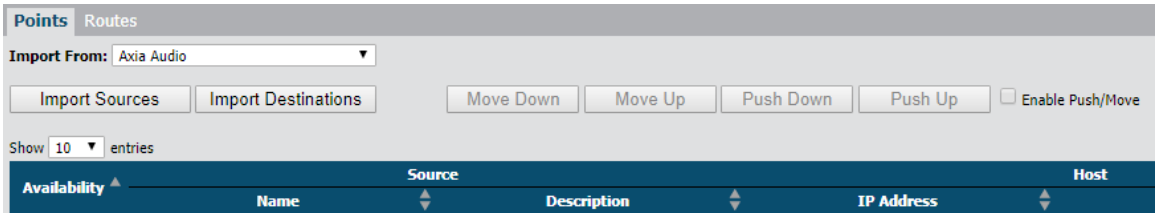
However, virtual routers can also go much deeper than just subsets of routes. For any specific virtual source or destination, you can also associate multiple real sources or destinations with that virtual source. That sounds complicated but actually it is relatively simple and extremely powerful. For 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. You could use a Virtual Router to do this. Create a virtual source that contains both an audio and a GPIO source, and 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, the virtual router will tell the underlying audio and GPIO routers to make both routes.

In order to create a virtual router, click the plus icon on the main routers page.

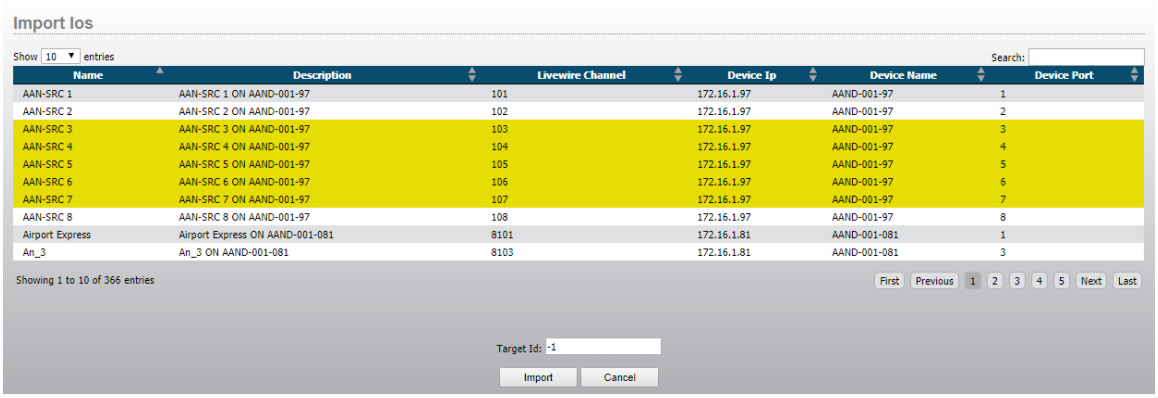


Provide a name and description for the router and click Add. The router id will autogenerate the next available router id number. That can be overridden with your own id number as well. Be careful to use a unique id that is not already in use by another router.

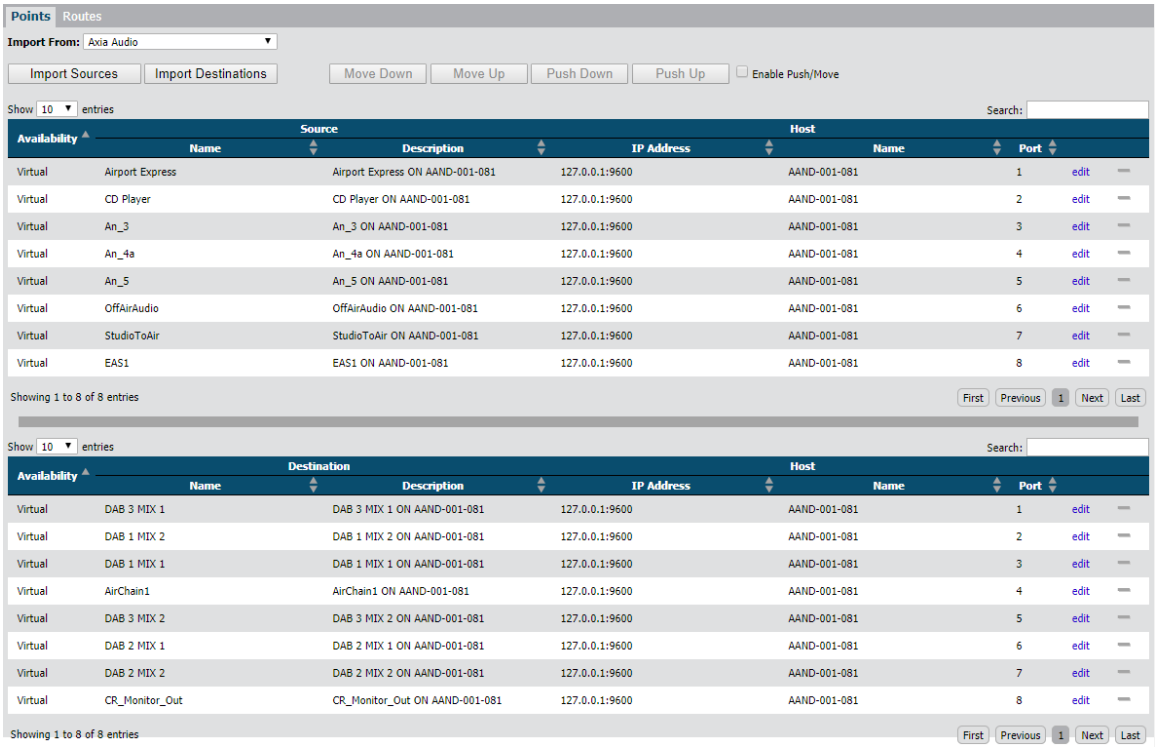
After clicking Add, the virtual router will appear in the routers list. However, this router is currently empty. It does not have any sources or destinations. Click the details button and select the points tab. You will notice that there are several controls at the top of the points tab.



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 from drop down and then click Import Sources. You will then be presented with a list of sources you can select to import into the virtual router.

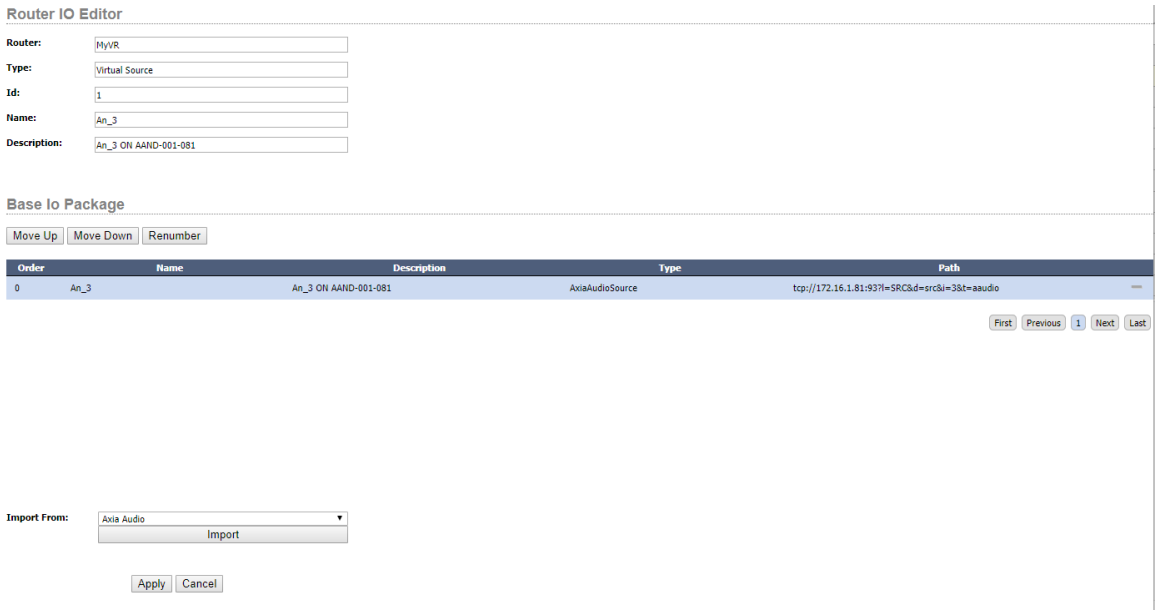


Click on the sources you wish to import (Shift-click to select a range) and click the import button. Repeat the procedure with the Import Destinations button to import destinations into the virtual router. Note that in general you will leave the target id at -1. This will import the IOs to the end of 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.

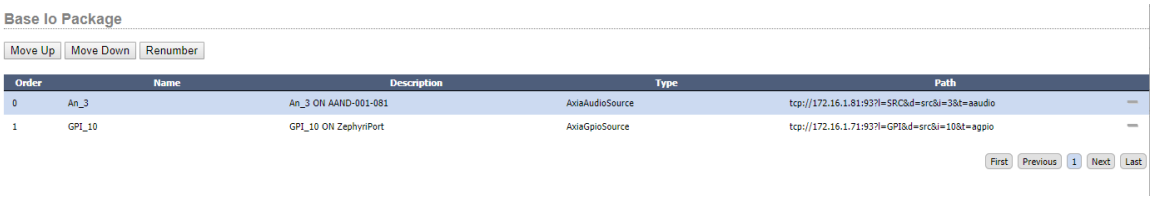


At this point you have a virtual router with only the sources and destinations you want available. This router can then be made available as the only router certain users have access to using the user rights section or it can be used to easily see and change certain critical routes without viewing the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



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. So, for example we could select the Gpio router from the import from dialog, click import and select a gpio source to include in this package.



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 supplies a plus icon for manually adding virtual sources and destinations and minus icons for deleting a virtual source or destination from the router.

Once the router is created, open the routes tab to make and view route changes on the virtual router or use PathfinderClient.

Deeper Tech: Packaging

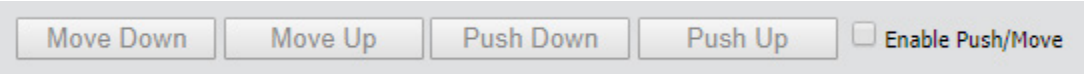
For those who are interested in what happens under the hood with virtual routers, this section is for you. Every virtual source or destination is a package. Within that package is a list of pointers to real sources or destinations. When you make a route change, the system does a number of things. First 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 means another way you can use these packages is 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 creating 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 so, 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, then the route may have two other states. One is “None”. This state means that none of the destination package’s destinations are assigned to anything. The other state is “Other”. The “Other” source 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 specify a 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.

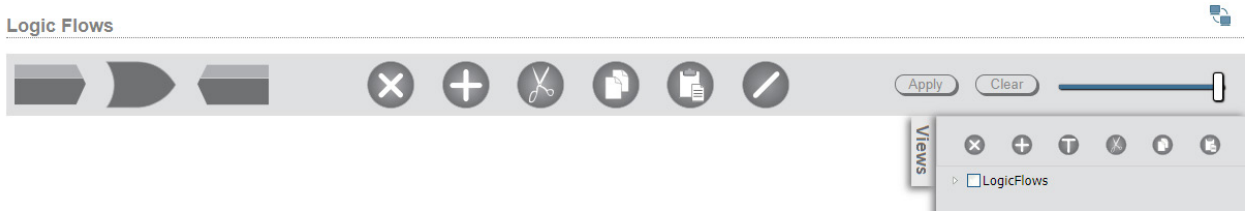
Push Up - moves the IO up in the router effectively decreasing its IO number and pushing any lower numbered IOs it runs into up as well decreasing their numbers until it reaches a hole in the numbering or 0.

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.

Chapter 6

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. Clicking on the Logic Flows link in the navigation bar will display the Logic Flows user interface.

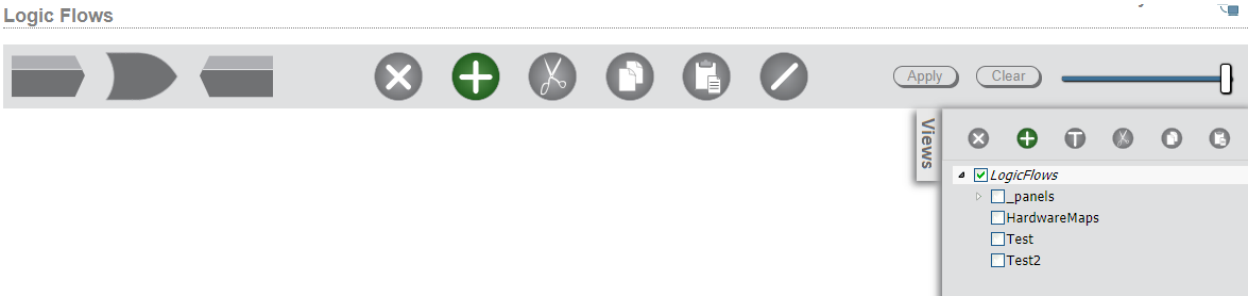


Each logic flow can be thought of as an event. More specifically, a flow is a set of property translations. During an event, we convert an input property value to an output property value.

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

- StartPoint - A property in the system which is being monitored for changes.
- EndPoint - A property in the system which will be changed by the 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.

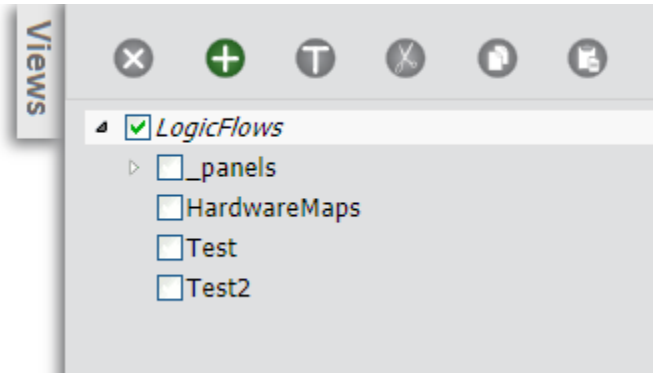
By using these tools, you can build logic visually to accomplish just about any task you want. Let's look at some examples.



When you first look at the Logic Flows Screen you will see a toolbar across the top of the screen and an organizational panel on the right called Views.

Views

The Views tab may slide in and out by clicking on it. You can think of views like folders in your computer's file system. They are organizational containers into which you can place flows. This allows you to keep all of the flows related to a certain subject in the same place. You can select multiple views to view the flows from multiple views together in the same screen. In order to create new flows, you have to select a single view since the system has to know which view to place the new flow into. Like a folder structure, you can nest views as deep as you like. Just check the parent view and then click the Add (plus) button in the views panel to create a new view.



While you can add flows to the master LogicFlows view, it is not recommended as this is the one view that you cannot completely delete. It is recommended that you devise a strategy for your view naming and create the appropriate views underneath Logic Flows to hold your flows. It is also important to note that each view is limited to displaying 100 flows.

It is also important to note that there will likely be 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 and in the chapter on legacy user panels, but it is important to understand that these views are special views managed by the system and should not be deleted.

Views can be deleted by selecting the view and clicking the X button in the Views panel (not to be confused with the x button in the main logic flows toolbar). When you delete a view, the system will ask for confirmation because all logic flows and descendent views of that view will be deleted as well.

The T icon allows you to rename a view and the other three icons allow you to cut, copy, and paste a complete view.

It is important to note that the cut copy and paste tools will only be enabled when a single view is selected.

To copy a view, select the view folder to copy and deselect any other views and then click the copy icon. Once a view is on the clipboard, the paste icon will be enabled. Select a new parent view folder, deselect all other folders, and click paste. The system will ask for a new name for the new view to be pasted, and it will then paste the copied view using the new name. You cannot select multiple views for copy and paste operations though if you select a view with child views those child views are a part of the copy operation. In the example above Test was copied to the clipboard. Then we selected the LogicFlows view and pasted using the name Test2. It is important to note that all pasted logic flows will be in a disabled state. After selecting the Test2 folder you will notice the pasted flows are disabled. 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 endpoints. It is also important to note that view Copy and Paste operations are completed by the Core PRO appliance rather than the browser and so do not require an apply.

Note on copy: 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 PathfinderCore PRO logic flow engine then handles the data cloning.

Cut works the same way as copy except that after pasting, the original view is deleted. It is important to note that the paste creates disabled flows whether the operation started as a cut or a copy. Therefore, if you are cutting and pasting you will still need to re-enable the pasted flows.

Toolbar

The toolbar presents a number of controls. These controls will be enabled or disabled depending on whether it is currently valid to use that control. For example, the scissors icon (cut) will not be available if you do not have a flow selected that may be cut.

From left to right the controls on this tool bar are:

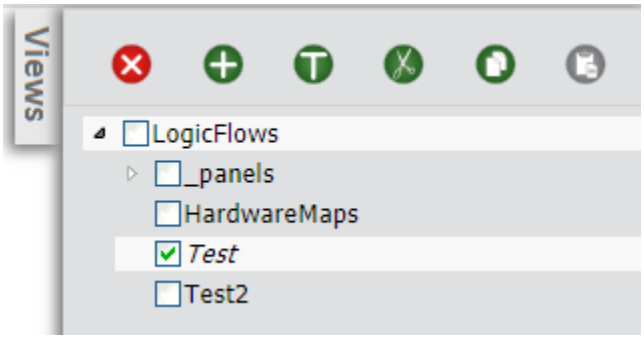
- Starting Translator - Allows you to add a translator to the left side of a combiner or to extend the flow's start to a new flow.
- Combiner - Allows you to add a combiner
- Ending Translator - Allows you to add a translator to the right side of a combiner or to extend the flow's end into a new flow.
- Delete - Allows you to delete a flow or a flow combiner or translator.
- Add - Create a new flow.
- Cut - Cut a flow to the clipboard.
- Copy - Copy a flow to the clipboard.
- Paste - Paste a flow from the clipboard.
- Disable /Enable - Disable or enable a flow or a combiner or translator.

On the Far right side of the tool bar are:

- Apply Button: Applies pending changes to the system
- Clear Button: Clears pending changes and reverts back to the actual state of the system
- Size Slider: Increases of decreases the size of the flow objects in order to display more items on one screen.

First Flow

Understanding how these tools are to be used will be easier if we create a flow and start explaining the process. First, create a view and make sure that view is selected. To do this, click the Logic Flows check box in the Views panel so that it is selected and its checkbox is checked, then click the plus button in the views control panel. The system will ask for a name for the new view and will create that view underneath the Logic Flows view when you click ok. At that point click the newly created view so that its checkbox is checked and click the root Logic Flows view to uncheck its checkbox. Now you should have only your new view selected. In the example below, I created a view called Test.

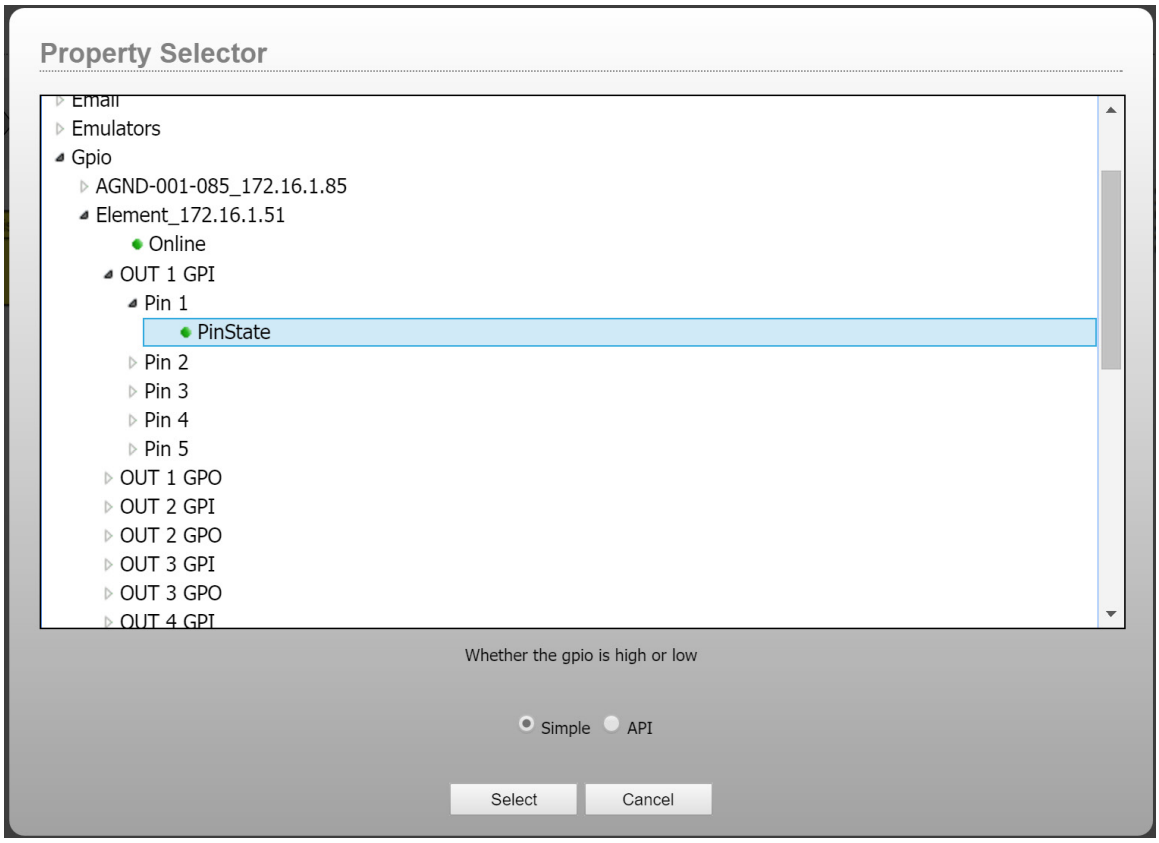


You will notice that once a single view is selected, the Add button (plus) in the main control panel for Logic Flows will switch from its disabled grey to Green indicating that you can now use that control to add a new flow.

If you click on the Add button, it will create a new flow consisting of a start point, a translator, and an endpoint.

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 end point, and one translator. This is the simplest flow you can create. 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. However, double clicking on the translator itself will not do anything until the start point and endpoint 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 will then open up the property selector dialog. This dialog is used to select a property in the system that the new flow will monitor for changes.

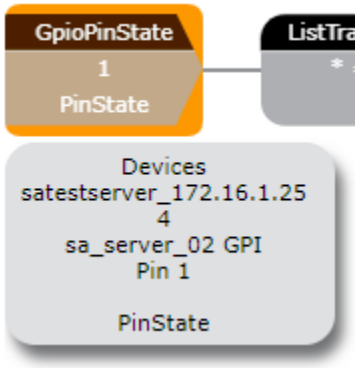


Use the arrows to expand and navigate into the objects in the system until you find the object and property you want to select. Properties will have a green dot next to them as opposed to the expansion arrow. You can either select the property you want to monitor and then click Select or just double click on the property to select it. In this case we are selecting the PinState property of GPI#1, Pin#1 on the Element at 172.16.1.51. It is useful to 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. If you are uncertain about the purpose of a property, selecting the property can provide you with the information you need to understand its purpose.

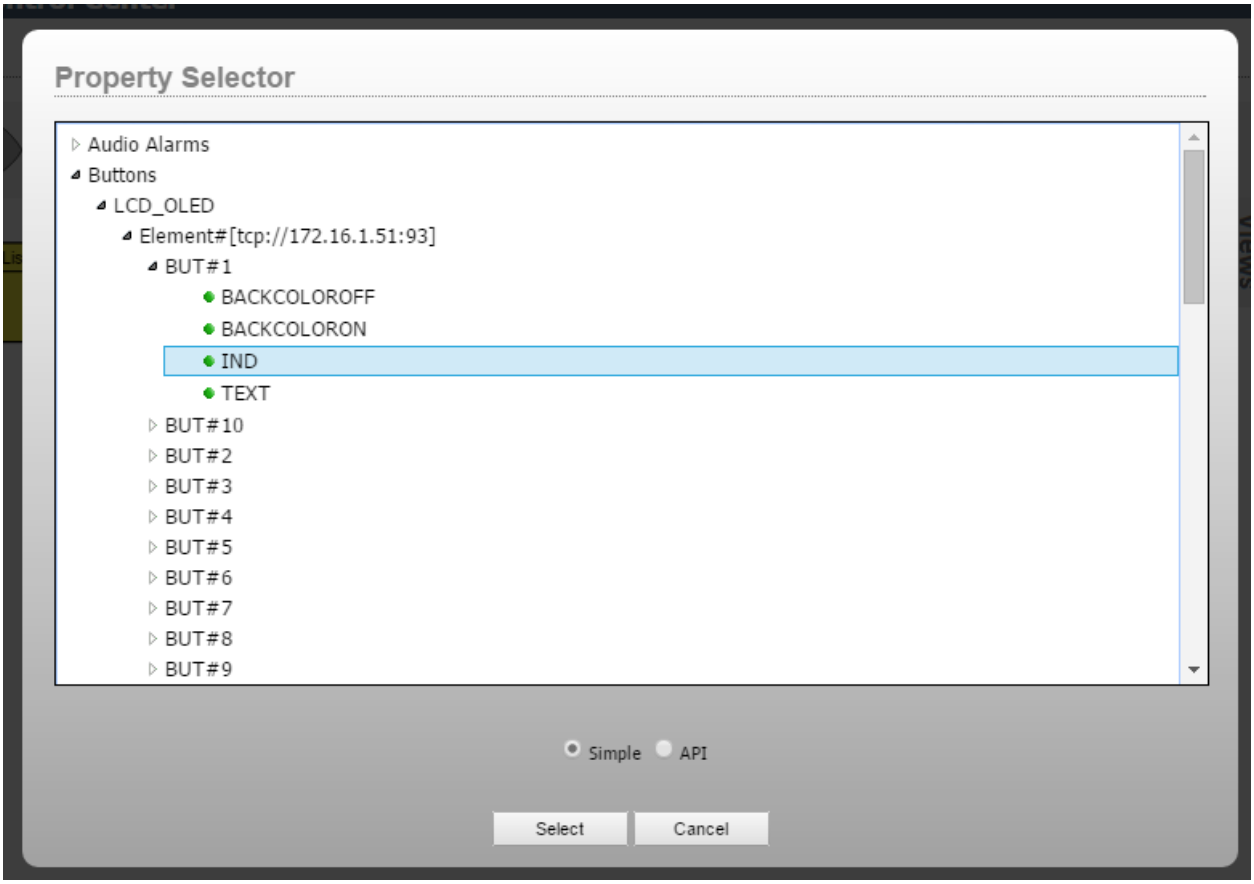
Note: The property selector dialog has a simple list and an API list. The API list is only recommended for advanced users and so will not be discussed until later in the documentation.

Select the start point property by clicking on it and clicking select or alternatively just double clicking on the property. Again, we are selecting properties which will have the green dot next to it. And in this example we are selecting the PinState property of a GPI. Once you have selected the start point property, the block will be filled in with the information you have selected.

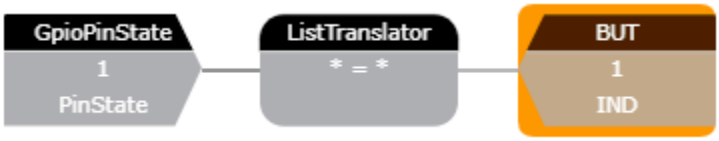
Hovering over the start point that has been assigned a property will also cause a pop-up hover box to appear with more detailed information about the selected property.



Double click on the ending block and repeat the procedure to select the endpoint or the property we want this flow to change. In this case, I am going to pick an LCD button, but you could also pick a GPO in order to make the GPO state follow the GPI state.

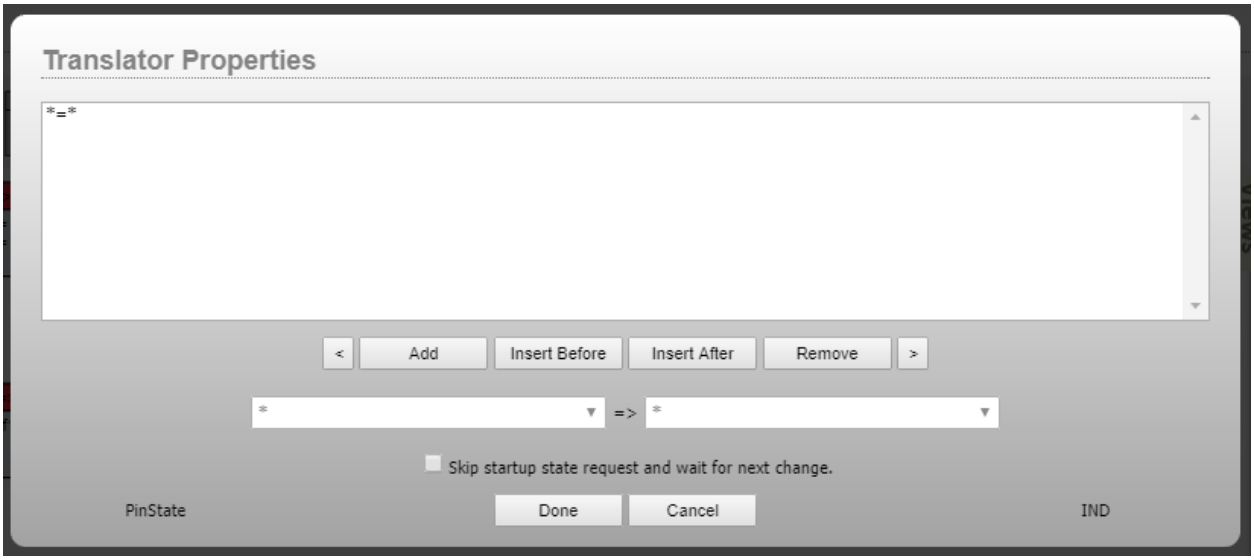


At this point, the flow would look something like:



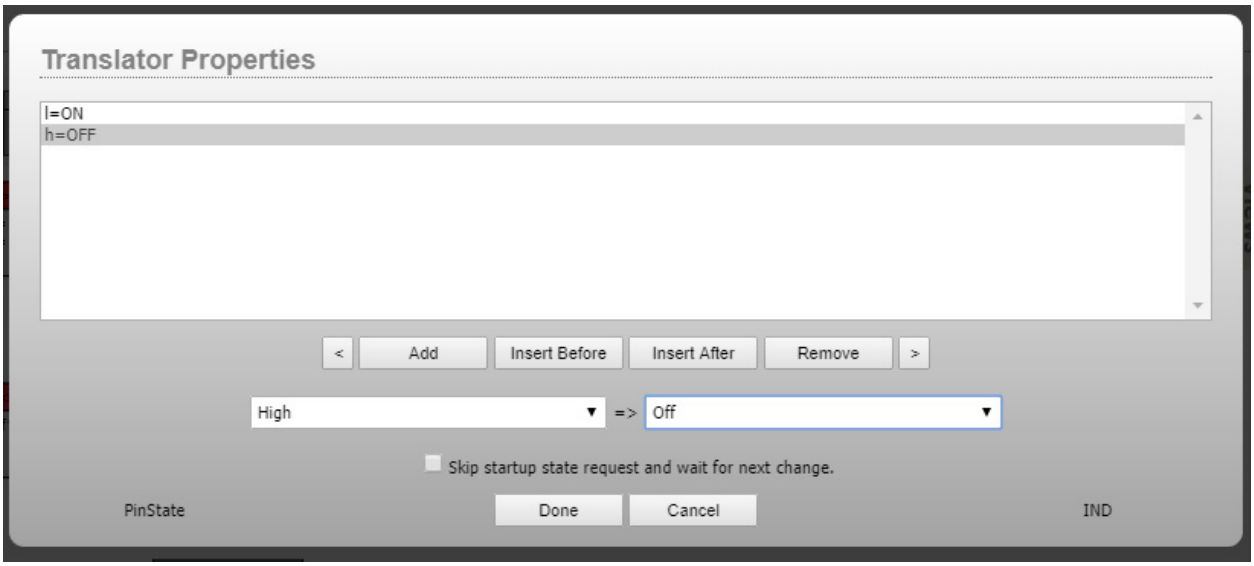
Note: You may find that the list of available properties differs depending on whether you are editing a start or an end point. This is because the system is intelligent enough to know if a property is read only or read write or write only. Properties that are read only may not be used as an end point (something you want to change). So the system will only display properties that make sense for the type of block you are editing.

Once you have a start and an endpoint, you can edit the translation table by double clicking on the ListTranslator block.



This dialog allows you to enter a list of translations which you want to take place. Select the Top translation point (*=*), this will enable the drop down boxes below. You will notice that the drop down on the left side now has Low and High as options, and the one on the right has a list of indicator states such as off, on, flash, etc. Select Low in the left box and On in the Right box. The translation point above will update with your changes. Now click the Add button to add another translation to the list.

For this one, select high in the Left and Off in the Right. It should look something like:



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 endpoint should be depending on what the start point currently is.

Note on the asterisk: The asterisk is a special catchall item that can be considered to be any. 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 than turn the indictor 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 a bunch of 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, then 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 any 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 endpoint or next step in the flow. If no match is found, then the endpoint is not changed.

The reason you were unable to edit the translation block until both the start and end point properties were chosen is because the system tries to help you with the list of possible options depending on the property selected. For example, in this case you got Low and High on the left, and a list of indicator states on the right. If the start point had been the CurrentSource property of a destination, then this drop down would present a list of available router sources.

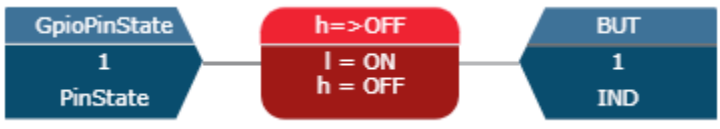
The other buttons on the translation list screen can be used to move translation definitions up and down in the list, insert new ones, and remove them.

The “skip startup state change and wait for next change” checkbox also requires some explanation. By default (this checkbox unchecked) when a flow is applied to the system or the system starts up, the flow asks for the current state of the start point and then applies the correct result to the end point. Occasionally you may not want this to happen when the flow is first created and/or loaded at startup but would rather wait for the next change in the system that would cause the flow to execute. Use this checkbox for these cases.

Once you are done click Done. The flow will now look something like:

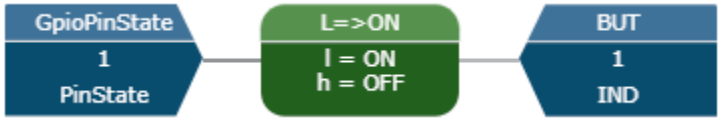
This flow is not active yet. The black and gray states are showing us that this flow is being designed but that the changes have not been committed to Pathfinder Core PRO yet. They only exist as objects in your browser. You should also see the Apply button flashing to indicate that changes are pending.

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:

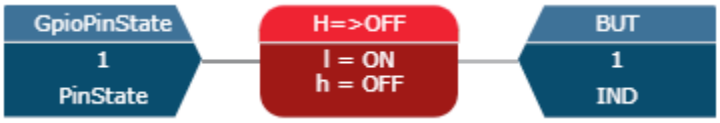


Dynamic Data

The flow diagram does not just 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:



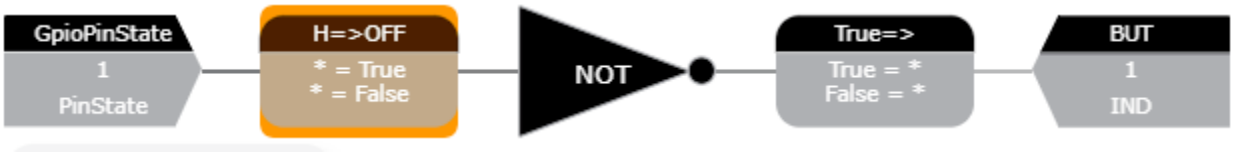
Within the top of the translation box, we see the currently active translation state in text and a color change depending on that state.

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

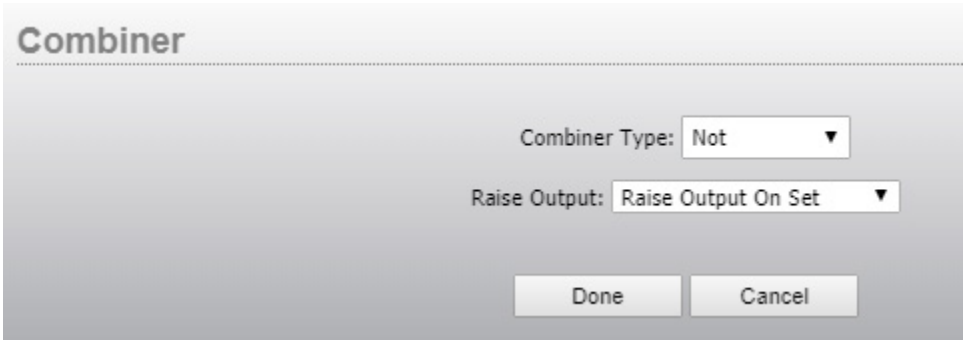
In this way, the system also dynamically displays the state of the flows for easier monitoring and troubleshooting.

Combiners and the second flow

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 endpoints and to make logical decisions based on more than one start point. In order to add a combiner, select a translator in an existing flow by clicking on the translation box in the flow. Once selected the combiner tool in the tool bar will become enabled. Clicking the combiner will insert a combiner into the flow. If you do this on the flow we just created in the example above, it will look like:

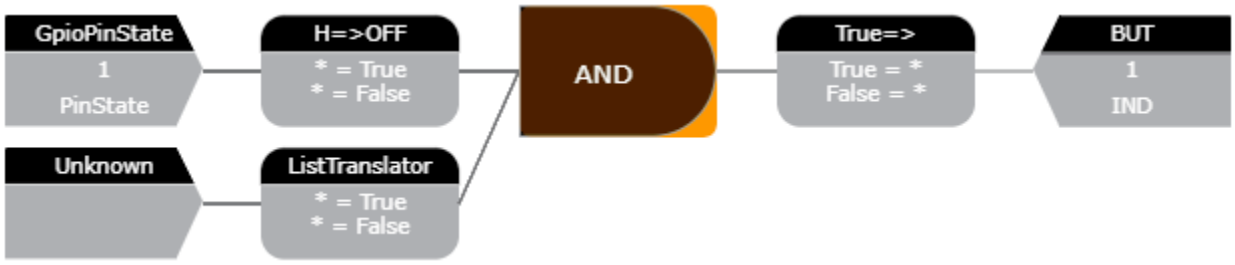


You can change the combiner type by double-clicking on it. This will present a dialog that allows you to select the combiner type you want and adjust any parameters depending on the type of combiner selected.

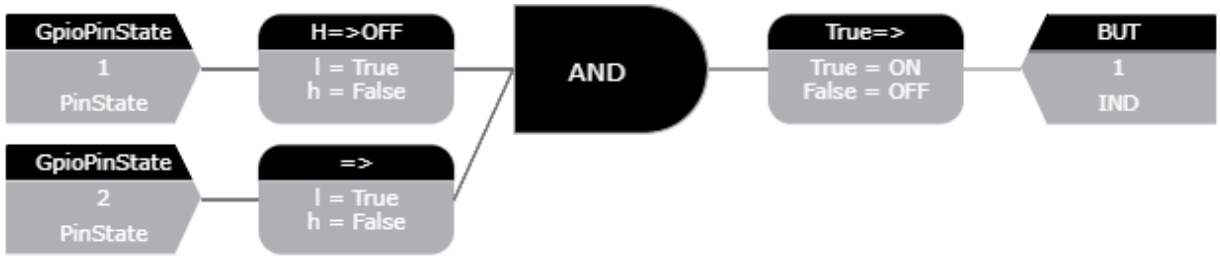


The Combiner Type drop down list will present available combiner types. However, each combiner has a maximum and minimum number of inputs that are required to use them. Only combiners that fit the number of inputs entering the combiner will be available. Currently, you will notice that the only options in the list are NOT, PassThru, and Delay. That is because these are the only combiners that are available with a single input. If you cancel out of the combiner settings dialog and click on the Combiner to select it, you will notice that both the left and right translator tools become available.

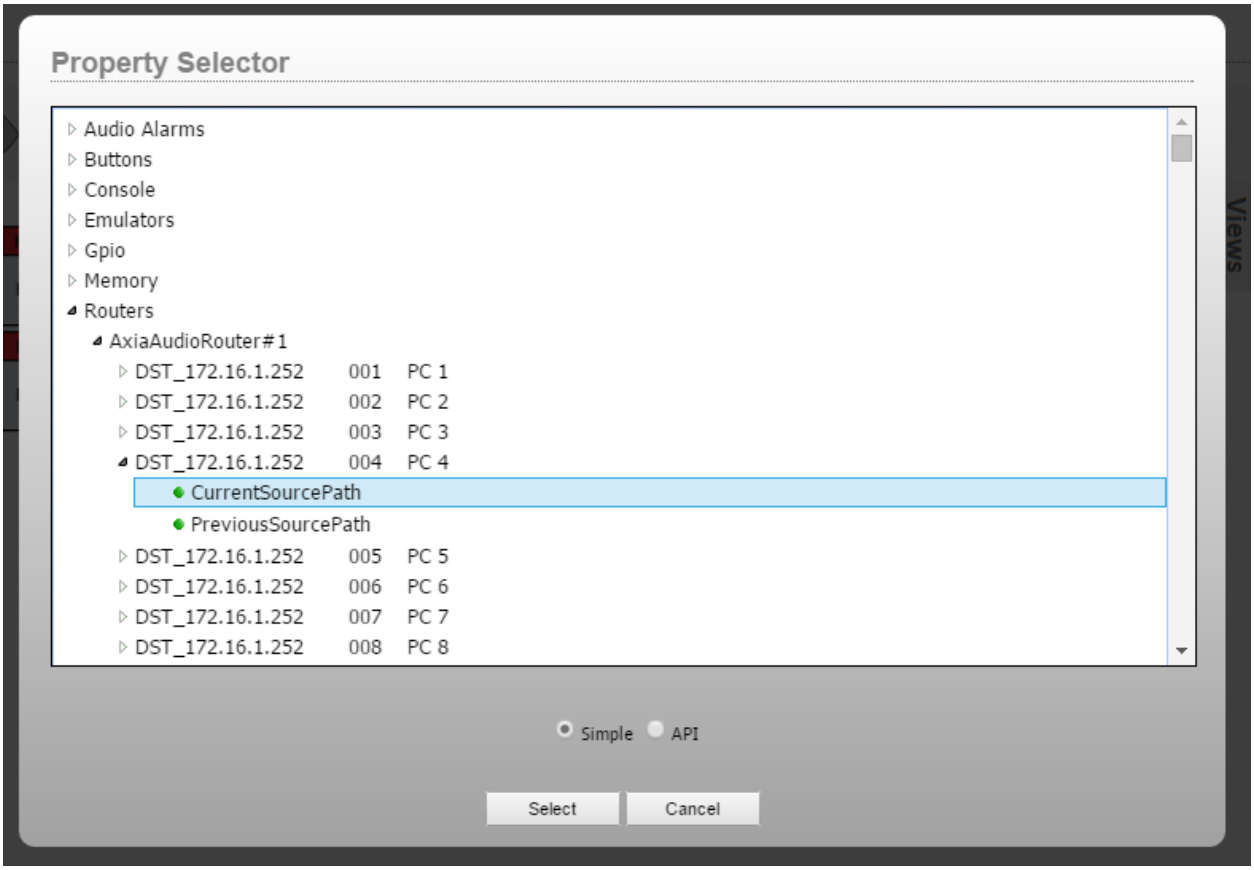
Clicking on either of these will either add a translator to the input side of the combiner or the output side of the combiner. If you add a second input, you will notice that the combiner switches to AND. Double clicking on the combiner to access its parameters now has a much larger number of combiner type options including: AND, OR, NAND, NOR, XOR, XNOR, EQUALITY, and RELAY.



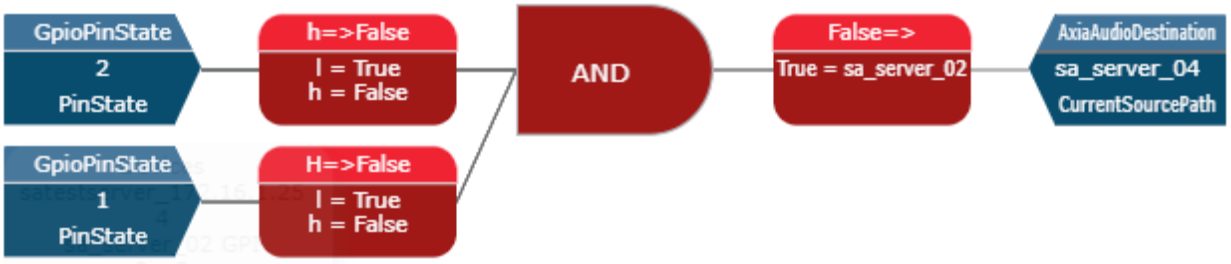
In this case, since the combiner is a logic gate, the translation types are automatically switched to True/False entering and exiting the combiner. By assigning a second GPI pin to the second start point, we can change this event so that two buttons must be held simultaneously in order for the end point change to happen. Edit the various translation lists as described above, and then when complete click apply and you should end up with something like:



Essentially this flow will light the indicator light on the button only if both GPI 1 and 2 are low. This does not look very useful, but if we were to change the endpoint to be a route destination, it becomes more interesting. To do this, double click on the endpoint box and instead of a GPIO, expand the routers section and select the CurrentSourcePath property of one of the router destinations.



Then edit the translation which is after the combiner and change the translation so that True= one of the sources from the sources that will be in the right hand side drop down of the translation dialog. Remove the false translation because we do not want to do anything if the buttons are released and you end up with:



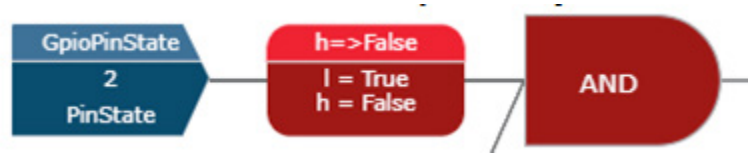
This essentially means a user has to hold two GPIO buttons down simultaneously, and then the PC4 destination will get source sa-server_02 routed to it. This is a common scenario where we don't want an accidentally bumped button to change the air chain routing so we require two buttons to be held together.

The available combiner types are shown in this chart:

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

Logical Combiners

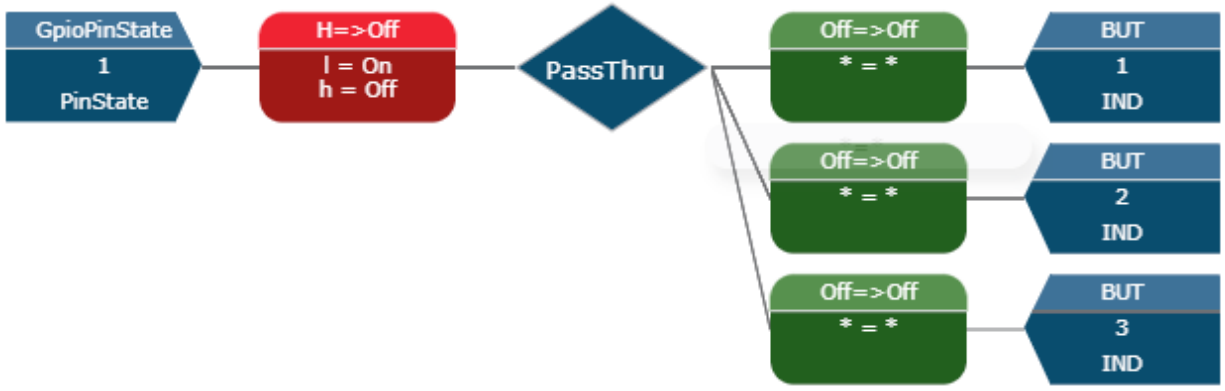
Logical combiners include AND, OR, NAND, NOR, XOR, XNOR, and NOT. All of these except NOT accept multiple inputs which must be translated to True or False on the input side. Not only accepts a single input. In each case, the translation that feeds the input of these combiners should be converted to a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

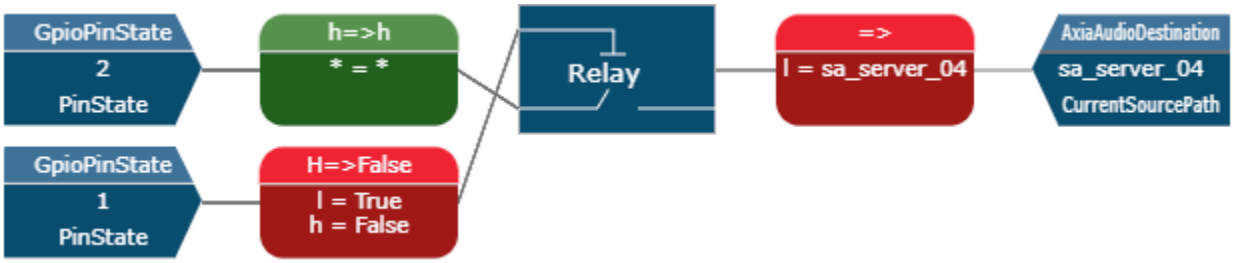
The PASSTHRU combiner only allows 1 input and passes the value from the start point translation to as many output translators as you like. This allows you to make multiple output properties change based on a single input property.



Relay

The relay combiner allows you to define one start point property that will translate to a true or false. If that evaluates to true then the value from the second start point is passed through the combination to any translations on the output of the combiner.

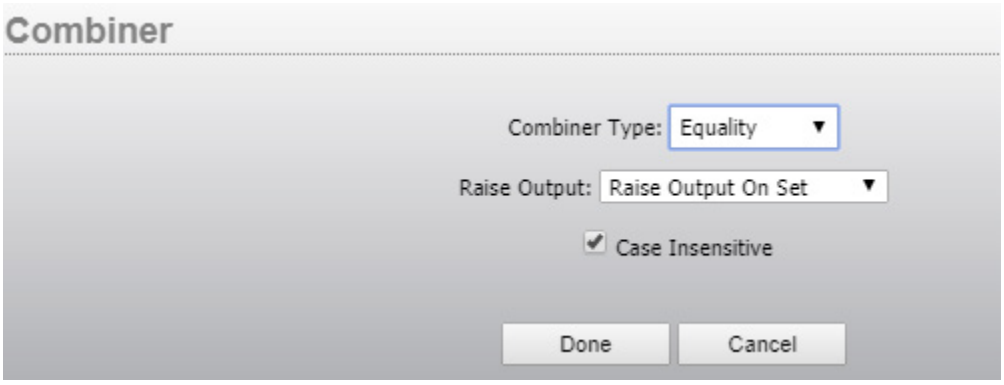
The diagram within the combination displays which start point translation is acting as the trigger and which is acting as the PASSTHRU.



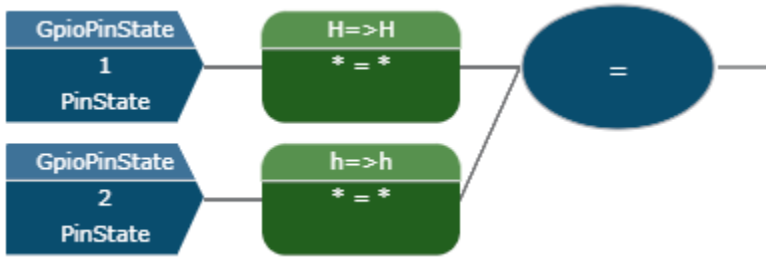
This will actually achieve the same functionality as our previous example, but in a different way. GPI 1 must be low in order for changes taking place on GPI 2 to pass thru. If it is low then changes from GPI 2 will be passed to the output translation which converts the Low to a route change.

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.



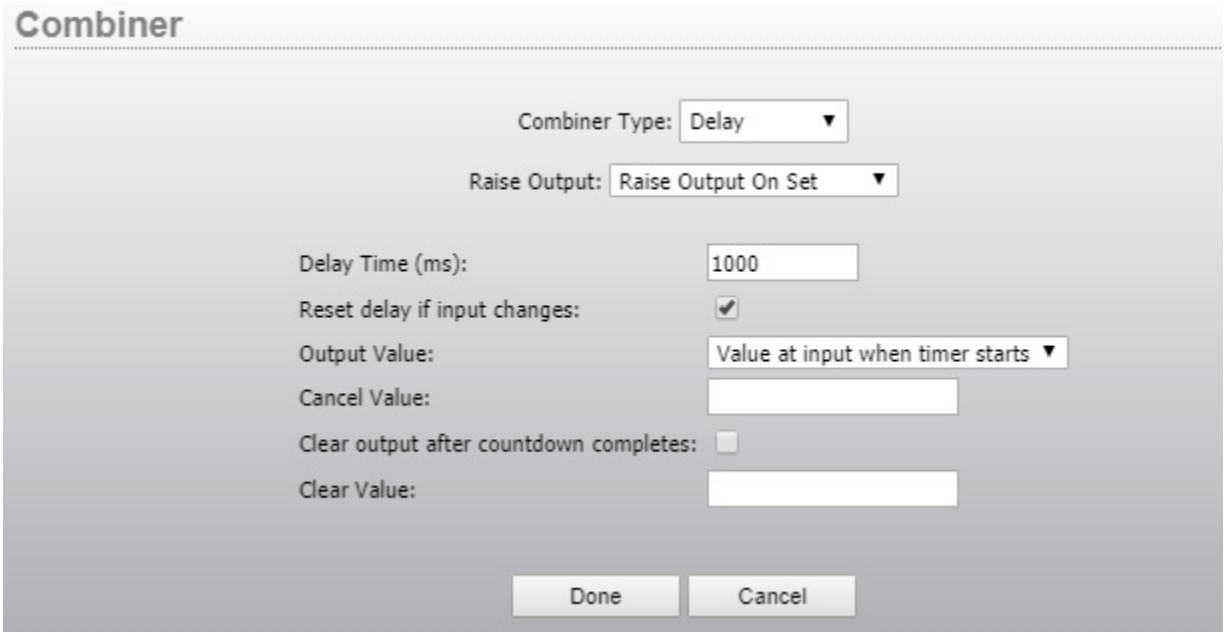
The equality combiner also has a case insensitive checkbox which defines whether comparisons are done in a case sensitive or insensitive way



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.

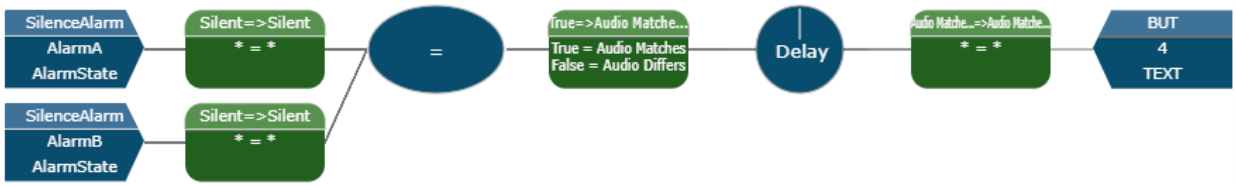


- Delay Time: Defines the number of milliseconds to delay.
- Reset delay if input changes: The delay countdown starts every time the input changes its value. If this option is selected the countdown will be reset whenever the input value changes. If you have a parameter that is fluttering and you only 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 milliseconds after a change even if additional changes happen during the countdown (delay time) then this option should be unchecked.
- Output Value: Defines whether the value that is passed through is the input value at the start of the delay countdown or the end of the countdown. 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. The options are:
  - » Value at input when timer starts
  - » Value at input when timer ends
- Cancel Value: Allows you to define 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 more on this below.
- Clear Value: used to assign 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 endpoint.

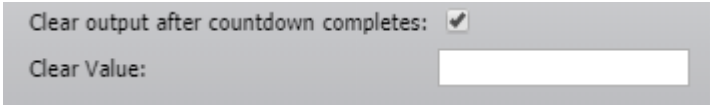
Interval timers are still useful in situations where a single delay needs to be stopped, started, reset, or manipulated by different flows.

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

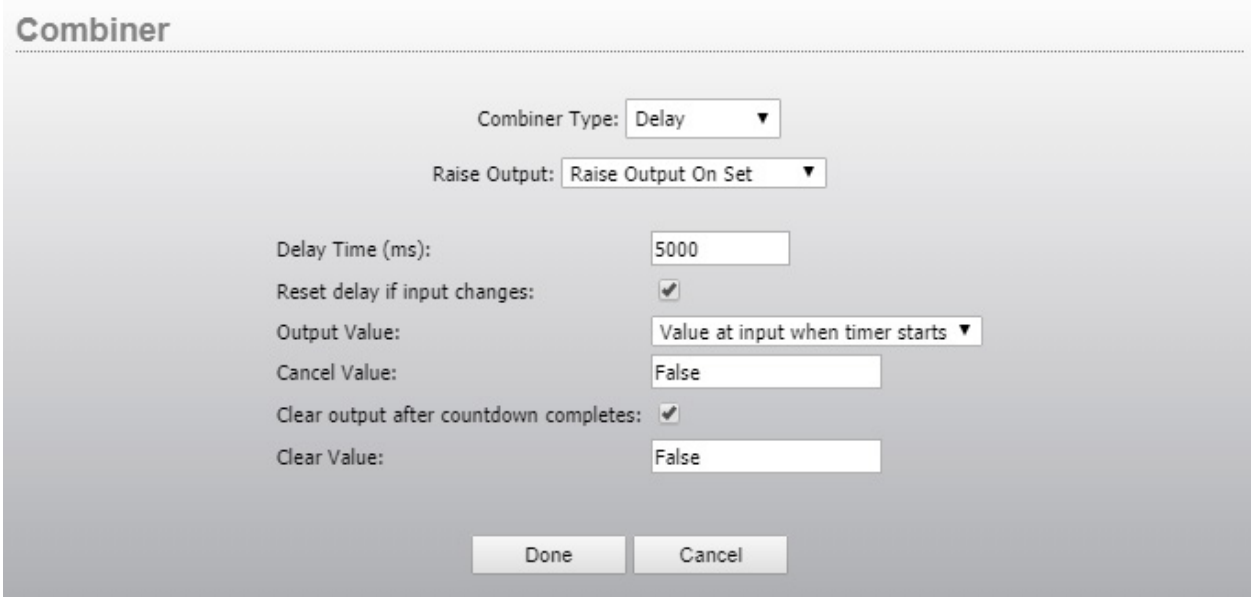


In many cases you may want the delay to pass a value through after the delay but then reset the output for the next change. For example, if we want a flow that requires a user to hold a button down for a length of time before making a change, the flow might look something like:

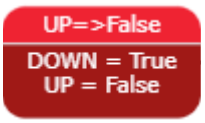
Our goal with this flow is the following:

- 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 like:



- 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.

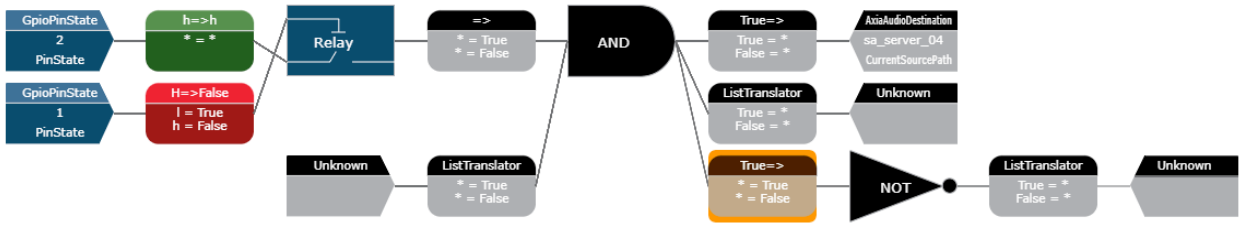
Without the clear value and clear output after countdown options, the output of the combiner would remain true and so the next press would not change the output and therefore would not trigger another route. If we cleared the Cancel value and did not use the clear options, then releasing the button would 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.

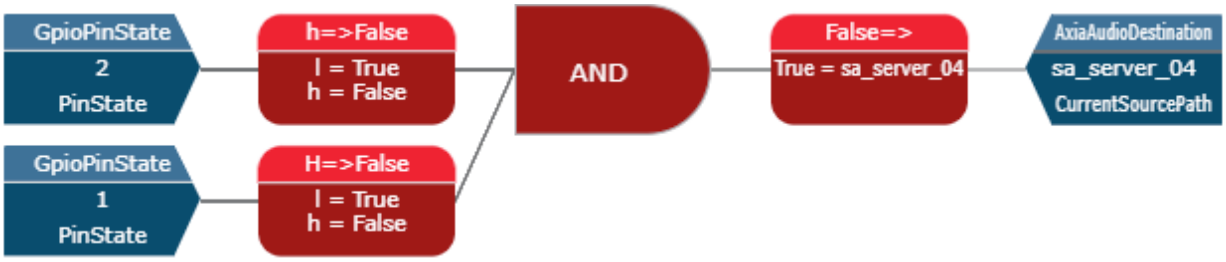


This example is not created to do anything specific, it is just shown to illustrate that you can theoretically continue to add translations and combinations as needed to get the particular logical job done. Translators and combiners must alternate. You cannot have a combiner next to a combiner or a translator 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.

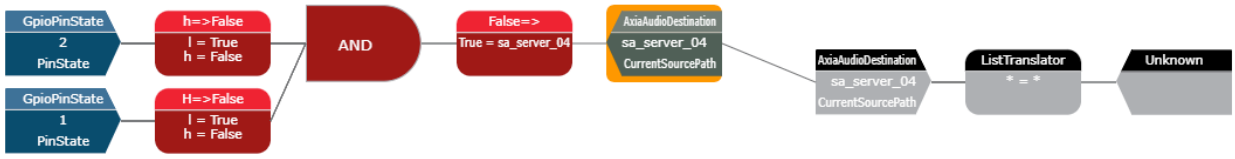
Extending Flows

If you select the endpoint of a flow, you will notice that the right translator control in the control panel is available (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 endpoint of the previous flow.

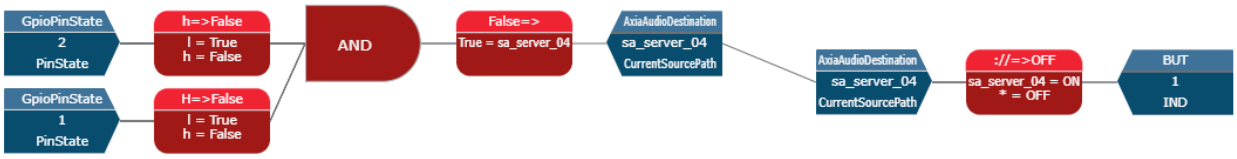
If we go back to this example flow:



Click on the endpoint and then click on the right translator control, you will get:



In this case the second flow is joined to the first 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 PC4 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 endpoint 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

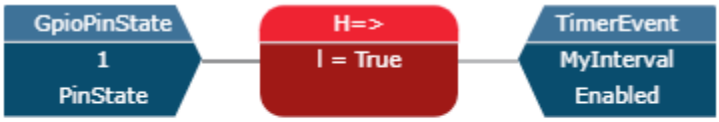
Cut, copy, and paste are standard tools, and require little explanation. You can select a flow by clicking in the area surrounding a flow. At that point the cut and copy icons will be available. Cut will remove the flow from the view and place it on the clipboard and copy will just make a copy of it on the clipboard. Once a flow exists on the clipboard it can be pasted back into the view or into 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 endpoints. So it will actually clear the outer start and endpoints 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. Then click apply to apply the title. Flows that have titles will display the titling text underneath the flow.



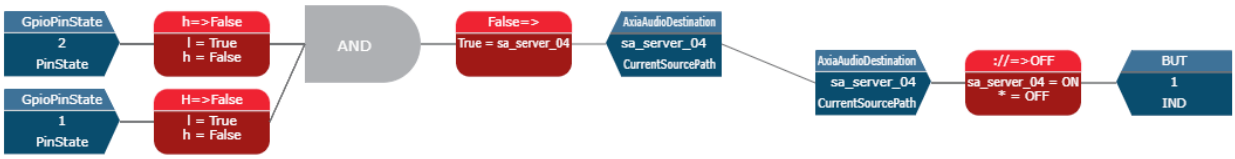
MyFlow

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.

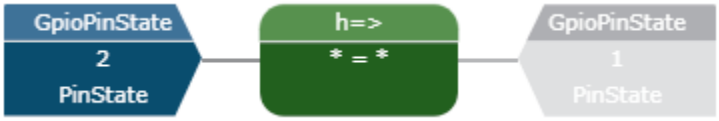
This change is the only one that takes place immediately within Pathfinder Core PRO logic flows without the need to click apply. The object will become greyed out and that particular object will no longer function, stopping the logic flow at that point. For example:



If you trigger the GPIs, their states will change within the attached translators, but the endpoint 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 endpoint (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

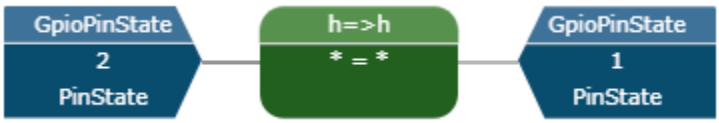
If your system is clustered and you are viewing logic flows on the secondary system you may see a lighter gray form of disabling.



Also with clustering the upper right corner will show and 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:

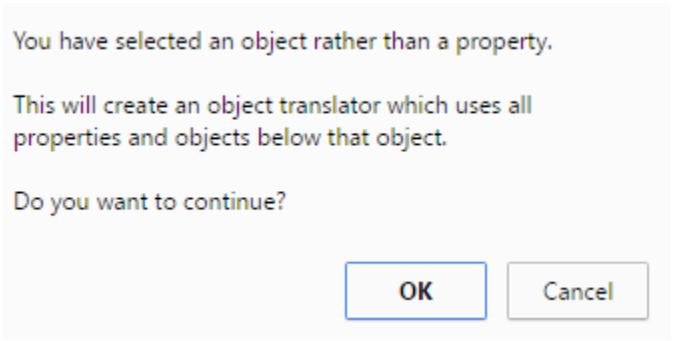
Event System: True

> **Note:** The event system is never actually off in PathfinderCore PRO. When a logic flow needs to make a change to an endpoint, 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 endpoints will not be disabled by the clustering system.

Object Translators

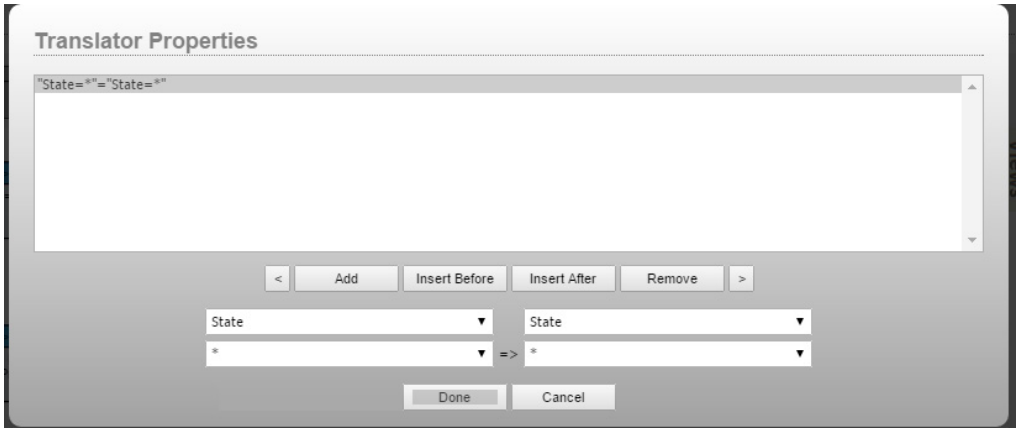
You may notice that when editing the start or end point of a flow, if you select an object in the property selection tree (arrow icon) instead of a property (green dot icon), you will get a warning message that looks like:



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 Endpoint is a different VMIX sub mixer on a different engine.

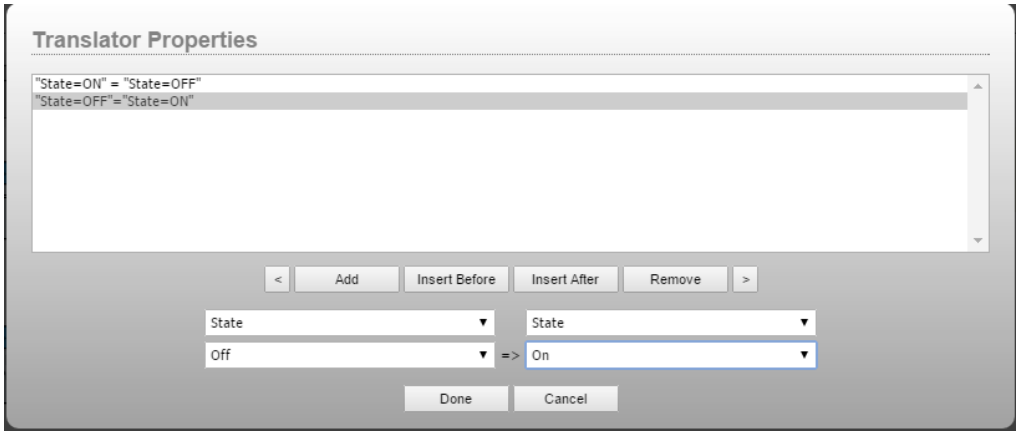
You will notice that this translator looks slightly different in that the start and endpoints only have two pieces of information rather than the usual 3. If we do not alter the translation pattern from the default *=*, this will mean that every change to every property of every sub object under the sub mixer 1 will get mirrored to the equivalent object and property under sub mixer 2. So if sub mixer 1, channel 1 gets turned on, then sub mixer 2, channel 1 will 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. In this way we can mirror settings of an object.

If you edit the translation list, you will notice that some additional fields appear when an object translation is selected. These additional drop downs allow us to specify properties of the selected object as well as values. For example, we could change the translation pattern from *=* to:



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.

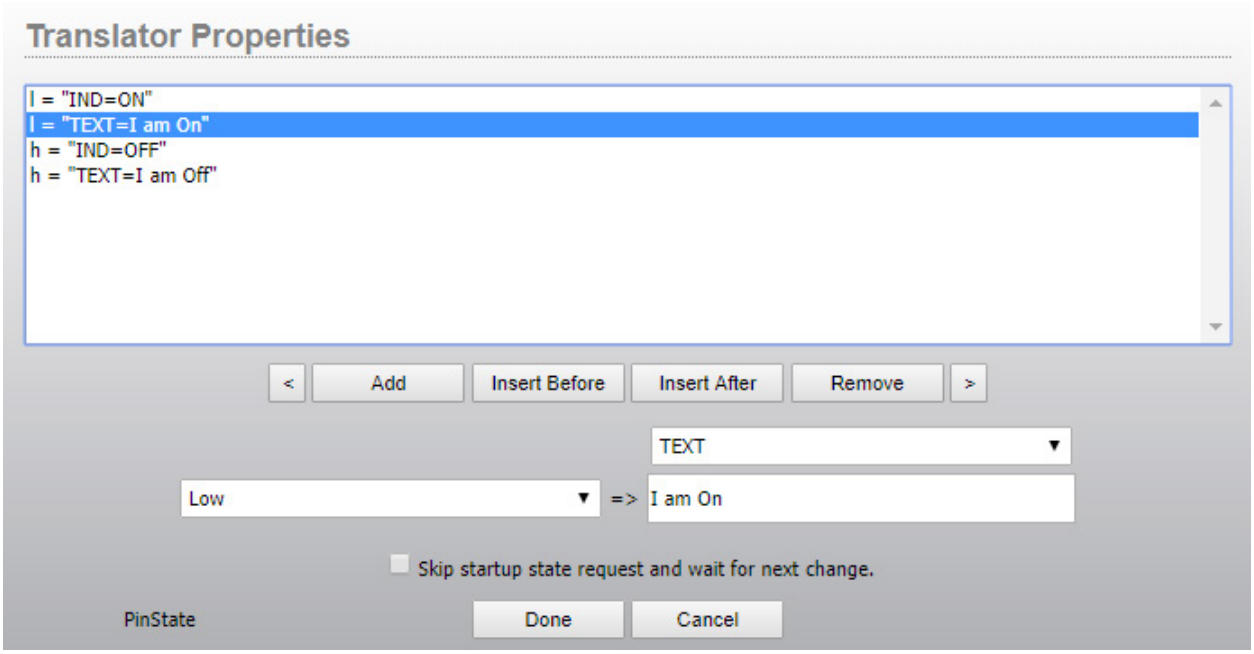
Or, we could apply values such as:



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:



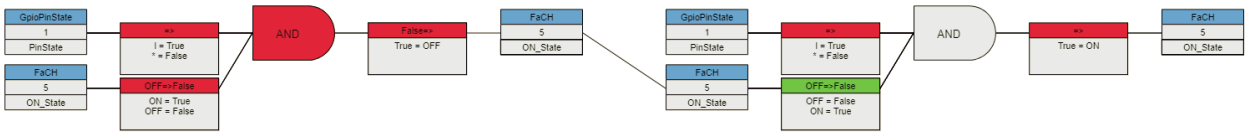
In this case we are setting two different properties on the button each time the gpio pin state changes high or low. This can simplify the flows and greatly reduce the licensing counts used when there are many button states that need to be set.

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

Latching

It is fairly common to need a button within the system to function in a latching fashion where each press of the button will switch something back and forth. Using the kind of logic shown above to try and 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 then turn it off and if the GPI is low and the button is off then turn it on. The problem with this flow is that it creates and endless loop for 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 trying to solve what on the surface appears to be a simple problem. As a result, we have crafted two much simpler ways to do latching in Pathfinder Core PRO.

Swap

The first is that many of the properties that have two states (on/off, Low/High, etc.) also have a value that you can set called swap. So for example:

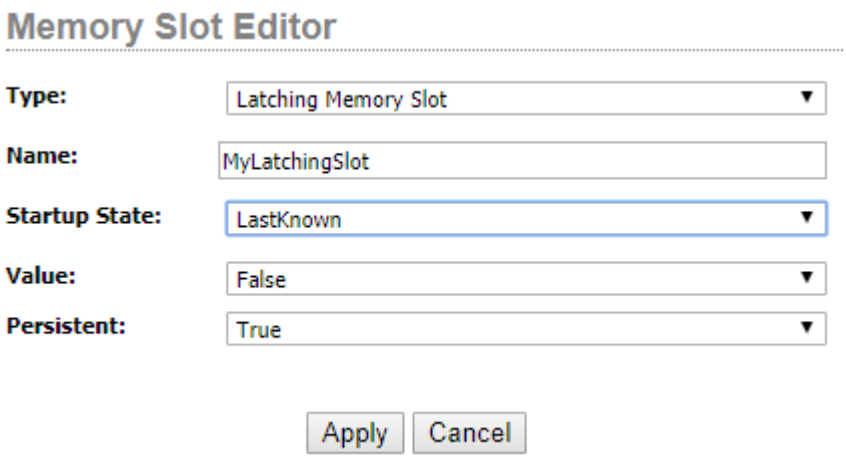
In this case each time the button is down the button state will be swapped (on to off or off to on).

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

Latching Memory Slot

However, In the case of properties that you want to use in a latching fashion that do not have the swap value, you can use the latching memory slot. To learn more, see Chapter 7 Memory Slots. However, since this can be a common logical conundrum that many novice users attempt to solve with logic flows in strange and frustrating ways, we wanted to specify the correct way to accomplish this.

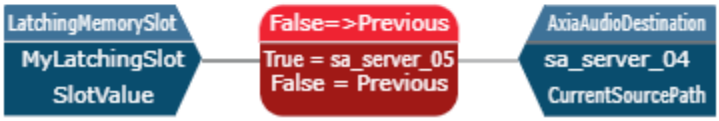
While basic memory slots can be created using the Logic Flows property tree as a shortcut, there are additional types of memory slots that can only be created from the memory slots page. These are outlined in more detail in Chapter 7. However, in order to create latching functionality, go to the memory slots link in the Navigation Bar and add a new memory slot using the plus icon. In the dialog that appears, switch the memory slot type to "Latching Memory Slot". Provide a name for the slot and click Apply to create the slot.



The Latching memory slot is a special type of slot that can only have a value of true or false. It also has a write only property called trigger. For example, in logic flows we could now add a flow such that whenever a button on the element is pressed the translator sets the trigger property to true.

The trigger property is write only. It cannot be read. And 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 the button 10 is pressed.

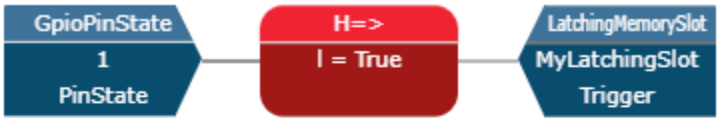
Next, we create a second logic flow that defines what we want to happen depending on whether the Latching memory slot is true or false. For example:



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

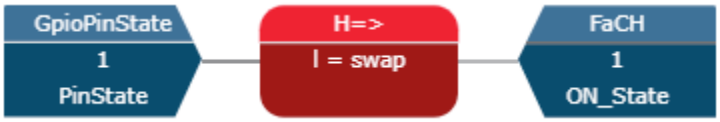
Swap over Latching Memory Slot

It is important to note again that swap is preferable where it is available. For example:



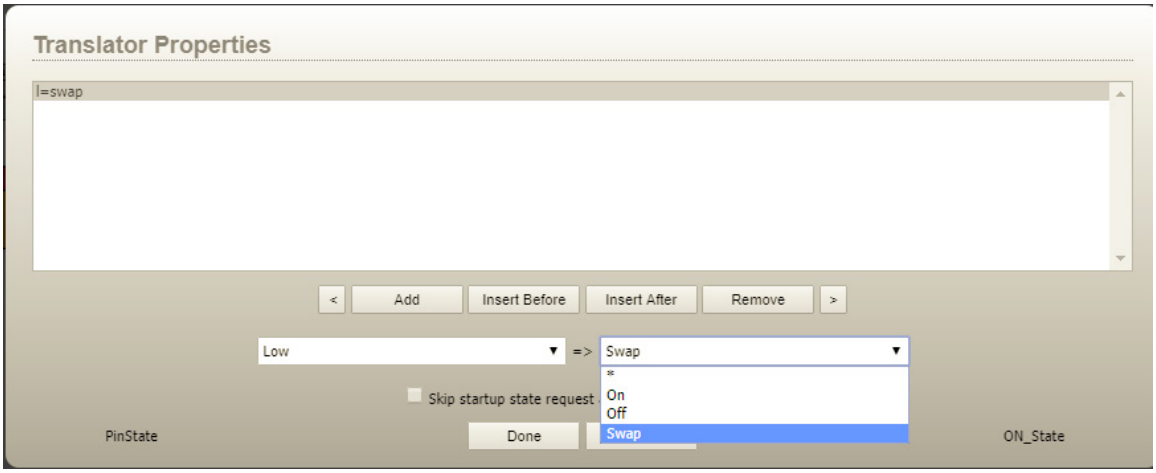
In this case the Trigger property causes the latching memory slot to swap between true and false each time the GPI goes low and then that true or false is translated to the on/off state of the fader. So engaging the gpio repeatedly will cause the fader to turn on and off. The problem with this flow is 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. It feels complicated to do what on the surface feels like it should be simple.

The swap value makes this much simpler:



That is all you need to solve the 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.

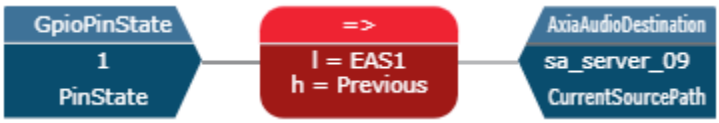
It is important to note that the swap value is an action and so will only appear in the translation options for an endpoint. It has no meaning for a start point because it is not a real actual state and so it will not appear in the options on a start point. The user interface will present the option where it is available. For example:



Note: Axia LCD button 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.

EAS Example

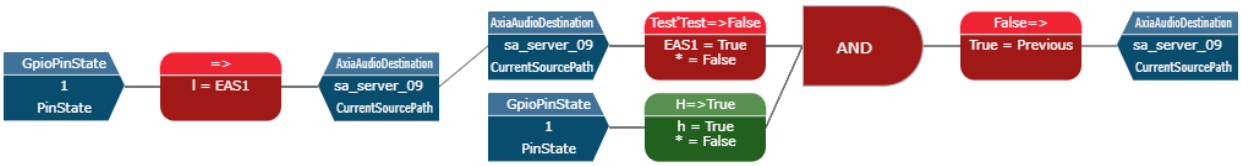
Another highly relevant example is EAS. This is a pretty common EAS flow that we see customers make.



First 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 section on this option.

Without 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.



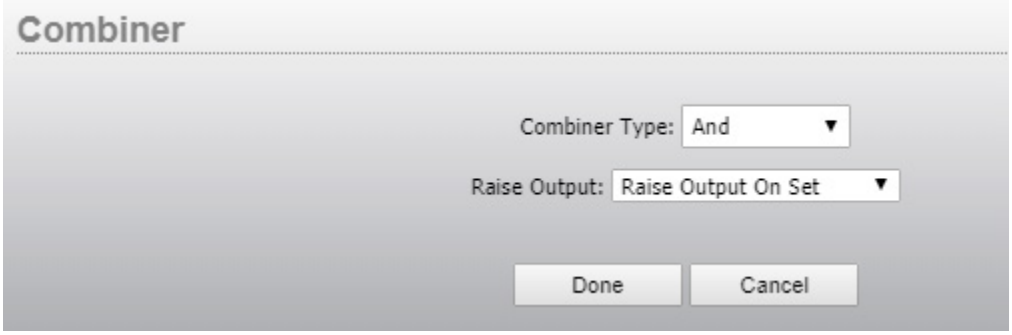
In this example, a route to previous will only happen if the GPI is high and we are currently on the EAS source. This will function in a much more reliable manner.

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.

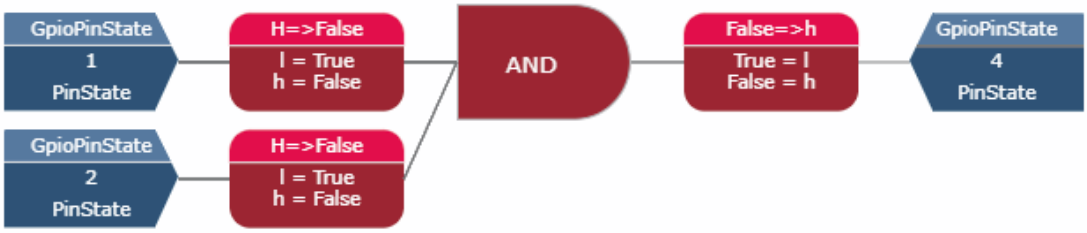
Advanced Options

Raise Output Property

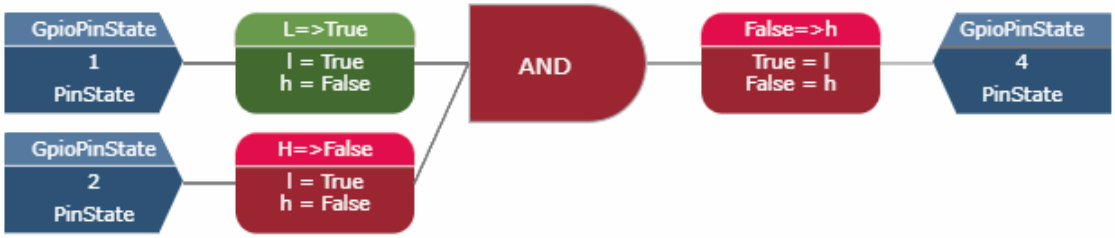
Over the course of the examples above, you may have noticed an additional property in combiners that we have not discussed yet called Raise Output.



The values for this property are Raise Output on Set and Raise Output on Change. The effects of this property are a bit subtle, but can be important in certain situations to understand.



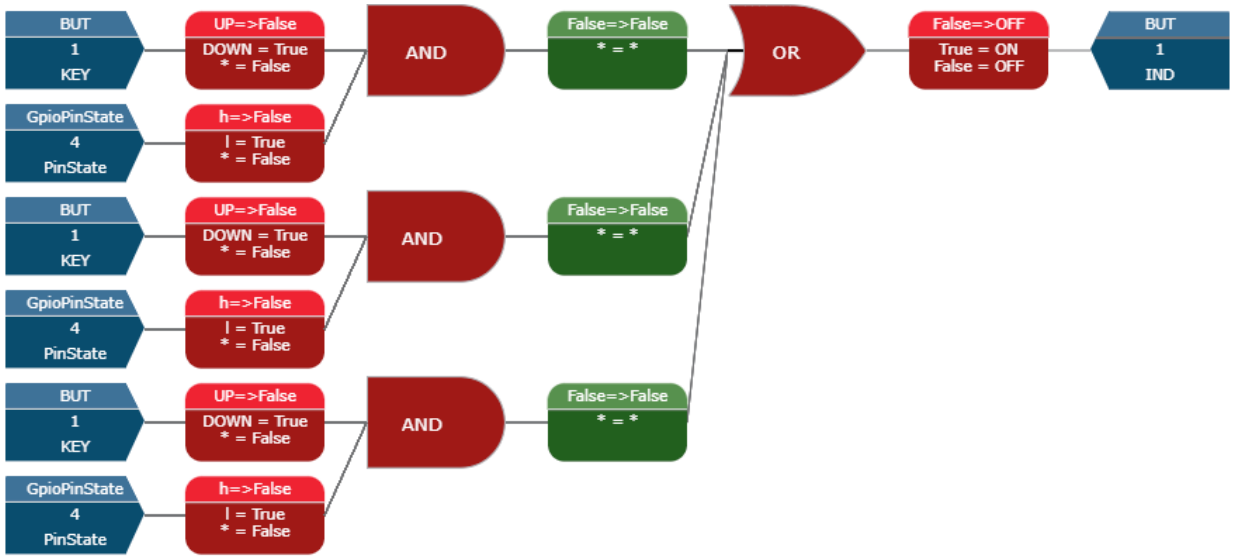
In this flow if both pin 1 and pin 2 are low, then pin 4 will be set low. And if either pin 1 or pin 2 is high, pin 4 will be set high. If the combiner is set to Raise Output on Set, then any time either pin 1 or pin 2 is changed a message will be sent to set pin 4 to the corresponding value. So, if 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 would 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 a bit subtle, but essentially with “Raise Output on Set”, a change message will be sent to the endpoint 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 that might be a better 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 endpoint 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 into the backing storage. For backwards compatibility, Raise on Set will store the same way it historically has making that backwards compatible with older software versions. 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.

Note: 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. Example:

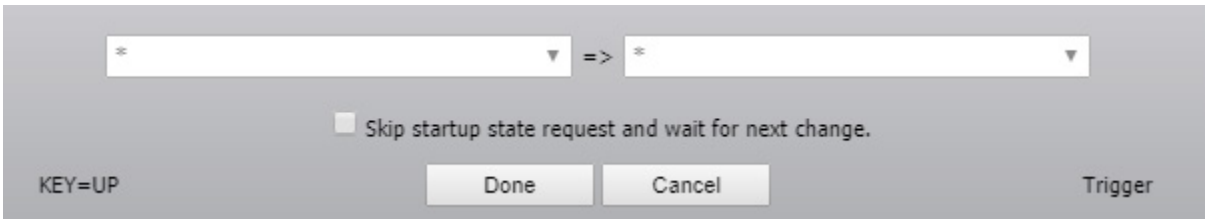
set LogicFlows#0.LogicFlowFolder#yyy ChangeAllCombinerRaiseOutputSettings= RaiseOutputOnSet

set LogicFlows#0.LogicFlowFolder#yyy ChangeAllCombinerRaiseOutputSettings= RaiseOutputOnChange

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

Translator Skip Startup State

One additional option that has not been discussed in the translator dialog is the “Skip startup state and wait for next change” option. You will find this in the translator configuration dialog:

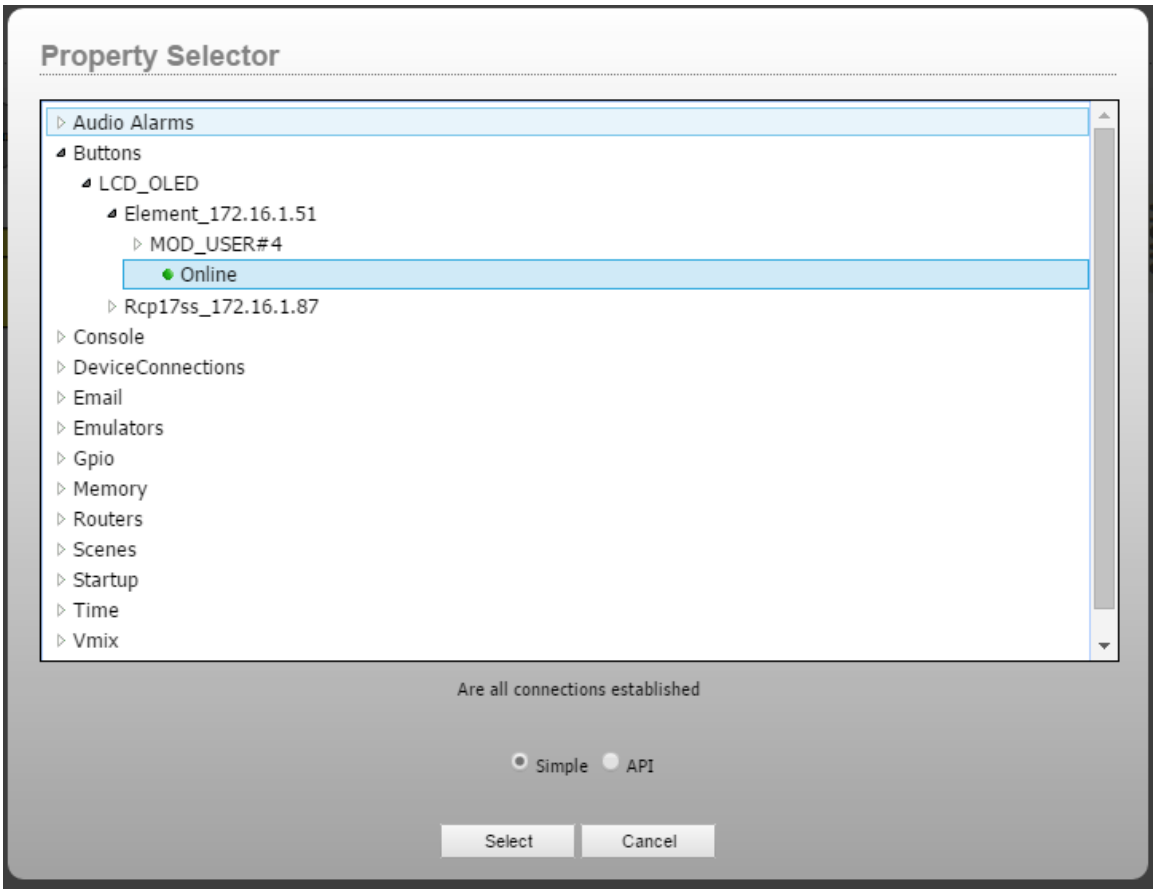


By default when an event is created and saved to the system or during system startup, the system will request the initial state of the start points. The response will then trigger the logic flow to set the correct value for whatever the state is. In most cases this is desirable. But in some you may not want to trigger it. 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. It is always important when designing a system and before it goes live on air to test that your logic does what you want in the case of a system restart and/or device restart.

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

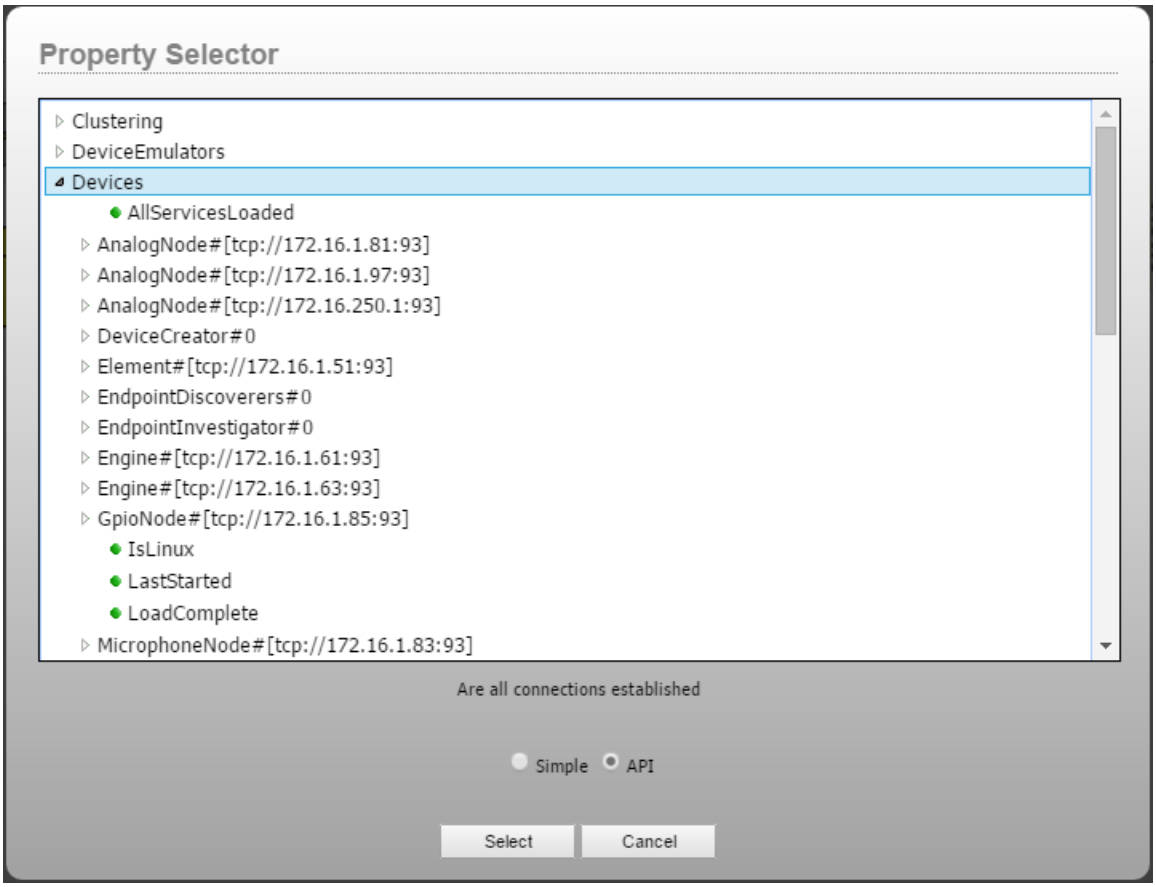
Property List Types: Simple versus API

When editing a start point or endpoint, the property selection dialog has two modes; Simple and 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. Selecting the API tree will present a much wider array of objects and properties.

Pathfinder Core PRO has an advanced protocol called SapV2. In reality it 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 not just a control protocol; it is the protocol which is used for all internal messaging between services within the system. If you want to learn more about this API, see the SapV2 appendix. In order 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.



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. This list is not inclusive. 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 if 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.
  - » ActivateScene: 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 setup 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 changed 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
  - » AUDIO_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

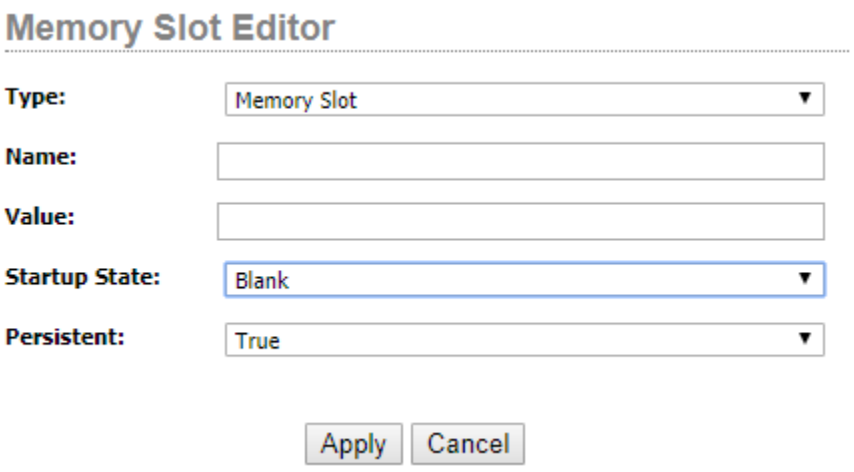
While this chapter has reviewed the details of logic flows, many of the other chapters of this manual will touch more on their uses as we detail how the aspects of the system functions and therefore what capabilities may be used by logic flows. There will be more examples of logic flows as we progress through the rest of this manual.

Chapter 7

Memory Slots

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.

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

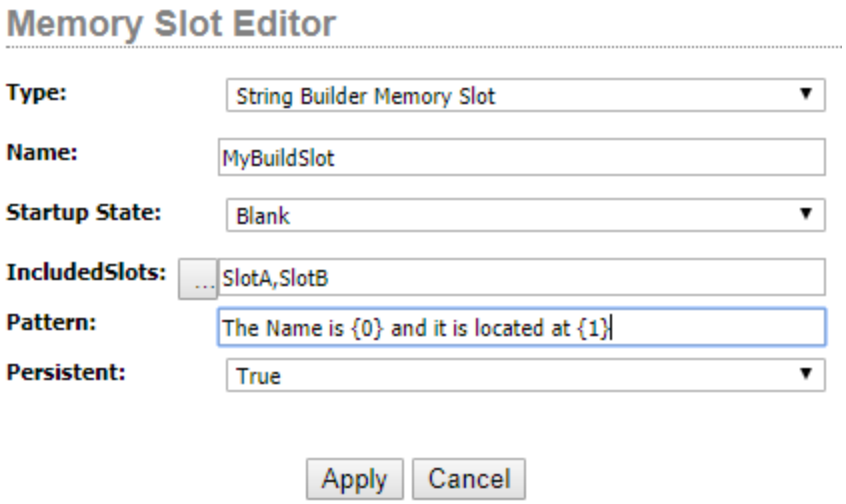


Pathfinder Core PRO has five different types of memory slots. Each one has its own options and properties:

- Memory Slot: This is the traditional memory slot into which any kind of data may be stored.
- Latching Memory: This memory slot can only have a value of True or False. Its write-only property is called Trigger. 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 false to true. See the latching example in the Logic Flows section of this manual for more details.
- Numeric Memory Slot: This memory slot only accepts numbers. Its write-only properties are increment and decrement. Use this in logic flows where you need to increment or decrement the value in the memory slot.

- Sap Property Memory Slot: This slot allows you to assign any property in the system to this memory slot. When this property value changes, then the slot is updated with that value. The ellipsis button can be used to select the property in the system to be used with this slot. It will present a property selection dialog similar to that of the dialog in Logic Flow endpoint editing.

- String Builder Memory Slot: This slot allows you to build a string based on values in other memory slots. The "IncludedSlots" field is used to enter a comma-delineated list of memory slot names to use in the builder. The pattern field is a text pattern with bracketed numbers used where slot values from the comma-delineated list should be inserted. For example:



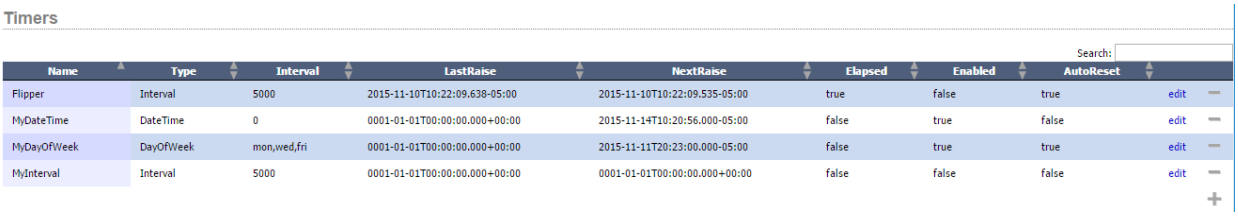
In this case {0} will be replaced with the value of slot A and {1} will be replaced with the value of slot B. If Slot A's value is Fred and Slot B's value is Telos Alliance, the value of this new memory slot called MyBuiltSlot would be: The Name is Fred and it is located at Telos Alliance. The value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

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 to 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 PathfinderCore PRO. As a result, use that option where it is needed and not 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.

Chapter 8

Timers

Timers allow flows to be executed at specific times. This list shows the timer that have been created and may be used in Logic Flows.



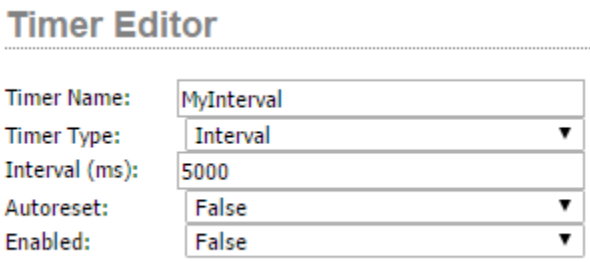
The page shows up to date information as the timer states changes. The following columns are presented:

- Name: The name of the timer.
- Type: The type of timer.
- Interval: In the case of interval timers, this column will display the time in milliseconds. In the case of Day of week timers, the column will display the days per week that are selected. This column is not used for Date/Time timers.
- LastRaise: This column displays the date and time when the timer last elapsed. If the timer has never elapsed, it will display a default minimum date time value.
- NextRaise: This column displays the date and time when the timer is next scheduled to elapse. If the timer has never been enabled, this column may also show a default minimum date time value.
- Elapsed: The elapsed column will turn true if a timer has elapsed. It is important to note that if a timer is set to Autoreset, this value will immediately turn false again.
- Enabled: This column displays whether the timer is currently enabled. A timer which is not enabled will never elapse.
- Autoreset: Interval timers may be configured to AutoReset. When an auto resetting timer elapses, it will automatically update its next raise time and starting counting down again.
- The edit link allows you to edit and change the parameters of a timer.
- The minus icon allows you to delete a timer from the system.
- The add icon is used to create new timers.

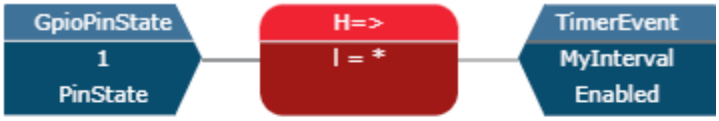
There are three types of timers that can be created in Pathfinder Core PRO:

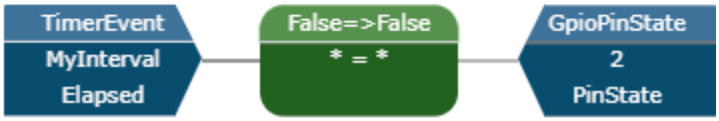
- Interval: An interval timer raises an elapsed property after a specified number of milliseconds. Interval timers can also be programmed to autoreset. An autoresetting interval timer will set elapsed to true after the interval and then immediately set it back to false and restart the countdown.
- Date/Time: This allows you to specify a specific day and time. The timer's elapsed property will be set to true at that day and time.
- Day of Week: This allows you to specify a time and the days of the week on which the timer should set its elapsed property to true.

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. Leave the autoreset and enabled properties as false. Give the timer a name and apply it.



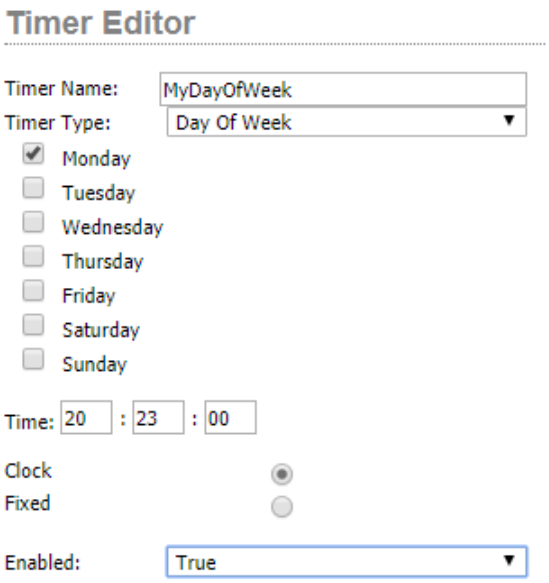
Now we can use this timer in the logic flows.



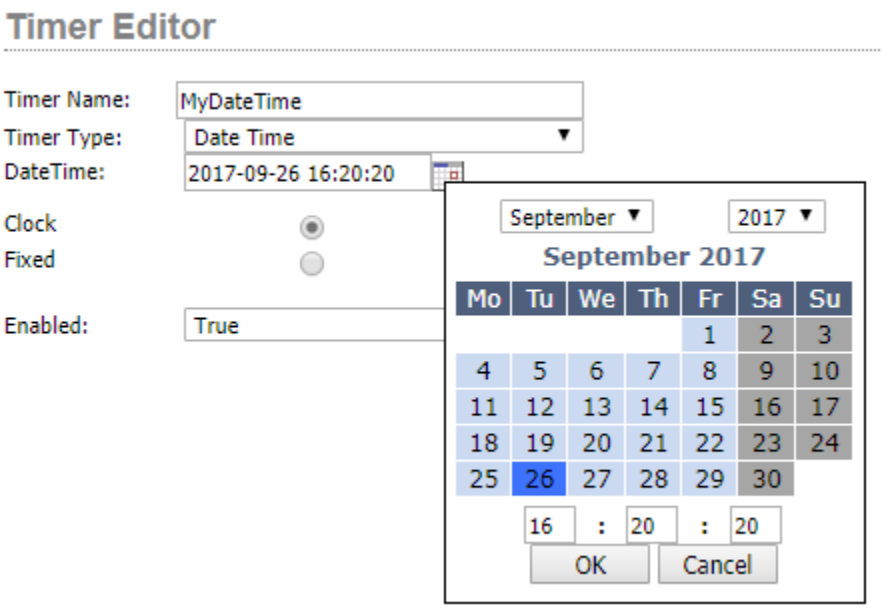


In this case, when GPI 1 goes low, it enables the interval timer. Then the elapsed property 5 seconds later causes GPI 2 pin to be set low.

If we want something to happen at a specific time every Monday, we could create a Day of Week Timer.



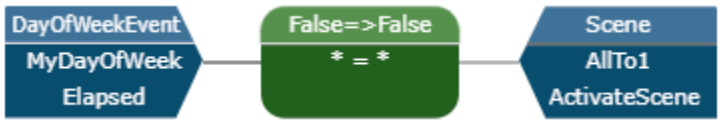
Or, if we wanted something to happen on a specific day and time, we could create a Date/Time timer.



Clicking the calendar icon will present a user-friendly calendar for configuring the date and time.

In both the Day Of Week and Date/Time timer, it is important to understand the difference between clock time and fixed time. Within PathfinderCore 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. So if you want an event to happen at 7PM according to what the clock shows both before and after a daylight savings shift, then you should pick the clock option. However, if you are picking up a satellite feed and the region from which the feed originates does not shift their clock with 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.

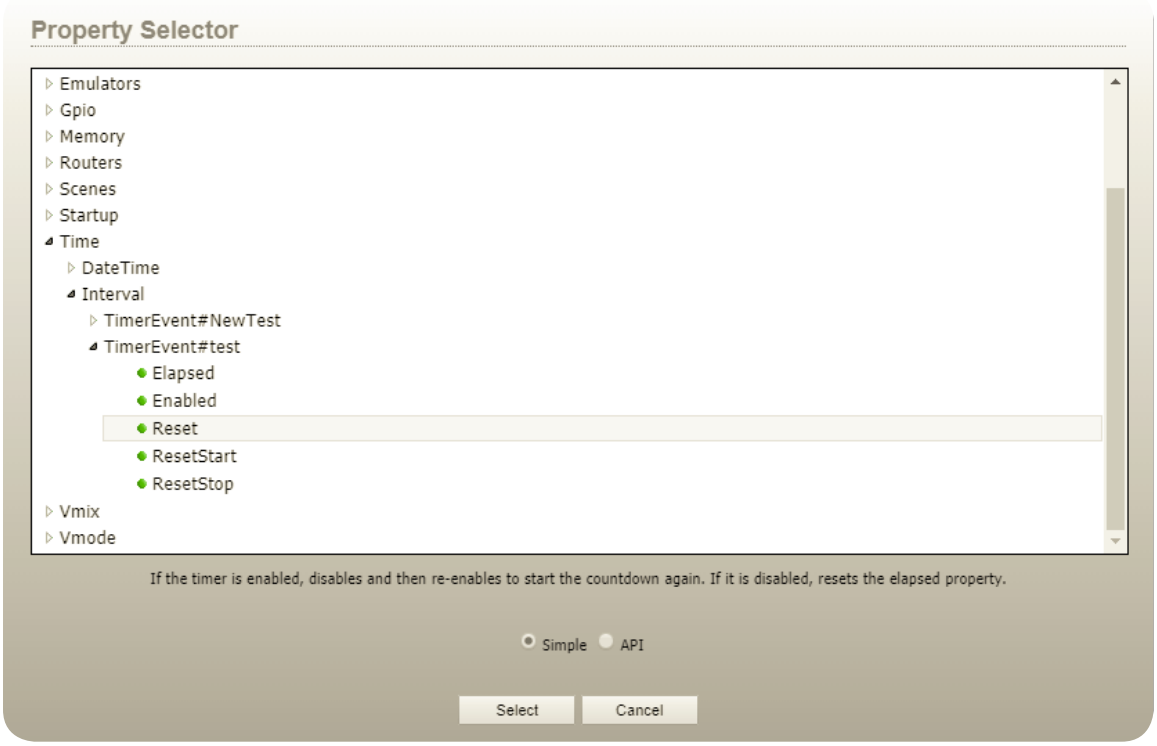
It is also important to understand that creating a timer in Pathfinder Core PRO will not actually cause any changes to take place in the system. To do that, you would need to create a logic flow as shown above with a start point that uses the timer's elapsed property to then change some other property 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 DayOfWeek 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.

Interval Timers Additional properties

Interval timers also have three additional write only properties available. These are Reset, ResetStart, and ResetStop. 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.



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.

The description of each of these properties is as follows:

- 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.
- ResetStart: Stops the timer (if it is running) and then starts the timer to reset the countdown.
- ResetStop: Disables the timer and changes the elapsed state to false.

Chapter 9

Scenes

In Pathfinder Core PRO, scenes can only be created using the PathfinderPC_Core client. See Chapter 18 PathfinderPC_Core Client to review how to create scenes.

There are a few differences related to scenes under the hood in Pathfinder Core PRO vs Pathfinder Pro. One is that there is a read only property which can be used in logic flows for each scene called IsActive. This means that in addition to activating a scene with a logic flow or a client application or the web UI, you can also create logic based on whether all of the points in the scene are currently in the requested state.

Another important difference is that under the hood a scene is nothing more than a set of API commands (See appendix on SapV2 API). So a scene does not have to be just route changes or even route changes at all. You could create a scene that sets a bunch of memory slots to specific values. Or you could create a scene that includes route changes, fader changes, memory slot changes and button state changes. Unfortunately, there is no user interface yet to make these kinds of scenes so in the current version unless you use the API, only route change scenes can be created. In the future we will have a scene creation interface that allows adding different kinds of changes.

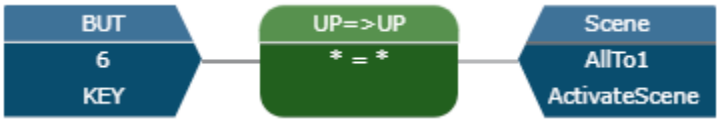
Once scenes are created using PathfinderPC_Core Client, they will be listed and may be activated using the Pathfinder Core PRO web pages in addition to the PathfinderPC_Core Client. Click on the Scenes link in the navigation bar to view the available list of scenes:

The columns on this page are dynamic and will show the state of the scenes as they change. The columns included are:

- SceneName: The name of the scene. Scene names may be preceded by a number and an underscore in some cases. These represent scenes created by PathfinderPC_Core Client that are related to a specific router number.
- IsActive: This shows dynamic information, and will display True or False depending on whether all of the entries in the scene are currently in their requested state. The IsActive property is also available to logic flows so that a flow can perform certain actions depending on whether a scene is active or not in addition to actually triggering a scene.
- The Activate button on each line will cause all of the changes within a scene to be executed in Pathfinder Core PRO.
- The edit link is for future use.
- The minus icon allows you to delete scenes from the system.
- The plus icon is also for future use. Use PathfinderPC_Core Client to create scenes.

It is important to understand that when a scene is activated, the requests are sent by Pathfinder Core PRO to the required equipment, but in some cases the scene change might not be successful. For example, if an Xnode that is involved in the scene is currently powered off, the scene change will not be completed. In this case the IsActive state will not change to True.

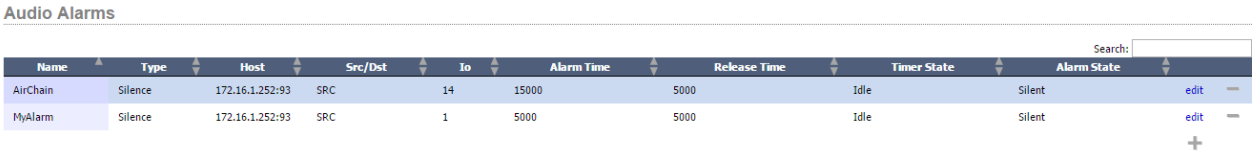
Scene changes may be activated using Logic Flows using the ActivateScene Property. This property is a write only property and will therefore only be available when editing end points. As an example, we could create a logic flow that executed a scene change every time a specific LCD button was pressed.



Chapter 10

Audio Alarms

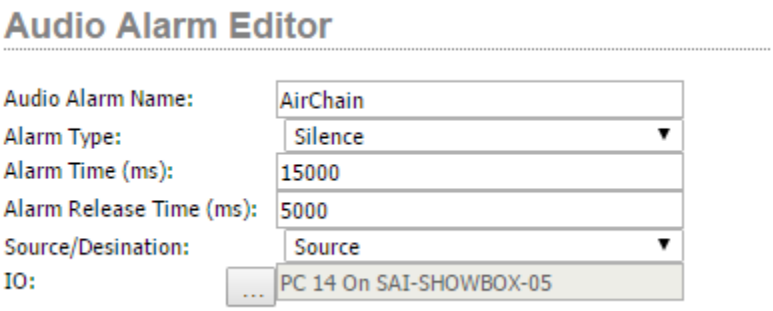
A very common use of logic flows in Pathfinder Core PRO is to perform critical actions when an audio source or destination falls silent. For example, you could have a logic flow send you an email if the feed to the transmitter becomes silent. In order to accomplish this, you should use the Audio Alarms 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 on the Audio Alarms link in the Navigation Bar to create, edit, and view audio alarms in the system.



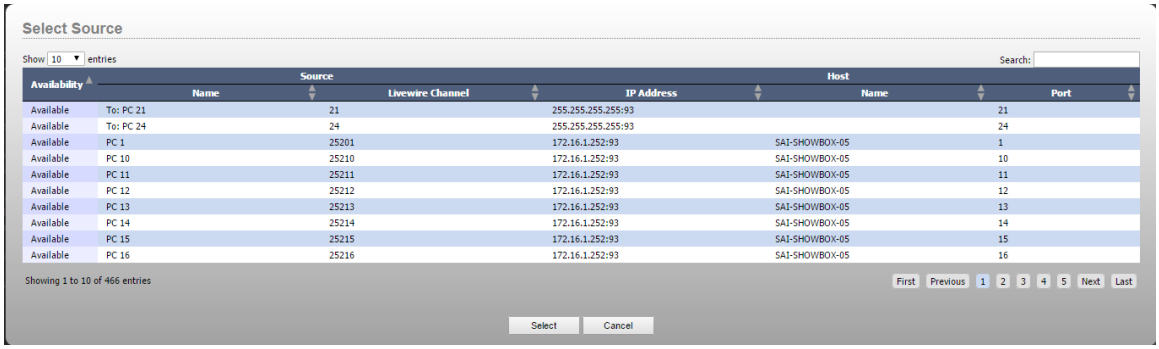
As with other Pathfinder Core PRO pages, this one will present dynamic data as the alarm states change. Nine columns are displayed for each audio alarm:

- Name: The name of the alarm.
- Type: The type of the alarm. Types can either be Silence or Clipping. If you wish to know when audio returns, use a silence alarm and then select the correct value for the Alarm state in logic flows. 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 that 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 goes silent, this column will show “CountingToAlarm”. Idle means that the source or destination is currently in the state represented by the AlarmState column.
- Alarm State: This column represents the current state of the alarm. Possible values include Silent, Audio-Present, and Clipping. This value will only change after the source or destination maintains the requested state for at least the amount of time in the Alarm Time or Release Time.

Let's create an example. While on the Audio Alarms page, click the plus sign to add an audio alarm.

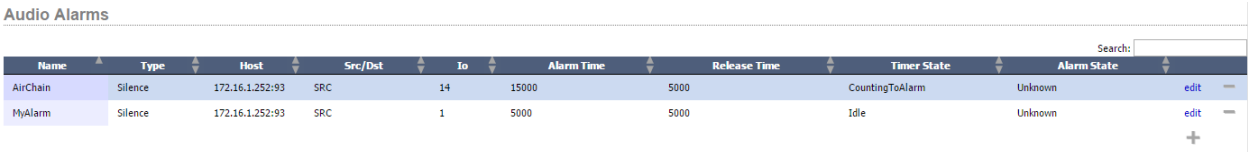


Provide a name for the alarm, in this case, AirChain. Select the type, in this case, silence. Alarm time is the number of milliseconds the audio must be in the requested state before the alarm becomes active. Let's make this value 15000 ms. Alarm release time is the amount of time it needs to be in the opposing state after an alarm has tripped before the alarm clears. We'll set that to 5000 ms. Select whether the alarm will be on a source or destination. Finally, use the ellipsis button to choose from the list of available sources or destinations.

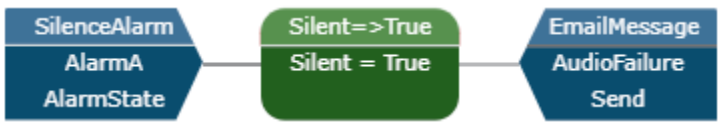


Apply the changes.

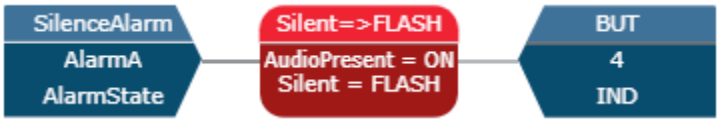
It is important to remember that the list view of the alarms (like most lists in Pathfinder Core PRO) show the alarms in real time.



For example, the timer and alarm states will dynamically update on this screen as the alarm states change. This can be an excellent troubleshooting tool. Once the alarm has been created, it is available for use in Logic Flows. In the logic flow listed below, we have created a flow that sends an email whenever the MyAlarm audio alarm has its alarm state switch to silent.



If we wanted to do a different action when the alarm releases (audio returns), just use the Audio Present 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 examples of possible audio alarm flows are covered in the section on email messages, since that is a common action to be taken when things get quiet.

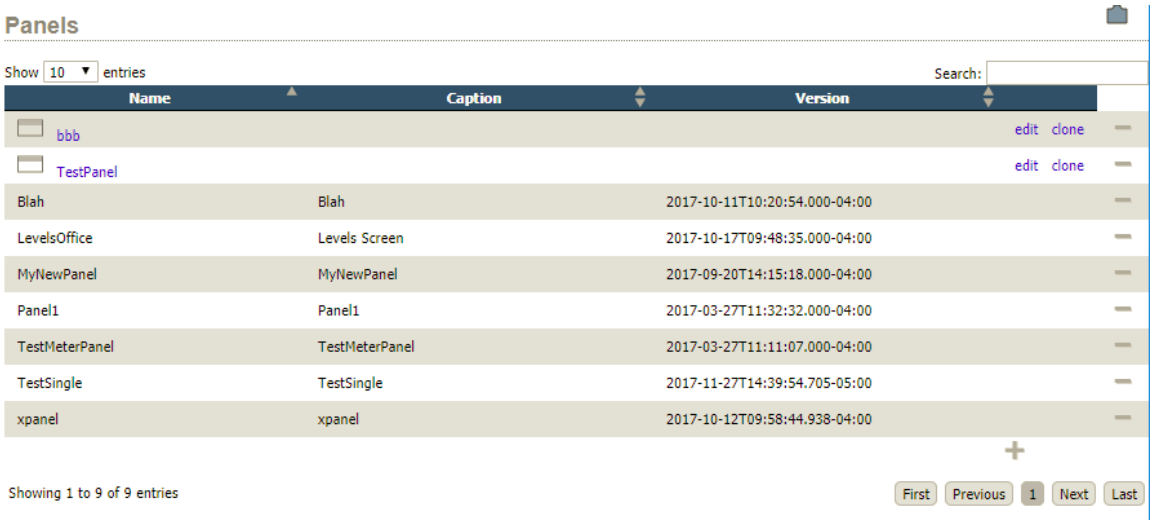
Chapter 11

User Panels

Html5 User Panels

Pathfinder Core PRO allows you to create your own custom user interface that can be displayed and used on any web browser capable device. This allows you to define precisely how your users are allowed to interact with your Axia system in a way that makes sense to your organization's workflow. Currently Chrome is the recommended browser of choice for this feature.

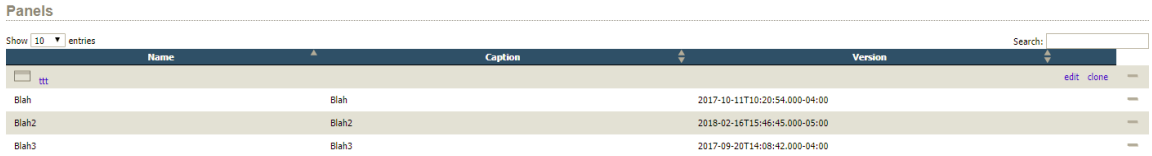
Html user panels may be used and edited from the User Panels menu item.



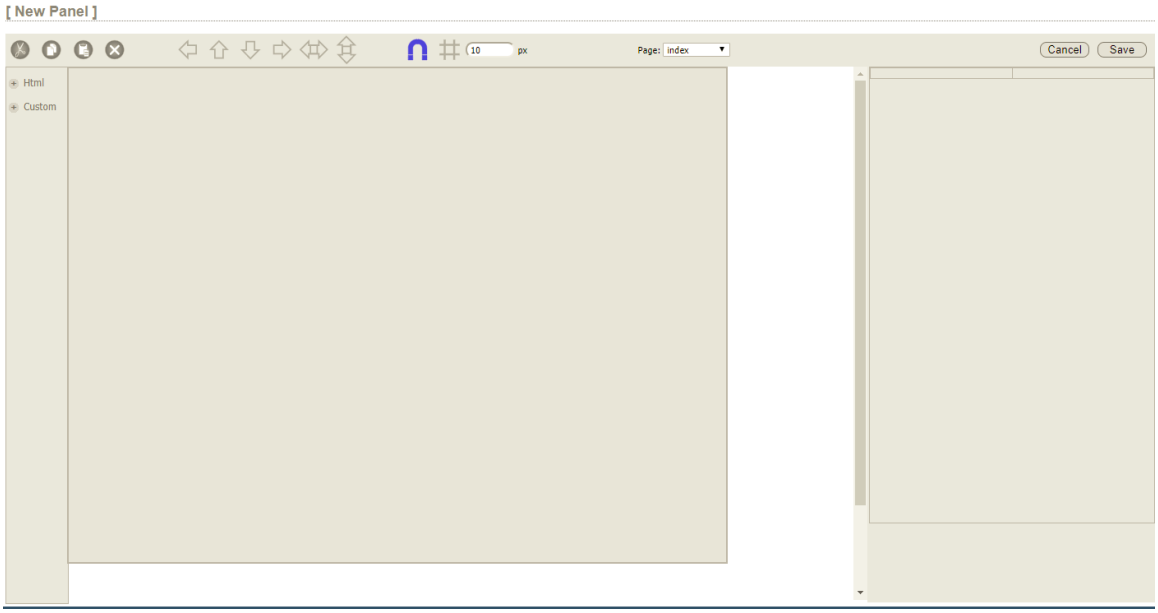
This list will show both the new Html5 panels and the original legacy panels created with the legacy panel designer. Html5 panels will have their name displayed as a link. Clicking that link will display the panel within the context of the PathfinderCore PRO web pages. Clicking the icon will open the panel in its own window without most of the browser menu systems. Finally, the html5 panels will have edit and clone links. The edit link will open the panel in the html panel designer and the clone link will make a duplicate of the panel as it exists under a new panel name. The icon will delete an existing panel. The icon will open the html panel designer for creating a new panel.

Creating a new panel

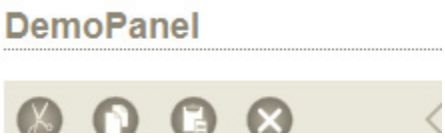
To create and use the new panels, select the User Panels menu item.



Click the + icon to create a new panel.

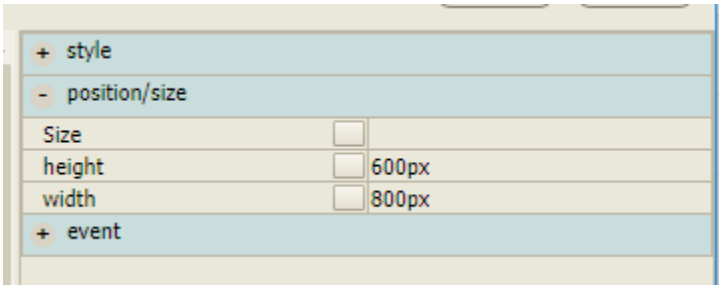


This will open a new panel in the panel designer. We recommend clicking the Save button next in order to give your panel a name and instantiate it before you begin adding components. After saving and giving the panel a name, the name of the panel should appear in the upper corner:



Next set the size of the panel. Click in the main panel so that the panel gets highlighted.

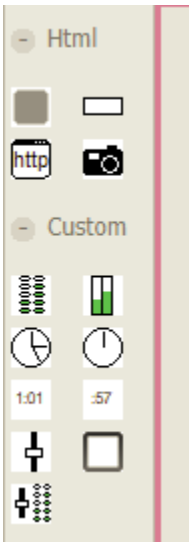
On the right side, the property window will fill with sections of properties that may be manipulated. Expand the position/size section.



Change the size of the panel by clicking in the size field and selecting from a pre-determined size or by altering the height and width properties. The size of the panel area should change accordingly.

Adding Components

To add components to your panel, expand the Html and/or custom sections in the left toolbar.



Hovering over any tool will show a hover balloon with the name of the component. To explain the tools for the components and how to use them this document will start with a simple html button. Later we will discuss the other components that may be used. Click and drag the top left html button (hover balloon says Button) from the tool bar into the panel. When you let go of the button at the end of the drag operation a new button should appear on the panel and it should be highlighted with the red box indicating it is the selected component:

Dragging the edges of the button will resize it whereas dragging from the center of the button will move it. You can also use the arrow keys on your keyboard to nudge the component by small amounts in any direction. It will move 1 pixel at a time unless the grid is enabled in which case it will move by the grid amount. We will discuss the grid more shortly. The selection handle displays just inside the actual edges of the object by design so when aligning objects, you can still see the actual edge of the object.

Note: Some components may by default resize both height and width when one or the other is dragged. These are generally more complex components such as the console fader component. This is because non-scaled resizing causes the component to look skewed and stretched so both sizes are changed to maintain aspect. Holding the shift key while resizing overrides this behavior and allows you to skew the component if desired.

Adjusting Properties

It is also important to note that the property list on the right will have updated to present the properties that may be changed for the currently selected component (our new button). Clicking in the field and changing one of the properties will make the corresponding changes to the button. For example, click in the caption field and add a caption:

Expand the style section and try adjusting the border-radius or the font-size.

As you can see there is a high degree of power to achieve exactly the desired design using the properties in the property grid.

Clicking cancel will return the panel to its previous save state and clicking save will save the changes.

> Note: In this version there is no undo or redo so frequent saves as you are working are recommended.

Tool Bar

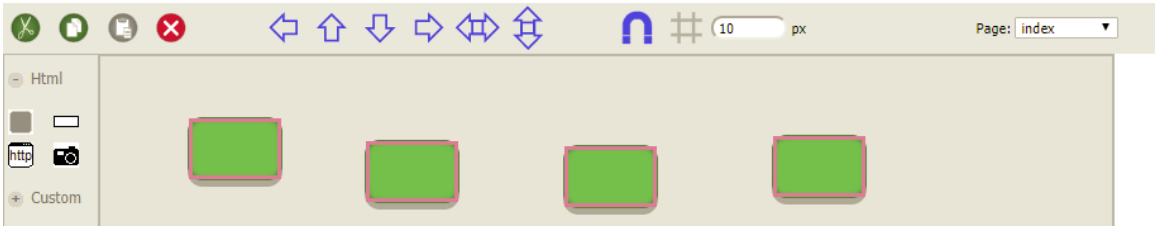
Cut, Copy, Paste, Delete

The top tool bar has several tools to help with the layout of components:

These are the standard Cut, Copy, Paste, and Delete tools. They can be used with any components currently selected on the panel. You can select multiple components by holding the shift key while you click

Alignment Tools

These tools are alignment tools and will only be available for use if you have multiple components selected in the panel. For example, drag three or four buttons into the panel. Then while holding the Shift or Control key click on each component until the red select box is around each of them:



Once more than one component are selected, the align tools will become enabled. These tools are:

Align Left

Align Top

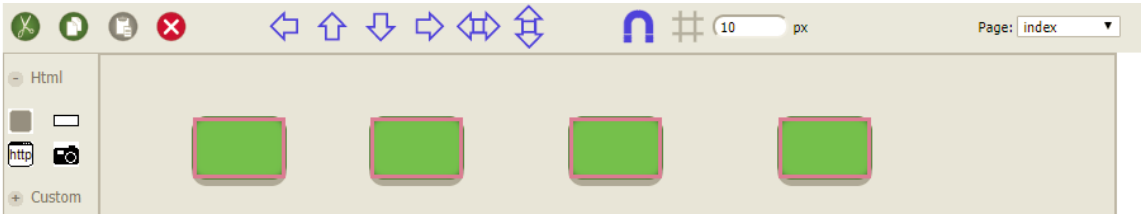
Align Bottom

Align Right

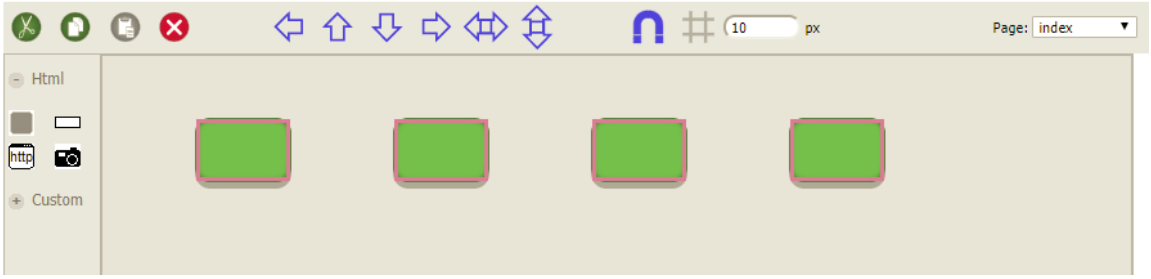
Spread Horizontally

Spread Vertically

In each case the system will look for the most extreme edge and align to that. For example, with the 4 buttons selected in the example above, clicking the align top will find the selected button that is closest to the top of the panel and align all of the buttons so that their tops match that button.

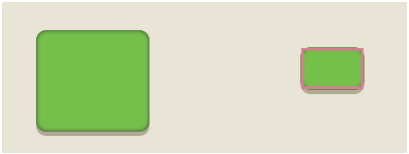


Clicking the Spread Horizontally tool will make them spread evenly between the left and right button

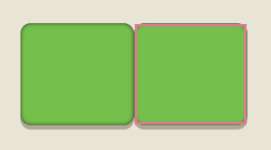


Magnet

The magnet tool is enabled by default but may be disabled by clicking on the tool. This tool helps to align and resize items of like kind edge to edge. For example, create a new button and resize it. And then add a second button and leave it at the default size.

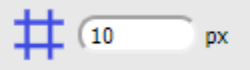


If you then drag the small button so that its edge meets the large button it will immediately snap to the size of the large button and align itself to the large button's edge.



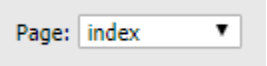
This is extremely useful when trying to build a panel with many same sized components lined up edge to edge. However, it can also be disconcerting when you want the components to be different sizes and/or not to line up. So always be aware of whether the magnet tool is on or off when trying to align components.

Grid

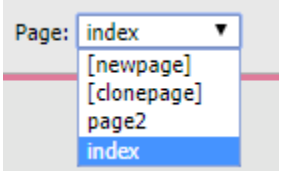


The grid tool if enabled will align component's location and size according to a grid of a specified pixel density. For example, if you enabled the grid with a pixel density of 10 pixels, you will notice that dragging components will jump by 10 pixels. This is useful when trying to evenly align components.

Page



The page tool allows you to create additional pages for a panel. Buttons may be created to switch between the pages. The method to make a panel switch pages will be described in more detail later. However, to design a new page, click on the down arrow and select the [newpage] option.

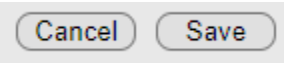


After clicking the new page option click save to give the new page a name. Then you can design a new page as if it was any other panel. Each panel has an index page which is the default page that will be loaded when the panel is first displayed. You can select any page to edit by clicking the page in the drop down list. You can clone your work to a new page by selecting the page to be cloned and then selecting the clone page 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.

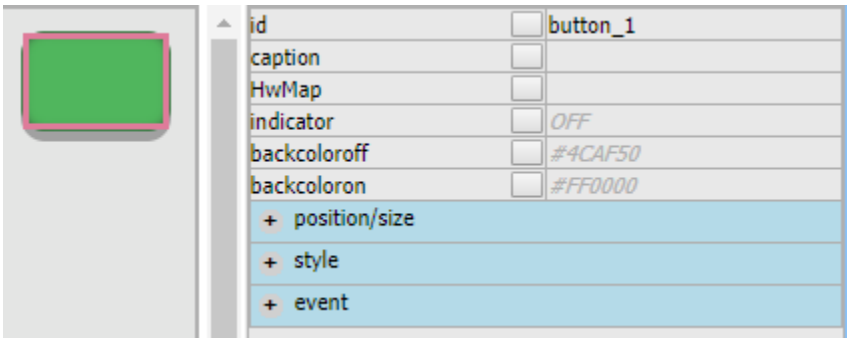
Cancel/Save

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.



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.



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 addition sections and properties.

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

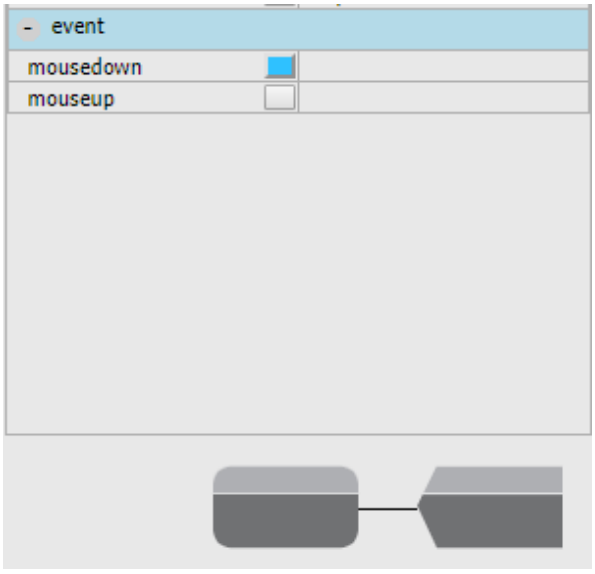
Each property in the property grid has a button between the property name and the property value at the end of the property name side of the grid. This is called a bind button.

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

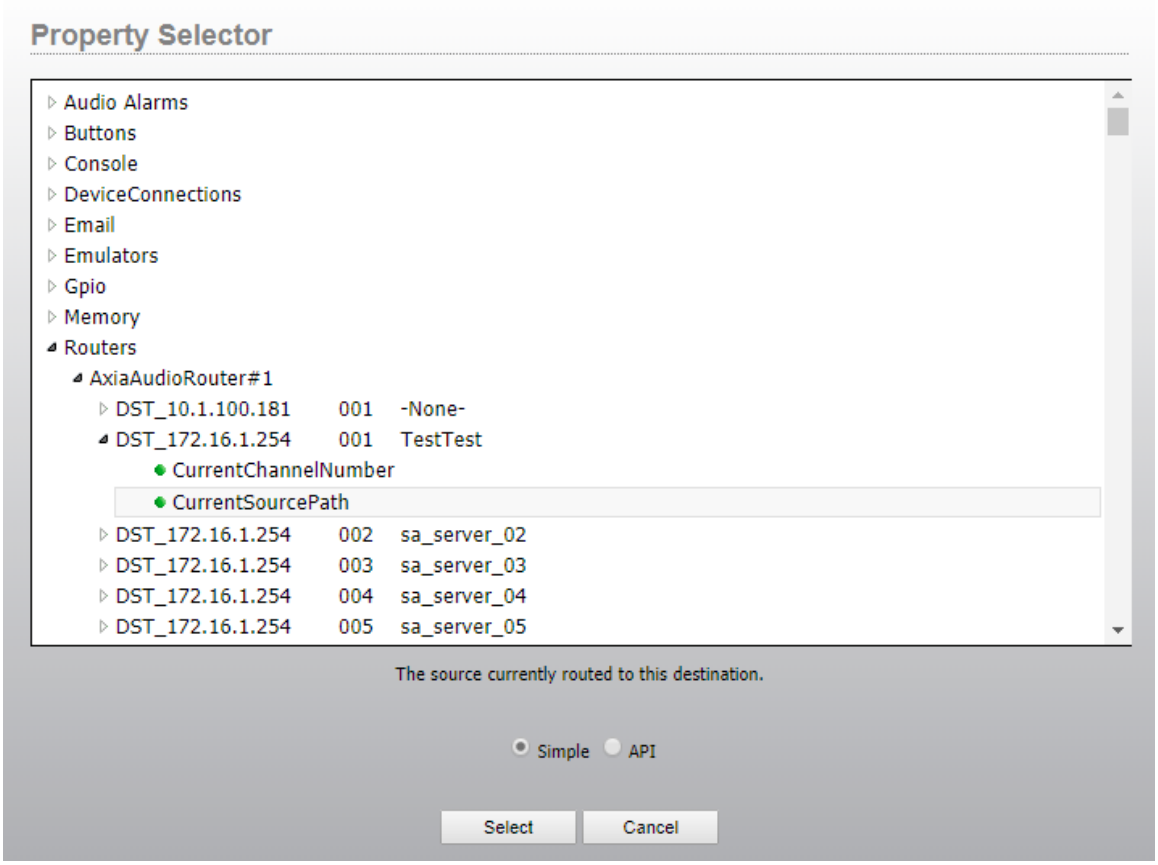
You may also notice that after enabling a property for binding that an image of a partial logic flow will appear at the bottom of the property grid:



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.

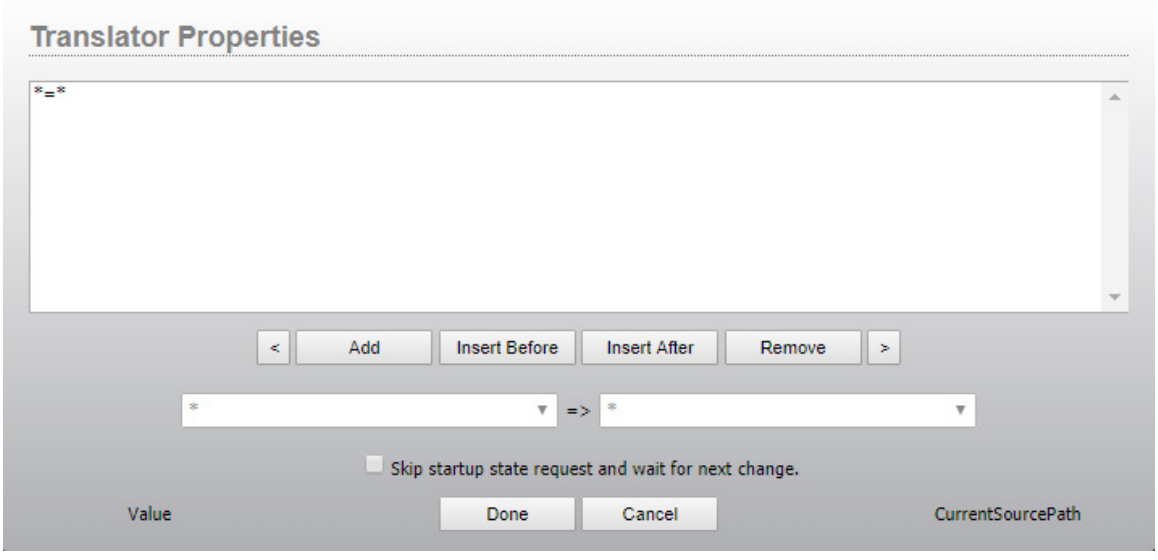
Let's say we want the button we have dragged onto the panel to trigger a route change. Select the button and enable the binding button on the mousedown event. We are defining what we want to have happen when the button is pressed. Then double click on the endpoint in the flow image.

This will open the normal property selection dialog used in logic flows:

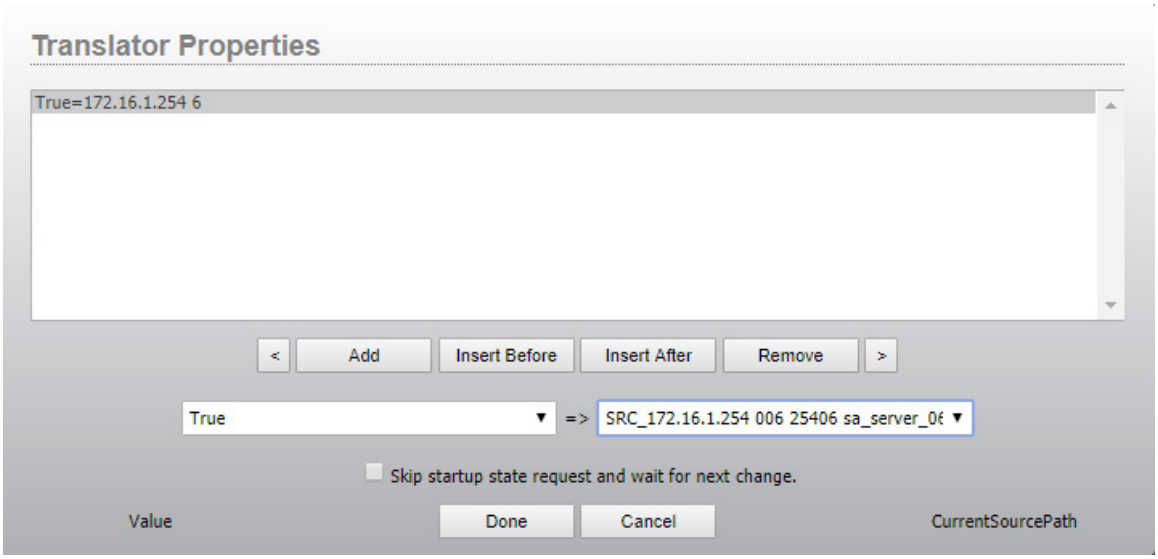


Expand the Routers section, expand a router, and expand the destination you want to change when the button is pushed, and then click on the CurrentSourcePath Property. Then click select.

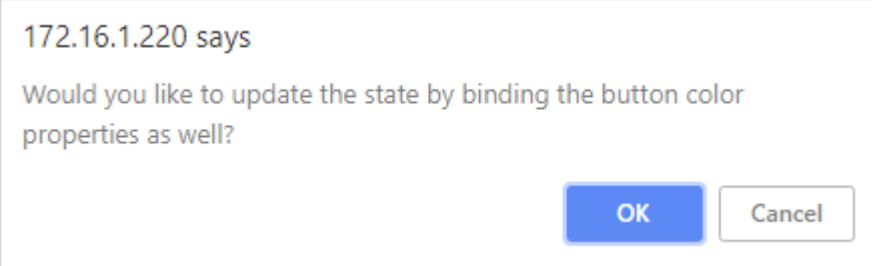
The system will automatically move to the translation dialog.



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.

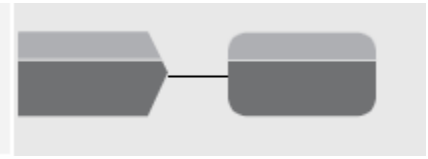


We have just defined that if the mousedown event is true, the sa_server_06 source will get routed to the TestTest destination. However, when you click Done you will get a pop up message.



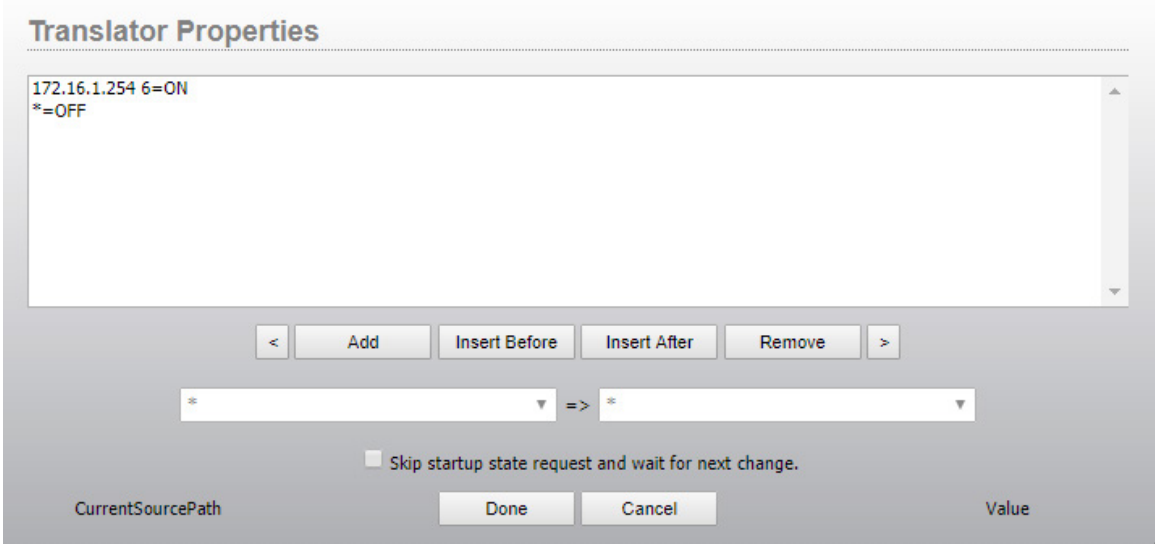
This message will only appear if you are generating flows on the mousedown or indicator properties of a button. In this case it knows that since we are defining what we want the button to do, we probably also want some indication on the button that the requested action has been done. If we click OK, 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 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 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.



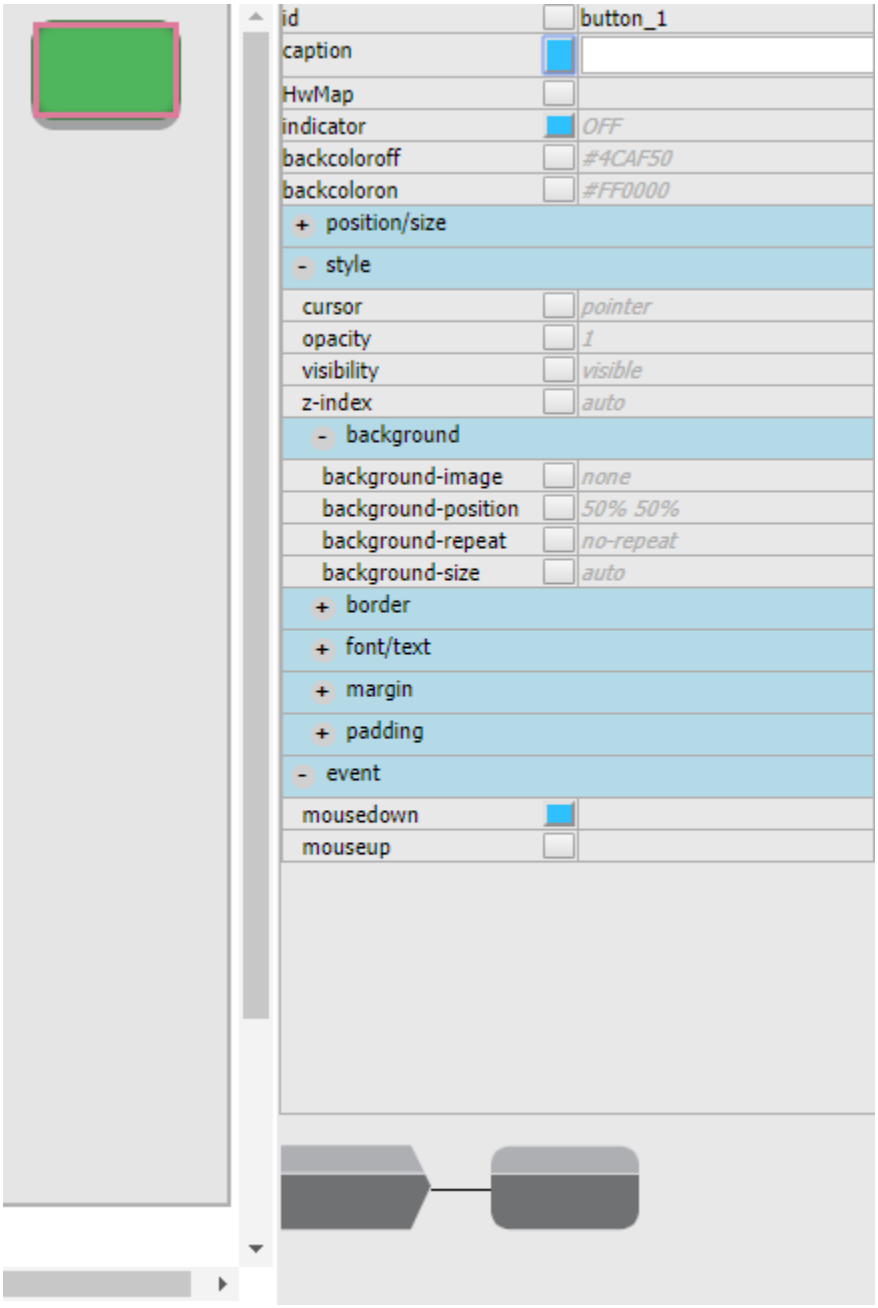
You will also notice that the system is assuming you will want the indicator to be on if the selected source is routed to the destination and off if it is not.

Click Done.

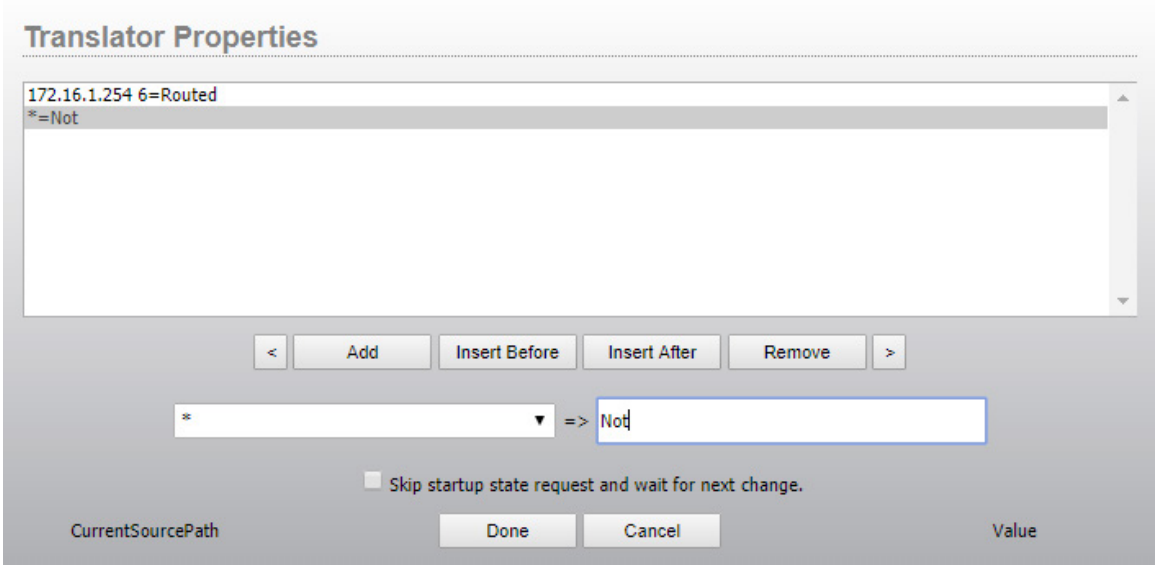
You will notice that the flows are no longer gray and have turned blue to indicate they have been defined. Saving the panel will cause the flows to be created and start working in logic flows. Flows created in this manner will be generated in a special folder in logic flows called _panels. The flows 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 sometimes useful to have this open in a separate browser tab while you are working. Then after you save 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.

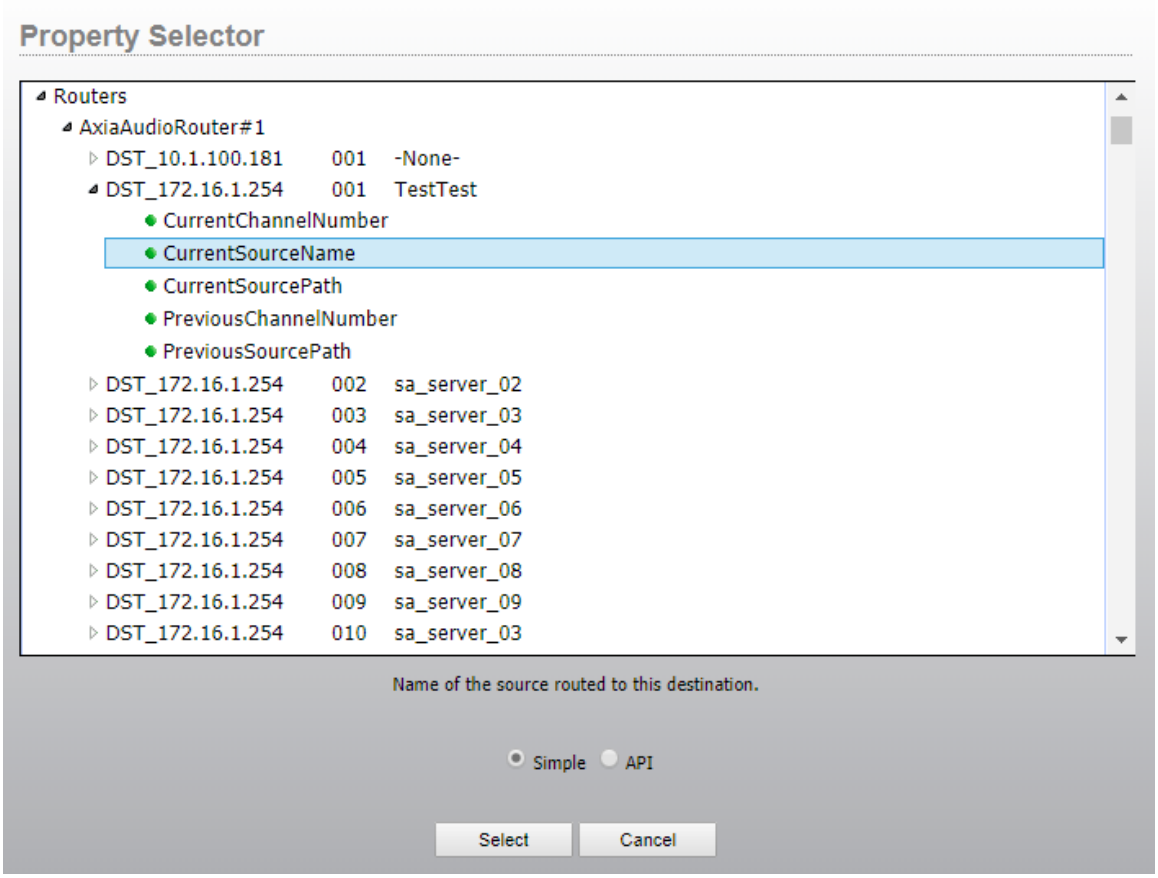


Now double click on the start point and select the same CurrentSourcePath property of the same destination. Now in the translation select what you want the button to say when the source is routed and what you want it to say when it is not.

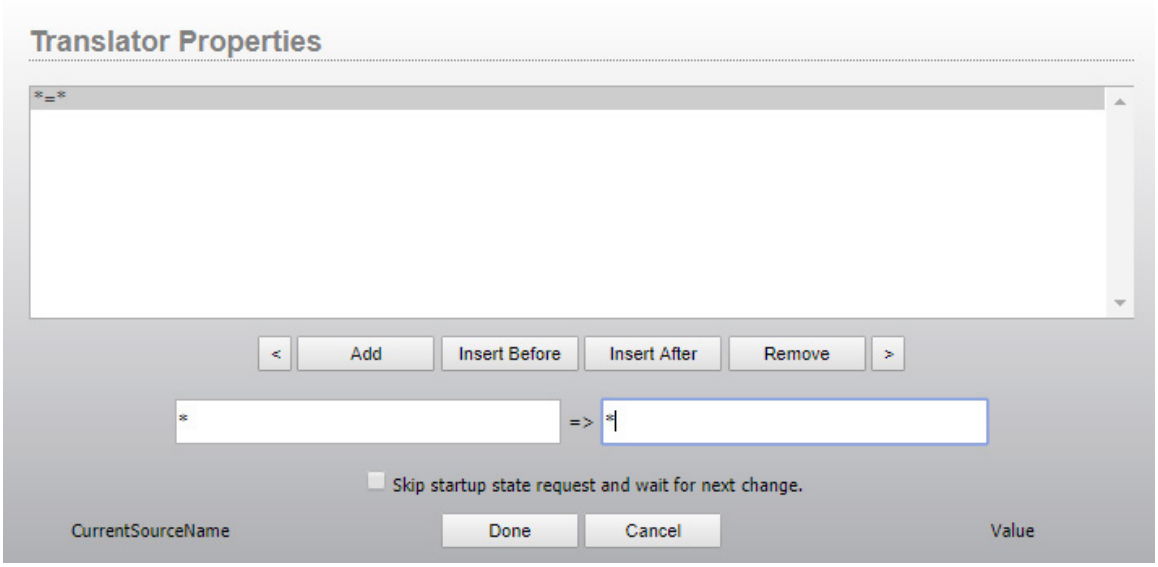


Click Done and Save to save your changes. Executing the panel should now display Routed or Not in the buttons caption depending on whether the selected source is routed to the destination.

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.



Now pick the currentSourceName Property instead of the CurrentSourcePath property and click select.



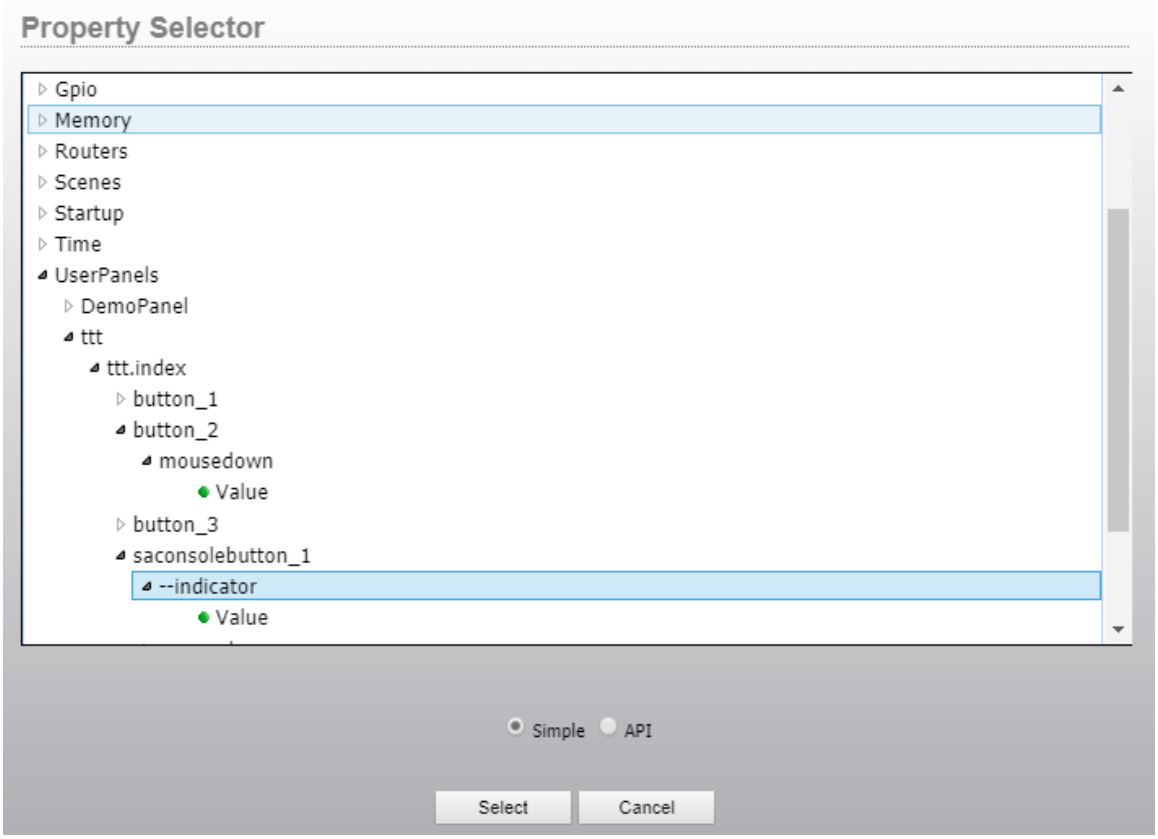
Change the translation to be *=*. Then click Done. Now the button's caption will be tied to the name of whatever source is currently routed to the destination.

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 light or unlight 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

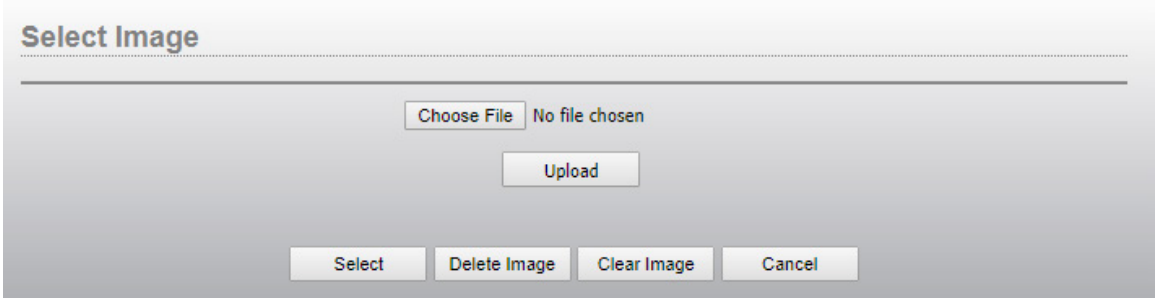
In many cases you may wish to create more complex flows than described in the examples above. For example, you may want your indicator state on a button to be the product of numerous conditions in the system. Those kinds of flows can still easily be created, but they must be created within the logic 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.



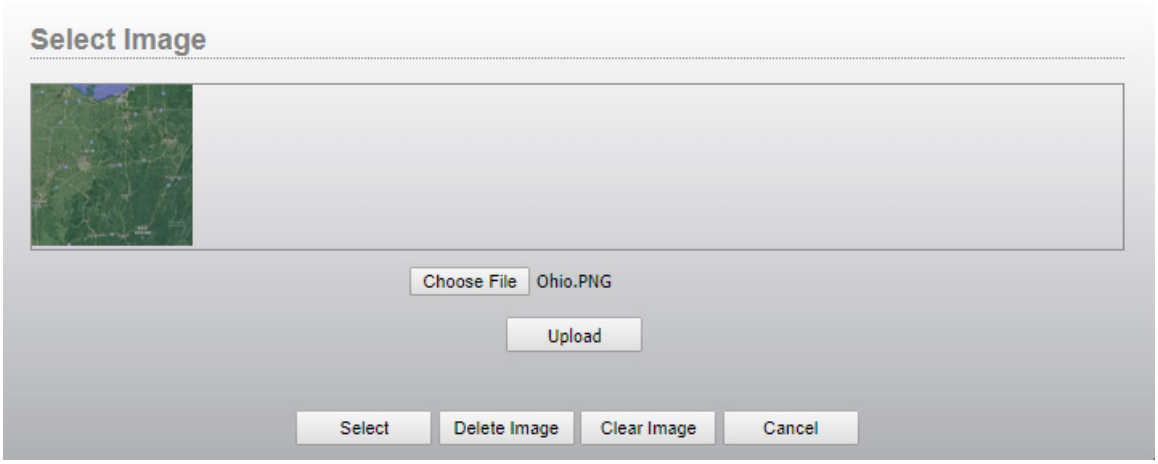
Changing Pages

The toolbars section above talked about multiple pages within a panel. This section will use the information learned above to create a panel that switches between multiple pages.

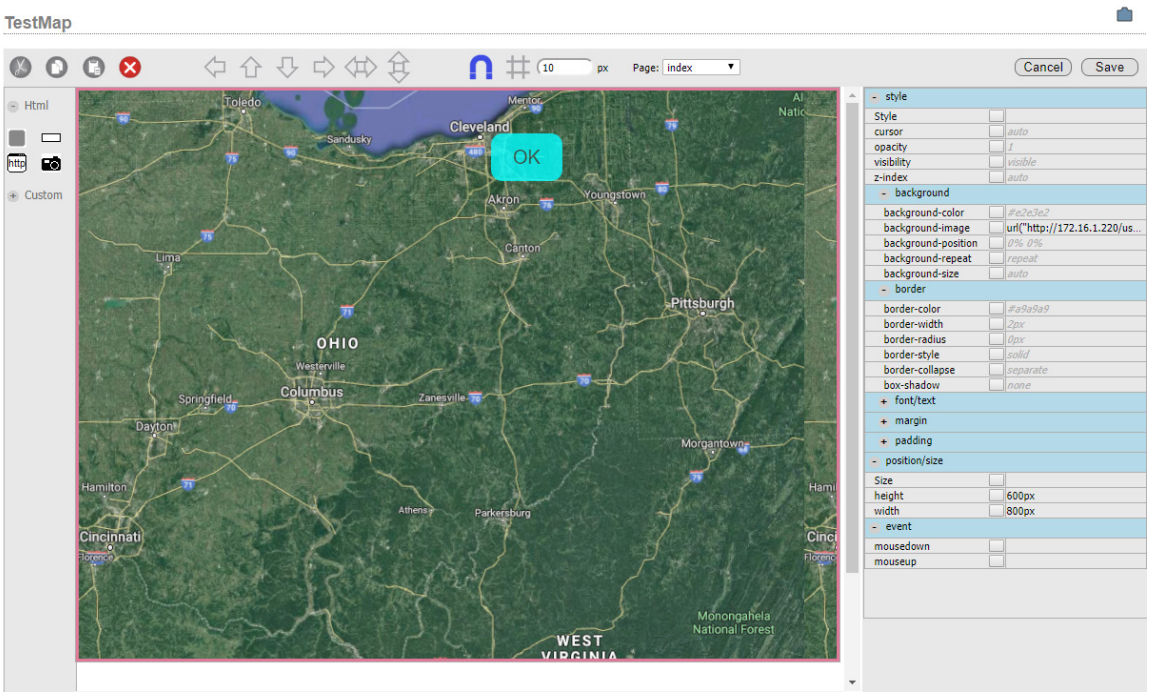
First create a new panel and Save it as TestMap. Click on the panel and then in the property grid, expand the style and background sections. Click on the value for the background-image property. A dialog will appear for selecting and uploading images.



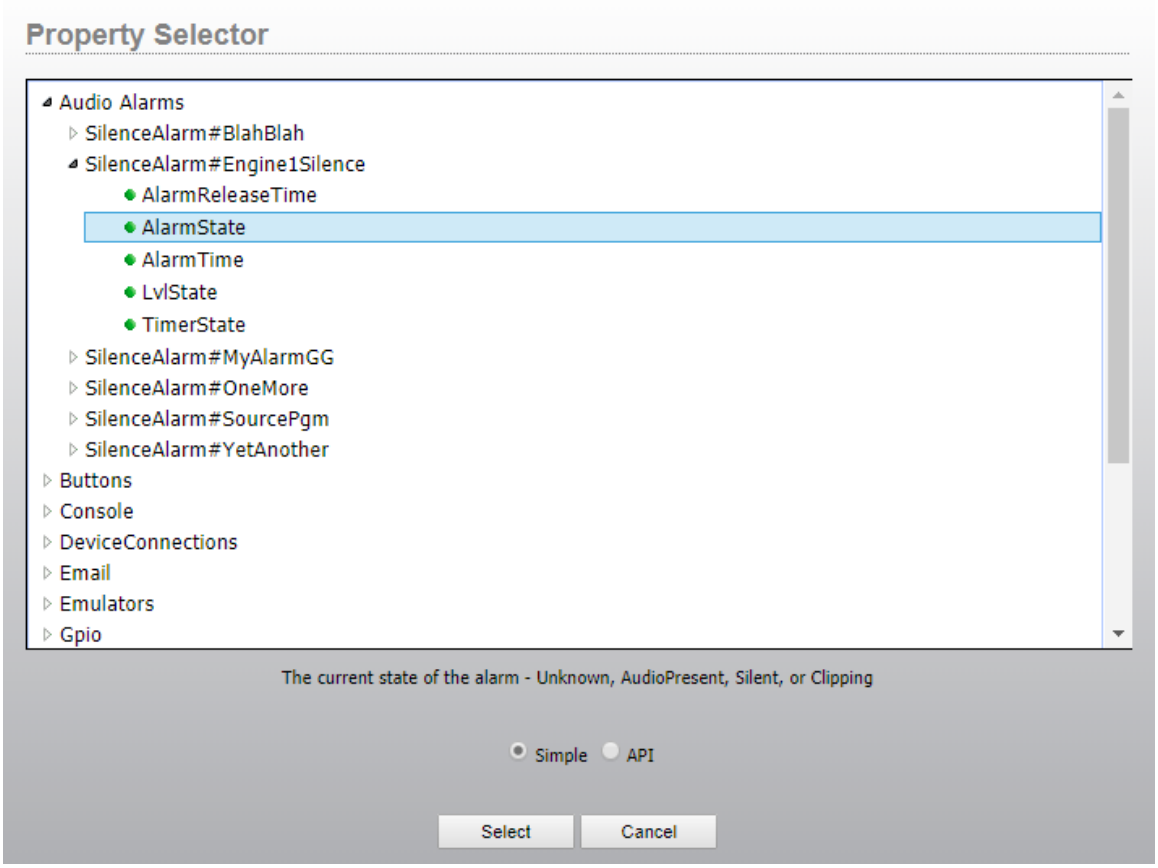
In this case I am going to upload an image of a map of Ohio. Click Choose File and select your stored image and then click Upload. The image will then appear in the selection list.

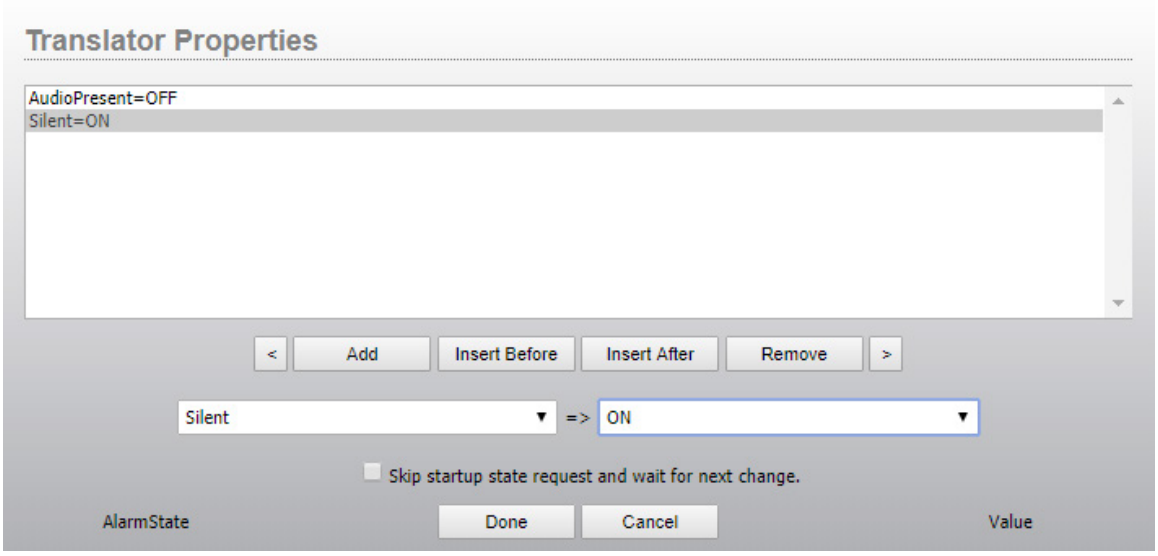


Click on the image and click Select. The background of the panel will now display the image of Ohio. Drag a button on and position it next to Cleveland and then update the caption and color properties of the button to read Ok and select cyan for the backcoloroff. Under the border section, set the box shadow to none. Then set the opacity property to .8. Your panel should now look something like:



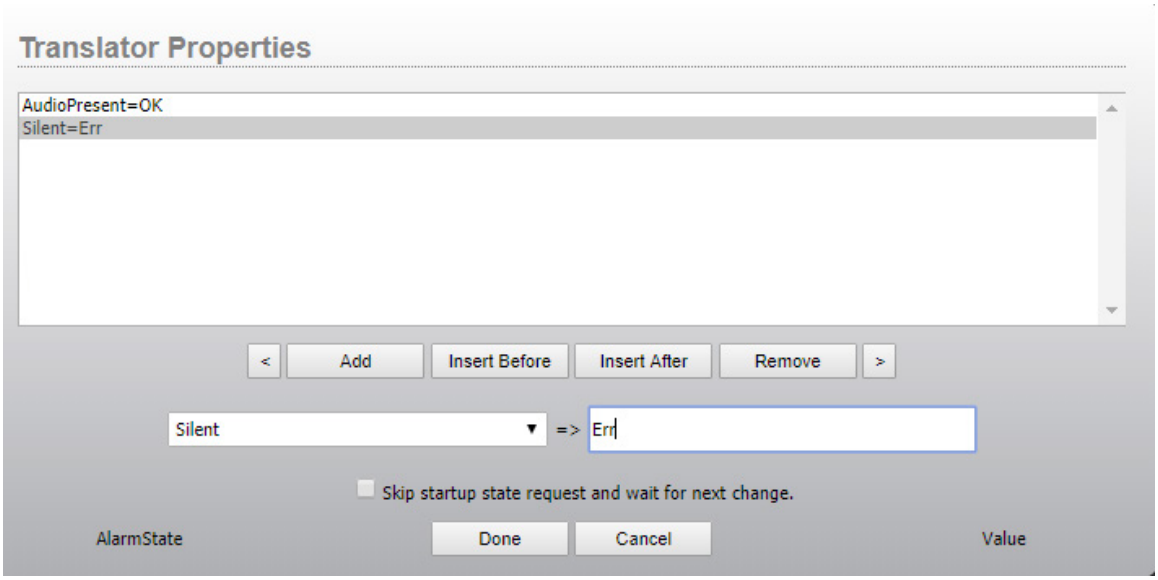
Next enable the binding for the indicator state of the button. Then click on the start point and select a silence alarm state as the start point. For the translation select audio presence as indicator off and silence for on.





After clicking Done, click cancel when it asks about the mouse down action. 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.

Next turn the binding for the caption property on and double click the start point of that flow. Select the same Alarm State property, but for the translation make silence convert to Err and Audio Present to OK.

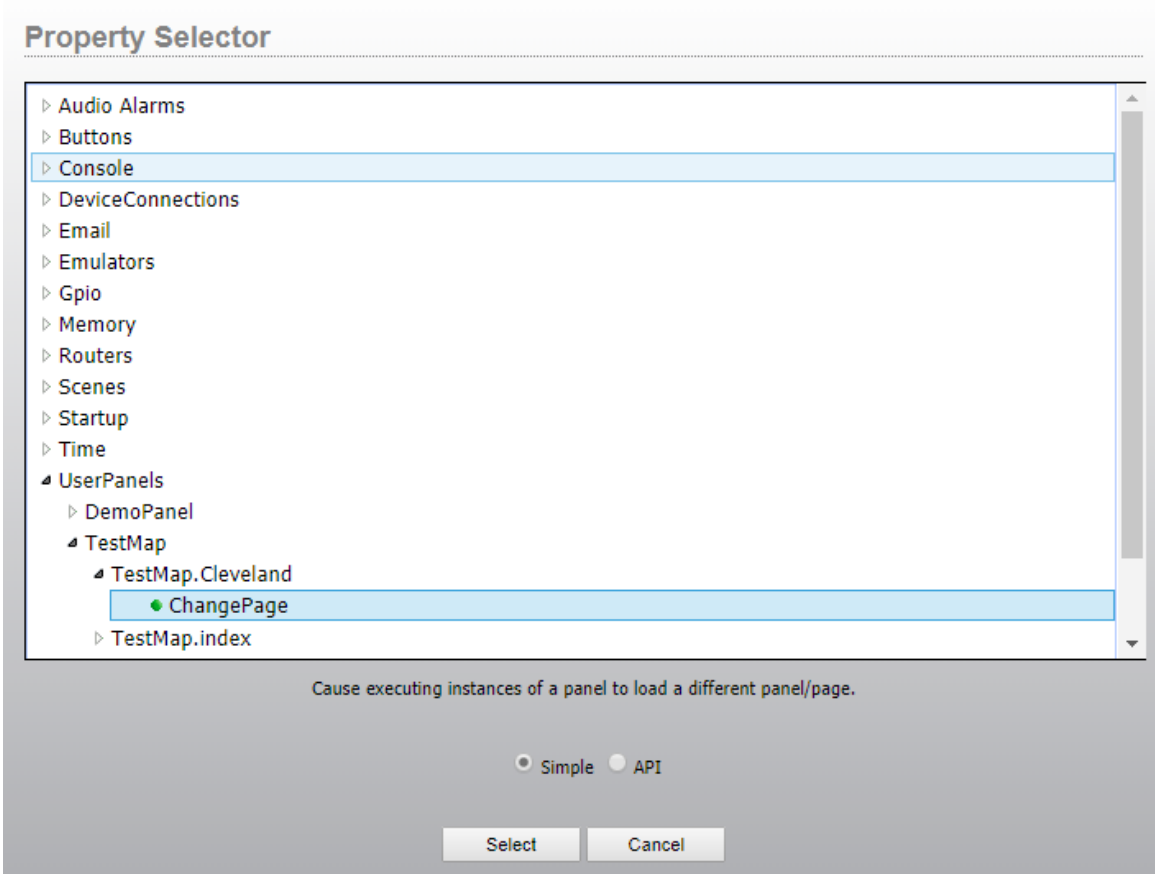


Click Done and the save the panel.

At this point if we were to execute this panel, the button next to Cleveland should show cyan and OK if the silence alarm has audio presence and Err and Red if it is silent.

Next from the page drop down select [NewPage]. Save the New page with the name Cleveland. Drag a new button onto the new page and set its caption to return. Turn on the binding for mouse down on this button and then click the end point. Expand the UserPanels section of the logic flow property selector and the TestMap.Cleveland section and select the ChangePage property.

Note: If you do not find TestMap.Cleveland in the logic flow property selector it means you have not saved the new page yet.



In the translation set True=TestMap.index. Click Done and click cancel for updating the indicator property. Then save the Cleveland page again. 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.

Now use the page drop down to select the index page again. Click the OK button. Turn the binding for the mouse down event on, click the endpoint, select the TestMap.index ChangePage property and then in the translation set true=TestMap.Cleveland. Click Cancel for the binding indicator question and save the panel.

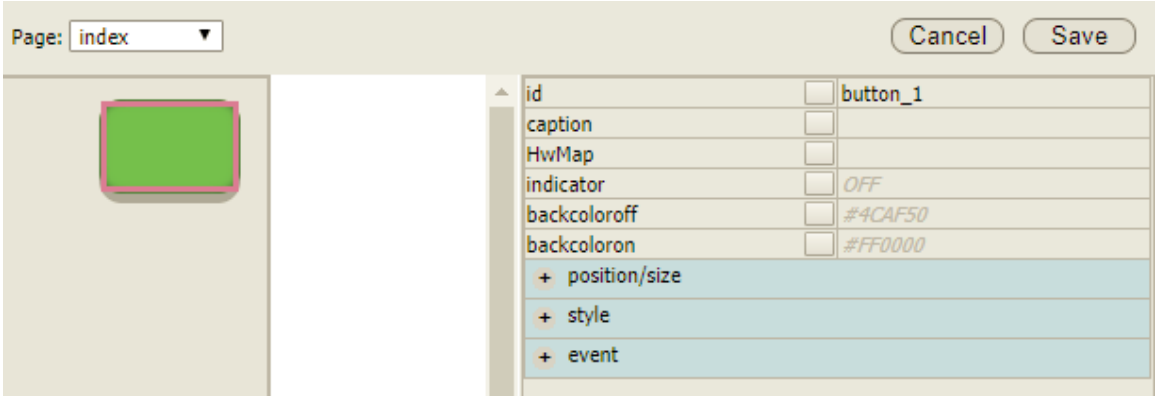
At this point, execute the panel. The button on the main map should switch caption and indicator color based on the silence alarm state and clicking this button should take you to a new panel called Cleveland. Clicking the return button should take you back to the main map index page.

It is important to note that you are always selecting the changepage property of the current page you are on and setting its value through the translation dialog to the page you want to move to. Another interesting point is that the change page property is also available to normal logic flows. So, for example rather than binding this to the mouse down 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 change page property is not limited to pages within the same panel. You could switch to a completely different panel.

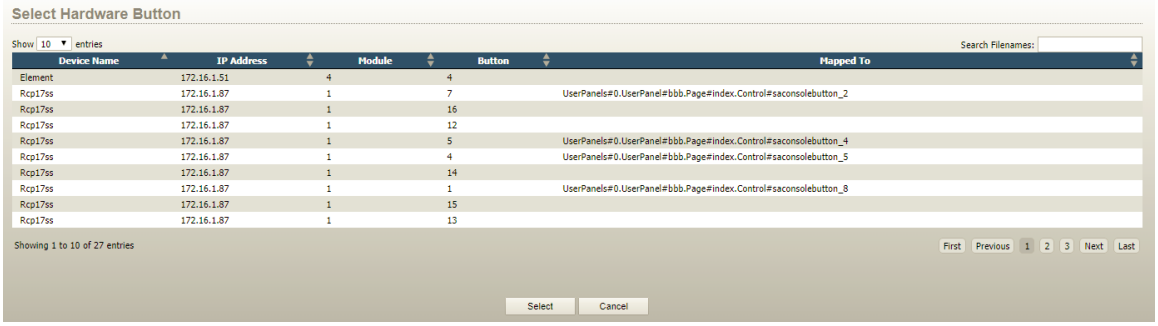
Obviously in a real use case we would fill the Cleveland page with more information than just a return button. For example, we might create a system flow chart page with meters and buttons and silence alarm states of various parts of the chain to more easily determine where the failure occurred.

Hardware mapping buttons

Buttons created in a user panel (html or console) may also be hardware 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 button created in a user panel.



Then click on the HwMap field. A dialog will appear from which you can select known buttons in the system.



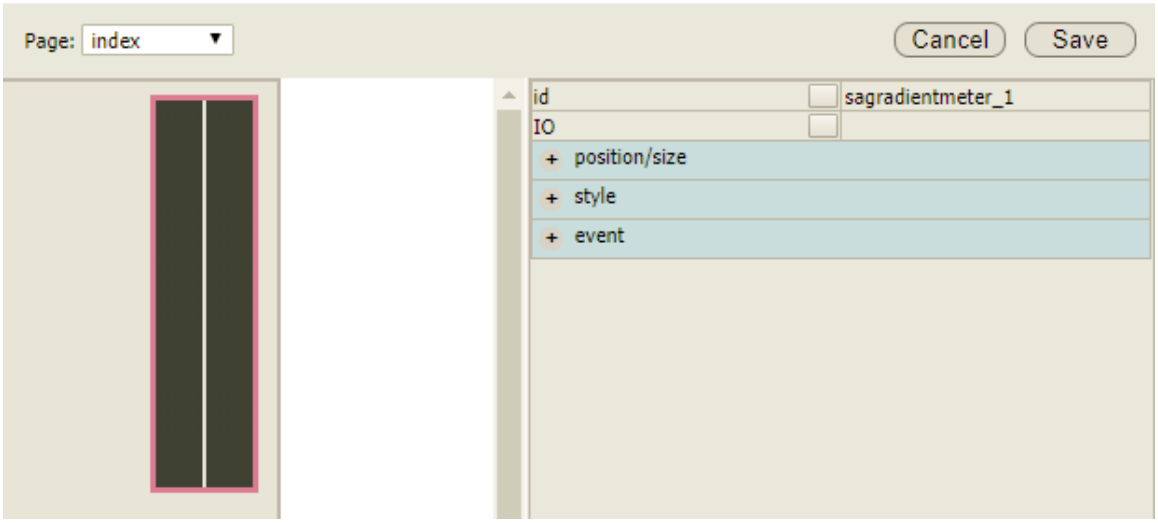
The mapped to field will display if the button has already been hardware mapped as each hardware button can only be mapped to a single software button. Physical buttons do not have to be hardware mapped. They can also just be used directly with logic flows. If the button you want is not shown, make sure the Lcd panel or console is in the devices list. If not, it needs to be discovered. See the PathfinderCore PRO manual for more details on discovering devices. Note that Lcd panels must be manually discovered in devices using the add and Lwcp discovery. If the device exists in the system and it is a console but it 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.

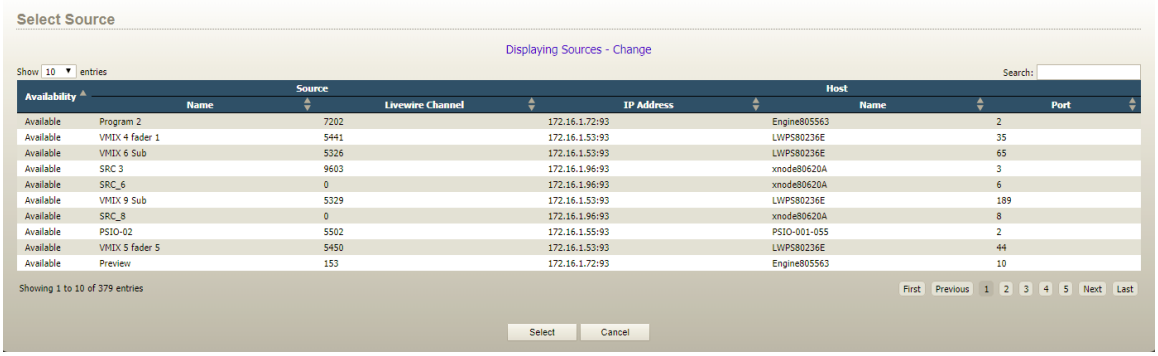
Once the panel is saved hardware mapping in html panels takes place natively in the application and does not require hardware map logic flows like the legacy panels did.

IO selection for meters and faders

Under the custom components there are two different kinds of meters and two different kinds of faders. The components section below will provide details on these components. But for all of these components the method for selecting the IO which the component will display is the IO property. For example, drag a gradient meter onto the palette and resize it to an appropriate size.



Clicking in the IO property field will open an IO selection dialog box.

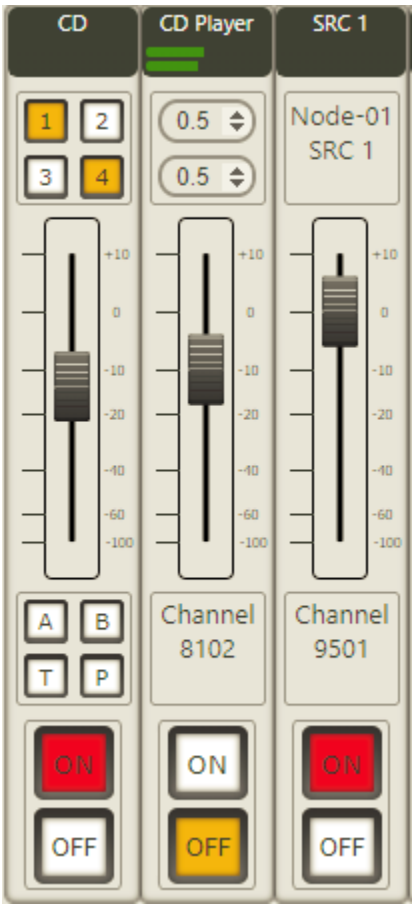


The link at the top of the page will display whether sources or destinations are currently being displayed and clicking the link will change between the two. You can use the search box to narrow down the list within either sources or destinations. Click an IO whose metering you want to be tied to the meter and click select. The IO field will fill with the path of the selected IO. Since we are in the designer, no metering will be displayed. It will only display when the panel is executed.

It is also important to note that turning the binding button on for the IO parameter will make it available to logic flows such that you can dynamically change the IO assigned to the meter or fader.

ConsoleFader

Some special attention should be given to the ConsoleFader component. 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 and so the component changes to show the time up and down parameters. And the third shows an XNode source.

Components

Each component has a whole variety of css properties and component events that are available. As mentioned previously in this document many of these can be understood by googling web page css descriptions. This section will describe each component as well as custom properties of that component that are beyond the standard css style properties. While we have included images of the components below it is important to note that the css properties can be used to make the components look much different than the default we are showing below. Feel free to adjust border, color, and shadowing properties to achieve the design desired.

Important Note: Each component has an ID property. This allows 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 – Button

The button component is a button that can be designed with different colors, borders, background pictures, etc and can then be used to control and indicate changes in the system. This component has been covered in detail in the examples above. It is important to note that 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.

Custom properties include:

- 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 behavior 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 properties.
- backccoloron: color used when the indicator state is on.
- backccoloroff: color used when the indicator state is off.

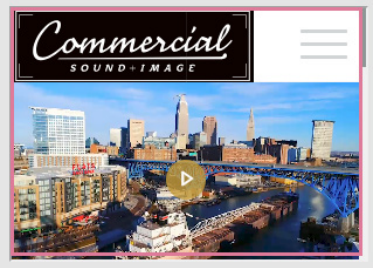
Html - Label

Labels can be used to generate textual label information in the system. The caption of the label may be dynamically updated by flows by binding the textContent property.

Custom properties include:

- textContent: The textual information displayed by the label.

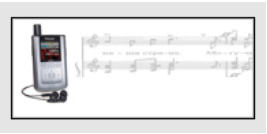
Html - Web page



The web page component allows you to embed a web page from another site into the panel. This component is an html iframe. It is important to note especially during testing that some sites (such as google) prevent their content from being displayed in an iframe. This component can be used to display video streams and other web page content. Use the src property to enter the url for 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.

Html – Image



This allows you to embed an image in a page. However, most elements also support the background-image property. For example, you can put an image on the panel itself without using an image component and you can put background images on buttons and labels as well. This component can also 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.

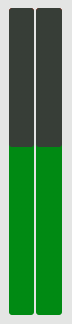
Custom – Led Meter

This component can be tied to audio ios that support metering in the system. Use the IO property to display a list of sources or destinations which can be assigned to the meter.

Custom properties include:

- IO: Used to select the audio io this meter will display.
- orientation: Used to select whether the meter will display horizontally or vertically.
- metercale: Used to select the scale of the meter. Options include standard, linear, and british.
- metrics: Used to define whether the numerical values for the 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 used for each of three sections of the meter.
- Led\border: used to adjust the border settings of the individual led blocks.

Custom – Gradient Meter



This component can be tied to audio ios that support metering in the system. Use the IO property to display a list of sources or destinations which can be assigned to the meter.

Custom properties include:

- IO: Used to select the audio io this meter will display.
- orientation: Used to select whether the meter will display horizontally or vertically.
- metercale: Used to select the scale of the meter. Options include standard, linear, and british.
- metrics: Used to define whether the numerical values for the meter are displayed next to the meter or not. Options include none, lefttop, middle, rightbottom.
- style\optimum percent: 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 used for each of three sections of the meter.

Custom – Analog Clock

This component displays the current time using an analog clock style.

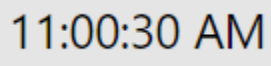
Custom – Analog Coutdown

This component allows you to define a trigger a countdown. The clock will display the countdown value.

Custom properties include:

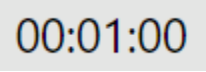
- countdownlength: time in seconds for the countdown.
- countdownstart: generally exposed via bindings for a logic flow to trigger the start of a countdown. Options are true or false.

Custom – Digital Clock



This component displays the current time using a digital clock style.

Custom – Digital Coutdown

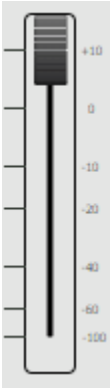


This component allows you to define and trigger a countdown. The clock will display the countdown value.

Custom properties include:

- countdownlength: time in seconds for the countdown.
- countdownstart: generally exposed via bindings for a logic flow to trigger the start of a countdown. Options are true or false.

Custom – Fader



This component can be tied to audio ios whose gain may be manipulated. Use the IO property to select the controllable source or destination.

Custom properties include:

- IO: used to elect the IO whose gain you wish to control.
- metrics: how the numbering for the fader is displayed. Options include none and lefttop.
- slider-height: percentage of the overall component height used for the slider height.
- slidebarwidth: width in pixels of the bar on which the slider rides.
- metricoffset: percent of width used for the metrics.
- metriclinecolor: color for the lines drawn for each metric line.
- slidebarcolor: color of the bar on which the fader rides.
- slidebarradius: radius in pixels
- slidebardisplay: whether the slider bar is displayed.
- faderimage: rather than using the default designed slider, an image may be used.
- slider-margin-left: margin offset for the slider.
- slider-border-style: border style for the slider.
- slider-border-width: border width for the slider.
- slider-border-radius: border radius for the slider.
- slider-border-color: border color for the slider.
- optimum: gain level which is designed to be optimum or unity.
- type: whether this is an audio slider or a slider for controlling other numeric values. Options are audio and linear. Note linear option and the ability to control other numeric values is for future use.
- autosizefont: whether the metric font automatically scales as the size of the fader is adjusted.

Important Note: Currently Qor faders are not supported by this component in the first release. Additionally it currently uses the fach object for fusion faders. Support for Qor and Lwch will be added in the near future.

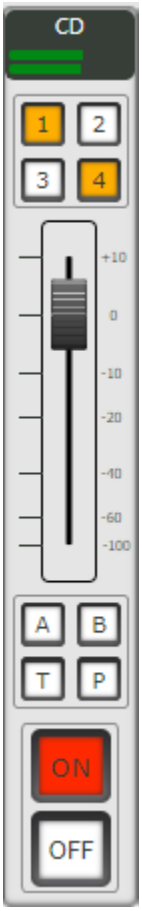
Custom – Console Button

This button works the same as the html button but has a more interesting look.

Custom properties include:

- 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 behavior 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 properties.
- 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.

Custom – Console Fader



This fader is a smart fader that displays a variety of things depending on the type of io assigned to it. In the example above, this has been assigned to a Fusion console input. When executed the component understands the type of IO to which it has been assigned and updates the controls accordingly. In this case it shows the name of the source profile, a meter obtained from the input stage of the fader, the program buss assignments, a fader which maps to the fusion fader, A and B user buttons, talk and preview buttons, and On and Off. The Io can be defined using the IO property.

Custom properties include:

- IO: used to elect the IO whose gain you wish to control.

Important Note: Currently Qor faders are not supported by this component in the first release. Additionally it currently uses the fach object for fusion faders. Support for Qor and Lwch will be added in the near future.

Launching the panel from a Desktop icon (windows)

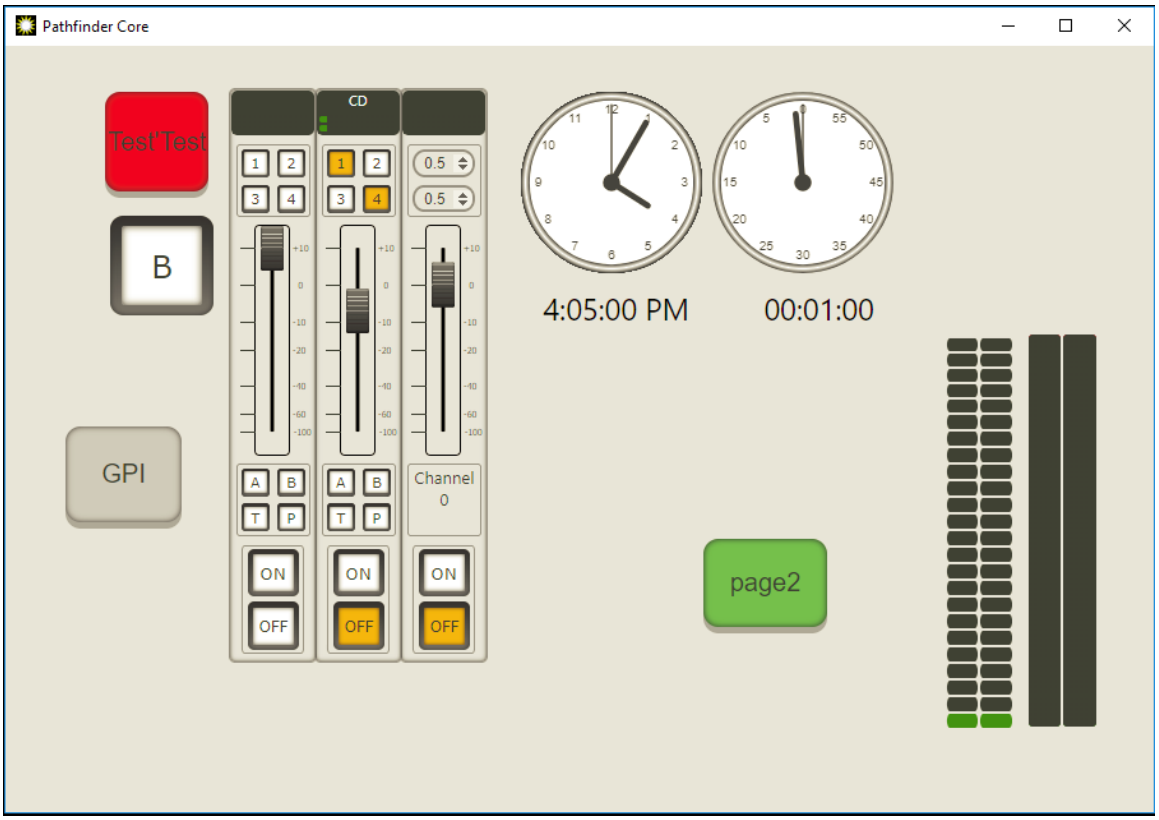
If you are using chrome, there are some command line options that will allow you to launch a PathfinderCore PRO panel as if it was an application. Copy and paste the following into a notepad or text editor:

"C:\Program Files (x86)\Google\Chrome\Application\chrome.exe" --app="data:text/html,<html><body><script>window.location='http://Admin:Admin@172.16.1.220/userpanelframemin.php?panel=ttt&page=index';</script></body></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 the Username:Password to be one that reflects your system. Also change the ip address (172.16.1.220) to match the ip address of your Pathfinder Core PRO. Finally, after panel= change the name of the panel you want to launch from ttt to the name of your panel. If you want to target a page other than the default index page, change that as well. Once you have the edits made, select the entire text again and copy it to your clipboard.

Now, right click on your desktop and select new shortcut. Paste the text into the location of the item field. Click next and input a name into the shortcut and click Finish.

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><body><script>
window.moveTo(580,240); window.location='http://Admin:Admin@172.16.1.220/userpanelframemin.
php?panel=ttt&page=index';</script></body></html>"

In the future we will investigate a way to generate the shortcut (or at least the shortcut text) automatically.

Legacy User Panels

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. While this is still supported and therefore covered in this manual, it is deprecated. And we would urge all users to migrate to the much more powerful HTML5 panels. The panels created in this manner are called Legacy User Panels and they 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.

The PanelDesigner application is used to create legacy user panels. See Chapter 19 PanelDesigner to review how to design and create panels.

After the legacy panels have been created and saved using PanelDesigner, the "User Panels" link in the navigation bar will display the Panels that exist in the system.

This page will display the name, caption, and most recent version number in the form of a date/time stamp for each panel. Panels may also be removed from the system using the minus icon next to each panel on this page.

Once the panel is created, properties for each of the panel's controls are available to be used within logic flows. There is a wide list of properties that are accessible to be used by logic flows in order to change colors, text, and pictures on a panel as well as to react to button clicks and other control changes.

The list of available properties may be reviewed by browsing the start and endpoint property selection tree within logic flows once the requisite panel has been created in the system. Additionally, most of these properties match the property list described in Chapter 19 PanelDesigner. Review that list of properties if you have questions as to a particular property's purpose.

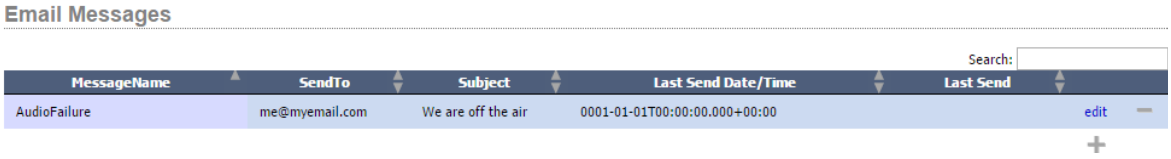
Once these legacy panels exist in Pathfinder Core PRO, they can be opened either in PathfinderPC Client or by using the configuration file with PathfinderPC Mini. See details on how to use these panels in Chapter 18 PathfinderPC_Core Client, and also Chapter 20 PathfinderPC_Core Mini.

Chapter 12

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 fairly common to send e-mails to a variety of engineers when silence is detected at a transmitter site. In order 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 Chapter 3 System, section 9 e-mail.

Once the e-mail host has been configured and test e-mails can be sent successfully, click on the Email Messages link in the navigation bar.

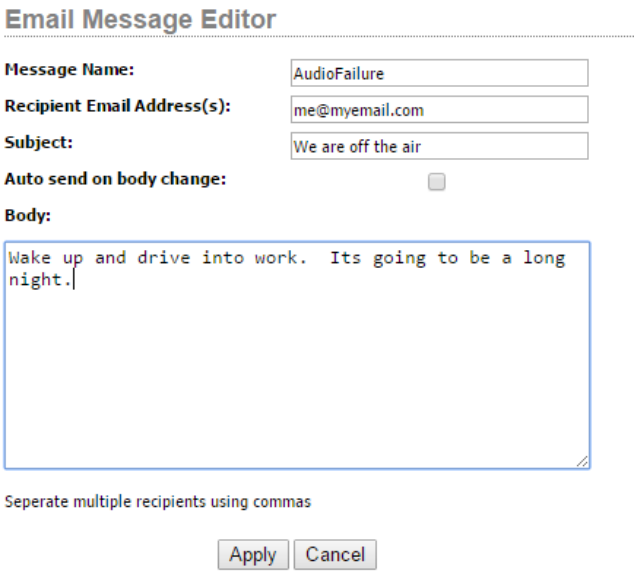


This is where you can create e-mail messages that can be sent when specific actions take place in the system. The information included on this page includes the following five columns for each e-mail message:

- Message Name: The unique name of the e-mail message.
- SendTo: Displays the e-mail addresses to which the message will be sent.
- Subject: Displays the subject to be used by the e-mail message.
- Last Send Date/Time: Displays the last time the message was sent.
- Last Send: Displays information about whether the last send attempt was successful or failed.
- Edit: Displays a link to edit the configuration of an e-mail message.

As usual, the minus and plus icon can be used to delete e-mail messages from the system or to create a new e-mail message.

When you add a new e-mail message, the following dialog will be presented:



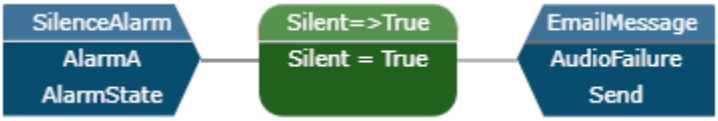
The Message Name property identifies the e-mail when it is accessed via logic flows or API. The Recipient Email Address field is where you can place a comma delineated list of e-mail addresses to which the message will be sent. The auto send on body change check box allows you to define new body content in your e-mail via a logic flow and have it send automatically without also having to set the Send Property. The Body is where you place the text of the e-mail message. You can embed the date and time into the subject or body of an email by using <%DateTime%>.

In this case when the email message gets sent, the subject will look like:
Hello YYY – 2017-12-14T13:51:28.033-05:00

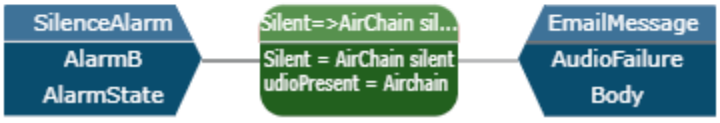
And the Body will look like:

Howdy YYY – 2017-12-14T13:51:28.033-05:00

Once created this e-mail message becomes available to be used by logic flows. In the following example, we will send the Audio Failure email message whenever MyAlarm's AlarmState becomes Silent. We accomplish this by setting the write only property called 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 particular start point. For example:



Here we are actually using 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.

Chapter 13

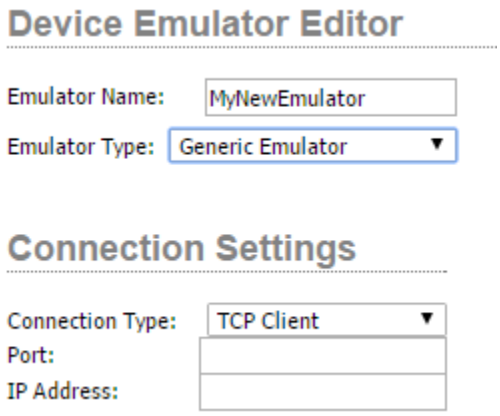
Device Emulators

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. Clicking on the Device Emulators link in the navigation bar will display the list of device emulators currently defined in the system.

The following four columns are present on this page:

- Name: The name of the emulator.
- Type: The type of the emulator (Generic, Probel General Router, Probel General Switcher).
- Connection Type: The connection type to be used with the emulator which can either be TCP Client or TCP Listener.
- Port: The TCP port to be used for the connection.

Click the plus icon to add a new emulator.



Pathfinder Core PRO currently supports three types of device emulation. Each emulator type may present slightly different configuration options depending on the emulator type, and those configuration options are described below. However, in each case you need to provide a name for the emulator, select the emulator type, and then define the connection parameters.

The connection type can either be TCP Client or TCP Listener. The primary difference between the two is whether the system that will be using this emulator will establish the connection to Pathfinder Core PRO or whether Pathfinder Core PRO should initiate the connection to the other device. If the other device will connect to

Pathfinder Core PRO, then Pathfinder Core PRO needs to listen for that device. In that case you should select the TCP Listener option and define the port on which Pathfinder Core PRO will listen. If Pathfinder Core PRO needs to initiate and maintain the connection, then select TCP Client and define the IP address and port of the device to which Pathfinder Core PRO should connect.

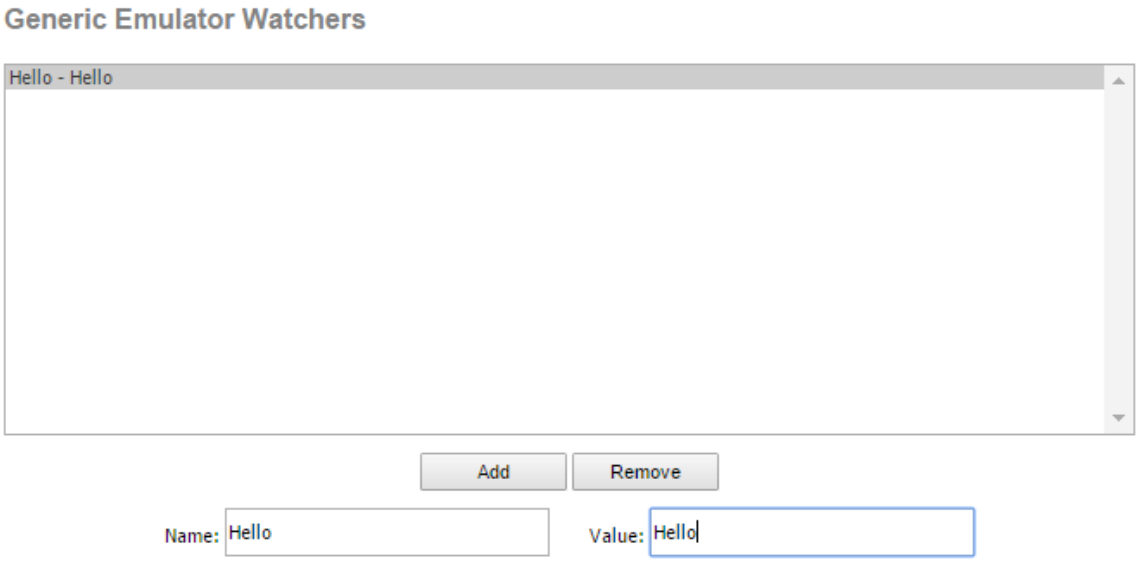
> **Important Note:** Pathfinder Core Pro does not currently support serial ports. If you need serial port control, investigate serial-to-TCP converter solutions, or use Pathfinder's Port Router application to bridge the serial data to TCP data.

After selecting the Emulator Type, the lower part of this dialog will present options that are specific to which emulator type has been selected.

Generic

The Generic device emulator is just a TCP connection. It can be used to send and receive custom messages from an automation system or some other third party system. 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.

This emulator can also be used to trigger actions depending on what data comes into the emulator. In order to define this, we can create watchers. Each watcher is looking for information coming to the emulator port, and if it sees that data, it has a property called Triggered which will briefly get set to True each time the data being watched for arrives. To define these watchers, use the section of the emulator editing dialog that appears when you select the Generic Emulator Type.

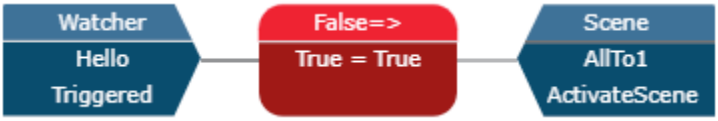


To create a new watcher, click the Add button and select the newly created watcher from the list above. Then, define the name of the watcher and the value being watched for using the name and value fields. You can edit any watcher by clicking it in the list and updating the values in these two fields. The Name of the watcher is used to identify the watcher in the system. It will appear as an object underneath the emulator in the Logic Flows property selection tree.

In the example above, we have created a single watcher on this device emulator called Hello, and it is watching for the value Hello. If we see the word Hello come into this emulator, then the watcher's Triggered property will be set

to True and then back to False so that it can wait for the next time it comes in.

Then we can create a logic flow where the start point is the triggered property of the watch point and do anything we like in the system when we see the word Hello come in through the Device Emulator port.



In this flow we have defined that every time we see the word Hello on the watcher in question, we will activate the scene AllTo1.

Escape Characters

Both the ToSend properties and the watcher values support a number of special character strings to represent special characters.

- \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 word MyName followed by a carriage return and line feed. However, in the following example, the double slashes are converted to a single literal slash for the case where you actually want to send the value \cr\lf.

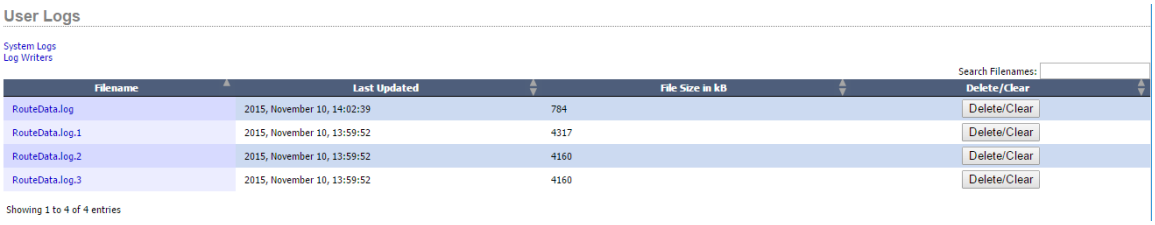
MyName \\cr\\lf

Probel General Router and General Switcher

There are two Probel routing protocols supported by Pathfinder Core PRO. They are Probel General Router and Probel General Switcher. The syntax of these protocols differs and 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. It is highly advised that you create a virtual router with only the sources and destinations from the overall system that you want the emulator to control, and then select that virtual router from this drop down.

Chapter 14 Logs

Pathfinder Core PRO provides a variety of different logging mechanisms as well as maintaining its own internal system logs. Click on the Logs link in the Navigation bar to view the current logs in the system as well as to define logging options.



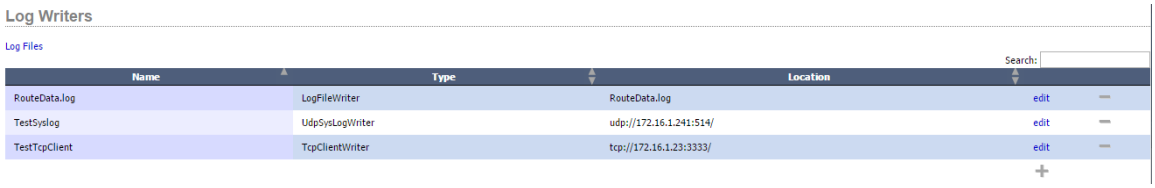
There are two kinds of log files created by the system. System logs and user logs. System logs are created by the system's internal services and primarily define information for use by the developers. You can view these logs by clicking on the System Logs link.

User logs can be created using the Log Writers link which will be discussed shortly. User logs are where you can define changes on your Axia network that you want to be entered into log files for later review.

Clicking on any of the logs will open the log for viewing in the browser. The rows for each log also display the file size and last updated date and time, and finally, a button to delete or clear the log.

You may notice that some logs (such as the one in the screen shot above) have multiple copies with a period and a number at the end. Due to space restrictions, Pathfinder Core PRO maintains an intelligent log rotation service which will rotate the log files if they are becoming too large, getting rid of the oldest information. In general, log files get rotated 4x and the rotation happens when the file size reaches approximately 4MB. 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. This will be outlined in more detail in the LogWriters section next.

In order to define logging parameters, click on the logs link in the navigation bar and then click on the Log Writers link on the page that is displayed. This will present a page listing the log writers that have been created in the system.



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.

First provide a name for the log writer. Next select the type. There are four types of log writer that can be created:

- Log File: This type of log will be stored locally on the Pathfinder Core PRO system. This means that the log is subject to regular rotation, deletion, and cleanup depending on space requirements. Therefore, it should not be relied upon for long term storage. It is very useful for recent events though.
- TCP Listener: This type of log writer will listen on a TCP port and 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 system and monitor log changes dynamically.
- TCP Client: This type of log writer will attempt to connect to an 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, storage used by many systems and IT scenarios. A number of syslog collection applications exist, both freeware and commercial. A quick internet search should provide you with a variety of options.

Once you choose the log writer type, you may be presented with a number of additional pieces of information to fill in regarding the connection specifics. These can include IP addresses and TCP port numbers. The required options will appear according to which log writer type you select.

Finally, you need to use the selection window at the bottom of this dialog to select what log options you wish to log. Expanding the tree in this window will present checkboxes for 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)
  - » 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.

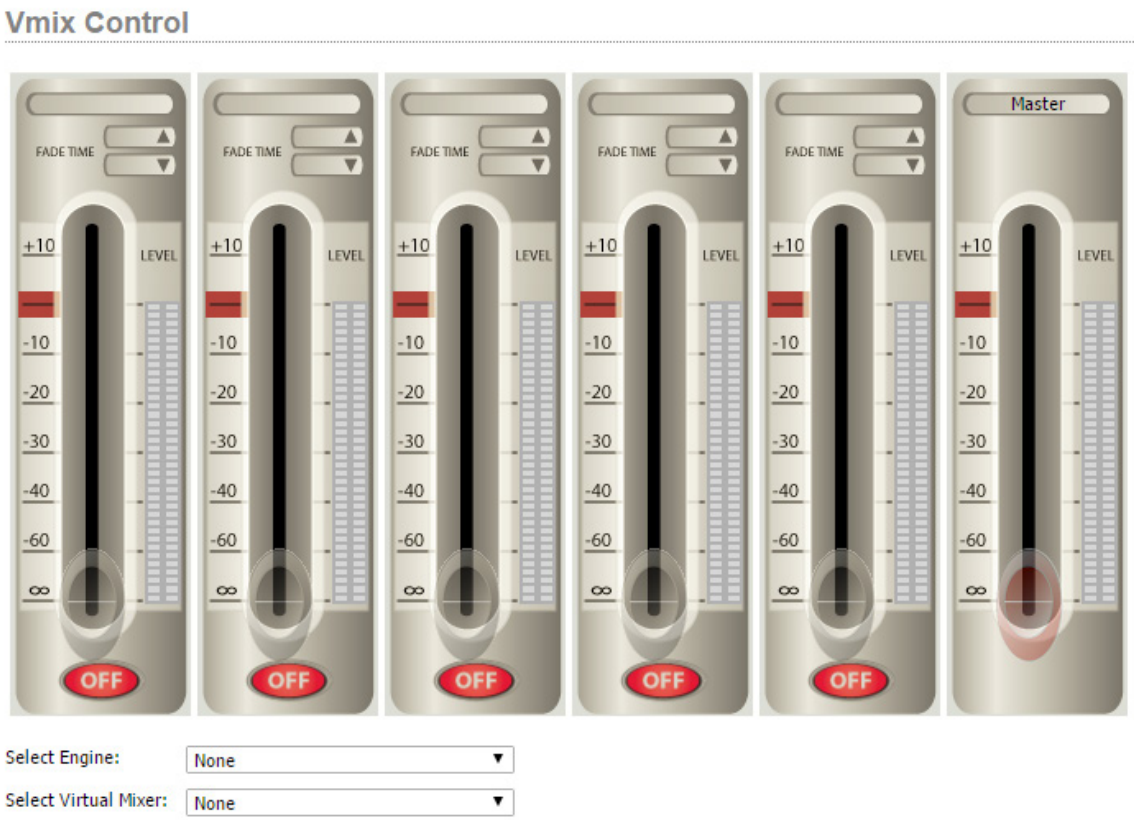
- RouterEvents
    - » GPIO State: Logs changes to GPIOs within the system.
    - » RouteState: Logs any route changes that take place within the system.
- TimeEvents
    - » Elapsed: Logs whenever timers elapse.
    - » Enabled: Logs changes to a timer’s enabled state.

Please note that this list is subject to change in future software revisions.

Chapter 15

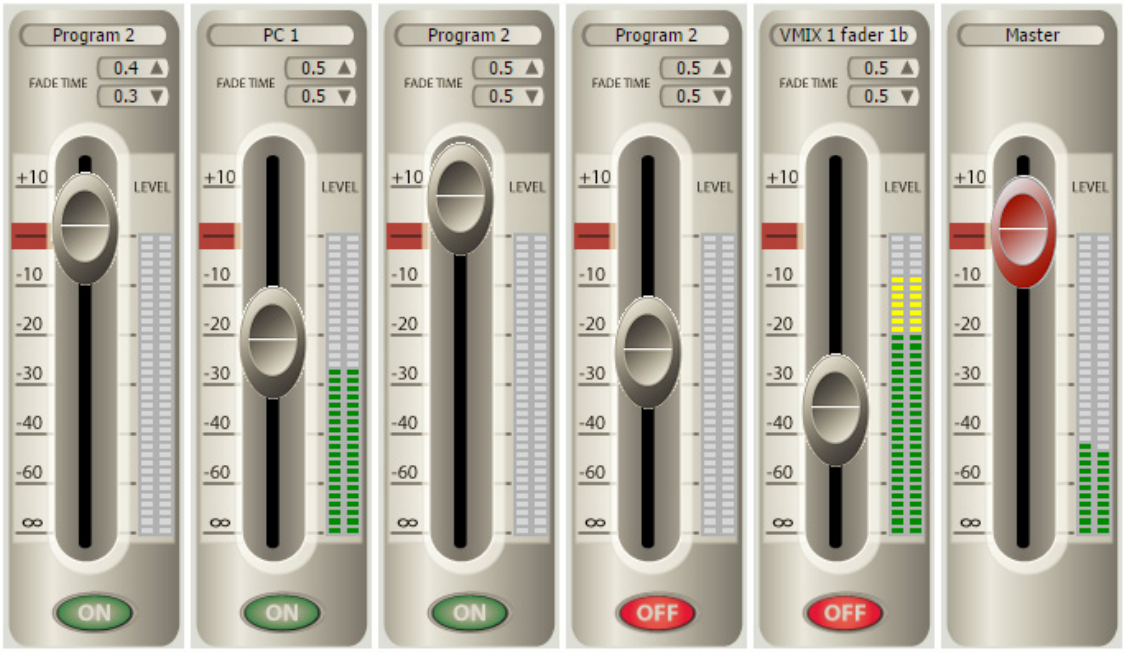
Vmix Control

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 mixer, in effect creating an on-screen mixer. To access this user interface, click on the Vmix Control link on the navigation bar.



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 tell the page 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

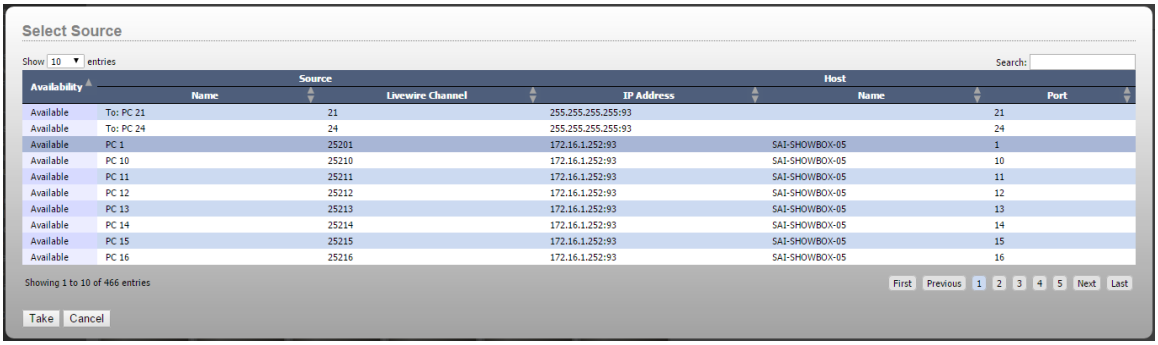
The interface will now update to display the state of the virtual mixer selected. You can use the drop-down lists to switch between different Vmixers.

Each Vmixer has 5 input channels and a master fader. The master fader is labeled as master and has the red fader. The meters for the input channels display the real-time metering of the source assigned to each of the faders. This metering is pre-fader so it is showing you the level of incoming audio. The meter next to the master fader is displaying the level at the output of the virtual mixer and is therefore post Master fader.

Dragging a fader on the screen up or down will adjust the VMixer channel's gain.

Clicking the On/Off buttons will turn a fader on and off. It is important to note that turning a virtual mixer fader on and 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.



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.

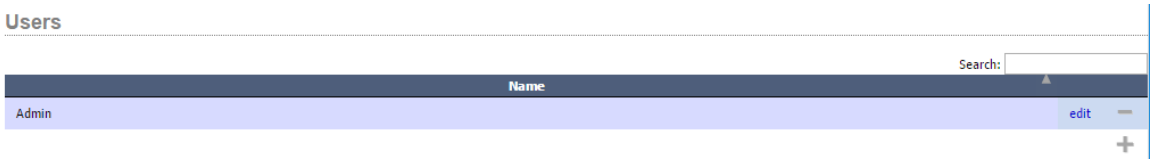
Chapter 16

Users

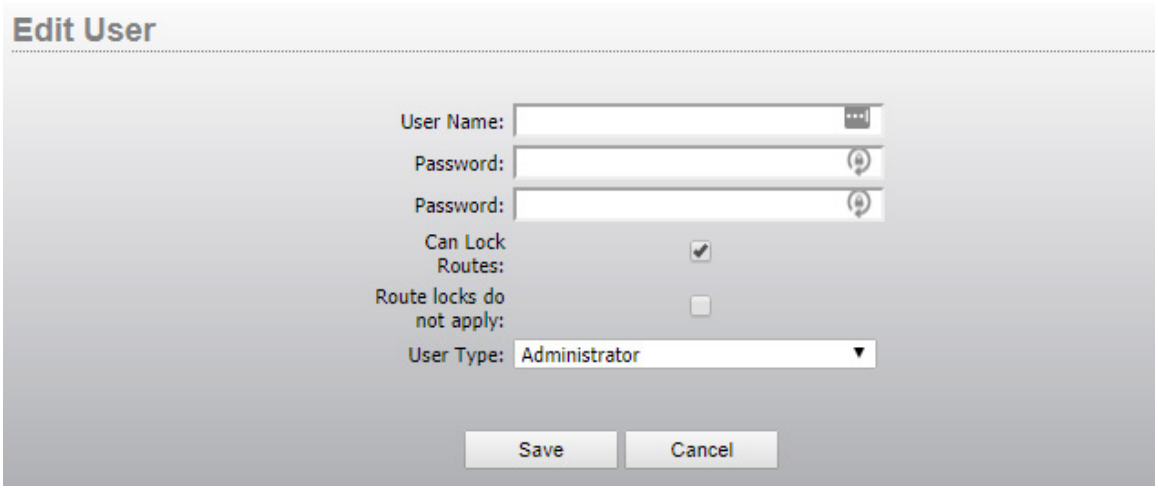
When you first log in to Pathfinder Core PRO, the username and password you use is:

Username: Admin
Password: Admin

However, this can be changed and controlled using the Users link on the navigation bar.

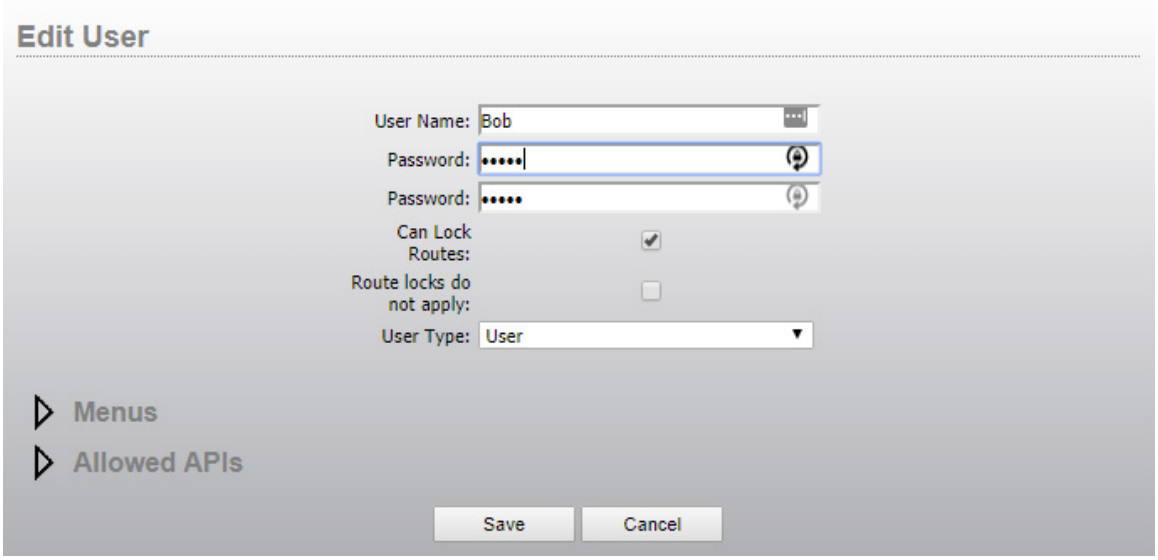


Click edit to change a user's password or click the plus icon to add users to the system. Use the minus icon to delete users from the system. The system will not allow you to delete all users. At least one user must always exist in the system or there is no way to use it. Therefore, the user interface will show an error message if there is only one user left and you try to delete it.



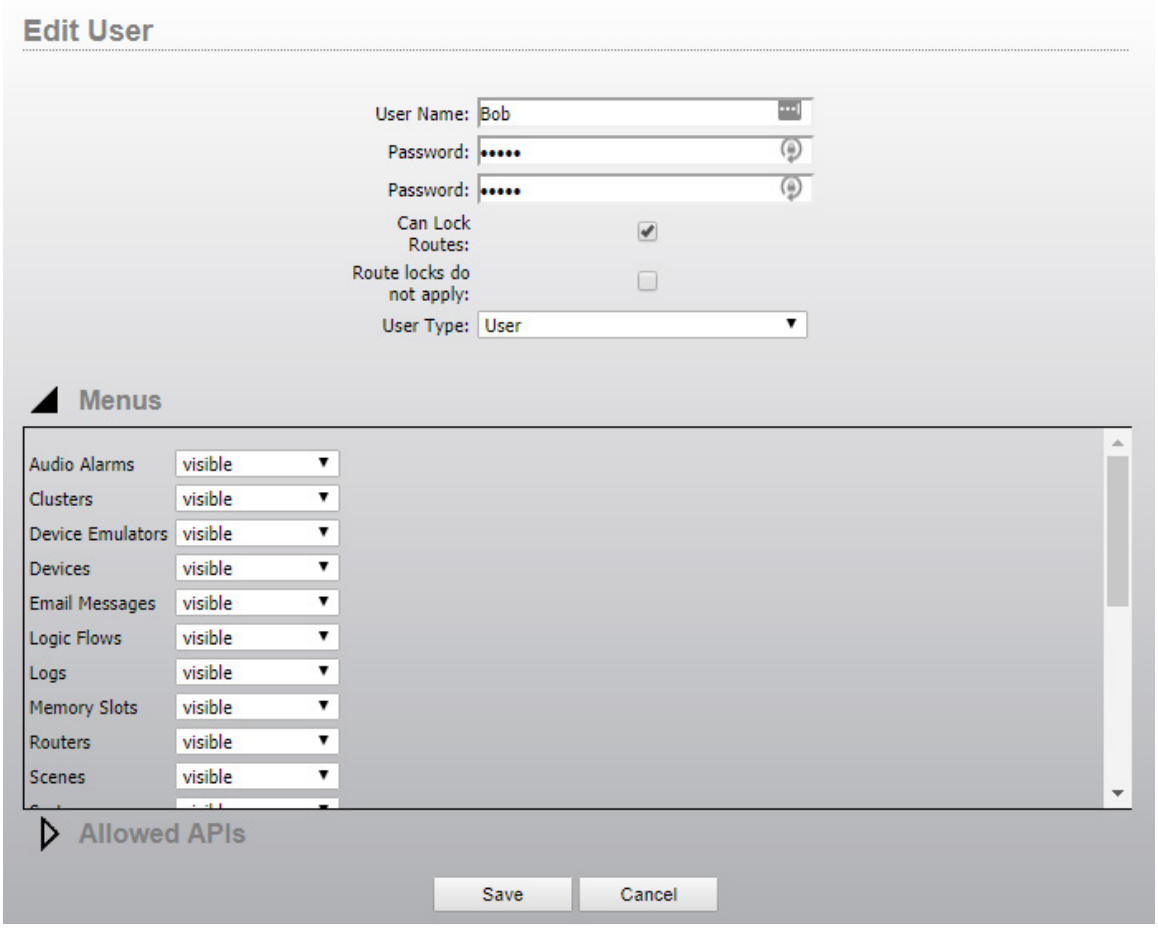
After creating a new user, that user's credentials may be used to log into the Pathfinder Core PRO's web pages or using PathfinderClient, PathfinderMini, or PanelDesigner.

Users may be Administrative users who have acces 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.



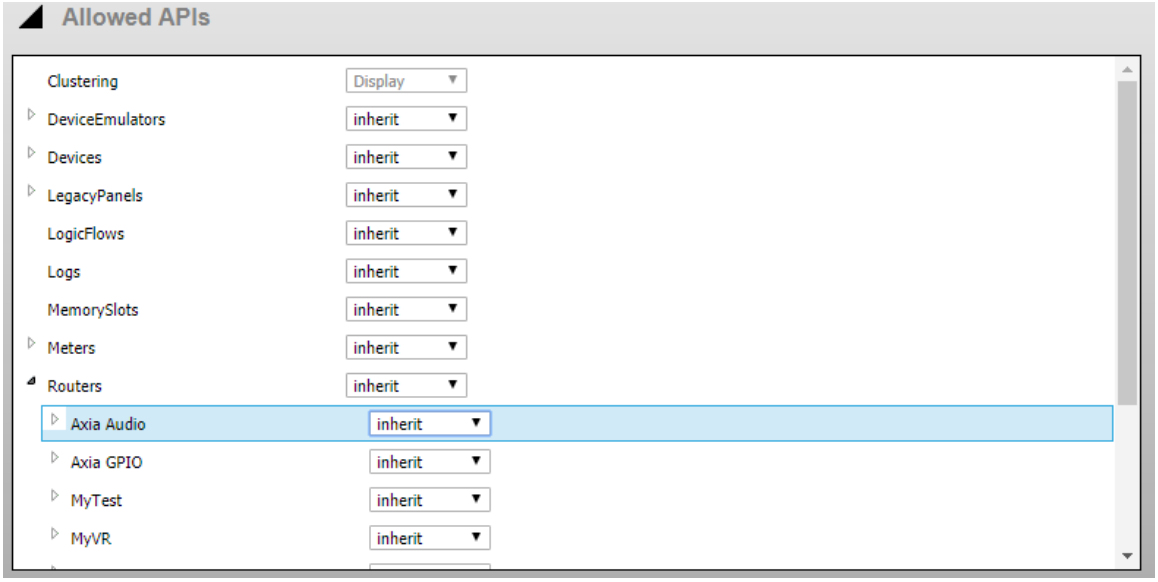
The arrows next to each section may be expanded to specify the user rights. By default a standard user will see all navigation menu items but will have no rights in the api so most or all of the menu items will not populate with any data.

The menus section allows you to define which navigation bar menu items are available to the user.



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:



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.

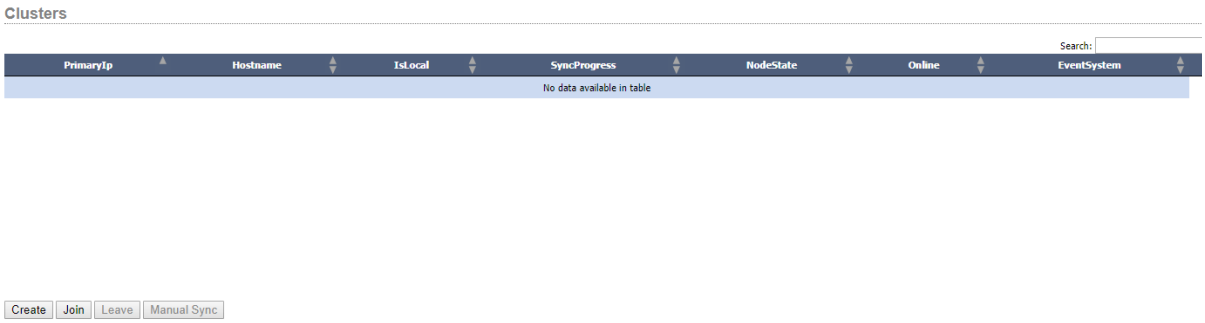
It is important to note that by default branches below a specific branch inherit the parent's properties. So, if you set the Routers branch to display, all routers will be displayed. You can then set no access on routers you do not want the user to see and/or Change/Use access to routers you want them to be able to use.

Chapter 17

Clusters

If you are concerned about hardware failures, then Pathfinder Core PRO clustering is for you. Pathfinder Core PRO allows you to add multiple Pathfinder Core PRO systems onto the network and connect them together so that they automatically and dynamically share configuration data and states. They also monitor each other, taking over processing events if one of the systems fails.

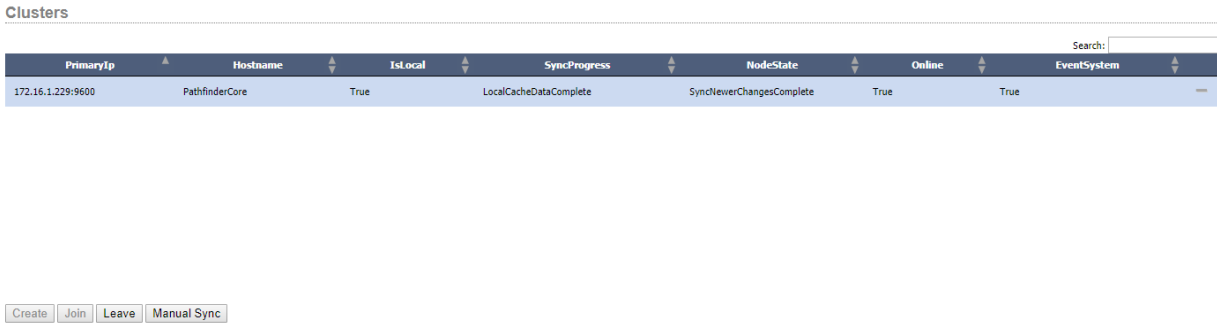
The first step is getting your system to work close to the way you want it to function with a single Pathfinder Core PRO. Allow the routers to be created and add other events and components you wish to have running on your system. Next, take a backup of your configuration and download it to your local 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. Next, click on the clusters link in the navigation bar.



Four buttons will be presented at the bottom of this page; Create, Join, Leave, and Manual Sync. The Leave and Manual Sync buttons are unavailable because the system is not currently a 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:



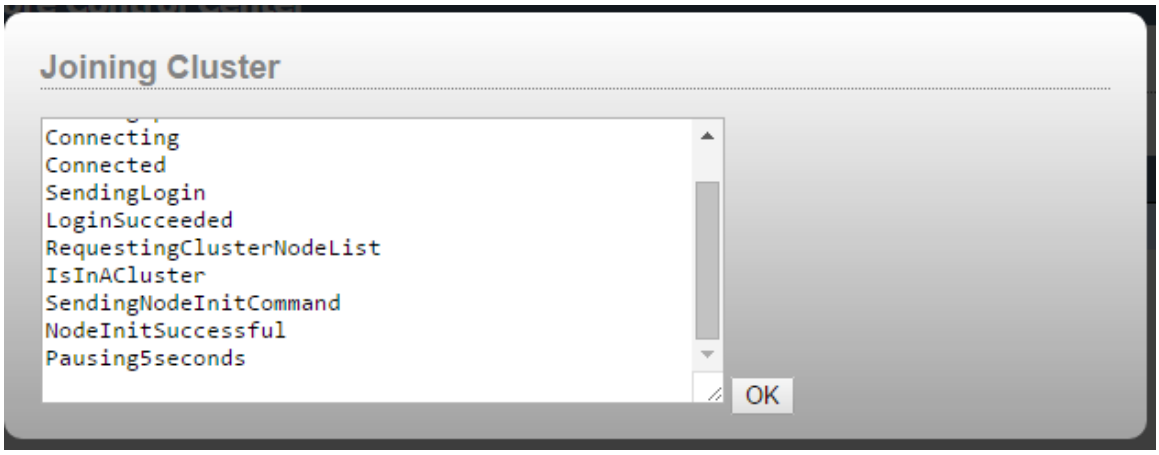
The system is now part of a one node cluster. This display will show the hostname and IP address for each node in the cluster. When there are more than one systems in the cluster, the IsLocal field will show which of the systems you are currently logged into. It will also display the state of each Pathfinder 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

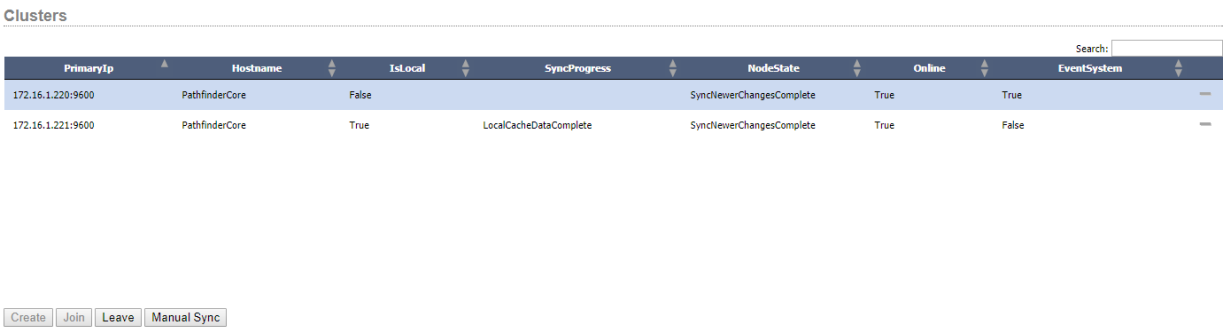
Now it's time to add the second system into the cluster. It is important to note that the second system's configuration will be overwritten during this process. If the system is new, make sure you first assign an IP address using the front panel. The IP must have network access to the first unit in the 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 will also ask for the cluster administration password you provided when you created the cluster.

The system will now pop up a progress message box as the system joins the cluster.



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.



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 procedure that would be performed in the context of an Axia support session.

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.

Chapter 18

PathfinderPC_Core Client

PathfinderPC_Core Client is a desktop application that users can use to make routing changes with the Pathfinder Core PRO system. It is important to note that as we move more and more capability directly into the browser this application is becoming deprecated. Web browsers are the preferred client for interacting with Pathfinder Core PRO.

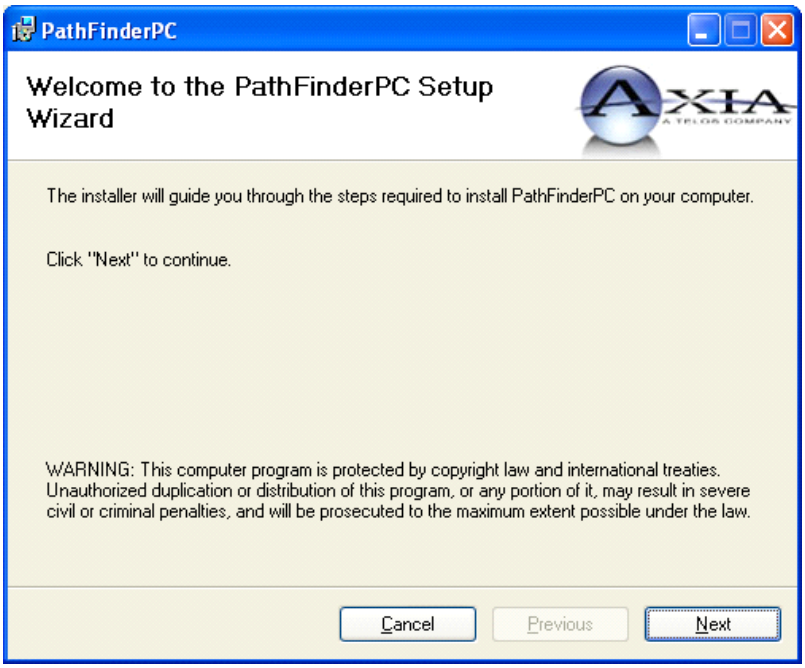
The PathfinderPC_Core Client communicates with Pathfinder Core PRO over TCP/IP on the local LAN and/or over the Internet. It can be used to make routes, create and edit virtual routers, and to create, edit, and activate scene changes. It also has a search engine for finding route points and scenes. Finally, it can display and manipulate user panels that have been created in the system.

PathfinderPC_Core Installation

The installer for PathfinderPC_Core Client may be downloaded from the Pathfinder Core PRO System web page. The client software may be installed on any Windows machine on the network that is to be used for making routing changes.

To Install PathfinderPC_Core Client:

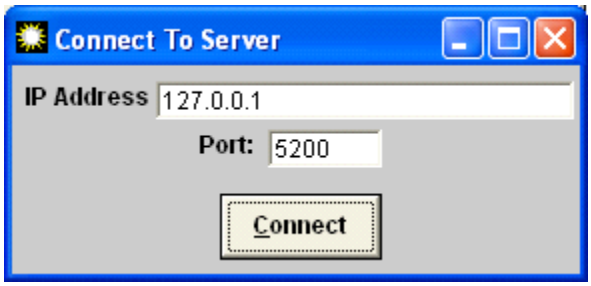
- Click on the System web page in PathfinderCore PRO.
- Click on the PathfinderPC Core Client link and download the PathfinderPC_Core Client installer.
- Double Click on the PathfinderPC_Core client installer to install it.



- Click **Next** to begin the installation.
- Click **I Agree** to the license.
- Click **Next** again.
- Select the location for installing the application or leave it at the default.
- Click **Next** a final time to finish the installation procedure.
- If the application is being installed on Windows 7 or later, it is recommended that the following option also be turned on to ensure PathfinderPC_Core Client has the ability to properly manage its own files:
    - » Find the PathfinderPC_Core.exe application located in the C:\Program Files(X86)\PathfinderPC_Core folder.
    - » Right click on the PathfinderPC_Core.exe application and select properties.
    - » Click on the compatibility tab.
    - » Enable the Run As Administrator checkbox.

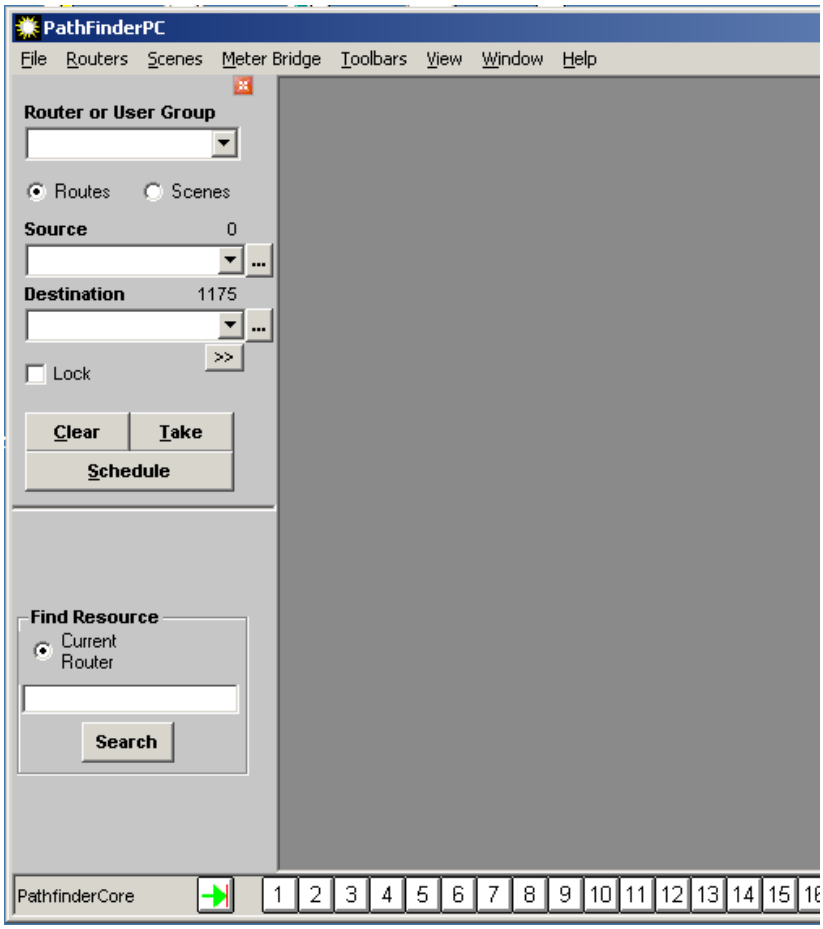
PathfinderPC_Core Main Application

After installing PathfinderPC_Core Client, launch the PathfinderPC_Core Client application. After a few seconds, one or two messages will appear that there is an error connecting to the server. Click OK on these and then select the "Connect To Server" menu item under the File menu. This will present the following window.

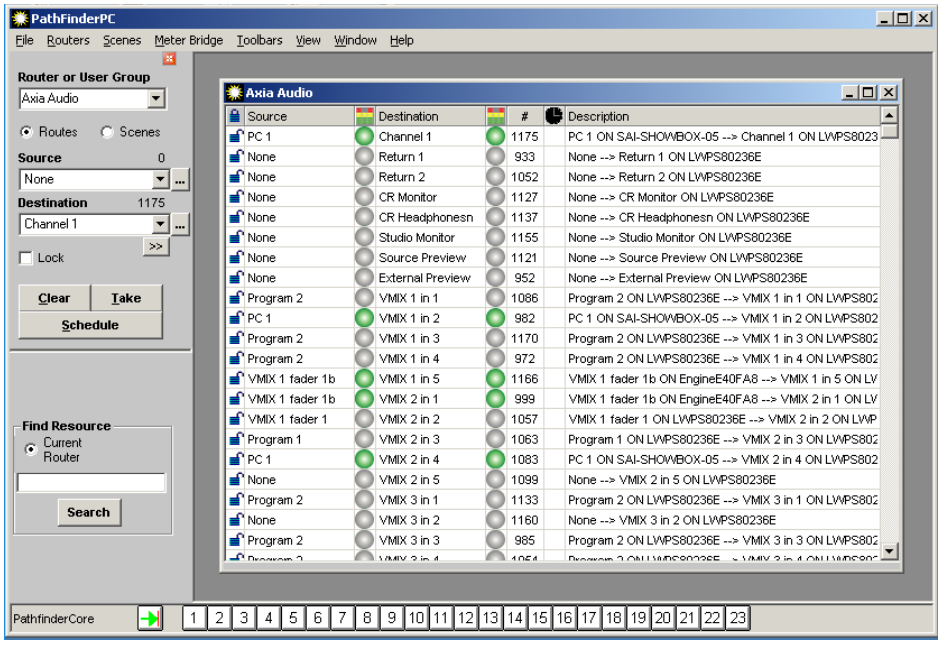


Enter the IP address or fully qualified DNS name for the Pathfinder Core PRO system. For use with Pathfinder Core PRO, the port should always be left at 5200.

If the connection is successful, all of the menu items on the main Patch Bay Control screen will become available.



Under the Routers drop down list will be a list of all of the available Routers defined in Pathfinder Core PRO. Selecting any of these routers will bring up the appropriate routing list for that router.

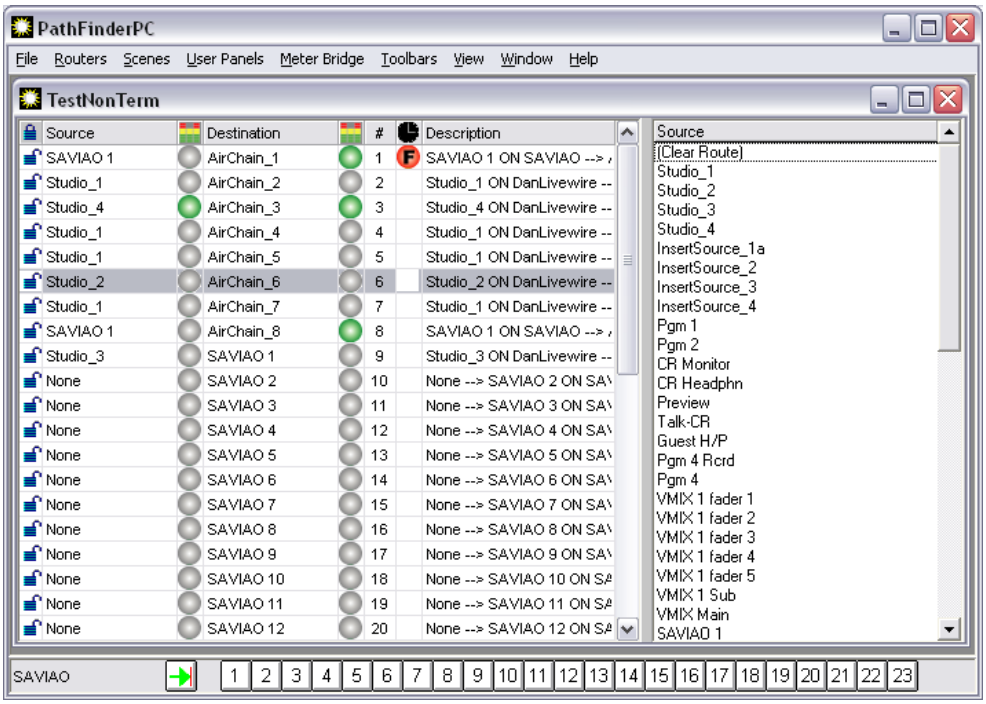


PathfinderPC_Core has two control panel toolbars. Along the bottom is the window preset toolbar. Each numerical button holds a window configuration. Since the user can have multiple router and meter windows open simultaneously, this panel can be used to store these window positions and sizes for quick retrieval. Clicking on one of the numerical buttons retrieves the window position and opens the correct windows, positioning them correctly. In order to store a set of window positions, click the green arrow button and then the preset number into which you wish to store the current window setup.

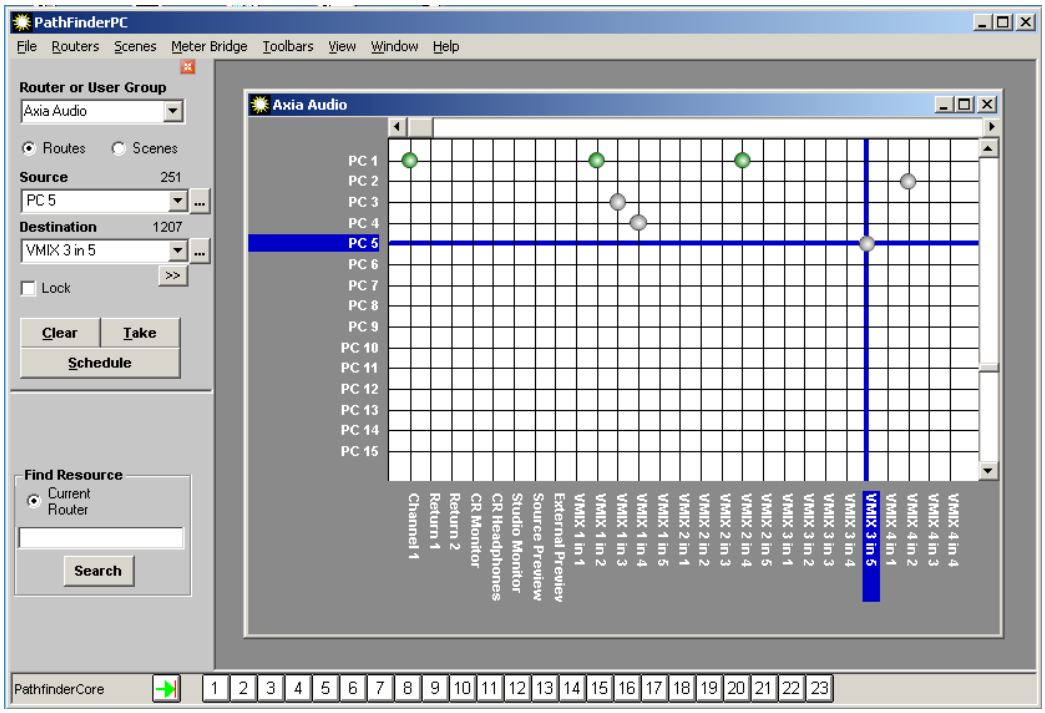
The route control toolbar is on the left side of the PathfinderPC_Core screen. This provides the primary routing control over the system. Routes may be made by selecting router and the destination, and then selecting the source you wish to be assigned to the destination. Finally, click Take or clear to establish the route or clear the destination. All source and destination drop down lists in the software can also display the description column by clicking on the header row (top fixed row) of the drop down list. Route points may also be locked or unlocked using this panel if the router supports locking. Locked route points may not be taken by another user without specific confirmation. Clicking on the padlock next to the line in the routing grid will also lock and unlock the point. When the padlock is closed the point may not be changed unless it is unlocked first. Clicking on a locked padlock to unlock the point will produce a warning asking if you really wish to unlock the route point. There are some destinations that are locked at the system level of the device and therefore cannot be unlocked or changed by PathfinderPC_Core. These include some Engine destinations that are involved in console operations. While PathfinderPC_Core will display the route status of these Engine Destinations, it is not allowed to make changes to their routing. That must be accomplished through the standard Console and Engine interfaces, or through console control logic flows which allow you to load different source or show profiles.

Selecting the scene option changes the control panel to show a drop down list of scene changes which are available for the chosen router. Any of these scenes may be selected and taken.

In addition, there are two other routing interfaces that may be selected depending on the user's preference. For users that prefer the legacy mode of routing, the route control toolbar may be closed. When this toolbar is closed, simply clicking on the line for the appropriate destination and then clicking on the correct source in the pop-up list which will appear on the right makes the route. Hitting escape closes the pop-up list without making a change.



Finally, the user can select the Chart View under the view menu to produce a graphical grid view of the routing status.



Selecting List under the view menu will return the routing status to the original mode.

Multiple routers may be open and arranged in the master parent window as desired.

The window menu allows the user to bring a particular window to the forefront or to arrange the windows automatically within the parent window.

There are several columns in the routing grid. The lock icon shows whether the point is locked or unlocked. Next is the name of the source name involved in the routing point. If the router is a Livewire Audio router, there will be an audio icon which shows whether audio is present on the source. Next is the Destination name in the route point. Again if the router is a Livewire Audio router, there will be an audio icon which shows whether audio is present on the destination.

A green dot shows that audio is present. A red dot shows clipping, and a grey dot shows audio absent. If no dot is present it signifies that the route point does not support audio metering.

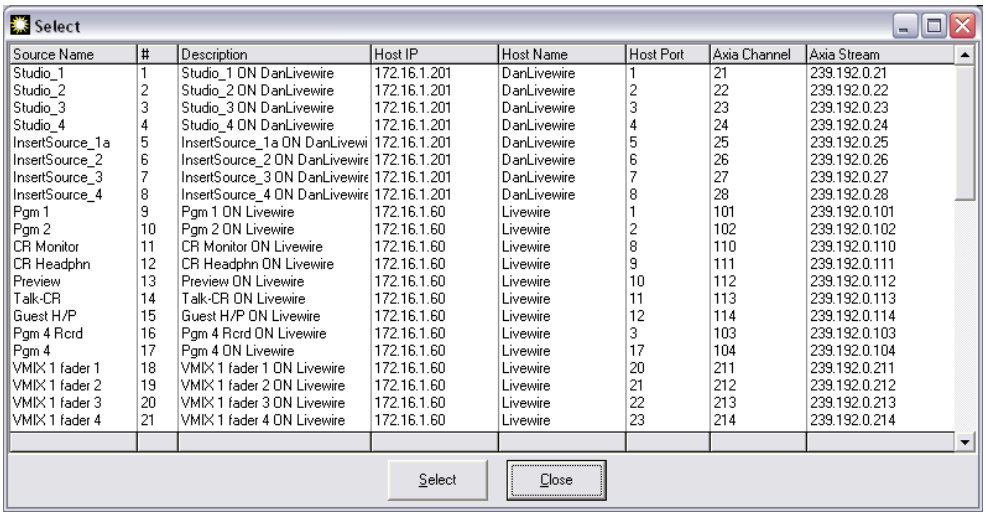
Next, there will be a column which shows the destination's ordinal number in the router. The eye column shows the status of any silence detection events. This column is not used by Pathfinder Core PRO. But since the application also works with PathfinderPro, the column is still present. A blank field means that the route point is not involved in any silence detection. A "P" means that the route is connected to the "Primary" source. A "B" means that the route is connected to the "Backup" source. An "F" means that the point has "Failed." This could either be due to audio failure on both primary and backup sources, or the current route is not set to either Primary or Backup. In addition, these indicators will change from colored to grey to indicate the status has entered the pending timeout period and is waiting the required timeout period before making a switch.

Finally, there is a column which shows the description field. This is a combination of the description information from both source and destination points.

Select List

Throughout the PathfinderPC_Core Client software, there will be drop down lists when sources and destinations need to be selected for events, stack events, route selections, etc. Next to these drop down lists will also be a button identified by an ellipsis.

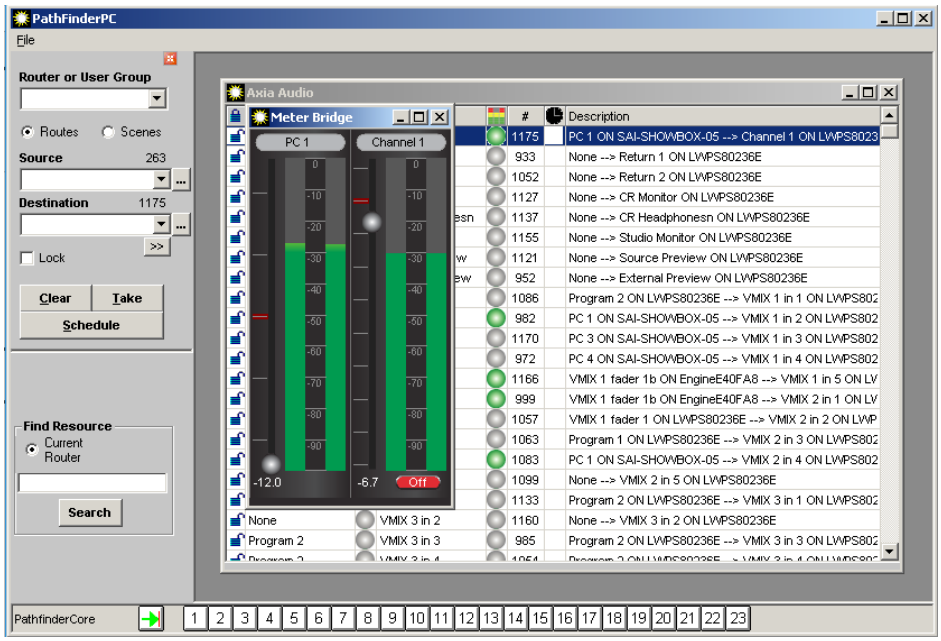
Clicking the ellipsis button will open a select window from which the source or destination may be selected as an alternative to finding the point in the drop down list. This window includes a grid with additional information about the sources or destinations including Name, Pathfinder Number, Description, Host IP, Host Name, Axia Channel Number, etc. The grid may be sorted by any of the columns and the columns may be resized and reordered.



Click the header bar with the column name for any column to sort the list by that column in ascending order. Click the footer bar for the particular column to sort the list by that column in descending order. Dragging a column header will allow you to reorder the columns. Press the Close button to close the window. Select the source or destination in the list that you want applied to the drop down list in the previous window. Then click “select” or double click the entry to select that entry and cause the select list to disappear. This process may be alternatively used to select sources and destinations in the client wherever a drop down list is presented.

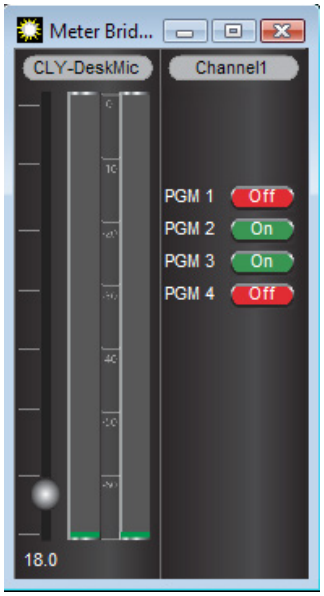
Metering

Right clicking on route points which support audio metering as indicated by the audio columns in the grid will open a fully functional set of meters for that route point.

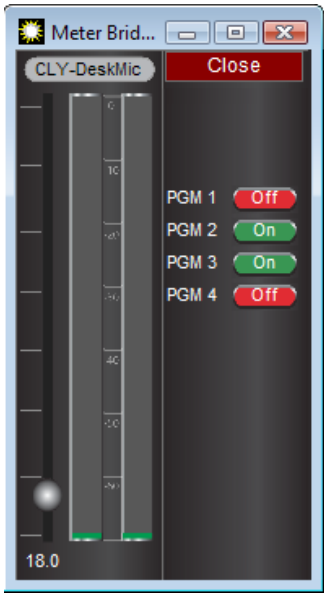


The meter bridge window may be enlarged and meters added, removed, and reorganized. Right clicking on the grid will add additional meters to the meter bridge.

The meters may also have some additional controls depending on the type of source or destination they are controlling. If the gain can be changed on the source or destination, a circular fader will be present next to the meter which will allow these changes. If the IO is a node IO, this will change the gain on the node. If it is a Vmix, it will change the Vmix fader gain. If it is a Console Fader Channel, it will change the actual level of the fader. In addition, Vmix and Console Channels will present an On/Off button for turning the channel on or off. Finally, if the destination is a Console Fader Channel, right clicking in the middle of the meter will cause the meter to show program buss assignments which can also be changed.



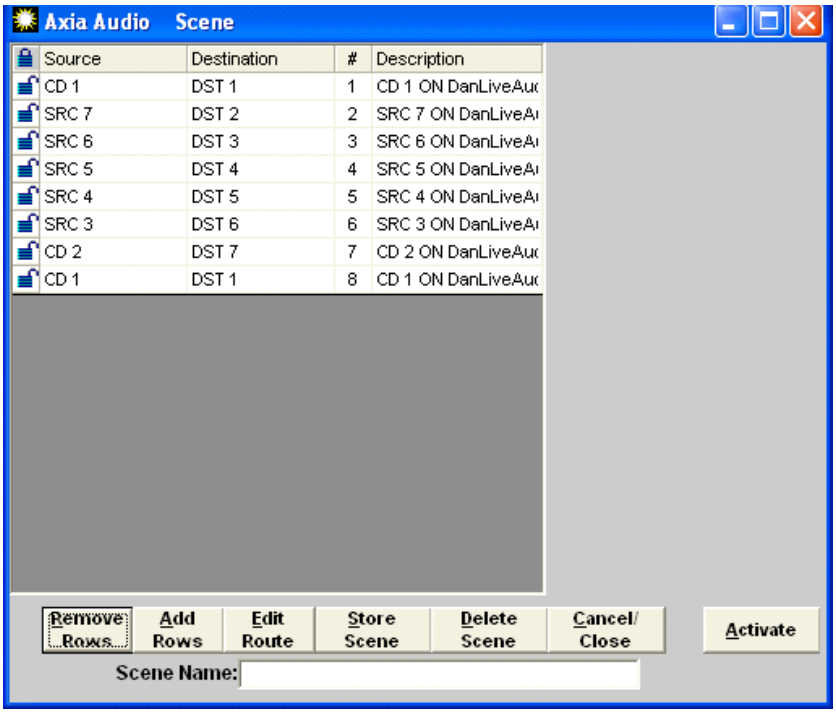
Right clicking again will return to the normal meter state. Meters can also be closed by positioning the mouse to the top of the meter. This will cause a close box to float down. Clicking that close box will close the meter.



These meters provide quick and easy monitoring of levels in your Axia system.

Scenes

The items within the scene menu and scene drop down list change automatically depending on which router is currently selected. These are a list of scenes stored on Pathfinder Core PRO for that particular router. A new scene may be captured and edited using the Edit and Capture Scene menu items.



At this point, the scene contains the current state of all rows in the router. Routes in the scene may be edited without changing the master router by clicking the edit route button or double clicking the line to change the source. Rows may be removed from the scene to create a partial scene. In this way, a scene may be created which just changes a few points to a particular state rather than changing the entire router. Removed rows may be added back using the Add Rows button. The Store Scene button stores this scene into the Pathfinder Core PRO database for the particular router. A valid name must be entered into the Scene Name field before it can be stored. The scene is then added to the list under the scene menu on the main screen when that particular router is selected. The scene may be activated, causing the routes to be taken by selecting that scene from the list under the scene menu item. Alternately, you can use the drop down scene list of the route control panel.

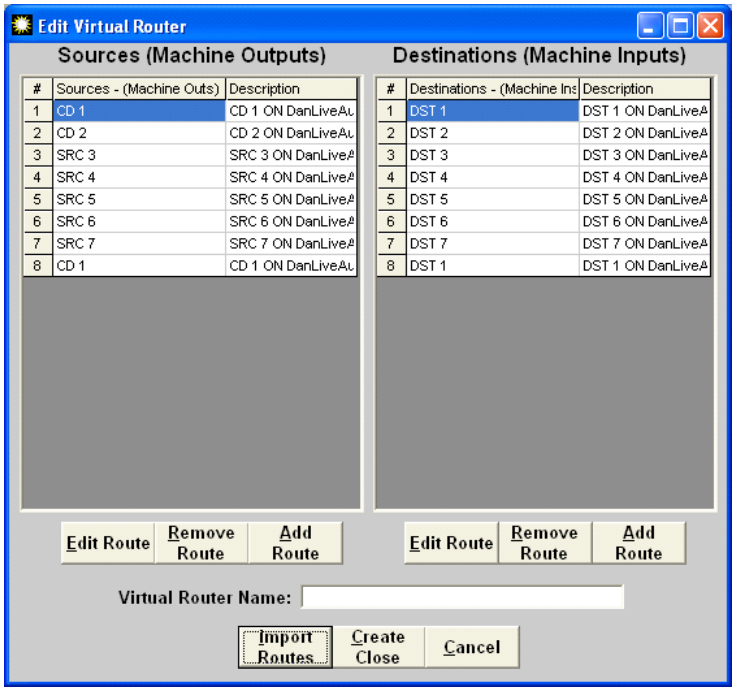
Please note: The scenes also store the state of the Lock point if locks are supported. Thus, a scene that makes routes and then locks them may be created. Also, if a scene is selected that uses points that are already locked, the scene will not complete the locked routes and will issue a warning.

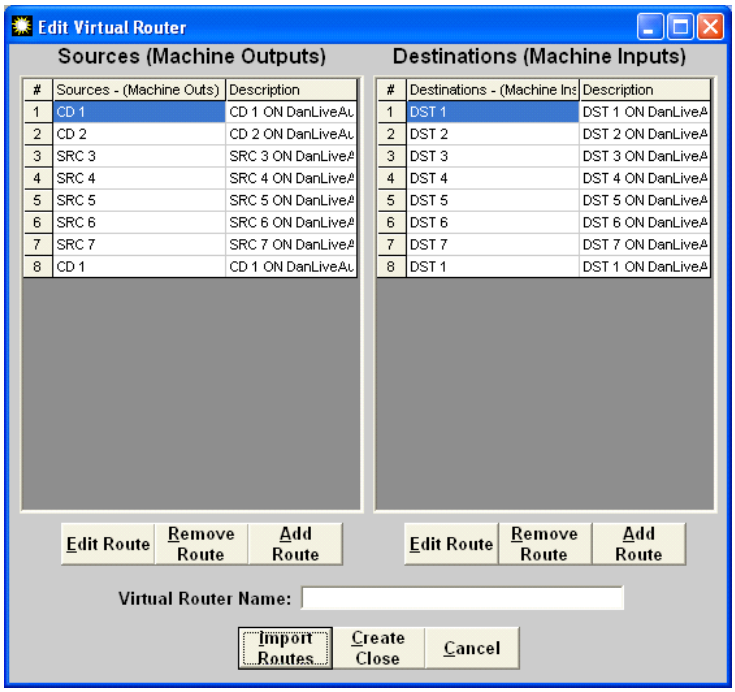
Virtual Routers

Virtual Routers are Routers that have points that correspond to points on other actual routers. For example, if a facility has a centralized Audio router with a variety of shared gear, as well as points from a number of edit suites, each individual suite may only want to see the routing points that are pertinent to that suite. Rather than seeing a huge list of points that are rarely used, a Virtual Router may be created which only contains the routing points on the main router that are most often used. The full router may still be opened when needed, but the virtual router would be used for everyday use. This also allows an Administrator to limit a user to a specific set of route points.

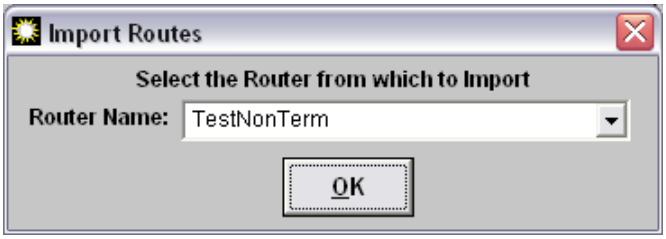
Each point on a virtual router may also correspond to multiple points on different actual routers. In this way, a machine router may be created where Audio, Video, and Machine Control can be routed with one click. To accomplish this, a point is created on a virtual router for a machine (i.e. Digital Beta Video deck). This point contains the correct routing point information for the actual audio router, video router, and machine control routers.

In the PathFinderPC_Core Client Software, Virtual Routers may be created by using the Create Virtual Router and Edit Virtual Router menu items under the Routers Menu. This will also display the Edit Virtual Router Window. Once a Virtual Router has been created it may only be deleted from the Routers link on the Pathfinder Core PRO web page.

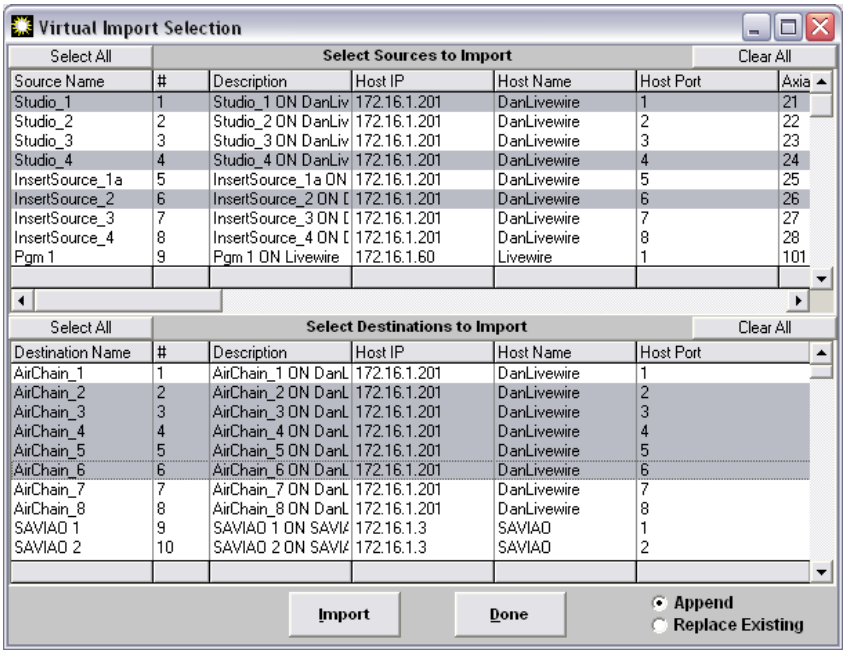




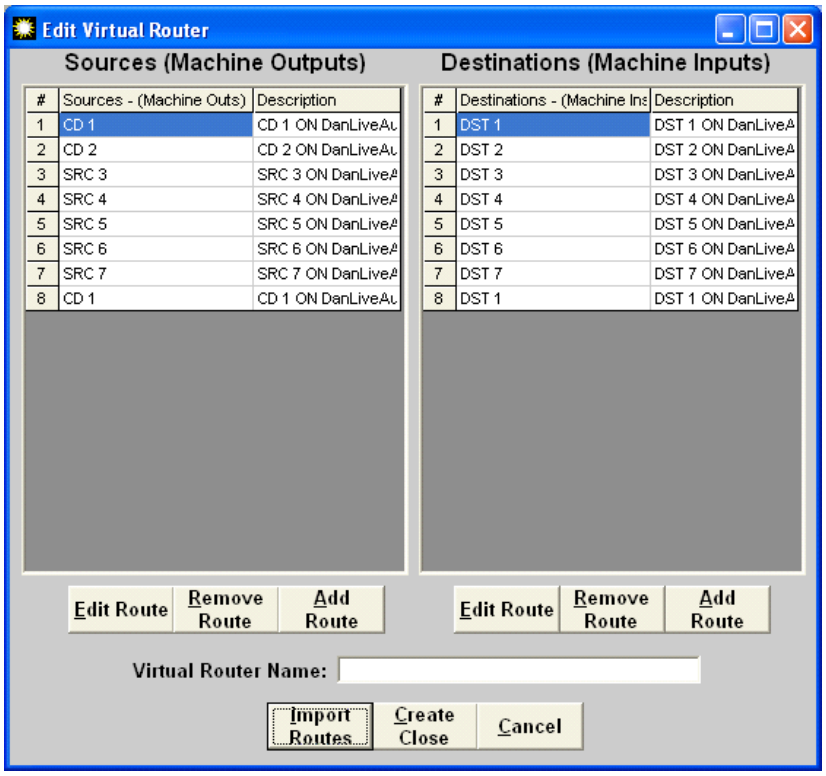
The Virtual Router Edit Window allows route points from other routers to be added, removed, edited and imported. The Import Routes buttons provides a list of the current routers on Pathfinder Core PRO, and allows the user to import any or all of the routes from a given router into the virtual router. The routes can either be appended to the Virtual Router or replace the existing routes on the router.



After selecting OK, a selection list will appear.



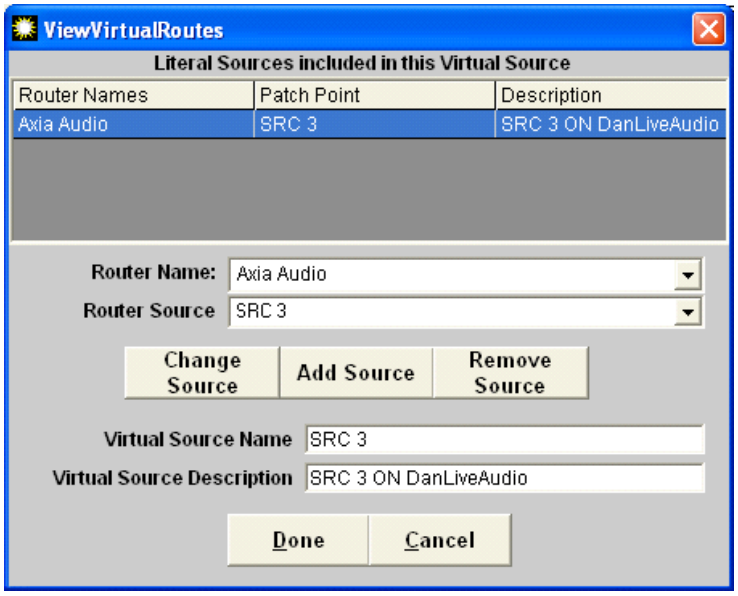
Use this list to select the routes you wish to import. Use the Ctrl and Shift keys to highlight multiple rows in the source and destination lists. Click Import to import the route points into the virtual router. Records may be sorted by any column by clicking the column header.



The Remove Route button removes a routing point from the router. A good way to create a Virtual subset of an existing router is to import all of the routes from the router into the new virtual router and then remove the points that are not desired. Multiple sequential routes may be removed by clicking the top point to be removed, then holding the shift key while clicking the bottom route. Once the routes to be removed are selected, click “Remove Routes”.

Right clicking on a point and dragging it to the desired position reorders the List.

The Add Route and Edit Route buttons display the View Virtual route window.



The View Virtual Routes window is where the specifics for a given virtual route may be edited and defined. Each Virtual point is made up of any number of actual points from other routers. These points are listed in the list at the top of the screen

A given route point may only have a single route point making it a direct map to a point on another router. But it also may have several router points tied to this single virtual point. This allows a single virtual point to include audio, video, machine control, and GPIO points for a given device. The Remove Route button removes the selected real route point from the virtual route point list. The Router Name and Router Point drop down lists provide access to all of the actual routers and their points. Simply select the router and the route point which is to correspond to this Virtual point and then click either Add Source/Destination or Change Source/Destination to either add a new point or edit the currently selected point.

The Patch Name is the name of the Virtual Point in the Virtual Router, and the description field allows a description to be created for the virtual route point. Clicking “Done” submits the Changes to the configuration.

The purpose of a Virtual Router is to provide an ordered subset of a real router where each point on the virtual router corresponds to an actual point on a real router. Some more explanation may be needed to understand why some points in the Virtual Router might have multiple corresponding points though. Here is an example.

Radio station WXYZ is using Pathfinder Core PRO, and has already allowed discovery and therefore has both an audio and a GPIO router. One of their automation systems has both a primary and a backup system - each with a Livewire driver. While interfacing with a satellite link, the system needs to be able to react to and trigger GPIO changes from a node on the network as well as audio. It does this by routing the node's closures to the automation system's Axia driver which the automation system is monitoring. And when the automation trips closures on its local drivers, those need to appear as GPOs on the remote node. Additionally, audio needs to be routed to and from the automation system. When the station needs to switch from the primary automation system to the backup system, all of these signals must travel together.

In order to accomplish this, a Virtual Router may be created. Create a new point in the Destination field (the device input side) by clicking on the Add Route button under the Destinations list. Assign it a patch name of "Automation System A". Next, select the Audio router from the router name drop down combo. Select the correct audio point for the first automation system's input from the router Point drop down combo. Click Add Destination. Next, Select the GPIO Router from the router name drop down combo. Select the correct GPIO input point for the Automation system from the router point list that is now displaying the list of routes for the GPIO router since that is what is selected above. Click Add Destination. Finally, click the "Done" button to add the point into the configuration.

Repeat the same procedure under the Sources List (Device outputs) and make a point for the audio source - for the sake of this example, we will use a satellite audio signal applied to a node. Also add the GPIO point as described above. You could also create an output point for the automation system and an input point for the satellite, so that routing could take place in both directions. Repeat this procedure for the second automation system. Finally, once all desired points are created, click the Create/Close button to submit the new router design to the server or Pathfinder Core PRO. At this point, the Virtual router will then appear in the list of Routers in the Routers Menu. Open the new virtual router.

Now if the engineers route Satellite Out to Automation System 1 In, the Audio and GPIO are routed simultaneously.

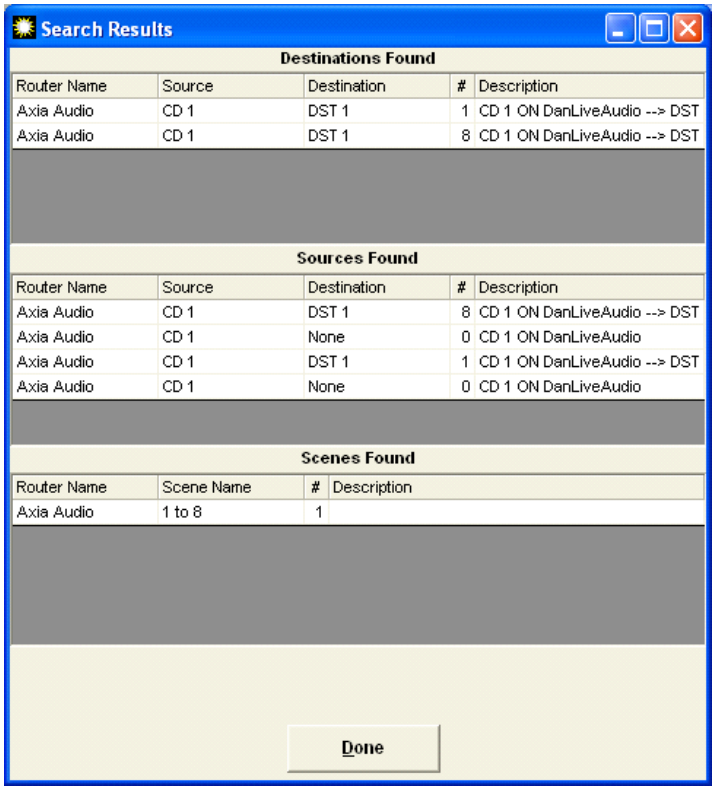
Finally, a scene could be created which makes two routes on the virtual router to route both the output of the Satellite to the Input of the Automation System and the output of the Automation System to the input of the Satellite with one mouse click, and all of the audio and GPIO would follow.

We could also extend this example to tie multiple audio points into a single virtual source or destination to simultaneously route multi-channel audio signals.

In this way, the virtual routers may be used to create any routing scenario imaginable among the various routers.

Search Functions

With large routing systems, it will often be necessary to search for route points. This can be accomplished either through the search part of the routing toolbar, or by selecting Search from the File Menu. Enter the search phrase into the text. The software will then open the search results window.

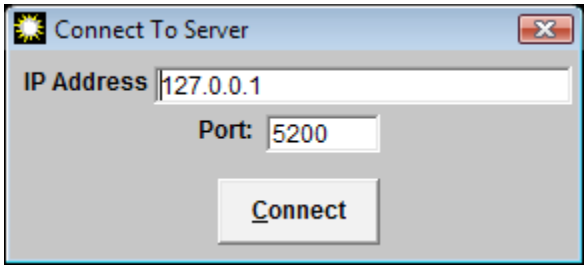


The search results show the sources, destinations, and scenes which match the search criteria. Selecting a particular source or destination point and clicking done will highlight that point on the routing grid.

File Menu

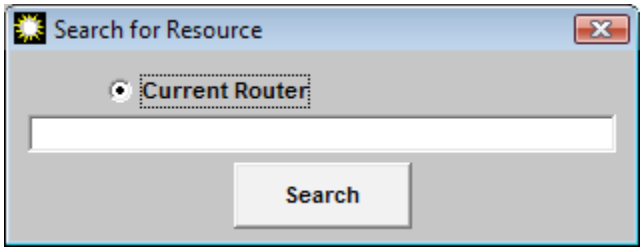
The File Menu provides access to a number of master functions.

The **Connect To Server** menu item opens a window where you can enter the IP address of the Pathfinder Core PRO to which the client should connect.



The **Clear Routes On Shutdown** is a seldom-used option. If this item is selected, any routes that are made by that instance of PathfinderPC_Core Client will be cleared when the client is shut down. This can be useful in a recording studio environment where you wish to clear the system at the end of the day, but is not recommended for radio station environments. This option should generally be left off.

The **Search** menu item opens a search window similar to the one in the Route Control toolbar. See the search section above for details on the PathfinderPC_Core Search functions.



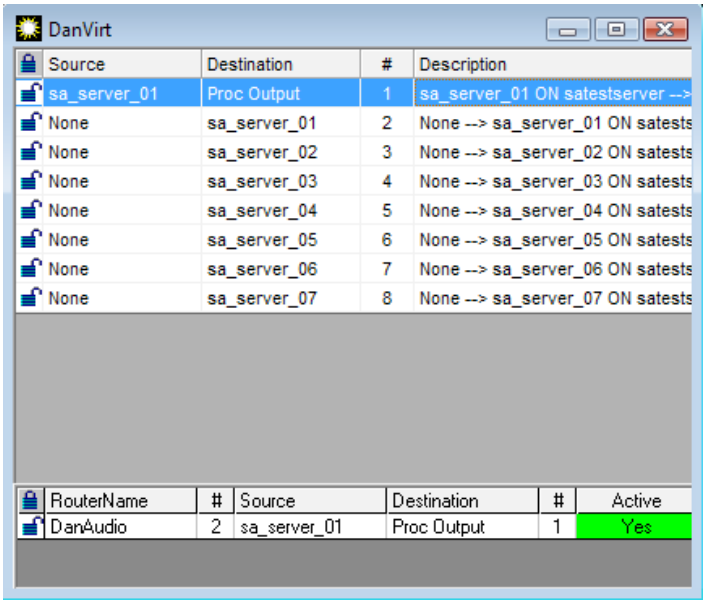
The **Exit** menu item will close the PathfinderPC_Core application.

Routers Menu

The Routers menu item lists the routers that are in the system. Clicking any Router in the list under this menu will open that router in the client.

The **Create Virtual Router** and **Edit Virtual Router** menu items will open the virtual router creation wizard. See the Virtual Router section above for details on creating and editing virtual routers.

The **Show Virtual Router Details** menu item can be turned on or off. If this option is turned on and a virtual router is open, there will be a section at the bottom of the list view of the virtual router. This will fill with data about the base routes involved in any virtual route selected in the router.



For any route selected in the virtual router that is selected this will show the base routers, sources, and destinations involved in the route.

The **View Hidden** menu item is no longer actively used. It displays routers that are hidden when they are involved in a gateway router.

Scenes Menu

The scenes menu will list any scenes available to the selected router. Clicking on one of the scenes in that menu will activate the scene. See the section above on editing and creating scenes.

The **Edit** menu item will open a scene in the scene editor wizard.

The **Capture Scene** will capture the existing router routing configuration and present in the scene editor for modification as a new scene.

The **New Empty Scene** menu item will open the scene editor with a new empty scene.

User Panels Menu

The User panel menu will only be present if user panels have been created using the Panel Designer application and saved to Pathfinder Core PRO. See Chapter 19 Panel Designer for details. Under this menu item, you will find a list of the custom user panels available to be opened and used by the system. Selecting any of the panels in this menu will open the panel in the PathfinderPC_Core Client.

Meter Bridge Menu

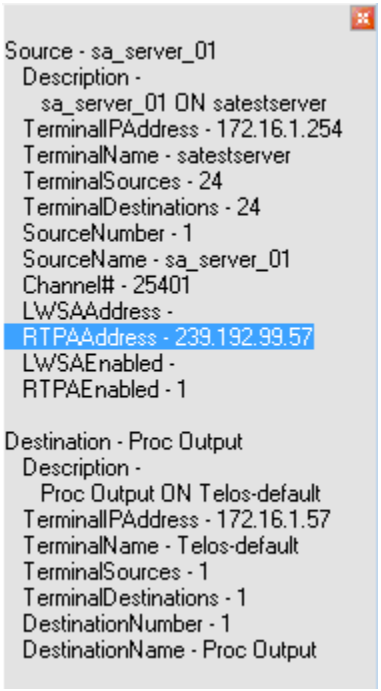
The meter bridge **Add** menu item will add meters for routes that are currently selected in an Axia Router to the meter bridge window. If the meter bridge window is not open, it will also open that window. This is equivalent to right clicking on the route line in the router window.

Toolbars Menu

The **Route Control** menu item displays and hides the Route Control toolbar on the left hand side of the screen.

The **Window Preset** menu item displays and hides the Window Preset toolbar on the bottom of the screen.

The **Route Information** menu item hides and displays a toolbar on the right hand side of the screen that displays additional information about Axia route points which have been selected in the router. This information includes such things as host IP, terminal names, and Livewire channel numbers.



View Menu

The **List** menu item presents the currently selected router in List view as described above.

The **Chart** menu item presents the currently selected router in Chart view as described above.

Window Menu

The Window menu displays all of the windows that are currently open in the PathfinderPC_Core Client. Selecting any of these windows from the list will bring that window to the front.

The **Cascade** menu item will reposition all of the currently open windows within the PathfinderPC_Core Client in a cascading fashion.

The **Tile** menu item will reposition all of the currently open windows within the PathfinderPC_Core Client in a tile fashion.

Help Menu

The **Help** menu item open the help system in PathfinderPC_Core Client.

The **About** Pathfinder PC menu item opens the splash screen which displays the version information for the PathfinderPC_Core Client.

PathfinderPC_Core Client Registry Settings

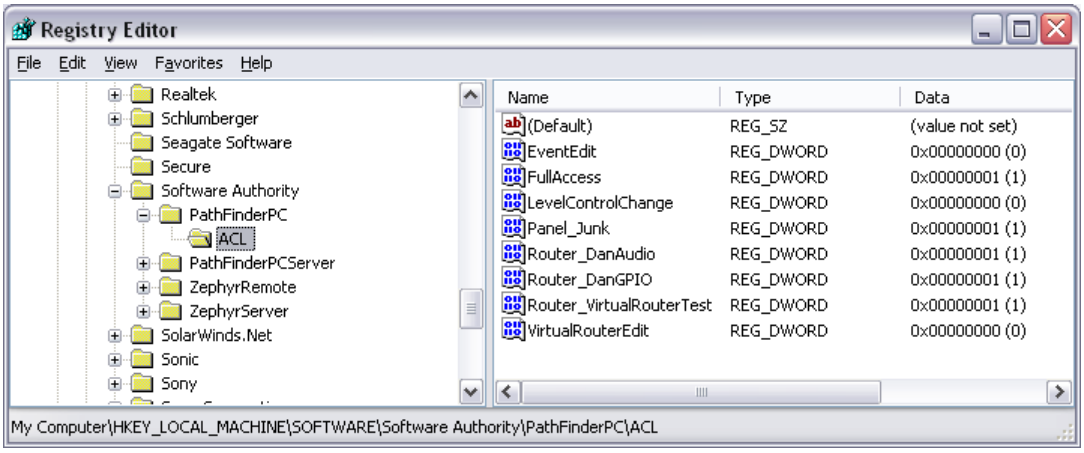
We have found that many clients do not want their users to have to login when running PathfinderPC_Core Client, but still want to restrict their access to particular routers, features, and panels. This restriction can be accomplished by modifying registry settings on the client computer. The registry key to edit is:

HKLM\Software\Software Authority\PathfinderPC\ACL\

All keys are of type DWORD. The following keys are available:

FullAccess (0 or 1) – This is set to 1 by default when the program starts and this registry key does not exist. 1 enables full access (according to user login rights if enabled on server). 0 limits access to the below entries. Setting this option to "1" overrides any of the other registry settings listed below.
Router_<RouterName> (0 or 1) for deny or allow
Panel_<PanelName> (0 or 1) for deny or allow
VirtualRouterEdit (0 or 1) for deny or allow
SceneChangeEdit (0 or 1) for deny or allow
LevelControlChange (0 or 1) for deny or allow

Setting the FullAccess Key to "0" will then allow granular control over client access restrictions to the system via the registry entries. If the Full Access key is set to 0, the other registry keys must be used to specifically deny or allow access to aspects of the system.



Chapter 19

Panel Designer

The PathfinderPC_Core Panel Designer application is a tool for creating custom user panels to be used either in the PathfinderPC_Core or PathfinderPCMini_Core applications. It should be noted that this is a deprecated method for working with panels that is being replaced by HTML5 panels as described earlier in this manual. This section is left in the manual for legacy use, but it is recommended that all customers start moving their panels to the new HTML5 panel format.

This allows you to create your own custom interfaces for your users with only the items you want them to use displayed. Using this tool, you can create a user panel with background colors and pictures, and drag and drop buttons, labels, meters, clocks, and web browsers onto the form. These controls can then be stretched to different sizes and moved and placed precisely on the form. Finally, each control has a list of properties that can be set to define the control's behavior when the panel is used in the system. It is important to understand that the Panel will not actually perform any routing or control functions from within the Panel Designer. The Panel Designer only allows you to design and define the panel. When you are finished working on the panel, it can be saved to Pathfinder Core PRO. Then any panel in this folder will become available for use by PathfinderPC_Core Client or PathfinderPCMini_Core.

The installer for PathfinderPC_Core Client may be downloaded from the Pathfinder Core PRO System web page. The client software may be installed on any Windows machine on the network that is to be used for making routing changes.

To Install Panel Designer:

- Click on the System web page in PathfinderCore PRO.
- Click on the Panel Designer link and download the Panel Designer installer.
- Double Click on the Panel Designer installer to install it.
- Click **Next** to begin the installation.
- Click **I Agree** to the license.
- Click **Next** again.
- Selection the location for installing the application or leave it at the default.
- Click **Next** a final time to finish the installation procedure.

Connecting Panel Designer to Pathfinder Core PRO

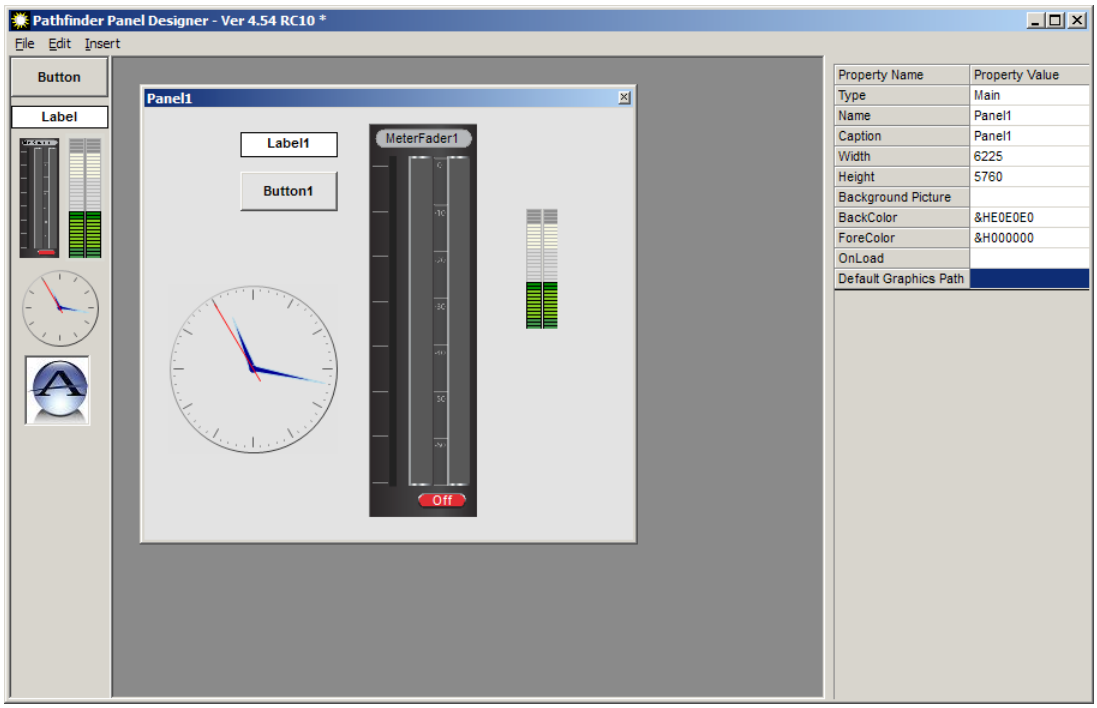
In order to work with the Panel Designer in Pathfinder Core PRO, you first must connect the two together. To do this, launch the Panel Designer Core application. Under the File Menu, Click Connect. Enter the IP address of the Pathfinder Core PRO device and leave the port as 5200. Click Connect.

The software will ask for a user name and password. Use a username and password defined under the User section of Pathfinder Core PRO. By default the username and password are:
UserName: Admin
Password: Admin

Once you are connected, there will be a green icon in the bottom left corner of the Panel Designer application showing Connected. If this icon is missing then you are not connected to Pathfinder Core PRO.

Creating a Panel

To begin creating a user panel, select New Panel from the File Menu. If you want to edit an existing panel, click the Open Panel menu item. As long as a connection has been established to Pathfinder Core PRO, you should see a list of panels on Pathfinder Core PRO to edit. Select the panel you wish to edit and click Open.



Once you have a panel open for editing, you can then resize the main panel form to be the correct size. At this point in time there are several objects available in the Panel Designer tool bar on the left of the application. These include buttons, labels, clocks, a meter control, a web browser control, and a MeterFader control which adds gain and console control features. Each Panel gets stored to a unique file and becomes a single user panel available to the PathfinderPC_Core Client. Buttons and Labels may be added to the panel by dragging the control from the left hand toolbox to the user panel being created.

Arranging controls on the form

Once a control is dropped into the panel, it becomes fixed on the form. In order to move a control, click the control to select it and then right click on the handle around the control. The mouse pointer icon will switch to the move pointer and allow you to drag the control around the screen. Using the shift and/or control keys, you can select multiple control, and then right click the handle of any one of them to move all of them as a group. In order to resize the control, click on it to select it and then left click and drag the handles. Again if you select multiple controls, you can resize all of them simultaneously.

The designer can select multiple controls using the shift and/or control keys and then use the Cut, Copy, and Paste functions from the menu or quick keys, to duplicate controls. The edit menu also provides automatic alignment tools. By selecting multiple controls and then selecting the Align Left, Right, Top, or Bottom, all of the selected controls can be aligned. The software will pick the farthest current edge to the selected direction out of the selected controls and align all of them to that edge. If the designer selects several controls and then selects align left, the software will find the control which currently has the left most edge, and it will align all of the selected controls to that edge. The spread vertically and horizontally menu item will spread the selected controls out evenly over the space occupied by the selected controls.

Setting Control Properties

Each object within a User Panel including the main form itself has a unique set of Properties that can be manipulated to define the look and behavior of the object when it gets used within the system. These are shown in the bottom of the Right hand panel. Whenever you click on an object in the User Panel, the appropriate set of properties for that control will display. It is these properties that must be set to make the control do anything in the Pathfinder System when the panel is opened and run using either PathfinderPC_Core Client or PathfinderPC_Core Mini.

For example, if you click on the main form of the newly created user panel you will see a list of 10 properties in the right hand column. If you grab the corner of the new user panel and drag to adjust the height and width of the form, you will notice that the height and width properties change to show the size of the form. Also if you double click in the caption property field and change the caption, it will change the caption at the top of the form. Each object has its own set of properties that do things within the system. The last part of this section will list all of their properties and their uses within the system

To continue with our first panel example, drag and drop a button control onto the panel. Next, Right click on the button and resize it to the desired size. Once you have gotten the button to the correct size, click somewhere else on the user panel to exit the resizing and moving mode. Then click on the button again to select it.

Next, with the button selected, double click on the caption property, and change the caption for the button. This will alter what the button says. You can also click in the BackColorOn and BackColorOff properties and then click on the button that appears in the property bar to open a color palette. This will allow you to define the colors to be used when the button is on or off. The button will change colors to the new color if you are changing the backcolor that matches the State drop-down for the control. You can change the state drop-down to on and off to view in the designer how the button will look when it is on or off. After changing the state drop-down you have to click on another property (leave the state property) before the change will appear. Make sure you leave the state property in the state you wish it to default to when the form is loaded. The state is usually used in routing panels to show whether a route exists, but when paired with stacking events or logic flows can be used to show any condition in the system you want. Finally, double-click in the name property and create a name for the button. If you are going to define custom roles for the user panel button through logic flows, the button will be identified in the logic flow by PanelName.ButtonName.

You could also drag and drop labels onto the panel. Labels are simply textual information that will be displayed in any color you like on the panel.

At this point, we have a panel that has a named button with a nice caption, but the button does not yet do anything in the system. Find the three properties MapRouter, MapSource, and MapDestination. If you enter numbers into these three properties that match the number of a router within Pathfinder, as well as the number of a source and destination within that router this button will activate that route when pushed from within PathfinderPC_Core Client and/or PathfinderPC_Core Mini.

To determine what numbers to use in these properties, open PathfinderPC_Core Client and find the source or destination you want using the ellipsis buttons next to source and destination in the routing tool bar. Find the number of the router that has the source and destination to be used. Enter this number into the MapRouter property. Find the numbers of the source and destination to be activated as a route when the button is pushed and enter those numbers into the MapSource and MapDestination properties.

Remember that pushing the button from within the Panel Designer will do nothing as it is just the designer program. The panel must first be saved to a Pathfinder Core PRO device, and then the panel should be opened using either PathfinderPC_Core Client or PathfinderPC_Core Mini. Assuming that our Audio Router is router 1 and our button is going to route Source 1 to Destination 1, make MapRouter, MapSource, and MapDestination all 1. Before we try to use this new panel, add one more button. Create a caption and name for this button as well, and set MapRouter and MapDestination also to 1 (or whatever number router your audio router is), but set the Source to 2. This will mean that when we run the panel clicking the top button will route 1 to 1 and clicking the second button will route 2 to 1. Furthermore, the Mapping process will automatically change the state of either button if the route for that button is active so that it will toggle the backcolor colors. Next, save the Panel, giving it a name.

Launch the PathfinderPC_Core Client application and make sure it is connected to Pathfinder Core PRO. If it is already started, you may have to restart it before the panel will be available. You will notice that there should now be a menu in PathfinderPC_Core Client called User Panels. Under that menu item you will find your newly created user panel. If you create additional panels and save to Pathfinder Core PRO, they also will appear under this menu item. Click on the name of the panel to open the panel. If the MapRouter, MapSource, and MapDestination fields have valid entries, clicking the buttons should now cause the routes to be made and the button colors to change accordingly.

To use this panel within PathfinderPC_Core Mini, edit the configuration file and be sure the server IP addresses are defined, and enter the PanelName to open in the configuration file. See the section on PathfinderPC_Core Mini for details. Launching PathfinderPC_Core Mini will run this panel as its own application.

If you close and reopen the panel designer, you can use the File menu to open and edit the Panel if you want to change it later. Once you re-save the changes you must close and open the panel again within PathfinderPC_Core Client to get the changes to display. In PathfinderPC_Core Mini, just close and re-launch the application to view your changes. You can also create additional panels and save them to the folder, and they will be available to these applications.

Remember that the MapRouter/Source/Destination is only one thing you can do with the panels. You could also leave these fields blank and use Logic Flows to determine the button functions. The section on latching buttons in the Logic Flows section of this document is a good example. In addition the detailed property descriptions below will provide more insight on the wide variety of functions these controls can perform.

To continue with our example, open the Panel you were just designing in Panel Designer again. One of the other powerful features of these panels is that once created, their button functions can be mapped to actual hardware controls as well using either GPIO circuits or Axia Element User Panel buttons. To accomplish this, select the button you wish to map to a hardware control, and then find the HWMapType property and select the Button, Label, or GPIO button option. You then must select a different property for the change to update. Essentially you must leave the property for the change to get written into the panel definition. This is true of most of the properties. The Button option means that this button will get mapped to an Axia User Panel button. The Label property means that this button will be mapped to the LCD display of an Axia User Panel. The GPIO Button option means that it will map to a GPIO Circuit. You will notice that once you make your selection and then leave the selection field the options in the HWMap section will change depending on whether a GPIO option or User Panel option was selected.

Next under HWMapHostIP enter the IP address of the Element CPU or GPIO box that hosts the hardware device to be used. The HWMapHostPort will have already been selected when the HWMapType was selected and should not be casually changed. The HWMapInternalName Property is also automatically defined and matches the software PanelName.ControlName to which you are mapping. The HWMapAxiaSysName is automatically defined by the last two properties on the list, and cannot be manually set. If you have selected a GPIO type map then these last two properties will be GPIO Port and GPIO Circuit. Enter the number of the port on the device defined by the HWMapHostIP, and the GPI/GPO pin to use.

If the HWMapType is a UserPanel Button or Label (LCD Display), then these properties will be UserPanelModule and UserPanelButton. The module number is related to the dip switch setting on the bottom of the module in the console. However, there is a much simpler method for obtaining the module number if you don't know it. Open a command prompt and type:

telnet ipaddress 4010

example:
Telnet 172.16.1.3 4010

For the IP address use the IP address of the Element CPU that hosts the user module. Then press a few buttons on the module. You should see commands coming back that look like:
EVENT MOD_USER#3.BUT#6 KEY=DOWN
EVENT MOD_USER#3.BUT#6 KEY=UP

The module number you are looking for is the number after MOD_USER#. In this case, it would be 3. The button numbers start from 1 at the bottom of the button panel strip and count up to the top of the strip. Enter the module number and the button number that should be mapped to the software button. Now save the panel.

When you save the panel, it will automatically generate custom logic flows under a special HardwareMaps View. This also allows you to view the mapping functionality in logic flows. If you have entered the Host IP and additional settings correctly, those hardware controls should do exactly the same thing as the software buttons. This is true whether or not the Client application has the panel open or is even running. As long as Pathfinder Core PRO is running those hardware controls will function. In the case of the GPIO control, whenever the route is active, the GPO will be low causing the GPIO button to be lit. Also, pushing the button and closing the GPI will cause the route to be activated. The same is true for the user panel buttons, except that in that case the button's captions and colors will also match the software panel color and caption selections. However, if you are using certain custom colors, it may take some tweaking to make the software panel color and hardware panel button color look the way you want them to. Because of the button backlighting, the same RGB color settings do not always look exactly the same between the hardware and software button.

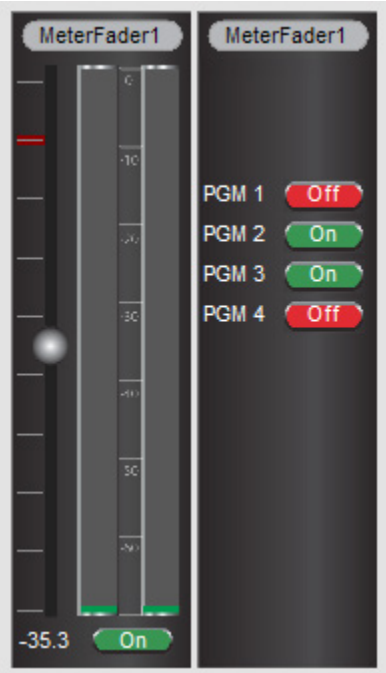
Again, remember that we could also leave all of the hardware map and MapRouter, Source, Destination fields blank and instead use logic flows to define the button behavior.

To do that, create a new logic flow on Pathfinder Core PRO using the logic flows link. Review Chapter 6 Logic Flows if you are unsure how to create a new flow. For the start point drill down into the buttons branch, and then into the legacy panels branch. The LegacyPanels branch will only exist if a panel with buttons exists in the system. Find your panel and expand it and then find the button you wish to use. Under the button will be a list of properties for that button. In this case, we will select the IsDown property. Then click the Select button. Now we have to define what will happen when the button is pressed. We need to double click the endpoint and select the property we want to change whenever the button is pushed. For example, we could select a route change so that every time the button is pushed a route change happens. Or we could change the color and/or caption of the button. Or do any other change we wish to make in the system.

You will also notice that some of the properties are picture-related. Pathfinder Core PRO will not currently serve out pictures, but the client applications will look in specific places for the graphic files. If you click on the main panel in Panel Designer, you will find a property called Default Graphics Path. In this field you can select a network share to which all of the client applications have access. Then if the graphics files are placed in this share they can be found by all instances running the program. The second possibility is that the files can be installed in a folder called PanelGraphics within the PathfinderPC_Core Client or Mini installation folder that will use them. When the panel is opened graphics files will be searched for in each of these two locations. By using custom background and button pictures we have already seen some incredibly slick and creative panels being used in the field.

There is a clock control with a variety of properties that can be positioned and resized on the form. You can display the clock with and without the face hashes and numbers and with your own background picture and/or colors so that the clock can be customized for the look your panel requires. It can also be switched between analog and digital.

Meter controls display dynamic metering for a source or destination. There are two meter controls in the toolbox. A standard meter and a MeterFader. The standard meter can be dropped onto a form and assigned to a router and source or destination and it will supply metering for the source or destination. The MeterFader control is configured the same way, but it adds some additional functionality. When the panel executes in PathfinderPC_Core Client or Mini, the MeterFader control is intelligent enough to know what kind of source or destination it is connected to. Depending on the source or destination's capabilities, the control may provide a fader for changing the gain, and controls for turning console or VMIX channels on or off. Additionally, if the MeterFader is tied to an Axia Console Channel, right clicking in the fader will present options for changing the program buss assignments.



These controls work exactly like the meters in PathfinderPC_Core Client that are available by right clicking on a specific route.

Finally, you can use the Web Browser control to display web pages which may have more dynamic content such as embedded video or audio players. In this way, you can add to the richness of the user panel.

Property Descriptions

Important Note: Typically, you must move off of the field you are editing to a different field before the changes will be stored into the panel configuration memory.

Panel Properties

Type	Defines the object type as main panel and cannot be changed by the designer.
Name	Defines the name of the Panel. When you save the Panel this name will automatically be set to the name of the file less the file-type extension. Also setting this property will cause the save function to default to a filename matching the panel name. The Panel Name and Filename should always match.
Caption	The caption that will be displayed at the top of the panel.
Width and Height	Displays the width and height of the form. Change these parameters by resizing the form.
BackgroundPicture	Select a background picture for the form. This picture file must be available to all clients either in the network share folder defined in the Default Graphics Path property, or it must exist on each client machine in the Panel-Graphics folder in the PathfinderPC_Core Client or Mini installation folder.
Backcolor	The background color of the form. Defining a picture in the BackgroundPicture property will override this setting.
ForeColor	Not used.
OnLoad	Used to run customized scripts and scripting commands when the panel loads. See the scripting documentation for details of commands that can be used in this field.
Default Graphics Path	Defines a network share where client applications may search for the graphics files defined throughout the panel sets.

Label Control Properties

Type	Defines the object type as label and cannot be changed by the designer.
Name	Defines the name of the label control. This name must be unique within the panel.
Caption	Sets the textual information to be displayed in the label.
Height	Displays the height of the control. Change this parameter by right-clicking on the control and resizing it or manually editing this property value.
Width	Displays the width of the control. Change this parameter by right-clicking on the control and resizing it or manually editing this property value.
Top	Displays the position of the top edge of the control on the panel. Change this parameter by right-clicking on the control dragging it to a new location, or manually editing this property value.
Left	Displays the position of the left edge of the control on the panel. Change this parameter by right-clicking on the control dragging it to a new location, or manually editing this property value.
BackcolorOff	The background color of the label when it is in the off state.
BackcolorOn	Defines the background color of the label when it is in the on state.
ForeColor	Defines the color of the text within the label. This will display in the software user panels, but the Axia Hardware User Panels only have the ability to change the background color, and not the text color.
Font Name	The font name to be used for text in the label.
Font Size	The font size to be used for text in the label.
Font Bold	Whether the text should be bold or not.
Font Underline	Whether the text should be underlined or not.
Font Italic	Whether the text should be italicized or not.
State	Sets the state of the control On or Off. The BackcolorOn and Off will follow this state change.
Flash	Will turn flashing on or off. Flashing will cause the background color to switch back and forth between the BackgroundOn color and the BackgroundOff color.
HwMapType	If the control is to be mapped to a hardware control, this selects the type of hardware control to which to map. The options are Button, Label, and GPIO. GPIO will map to a GPIO circuit, Button will map to an Axia User Panel Button, and label will map to the LCD label strip on an Element User button panel.
HWMapHostIP	The IP address of the Element CPU or GPIO device that hosts the control for the hardware map device
HWMapHostPort	This is the TCP port to which to connect to control the Hardware device. This should not generally be changed by the designer. It is automatically selected when the HWMapType is selected

HWMapInternalName	This is the name of the software object that is being mapped to the hardware object. Again, this field is automatically set according to the name of the Panel and the name of the control. It should generally not be changed by the designer.
HWMapAxiaSysName	This is the name of the hardware object being mapped to the software object. This cannot be directly changed by the designer, but is created based on the remaining two drop down properties (For GPIO – Port Number and Circuit Number, For Button and Label – UserPanel Module and UserPanel Button).

GPIO Port	When the HWMapType is GPIO this selects the GPIO port on a GPIO device that will be used in the software to hardware map.
GPIO Circuit	When the HWMapType is GPIO this selects the GPIO circuit on the GPIO port above that will be used in the software to hardware map. This GPIO port and circuit then acts like the software button. The GPO will be closed when the button state is On and closing the GPI will do the same function as pressing the software button. Once the GPIO Port and GPIO Circuit properties are both entered, the HWMapAxiaSysName will automatically be filled in accordingly
UserPanel Module	Defines the Panel ID within the Element Console to address. This number can be determined by telnetting to the IP address of the element CPU on Port 4010, and tapping several of the buttons. For Example: *Open a command prompt on the windows PC and type Telnet IPAddress 4010 - "Telnet 172.168.2.3 4010"* *Press Enter. This should open a Telnet session with the CPU.* *Now tap several of the user panel buttons. You should see messages that look like:* *EVENT MOD_USER#3.BUT#6 KEY=DOWN* *EVENT MOD_USER#3.BUT#6 KEY=UP* *The Module number is the one after the first # sign. In this case 3. So 3 would be the number to enter in the UserPanel Module field.*
UserPanel Button	Identifies the number of the button on the panel to map to the software control. The bottom button is 1, and they count up to the top of the panel. Thus on a ten button panel the top button is ten. Once the UserPanel Module and UserPanel Button properties are both entered, the HWMapAxiaSysName will automatically be filled in accordingly.

Button Control Properties

Type	This defines the object type as button and cannot be changed by the designer.
Name	Defines the name of the button control. This name must be unique within the panel.
Caption	Sets the textual information to be displayed in the button.
Height	The height of the control. Change this parameter by right-clicking on the control and resizing it or manually editing this property value.
Width	Displays the width of the control. Change this parameter by right-clicking on the control and resizing it or manually editing this property value.
Top	Displays the position of the top edge of the control on the panel. Change this parameter by right-clicking on the control dragging it to a new location, or manually editing this property value.
Left	Displays the position of the left edge of the control on the panel. Change this parameter by right-clicking on the control dragging it to a new location, or manually editing this property value.
BackcolorOff	Defines the background color of the button when it is in the off state. This color will be overridden if a picture is assigned to the button.
BackcolorOn	Defines the background color of the button when it is in the on state. This color will be overridden if a picture is assigned to the button.
ForeColor	Defines the color of the text within the button. This cannot be changed for button controls.
Font Name	The font name to be used for text in the button.
Font Size	The font size to be used for text in the button.
Font Bold	Whether the text should be bold or not.
Font Underline	Whether the text should be underlined or not.
Font Italic	Whether the text should be italicized or not.
State	Sets the state of the button to On or Off. The BackcolorOn and Off will follow this state change
OnClick	Used to run customized scripts and scripting commands when the button is clicked. See the scripting documentation for details of commands that can be used in this field.

OnMouseDown	Used to run customized scripts and scripting commands when the button is pushed down. See the scripting documentation for details of commands that can be used in this field.
OnMouseUp	Used to run customized scripts and scripting commands when the button is released. See the scripting documentation for details of commands that can be used in this field.
Picture_Up_On	Defines a picture to be used in the button when the button is up and the State is On. In order for pictures to be used they must be located either in the network share defined by the Panel Form's Default Graphics Path or within the PanelGraphics folder within each PathfinderPC_Core Client's or PathfinderPC_Core Mini's installation folder. You can view the changes to these properties in the designer by assigning a picture and then pressing and releasing the button with the state property either on or off.
Picture_Up_Off	Defines a picture to be used in the button when the button is up and the State is Off. In order for pictures to be used they must be located either in the network share defined by the Panel Form's Default Graphics Path or within the PanelGraphics folder within each PathfinderPC_Core Client's or PathfinderPC_Core Mini's installation folder. You can view the changes to these properties in the designer by assigning a picture and then pressing and releasing the button with the state property either on or off.
Picture_Down_On	Defines a picture to be used in the button when the button is down and the State is On. In order for pictures to be used they must be located either in the network share defined by the Panel Form's Default Graphics Path or within the PanelGraphics folder within each PathfinderPC_Core Client's or PathfinderPC_Core Mini's installation folder. You can view the changes to these properties in the designer by assigning a picture and then pressing and releasing the button with the state property either on or off.
Picture_Down_Off	Defines a picture to be used in the button when the button is down and the State is Off. In order for pictures to be used they must be located either in the network share defined by the Panel Form's Default Graphics Path or within the PanelGraphics folder within each PathfinderPC_Core Client's or PathfinderPC_Core Mini's installation folder. You can view the changes to these properties in the designer by assigning a picture and then pressing and releasing the button with the state property either on or off.
FlashPicture	This property is not currently enabled.
Flash	Will turn flashing on or off. Flashing will cause the background color to switch back and forth between the BackgroundOn color and the BackgroundOff color.
Enabled	Will make the button enabled or disabled for actual use in the panel.
MapRouter	Specifies a router when mapping a route point to a button. When the panel is running, a mapped route point will turn the button on when the route exists and off when it does not. And pressing the button will send a route change request to the equipment using the MapRouter, MapSource, and MapDestination fields. All three of these fields must be entered for this to work properly.
MapSource	Specifies a source when mapping a route point to a button. When the panel is running, a mapped route point will turn the button on when the route exists and off when it does not. And pressing the button will send a route change request to the equipment using the MapRouter, MapSource, and MapDestination fields. All three of these fields must be entered for this to work properly.
MapDestination	Specifies a destination when mapping a route point to a button. When the panel is running, a mapped route point will turn the button on when the route exists and off when it does not. Pressing the button will send a route change request to the equipment using the MapRouter, MapSource, and MapDestination fields. All three of these fields must be entered for this to work properly.
HwMapType	If the control is to be mapped to a hardware control, this selects the type of hardware control to which to map. The options are Button, Label, and GPIO. GPIO will map to a GPIO circuit, Button will map to an Axia User Panel Button, and label will map to the LCD label strip on an Element User button panel.
HWMapHostIP	The IP address of the Element CPU or GPIO device that hosts the control for the hardware map device
HWMapHostPort	This is the TCP port to which connects to the Hardware device. It should not generally be changed by the designer. It is automatically selected when the HWMapType is selected.
HWMapInternalName	This is the name of the software object that is being mapped to the hardware object. This field is automatically set according to the name of the Panel and the name of the control. It should generally not be changed by the designer.
HWMapAxiaSysName	This is the name of the hardware object being mapped to the software object. This cannot be directly changed by the designer, but is created based on the remaining two drop down properties (For GPIO – Port Number and Circuit Number, For Button and Label – UserPanel Module and UserPanel Button).
GPIO Port	When the HWMapType is GPIO, this selects the GPIO port on a GPIO device that will be used in the software to hardware map.
GPIO Circuit	When the HWMapType is GPIO this selects the GPIO circuit on the GPIO port above that will be used in the software to hardware map. This GPIO port and circuit then acts like the software button. The GPO will be closed when the button state is On and closing the GPI will do the same function as pressing the software button. Once the GPIO Port and GPIO Circuit properties are both entered, the HWMapAxiaSysName will automatically be filled in accordingly

UserPanel Module	Defines the Panel ID within the Element Console to address. This number can be determined by telnetting to the IP address of the element CPU on Port 4010, and tapping several of the buttons. For Example: Open a command prompt on the windows PC and type Telnet IPAddress 4010 - "Telnet 172.168.2.3 4010" Press Enter. This should open a Telnet session with the CPU. Now tap several of the user panel buttons. You should see messages that look like: EVENT MOD_USER#3.BUT#6 KEY=DOWN EVENT MOD_USER#3.BUT#6 KEY=UP The Module number is the one after the first # sign. In this case 3. This would be the number to enter in the UserPanel Module field.
UserPanel Button	Identifies the number of the button on the panel to map to the software control. The bottom button is 1, and they count up to the top of the panel. On a ten-button panel, the top button is ten. Once the User-Panel Module and UserPanel Button properties are both entered, the HWMapAxiaSysName will automatically be filled in accordingly.

Meter Control Properties

Type	This defines the object type as Meter and cannot be changed by the designer.
Name	Defines the name of the Meter control. This name must be unique within the panel.
Caption	Sets the textual information to be displayed in the Meter. This is not currently used.
Height	The height of the control. Change this parameter by right-clicking on the control and resizing it or manually editing this property value.
Width	Displays the width of the control. Change this parameter by right-clicking on the control and resizing it or manually editing this property value.
Top	Displays the position of the top edge of the control on the panel. Change this parameter by right-clicking on the control dragging it to a new location, or manually editing this property value.
Left	Displays the position of the left edge of the control on the panel. Change this parameter by right-clicking on the control dragging it to a new location, or manually editing this property value.
Source/Dest	Defines whether the monitored I/O will be a source or a destination.
Router Number	Defines number of the router that will be used for this meter source or destination.
IO Number	Defines the number of the source or destination to monitor.
Scale	Used to select the numeric scale to be used for the meter – Standard, British, or Linear.
Style	Used to select the style as either led or gradient.
Line Color	Defines the color of level marking lines.
Decay Rate	Defines the decay rate for more natural metering. Approximately 3000ms generally yields good results.
Center Pad	Defines the space between the left and right meter. If expanded this also becomes the space designated for level markings.

MeterFader Control Properties

Type	This defines the object type as MeterFader and cannot be changed by the designer.
Name	Defines the name of the Meter control. This name must be unique within the panel.
Caption	Sets the textual information to be displayed in the MeterFader.
Height	The height of the control. Change this parameter by right-clicking on the control and resizing it or manually editing this property value.
Width	Displays the width of the control. Change this parameter by right-clicking on the control and resizing it or manually editing this property value.
Top	Displays the position of the top edge of the control on the panel. Change this parameter by right-clicking on the control dragging it to a new location, or manually editing this property value.
Left	Displays the position of the left edge of the control on the panel. Change this parameter by right-clicking on the control dragging it to a new location, or manually editing this property value.
Source/Dest	Defines whether the monitored I/O will be a source or a destination.
Router Number	Defines number of the router that will be used for this meter source or destination.
IO Number	Defines the number of the source or destination to monitor.
Scale	Used to select the numeric scale to be used for the meter – Standard, British, or Linear.
Style	Used to select the style as either led or gradient.
Line Color	Defines the color of level marking lines.
Decay Rate	Defines the decay rate for more natural metering. Approximately 3000ms generally yields good results.
Center Pad	Defines the space between the left and right meter. If expanded this also becomes the space designated for level markings.

WebBrowser Control Properties

Type	This defines the object type as WebBrowser and cannot be changed by the designer.
Name	Defines the name of the Web Browser control. This name will be used to identify the control in stack events in the form PanelName.ControlName.
Caption	Sets the textual information to be displayed in the WebBrowser. This is not used at this point in time.
Height	The height of the control. Change this parameter by right-clicking on the control and resizing it or manually editing this property value.
Width	Displays the width of the control. Change this parameter by right-clicking on the control and resizing it or manually editing this property value.
Top	Displays the position of the top edge of the control on the panel. Change this parameter by right-clicking on the control dragging it to a new location, or manually editing this property value.
Left	Displays the position of the left edge of the control on the panel. Change this parameter by right-clicking on the control dragging it to a new location, or manually editing this property value.
URL	Defines the URL to display in the web browser.

Clock Control Properties

Type	This defines the object type as Clock and cannot be changed by the designer.
Name	Defines the name of the Clock control. This name must be unique within the panel.
Caption	Sets the textual information to be displayed in the Clock. This is not currently used.
Height	The height of the control. Change this parameter by right-clicking on the control and resizing it or manually editing this property value. If the clock is Analog, the height and width will be forced to be equal. If it is set to Digital, height and width may be different.
Width	Displays the width of the control. Change this parameter by right-clicking on the control and resizing it or manually editing this property value. If the clock is Analog, the height and width will be forced to be equal. If it is set to Digital, height and width may be different.
Top	Displays the position of the top edge of the control on the panel. Change this parameter by right-clicking on the control dragging it to a new location, or manually editing this property value.
Left	Displays the position of the left edge of the control on the panel. Change this parameter by right-clicking on the control dragging it to a new location, or manually editing this property value.
BackColor	Defines the background color of the clock. This color will be overridden if a picture is assigned to the clock.
ForeColor	Defines the Foreground color of the clock for the numbers if displayed.
FaceVisible	Defines whether the face circle and hash marks are visible. You may choose to use a background image of your own design for the clock instead of the drawn face.
Digital	If True, the clock will display digital time instead of an analog face.
24 Hour	If the clock is in digital mode, this defines if the time will be in military 24 hour mode.
Background Picture	Defines a background picture for the clock.
Stretch Image	Determines if a background image for the clock should be centered or stretched to fill the clock space.
Font Name	Font to be used for numbers.
Font Size	Font Size to be used for the numbers.
Font Bold	Whether the font is bold.
Font Underline	Whether the font is underlined.
Font Italic	Whether the font is italic.
Long Hash Color	The color of the longer hash marks at the 5 minute intervals on the analog clock face.
Short Hash Color	The color of the shorter hash marks at the minute intervals on the analog clock face.
Face Border Color	The Color of the main clock face circle.
Time Offset	The amount of time the clock should be offset from the computer's time if desired.
Face Numbers	If True, the numbers 1 through 12 will be displayed on the analog clock face. Use the font settings to modify the look and size of the numbers.
Element Hands	Defines whether the clock hands should be simple black lines or look more like the clock hands on the Axia Element Console screen.

Chapter 20

PathfinderPC_Core Mini

PathfinderPC_Core Mini is a small application that displays a specified user panel as its own application. It should be used in situations where the PathfinderPC_Core Administrator wants the user to only have access to the routes and control offered by a specific user panel and nothing else.

It should be noted that this is a deprecated method for working with panels that is being replaced by HTML5 panels as described earlier in this manual. This section is left in the manual for legacy use, but it is recommended that all customers start moving their panels to the new HTML5 panel format.

The installer for PathfinderPC_Core Mini may be downloaded from the Pathfinder Core PRO System web page. The client software may be installed on any Windows machine on the network that is to be used for making routing changes.

To Install PathfinderPC_Core Mini:

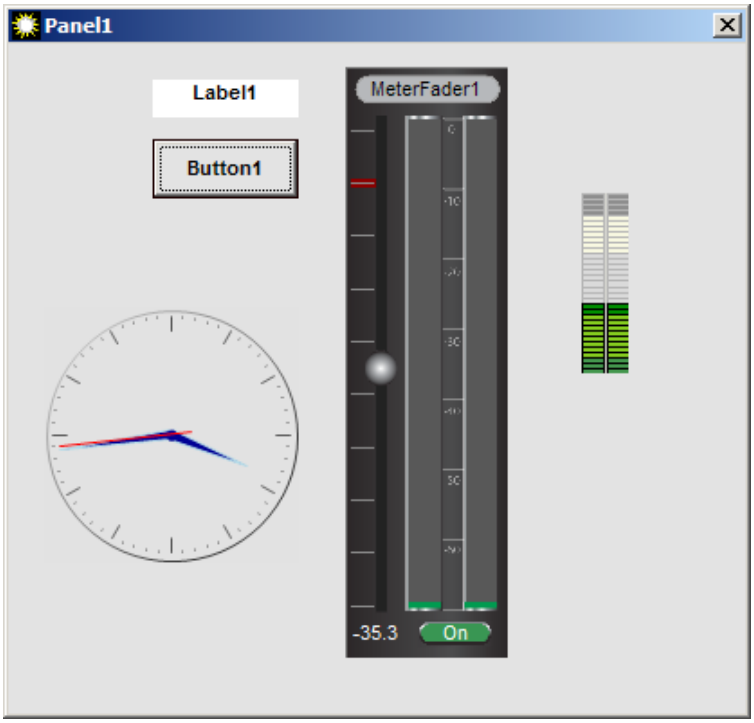
- Click on the System web page in PathfinderCore PRO.
- Click on the PathfinderPC Core Mini link and download the PathfinderPCMini_Core installer.
- Double Click on the PathfinderPCMini_Core installer to install it.
- Click **Next** to begin the installation.
- Click **I Agree** to the license.
- Click **Next** again.
- Selection the location for installing the application or leave it at the default.
- Click **Next** a final time to finish the installation procedure.
- If the application is being installed on Windows 7 or later, it is recommended that the following option also be turned on to ensure PathfinderPC_Core Mini has the ability to properly manage its own files:
    - » Find the PathfinderMini_Core.exe application located in the C:\Program Files(X86)\ Pathfinder-Mini_Core folder.
    - » Right click on the PathfinderMini_Core.exe application and select properties.
    - » Click on the compatibility tab.
    - » Enable the Run As Administrator checkbox.

To use PathfinderMini_Core, the administrator needs to complete the details in a small configuration file stored in the PathfinderMini_Core installation folder. This file is called PFMiniConfig.txt. Open this file with any text editor. It should look something like this.

PrimaryServerIP=172.16.1.3
PrimaryServerPort=5200
BackupServerIP=172.16.1.1
BackupServerPort=5200
PanelName=Studio_1
UserName=Admin
UserPassword=Admin

Change the Server IP Addresses for Pathfinder Core PRO and a second clustered Pathfinder Core PRO if desired. Also enter the name of the Panel to be displayed. If you have changed or created specific user accounts within Pathfinder Core PRO, you may also need to enter a valid user name and password.

Save the file and launch PathfinderMini_Core. As long as the application can find the server and finds the Panel within Pathfinder Core PRO, the panel will be displayed and run as its own application.



Appendix A:

SapV2

SapV2 stands for Software Authority Protocol Version 2. It is the glue that provides all of 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. To give you an idea, 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!

Content Note: Because of the width of the page in this document, many of the example commands will wrap to the next line. That is not the case with the actual messages as they appear in the system.

In order to interact with SapV2, open up 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 login:.

Login <username> <password>

Replace the bracketed username and password with a valid user and password.

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 are listed and described later in this document.

The object path represents branches of a tree similar to 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", HasGpios="True", HasAudio="False", Name="Element", FriendlyName="Element_172.16.1.51", Description="", DiscoveredDateTime="02/06/2015 13:08:41", PathId="tcp://172.16.1.51:93", ShortId="2", AutoReconnect="20000"

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", FriendlyName="EndpointInvestigator" $MAX_DEPTH=2
indi Devices#0.EndpointDiscoverers#0 FriendlyName="EndpointDiscoverers" $MAX_DEPTH=2 indi Devices#0.DeviceCreator#0 Abort="False", OpenInvestigatorCount="0", FriendlyName="DeviceCreator" $MAX_DEPTH=2
indi Devices#0.MicrophoneNode#[tcp://172.16.1.99:93] DeviceType="livemic", HasGpios="False", HasAudio="True", Name="AAMD-001-99", FriendlyName="AAMD-001-99_172.16.1.99", Description="", DiscoveredDateTime="02/06/2015 13:08:42", PathId="tcp://172.16.1.99:93", ShortId="12", AutoReconnect="20000" $MAX_DEPTH=2
indi Devices#0.AnalogNode#[tcp://172.16.1.81:93] DeviceType="liveio", HasGpios="False", HasAudio="True", Name="AAND-001-081", FriendlyName="AAND-001-081_172.16.1.81", Description="", DiscoveredDateTime="02/06/2015 13:08:41", PathId="tcp://172.16.1.81:93", ShortId="5", AutoReconnect="20000" $MAX_DEPTH=2
indi Devices#0.MicrophoneNode#[tcp://172.16.1.99:93].LwrpInterpreter#0 FriendlyName="LwrpInterpreter" $MAX_DEPTH=2
indi Devices#0.AnalogNode#[tcp://172.16.1.97:93].LwrpInterpreter#0 FriendlyName="LwrpInterpreter" $MAX_DEPTH=2

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", FriendlyName="EndpointInvestigator" $MAX_DEPTH=2
indi Devices#0.EndpointDiscoverers#0 FriendlyName="EndpointDiscoverers" $MAX_DEPTH=2
indi Devices#0.DeviceCreator#0 Abort="False", OpenInvestigatorCount="0", FriendlyName="DeviceCreator" $MAX_DEPTH=2
indi Devices#0.MicrophoneNode#[tcp://172.16.1.99:93] DeviceType="livemic", HasGpios="False", HasAudio="True", Name="AAMD-001-99", FriendlyName="AAMD-001-99_172.16.1.99", Description="", DiscoveredDateTime="02/06/2015 13:08:42", PathId="tcp://172.16.1.99:93", ShortId="12", AutoReconnect="20000" $MAX_DEPTH=2
indi Devices#0.AnalogNode#[tcp://172.16.1.81:93] DeviceType="liveio", HasGpios="False", HasAudio="True", Name="AAND-001-081", FriendlyName="AAND-001-081_172.16.1.81", Description="", DiscoveredDateTime="02/06/2015 13:08:41", PathId="tcp://172.16.1.81:93", ShortId="5", AutoReconnect="20000" $MAX_DEPTH=2
indi Devices#0.MicrophoneNode#[tcp://172.16.1.99:93].LwrpInterpreter#0 FriendlyName="LwrpInterpreter" $MAX_DEPTH=2
indi Devices#0.AnalogNode#[tcp://172.16.1.97:93].LwrpInterpreter#0 FriendlyName="LwrpInterpreter" $MAX_DEPTH=2 $DONE

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.GpioPinState#4 Pinstate=”h” $MAX_DEPTH=-1
indi Devices#0.WinDriver#[tcp://172.16.1.254:93].LwrpInterpreter#0.LwrpRoot#0.Gpi#10.GpioPinState#3 Pinstate=”h” $MAX_DEPTH=-1
indi Devices#0.WinDriver#[tcp://172.16.1.254:93].LwrpInterpreter#0.LwrpRoot#0.Gpi#10.GpioPinState#1 Pinstate=”h” $MAX_DEPTH=-1
indi Devices#0.WinDriver#[tcp://172.16.1.254:93].LwrpInterpreter#0.LwrpRoot#0.Gpi#10.GpioPinState#5 Pinstate=”h” $MAX_DEPTH=-1
indi Devices#0.WinDriver#[tcp://172.16.1.254:93].LwrpInterpreter#0.LwrpRoot#0.Gpi#14.GpioPinState#2 Pinstate=”h” $MAX_DEPTH=-1
indi Devices#0.WinDriver#[tcp://172.16.1.254:93].LwrpInterpreter#0.LwrpRoot#0.Gpi#14.GpioPinState#4 Pinstate=”h” $MAX_DEPTH=-1
indi Devices#0.WinDriver#[tcp://172.16.1.254:93].LwrpInterpreter#0.LwrpRoot#0.Gpi#14.GpioPinState#3 Pinstate=”h” $MAX_DEPTH=-1 $DONE

Please note: The example above is abbreviated. In reality this would 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.GpioPinState#2 Pinstate="h", FriendlyName="PIn 2" $MAX_DEPTH=-1
indi Devices#0.WinDriver#[tcp://172.16.1.254:93].LwrpInterpreter#0.LwrpRoot#0.Gpo#18.GpioPinState#3 Pinstate="h", FriendlyName="PIn 3" $MAX_DEPTH=-1
indi Devices#0.WinDriver#[tcp://172.16.1.254:93].LwrpInterpreter#0.LwrpRoot#0.Gpo#18.GpioPinState#5 Pinstate="h", FriendlyName="PIn 5" $MAX_DEPTH=-1
indi Devices#0.WinDriver#[tcp://172.16.1.254:93].LwrpInterpreter#0.LwrpRoot#0.Gpo#13.GpioPinState#5 Pinstate="h", FriendlyName="PIn 5" $MAX_DEPTH=-1
indi Devices#0.WinDriver#[tcp://172.16.1.254:93].LwrpInterpreter#0.LwrpRoot#0.Gpo#13.GpioPinState#4 Pinstate="h", FriendlyName="PIn 4" $MAX_DEPTH=-1 $DONE

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.GpioPinState#2 Pinstate="l", FriendlyName="PIn 2" $MAX_DEPTH=-1
indi Devices#0.WinDriver#[tcp://172.16.1.254:93].LwrpInterpreter#0.LwrpRoot#0.Gpo#18.GpioPinState#3 Pinstate="l", FriendlyName="PIn 3" $MAX_DEPTH=-1 $DONE

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.GpioPinState#2 Pinstate
sfr Devices#0.WinDriver#[tcp://172.16.1.254:93].LwrpInterpreter#0.LwrpRoot#0.Gpo#6.GpioPinState#2 Pinstate=
”[ReadWrite=RW,SyntaxType=OPT,IsStable=True,SyntaxUsage=\”Single:l(Low),h(High)\”,IsSimpleUi=True]”

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 it’s syntax should not be expected to change in future versions of PFCore unless carefully noted and/or with backwards 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 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 an alphabetical way that is conducive to reading 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.GpioPinState#2 Pinstate="l"

** 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.GpioPinState#2 Pinstate="h"

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.GpioPinState#2 Pinstate

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.GpioPinState#2 Pinstate="l"

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.GpioPinState#2 Pinstate

sfr Devices#0.WinDriver#[tcp://172.16.1.254:93].LwrpInterpreter#0.LwrpRoot#0.Gpo#6.GpioPinState#2 Pinstate=”[ReadWrite=RW,SyntaxType=OPT,IsStable=True,SyntaxUsage=\”Single:l(Low),h(High)\”,IsSimpleUi=True]”

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
- IsSimplUI: 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 SyntaxTyp-e=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"

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 if multiple properties in the message exist, a match is made if either of them match or only if all of them match.

Warranty

Telos Alliance Limited Warranty

For the latest Telos Alliance warranty, visit: telosalliance.com/warranty

Telos Alliance • 1241 Superior Ave. • Cleveland, Ohio, 44114, USA • +1.216.241.7225 • TelosAlliance.com

1490-00079-003 --- USER MANUAL: PATHFINDER CORE PRO