

PathfinderPC Router Control Software

User's Guide



Software
AUTHORITY

9211 West Creek Road - Independence, Ohio 44131



August, 2008

Customer Service

We support you...

By Phone/Fax in the USA.

- Customer service is available from 9:30 AM to 6:00 PM USA Eastern Time, Monday through Friday at +1 216.241.7225. Fax: +1 216.241.4103.

By Phone/Fax in Europe.

- Service is available from Axia Europe in Germany at +49 81 61 42 467. Fax: +49 81 61 42 402.

By E-Mail.

- The address is **Support@AxiaAudio.com**.

Via World Wide Web.

- The Axia Web site has a variety of information which may be useful for product selection and support. The URL is **http://www.AxiaAudio.com**.

Feedback

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

Updates

Periodic updates of this software may become available. To determine if this is the case, please check our web site. Our electronic newsletter has announcements of major software updates for existing products, as well as keeping you up to date on the latest Axia, Telos, and Omnia product releases. To subscribe go to:

<http://www.axiaaudio.com/news/eNews.htm>

Trademarks

Telos Systems, Axia Audio, Path nderPC, Livewire, the Livewire Logo, the Axia logo, SmartSurface, SmartQ, Omnia, the Omnia logo, and the Telos logo, are trademarks of TLS Corporation. All other trademarks are the property of their respective holders.

Axia Audio

2101 Superior Ave. Cleveland, OH 44114 USA +1 (216) 241-7225
Inquiry@AxiaAudio.com

Axia Europe

Johannisstra@e 6, 85354 Freising, Germany +49 81 61 42 467
Inquiry@AxiaAudio.com

Copyright © 2007 by TLS Corporation. Published by Axia Audio. 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. All rights reserved.

Notice

All versions, claims of compatibility, trademarks, etc. of hardware and software products not made by Axia 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.

Warranty

This product is covered by a one year limited warranty, the full text of which is included in the rear section of this manual.

Service

When calling regarding PathfinderPC, we strongly recommend being near your computer when you call, so our Support Engineers can verify information about your computer, your configuration and the conditions under which the problem occurs. Have your serial number handy prior to calling.

About This Manual

This manual covers the details of the Axia PathfinderPC software for use with computers running the Windows® operating system. However it is assumed in this document that you are familiar with Livewire's basic concepts, as outlined in the companion Introduction to Livewire manual.

If you have not done so, please review that material first. In it we explain the ideas that motivated Livewire and how you can use and benefit from it, as well as nitty-gritty details about wiring, connectors, and the like. Since Livewire is built on standard networks, we also help you to understand general

network engineering so that you have the full background for Livewire's fundamentals. After reading Introduction to Livewire you will know what's up when you are speaking with gear vendors and the network guys that are often hanging around radio stations these days.

As always, we welcome your suggestions for improvement. Contact Axia Audio with your comments:

Axia Audio, a Telos Company

2101 Superior Avenue

Cleveland Ohio 44114 USA

Phone: +1.216.241.7225

Web: www.AxiaAudio.com

E-Mail: Inquiry@AxiaAudio.com

Table of Contents

OVERVIEW	10
SYSTEM REQUIREMENTS.....	12
PATHFINDERPC CLIENT APPLICATIONS.....	12
PATHFINDERPC/PRO SERVER	12
VERSION COMPARISON (PATHFINDERPC VERSUS PATHFINDERPRO).....	14
PATHFINDERPC/PRO SERVER INSTALLATION	16
PATHFINDERPC/PRO LICENSING	17
MAIN PATHFINDERPC SERVER CONTROL PANEL	17
OVERVIEW	17
ADDING/EDITING ROUTERS IN THE SYSTEM	18
<i>Axia Livewire Routing System</i>	20
<i>ZSystems/Videoquip/Sigma</i>	25
<i>Virtual Routers</i>	25
EDITING ROUTE POINT NAMES.....	26
PATHFINDERPC/PRO SERVER MENU ITEMS.....	27
FILE MENU.....	27
<i>Master TCP/IP Socket #</i>	27
<i>User Database and Password Mode</i>	28
<i>Email Alert Settings</i>	30
<i>Licensing</i>	31
PATHFINDERPC SERVER MENU ITEMS - VIEW MENU	31
<i>Script Debugger</i>	31
<i>Device List</i>	31
PATHFINDERPC SERVER MENU ITEMS - PREFERENCES MENU	32
<i>Silence Threshold</i>	32
<i>Clipping Threshold</i>	32
<i>Log Settings</i>	32
<i>Log to File</i>	32
<i>Log to TCP Port 5400</i>	32
<i>Log to UDP Syslog</i>	32
PATHFINDERPC SERVER MENU ITEMS - HELP MENU.....	32
<i>Help</i>	32
<i>About PathfinderPC Server</i>	32
SELECT LIST	33
STANDARD EVENT SYSTEM.....	34
TIME BASED EVENTS	34
AUDIO BASED EVENTS.....	36
GPIO EVENTS.....	38
STACKING EVENTS	40
STACK EVENT EXAMPLE 1 – GPI TRIGGERED ROUTE WITH EMAIL	40
ABOUT THE STACK LISTS SCREEN	43
MORE EXAMPLES	47
<i>Example 2 – Time/Date</i>	47
<i>Example 3 – Talkback Button</i>	48
<i>Endless Loops</i>	51

EXAMPLE 3 – LATCHING TALKBACK BUTTON.....	51
STACKEVENT QUALIFIERS	54
<i>RouteStatus</i>	54
<i>Audio Level Trigger</i>	54
<i>GPI State Change</i>	55
<i>GPO State Change</i>	56
<i>User Button Press</i>	57
<i>Time/Date Range</i>	58
<i>User Command Received</i>	59
<i>Memory Value</i>	60
<i>Event Engine Start</i>	61
<i>Element Profile Change</i>	61
<i>VMIX Change</i>	62
STACKEVENT ACTIONS	62
<i>Activate Route</i>	62
<i>Activate Scene</i>	63
<i>GPIO State Change</i>	63
<i>User Button Property</i>	64
<i>Send Email</i>	65
<i>Run Script</i>	65
<i>Send User Command</i>	66
<i>Memory Value</i>	67
<i>SA Command</i>	68
<i>Element Profile Change</i>	68
<i>VMIX Change</i>	69
<i>Execute Shell Command</i>	69
PROTOCOL TRANSLATORS	69
SUPPORTED TRANSLATORS	71
<i>Software Authority Protocol</i>	71
<i>Pro-Bel General Router and Pro-Bel General Switcher Protocol Translators</i>	71
<i>Generic Protocol</i>	71
<i>Sine ACU-1 Prophet Version</i>	71
<i>BTools Switcher</i>	74
SOFTWARE AUTHORITY BRIDGE APPLICATIONS	78
SOFTWARE AUTHORITY GENERIC BRIDGE APPLICATION.....	79
SOFTWARE AUTHORITY ACU-1 PROPHET VERSION BRIDGE APPLICATION.....	82
LOGGING	84
<i>Millisecond Counter</i>	85
<i>Date of Message</i>	85
<i>Time of Message</i>	85
<i>ServerIPAddress</i>	85
<i>Message Severity</i>	85
<i>Message ID Number</i>	85
<i>Message Source</i>	85
<i>Remote Device</i>	86
<i>Message Data</i>	86
LOGGING CONFIGURATION	86
<i>ConnectMessage Section</i>	89
<i>RouterDeviceReceivedCommand Section</i>	90
<i>RouterEvents Section</i>	91
<i>ServerActionSent</i>	92
<i>ServerEvents</i>	94
<i>ClientEvents</i>	98
LOG MESSAGE CUSTOM GROUP TYPES.....	100

Router Device.....	100
Stack Event.....	100
Event.....	100
Protocol Translator.....	100
Client.....	100
CLUSTERING.....	100
CLUSTER COMPUTER DESIGN CONSIDERATIONS.....	101
CREATING A CLUSTER.....	102
NOTES OF IMPORTANCE.....	107
PATHFINDERPC.....	109
PATHFINDERPC INSTALLATION.....	109
PATHFINDERPC MAIN APPLICATION.....	110
<i>Select List</i>	114
<i>Metering</i>	115
<i>Scenes</i>	116
<i>Virtual Routers</i>	117
<i>Search Functions</i>	122
<i>Scheduling</i>	123
<i>User Panel Menu</i>	124
PATHFINDERPC CLIENT REGISTRY SETTINGS.....	124
PANEL DESIGNER.....	126
CREATING A PANEL.....	126
<i>Arranging controls on the form</i>	127
<i>Setting Control Properties</i>	127
PROPERTY DESCRIPTIONS.....	131
<i>Panel Properties</i>	131
<i>Label Control Properties</i>	132
<i>Button Control Properties</i>	134
<i>Meter Control Properties</i>	135
<i>WebBrowser Control Properties</i>	135
PATHFINDERPC MINI.....	137
PATHFINDERPRO VMIX CONTROL.....	138
OVERVIEW.....	138
LICENSING.....	138
INSTALLATION.....	138
CONFIGURING VMIXCONTROL.....	139
USING VMIXCONTROL.....	142
<i>Using a Fader</i>	143
SAPORTROUTER.....	145
OVERVIEW.....	145
LICENSING.....	145
INSTALLATION.....	145
CONFIGURING SAPORTROUTER.....	146
<i>Creating Ports</i>	147
PORT TYPES.....	147
<i>Serial</i>	147
<i>TCPServer</i>	148
<i>TCPClient</i>	149
<i>GPIOPort</i>	150
COMPLETING THE CONFIGURATION.....	150
SERVICE STATE.....	151

ROUTING.....	152
<i>Multiplexing</i>	153
PATHFINDERPRO CONTROL.....	153
THIRD PARTY REMOTE CONTROL.....	153
<i>Remote Control Command List</i>	154
VISTA USERS.....	156
<u>DEMO LIMITATIONS AND LICENSING.....</u>	157
<u>CONTACT INFORMATION.....</u>	157
<u>APPENDIX A.....</u>	158
SAPROTOCOL - SOFTWARE AUTHORITY CONTROL PROTOCOL.....	158
<i>Overview</i>	158
<i>Command List – Client to PathfinderPC Server</i>	158
<i>Server To Client Commands Only</i>	166
<u>APPENDIX B - GETLIST FUNCTION FOR PROTOCOL TRANSLATOR.....</u>	168
OVERVIEW.....	168
SCENARIOS.....	168
<i>Nongoals</i>	168
<i>Usage Notes</i>	168
<i>Key</i>	169
<i>Syntax</i>	169
<i>Required Lists</i>	170
<i>GetListVersion</i>	176

A Note From The Founder/CEO of Telos

It's been a tradition since Telos' very first product, the Telos 10 digital phone system, that I share a few words with you at the beginning of each manual. So here goes.

In radio broadcast studios we're still picking up the pieces that have fallen out from the digital audio revolution. We're not using cart machines anymore because PCs are so clearly a better way to store and play audio. We're replacing our analog mixing consoles with digital ones and routing audio digitally. But we're still using decades-old analog or primitive digital methods to connect our gear. Livewire has been developed by Telos to provide a modern PC and computer network-oriented way to connect and distribute professional audio around a broadcast studio facility.

Your question may be, "Why Telos? Don't you guys make phone stuff?" Yes, we certainly do. But we've always been attracted to new and better ways to make things happen in radio facilities. And we've always looked for opportunities to make networks of all kinds work for broadcasters. When DSP was first possible, we used it to fix the ages-old phone hybrid problem. It was the first use of DSP in radio broadcasting. When ISDN and MP3 first happened, we saw the possibility to make a truly useful codec. We were the first to license and use MP3 and the first to incorporate ISDN into a codec. We were active in the early days of internet audio, and the first to use MP3 on the internet. Inventing and adapting new technologies for broadcast is what we've always been about. And we've always been marrying audio with networks. It's been our passion right from the start. In our genes, if you will. As a pioneer in broadcast digital audio and DSP, we've grown an R&D team with a lot of creative guys who are open-eyed to new ideas. So it's actually quite natural that we would be playing marriage broker to computer networks and studio audio.

What you get from this is nearly as hot as a couple on their wedding night: On one RJ-45, two-way multiple audio channels, sophisticated control and data capability, and built-in computer compatibility. You can use Livewire as a simple sound card replacement - an audio interface connecting to a PC with an RJ-45 cable. But



add an Ethernet switch and more interfaces to build a system with as many inputs and outputs as you want.

Audio may be routed directly from interface to interface or to other PCs, so you now have an audio routing system that does everything a traditional "mainframe" audio router does - but at a lot lower cost and with a lot more capability. Add real-time mixing/processing engines and control surfaces and you have a modern studio facility with many advantages over the old ways of doing things. OK, maybe this is not as thrilling as a wedding night - perhaps kissing your first lover is a better analogy. (By the way, and way off-topic, did you know that the person you were kissing was 72.8% water?)

While we're on the subject of history... you've probably been soldering XLRs for a long time, so you feel a bit, shall we say, "attached" to them. We understand. But no problem - you'll be needing them for microphones for a long while, so your withdrawal symptoms won't be serious. But your facility already has plenty of Ethernet and plenty of computers, so you probably already know your way around an RJ-45 as well. It's really not that strange to imagine live audio flowing over computer networks, and there's little question that you are going to be seeing a lot of it in the coming years.

The 20th century was remarkable for its tremendous innovation in machines of all kinds: power generators, heating and air conditioning, cars, airplanes, factory automation, radio, TV, computers. At the dawn of the 21st, it's clear that the ongoing digitization and networking of text, audio, and images will be a main technology story for decades to come, and an exciting ride for those of us fortunate to be in the thick of it.

Speaking of years, it has been a lot of them since I wrote the Zephyr manual intro, and even more since the Telos 10 - 20 years now. Amazing thing is, with all the change around us, I'm still here and Telos is still growing in new ways. As, no doubt, are you and your stations.

Steve Church

A Note From The President of Axia

20 years ago, I designed my first broadcast console for PR&E. I look back on that time with great fondness; we were building bullet-proof boards for the world's most prestigious broadcasters, making each new console design bigger and fancier to accommodate a wider variety of source equipment and programming styles. The console was the core of the studio; all other equipment was on the periphery.

Then things changed: the PC found its way into broadcast audio delivery and production. At first, PC audio applications were simple, used only by budget stations to reduce operating expenses. But soon the applications evolved and were embraced by larger stations. Slowly, the PC was taking center stage in the radio studio. Like many, I was captivated by the PC. Stations retired carts, phonographs, open-reel decks, cassettes — even more modern digital equipment such as DAT and CD players, replacing all with PC apps. Client/server systems emerged and entire facilities began using PCs to provide most - or all - of their recorded audio. Yet consoles continued to treat PCs as nothing more than audio peripherals. I knew that we console designers were going to have to rethink our designs to deal with computer-centric studios. During this time, traditional broadcast console companies began producing digital versions. But early digital consoles were nearly identical in form and function to their analog predecessors. It took a fresh look from a European company outside broadcasting to merge two products - audio routing switchers and broadcast consoles - into a central processing engine and attached control surface. Eventually nearly every console and routing switcher company followed suit, and a wide variety of digital “engines” and control surfaces flooded the market.

But, advanced as these integrated systems were, they still handled computer-based audio sources like their analog ancestors. Sure, the router and console engine were now integrated, but the most important studio element - the PC - was stuck in the past, interfaced with 100-year-old analog technology. The PC and console couldn't communicate in a meaningful way - strange,

considering that PCs everywhere were being networked, fast becoming the world's most popular and powerful communication tool.

Then a group of Telos engineers developed a method of using Ethernet to network real-time audio devices, allowing computers and consoles, controllers and peripherals to interact smoothly and intelligently. Powerful, flexible networks had finally come to our studios. As with the transition from carts to computers, the benefits are many and impressive. A few networked components can replace routing switchers, consoles, processing peripherals, sound cards, distribution amps, selector switches and myriad related devices.

This deceptively simple networked system costs a fraction of other approaches, yet has capabilities surpassing anything else. The system is modular and can be used to perform discrete functions in a traditional environment. Concurrently, it easily scales to serve both

the humblest and the very largest of facilities. Console, router, and computer work in harmony.

So, equipped with this new technology and countless ideas, we launch *Axia*, the newest division of Telos. *Axia* is all about delivering innovative networked audio products to

future-minded broadcasters. On behalf of our entire team, I welcome you as a charter client. *Axia* is the culmination of nearly 40 man-years of some of the most ambitious R&D ever applied to the radio industry. And this is only the beginning. We have more products, innovations, and partnerships in the pipeline.

You already know your *Axia* system is unlike anything else. So it shouldn't be surprising that your new system is loaded with new thinking, new approaches, and new ideas in virtually every conceivable area. Some concepts will challenge your traditional ideas of studio audio systems, but we're certain that once you have experienced the pleasures of the networked studio, you'll never want to go back. And now, for something completely different...

Michael “Catfish” Dosch



Overview

The **PathFinderPC** software is a Client/Server Router control software package. It is designed to provide facility-wide control over any number of Audio, Video, and Machine Control routers including the Axia Livewire distributed routing system. At the time of this writing, the package supports the Axia Livewire products, ZSystems 64X64 and 128X128 Detanglers, the VideoQuip DS1224 machine control router, NTI VGA Router, and the Sigma Series of routers. In addition, we also have a Software Authority Port Routing application that can be controlled by PathfinderPC Server which allows Serial to TCP conversion and routing.

The software consists of a server product called PathFinderPC Server that communicates via Serial port or Ethernet with the routers, and a client application called PathfinderPC Client which presents the graphical user interface for making routing changes. The client application connects to the server using the TCP/IP protocol over any established LAN or even over the Internet. The server application then makes the client requested changes to any of the routers in the routing system and provides updates to the client application interface. The PathfinderPC package also includes a tool called Panel Designer and an additional application called PathfinderPC Mini. The Panel Designer tool allows a designer to create custom software routing panels. These panels are then available in the PathfinderPC Client application for end user use. The PathfinderPC Mini application is used in a situation where the only controls the end user should access are those available through one of the custom designed panels. An administrator defines which panel will be displayed by any given installation of PathfinderPC Mini through a small configuration file, and then the mini panel is the only access to the PathfinderPC system that will be available to that user.

The PathfinderPC Server system is configured using intuitive wizards to inform the system about routers that are connected to the system, and to modify the names of route points in the system. PathfinderPC Server also includes the following additional features:

1. User database which an Administrator may use to control access to certain functions and routing lists in the system
2. An integrated event system to provide system and routing changes at specific times or based on silence detection and/or GPIO triggers
3. A more advanced Stacking Event System which allows an Administrator to create very customized event logic
4. A Protocol Translator to provide access to the routing system by automation systems and other controllers which use different protocols
5. An advanced logging system to log events throughout the routing system
6. Serves Custom User Panel definitions to Client applications
7. Provides a scripting engine for custom scripts if needed
8. Clustering so that two servers can run the PathfinderPC Server application and provide failover if one of the servers fails (requires two server licenses)

PathfinderPC Client is the application that end users will use to make route changes. The interface is simple to use and allows source and destination route changes to be made quickly and easily with a few mouse clicks. This client application also includes the following features:

1. Three different routing views and methods for viewing and making route changes
2. Simple interface for creating and activating entire scene changes to quickly change multiple routes in the system
3. Ability to lock routes so they may not be changed by another user during a show
4. Search engine for finding sources and destinations within large route tables
5. Resource sorting
6. Audio Absent, Present, and clipping metering
7. Full Metering
8. GPIO viewing and control
9. Virtual Router Creation to create custom route lists for a particular room and/or to tie points together from different kinds of routers (Audio/Video/Machine Control mapped as single point)
10. Event Programming (duplicate interface as server for creating standard events)
11. User Panel Viewing and manipulation

Panel Designer is a graphical tool to create custom user panels. It allows the user to drag and drop controls onto a panel and then assign colors, pictures, and textual properties to them. These controls can then be made to activate routes or scene changes or to trigger more complex tasks through Stacking Events. The software buttons can also be mapped to either GPIO hardware buttons or Element User Panel buttons. These hardware buttons will then perform the same tasks as the software buttons.

PathfinderPC Mini is an application that will run and display a single Custom User panel designed by the Panel Designer. This should be used on computers where the Administrator does not want the end user to have all of the power and options of the full PathfinderPC Client application. The Administrator can design a custom user panel for the end user that provides access to the few needed resources. PathfinderPC Mini is then used to display that panel on the end user's computer.

The PathfinderPC software suite has also been augmented with two different software bridge products. These bridges convert serial port commands to TCP commands for controlling devices that need to use the PathfinderPC Server protocol translator system and need to do that in a cluster-aware manner. These applications are called Software Authority Generic Bridge, and Software Authority ACU-1 Prophet Version Bridge, and are described in detail later in this document.

In addition PathfinderPro users will be able to use two additional applications, VMIXControl and SAPortRouter. The VMIXControl application provides an on-screen mixing application which controls the 40 Channel virtual mixer in each Axia Mix Engine. An Administrator can define which Mix engine and VMIX faders the user may use, and basic mixing can be accomplished on-screen.

The SAPortRouter application is a serial data routing application which allows the user to dynamically route textual data between serial and TCP ports. It runs as a service and allows a user to create Serial, TCP Client, TCP Server, and GPO driver ports. Data from any of these ports can then be routed to any of the other ports. This is known as the Software Authority Port Router application.

All of these PathfinderPC tools combine to form an incredibly powerful, facility-wide routing control software suite.

System Requirements

PathfinderPC Client Applications

The PathfinderPC Client, PathfinderPC Mini, Panel Designer, SApPortRouter, VMIXControl, and the bridge application programs will all run on Windows 2000, XP, 2000 Server, and 2003 Server. Minimum hardware system requirements for these operating systems are also acceptable to run these applications. In addition, at this point in time the operating system must be running with an English code-base. While some other language code-bases may work properly, the current protocol between the client and server relies on characters in the higher Ascii range that are reserved in some language code-bases.

In addition the SApPortRouter and VMIXControl applications require .net Framework 2.0. This is usually installed automatically on current versions of Windows XP and above. You can also obtain this framework for free through the Microsoft Windows Update system.

PathfinderPC/Pro Server

PathfinderPRO Server will also run under Windows 2000, XP, 2000 Server, and 2003 Server. However, there are some limitations that need to be considered when choosing an operating system for PathfinderPC Server. All Microsoft workstation operating systems such as Windows 2000 and Windows XP have a limit as to the number of simultaneous incoming connections they will accept. Therefore, if you are going to have more than 10 clients (PathfinderPC Client, PathfinderPC Mini) accessing the PathfinderPC Server application at the same time, you must use a Server operating system such as Windows 2000 server or Windows 2003 Server.

In addition, Microsoft XP Service Pack 2 added a security feature that limits the number of outgoing connections to devices that can occur in any short period of time. This means in an Axia environment, if the server application needs to connect to more than 6 Axia devices on the network, and there is any chance that those devices might not be reachable, XP should not be used for the PathfinderPC Server operating system.

Due to these restrictions, it is highly recommended that either Microsoft 2000 Server or Microsoft 2003 Server is used as the host operating system for PathfinderPC Server. It is also recommended that the machine have at least 512 MBytes of RAM.

If the clustering option is to be used, at least 2 NIC cards should be used, and we recommend using 4. See the section in this document on clustering for details.

PathfinderPC Server (as opposed to PRO) is restricted in the number of clients it can accept and the number of devices it can control. Because of these restrictions, this version will run adequately on an XP or workstation operating system. It is designed for smaller systems.

Finally, at this point in time the operating system must be running with an English code-base for either PathfinderPC Server or PathfinderPro server. While some other language code-bases may work properly, the current protocol that we use between the client and server relies on characters in the higher Ascii range that are reserved in some language code-bases.


Version Comparison (PathfinderPC versus PathfinderPRO)

PathfinderPC now ships in 2 different versions, PathfinderPC and PathfinderPRO. PathfinderPC is designed for smaller installs and comes with a single server license. PathfinderPro is designed for larger installations, ships with 2 Server licenses so that it is immediately ready for clustering, and has several additional features and programs. Here is a complete feature set comparison:

Feature	PathfinderPC	PathfinderPCPro	PathfinderPCPro Upgrade
Server Licenses	1	2	1 Additional + upgrade of existing
Server OS Required	XP/2003 Server	2003 Server	2003 Server
Client Connections (Client/Mini)	10	No limit other than CPU speed	No limit other than CPU speed
Axia Devices it will Control	25	No limit other than CPU speed	No limit other than CPU speed
Clustering		X	X
Routers Supported			
Axia Audio	X	X	X
Axia GPIO	X	X	X
Virtual	X	X	X
SA Port Router		X	X
VideoQuip DS 1224	X	X	X
ZSystems Detangler	X	X	X
Sigma	X	X	X
NTIVeemux VGA	X	X	X
Protocol Translators			
Software Authority Protocol	X	X	X
Generic Protocol	X	X	X
Pro-Bel General Router		X	X
Pro-Bel General Switcher		X	X
Sine ACU-1		X	X
BTools		X	X
Standard Events	X	X	X
Time Based Route Change	X	X	X
Time Based Scene Change	X	X	X
GPIO Based Route Change	X	X	X
GPIO Based Scene Change	X	X	X
Silence Detect Backup Route	X	X	X
Stack Events			

Detect/Activate Routes	X	X	X
Activate Scenes	X	X	X
Audio Level/Silence Detect	X	X	X
Detect/Activate GPIO	X	X	X
Software Panel Button Control	X	X	X
LCD Panel Button Control	X	X	X
Element Fader On/Off	X	X	X
Element Motorized Fader Control		3 rd Quarter 2008	3 rd Quarter 2008
Time Date Range	X	X	X
User Command Send/Receive (Generic Translator)	X	X	X
Memory Values	X	X	X
Event Engine Start	X	X	X
Element Profile Change	X	X	X
VMIX Control		X	X
Send Email	X	X	X
Run Script	X	X	X
SA Command	X	X	X
Shell Command		X	X
Node Gain Control	3 rd Quarter 2008	3 rd Quarter 2008	3 rd Quarter 2008
Panels (Mini, Client, Panel Designer)			
Buttons	X	X	X
Labels	X	X	X
Web Browser (Supports video)		X	X
Meters		X	X
Faders for VMIX/Motorized/Node Gain		4th Quarter 2008	4th Quarter 2008
SA Generic Bridge	X	X	X
SA ACU Bridge		X	X
VMIX Control Mixer Application		X	X
SA Port Router Service		X	X

PathFinderPC/PRO Server Installation

1) Run the PathfinderPC Server setup application to install the PathfinderPC Server Software.	
2) Click Next to begin the installation.	
3) Select I Agree to the license agreement and the Select Next to continue.	
4) Select the folder where PathfinderPC Server should be installed. Click Next to complete the installation.	

After the PathfinderPC Server software is installed, run it for the first time. The software will automatically display the Main Server Control Panel and start the Add Router wizard.



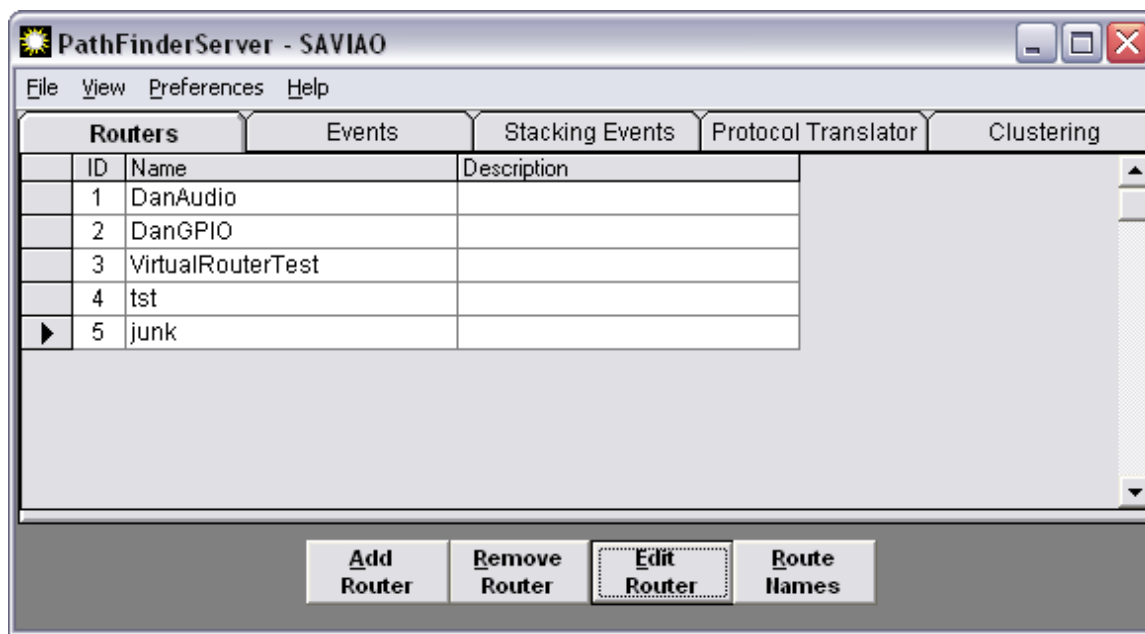
Please Note:

The server computer must have the appropriate hardware connections to communicate with all routers which are to be included in the system. For example, if the routers are to be controlled serially, the computer must have a serial port for each router to be controlled. Furthermore, the port must have the correct protocol. Both the Z-Systems Detangler and the VideoQuip routers, for example, require RS422 control. Most PCs have RS232 ports, thus an RS232 to RS422 adapter is required. The Axia line of products uses Ethernet connections and therefore the Ethernet connection must have access to the logical LAN on which the Axia products are installed. Feel free to contact Software Authority or Axia technical support for advice on products to accomplish these goals.

PathfinderPC/Pro Licensing

When you first launch PathfinderPC Server, a license window will appear. At the top of the license window is a link which will take you to the Pathfinder licensing site. If you have not created an account, click the create account button on the web page and use your request code from the CD packaging to generate an account and the correct license key. Then copy and paste the request code and license key into the software. Your copy of PathfinderPC Server will then be active.

Main PathfinderPC Server Control Panel



Overview

This panel provides access to all of the configuration menus and wizards necessary to design and configure the PathfinderPC Routing System. Routers may be added or removed from the server using the buttons on the main screen. The Edit Router button allows the Administrator to edit the communication parameters for the router whereas the Route Names button allows the user to edit the names of the individual routes within each router in the system. The events tab provides access to the event system to add and remove system events. The Stacking Events tab provides access to the Stacking Events Engine to create custom user defined events. The Protocol Translator tab lists the protocol translators that are active on the system and what form of communication PathfinderPC Server will use (Serial or TCP/IP) to answer automation requests under a given protocol. Finally, the clustering tab shows the list of active PathfinderPC Server computers involved in the cluster.

Under the File Menu an Administrator can control several options. First the Master TCP/IP Winsock port may be changed. This should only be changed if the port conflicts with another application. If the default port is changed on the server, it will also have to be changed on all instances of the client. The File menu also provides

access to the User Database. See the section later in this help document on creating users and modifying their rights on the system.

Next there is an option for setting email server settings for email alerts. Finally, the File menu lets the end user determine whether a username and password is needed for the client application to log into the server. If this option is not turned on, all copies of the client application will log in using full Administrator privileges. Thus it is highly recommended that the password mode be set to Manual login required, and the user database will then be used to authenticate client access. There is also a menu item to manage licensing so that upgrade licenses may be applied or license information retrieved.

The WebServer option is not complete at this point in time. In the future, it will provide a browser-based client interface for computers that can not run the PathfinderPC Client application.

The view menu provides a script debugger to which debug and error messages may be sent when writing and testing custom scripts. It also provides access to a device list. In an Axia network this will show a list of all of the devices on the network that the PathfinderPC Server knows about along with the type of device, its IP address, and links to the device's configuration page.

Under the Preferences menu, the silence and clipping threshold may be set. This is the audio threshold which will trigger either a clipping indication or a silence state in the system. This menu also provides access to the logging system in PathfinderPC Server.


Adding/Editing Routers in the system

In order to use the PathfinderPC system, the PathfinderPC Server Administrator must add the routers to the system by defining the router model and communications settings. With most of the routers PathfinderPC supports, this is just a simple task of selecting the router model number from a drop down list and defining the serial port or TCP settings to communicate with the router. This is true for the Z-Systems Detanglers, Videoquip DS1224, NTI VGA Router, and the Sigma routing equipment.

The Axia Livewire system, however, is different from most routers and requires a little more discussion. The Axia Livewire routing system uses many devices spread across a facility and all interconnected through a standard central Ethernet Switch. Thus, when we discuss an Axia Livewire Router in PathfinderPC Server, we are really talking about a network of Axia equipment that makes up a single logical router. In most Axia networks there will usually be two main logical routers created within PathfinderPC Server. One will encompass all of the audio points across the Axia Network, and the other will encompass all of the GPIO points. Obviously, it is not possible to route a GPIO point to an Audio point so they are treated as two different routers within the system.

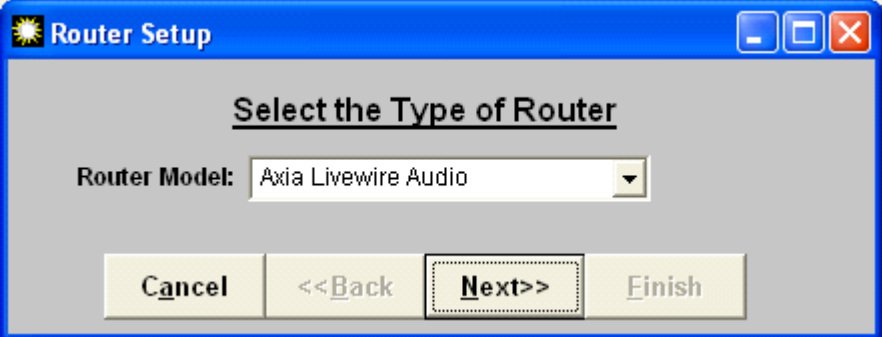
All of the routers listed above will be considered throughout this document as Real routers. Real routers are to be differentiated from virtual routers. A real router is defined as a router in the PathfinderPC System that directly communicates with the physical routing hardware in the system. Virtual routers on the other hand are a special kind of router that only exists within PathfinderPC. They are used to create subsets and supersets of the physical route points within the system. For example, a virtual router could be used to display only a select number of route points in an overall system that pertains to a specific room. The Administrator could define a router for a specific studio and map only the points on the master router to which the specific room should have access.

Virtual routers can also be used to marry points from different kinds of routers together. For example, the PathfinderPC administrator could create a source in a virtual router that has a source from a real audio router, a source from a real GPIO router, and a source from a real video router as a single virtual machine source. The same can be done on the destination side. Then when a route on the virtual router is made, audio, video, and GPIO will all travel together. This is much like the level concept some routers use except that it is totally dynamic in the way it can be defined by the PathfinderPC Administrator.

<p>1) To begin setting up your PathfinderPC system click on the Add Router button on the Server.</p>	
<p>2) Select the Type of Router to add to the system. The drop down list displays a list of the routers which are currently supported by PathfinderPC. The server can provide control for any of these routers and can even provide access for multiple types of routers at the same time.</p>	
<p>3) Click “Next” to be presented with a screen to enter a name and description for the router</p>	
<p>4) Click “Next” after entering the name and description by which the logical router will be known within the PathfinderPC system and you will be presented with a series of screens that will be different depending on the kind of router you selected in the previous step to create.</p>	



Please Note:

<p>If you have just started the server application for the first time, the screen below may appear automatically without clicking the Add Router button prompting you to create your first router.</p>	
--	--

The following sections of this document discuss the options for setting up the different kinds of routers PathfinderPC supports.

Axia Livewire Routing System

The Axia Livewire routing system is unique in the world of routers in that it is a distributed system. This means instead of all of the facility's audio wires coming to a single router in a central location, the audio wires can be wired to individual Livewire terminals in the separate rooms. These Livewire terminals then each connect to a central Ethernet Switch with a single Ethernet cable for each unit. As a routing system, instead of the PathfinderPC Server communicating to one central unit, the server knows about all of the individual terminals on the LAN and monitors each of them, presenting the client application with the appearance of a single router. In order to accomplish this, each Livewire device must have a unique IP address.

Before setting up the Axia system within Pathfinder, it is important to be sure that all Axia Devices are turned on and connected to the Switch, and that the PC which runs PathfinderPC Server is connected to that switch as well. It is also important that each Axia device in the system has a unique IP Address and name. If you purchased your Axia System from a distributor or Systems Integrator that preconfigured your nodes, then this has already been accomplished. If IP addresses and names have not already been programmed into the Terminals, PathfinderPC Server includes a tool to aid in this process. To access the tool, select the Utilities Menu from the Axia Livewire setup screen and select IP Address Helper. If this step has already been accomplished, skip to the section on Axia Livewire Configuration. Please note that at this point in time due to changes and additions to the Axia product line this tool is somewhat limited in its capabilities. It is recommended that the initial IP address and Channel assignment configuration be accomplished through each device's web page interface. The IP address helper tool below only works with Axia nodes at this point in time. It will not automatically configure Axia Elements, Engines, or Drivers. These must be initially configured through their own interfaces.

Assigning IP Addresses

If the system terminals have been pre-configured, please skip to the section on adding Axia devices to the routing table. To access the IP Address Helper tool, select the Utilities Menu from the Axia Livewire setup screen and select IP Address Helper. While the IP address helper tool below can be useful, it is often recommended to set the nodes up directly through their web pages. That option will work as well, and is the only option for configuring the Element CPUs and Engines.

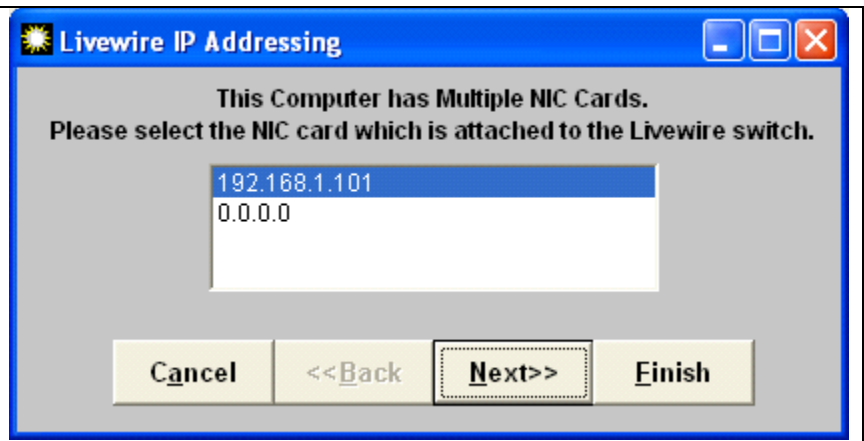


Please Note:

If the server computer has multiple NIC cards installed, the following screen will be presented.

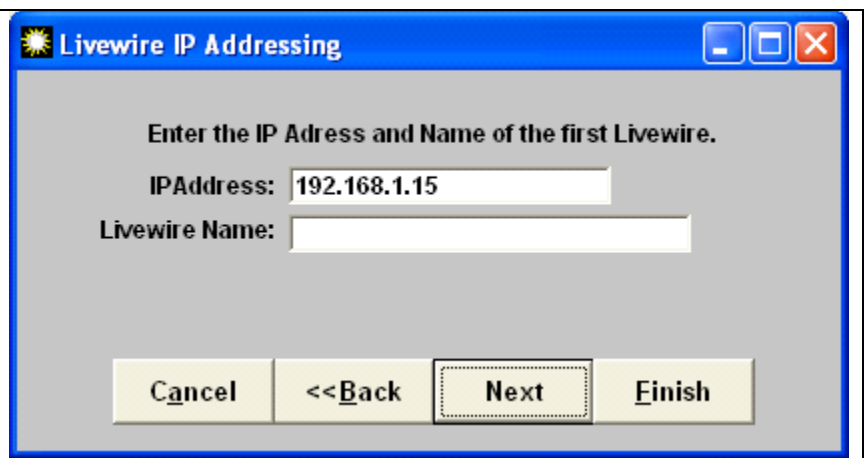
Select the IP Address of the NIC card that is connected to the switch which has the Livewire units attached to it.

Click Next.



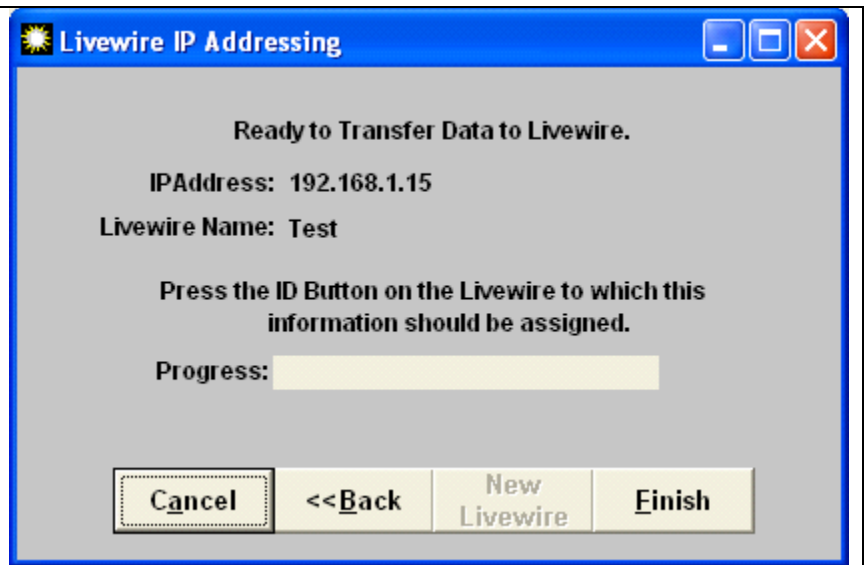
Enter the IP Address and unique name for the first Livewire in the system.

Click Next.



The server is now ready to program the first unit and is listening for a request from the unit. Press the ID button on the unit to which you want this IP Address and Name to be assigned. The server should hear the request and program the unit. In order to accomplish some of the programming, the software needs to know the password of the unit. By default it displays the factory default password. If you have changed this password already, you will have to provide the actual password in order to complete the programming. Finally, the software will offer to change the password to a new one if you so desire.

When the Progress Bar shows programming complete, you have the option to **click New Livewire** to program another unit, or **Finish** to finish assigning addressing to the livewire terminals.

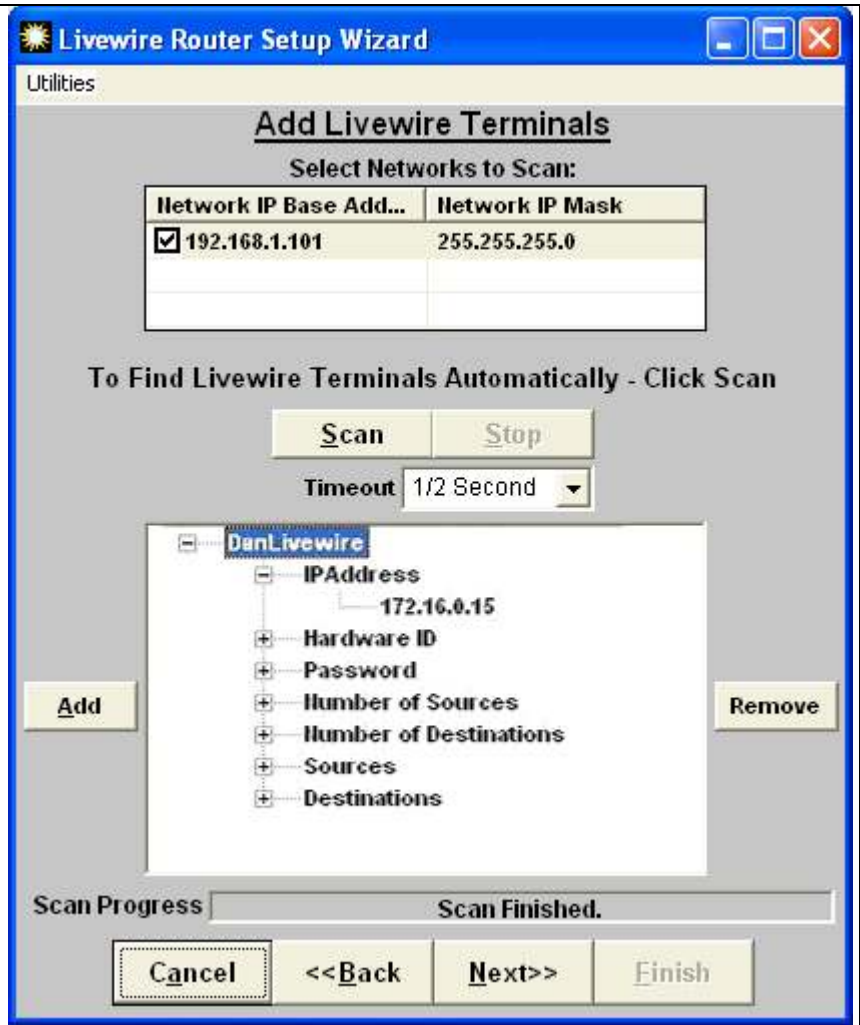


Once the IP addresses and names have been assigned to all terminals, the terminals should be rebooted to make sure the changes are stored properly.

Adding Axia Devices to the Routing Table

Once the Axia Livewire type of router has been selected and a name and description has been assigned to the router, the following wizard screen will be presented.

This screen is used to scan for Axia Devices on the LAN of the particular kind (Audio or GPIO) selected in the first step of the router addition process. Select the IP address of the Network Card on your computer that is connected to the Axia switch. Then **click Scan**. The system will spend 30 seconds listening for network broadcasts of Axia equipment on the LAN. It will then try to access and log into each of those devices to determine its capabilities and route lists.



The devices that are found that match the router type you are adding will be displayed in the lower window. When the scan is complete, you should see all Axia Devices that have resources that match the router type you are adding. For example, if you are adding an Axia Audio router to the PathfinderPC System, you should see all Axia devices that have audio resources and route points on the network in the lower window. If you are adding a GPIO router, you will see all GPIO devices and points. You may also manually add and remove terminals by clicking Add and entering the IP Address of the Axia device, or clicking on the device in the list and selecting Remove.

When you are finished adding all of the Livewire terminals, click Next.

Please Note:

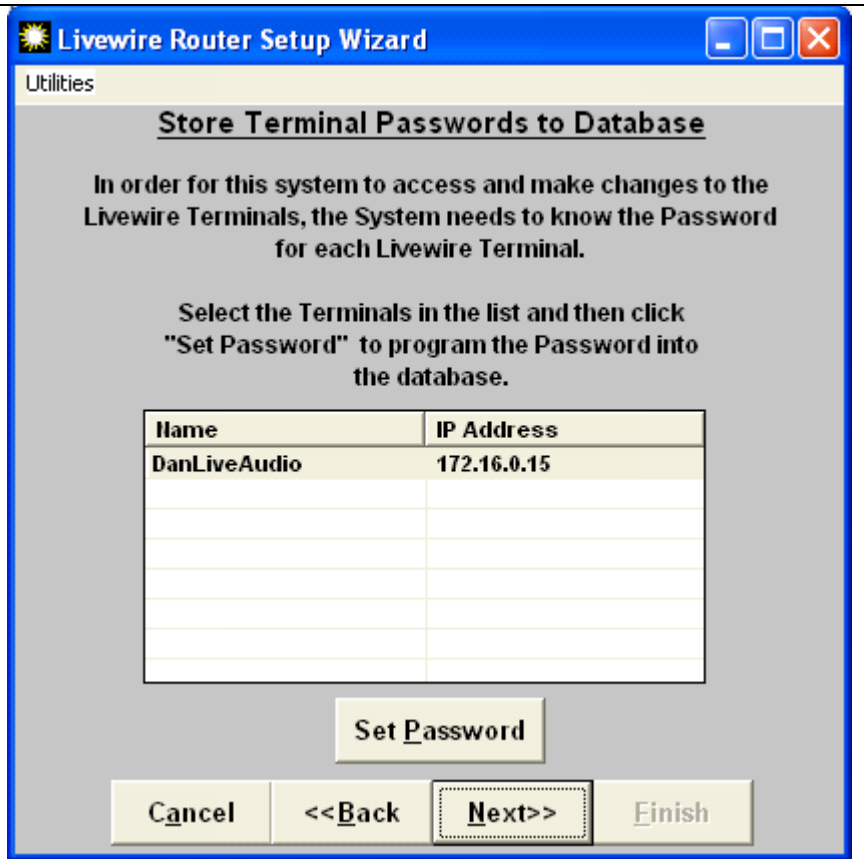
If you have not changed the password in any of the Axia devices you may skip this step.

In order for the server to control the Axia devices, it must know the access password for each Livewire Terminal in the system.

Select any number of the Livewire terminals in the list (so that the lines turn blue), and click Set Password.

This is not changing the password in the unit, but just informing the database as to what password each unit needs for access.

Click Next when all passwords have been assigned.



At this point the system will attempt to log in to each Axia device in the System using the defined password. If it fails to log into any of the units, you will see False in the Login Success column for that unit. You may click Back to correct the problem.

Once all terminals have the correct password programmed into the database, and the system can correctly log into each terminal,

Click Finish to finish adding the Axia Livewire router to the Router table. You may then **click on the Route Names button** on the main server page to edit the individual names of each router point in the system as described

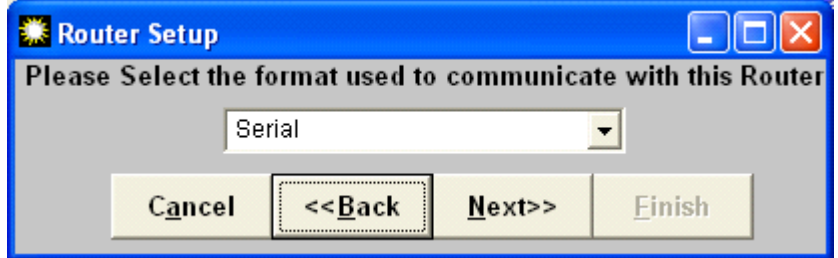


later in this document.

ZSystems/Videoquip/Sigma

First select the connection type. In almost all cases for these routers the connection type will be Serial. The only time TCP/IP would be used for Sigma, Videoquip, or Z-Systems routers would be if you are using an Ethernet IP to Serial converter.

Click Next.



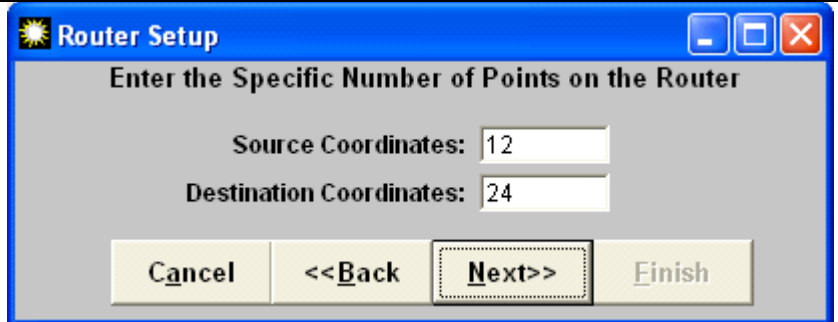
Next verify that the serial settings are correct for the given router and select the serial port on the computer that is connected to the router.

Click Next.

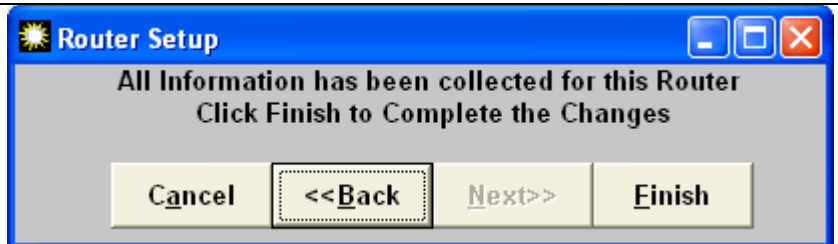


Verify that the number of sources and destinations is correct for the given router.

Click Next again.



Finally **click Finish** to add the router into the system.



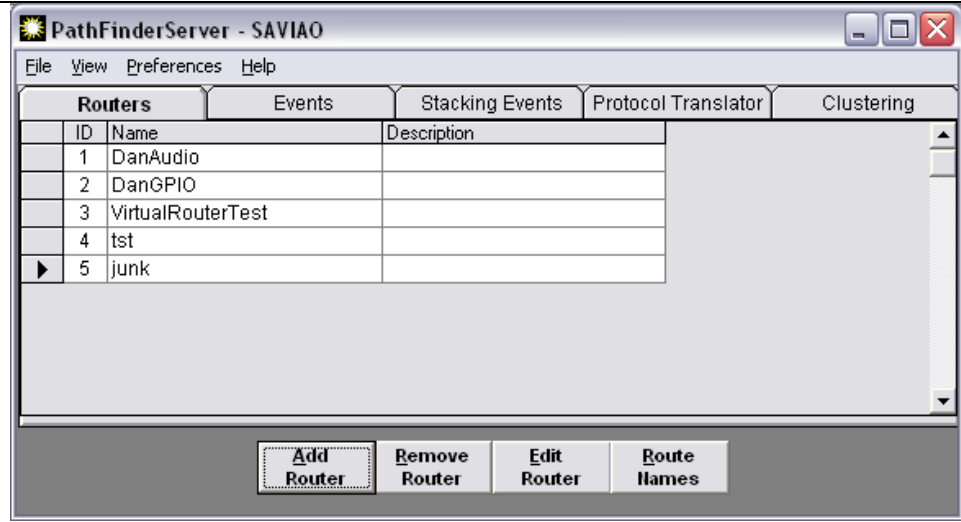
Virtual Routers

The Virtual Router Model is a unique model. It allows subsets and supersets of real routers to be created. This will be discussed in detail in the PathfinderPC client section of this document. The Virtual router is the only router that may be created and edited using the PathFinderPC Client software. All real routers should be defined within the server first.

Editing Route Point Names

To edit the names of source and destination points in the PathfinderPC system from the routers tab,

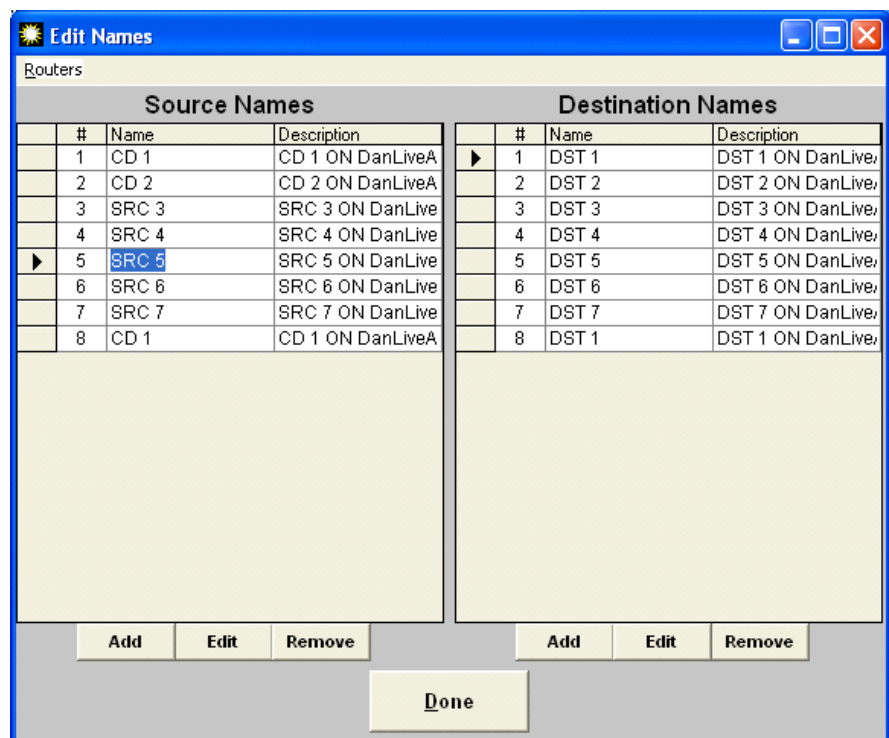
Select the router in the list and then click the Route Names button.



Source and Destination names and descriptions can be edited by clicking on the particular name or description and simply editing the text. **Click Done** when you are finished with all changes.

On an Axia router these names are actually retrieved from the Axia equipment itself, and changing the names in this table will push the new names out to the Axia equipment when you **click done**. This provides an easy place to modify source and destination names for all of your Axia equipment throughout the network.

Please Note: Engine names may not be modified by Pathfinder so changes on these points will not get published to the Engine.



If the router being edited is an Axia Livewire router, the Add, Edit, and Remove buttons will also appear.

By pressing the Add, Edit, and Remove buttons, this screen allows the end user to manually add route points and terminals into the system which were not automatically detected. Only use these manual add and edit buttons if you are knowledgeable about the inner workings of your Axia Livewire system.

Add/Edit Livewire Source

Index

Name

Standard Stream Address

(Livewire Terminals Only - Leave Blank for Non-Livewire PC Streams)

Terminal Control IP

Terminal Name

Terminal Password

IO Number on Terminal

Total Sources on Terminal

Total Destinations on Terminal

Livewire Stream Address

Done Cancel

PathfinderPC/PRO Server Menu Items

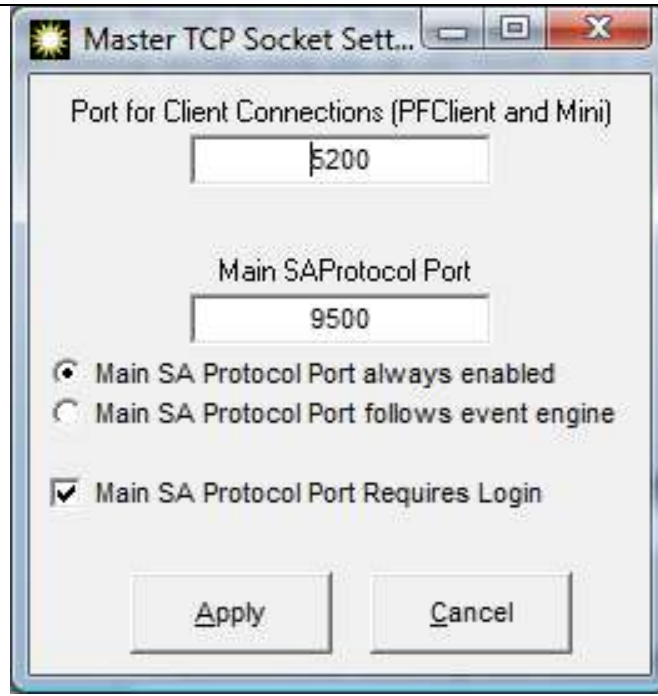
File Menu

Master TCP/IP Socket #

This menu item allows an Administrator to change the TCP socket number on which PathfinderPC listens for client connections. This should only be changed by an Administrator knowledgeable in TCP/IP when the computer is already using the default 5200 port for another application. If this port number is changed, it must also be changed in all client applications which attempt to connect to PathfinderPC Server.

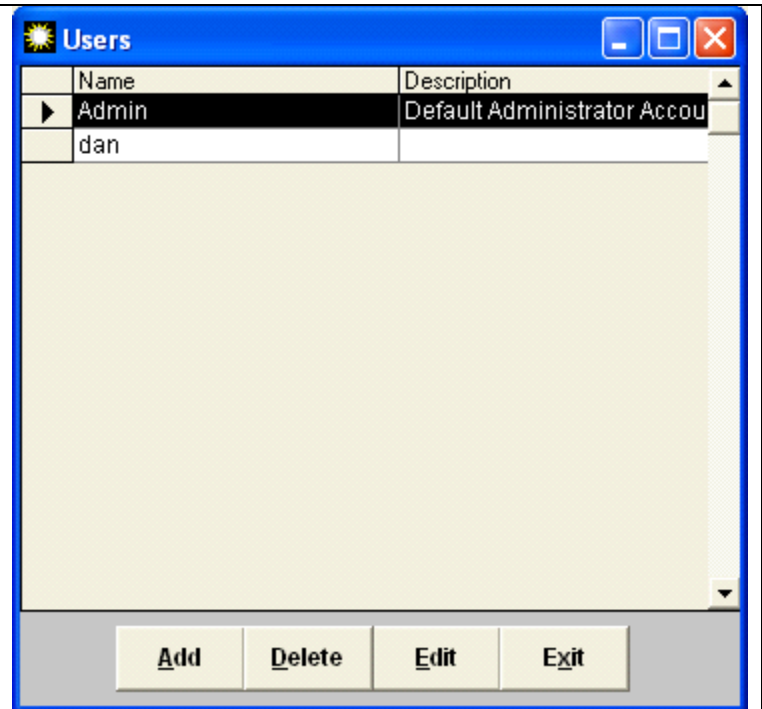
In addition, there is a master control port

(default 9500) which third party applications can use to control Pathfinder using the Software Authority Protocol. Define whether the port requires a Login and whether it is always enabled, or only enabled on the server that owns the event system.



User Database and Password Mode

The User database may be edited by selecting it from the File Menu of the PathfinderPC Server Main Control Panel.



Use the buttons at the bottom of the screen to Add, Delete, or Edit users.

Each user can either be an Administrator or a General user. Administrators have full privileges on the system for all routers. If General User is selected, the screen will expand to show the limitations available for general users.

Edit User

User Name: Admin

Password: *****

Confirm Password: *****

Description: Default Administrator Account

Administrator
 General User

Commit **Cancel**

The Administrator who sets up user accounts may define which routers in the system a general user may access and whether they have the right to override lock conditions on route points. In addition, only Administrators (not general users) are allowed to create or modify virtual routers.

In as normal operating environment, all routers will be set up on the system. The Administrator will then log into PathfinderPC client and create virtual routers which contain only the route points that should be available to general users in specific rooms or for specific shows. Finally, the Administrator will create the accounts for these general users and only allow them to have access to these virtual routers. This will limit the general user's ability to change critical routes in the system. In order for the User database to be used the menu item Password Mode – Manual Login Required under

Edit User

User Name: Peter

Password: *****

Confirm Password: *****

Description: ML Show producer

Administrator
 General User

User may NOT access:

#	RouterName

User may Access:

#	RouterName	Lock
1	Axia Audio	Yes
2	MachineControl	Yes
7	TestVirtual	Yes

Can modify Locks

Commit **Cancel**

the File menu must be selected.



Please Note:

Using the user database is a powerful tool for limiting a user's rights on the system. However, some Administrator's would prefer that their users not have to use a login user name and password each time they open up a PathfinderPC Client. In that situation, select the Password Mode – No Password required menu item in the File menu. Then there are several registry entries that may be used on the PathfinderPC client workstation to limit the client application's access to the system instead. See the registry settings section in the PathfinderPC section of this document for details.

Email Alert Settings

This menu item opens a screen where the Administrator can define the email server settings required for the PathfinderPC system to send email alerts.

These settings are similar to the ones required by most email client applications to send mail through your corporate email server. Contact your email server administrator for the correct settings for this window.

Email Master Settings

Enter the Email Server Settings to send Alert Emails
(These settings are the same you as you would use in your mail client application such as Outlook Express to send email)

Email Server
mail.comrec.com
(Example: mail.yourdomain.com)

Email Server Port (SMTP)
25

Email User Account to use to send mail
(Example: henry)

Password for Sending user account
(Example: myEmailPassword)

From Address
dbays@comrec.com
(Example: henry@yourdomain.com)

Test **OK** **Cancel**

Licensing

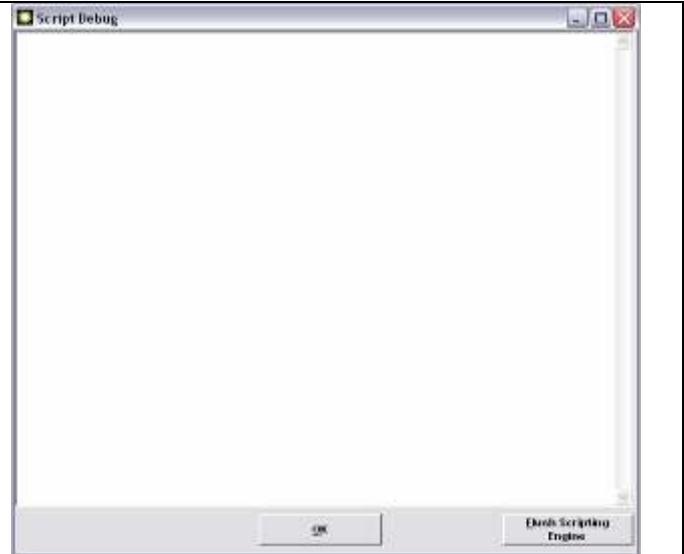
This menu item will display the licensing screen. This screen will show your current license and license type, as well as provide a link to the PathfinderPC license activation web site.



PathfinderPC Server Menu Items - View Menu

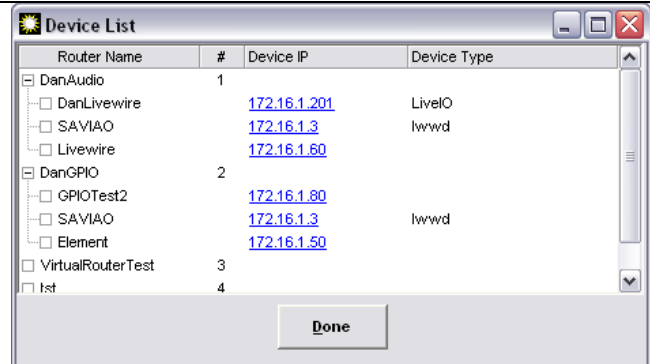
Script Debugger

The script debugger window is used when designing custom PathfinderPC scripts to view error and debugging messages.

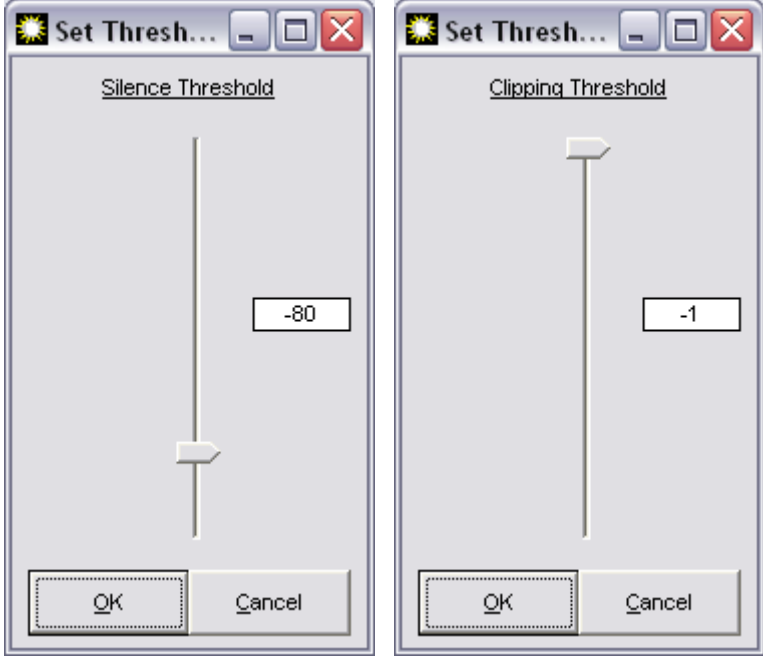


Device List


The device list will display all devices in a particular router. For an Axia Router, this will display all of the Axia devices along with their IP address and device type that are incorporated in the particular PathfinderPC Real Router. Clicking on the blue IP address link for any of these devices will open the device's web configuration page. Also, any device that does not have the device type listed within the first 60 seconds after the server launches should be tested for proper operation, as this is usually a signal that PathfinderPC Server cannot contact it properly. The logging and log files if configured can also supply more information.



PathfinderPC Server Menu Items - Preferences Menu

<p><u>Silence Threshold</u> The silence threshold sets the audio level threshold that will be used for any silence detection events. It also defines the threshold that will cause the metering bubbles in the client to display green for Audio Present.</p>	
<p><u>Clipping Threshold</u> The clipping threshold sets the audio level threshold that will be used to determine if clipping is occurring in the system.</p>	
<p><u>Log Settings</u></p>	<p>See the Logging section of this document for an explanation</p>
<p><u>Log to File</u></p>	<p>See the Logging section of this document for an explanation</p>
<p><u>Log to TCP Port 5400</u></p>	<p>See the Logging section of this document for an explanation</p>
<p><u>Log to UDP Syslog</u></p>	<p>See the Logging section of this document for an explanation</p>

PathfinderPC Server Menu Items - Help Menu

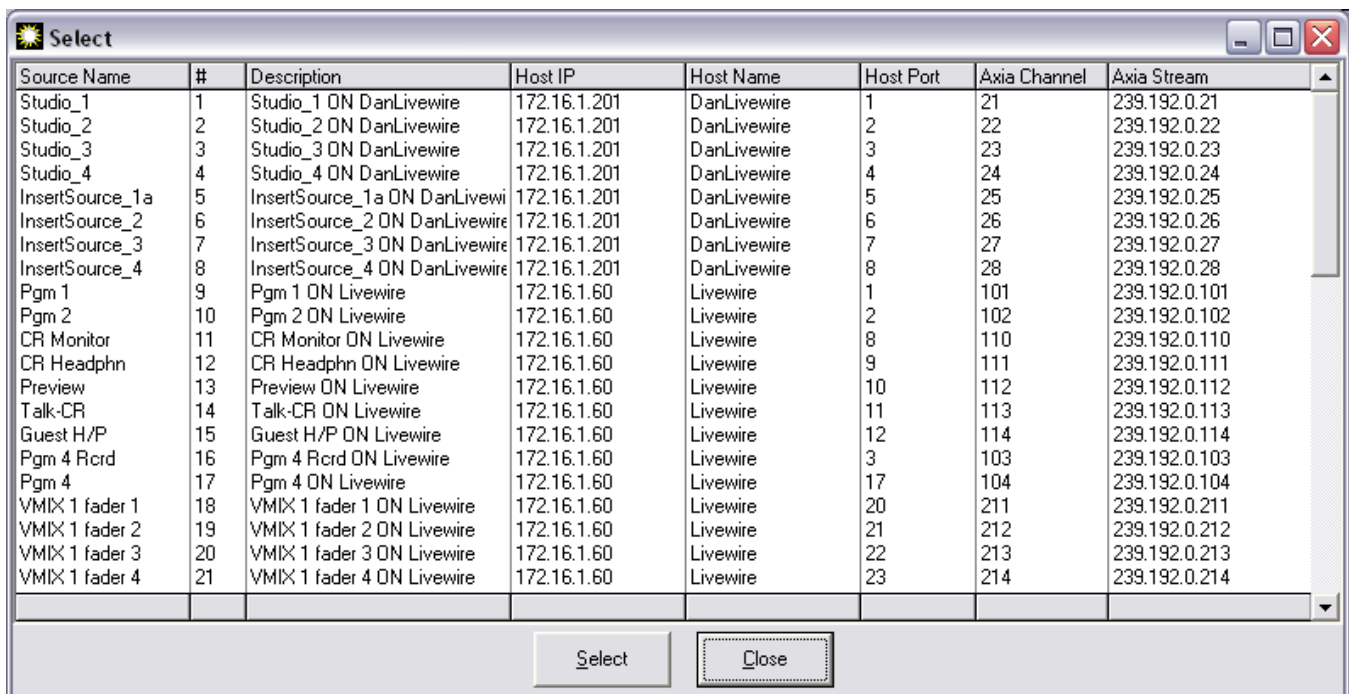
<p><u>Help</u></p>	<p>Displays this help file.</p>
<p><u>About PathfinderPC Server</u> Displays the opening splash screen which includes the version information for PathfinderPC Server.</p>	

Select List

Throughout the PathfinderPC Server and Client software, there will be drop down lists when sources and destinations need to be selected for events, stack events, 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.



Source Name	#	Description	Host IP	Host Name	Host Port	Axia Channel	Axia Stream
Studio_1	1	Studio_1 ON DanLivewire	172.16.1.201	DanLivewire	1	21	239.192.0.21
Studio_2	2	Studio_2 ON DanLivewire	172.16.1.201	DanLivewire	2	22	239.192.0.22
Studio_3	3	Studio_3 ON DanLivewire	172.16.1.201	DanLivewire	3	23	239.192.0.23
Studio_4	4	Studio_4 ON DanLivewire	172.16.1.201	DanLivewire	4	24	239.192.0.24
InsertSource_1a	5	InsertSource_1a ON DanLivewi	172.16.1.201	DanLivewire	5	25	239.192.0.25
InsertSource_2	6	InsertSource_2 ON DanLivewire	172.16.1.201	DanLivewire	6	26	239.192.0.26
InsertSource_3	7	InsertSource_3 ON DanLivewire	172.16.1.201	DanLivewire	7	27	239.192.0.27
InsertSource_4	8	InsertSource_4 ON DanLivewire	172.16.1.201	DanLivewire	8	28	239.192.0.28
Pgm 1	9	Pgm 1 ON Livewire	172.16.1.60	Livewire	1	101	239.192.0.101
Pgm 2	10	Pgm 2 ON Livewire	172.16.1.60	Livewire	2	102	239.192.0.102
CR Monitor	11	CR Monitor ON Livewire	172.16.1.60	Livewire	8	110	239.192.0.110
CR Headphn	12	CR Headphn ON Livewire	172.16.1.60	Livewire	9	111	239.192.0.111
Preview	13	Preview ON Livewire	172.16.1.60	Livewire	10	112	239.192.0.112
Talk-CR	14	Talk-CR ON Livewire	172.16.1.60	Livewire	11	113	239.192.0.113
Guest H/P	15	Guest H/P ON Livewire	172.16.1.60	Livewire	12	114	239.192.0.114
Pgm 4 Rcrd	16	Pgm 4 Rcrd ON Livewire	172.16.1.60	Livewire	3	103	239.192.0.103
Pgm 4	17	Pgm 4 ON Livewire	172.16.1.60	Livewire	17	104	239.192.0.104
VMIX 1 fader 1	18	VMIX 1 fader 1 ON Livewire	172.16.1.60	Livewire	20	211	239.192.0.211
VMIX 1 fader 2	19	VMIX 1 fader 2 ON Livewire	172.16.1.60	Livewire	21	212	239.192.0.212
VMIX 1 fader 3	20	VMIX 1 fader 3 ON Livewire	172.16.1.60	Livewire	22	213	239.192.0.213
VMIX 1 fader 4	21	VMIX 1 fader 4 ON Livewire	172.16.1.60	Livewire	23	214	239.192.0.214

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 both the client and server wherever a drop down list is presented.

Standard Event System

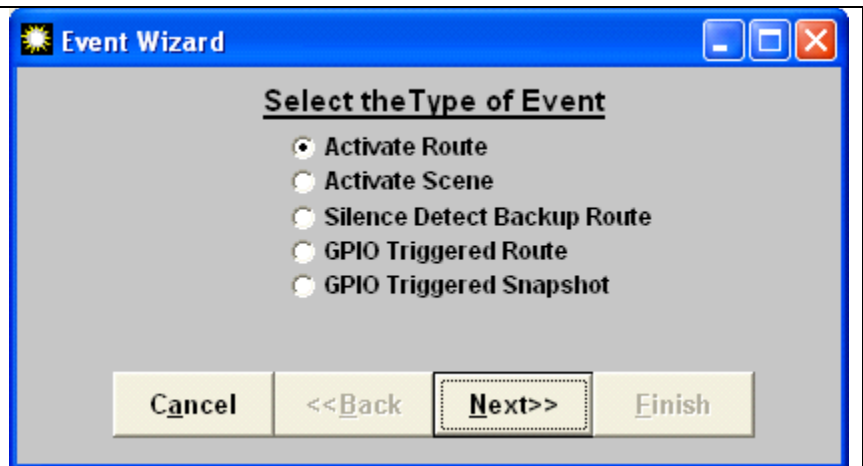
Standard Events may be created by clicking on the Events Tab in the Main Server Page and then **clicking Add Event**. There are three categories of events – Time Based Events, Audio Events, and GPIO events. Time based events are events which occur at a certain time either on a regular schedule or occur one time only. Audio events allow you to program a silence detection point. If the audio content on a particular point disappears for more than a certain amount of time, a backup route may be engaged. GPIO Events are events that are triggered by a GPI or GPO contact closure.



Time Based Events

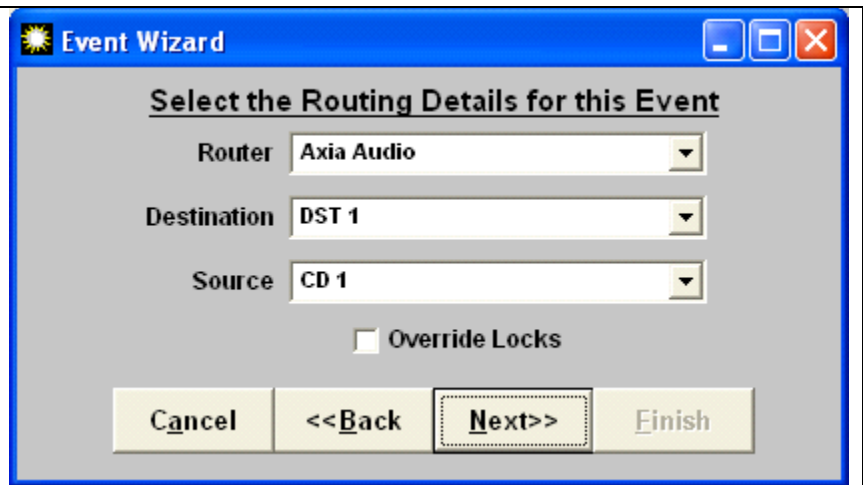
There are two types of time based events – Activate Route and Activate Scene. Activate Route switches a route at a particular date and time or on some daily schedule and Activate Scene does the same for a scene. To select a Time Based event, add a new event, and select either Activate Route or Activate Scene.

Click Next



Select the routing system on which the event is to take place, and the route details - including both a destination and source point, or a scene according to which type of event has been selected. Selecting the Override Locks checkbox will force the route to take place regardless of whether locked route points need to be changed to accomplish this goal.

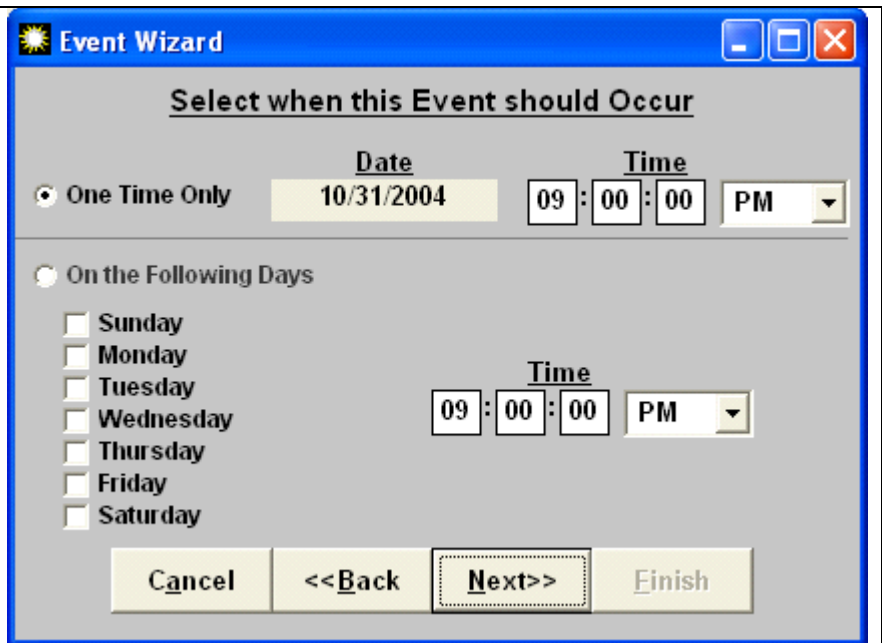
Click Next again



The dialog box is titled "Event Wizard" and "Select the Routing Details for this Event". It contains three dropdown menus: "Router" set to "Axia Audio", "Destination" set to "DST 1", and "Source" set to "CD 1". There is an unchecked checkbox labeled "Override Locks". At the bottom are four buttons: "Cancel", "<<Back", "Next>>" (highlighted with a dashed border), and "Finish".

Use this screen to schedule when the event should occur. Select whether the event should happen once or on a recurring schedule.

Click Next again



The dialog box is titled "Event Wizard" and "Select when this Event should Occur". It has two radio button options: "One Time Only" (selected) and "On the Following Days". Under "One Time Only", there is a "Date" field with "10/31/2004" and a "Time" field with "09:00:00 PM". Under "On the Following Days", there is a list of days from Sunday to Saturday, all unchecked, and a "Time" field with "09:00:00 PM". At the bottom are four buttons: "Cancel", "<<Back", "Next>>" (highlighted with a dashed border), and "Finish".

Enter a Name and Description for the event.

Click Finish to add the event.



The dialog box is titled "Event Wizard" and "Select a Name and Description for this Event!". It has two text input fields: "Event Name" containing "Sports Talk" and "Description" containing "Change options for ML Sports show". At the bottom are four buttons: "Cancel", "<<Back", "Next>>" (highlighted with a dashed border), and "Finish".



Please Note:

Time based events which occur only once will be deleted out of the events database the day after they have been programmed to occur.

Audio Based Events

An Audio based event is a great way to introduce redundancy into a system. At this point in time, the Axia Livewire system is the only system which supports audio based events, as it is the only system which allows the server application to check and monitor audio levels in the routing devices themselves.

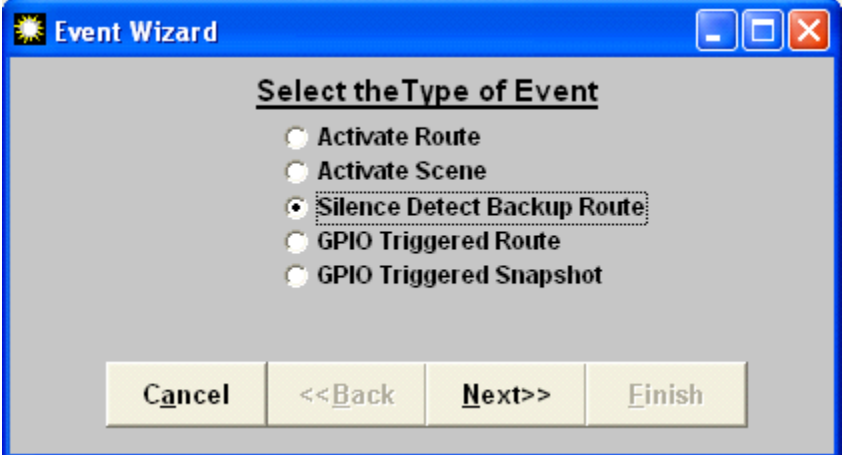
With this event you basically program the system to monitor a certain source/destination route. If the defined route is in place, and if audio falls below a certain threshold on the destination for longer than a certain amount of time the router attempts to switch to a backup source. If there is no audio on the backup source, then the server fails the route until it is able to determine that audio is present on either the primary or backup source, at which point it switches to that source. If a user routes a source other than the primary or backup source to a monitored destination, it is also considered a forced failure and the system stops attempting to make audio based routes until either the primary or backup source is again routed to the monitored destination.

Remember that since an Axia system is a distributed system, there are multiple Axia units in a single Server router. Thus, hardware redundancy may be built into the system by selecting Primary and Backup routes on different Axia nodes. Thus, if the Axia device that houses the primary source becomes unavailable, the system will automatically switch to the backup source which may be owned by a totally different Axia device.



Please Note:

At this point in time Axia Windows Driver sources and destinations do not support metering or silence/clipping detection. Thus Windows Driver sources and destinations should not be used in any silence detection events.

<p>To set up an Audio Based event</p> <p>Click Add Event</p> <p>Select Silence Detect Backup Route.</p> <p>Then click Next</p>	
--	--

Next, enter the Router, Source and Destination points for both the Primary and Backup route. While any drop down box is open, you can click on the grey header bar within the drop down window to make the description field visible or hide it. This is often useful to find the source or destination you want in a large list. Also, select Override locks if the route should be made regardless of the lock state of points involved.

Click Next again

The screenshot shows the 'Event Wizard' dialog box with the title 'Select the Routing Details for this Event'. It features a 'Router' dropdown menu set to 'DanAudio'. Below this, there are two sections: 'Primary Route' and 'Backup Route'. Each section has 'Destination' and 'Source' dropdown menus, both set to 'AirChain_1' and 'Studio_1' respectively. At the bottom, there is an 'Override Locks' checkbox which is unchecked. Navigation buttons include 'Cancel', '<<Back', 'Next>>', and 'Finish'.

Select the amount of time in seconds that the audio should be missing before switching to the backup route.

Also select GPIO settings if desired as described below.

Click Next

The screenshot shows the 'Event Wizard' dialog box with the title 'Select Audio Threshold'. It explains that the system will switch to the Backup Route when the audio level falls below the silence threshold for more than a specified duration. A dropdown menu is set to '10' with the unit 'Seconds'. Below this is the 'GPIO Closure' section with three radio button options: 'None' (selected), 'When primary fails only', and 'For all silence detect events'. The 'Router' dropdown is set to 'DanGPIO', and the 'Destination (GPIO)' dropdown is set to 'OUT 1'. There are also dropdowns for 'Contact Closure' (set to '1') and 'Change State To' (set to 'Low'). A text input field for 'Pulse Length in Milliseconds' is set to '0', with a note '(0 = Steady State)'. Navigation buttons include 'Cancel', '<<Back', 'Next>>', and 'Finish'.

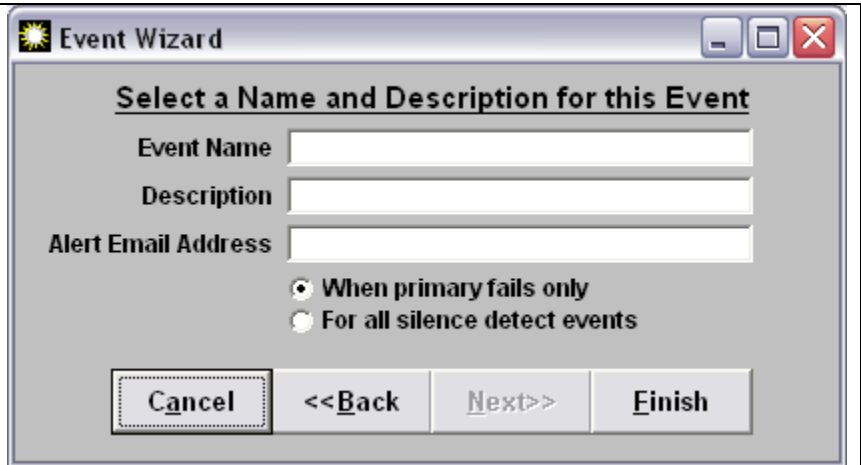


Please Note:

Some care should be taken in choosing the time value. In a radio station environment, consider the maximum amount of dead air that a host might naturally have in the middle of a conversation. If a GPIO router also exists in the PathfinderPC system you can also select a GPIO port and pin to activate. This allows you to set up a light or other GPIO alert to trigger in the event of silence detection. If you select the “When Primary Fails Only” option the GPIO will only get tripped when the primary route exists and silence is detected. If you select the “For all silence detect events,” the GPIO will get tripped when switching from primary to backup, backup to primary, or either state to failed. Finally click **Next** and add a name and description for the event.

Add a name and description for the event. You can also add an alert email address with the same logic selections as the GPIO trigger above. This will send an email to the address in the Alert Email address box. Emails will only be sent if the Email server information is entered into the system using the Email Alert Settings menu item under the PathfinderPC Server File menu.

Click Finish to add the event



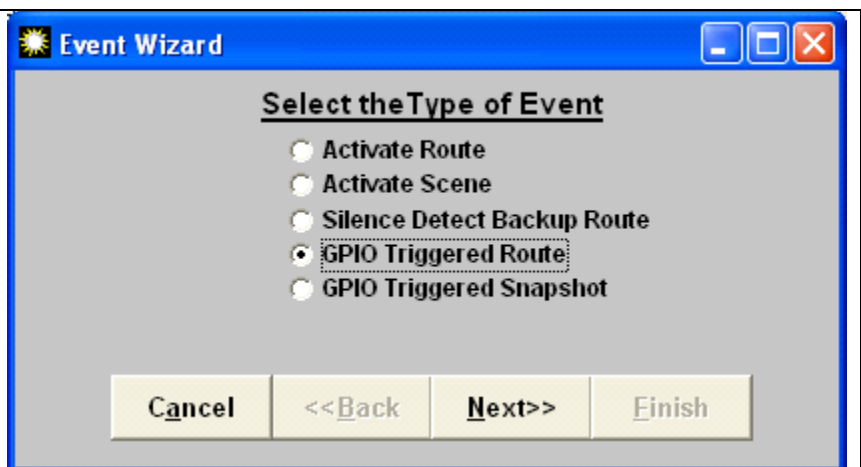
GPIO Events

Like the Time Based Events, there are two types of GPIO Events:

- **Route GPIO**
- **Scene (Snapshot) GPIO**

GPIO events are either individual routes or entire scene changes that are triggered by a change in a GPIO source contact closure.

Click Next



Select the route point information or scene to change.

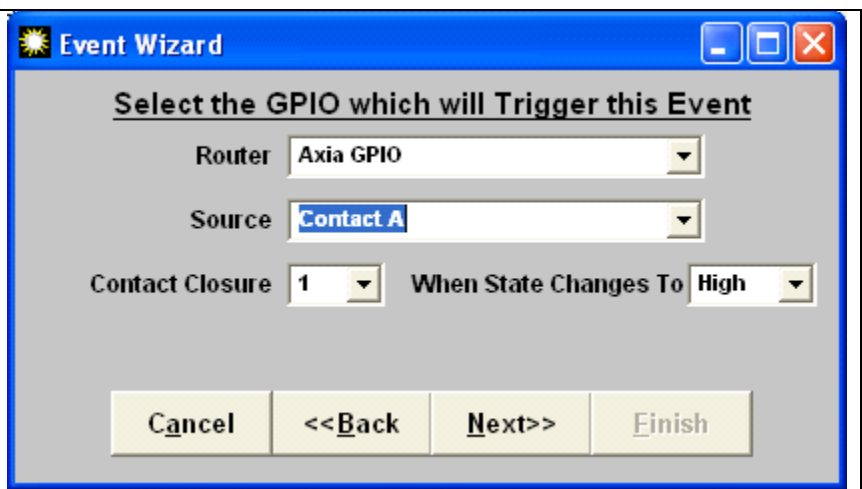
Click Next



The dialog box is titled "Event Wizard" and "Select the Routing Details for this Event". It contains three dropdown menus: "Router" set to "Axia Audio", "Destination" set to "DST 1", and "Source" set to "CD 1". There is an unchecked checkbox labeled "Override Locks". At the bottom are four buttons: "Cancel", "<<Back", "Next>>", and "Finish".

Select the GPIO source, contact closure, and whether the change to high or low triggers the event.

Click Next



The dialog box is titled "Event Wizard" and "Select the GPIO which will Trigger this Event". It contains three dropdown menus: "Router" set to "Axia GPIO", "Source" set to "Contact A", and "Contact Closure" set to "1". There is also a dropdown menu for "When State Changes To" set to "High". At the bottom are four buttons: "Cancel", "<<Back", "Next>>", and "Finish".

Finally enter a name and a description for the event.

Click Finish to add the event

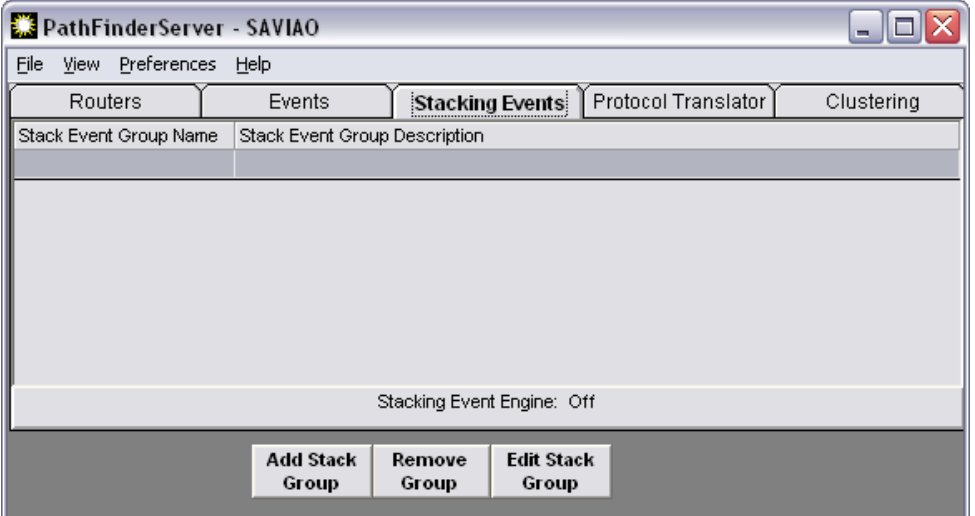


The dialog box is titled "Event Wizard" and "Select a Name and Description for this Event!". It contains two text input fields: "Event Name" with the text "MusicReelA" and "Description" with the text "Switch to Music". At the bottom are four buttons: "Cancel", "<<Back", "Next>>", and "Finish".

Stacking Events

The Stacking Event Engine in PathfinderPC Server is an exciting new technology that allows an Administrator to create highly customized events through a simple to use graphical interface. It provides the power of a scripting language without the need to know how to write in a programming language. The concept of a stacking event is that for each event the Administrator defines a list of qualifiers (conditions) and a list of Actions. If all of the qualifiers are met, all of the actions are run. This section of the document will describe the various Qualifiers and Actions available and how to use them. It will also provide some useful examples. It will take you through the concepts of creating a stack event and then present a detailed description of the various qualifiers and actions available.

Stack Event Example 1 – GPI triggered route with Email

<p>To begin creating a stacking event, select the stacking event tab within PathfinderPC Server.</p> <p>Click the “Add Stack Group” button</p>	
---	---

Stack Events are organized into Stack Event Groups. These groups have nothing to do with the actual operation of the stack events, but rather are organizational tools to make browsing the events easier. The Stack Event group can be thought of as a folder similar to a folder in windows.

Each stack event group is written to its own xml file within the PFSStackEvents folder in the PathfinderPC Server installation folder. All of the events that are in the particular group are written into that group’s xml file. This is important to know because if you create an event that you suspect is having undesired results, rather than deleting the event using the server’s user interface, you can temporarily move the file out of the folder and restart the server application. This will remove the stack event from the system during testing without deleting your work. You can then move the file back at a later time, restart the server application again and the event will be active again.

It is also important to know that in a clustered environment the Stack Events must be edited on the server that currently owns the event system. Furthermore, while the changes to Stack Events are automatically synchronized between clustered servers, the new events may not display in the Stack Event Window on the server that does not own the event system. This is because this window is not updated until and unless the Event System starts up on that server as a result of a failover. However if you browse into the stack events folder on the secondary server, you will see that in fact the XML files have been updated through the cluster synchronization.

Enter a name for the stack event group and a description if desired.

Now click **Add Event** to create your first event to be included in this Stack Event Group



The image shows a dialog box titled "Stacking Event Group List". It contains two text input fields: "Stack Event Group" and "Group Description". Below these is a section titled "Stack Events in this Group" which contains a table with two columns: "Event Name" and "Event Description". The table is currently empty. At the bottom of the dialog, there are three buttons: "Add", "Edit", and "Remove". Below these buttons are two checkboxes: "Force a run of actions met" and "Force a run of actions not met". At the very bottom are "OK" and "Cancel" buttons.

The Event creation wizard screen is shown below. Each event must have a unique name within the stack event group. The name is entered into the Event Name field, and a description field is available if desired as well. You will notice 5 boxes, each with **Add, Edit, and Remove buttons below them**. The first box is the qualifiers box. These are the conditions that will be analyzed to decide if actions are to be run. The remaining 4 boxes are what actions to run depending on whether the qualifiers are met or not met. These boxes will contain actions that will be run if the qualifiers are met, if they are not met, as well as actions that can be run on a delay after the qualifiers are met or not met.

Clicking on the Add button under any of these boxes will present a list of the available qualifiers or Actions to add.

Let's create an example. In this example stack event we are going to create an event that makes a route and sends an email when a particular GPI is closed. To run this example you will need an Axia Audio Router and an Axia GPIO Router already configured in your system. See the Add Routers section of this document for instructions if these do not already exist.

Obviously in this example our qualifier is the GPI contact closure.

About the Stack lists Screen

The screenshot shows the 'StackLists' application window with the following sections:

- Event Name:** Default
- Event Description:** [Empty text box]
- Qualifiers:** A table with columns: Qualifier Name, Qualifier Description, Qualifier Type. Below the table are 'Add', 'Edit', and 'Remove' buttons.
- Run actions if ALL qualifiers are Met** (selected) / **Run actions if ANY qualifiers are met**
- Actions if all Qualifiers are met:** A table with columns: Action Name, Action Description, Action Type. Below the table are 'Add', 'Edit', and 'Remove' buttons.
- Actions if All Qualifiers are NOT met:** A table with columns: Action Name, Action Description, Action Type. Below the table are 'Add', 'Edit', and 'Remove' buttons.
- Stack Event Name:** [Empty text box]
- Run Actions On Startup:** [Unchecked checkbox]
- Delayed action if all Qualifiers are met:** A table with columns: Action Name, Action Description, Action Type. Below the table is a 'Delay Time in Milliseconds' field set to 5000 (range 0 to 60000) and 'Add', 'Edit', 'Remove' buttons.
- Delayed action if all Qualifiers are NOT met:** A table with columns: Action Name, Action Description, Action Type. Below the table is a 'Delay Time in Milliseconds' field set to 5000 (range 0 to 60000) and 'Add', 'Edit', 'Remove' buttons.
- OK** and **Cancel** buttons at the bottom.

Click the **add button** under qualifiers.

The Stack Item Editor appears.

This close-up screenshot shows the 'StackLists' application window with the 'Qualifiers' section highlighted. It includes the 'Event Name' (Default) and 'Event Description' fields, the 'Qualifiers' table with 'Add', 'Edit', and 'Remove' buttons, and the radio button options for 'Run actions if ALL qualifiers are Met' (selected) and 'Run actions if ANY qualifiers are met'.

Provide a name and description for the qualifier. You will notice a powerful list of the kinds of qualifiers you can add. In this case we will select GPI State Change.

Click Next



Stack Item Editor

Group Name: _____

Event Name: Default

Qualifier Name: GPITrigger

Qualifier Description: Trigger From GPI

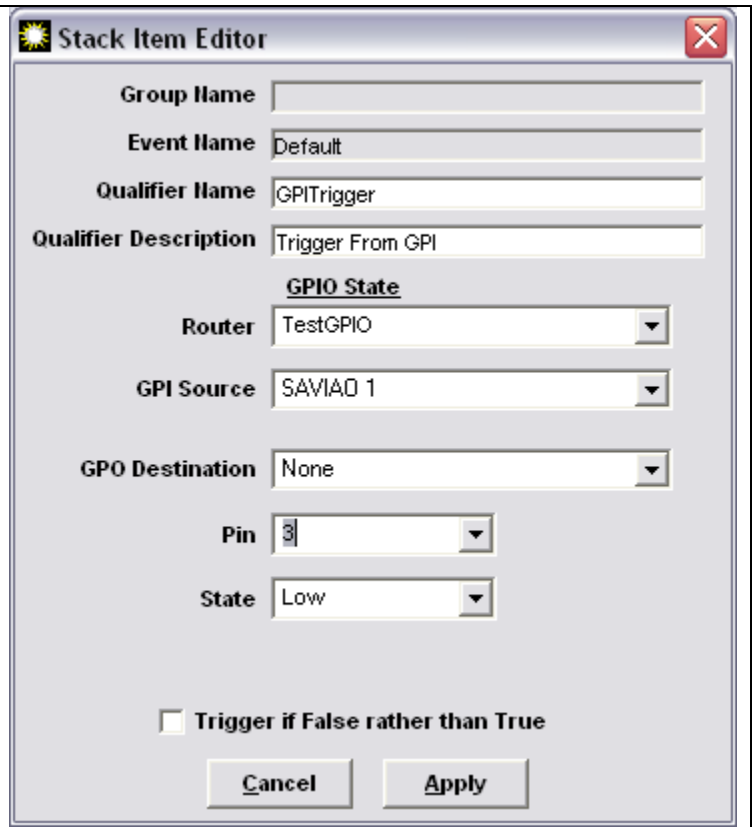
Qualification Type

- Route Status
- Audio Level Trigger
- GPI State Change
- GPO State Change
- User Button Press
- Time/Date Range
- User Command Received
- Memory Value
- Event Engine Start
- Element Profile Change
- Element VMIX Change

Cancel Next

The qualifier window now changes to show the options available for a GPI State Change qualifier. Select the GPIO Router and the GPI Source Port from the drop down menus that will trigger the event. Also select the pin and whether the GPI state going low or high will trigger the event.

Then click Apply



Stack Item Editor

Group Name: _____

Event Name: Default

Qualifier Name: GPITrigger

Qualifier Description: Trigger From GPI

GPIO State

Router: TestGPIO

GPI Source: SAVIAO 1

GPO Destination: None

Pin: 3

State: Low

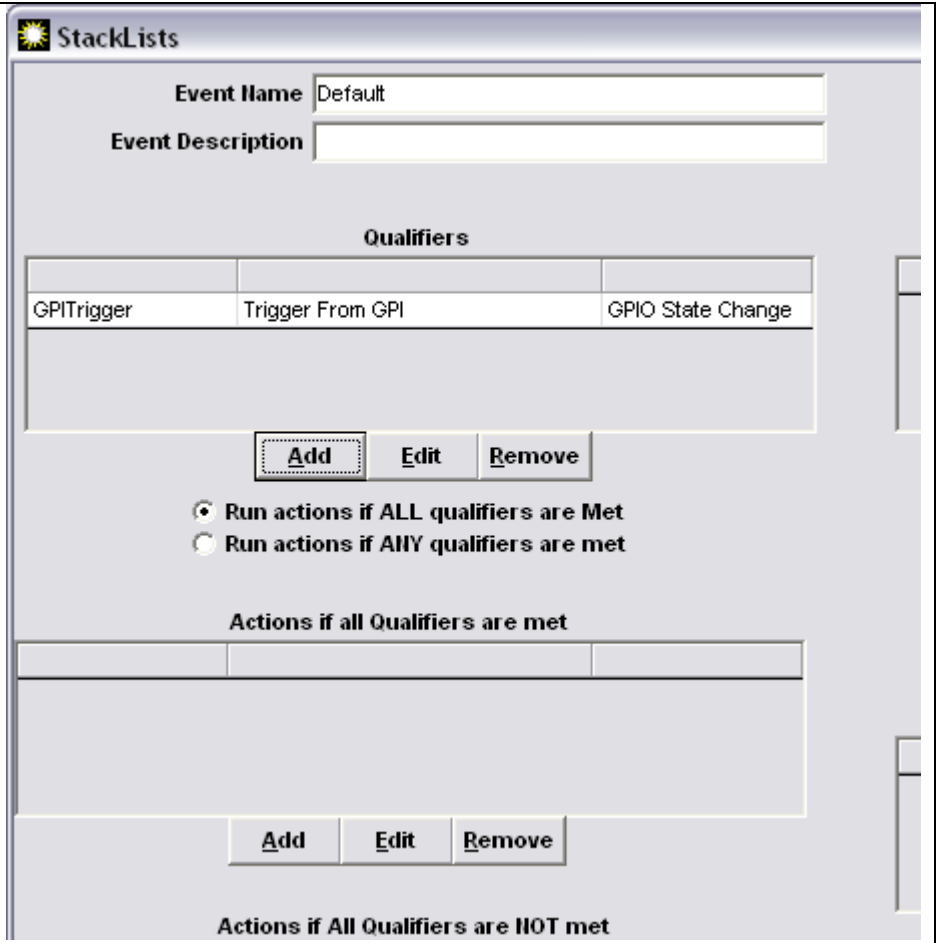
Trigger if False rather than True

Cancel Apply

The qualifier settings window will disappear and you will see a qualifier in the qualifier list in Stack Lists screen.

So now we know that a specific GPI is going to trigger this event, but we must determine what the event will do when that GPI is closed.

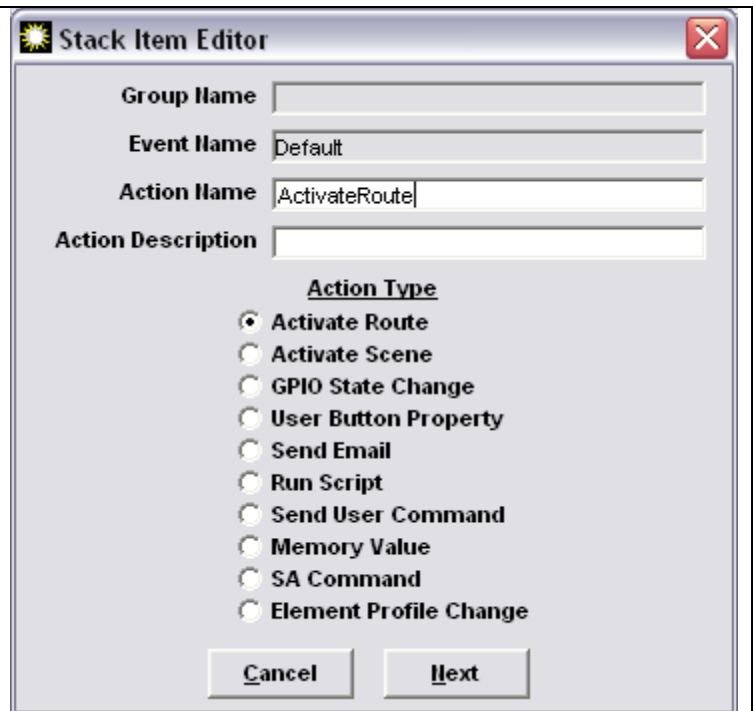
Click on the Add button under the Actions if Qualifiers are met list



Again you will see a list of the available kinds of actions that can be run. In the case of this demo enter a name for the action.

Select the Activate Route action type

Click next



Again a list of the available options for the Activate Route action appears. **Select the Router** on which you want the route to take place and the source and destination you want to be activated. You can also select to lock the route once it is made and to override a lock and force the route if the destination is locked already.

Click Apply

Stack Item Editor

Group Name: _____

Event Name: Default

Action Name: ActivateRoute

Action Description: _____

Route State

Router: DanAudio

Source: Studio_1

Destination: AirChain_1

Lock

Lock Override

Cancel Apply

The editor window will disappear and there will be an action in the Actions if qualifiers are met list.

Now click the Add button below the Actions if qualifiers are met list again.

Run actions if ALLY qualifiers are met

Actions if all Qualifiers are met

ActivateRoute	Activate Route
---------------	----------------

Add Edit Remove

Actions if All Qualifiers are NOT met

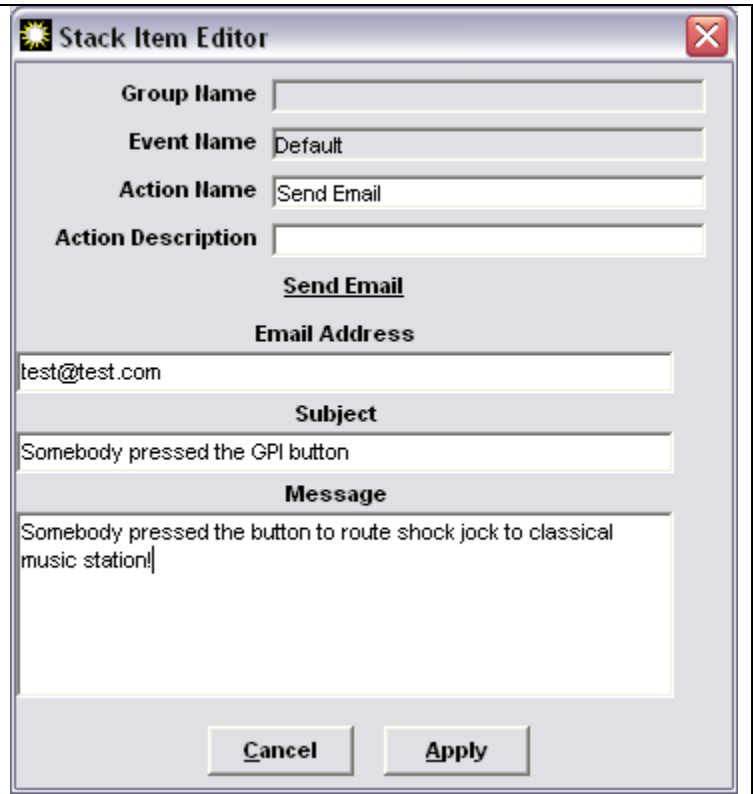
--	--

Give the action a unique name and this time **select the Send Email** type of action.

Click next

This time the editor displays options for sending an email. Fill in where you want the message to be sent, the subject, and content of the message.

Click Apply



Stack Item Editor

Group Name

Event Name

Action Name

Action Description

Send Email

Email Address

Subject

Message

Cancel Apply

Again the editor window will disappear and you should now have two actions that will take place when the GPI button is pressed. Now click OK on the Stack List Editor window and OK on the Stacking Event Group List window. The events will get created and the Stack Event Group will display in the main window of PathfinderPC Server under the Stacking Events tab. Now when you close the GPI, the route should change and an email will be sent.

MORE EXAMPLES

Let's expand this example slightly now.

Example 2 – Time/Date

Select the event in the main PathfinderPC Server window and click edit. This will open the Stack Event Group List window. **Now select** the one stack event we have in this group at this time and click edit again. You should now see your stack event with one qualifier and two actions.

Let's add another qualifier.

Click **add** under the qualifier list and select Time/Date range making sure to also fill in the qualifier name field.

Click Next

Stack Item Editor

Group Name

Event Name Default

Qualifier Name DuringSpecificTimes

Qualifier Description

Time.Date

Start Date

One Time Only Date

7/16/2006

On the Following Days

Sunday

Monday

Tuesday

Wednesday

Thursday

Friday

Saturday

End Date

One Time Only Date

7/16/2006

On the Following Days

Sunday

Monday

Tuesday

Wednesday

Thursday

Friday

Saturday

Start Time 5:00:00 PM

End Time 9:00:00 PM

Trigger if False rather than True

Cancel Apply

In this case we are adding a qualifier with specific days and times - in this case Monday, Tuesday, and Thursday between 5 PM and 9 PM. Adding this qualifier into the event will mean that the route and email will only occur if the GPI button is pressed between the times 5 PM and 9 PM. Apply the qualifier and click OK on the Stack List and Stack Event Group List screens to write the stack event file and test your results.

In the example above, you will notice that each qualifier has also had a checkbox that can be checked that says “Trigger if False rather than True.” This checkbox will effectively invert the meaning of the qualifier. So, for example, if we had checked that box on the time range qualifier it would mean that the actions would be run if the GPI were pressed and it was any time of day other than Monday, Tuesday, or Thursday between 5 PM and 9 PM.

Test your results. Of course they should now only work if it is Monday, Tuesday, or Thursday, and between 5 and 9 PM.

Example 3 – Talkback Button

In this example we will use a stack event and a custom user panel to create a talkback button. To run this example you must first create a custom user panel. See the section of this document on the Panel Designer to create the panel. The panel must have a name for the panel and a unique name for each control on the panel, as this is what will be used in the stack event. So for this demo create a panel called DemoPanel with a button called DemoButton. Also for this example Destination 1 on Router 1 will be the headphone destination we are

switching, and Source 1 will be the program material and Source 2 will be the Talkback microphone. Now as described in the previous section add a Stack Event group and give it a name, or edit your existing group and add this as an additional stack event within that group.

Next add a qualifier of type UserButtonPress

Enter a name for the qualifier, and enter the panel and button control name in the button name field in the form: PanelName.ButtonName

In our example this should be:
DemoPanel.DemoButton

Stack Item Editor

Group Name

Event Name

Qualifier Name

Qualifier Description

User Panel Button

Use Named Button

Button Name
(Format: PanelName.ButtonName)

Use Button Address

Host IP

Button Module

Button Number

Key State

Back Color On

Back Color Off

Fore Color

State

Flash

Button Caption

Set Button Name

Trigger if False rather than True

Cancel Apply

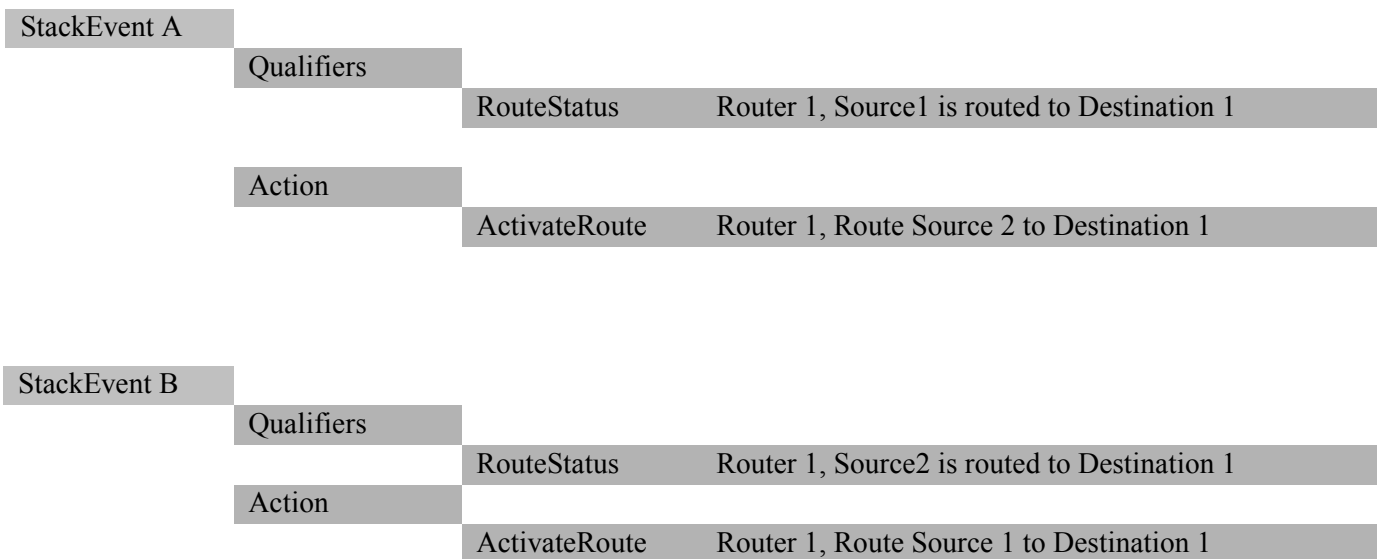
Make sure the KeyState is set to down, and leave the other parameters as they are defaulted. This means that the only qualifier we are looking for is the button being down. We could if we desired add additional aspects of the button qualification such as a specific state as will be called for in Example 3. **Click Apply**. Our qualifier will now be true whenever the user panel button is pressed on any open copy of the DemoPanel user panel in a client application, or hardware device that is mapped to that panel control. For Actions in the Actions if qualifiers are met section, add an Action of type Activate Route and set the router to the correct audio router and the destination to the Destination 1 and the Source to Destination 2.

This will make the route from the Talkback Source to the Headphone Destination whenever the button is pressed. Next, add an Action to the Actions if qualifiers are not met section. This will return the route whenever the button is not down as the qualifier dictates. In this Action, select an ActivateRoute Action and set the Source to 1 and the Destination to 1.

Thus when the button is pushed, the talkback source will be routed to the headphone destination, and when it is released the program content will be sent to the destination.

Endless Loops

One thing that must be kept in mind when creating stack events is that there currently is no logic within PathfinderPC Server to prevent an endless loop. For example,



You will notice the outline style diagram. This is often an advantageous way to design Stack Event logic on paper before you create your events - especially if your event will be complex. In the example above the action for StackEventA will trigger the qualifier for StackEventB which will in turn trigger the Action for StackEventB which triggers the qualifier for StackEventA. You now have an endless loop causing the route to switch back and forth forever. There is nothing to prevent you from creating such an event, and it can bring the server application to its knees as it tries vainly to toggle the route back and forth as fast as it can. If you think you have accidentally created an endless loop with a stack event, it is easy to diagnose. Simply move the stack event group file out of the PFSSStackEvents folder and re-launch the server application. If the problem goes away, the stack event needs to be redesigned.

Example 3 – Latching Talkback Button

In this example we will create a talkback button similar to the talkback example above, except this time we want it to switch to talkback when you press the button the first time, and switch back to program content when you press it a second time. In this case we are going to outline the logic on paper first as we did in the endless loop example to make sure we get it correct. To create our latching talkback button we need something to keep track

of the state of the button, and since the button has a state option we can use that. Again in this example Source 1 is program material and Source 2 is the talkback source. Destination 1 is our headphone destination.

StackEvent A	
Qualifiers	User Panel Button DemoPanel.DemoButton – ButtonDown – State Off
Action	ActivateRoute Route Source 2 to Destination 1 User Panel Button Set DemoPanel.DemoButton State to On

StackEvent B	
Qualifiers	User Panel Button DemoPanel.DemoButton – ButtonDown – State On
Action	ActivateRoute Route Source 1 to Destination 1 User Panel Button Set DemoPanel.DemoButton State to Off

While at first glance this logic looks correct, in actuality it will create an endless loop for as long as the button is held down. This is because as soon as any item involved in the qualifier list is changed for an event, the event is analyzed. Thus, in EventA the button is pushed down when its state is off. The Action sets the State to On and makes the route. As soon as the state is set to On the button is still down because the computer is much faster than the finger’s button release, therefore the qualifiers for StackEventB are true and its actions are run – triggering the qualifiers of Action A. Therefore, we must find a way to only allow one event to run each time the button is pushed. The solution is to use a memory slot. Memory slots can be defined in an action by using a name and a value for the slot. Then they can also be qualifiers.

StackEventA	
Qualifier	User Panel DemoPanel.DemoButton – ButtonDown – State Off MemorySlot MyButtonMem - Value field blank
Action	Memory Slot MyButtonMem = “ENGAGED” Route Change Route Source 2 to Destination 1 User Panel Set DemoPanel.DemoButton State to On

StackEventB		
Qualifier	User Panel	DemoPanel.DemoButton – ButtonDown – State On
	MemorySlot	MyButtonMem - Value field blank
Action	Memory Slot	MyButtonMem = “ENGAGED”
	Route Change	Route Source 1 to Destination 1
	User Panel	Set DemoPanel.DemoButton State to Off

StackEventC		
Qualifier	User Panel	DemoPanel.DemoButton – ButtonUp
Action	Memory Slot	MyButtonMem = Value field blank

It is very important to understand that the actions in a stack event are processed in the order in which they appear in the list. Thus, when the button is pressed for the first time in StackEventA, the memory slot MyButtonMem has not been defined, and so it is empty. Therefore, since it is empty and the current state of the button is off and it is down the qualifiers are true. The first action that the StackEventA runs is to define a value for MyButtonMem. Now that memory slot is no longer blank so StackEventB will not run when Stack Event A makes the Route Change and button state change. Then when the button is released, StackEventC clears the memory slot, but StackEventB will not run because the button is now up. The next time the button is pushed the memory slot is empty again, and the button state is On, and the button is down, so StackEventB runs, which again first sets the memory slot to a value so StackEventA will not run again until the button is released. If the memory slot change does not precede the Button state change in the action list however, the whole thing will fall apart.

This shows the huge possibilities for the advanced event capabilities that are possible through the Stack Event Engine of PathfinderPC Server. The next section will step through the parameters that are available through each of the qualifiers and actions in the Stack Event Engine. In reality, the creative Administrator that takes the time to become acquainted with all of the possible qualifier and action tools in the PathfinderPC Server Stack Event arsenal should rarely or ever have the need for custom scripting. However, in the rare instances where the customized needs would require more advanced logic than can be easily accomplished with these tools, PathfinderPC can also access and run custom scripts. Contact Axia and/or Software Authority if you have need of custom scripting to drop into your PathfinderPC Server installation.

StackEvent Qualifiers

RouteStatus

The RouteStatus qualifier allows you to define a source and destination on a particular PathfinderPC router. The stack event will be analyzed whenever the destination has its source route assignment changed. If the source point that is assigned to the destination corresponds to the source in the qualifier, the qualifier will be considered True. Otherwise it will be considered False.

If the **Use Lock State as Part of Qualifier** is checked the qualifier will only be True if the lock state of the route also matches the lock state of the lock checkbox located beneath this item.

Trigger if False rather than True reverses the qualifier state making the qualifier False if the route conditions are met and True if they are not.

The screenshot shows the 'Stack Item Editor' dialog box. It has a title bar with a sun icon and a close button. The fields are: Group Name (empty), Event Name (Default), Qualifier Name (TestRoute), and Qualifier Description (empty). Under the 'Route State' section, Router is set to 'DanAudio', Source to 'Studio_1', and Destination to 'AirChain_1'. There are three checkboxes: 'Use Lock State as Part of Qualifier' (unchecked), 'Lock' (unchecked), and 'Trigger if False rather than True' (unchecked). At the bottom are 'Cancel' and 'Apply' buttons.

Audio Level Trigger

The Audio Level Trigger qualifier allows you to create a condition based on the audio state of a source and or destination. If the audio is silent or present or in clipping for more than a certain amount of time the qualifier is considered to be True. Only routers that have audio metering capabilities will be available in the router list. If you leave either the **Source** or the **Destination** field set to None the qualifier will only trigger off of the selected source or destination. If you create an entry for both source and destination the qualifier will only be True if both the source and the destination selected fulfill the audio state. The option buttons allow you to select whether the audio qualifier is based on **Silence**, **Audio Present**, or **Clipping**. The **Timeout** defines how long in seconds the audio state must exist before the qualifier is considered to be True.

It is important to note that at this point in time Axia windows driver sources and destinations do not support metering and/or silence/clipping

The screenshot shows the 'Stack Item Editor' dialog box. It has a title bar with a sun icon and a close button. The fields are: Group Name (empty), Event Name (Default), Qualifier Name (TestAudio), and Qualifier Description (empty). Under the 'Audio Level State' section, Router is set to 'DanAudio', Source to 'None', and Destination to 'None'. There are three radio buttons: 'Silence Detect' (selected), 'Audio Present' (unchecked), and 'Clipping' (unchecked). There is a 'Timeout in Seconds' field with the value '5'. There is a checkbox for 'Trigger if False rather than True' which is unchecked. At the bottom are 'Cancel' and 'Apply' buttons.

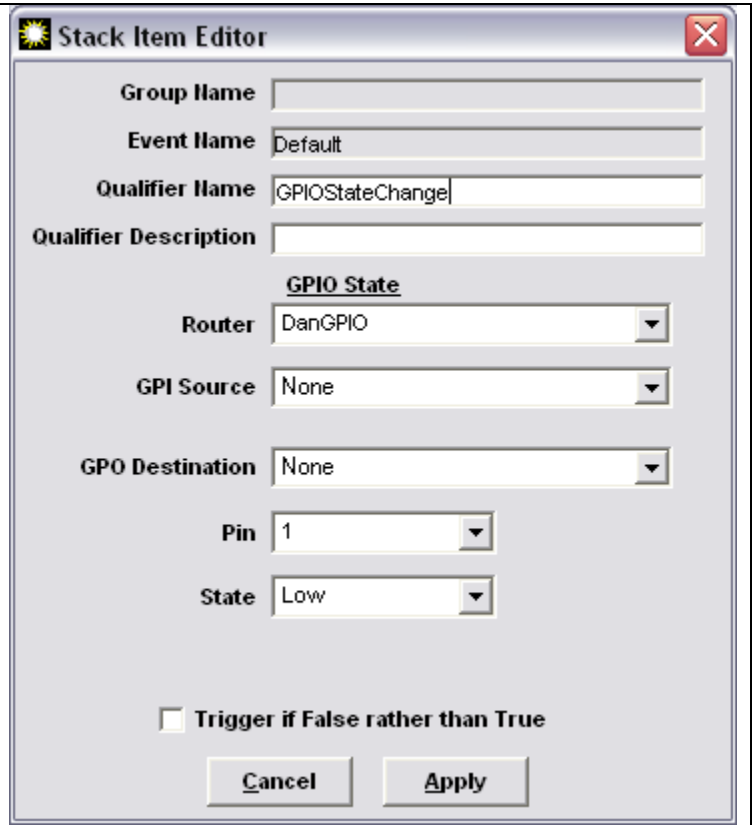
detection. These sources and destinations will always be seen as silent at this point in time

Trigger if False rather than True reverses the qualifier state making the qualifier False if the audio conditions are met and True if they are not.

GPI State Change

The GPI State Change qualifier allows you to select a GPI Source pin from a GPIO router and monitor its state. Select the GPIO Router and the GPI Source to be monitored from the Router and GPI Source drop down lists. Then select the Pin (1 through 5) to monitor and the state (Low or High) that will make the qualifier True. A stack event using this qualifier will get analyzed every time the pin on the selected port changes its state from high to low or low to high. If the pin state matches the one selected in the qualifier the qualifier will be considered True. If it does not it will be considered False.

Trigger if False rather than True reverses the qualifier state effectively reversing the State option.



The screenshot shows the 'Stack Item Editor' dialog box. It contains the following fields and options:

- Group Name:** (Empty text field)
- Event Name:** Default
- Qualifier Name:** GPIOStateChange
- Qualifier Description:** (Empty text field)
- GPIO State:** (Section header)
- Router:** DanGPIO
- GPI Source:** None
- GPO Destination:** None
- Pin:** 1
- State:** Low
- Trigger if False rather than True**
- Buttons:** Cancel, Apply

GPO State Change

The GPO State Change qualifier allows you to select a GPO Destination pin from a GPIO router and monitor its state. Select the GPIO Router and the GPO Destination to be monitored from the Router and GPO Destination drop down lists. Then select the Pin (1 through 5) to monitor and the state (Low or High) that will make the qualifier True. A stack event using this qualifier will get analyzed every time the pin on the selected port changes its state from high to low or low to high. If the pin state matches the one selected in the qualifier the qualifier will be considered True. If it does not it will be considered False.

Trigger if False rather than True reverses the qualifier state effectively reversing the State option.

The screenshot shows the 'Stack Item Editor' dialog box with the following fields and options:

- Group Name:** [Empty text box]
- Event Name:** Default
- Qualifier Name:** [Empty text box]
- Qualifier Description:** [Empty text box]
- GPIO State:** [Section header]
- Router:** DanGPIO
- GPI Source:** None
- GPO Destination:** None
- Pin:** 1
- State:** Low
- Trigger if False rather than True**
- Buttons:** Cancel, Apply

User Button Press

The User button Qualifier allows a defined user button in a Panel created with the panel designer to act as a qualifier. This qualifier will be analyzed both if the changes happen on the software panel button, and if they happen on a hardware mapped hardware button. To use this qualifier enter a button name in the Button Name field in the following format:

PanelName.ButtonName

The HostIP, ButtonModule, and ButtonNumber fields may be used instead of a Panel Designer button name to make this qualifier react to an Axia hardware User Panel or Accessory Panel button. Enter the Host IP of the Element CPU hosting the button and select the module number and button number to use. In addition, scrolling to the bottom of the button module list will display a LWChannel option. Selecting this will allow you to add a channel number and respond to fader On/Off button pushes from a fader with that LWChannel loaded.

Next select the button states that will make the qualifier True. Select the Key State (Up, Down, or No Change). You can also enter colors and caption information as part of the qualifier.

The primary colors can be entered by name – for example Red, Green, Blue. You can match subtler color shades by using RGB values in the hexadecimal format:

&HBBGGRR

For example:

&H00FF00 – would produce green

Any field that is set to No Change or is blank will not be considered when analyzing the qualifier. Any field that does have information in it will have to match the actual button condition for the qualifier to be True. Any fields that have information in them and are therefore being considered as part of the qualifier will cause the stack event to be analyzed any time one of those

Stack Item Editor

Group Name: _____

Event Name: Default

Qualifier Name: _____

Qualifier Description: _____

User Panel Button

Use Named Button

Button Name: _____
(Format: PanelName.ButtonName)

Use Button Address

Host IP: _____

Button Module: USER#1

Button Number: BUT#1

Key State: Down

Back Color On: _____

Back Color Off: _____

Fore Color: _____

State: No Change

Flash: No Change

Button Caption: _____

Set Button Name: _____

Trigger if False rather than True

Cancel Apply

pieces of information changes.

In most cases the only field you will use is the Key State field as that is how you will make specific actions happen when a button is pushed.

Trigger if False rather than True reverses the qualifier state making the qualifier False if all of the various fields' conditions are met, and True if they are not.

Time/Date Range

The Time and Date Range qualifier allows you to make a qualifier based on a certain time. This allows you to create a start Time and Date and an End Time and Date. If the time is currently between these times the qualifier is True. Otherwise it is False. You can specify a specific date or a day of the week for the start and end dates depending on whether the event only happens once or recurs on a weekly schedule.

The qualifier will be analyzed whenever the start time/date or end time/date is crossed. In addition this qualifier is often married to a second qualifier. For example, you could create a stack event with a GPI qualifier and a Time/Date Range qualifier. This would mean that when the GPI happens and it falls in the certain time range, perform the Actions.

Trigger if False rather than True reverses the qualifier state making the qualifier True if the date and time are any time other than the selected range.

Stack Item Editor

Group Name

Event Name

Qualifier Name

Qualifier Description

Time/Date

Start Date

One Time Only Date

On the Following Days

Sunday

Monday

Tuesday

Wednesday

Thursday

Friday

Saturday

End Date

One Time Only Date

On the Following Days

Sunday

Monday

Tuesday

Wednesday

Thursday

Friday

Saturday

Start Time

End Time

Trigger if False rather than True

User Command Received

This qualifier looks for certain textual commands to enter a Protocol Translator. For example, you could create a Generic translator and have an automation system send text commands to it. Then in a Stack Event, define the translator to listen to and the command you are listening for in the qualifier. When the command is received the qualifier is considered to be True.

Escape characters can be used to look for non-typable characters as follows.

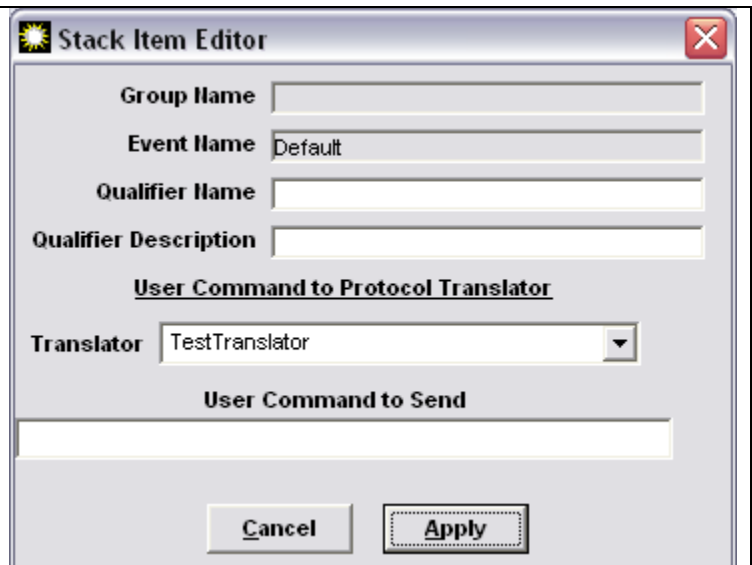
\\ = \

\cr = Carriage Return

\lf = Line Feed

\t = Tab

%xx where xx is a hexadecimal number that represents an Ascii character of the number represented by xx (PFS 3.58 and later only)



The screenshot shows a dialog box titled "Stack Item Editor" with a close button in the top right corner. The dialog contains several input fields and a dropdown menu:

- Group Name:** An empty text input field.
- Event Name:** A text input field containing the word "Default".
- Qualifier Name:** An empty text input field.
- Qualifier Description:** An empty text input field.
- User Command to Protocol Translator:** A section header.
- Translator:** A dropdown menu with "TestTranslator" selected.
- User Command to Send:** A section header followed by an empty text input field.
- At the bottom, there are two buttons: "Cancel" and "Apply".

Memory Value

PathfinderPC Server has 1000 memory slots and 99 memory matrixes available for Stack Event and custom scripting use. Each slot or Matrix is accessed using a name. A memory slot holds a single textual value. A Memory matrix is basically a table with rows and columns.

Use the option to select whether you are basing your qualifier on a memory slot or a memory matrix. Then enter the memory slot or matrix name into the correct field.

When using a memory slot the stack event will be analyzed every time the named memory slot's value changes. If the qualifier is analyzed due to the action of another qualifier, the qualifier will be considered True if a slot with the assigned name exists and it contains the value specified. If the slot does not exist or is a different value it will be considered False.

The memory matrix section allows you to enter a value. The **"Is Data In Memory List Under Column"** option allows you to select a column number in the matrix and determine if the value exists in any row under that column. If it does, the qualifier is True. If it does not, it is False.

The **"Does Data equal value at Row/Col"** option allows you to compare the value at the specified row and column with the defined value. If they are equal, the qualifier is True. Otherwise, it is False.

Trigger if False rather than True reverses the qualifier state making the qualifier True memory values do not match and/or the memory slot or matrix does not exist.

Stack Item Editor

Group Name

Event Name

Qualifier Name

Qualifier Description

Memory Parameters

Compare To Memory Slot

Memory Slot Name

Memory Slot Value

Add To Memory List

Memory List Name

Value to Compare to Memory

Is Data In Memory List Under Column

Does Data equal value at Row/Col

Row

Col

Trigger if False rather than True

Event Engine Start

This qualifier is only analyzed every time the Stack Event System Starts up or shuts down. It is considered True if the Engine has just started and False if it has just stopped.

Trigger if False rather than True reverses this state.




The screenshot shows the 'Stack Item Editor' dialog box. It has a title bar with a close button (X) in the top right corner. The dialog contains the following fields: 'Group Name' (empty text box), 'Event Name' (text box with 'Default' entered), 'Qualifier Name' (empty text box), and 'Qualifier Description' (empty text box). Below these fields is the text 'Event System Startup'. At the bottom, there is a checkbox labeled 'Trigger if False rather than True' which is currently unchecked. At the very bottom are two buttons: 'Cancel' and 'Apply'.

Element Profile Change

This qualifier allows you to select a specific Axia Element and Profile on that Element in the Axia network. If the specified profile is loaded then the qualifier is considered to be True. If it is not then it is False. Loading a Profile on the specified Element will cause a stack event with this qualifier to be analyzed.

Trigger if False rather than True will cause the qualifier to be True if any Profile other than the selected one is loaded.



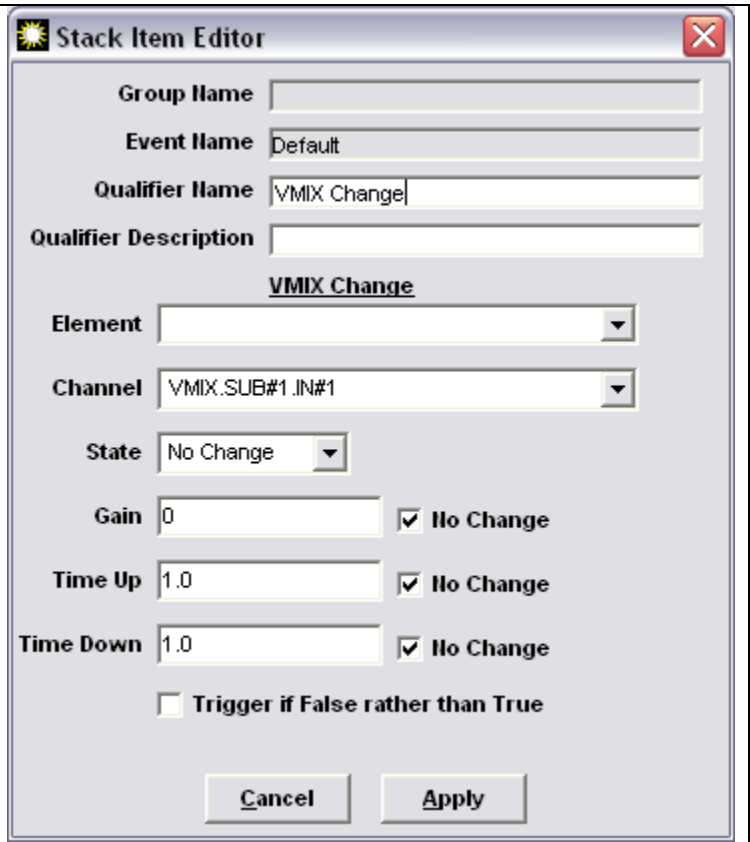
The screenshot shows the 'Stack Item Editor' dialog box. It has a title bar with a close button (X) in the top right corner. The dialog contains the following fields: 'Group Name' (empty text box), 'Event Name' (text box with 'Default' entered), 'Qualifier Name' (empty text box), and 'Qualifier Description' (empty text box). Below these fields is the text 'Profile Change'. There are two dropdown menus: 'Element' (with 'None' selected) and 'Profile' (with 'None' selected). At the bottom, there is a checkbox labeled 'Trigger if False rather than True' which is currently unchecked. At the very bottom are two buttons: 'Cancel' and 'Apply'.

VMIX Change

This qualifier allows you to select a specific Axia Element and VMIX Channel on that Element in the Axia network. Typically the field that will be most used for this qualifier is the State Field. This allows you to select On or Off states for the Channel to trigger an Action. If any of the “No Change” check boxes are unchecked then the corresponding fields must also match the state of the VMIX Channel in order for the qualifier to be considered True. A Stack Event with a VMIX qualifier will be analyzed whenever the selected VMIX Channel State changes.

Trigger if False rather than True will cause the qualifier to be True any of the VMIX states are different from the ones selected in the qualifier.

This feature only available with a PRO license.

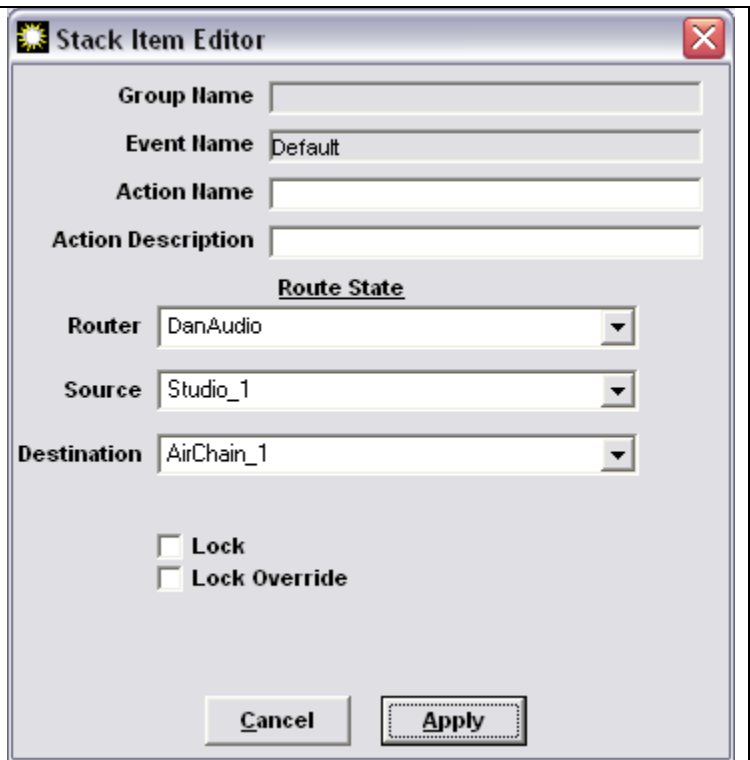


The screenshot shows the 'Stack Item Editor' dialog box. It has a title bar with a close button. The fields are as follows: Group Name (empty), Event Name (Default), Qualifier Name (VMIX Change), and Qualifier Description (empty). Below these is a section titled 'VMIX Change' with a dropdown for 'Element' (empty), a dropdown for 'Channel' (VMIX.SUB#1.IN#1), a dropdown for 'State' (No Change), a text field for 'Gain' (0) with a checked 'No Change' checkbox, a text field for 'Time Up' (1.0) with a checked 'No Change' checkbox, and a text field for 'Time Down' (1.0) with a checked 'No Change' checkbox. At the bottom, there is an unchecked checkbox for 'Trigger if False rather than True' and 'Cancel' and 'Apply' buttons.

StackEvent Actions

Activate Route

This action will activate the route selected using the Router, Source, and Destination lists. If the Lock Override is selected the route will be made regardless of whether the selected destination is currently locked. If the lock override is not selected the route will only be made if the destination is currently unlocked. If the route can be made and the lock option is checked the software will also lock the destination after making the change.



The screenshot shows the 'Stack Item Editor' dialog box. It has a title bar with a close button. The fields are as follows: Group Name (empty), Event Name (Default), Action Name (empty), and Action Description (empty). Below these is a section titled 'Route State' with dropdowns for 'Router' (DanAudio), 'Source' (Studio_1), and 'Destination' (AirChain_1). At the bottom, there are two unchecked checkboxes: 'Lock' and 'Lock Override', and 'Cancel' and 'Apply' buttons.

Activate Scene

This action will activate the selected scene on the selected router. If the Lock Override is selected all routes in the scene will be made regardless of whether the destinations involved are locked or not. If the Lock Override is not checked only those routes in the scene whose destinations are not locked will be made.

The screenshot shows the 'Stack Item Editor' dialog box. It has a title bar with a close button (X) in the top right corner. The dialog contains the following fields and controls:

- Group Name:** An empty text input field.
- Event Name:** A text input field containing the text 'Default'.
- Action Name:** An empty text input field.
- Action Description:** An empty text input field.
- Scene Change:** A section header.
- Router:** A dropdown menu with 'DanAudio' selected.
- Scene:** A dropdown menu with '1Off' selected.
- Lock Override:** An unchecked checkbox.
- Buttons:** 'Cancel' and 'Apply' buttons at the bottom.

GPIO State Change

This action will change the state of a GPI or GPO pin. Please note that the only GPIs that can be changed are Axia Driver GPIs. All GPOs on an Axia network can be changed however. Select the GPIO router and the GPO Destination Port (or GPI Driver Source Port) to be changed. Then select the Pin and State to which it should be changed. If it is to be a constant change leave the pulse length at 0. If the change is to be a pulse that then returns to the original state, enter the pulse length in milliseconds in the Pulse field.

The screenshot shows the 'Stack Item Editor' dialog box. It has a title bar with a close button (X) in the top right corner. The dialog contains the following fields and controls:

- Group Name:** An empty text input field.
- Event Name:** A text input field containing the text 'Default'.
- Action Name:** An empty text input field.
- Action Description:** An empty text input field.
- GPIO State:** A section header.
- Router:** A dropdown menu with 'DanGPIO' selected.
- GPI Source:** A dropdown menu with 'None' selected. Below it is the text '(GPI Actions only Valid for Drivers)'. Note: The text in the image says 'GPI Source' but the description refers to GPO actions.
- GPO Destination:** A dropdown menu with 'None' selected.
- Pin:** A dropdown menu with '1' selected.
- State:** A dropdown menu with 'Low' selected.
- Pulse In Milliseconds:** A text input field containing '0'. Below it is the text '(0 = Steady State)'.
- Buttons:** 'Cancel' and 'Apply' buttons at the bottom.

User Button Property

This action allows the user to change the state of software buttons defined in a Panel using Panel Designer. It will also change the state of any hardware buttons mapped to these software user panel buttons. See the Panel Designer section of this document for details on software panels and hardware mapping.

Enter the User Panel and button name in the Button Name field in the format:
PanelName.ButtonName

The HostIP, ButtonModule, and ButtonNumber fields may be used instead of a Panel Designer button name to make this Action change an Axia hardware User Panel or Accessory Panel button. Enter the Host IP of the Element CPU hosting the button and select the module number and button number to use. In addition, scrolling to the bottom of the button module list will display a LWChannel option. Selecting this will allow you to add a channel number and activate fader On/Off button pushes from a fader with that LWChannel loaded.

You can use the fields to select colors for the buttons. The primary colors can be entered by name – for example Red, Green, Blue. You can create subtler color shades by using RGB values in the hexadecimal format:

&HBBGRR

For example:

&H00FF00 – would produce green

You can also change the Button State to On or Off, and turn flashing On or Off if desired. The button caption (text displayed in the button) can also be modified using this action.

The Button Name should be used very carefully as it will change the internal running name of the button making it no longer available to the selected Action and/or qualifier. Use the Button Caption property to change the text on the

Stack Item Editor

Group Name

Event Name

Qualifier Name

Qualifier Description

User Panel Button

Use Named Button

Button Name
(Format: PanelName.ButtonName)

Use Button Address

Host IP

Button Module

Button Number

Key State

Back Color On

Back Color Off

Fore Color

State

Flash

Button Caption

Set Button Name

Trigger if False rather than True

button, not Button Name. Button Name does not change the text in the button, but rather the actual name of the button within the system.

Send Email

Use this action to send an email. Please note that in order for this action to work the master email server settings must be configured for the server. This is described elsewhere in this document, and can be found under the File menu of PathfinderPC Server. As long as the server has the correct Email server settings defined, this action will send an email.

Enter the Address, subject, and message into the appropriate fields. Multiple email addresses may be assigned by creating multiple email Actions, or by separating the email addresses by semicolons in the Email Address field. The Email address field must be less than 255 characters long.

The screenshot shows the 'Stack Item Editor' dialog box with the following fields and options:

- Group Name:** [Empty text box]
- Event Name:** Default
- Action Name:** [Empty text box]
- Action Description:** [Empty text box]
- Send Email:** [Selected radio button]
- Email Address:** [Empty text box]
- Subject:** [Empty text box]
- Message:** [Empty text area]
- Buttons:** Cancel, Apply

Run Script

Use this action to run a custom script. The script file must exist within the PFSScripts folder in PathfinderPC Server.

The screenshot shows the 'Stack Item Editor' dialog box with the following fields and options:

- Group Name:** [Empty text box]
- Event Name:** Default
- Action Name:** [Empty text box]
- Action Description:** [Empty text box]
- Run Script:** [Selected radio button]
- Script Name:** [Empty text box]
- Buttons:** Cancel, Apply

Send User Command

This Action will send a textual message out a Protocol Translator port. Select the Translator and enter the command to send.

Escape characters can be used to look for non-typable characters as follows.

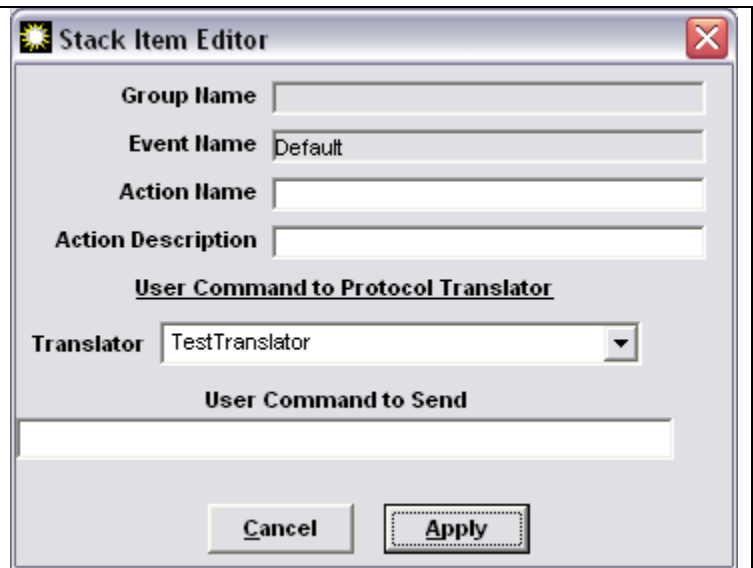
\\ = \

\cr = Carriage Return

\lf = Line Feed

\t = Tab

%xx where xx is a hexadecimal number represents an Ascii character of the number represented by xx (PFS 3.58 and later only)



The screenshot shows a dialog box titled "Stack Item Editor" with a close button in the top right corner. The dialog contains several input fields and a dropdown menu:

- Group Name:** An empty text input field.
- Event Name:** A text input field containing the word "Default".
- Action Name:** An empty text input field.
- Action Description:** An empty text input field.
- User Command to Protocol Translator:** A section header above a dropdown menu.
- Translator:** A dropdown menu currently showing "TestTranslator".
- User Command to Send:** A large empty text input field.
- Buttons:** "Cancel" and "Apply" buttons at the bottom.

Memory Value

The Memory Value Action allows you to create and assign values to memory slots and matrixes. PathfinderPC Server has 1000 memory slots and 99 matrixes that are available for use by Stack Events and custom scripts. Each memory slot consists of a slot name and a text value. Each Matrix consists of a name and a grid with columns and rows.

To assign a value to a memory slot, select the **Write To Memory Slot** option and fill in the **Memory Slot Name** and **Memory Slot Value** fields. When the action is run, PathfinderPC Server will look through its memory slots for a slot with the name in the Memory Slot Name field. If it finds one, it will assign the value to that slot. If it does not find a field with the specified name, it will use the next available memory slot and assign that slot the name and value from the action. If the Delete Memory Slot checkbox is selected, PathfinderPC Server will look for a memory slot with the correct name and clear out both the name and value making it available to be reused.

A PathfinderPC Server memory matrix is a table in memory with columns and rows. To create one, select the **Add To Memory List** option, and define a name for the list. PathfinderPC Server will then look for a list with the selected name. If it cannot find one, it will assign the name to the next available list. Data is added to a memory list by entering fields separated by commas.

Dog, Cat, Fight, Dog Wins

Would create a list with 4 columns and would enter the data into the 4 columns. If the data were then retrieved in a qualifier, row 0, column 0 would return the word Dog.

Additional rows can also be added. Selecting the Delete Memory List checkbox will delete the

Stack Item Editor

Group Name

Event Name

Action Name

Action Description

Memory Parameters

Write To Memory Slot

Memory Slot Name

Memory Slot Value

Delete Memory Slot

Add To Memory List

Memory List Name

Data to Add To the List

Remove Row

Delete Memory List

memory list. Selecting the Remove Row check box will try and find a row with all columns left to right that match, and it will delete that row. So if in the value field you enter Dog, Cat

All rows that have the first two columns with Dog and Cat will be removed.

Memory slots and matrixes are also automatically synchronized between clustered servers, so that on a failover the server that takes over the event system has the correct information in its memory slots and matrixes.

SA Command

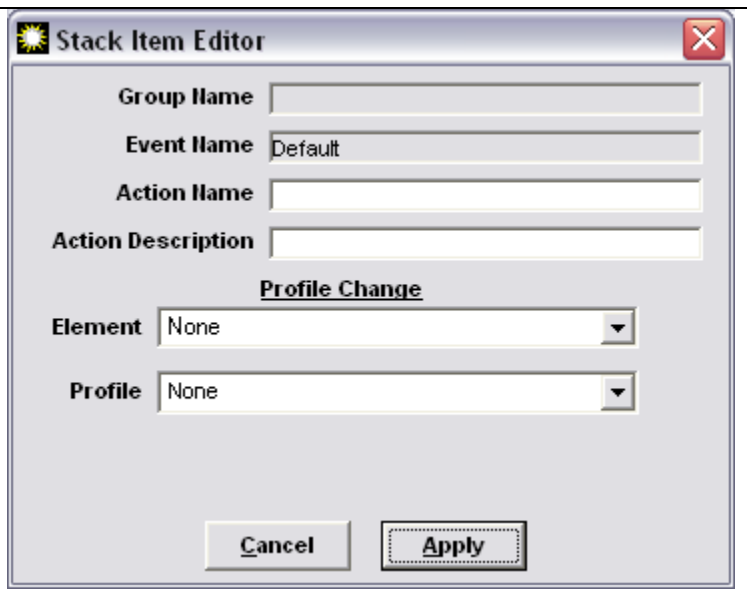
The SA Command Action runs a Software Authority command as defined in the Software Authority Protocol Translator Protocol document installed in the PathfinderPC Server folder. Enter the command to run in the defined field.



The screenshot shows the 'Stack Item Editor' dialog box. It has a title bar with a sun icon and a close button. The dialog contains several text input fields: 'Group Name' (empty), 'Event Name' (containing 'Default'), 'Action Name' (empty), and 'Action Description' (empty). Below these fields is a section titled 'SA Command To Run' with a large text input field. At the bottom are 'Cancel' and 'Apply' buttons.

Element Profile Change

This action will change an Element Profile. Select the Element and Profile from the drop down lists that you wish to load when the Action is run.

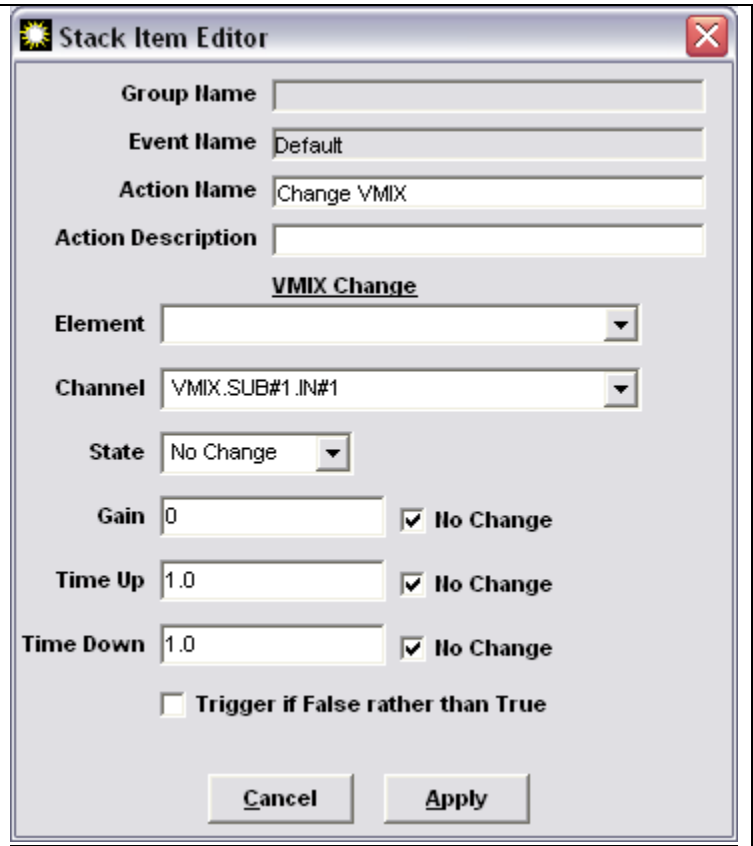


The screenshot shows the 'Stack Item Editor' dialog box. It has a title bar with a sun icon and a close button. The dialog contains several text input fields: 'Group Name' (empty), 'Event Name' (containing 'Default'), 'Action Name' (empty), and 'Action Description' (empty). Below these fields is a section titled 'Profile Change' with two dropdown menus: 'Element' (containing 'None') and 'Profile' (containing 'None'). At the bottom are 'Cancel' and 'Apply' buttons.

VMIX Change

This action will make a change to an Element VMIX Channel. Select the Element and Channel from the drop down lists. You can then turn the Channel On or Off by modifying the state field. In addition, you can deselect the No Change checkbox for the Gain, TimeUP, and or TimeDown fields, and enter new values for those fields. When the action is run, any of these fields that is not set to “No Change” will be updated to the assigned value on the selected Vmix channel. In this way the action can turn VMIX channels on and off, change their gain, and/or adjust their fade times.

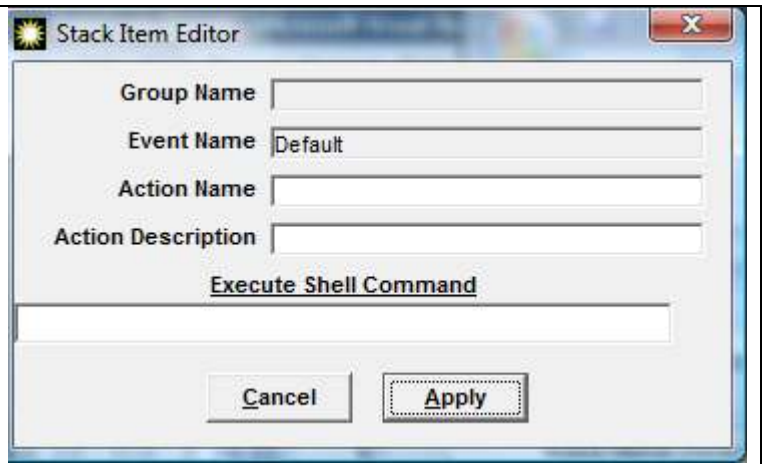
This feature is only enabled with a PRO license.



The screenshot shows the 'Stack Item Editor' dialog box. It has a title bar with a sun icon and a close button. The fields are: Group Name (empty), Event Name (Default), Action Name (Change VMIX), and Action Description (empty). Below these is a section titled 'VMIX Change' with a dropdown for Element (empty) and a dropdown for Channel (VMIX.SUB#1.IN#1). There are three rows of controls: Gain (0) with a checked 'No Change' checkbox, Time Up (1.0) with a checked 'No Change' checkbox, and Time Down (1.0) with a checked 'No Change' checkbox. At the bottom is an unchecked checkbox labeled 'Trigger if False rather than True'. There are 'Cancel' and 'Apply' buttons at the bottom.

Execute Shell Command

This action will make will execute a call as if it was from the windows command line. It can be used to launch other applications, batch files, or windows scripts.



The screenshot shows the 'Stack Item Editor' dialog box. It has a title bar with a sun icon and a close button. The fields are: Group Name (empty), Event Name (Default), Action Name (empty), and Action Description (empty). Below these is a section titled 'Execute Shell Command' with a large empty text area for the command. At the bottom are 'Cancel' and 'Apply' buttons.

Protocol Translators

Protocol Translators allow the server to communicate with other controllers as if it were a specific type of router. Essentially it allows PathfinderPC Server to pretend to be some other manufacturer’s router as far as a controlling device is concerned. The PathfinderPC Server Protocol Translator Engine translates commands from one type of router protocol to the routers which are set up on the system. Create a protocol translator by clicking on the Protocol Translator tab and clicking Add Translator.

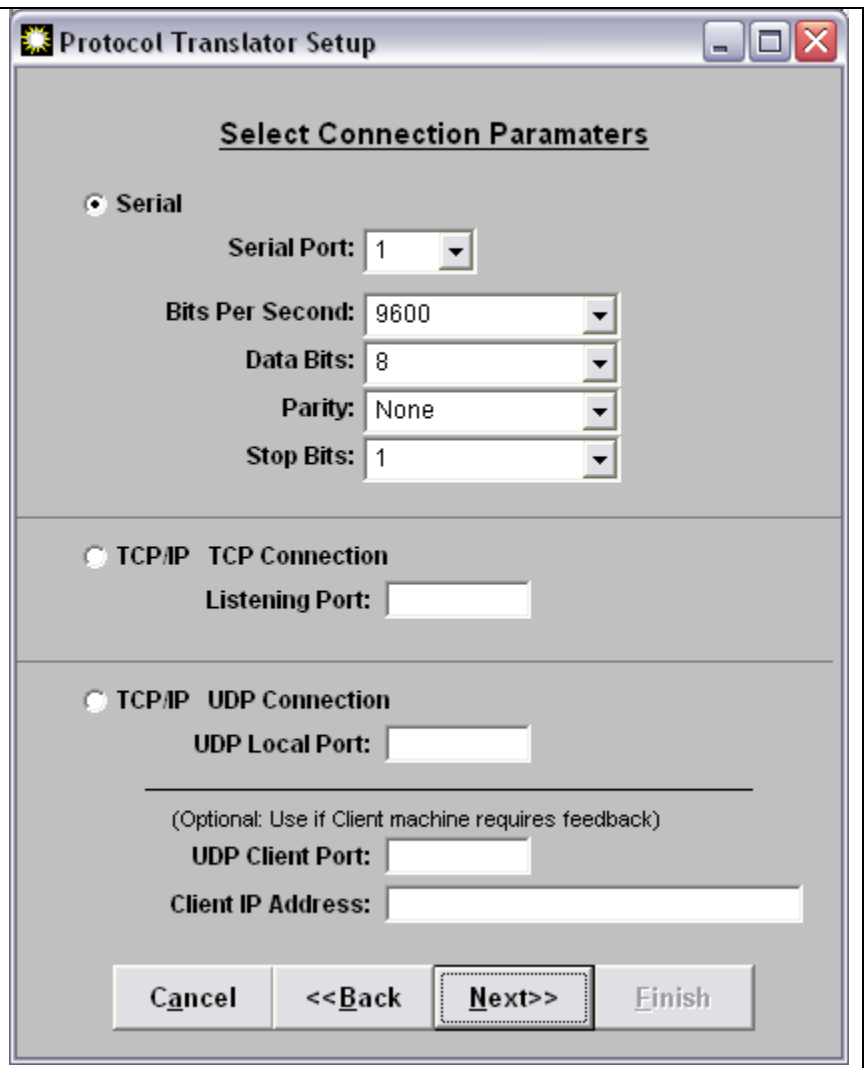
Select the protocol to use for the translation.

Click next



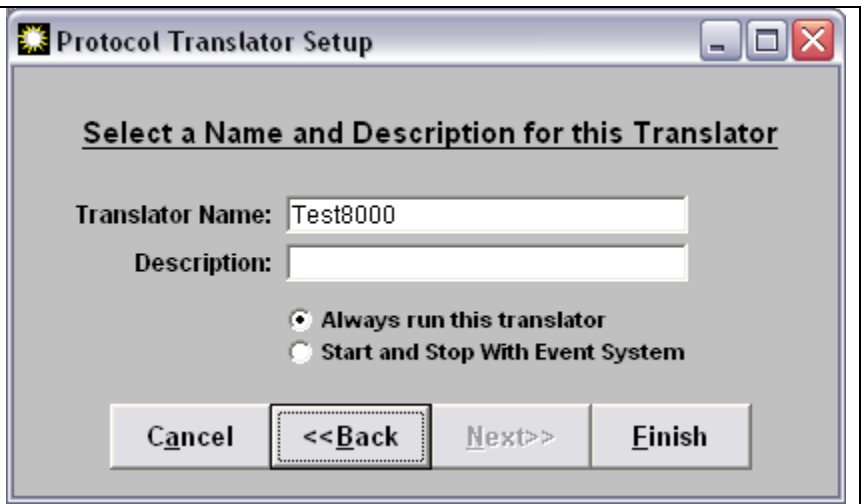
Decide how the communication will take place – serial port, TCP, or UDP connection, and enter the appropriate settings.

Click Next Again



Finally **add a name and description** for the translator.

In addition on this final screen you can select whether the translator will be open and responding to commands and connection attempts whenever the server is running, or only when the event system is running. This is useful in a Cluster where you may only want the translator answering on the machine that currently owns the event system. See the section later in this documentation on Software Authority Bridges for a more detailed explanation on this option.



Supported Translators

Software Authority Protocol

The Software Authority Protocol is a simple text based protocol for controlling any of the routers set up in PathFinderPC Server. This is one of the most simple and versatile control protocols in the system for controlling PathfinderPC resources as it is designed specifically for PathfinderPC. It has help available by typing a simple question mark and all commands are simple text commands. There is a text document included in your PathfinderPC Server installation folder outlining this protocol. This translator is often used by automation systems that can send simple text commands over TCP or Serial to control PathfinderPC routers.

Pro-Bel General Router and Pro-Bel General Switcher Protocol Translators

The Pro-Bel protocol translators only provide certain basic functions at this time. You will want to determine whether the controlling device you will be using with this translator requires the General Router or the General Switcher Protocol. For Prophet Router control, use the Pro-Bel General Router Protocol. Also see the section on the SAGenericBridge application for a discussion for how to use this in a clustered system with controllers that can only use a serial port.

Generic Protocol

The generic protocol does not do any translation at all. It is used to send and read commands into Stack Events. Stack Events can be set up to perform certain actions when certain user defined textual commands are received through the translator, and Stack Event Actions can be set up to send user defined commands back to a controlling device.

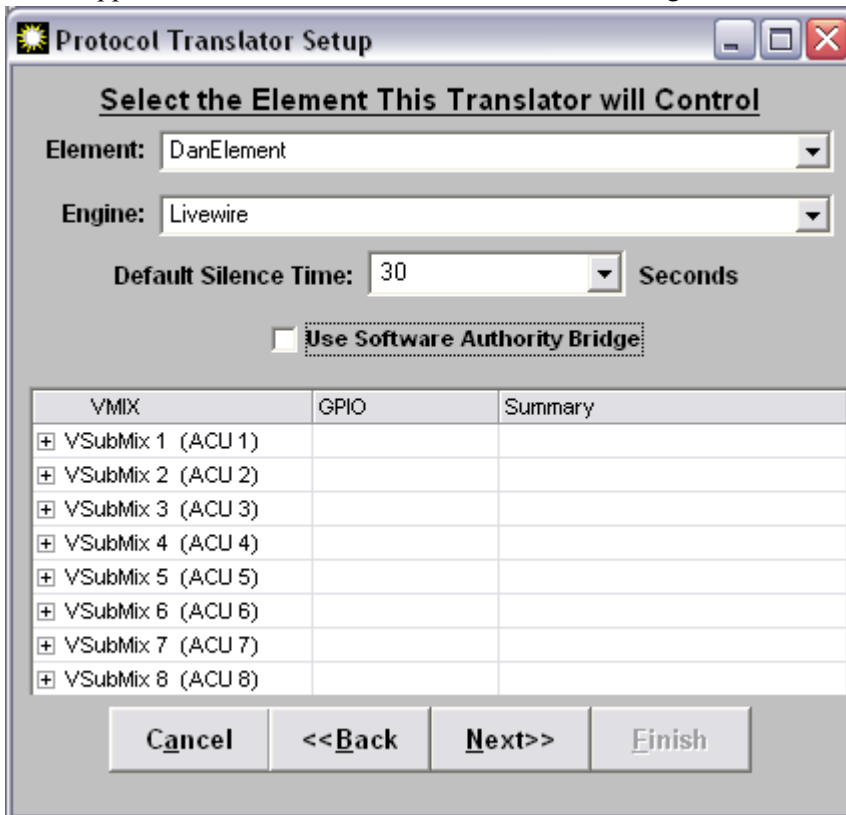
Sine ACU-1 Prophet Version

The Sine ACU-1 Prophet Version translator is designed to act like the Prophet version of the Sine Systems ACU-1 switcher. It converts the commands usually destined to an ACU-1 to a bank of VMIX channels on an Axia

Element and a bank of GPIO ports on the Axia network. An ACU-1 is comprised of 8 stereo Channels of audio inputs and a single stereo output. Any Channel on the input side can be turned on, and it will fade up to the predetermined level over a predetermined time period. If multiple channels are turned on, the ACU-1 will mix them together to the stereo output. In that way the ACU-1 acts both as a switcher and as a mixer. When the Channel is turned off, it will fade down over a predetermined period of time as well. In addition, there are 8 output relays (GPOs) which can be tripped for each ACU-1. Finally there is a bank of 16 Optos (GPIs) that can be wired up for any of the Input Channels. The Optos for any Channel that is turned on are then passed through to the controlling device. Each Prophet can address up to 4 ACU-1s on a single serial port since the serial cables can be chained.

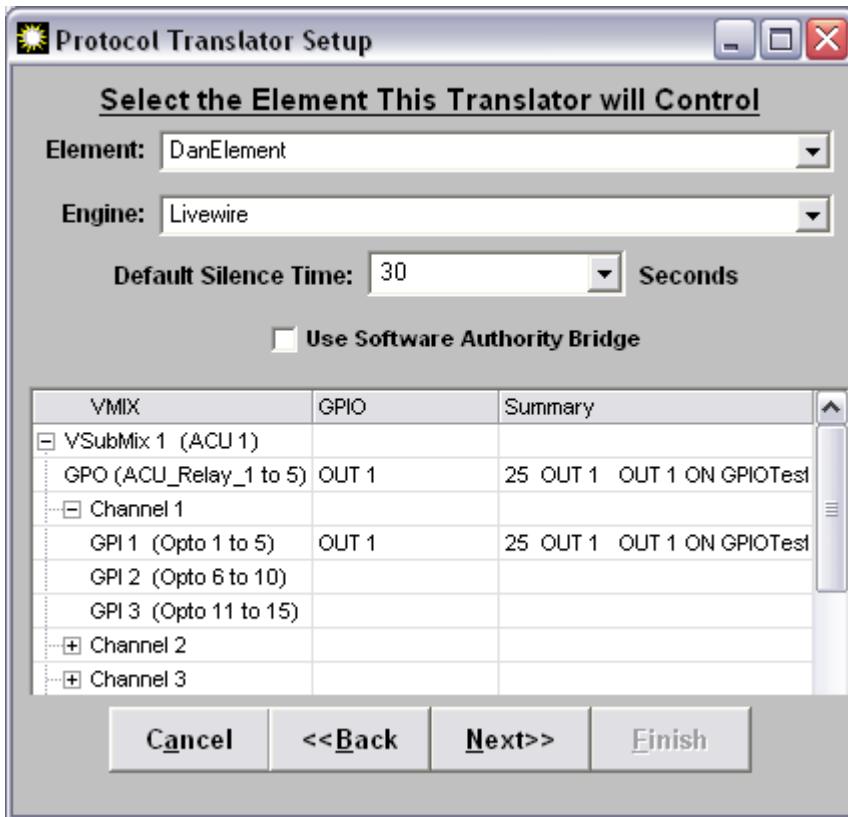
So how is this functionality mimicked in the Axia world? Each Axia Element has a 40 Channel Virtual Mixer that can be accessed through its web page. These 40 Channels are split up into 8 sub-mixers with 5 Channels each. There is an output from each of the 8 sub-mixers as well as a main mix out. Thus the ACU-1 Protocol Translator makes each of the VMIX sub-mixers on an Element equivalent to a single ACU-1. Prophet can address all of the VMIXers on a single Element through a single serial cable connection to PathfinderPC Server. Please note at this point in time Prophet has a limit of 4 ACUs per serial connection so only the first 4 VMIXers on any Element are actually addressable by Prophet. Since the Axia VMIX sub-mixers are 5 channel rather than 8, only the first 5 channels in any ACU setup in Prophet should be used. In addition you can assign 1 GPO port (5 closures) on the Axia network to each VMIX/ACU. Finally you can assign up to 3 GPI ports on the Axia network (15 closures) to each VMIX Channel. These closures will then be passed through as Optos to Prophet just as if they were ACU closures if the VMIX Channel is on. With a serial cable between the Prophet and PathfinderPC Server, PathfinderPC can create the same functionality as a bank of ACU-1s.

In order to do this the Sine ACU-1 Prophet Version Protocol Translator requires an extra configuration screen which appears before the Protocol Translator Name assignment screen.

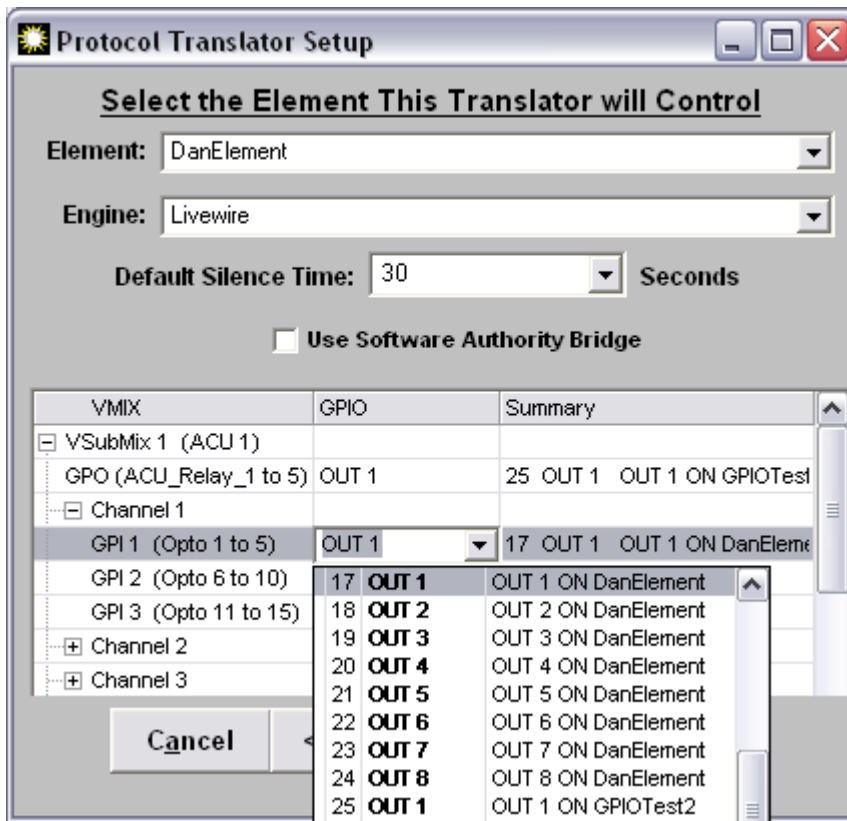


An Element must be present in a PathfinderPC System GPIO Router before you will be allowed to create an ACU-1 Translator. Select the Element and Engine that is associated with the Element to use for this bank of VMIX/ACU translations. The VMIX sub-mixers will be shown in the expandable tree below. Silence detection commands for the various channels will also be passed through to the controlling device based on the timeout value entered.

Expanding the trees in the bottom half of the window will supply the interface for associating GPIO closures with the VMIX sub-mixers and Channels.



Click in the GPIO column for each VMIX sub-mixer GPO (ACU-Relay) to assign the GPO port (5 closures) to be fired as relays by the ACU relay translation. Then select up to three GPI ports (15 closures) on the Axia network for each VMIX Channel. These GPIs will be passed through as Optos to the controlling device (Prophet) if the VMIX Channel is on and the closure occurs.



Finally if the controlling device will communicate directly with the PathfinderPC Server protocol translator using a serial port, leave the “**Use Software Authority Bridge**” option unchecked. If you are going to use the Software Authority ACU Bridge application to convert the serial communications to TCP and thereby enable communications to both servers in a cluster, turn this option on. See the section on Software Authority bridges for more information.

Finish adding the translator into the system and use the Prophet to test turning VMIX channels on and off.

BTools Switcher

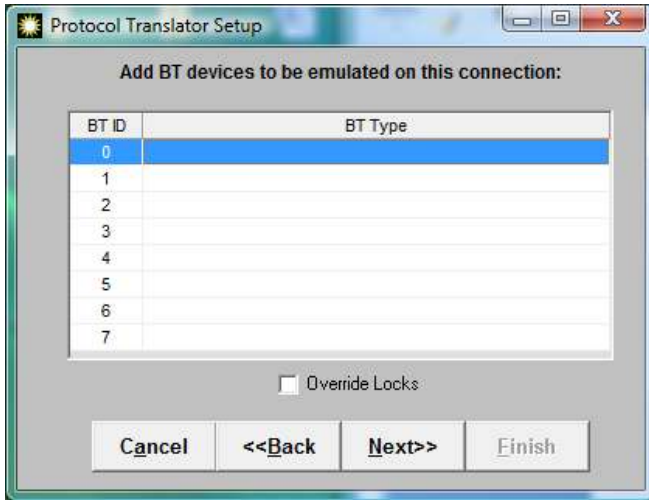
The BTools Switcher protocol translator is designed to emulate a variety of Broadcast Tools switchers. This translator handles switching and GPIO only. It does not currently handle mixing like many of the actual BTools boxes do. The first step to using the BTools translator is to create an audio virtual router that is the size of the BTools box you are trying to emulate. The boxes this translator emulates are:

- ACS8x2 - 8 sources, 2 destinations
- ACS8x4 – 8 sources, 4 destinations
- SS16x4 – 16 source, 4 destinations
- SS12x4 – 12 sources, 4 destinations
- SRC32 – GPIO only

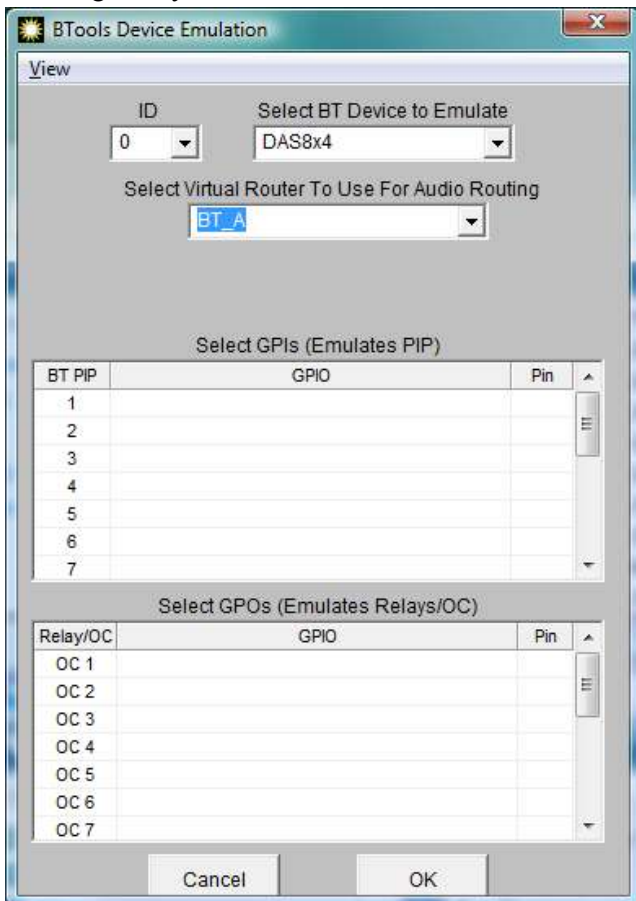
The Administrator needs to create a virtual router for each BTools box they wish to emulate and import the correct sources and destinations from the main audio router that the translator will control. One thing to remember is that up to 8 BTools devices can be chained on the same serial port. Therefore in the protocol

translator you can assign up to 8 BTools box emulators on the same protocol translator. The Administrator will need to create a separate virtual audio router for each instance of a Btools box you want to emulate. See the section in this document on creating virtual routers for details. The Administrator does not need to deal with GPIO in the virtual routers. This is handled in the protocol translator setup. Since the SRC32 is a GPIO only device, a virtual router is not required to emulate this either.

After configuring the serial/tcp settings, the following screen will appear in the wizard.

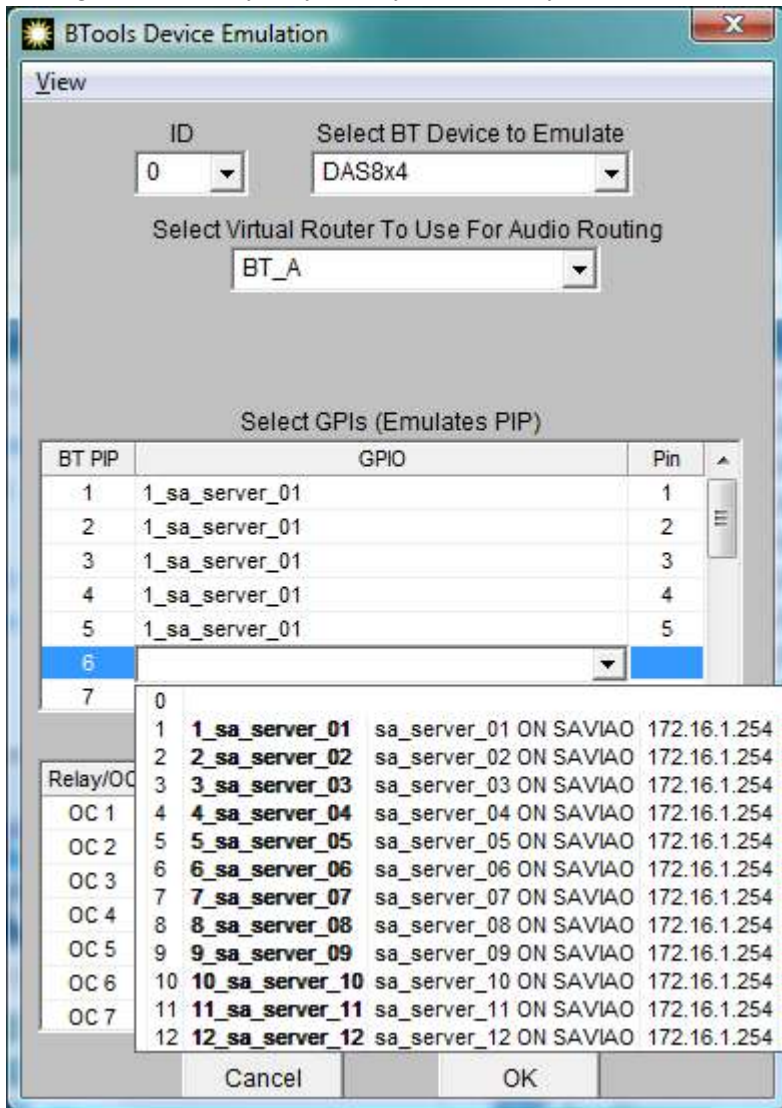


Clicking in any row will allow the Administrator to add a BTools emulator. The following window will appear.

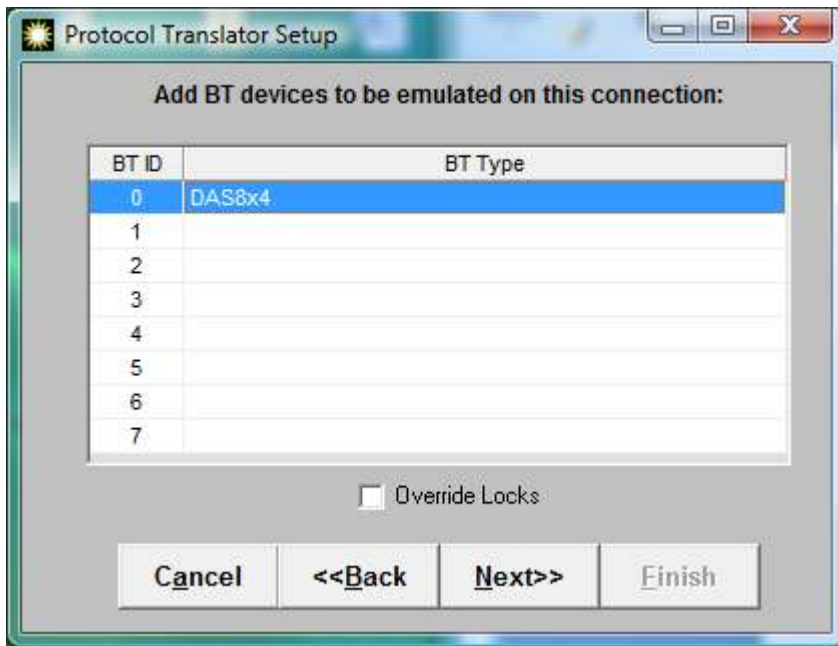


First select the type of device to emulate in the top box. As you change these device types, the number of available GPIOs and GPOs will change to match the capabilities of the emulated BTools device. Next select the virtual router that was created for this emulator.

Finally, fill in the GPIO information to define the GPIOs on the network that will be used to emulate the PIPs and Relays for this device. You will notice as you select a GPIO, the next 5 will automatically get filled in with the remaining pins for that port. This is to speed the data entry process, but you can change the settings at will. Hitting the delete key on your keyboard on any row in the GPIO section will clear that row out.

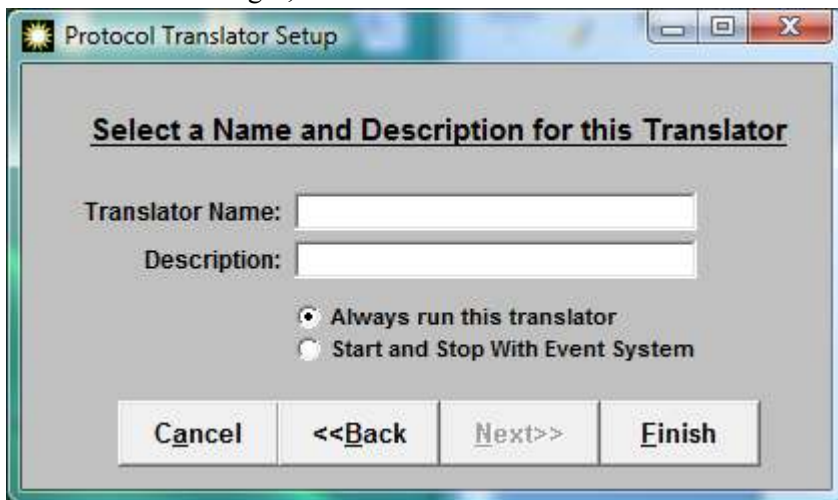


After clicking OK, this device will appear in the correct ID slot on the master list for the translator



You can fill in as many or as few as you like for each translator. If you have 1 automation system that needs to access multiple emulated BT devices, they can all be included in the same translator and addressed with a single serial port. If you have multiple automation system, each which must address a single emulated BT device, then create multiple translators on different serial/tcp ports, and put one emulated device in each translator. The configuration depends on the requirements.

The “Override Locks” checkbox will allow the translator to make the route changes whether the destinations are locked or not. This applies to user level locks. System level locks are destinations where the device itself will not allow route changes, and these cannot be overridden. Click Next to display the last configuration screen.

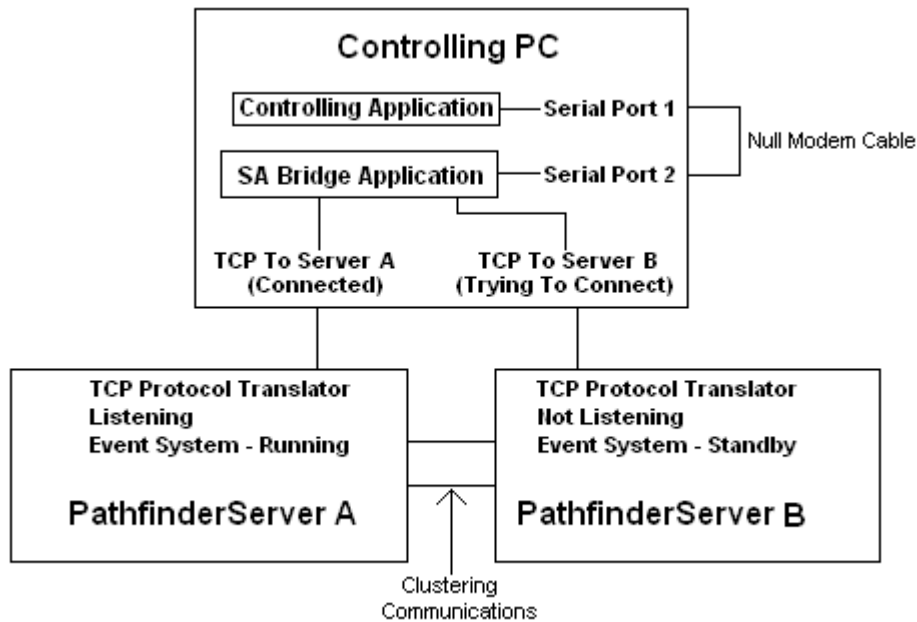


Enter the name and description for the translator and select whether it should run constantly or follow the event system. Finally, click Finish to create the translator.

An automation system that is set up to control BTDevices should now be able to address the Axia system through Pathfinder as if it were a BTDevice. The closures should be able to be tripped and signaled, and turning channels on and off should trigger the appropriate route changes in Pathfinder.

Software Authority Bridge Applications

Serial communications present an interesting dilemma to a high availability environment. Specifically, a standard RS-232 port can not be split effectively. Even if the two listening servers are clustered in a way such that only one has its serial port open at any given point in time (as PathfinderPC can be set up to do), creating an RS-232 splitter cable causes electrical problems that often causes communications problems in the RS-232 protocol. As a result Software Authority has added two bridging applications to its software suite. These applications take a serial port signal and convert it to TCP. Furthermore the bridging application includes two TCP connections so it will attempt to connect to multiple servers in a cluster.

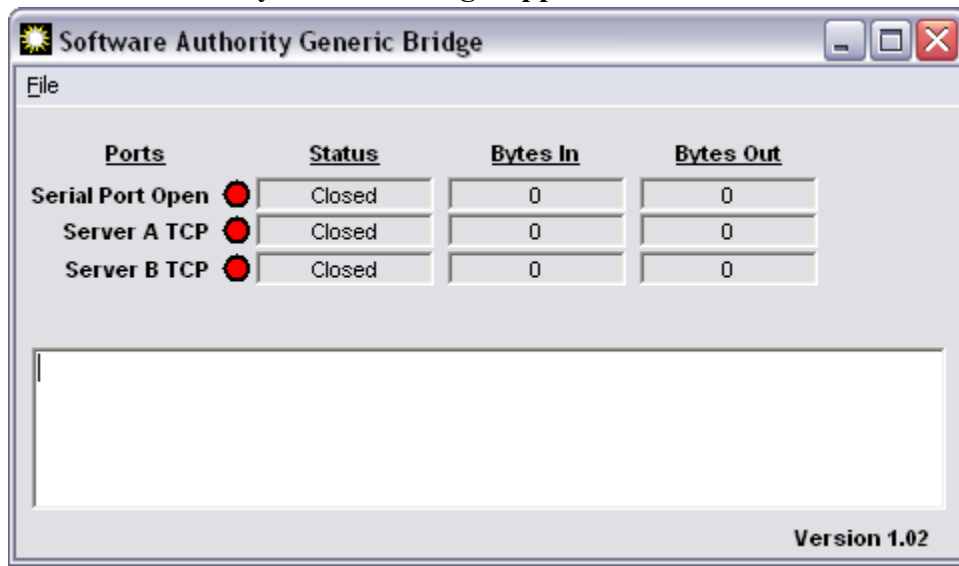


The first of these applications is the Software Authority Generic Bridge. The Generic bridge application passes all data to and from the serial port to whatever TCP connection can be established with the PathfinderPC Servers. The second of these applications is the Software Authority ACU-1 Bridge. This bridge works in the same manner as the Generic Bridge except that it is optimized for the Sine Systems ACU-1 Prophet version protocol. This protocol is very verbose with polling commands every 20 milliseconds. To reduce this traffic this bridge application uses a much less verbose event driven system to maintain the status between the PathfinderPC Servers and the bridge. Then the bridge application stores the correct responses for the polls and answers the polls on the serial port directly rather than forwarding each poll request on to the servers. The ACU version also has a screen that shows detailed information about the current state of the VMIX, GPI, and GPO channels involved in the ACU translation.

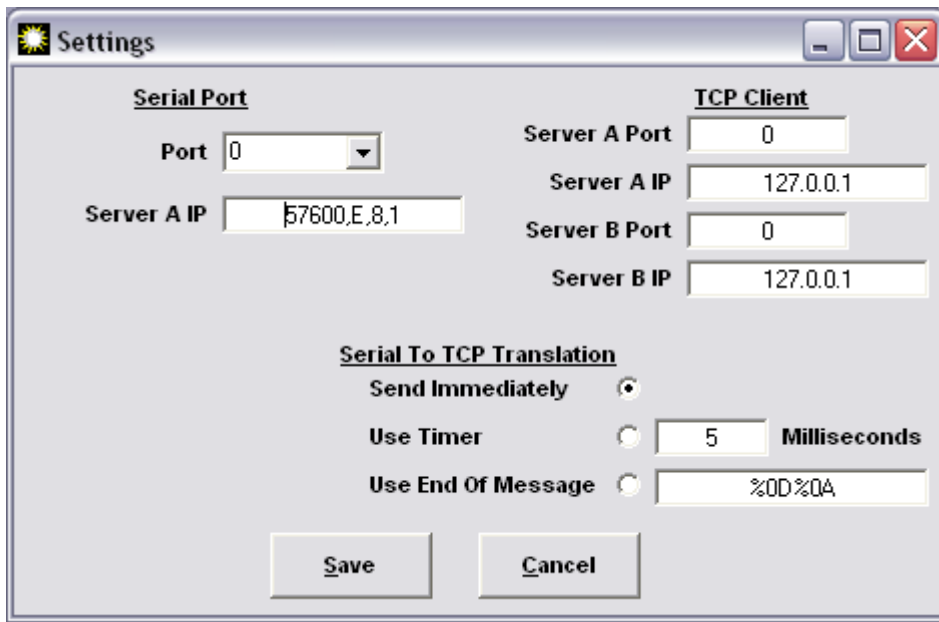
To use either bridge download the appropriate installer .msi file from the Software Authority web site and install it on the machine that is to handle the bridge application. This can be a dedicated computer or in many cases it can be the original controlling computer. In several installations we have used the bridge in the background of a Prophet automation or routing system. The Prophet uses Serial Port 1 on the computer and the bridge uses Serial port 2. Then a simple null modem cable between the ports allows the Prophet to think it is addressing a different device. See the diagram above. The bridge then converts the traffic to TCP and sends it on to the PathfinderPC Servers.

Before you use the bridge application you need to set up the Protocol Translator in Pathfinder. For example if you are going to use the bridge on a Prophet Router to convert the Probel protocol from serial to TCP, add a Probel translator in PathfinderPC and select a TCP port rather than a serial port. When you create the translator in PathfinderPC Server, be sure to select the option to make the translator stop and start with the event system on the last screen of the wizard. This will ensure that the translator is only open on the Server in the cluster that currently owns the event system. If you do not select this option the bridge will be able to connect to both servers at the same time and will send duplicate information to both servers, and both servers will try to act upon those commands. After the installation is complete and the protocol translator is set up in Pathfinder, run the bridge application. Let's look at the configuration and usage of the Generic bridge application first.

Software Authority Generic Bridge Application



This is the main screen of the bridge application. The indicator dots will be either red indicating that the port is not open or connected, or green meaning the port is open and connected. The Status Field shows the current Status of the connection. The Bytes In and Out fields show a byte count which resets when it reaches a certain size, but is a good indication that traffic is flowing through the bridge properly. The text window at the bottom of the screen displays event messages. When you start the application for the first time, the configuration screen will also be displayed. This screen is also available through the File Menu.



This screen allows you to define the serial, TCP, and translation settings for the bridge. Select the serial port number to use as well as the serial port settings. The serial port settings are entered in the format: Baud, Parity, Data Bits, Stop Bits

Valid settings for Parity are:

- E = Event
- O = Odd
- S = Space
- N = None
- M = Mark

Valid Settings for Data Bits are:

- 4
- 5
- 6
- 7
- 8

Valid Settings for Stop Bits are:

- 1
- 1.5
- 2

So a complete serial port settings field might look like:
57600,N,8,1

For 57600 baud, No Parity, 8 Data Bits, and 1 Stop bit

Next enter the TCP Port and IP address for the Primary and Secondary PathfinderPC Servers into the TCP Client fields.

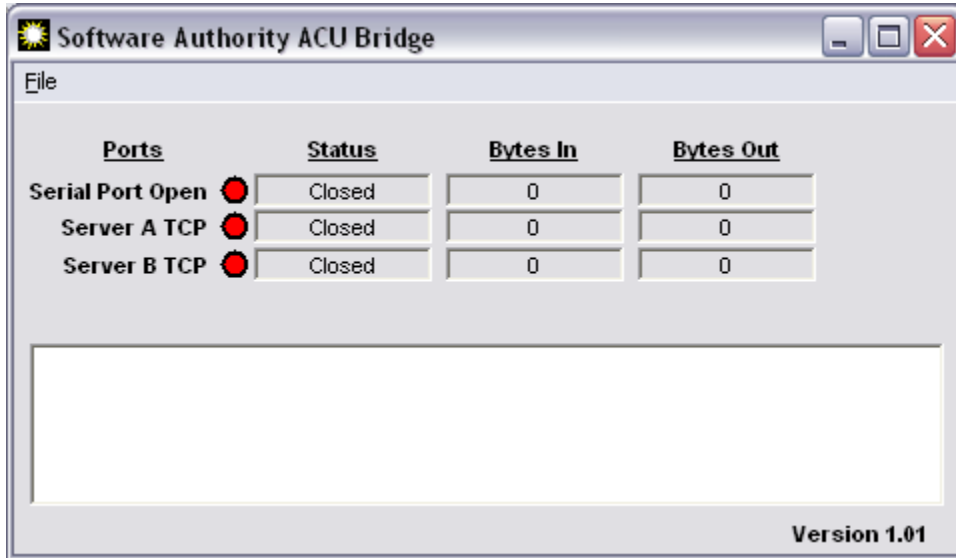
Finally select the translation settings. The translation settings define how the serial data will be packaged into TCP packets to send to the PathfinderPC Server. The first option, Send Immediately, forwards the information as soon as it receives it. While this may sound ideal, it can actually be very inefficient. The computer is fast enough to know about every byte that comes into a serial port, and this option will send each byte as its own TCP packet with any overhead involved in the TCP header and trailer information. So while this can work for certain situations, it is not the most recommended setting.

The second setting, Use Timer, allows you to set a millisecond counter. This will hold any data that comes in for the specified number of milliseconds while waiting for additional data, and then package all of it as a single TCP packet. This will be more efficient than the previous setting as long as the controlling application does not consider the delay to be a problem.

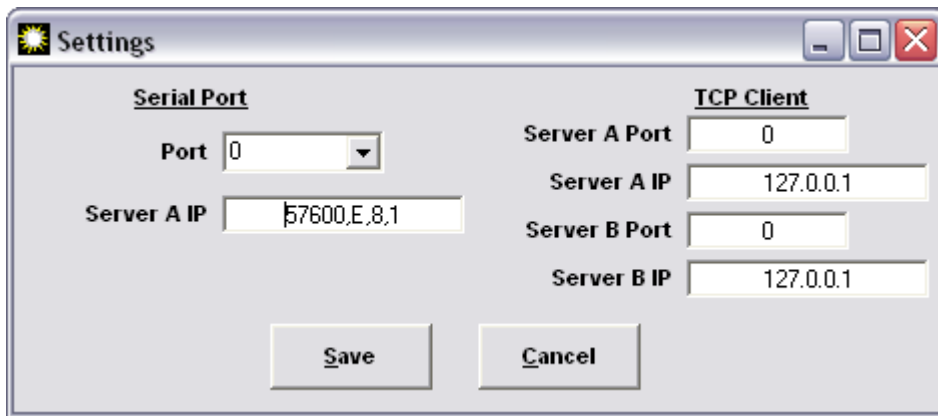
The last setting, Use End Of Message, allows you to define a text message which represents the end of a complete message. Many communications protocols have a certain set of bytes that are used to indicate the end of a message. This can be entered in the text box for this option, and the software will then wait until it sees the end of message characters before packaging the complete message and sending it on. To indicate non-printable characters use the percent sign with a 2-digit hexadecimal number to represent the Ascii characters. For example %0D%0A equates to a carriage return followed by a line feed. Since the percent sign is used as an escape character, if it is used in the end of message it also must be escaped (%25). This last option is the most efficient if an end of message string exists in the protocol to be used. You may need to try a number of these settings before finding the correct one for a given bridge configuration.

Select Save and the translator will start with the new settings. You should see the serial port open and attempts to connect to the servers. When successful connections are made, the indicator dot will turn green. It is normal and desirable that only 1 of the two TCP connections will be connected at any point in time. As stated earlier, the protocol translators on the PathfinderPC Server should be set up to only listen on the Server that currently owns the event system. This guarantees that only one server is processing the commands at any point in time. You should also see data counters changing on the main screen showing successful data flow. If you do not, you may need to use the File Menu and select Settings to rework the configuration until you get it to work correctly.

Software Authority ACU-1 Prophet Version Bridge Application

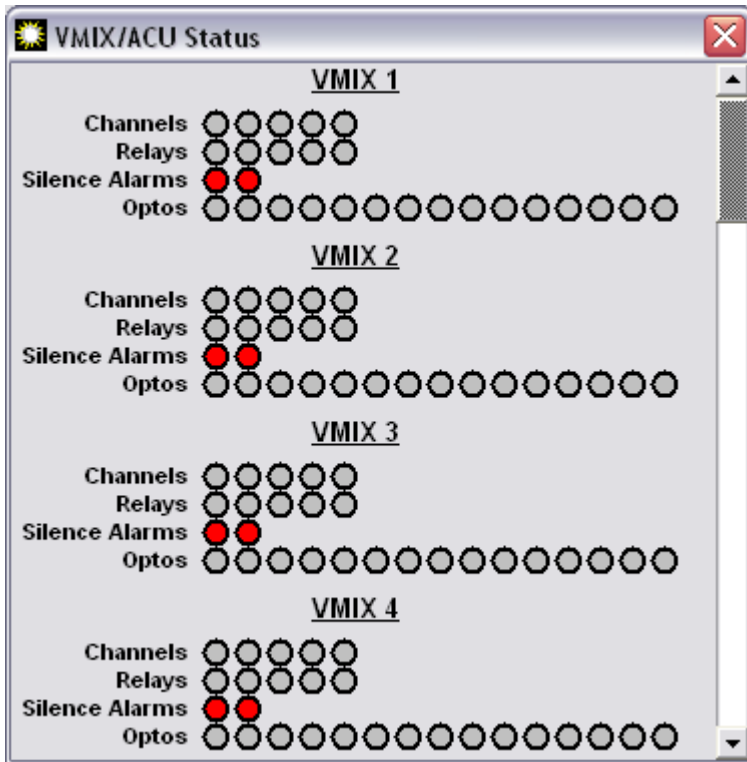


The ACU-1 Bridge application works and is set up in exactly the same fashion as the generic bridge application. The only exception is that the configuration screen for this bridge does not include the message packaging settings. The protocol is known and defined, and the bridge automatically takes care of this for you.



In order to use the ACU-1 Bridge application, it is important to set the Protocol Translator properly in PathfinderPC Server. Be sure to select the Sine ACU-1 Prophet Version translator and set it up on a TCP port. Also be sure to turn on the checkbox to Use Software Authority Bridge. If this is not turned on, the proper status information will not get exchanged between PathfinderPC Server and the bridge. Finally be sure to set the Translator to start and stop with the event system. This will ensure that only one server at a time in the cluster is processing commands. When you have completed setting the serial port and TCP client settings in the Bridge application, save your changes and watch the bridge start. Be sure data is flowing through the data counters and the serial port is opening properly, and the TCP connection is establishing a connection with the server that currently owns the event system.

The ACU Bridge also has a very useful additional screen available through the File Menu of the Bridge. From the File Menu select View Status.

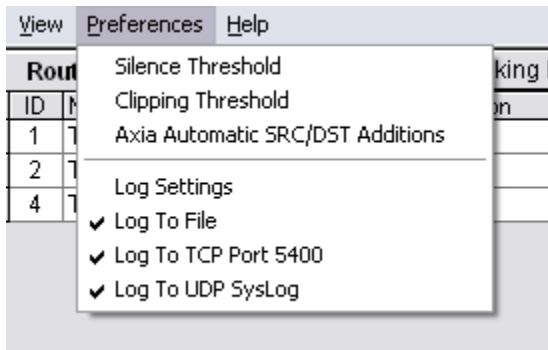


This screen shows the status of all VMIX channels, GPIs, GPOs, and Silence Alarms that are being translated with the ACU-1 Translator. The indicator dots will change color to display VMIX Channels that are on, GPIs (Optos) that are Low, GPOs (Relays) that are Low, and silence alarm settings. This display screen provides an intermediary step to show the states PathfinderPC Server is reporting. These are the states that will be reported through the serial port to polling requests from the ACU controller (Prophet).

To test failover between clustered PathfinderPC Servers for the Bridge Application, shut down PathfinderPC Server on the primary server that owns the event system. When the Event System on the secondary server starts up and comes out of Standby, its protocol translator port should also open, and you will see the green dot appear on the secondary server TCP connection on the bridge. Then when the Primary Server is restarted and takes over the Event System, the connection dot should move back to the Server A TCP Connection on the bridge.

Logging

One of the best keys to troubleshooting any advanced system is detailed log information. And therefore a great deal of time and energy has been devoted to the PathfinderPC Server logging system. PathfinderPC Server's logging system is accessed through the Preferences menu.



There are three places to which log data may be sent by PathfinderPC Server, and each may be selectively turned on and off by clicking the appropriate menu items. The **Log To File** menu item will create a new log file each day and store it in the Syslogs folder located within the PathfinderPC Server folder. The **Log to TCP Port 5400** menu item will send log commands in the standard syslog format to any clients connected to the TCP port 5400, and the **Log to UDP Syslog menu item** allows you to send the log entries to a centralized syslog capture computer. This option also sends the messages in standard syslog format.

The syslog log format is an Internet standard and there are many available capture and data manipulation tools for standard syslog data. Some are even free and/or open source tools available on the Internet for Syslog capture and data manipulation. For an example see <http://www.kiwisyslog.com>. By using a standard log capturing format, the PathfinderPC Server logs are open to all of the myriad of tools available on the Internet for the manipulation of this data. The UDP syslog also allows you to send syslog data from a variety of devices to the same server, and thereby combine all of a facility's log information. For example if you are using a clustered PathfinderPC installation, you could have the syslog server capture the data from both servers in the same location according to the message time stamps and sources. See the instructions for the syslog capture tool for more details. Both the TCP and UDP ports send data in the standard syslog format. The File based option is similar but contains some additional information for each log message, and therefore does not strictly adhere to the syslog format.

Let's look at the format of the log messages in both the text files and the TCP/UDP syslog format. First we will look at the text files created by the PathfinderPC Server log system. Each log message is posted to a single line within the text file, and is made up of the following columns of information separated by tabs:

MillisecondCounter

Date of Message

Time of Message

ServerIPAddress

Message Severity

Message ID Number

Message Source

Remote Device Message Data

Let's look at each of these columns in detail.

Millisecond Counter

The Millisecond counter shows the result of a call to the operating system which returns the time in milliseconds since the machine has been booted. Because of the variable in the operating system used to hold this value, it will sometimes return a negative value, but it can be used to get more accurate differences in time between events.

Date of Message

Date the message was generated

Time of Message

Time the message was generated

ServerIPAddress

The IP address of the PathfinderPC server which generated the message

Message Severity

A number from 0 to 7 representing the severity of the message

0 = Emergency

1 = Alert

2 = Critical

3 = Error

4 = Warning

5 = Notice

6 = Informational

7 = Debug

Message ID Number

Each message type within the PathfinderPC Server System is given an ID number. A user who is analyzing the logs can import them into Excel or Access or another log analysis tool and search for particular types of events. Each ID number is 4 digits. There are several sections of messages and each uses a different first digit. Therefore the 1000 through 1999 ID messages belong to one section and the 2000 through 2999 belong to a different section of messages. The sections and each message number along with a description of the message are listed later in this document.

Message Source

This lists the source device or engine in the system which generated the message. This could be a variety of different items from a particular router number within the server to the event engine to the clustering system, etc. This source could have several elements separated by colons. For example a route change event might have a source that looks like

Router:2:TestGPIO

This means that the source of the message is router 2 in the system which is also named TestGPIO.

Remote Device

If the message involves a remote device the device will be listed in this column. For example on an Axia network this column might list the IP address and port of a particular Axia node that to which the log message refers.

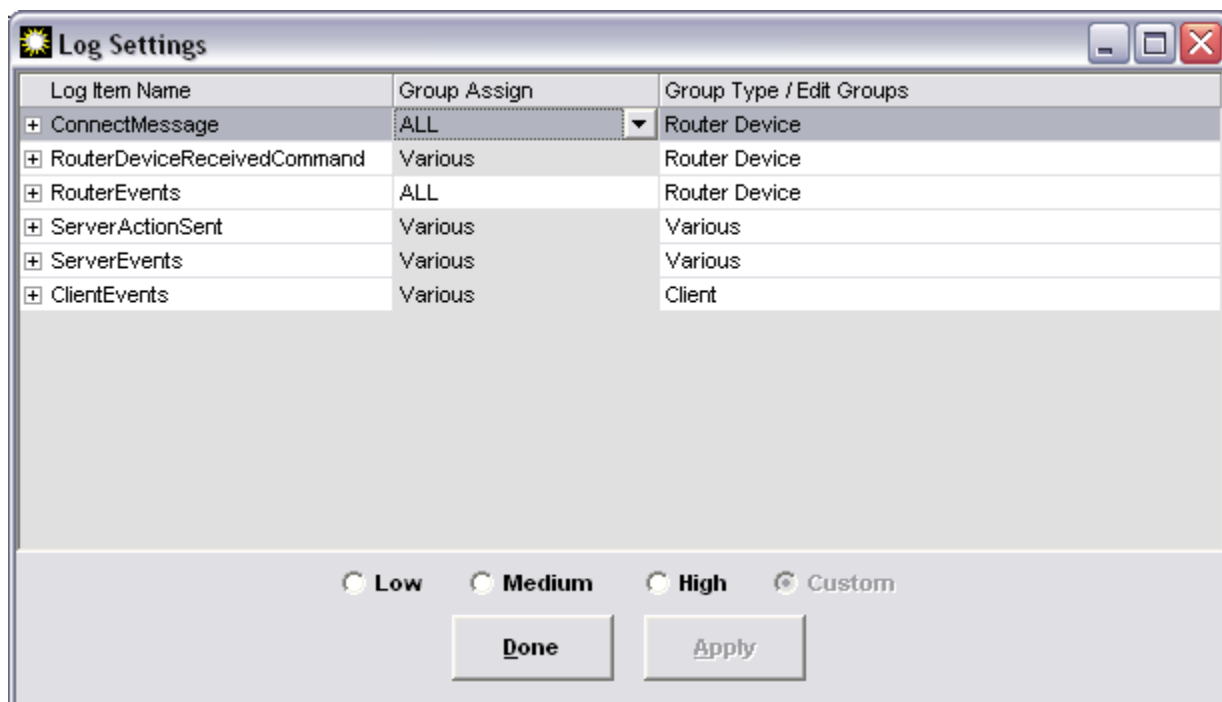
Message Data

This column contains the textual information for the specific log message.

Except for the MillisecondCounter, the TCP and UDP syslog messages contain the same information except that they are forced to comply with the standard syslog format. See <http://tools.ietf.org/html/rfc3164> for a description of this format. Again there are a host of available syslog capture tools for capturing, displaying, and working with the syslog messages sent to these ports.

Logging Configuration

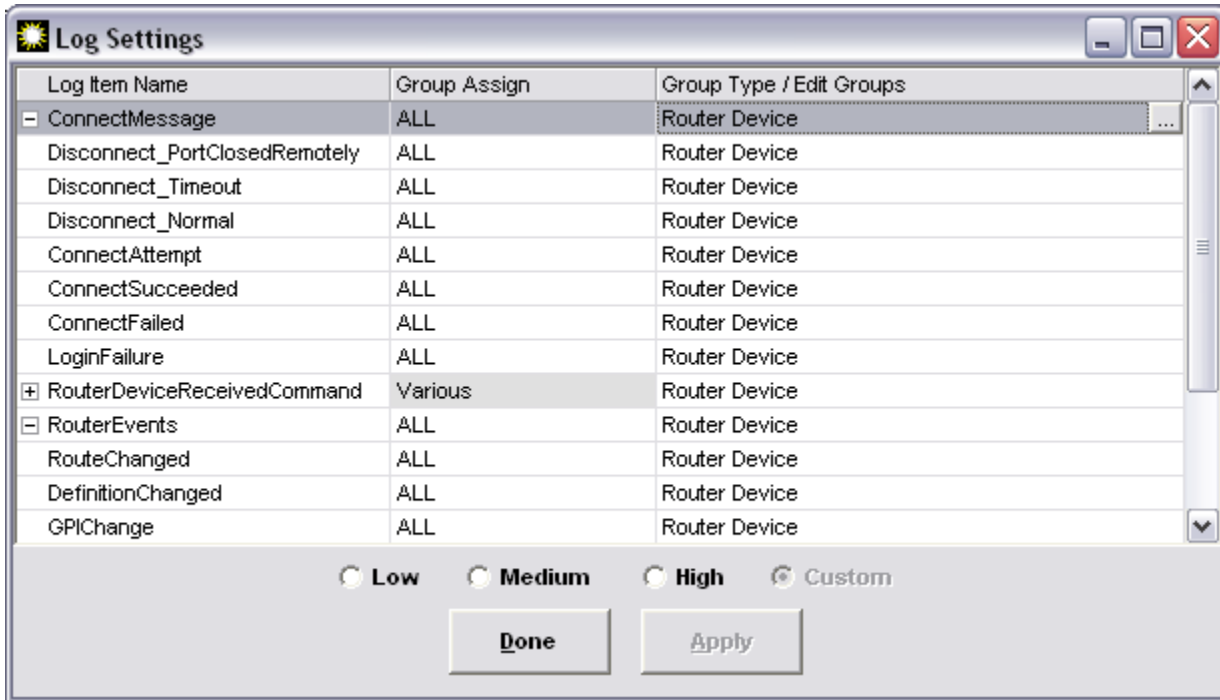
The PathfinderPC Server system allows you great flexibility in deciding which messages should be logged and which should be ignored. Some messages are really more for debugging a certain problem, and due to the volumes of log information they generate should only be turned on in those circumstances. The interface for making these configuration settings is available by clicking on the Log Settings menu item under the Preferences menu. This will display the following configuration screen.



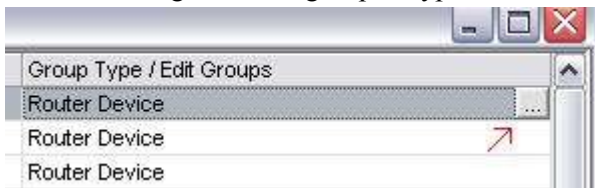
First you will notice that we have provided a Low, Medium, and High setting. This allows the general user to automatically get a basic logging setup quickly. For many users the medium log setting will be all they ever need. The High setting should only be used for extreme debugging as it turns every log message in the system

on, and will generate a tremendous amount of information and can also cause the server to respond slower in some high volume situations.

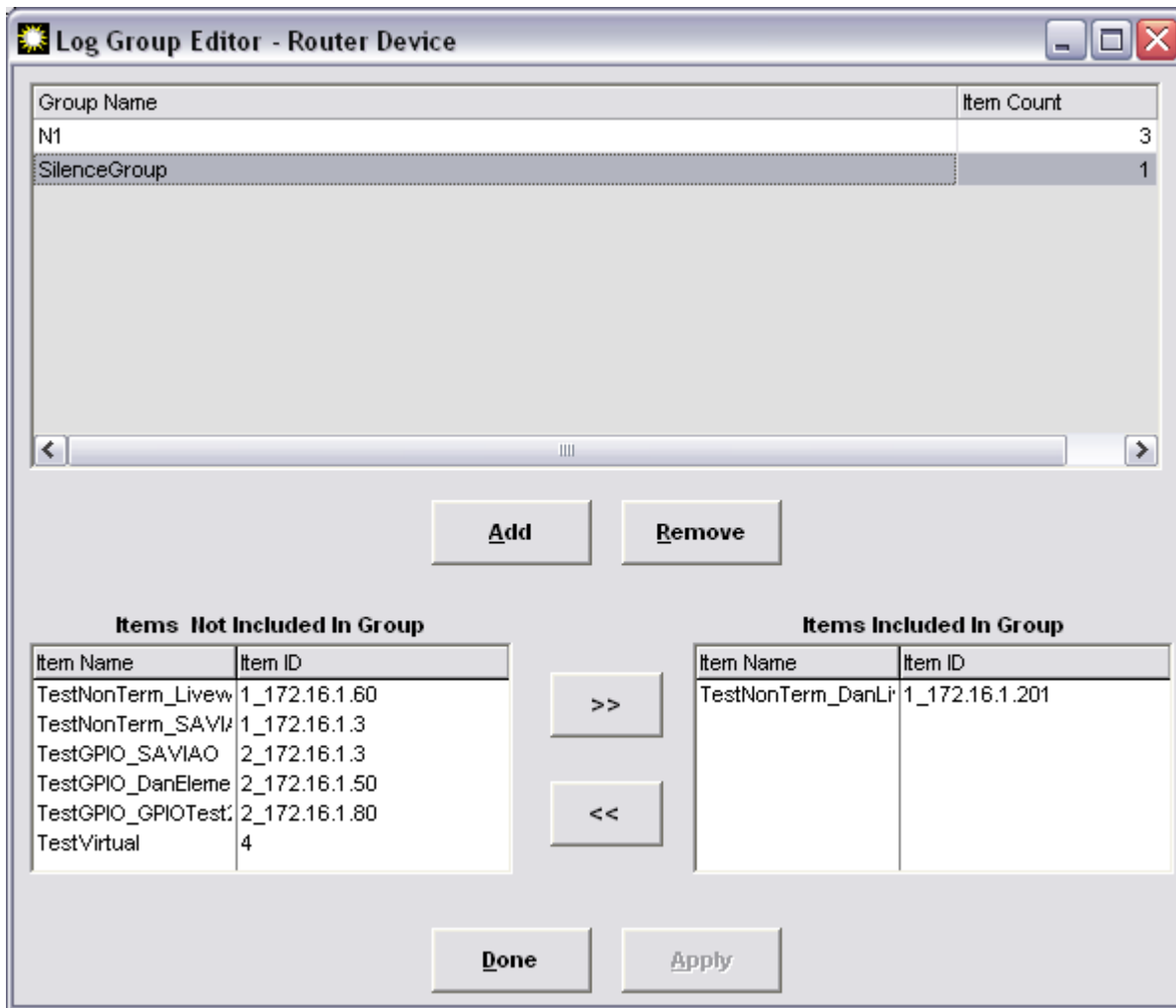
More detailed logging can be set up by manipulating the settings in the top part of the screen. There are six logging sections each of which may be expanded to manipulate the log types that belong to each section. Click on the plus sign to expand each section and make changes to the various log messages that the section of messages can generate.



Each section or log message type within the section can be set to None which means the event will not be logged at all, or ALL which means all events of that type will be logged. Selecting a drop down on the section title (with the + or – next to it) will set that type of logging for all messages within that section. In addition many of the messages have a group type. If the **Group Type/Edit Groups** column has a group type name other than None, that means that a custom group can be created and selected for items that share that group type. For example the Connect messages all have a group type of Router Device. If you click in the **Group Type/ Edit Groups** column, a button will appear that can be selected to create a custom group of the type in the column. In this case we are creating a custom group of type Router Device.



Clicking the button will open the group editor for Router Devices.



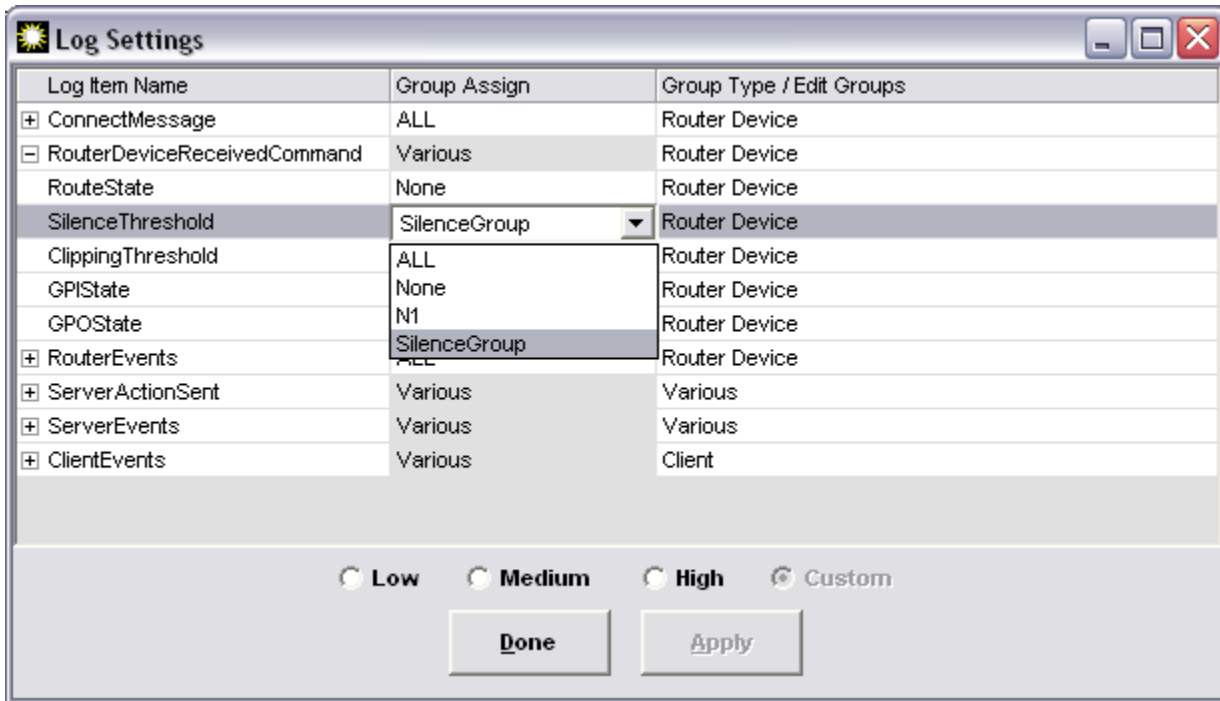
We can now create groups of Router Device objects and give that group a name. This group will then be available in the log message selection drop down in addition to the All and None selections in the previous screen. You can use this to create a group of specific objects (Axia nodes) for which you want to collect log information.

For example, let's say you want to log all of the GPI changes that take place on a couple of nodes. If in the log settings screen you set logging of GPI changes to ALL, every GPI change on the Axia network will get logged. Instead you could create a custom group and include only the GPIO devices you wish to log in that group. Then for the GPI log message, select that custom group name from the drop down list instead of All or None.

To create a group from the log group editor Click Add in the editor and then edit the name of the group in the top box. Select the items to include in the group from the list using the shift and control keys to select multiple lines if desired, and then use the buttons to move the items in and out of the included list. Then Apply the changes.

Please note that there are several types of groups that can be created depending on the log message type to which the group will be applied. This is indicated by the **Group Type/Edit** Groups column on the previous screen and the last part of the title bar of the group editor. For example Router Device group types will allow you to create custom groups of router device objects. On the other hand the Stack Event Group Type, which you will find on

some of the Server Events log messages, allows you to create custom groups of Stack events to log. The kind of group you are editing in the group editor will define the items that can be included. When you open the group editor from a Stack Event column type, the options to include will not be router devices, but stack events in the system. These custom groups will then be available in the main Log Settings screen under the group assign column for any log message of the correct Group Type.



Using these methods you can customize your logging messages to your facility’s individual needs.

Next we will list and describe all of the log message sections and their underlying messages. This next section will also list an ID number for each message. This is the ID number used to uniquely identify the particular type of log message in the logs, and can be used when analyzing the log to find certain kinds of events. The ID Numbers are all 4 digit numbers, and the first number represents the master section of log message. For example all messages that fall under the Connect Message section are between 1000 and 1999. All messages that fall under the RouterEvents section get IDs in the range between 3000 and 3999.

ConnectMessage Section

ID – 1000 to 1999

This section of messages logs connect and disconnect events to pieces of gear within the system. For example, these messages will create log entries if the server is having difficulty communicating with a particular Axia node. The server uses a heartbeat with each piece of equipment in the system, and will try to disconnect and reconnect with the unit if the heartbeat fails or the port is closed.

Disconnect_PortClosedRemotely

ID – 1001

Group Type – RouterDevice

Description – This message indicates that the TCP connection to the remote device was closed by the remote device.

Disconnect_Timeout

ID – 1002

Group Type – RouterDevice

Description – This message indicates that there was a timeout waiting for a valid response from the remote device.

Disconnect_Normal

ID – 1003

Group Type – RouterDevice

Description – PathfinderPC Server has disconnected from the remote device as a normal part of operations.

ConnectAttempt

ID – 1004

Group Type – RouterDevice

Description – PathfinderPC Server is attempting to connect to a Router device.

ConnectSucceeded

ID – 1005

Group Type – RouterDevice

Description – PathfinderPC Server has successfully established a connection with the specified device.

ConnectFailed

ID – 1006

Group Type – RouterDevice

Description – PathfinderPC Server was unable to establish a connection to the specified Router Device.

LoginFailure

ID – 1007

Group Type – RouterDevice

Description – The remote device has rejected a login request.

RouterDeviceReceivedCommand Section

The RouterDeviceReceivedCommand Section of log messages logs commands that are received from the external equipment. This section should be used with caution and is generally used for debugging purposes only. This is because this section logs all return communications from the external equipment including the responses to polling data. While PathfinderPC Server relies primarily on event driven commands from such gear as the Axia nodes, it does also send some polling requests both as a heartbeat to make sure the gear is still responding properly, and to double check that no changes have been missed. The return from these commands will get reported in this log section, and so on a busy network there will be a huge amount of traffic through this section of logs, and it can even slow the server response time down. So this section is used more for debugging than daily use. If you want to log when routes and GPIOs change (which is a good idea), it is better to accomplish this through the RouterEvents section as that section only writes log entries when one of the statuses changes rather than noting every communication response.

RouteState

ID – 2001

Group Type – RouterDevice

Description – Logs each command from a device that has to do with the state of the device's routing table

Silence Threshold

ID – 2002

Group Type – RouterDevice

Description – Logs each command relating to silence and/or audio present from the external device.

Clipping Threshold

ID – 2003

Group Type – RouterDevice

Description – Logs each command relating to clipping from the external device

GPIState

ID – 2004

Group Type – RouterDevice

Description – Logs each command relating to a GPI state from the external device. This includes responses to periodic GPI state polls.

GPOState

ID – 2005

Group Type – RouterDevice

Description - Logs each command relating to a GPO state from the external device. This includes responses to periodic GPO state polls.

RouterEvents Section

This section of log events describes changes to router states within the PathfinderPC system. It will log routing changes and GPIO changes within the system.

RouteChanged

ID – 3001

Group Type – RouterDevice

Description – This route has changed from the previously known route to the one specified in this log entry.

DefinitionChanged

ID – 3002

Group Type – RouterDevice

Description – Some aspect of the router definition has changed.

GPIChanged

ID – 3003
Group Type – RouterDevice
Description – The specified GPI state has changed.

GPOChanged
ID – 3004
Group Type – RouterDevice
Description – The specified GPO state has changed.

DeviceStateChanged
ID – 3005
Group Type – RouterDevice
Description – The state (usually online or offline) of a specific device has changed.

ServerActionSent

This section of log events will log commands that are sent by PathfinderPC Server.

RouteChangeRequest
ID – 4001
Group Type – RouterDevice
Description – PathfinderPC Server is sending a route change request to a device.

GainChangeRequest
ID – 4002
Group Type – RouterDevice
Description - PathfinderPC Server is sending a gain change request to a device.

GPIChange
ID – 4003
Group Type – RouterDevice
Description – A Request to change a GPI is being sent by PathfinderPC Server. The only GPIs that PathfinderPC can change are driver GPIs.

GPOChange
ID – 4004
Group Type – RouterDevice
Description – A request to change a GPO is being sent by PathfinderPC Server.

GPIStatusRequest
ID – 4005
Group Type – RouterDevice
Description – PathfinderPC Server is asking for the current status of GPIs on a specific device. Use with caution as this will log any polling requests as well.

GPOStatusRequest

ID – 4006

Group Type – RouterDevice

Description - PathfinderPC Server is asking for the current status of GPOs on a specific device. Use with caution as this will log any polling requests as well.

GPISubscription

ID – 4007

Group Type – RouterDevice

Description – This log message is not used at this time.

GPOSubscription

ID – 4008

Group Type – RouterDevice

Description – This log message is not used at this time.

LoginRequest

ID – 4009

Group Type – RouterDevice

Description – A login request is being sent to a specified device.

CheckNames

ID – 4010

Group Type – RouterDevice

Description – A request to get the names assigned to sources and destinations on a specific device is being sent.

RouteStatusRequest

ID – 4011

Group Type – RouterDevice

Description – PathfinderPC Server is sending a request to get the current routing status of the specified device.

SendEmail

ID – 4012

Group Type – None

Description – PathfinderPC Server is sending an email. The email address that it is sending to is included as part of the log data.

SendUserCommandToPT

ID – 4013

Group Type – Protocol Translator

Description – PathfinderPC Server is sending a user defined command message to a Protocol Translator port (Software Authority Protocol and Generic Protocol only).

SendRouterStateChangeToPT

ID – 4014

Group Type – Protocol Translator

Description – PathfinderPC Server is sending a route state change command to a Protocol Translator port (Software Authority Protocol only).

ServerEvents

This section of log messages pertains to events that occur on the PathfinderPC Server.

LogEngineStarted

ID – 5001

Group Type – None

Description – The Logging engine has started.

EventEngineStarted

ID – 5002

Group Type – None

Description – The Event Engine has started. In a cluster only one server in the cluster will have the Event Engine started at any point in time. See the clustering section of this document for details.

EventEngineStopped

ID – 5003

Group Type – None

Description – The Event Engine has stopped. In a cluster only one server in the cluster will have the Event Engine started at any point in time. See the clustering section of this document for details.

StackEventEngineStarted

ID – 5004

Group Type – None

Description – The Stack Event Engine has started. In a cluster only one server in the cluster will have the Stack Event Engine started at any point in time. See the clustering section of this document for details.

StackEventEngineStopped

ID – 5005

Group Type – None

Description - The Stack Event Engine has stopped. In a cluster only one server in the cluster will have the Event Engine started at any point in time. See the clustering section of this document for details.

ClusterLostConnection

ID – 5006

Group Type – None

Description – The cluster has lost its connection to the other server.

ClusterGainedConnection

ID – 5007

Group Type – None

Description - The cluster has established its connection to the other server.

ClusterLicenseKeyFailure

ID – 5008

Group Type – None

Description – There is an invalid licensing situation between the clustered servers. Each Server in a cluster must have a unique license. If both servers have the same license this log message will result, and proper connection and synchronization will not take place. Contact your reseller to purchase another copy of PathfinderPC Server.

StackEventQualifiersAnalyzed

ID – 5009

Group Type – Stack Event

Description – One of the states involved in the qualifier section for the stack event has changed, so the stack event is being analyzed to see if Actions need to be run.

StackEventQualifiersMet

ID – 5010

Group Type – Stack Event

Description – All of the qualifiers for a particular stack event have been met.

StackEventQualifiersNotMet

ID – 5011

Group Type – Stack Event

Description – The specified stack event has been analyzed due to a change in one of the qualifiers states, and the qualifiers have not been met.

RunStackEventQualifiersNotMetActions

ID – 5012

Group Type – Stack Event

Description – The stack event qualifiers have been analyzed and they have not been met, so the actions specified for this condition are being run.

RunStackEventQualifiersMetActions

ID – 5013

Group Type – Stack Event

Description - The stack event qualifiers have been analyzed and they have been met, so the actions specified for this condition are being run.

RunStackEventQualifiersNotMetActionsDelayed

ID – 5014

Group Type – Stack Event

Description – Delayed Actions are being run on a stack event whose qualifiers have been met.

RunStackEventQualifiersMetActionsDelayed

ID – 5015

Group Type – Stack Event

Description - Delayed Actions are being run on a stack event whose qualifiers have not been met.

SilenceDetectEvent

ID – 5016

Group Type – Event

Description – A silence detect event state has changed.

GPIOEvent

ID – 5017

Group Type – Event

Description – A standard GPIO event is running.

ActivateRouteEvent

ID – 5018

Group Type – Event

Description – An Activate Route event is running.

ActivateSceneEvent

ID – 5019

Group Type – Event

Description – An Activate Scene event is running.

UserPanelPropertyChanged

ID – 5020

Group Type – None

Description – A user panel property has changed.

RunScript

ID – 5021

Group Type – Scripts

Description – A script is being run.

RunScriptCommand

ID – 5022

Group Type – Scripts

Description – A single script type command is being run

ScriptTimerTripped

ID – 5022

Group Type – Scripts

Description – Script Timer completed count down

ScriptTimerGetName

ID – 5022

Group Type – Scripts

Description – Name of script timer requested

ScriptTimerSetName

ID – 5022

Group Type – Scripts

Description – Name of a script timer was set

ScriptTimerGetInterval

ID – 5022

Group Type – Scripts

Description – The interval of a script timer was requested

ScriptTimerSetInterval

ID – 5022

Group Type – Scripts

Description – The interval of a script timer was set

ScriptTimerGetEnabled

ID – 5022

Group Type – Scripts

Description – Checked to see if a Script timer is enabled

ScriptTimerSetEnabled

ID – 5022

Group Type – Scripts

Description – Script Timer was enabled

ScriptTimerGetAction

ID – 5022

Group Type – Scripts

Description – The Action that will be run upon a script timer countdown was requested

ScriptTimerSetAction

ID – 5022

Group Type – Scripts

Description – The Action that will be run upon a script timer countdown was set

ScriptTimerClear

ID – 5022

Group Type – Scripts

Description – Cleared a script timer location

LicenseExceptionNumClients

ID – 5022

Group Type – Scripts

Description – More clients than the license allows tried to access this server simultaneously

ClientEvents

This section of events pertains to requests that a client application (usually PathfinderPC Client or Mini) makes of PathfinderPC Server.

RequestLogin

ID – 6001

Group Type – Client

Description – A client application has sent a login request to PathfinderPC Server.

OpenRouter

ID – 6002

Group Type – Client

Description – A Client application has opened the specified router.

CloseRouter

ID – 6003

Group Type – Client

Description – A client application has closed the specified router.

RequestRouteChange

ID – 6004

Group Type – Client

Description – A client application is requesting a route change.

RequestLockChange

ID – 6005

Group Type – Client

Description – A client application is requesting a lock for a particular destination on a router.

AddSceneChange

ID – 6006

Group Type – Client

Description – A client application is attempting to add a scene to the system or update an existing scene in the system.

DeleteSceneChange

ID – 6007

Group Type – Client

Description – A client application is attempting to delete a scene from the system.

ActivateScene

ID – 6008

Group Type – Client

Description – A client application has asked to activate a specific scene.

Create_UpdateVirtualRouter

ID – 6009

Group Type – Client

Description – A client application is attempting to add a virtual router to the system.

OpenMeter

ID – 6010

Group Type – Client

Description – A client application has opened detailed meters.

CloseMeter

ID – 6011

Group Type – Client

Description – This log message is not used at this point in time.

ChangeGain

ID – 6012

Group Type – Client

Description – A client application has requested a gain change.

Add_UpdateEvent

ID – 6013

Group Type – Client

Description – A client application is attempting to add or update a standard event.

DeleteEvent

ID – 6014

Group Type – Client

Description – A client application is attempting to delete a standard event.

OpenUserPanel

ID – 6015

Group Type – Client

Description – A client application is opening a user panel.

CloseUserPanel

ID – 6016

Group Type – Client

Description – This message is not used at this point in time.

PanelActionCommand

ID – 6017

Group Type – Client

Description – An action associated with a user panel is being run.

SoftwareUserPanelButtonPush

ID – 6018

Group Type – Client

Description – A software user panel button is being pushed.

HardwareUserButton

ID – 6019

Group Type – Client

Description – A hardware mapped user panel button is being pushed.

Log Message Custom Group Types

None – No custom group can be created for log messages with this group type. The only available selections for messages with this group type are All and None.

Router Device

Custom Groups can be created and will be comprised of router device objects. On an Axia network this can be used to create a group of Axia nodes. The log messages which use this group will then only occur if the particular devices in the group are involved in the particular log event.

Stack Event

Custom groups of stack events can be created. Stack event log messages that are associated with the named group will then only be logged if they involve the stack events selected in the group.

Event

Custom groups of events can be created. Event log messages that are associated with the named group will then only be logged if they involve the stack events selected in the group.

Protocol Translator

Custom groups of Protocol Translators that exist on the server can be created. Log messages that are associated with the named group will then only be logged if they involve the Protocol Translators selected in the group.

Client

This group allows you to select hardware mapped clients and create IP addresses for other clients and add those to a custom group. The log messages associated with these custom groups will only be logged if the event involves the selected clients.

Clustering

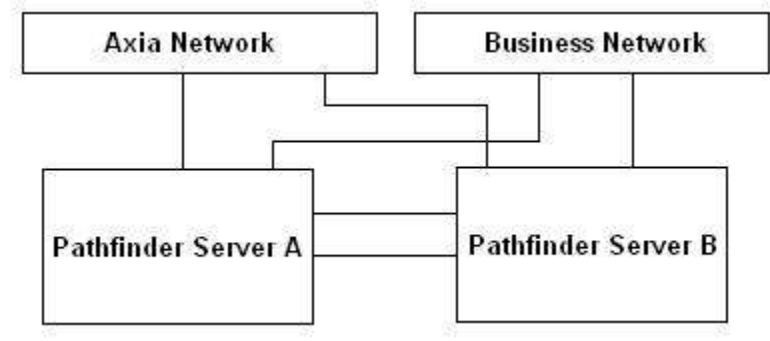
Almost all mission critical operations require redundancy to ensure additional reliability and guaranteed up time. Equipment can break, hard drives can fail, and unforeseen human factors can also play roles in system outages. Thus, it is always important to analyze your system for possible failures and attempt to isolate and mitigate the affects of those failures. To help in this process PathfinderPC includes a clustering option.

PathfinderPC Clustering allows the PathfinderPC Server software to run on two servers. Only clusters of two servers are supported at this time. The servers are made aware of each other and will synchronize database settings and event system changes automatically. This allows both machines to have a continuously accurate set of data. If one or the other server fails, the other will automatically take over critical operations. In addition PathfinderPC Server clustering will automatically load balance incoming PathfinderPC Client connections between the two servers. If one server has a higher load of client connections, additional connections will be redirected to the other server. In this manner PathfinderPC Server clustering offers both reliability and redundancy.

PathfinderPC Server Clustering is an application level clustering solution. This means that a cluster can be created with any operating system that supports PathfinderPC Server. It does not use Microsoft clustering so you do not need the expensive Advanced Server operating systems to create a PathfinderPC Server Cluster. See the System requirements section at the beginning of this document for details and assistance in choosing the PathfinderPC Server operating system. Microsoft Standard Server is the recommended operating system to use for scalability reasons. Clustering is included in the PathfinderPC Server application, but requires a separate PathfinderPC Server license to be purchased for each computer on which PathfinderPC Server is to be run. In fact two servers with the same license will not synchronize properly. They must have different license keys.

Cluster Computer Design considerations

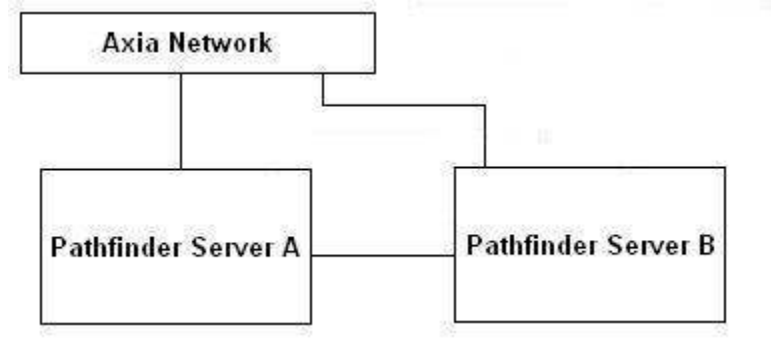
In most cases we recommend installing either 3 or 4 NIC cards into computers that will be used in a PathfinderPC Server Cluster. 1 NIC card in each of the PathfinderPC Servers will be connected to the Axia network. 2 NIC cards will be cross connected with crossover wires for clustering communications. And the fourth NIC card can be connected to the business LAN if desired for PathfinderPC Clients that need to control routing without allowing those PCs direct access to the Axia network. Standard computer networking dictates that each NIC card must be in its own subnet so that the computer knows which card to use when trying to reach any specific address. A default Gateway should only be configured on the NIC card that is connected to a network that has a gateway (usually the business network).



While both servers are continuously connected to and monitoring all of the Axia devices in the PathfinderPC databases, only one server at any point in time will have it's event system running. This is so that you will not have events executed twice. This role falls to the active machine in the cluster with the lowest clustering IP address. We will call this ServerA, and the other server ServerB. In the case that ServerA goes down or the PathfinderPC Server application on that machine is terminated, ServerB becomes the only active member of the cluster and will then take over the Event System role. Thus, it is absolutely critical that each server be able to determine if the other has PathfinderPC Server running and operational. This is why we recommend supplying redundant paths of communication for the clustering data. This ensures that if one of the clustering NIC cards or

cables is severed or goes down for some reason, the servers can still communicate. Otherwise both Servers will think the other has gone down and both will start trying to execute events in the event system, leading to duplicate events. The event system includes both standard events and stacking events.

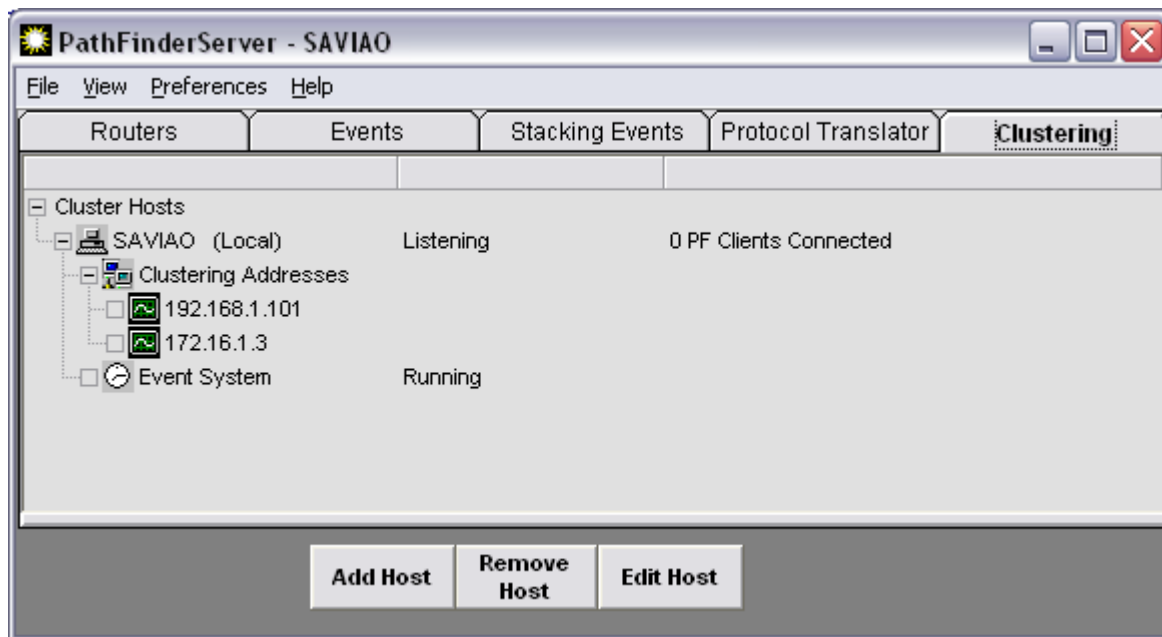
It is possible to create a PathfinderPC cluster with fewer than 4 NIC cards. For one thing if the business network does not need to be connected directly to the PathfinderPC Server then that NIC card can be eliminated. In addition you can set the Axia network to double for backup clustering communications and therefore remove the need for one of the cross-connect cards in each server. In this case only two NIC cards in each server would be necessary.



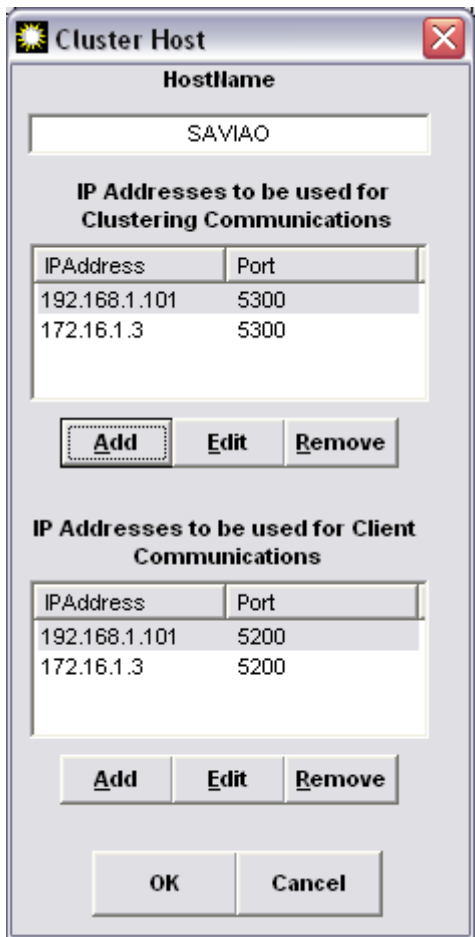
While the first design is preferred, this design is also acceptable.

Creating a cluster

When you first start PathfinderPC Server, it creates a file called Cluster.xml within the PathfinderPC Server folder. The results of this file are displayed when you select the Clustering Tab in the PathfinderPC Server application.

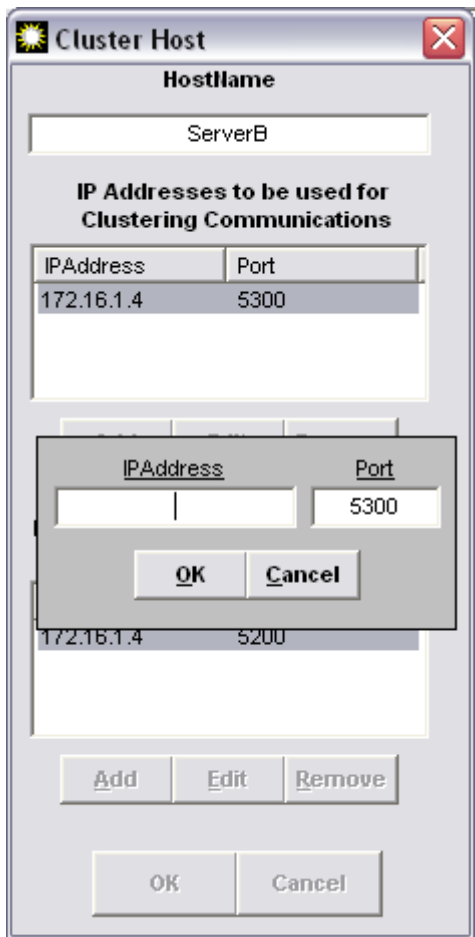


This picture shows the name of the machine on which PathfinderPC Server is running and the various NIC cards that are in the system. By default, all of these NIC cards are enabled for clustering communications. Selecting the host name and clicking edit allows these settings to be fine-tuned.

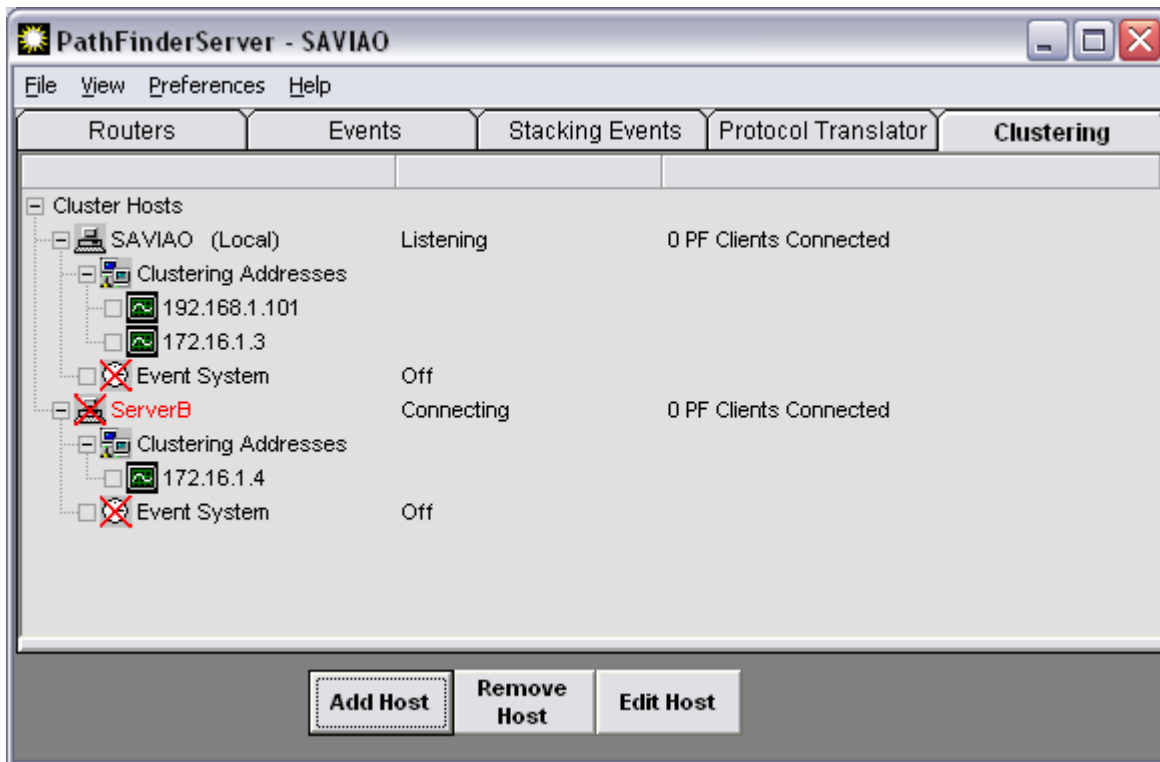


Using this screen you can define which NIC cards are to be used for clustering communications and the order in which they will be used. You can also define which of the NIC cards can be used for copies of the PathfinderPC Mini or Client to communicate with the server.

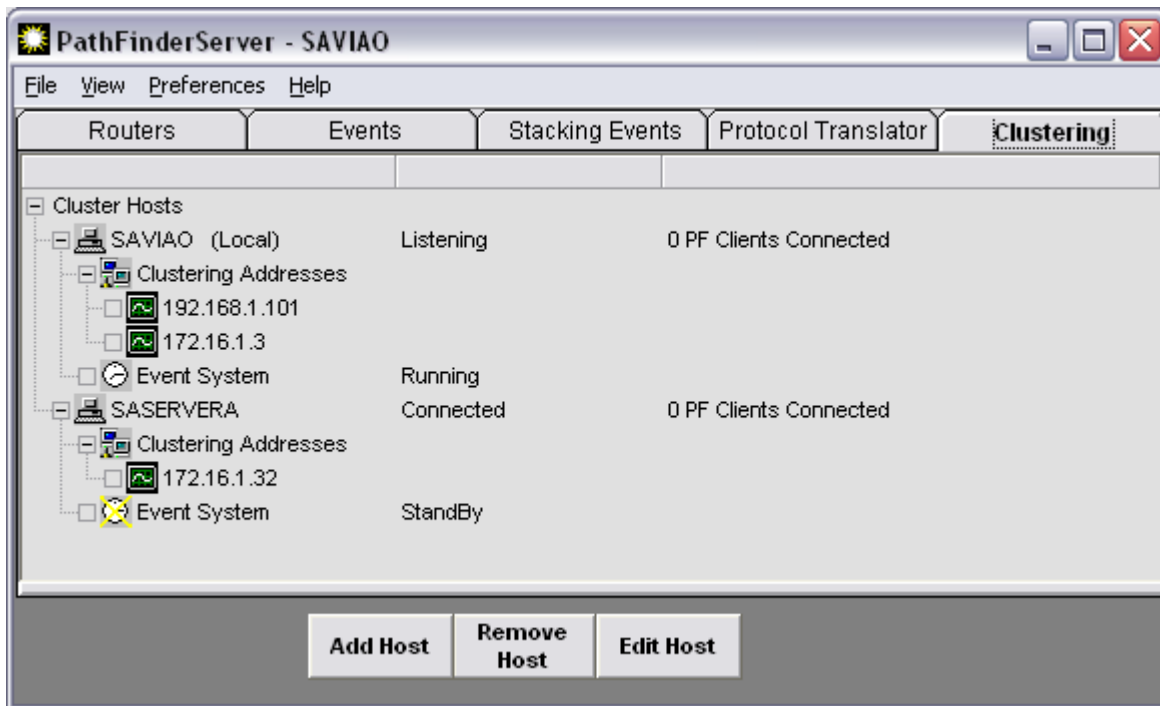
Clicking Add Host from the main PathfinderPC Server screen in the clustering tab will allow you to Add another PathfinderPC Server to the cluster. Enter the host name of the other server and the IP addresses of the other server to be used for clustering communications and client access.



It is important that the hostnames entered match correctly between the two servers. In addition, it is recommended to add the Axia network as a final last path for clustering communications if all else fails. Select OK when you are done entering the server information for the other server. You will then be prompted with a message asking if the local machine has the accurate configuration files. This question determines whether the configuration will get pulled from the remote machine, or whether the remote machine when it is set up will pull the initial router configuration from this one. You will now see icons for the new server in this cluster tab.



You will notice the red X through the Server B. This means that Server A has not been able to contact Server B yet. You will also notice the (local) word on the top node indicating which computer is the local machine. Finally, you may notice that the Event system is disabled (red x) on both nodes. After a period of time of waiting to establish a connection with the B Node, the Server A will go ahead and start the Event System. This is indicated by the removal of the red x through the Event System and the status will switch from Off to On. Now we must go through the same procedure on the other server. Once both servers have their own and the opposing server's information in their cluster file, they will be able to connect to one another. When this happens you will see the statuses on the machines oscillate through a number of states including converging and synchronizing data and they will eventually all settle on Connected and Listening for the local host. Furthermore, the Event System will be shown as running on the server with the lowest IP address, and will be shown as pending (with a yellow X through the event system) on the other machine.



Now if we shut down the PathfinderPC Server application on the ServerA, Server B will show the loss of connection and will try to re-establish the connection. If it can't it, it will start up its own event system. Then when the ServerA is restored, the event system will go back into pending on ServerB and ServerA will take over again once the converging and synchronization of data has taken place. Furthermore PathfinderPC clients automatically get the list of servers in the cluster when they connect. Thus if the server that the client is connected to fails, the client will automatically roll to the other server. The current server a client is using is shown in the bottom left corner of the PathfinderPC screen, and will flash yellow when it is attempting to reconnect or connect to a different server.

Here is a review if the basic steps for creating a PathfinderPC Server cluster.

1. Obtain 2 PCs preferably running Microsoft Standard Server
2. Install 4 NIC cards in each server
 - a. Assign 1 NIC card in each server an IP address on the Axia Network
 - b. Assign 1 NIC card in each server an IP address on the business Network if desired
 - c. Assign 1 NIC card in each server a private IP address for crossover clustering communications – for example
 - i. 172.16.10.1 - Netmask 255.255.255.0
 - ii. 172.16.10.2 - Netmask 255.255.255.0
 - d. Assign final NIC card in each server a private IP address in a different network block for redundant crossover clustering communications – for example
 - i. 172.16.11.1 – Netmask 255.255.255.0
 - ii. 172.16.11.2 – Netmask 255.255.255.0
3. Connect the business NIC card to the Business network and the Axia NIC card to the Axia network in each computer
4. Use an Ethernet crossover cable to cross-connect the remaining NIC cards between the two PathfinderPC servers

5. Use ping tests to make sure the Servers can access the Business network, Axia network, and each other via the crossover IP addresses and the Axia IP address.
6. Install PathfinderPC Server on Server A
7. Completely set up PathfinderPC Server and create the routers on Server A
8. When you are happy with the PathfinderPC Server setup on ServerA, click on the clustering tab and edit the settings for the ServerA.
 - a. Remove and re-add the clustering NIC IP addresses for this server in the preferred order
 - i. Cross-Over NIC 1
 - ii. Cross-Over NIC 2
 - iii. Axia Network NIC
 - b. Remove and re-add the client communication NICs as desired
 - i. Business NIC address
 - ii. Axia NIC address
9. Click OK
10. On Server A - Click Add Host
 - a. Enter the host name for ServerB
 - b. Enter the IP addresses for the Cross-over NICs of and Axia NIC on Server B into the clustering communications section
 - i. Server B Cross-Over NIC 1
 - ii. Server B Cross-Over NIC 2
 - iii. Server B Axia Network NIC
 - c. Enter the IP addresses for the Business and Axia Networks for Server B into the Client Communications Section
 - i. Server B Business NIC address
 - ii. Server B Axia NIC address
11. Click OK – and Select Yes to say that this server has the definitive configuration
12. Install PathfinderPC Server on Server B
13. Next you have two options.
 - a. Option A – use a thumb drive to copy the cluster.xml file from ServerA to Server B
 - i. Launch the Server and cancel the router config, then select the cluster tab and watch the servers synchronize
 - b. Option B – launch the PathfinderPC Server application and cancel the router config, and manually enter the clustering information into Server B so that it matches what we did in Server A.
 - i. When done select No to indicate that the router configuration files should be pulled from ServerA.
 - ii. Watch the synchronization take place

Notes of importance

It is always recommended to include the Axia NIC card as the NIC of last resort for clustering communications. This is because PathfinderPC Server determines which of the hosts listed in the Cluster.xml file is the local instance of PathfinderPC Server by comparing that IP list with the IP addresses on the local computer. If both cross-over NICs are down as they may be if the other host computer is turned off, no cluster NIC will come back as local, and the event system will not start.

The hostnames must match in both cluster configuration files in order for clustering to synchronize properly.

The IP addresses in the cluster file configuration must match those on the local machine or the event system will not start, even if the server is a stand-alone server. The cluster.xml file can be recreated automatically by deleting it and then restarting the PathfinderPC Server application.

PathfinderPC Server will automatically synchronize all router databases including the router settings, Events database, Stack event database, and protocol translator database between the 2 servers in the cluster. Stack Events can only be edited on the server that owns the event system, and the list of events may not display on the other server until it takes over the event system. This is normal. The synchronization happens automatically for any changes in any of these databases. The one exception is that each server maintains its own connection to all Axia equipment and therefore maintains its own route status table, but since both are getting this information from the same sources, they will always be in sync. In addition, there are a couple of things that are not automatically synchronized and have to be manually copied and/or setup between the servers at the time of this writing. These are log settings, Panels in the PFSPanels folder, and any custom scripts in the PFSScripts folder.

If your route database is very large, it is sometimes better to copy the database files from the primary server to the secondary server manually before you first launch the secondary server. This will reduce the time and effort needed for the very first initial synchronization.


Finally, Clustering requires a license for each machine, and the machines will not connect and synchronize properly if both machines have the same IP address.

PathFinderPC

PathfinderPC Client is the application that users will use to make routing changes with the PathfinderPC system. It communicates with the PathfinderPC Server over TCP/IP on the local LAN and/or over the Internet. It can be used to make routes, create and edit virtual routers, create, edit, and activate scene changes, as well as schedule events. It also has a search engine for finding route points and scenes throughout the entire routing system.

PathfinderPC Installation

Once the server software is installed, the client software may be installed on any Windows machine on the network that is to be used for making routing changes.

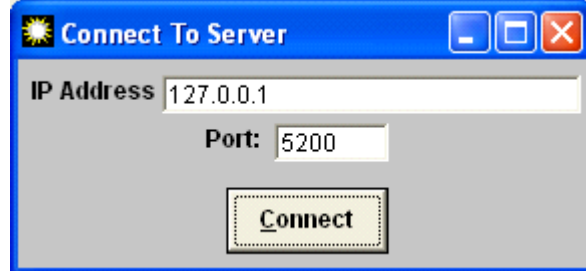
<p>Run Setup for PathfinderPC Client to start the installation.</p> <p>Click Next to begin the installation. On the next screen</p> <p>Click I Agree to the license.</p> <p>Click Next again.</p> <p>Select the location for installing the application.</p> <p>Click Next a final time to finish the installation procedure.</p>	
--	---

PathfinderPC Main Application

After installing PathfinderPC PC, be sure the server application is set up and running on the server PC, and then start PathfinderPC PC. 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.

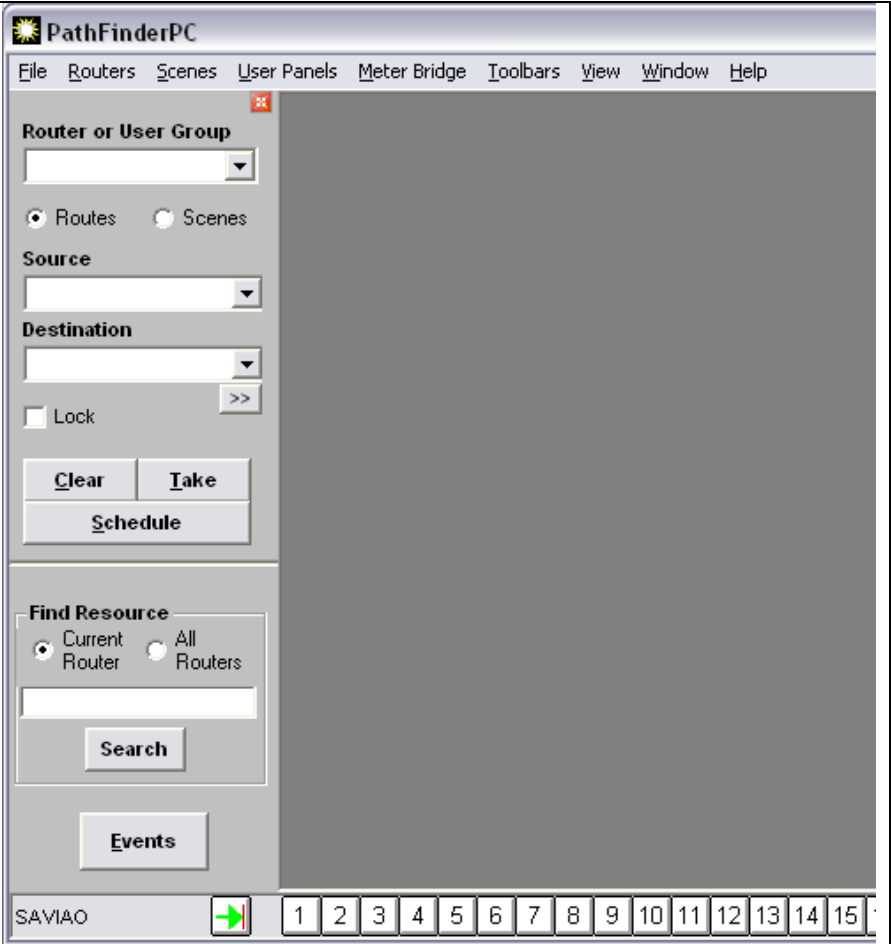
Enter the IP address or fully qualified DNS name of the computer running the server application. The port should match the Master TCP/IP Socket number in the server application. The default is 5200. If this has been changed on the server, you will need to change it here to match.

Click Connect

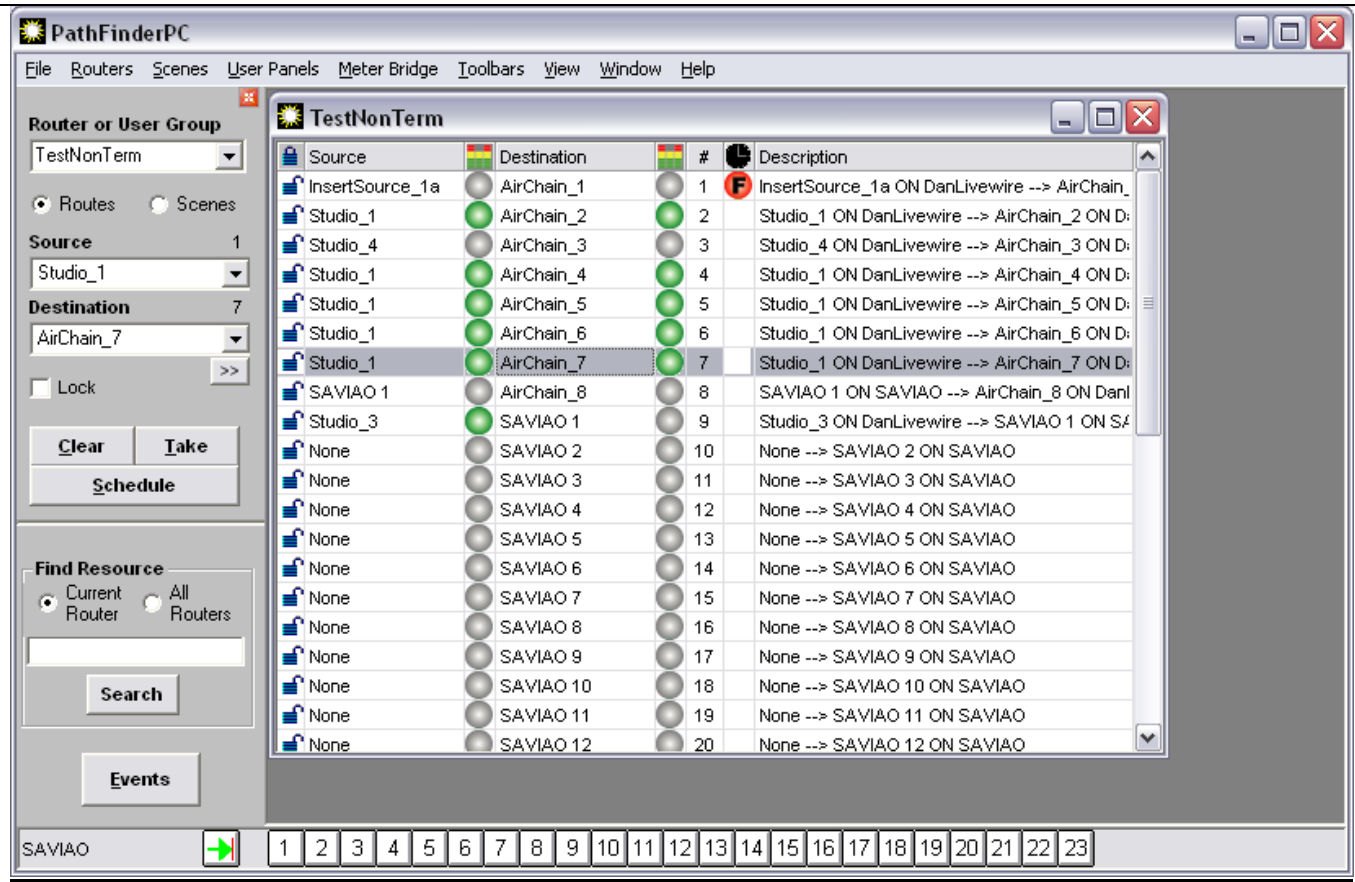


If the connection is successful all of the menu items on the main Patch Bay Control screen will become available.

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



Patch Bay Control screen with router list open.

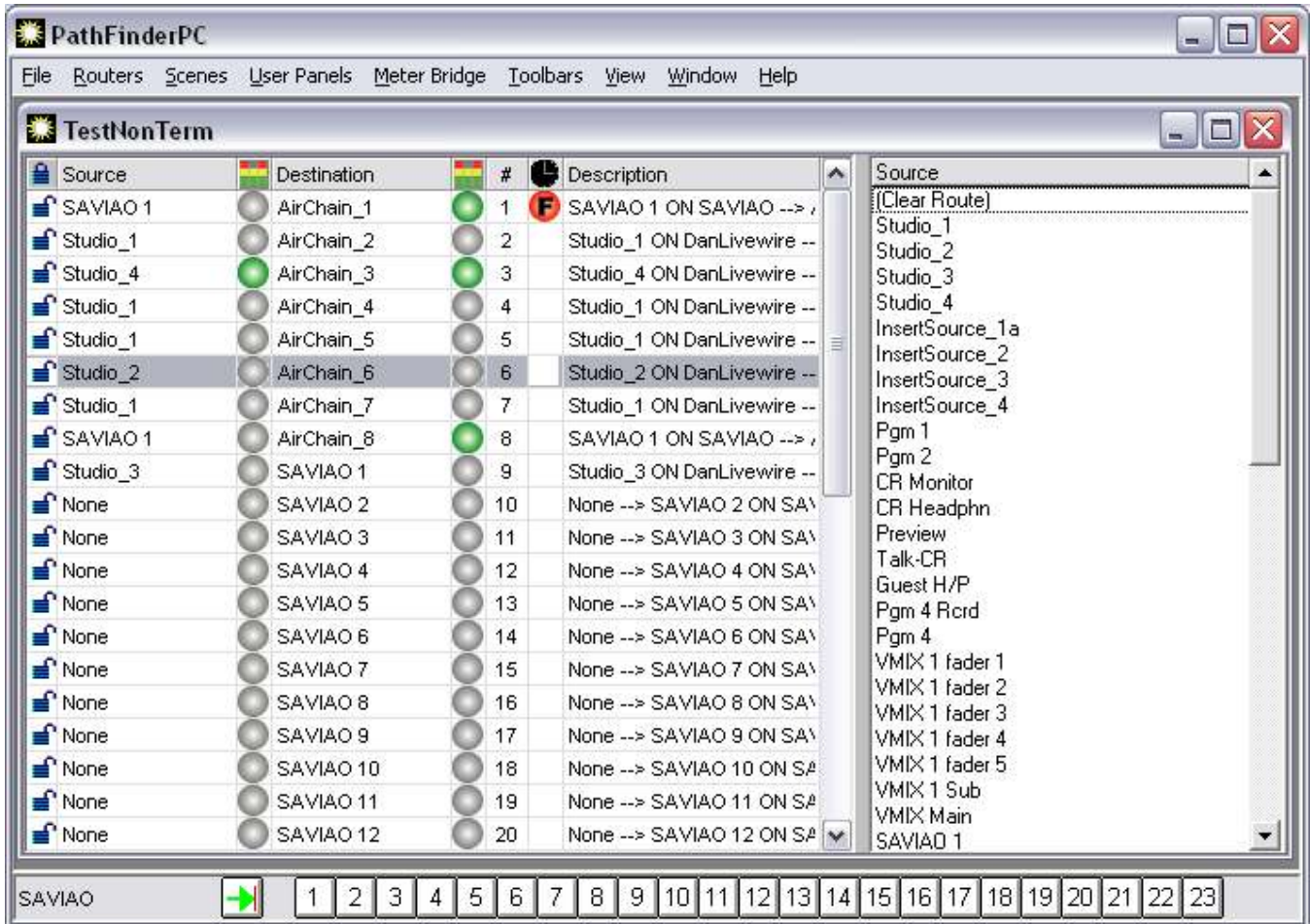


PathfinderPC has two control panel toolbars. Along the bottom is the window preset toolbar. Each numerical button holds a window configuration. Since the user can have multiple router and meter windows open simultaneously, this panel can be used to store these window positions and sizes for quick retrieval. 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.

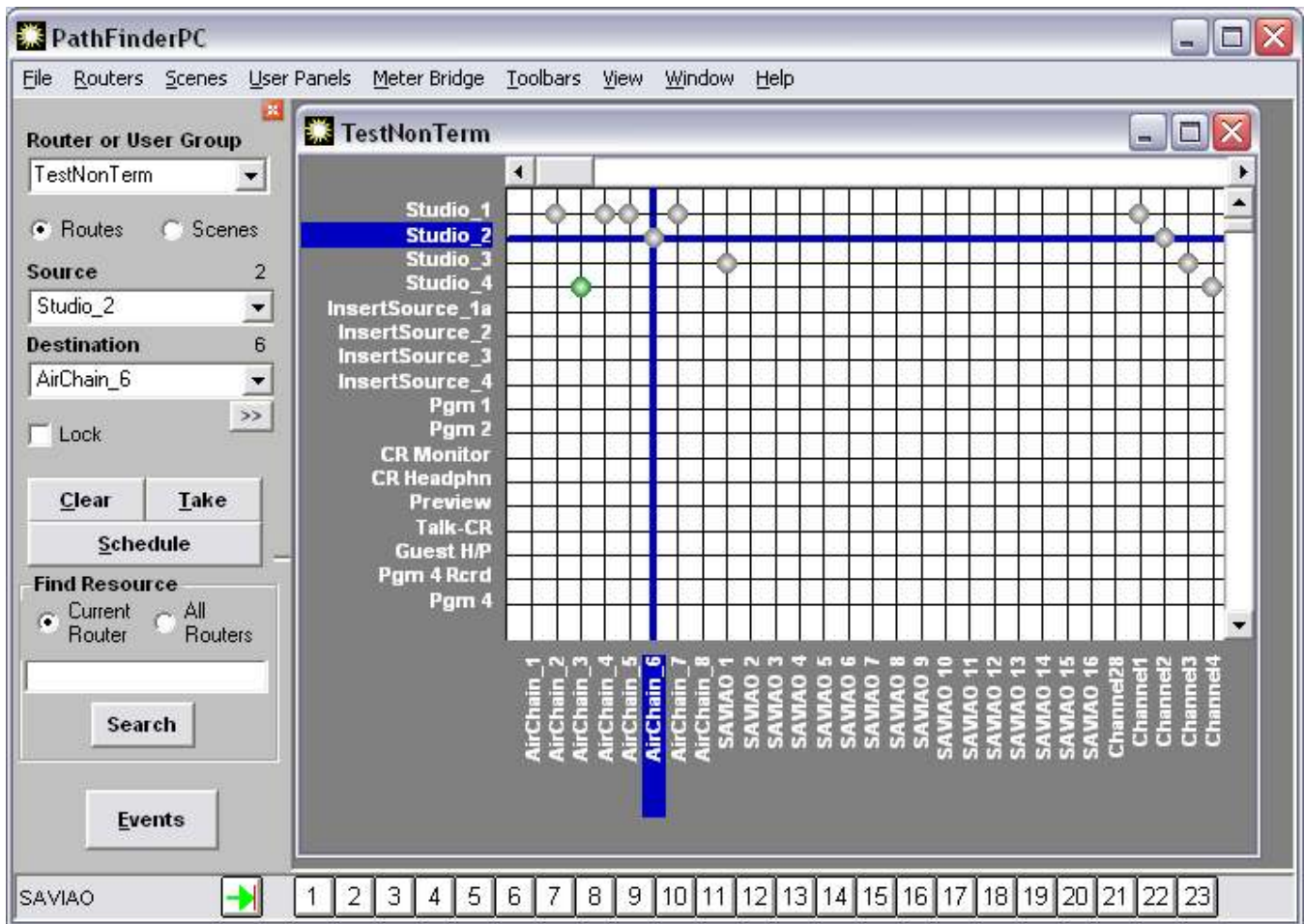
The route control toolbar is on the left side of the PathfinderPC screen. This provides the primary routing control over the system. Routes may be made by selecting router and the destination, and then selecting the source you wish to be assigned to the destination. Finally click Take or clear to establish 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. You can also click on the schedule button to enter the scheduling menus for scheduling a route to happen at a particular time. Route points may also be locked or unlocked using this panel. 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 Pathfinder. These include all Engine destinations. While PathfinderPC will display the route status of Engine Destinations, it is not allowed to make changes to Engine routing at this point in time. That must be accomplished through the standard Element and Engine interfaces.

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 or scheduled.

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. One exception is that at this point in time Axia windows driver sources and destinations do not support metering, but will still be displayed as grey dots.

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. 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 Server and 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.

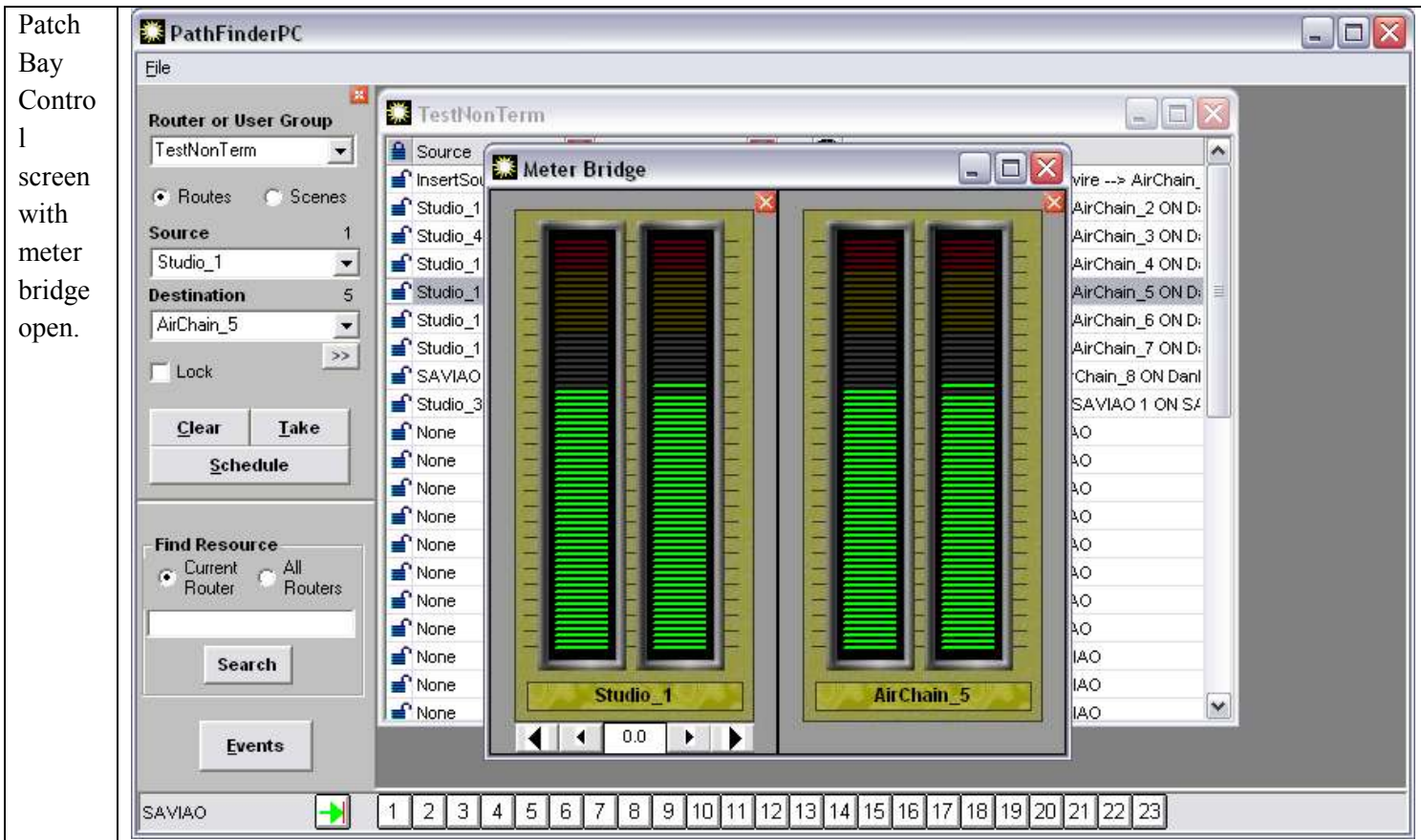
Source Name	#	Description	Host IP	Host Name	Host Port	Axia Channel	Axia Stream
Studio_1	1	Studio_1 ON DanLivewire	172.16.1.201	DanLivewire	1	21	239.192.0.21
Studio_2	2	Studio_2 ON DanLivewire	172.16.1.201	DanLivewire	2	22	239.192.0.22
Studio_3	3	Studio_3 ON DanLivewire	172.16.1.201	DanLivewire	3	23	239.192.0.23
Studio_4	4	Studio_4 ON DanLivewire	172.16.1.201	DanLivewire	4	24	239.192.0.24
InsertSource_1a	5	InsertSource_1a ON DanLivewire	172.16.1.201	DanLivewire	5	25	239.192.0.25
InsertSource_2	6	InsertSource_2 ON DanLivewire	172.16.1.201	DanLivewire	6	26	239.192.0.26
InsertSource_3	7	InsertSource_3 ON DanLivewire	172.16.1.201	DanLivewire	7	27	239.192.0.27
InsertSource_4	8	InsertSource_4 ON DanLivewire	172.16.1.201	DanLivewire	8	28	239.192.0.28
Pgm 1	9	Pgm 1 ON Livewire	172.16.1.60	Livewire	1	101	239.192.0.101
Pgm 2	10	Pgm 2 ON Livewire	172.16.1.60	Livewire	2	102	239.192.0.102
CR Monitor	11	CR Monitor ON Livewire	172.16.1.60	Livewire	8	110	239.192.0.110
CR Headphn	12	CR Headphn ON Livewire	172.16.1.60	Livewire	9	111	239.192.0.111
Preview	13	Preview ON Livewire	172.16.1.60	Livewire	10	112	239.192.0.112
Talk-CR	14	Talk-CR ON Livewire	172.16.1.60	Livewire	11	113	239.192.0.113
Guest H/P	15	Guest H/P ON Livewire	172.16.1.60	Livewire	12	114	239.192.0.114
Pgm 4 Rcrd	16	Pgm 4 Rcrd ON Livewire	172.16.1.60	Livewire	3	103	239.192.0.103
Pgm 4	17	Pgm 4 ON Livewire	172.16.1.60	Livewire	17	104	239.192.0.104
VMIX 1 fader 1	18	VMIX 1 fader 1 ON Livewire	172.16.1.60	Livewire	20	211	239.192.0.211
VMIX 1 fader 2	19	VMIX 1 fader 2 ON Livewire	172.16.1.60	Livewire	21	212	239.192.0.212
VMIX 1 fader 3	20	VMIX 1 fader 3 ON Livewire	172.16.1.60	Livewire	22	213	239.192.0.213
VMIX 1 fader 4	21	VMIX 1 fader 4 ON Livewire	172.16.1.60	Livewire	23	214	239.192.0.214

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 both the client and server 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.

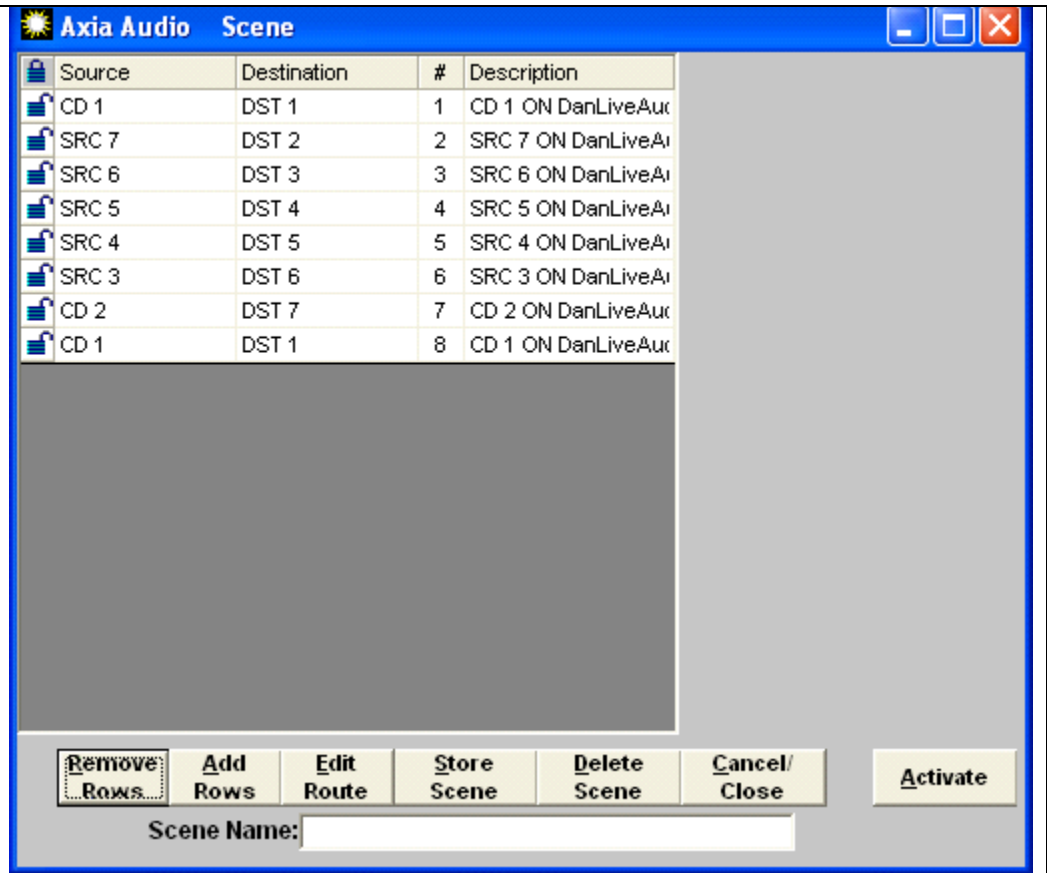


The meter bridge window may be enlarged and meters added, removed, and reorganized. Right clicking on the grid will add additional meters to the meter bridge. Source meters also have buttons to change the input gain of the source channel.

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 the server for that particular router. A new scene may be captured and edited using the Edit and Capture Scene menu items.

Clicking capture scene will produce the following window.



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 PathfinderPC Server's 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 or using the drop down scene list of the route control panel.

Please note: The scenes also store the state of the Lock point. 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.

Virtual routers may be created and edited either from the server software or the client software. To create a virtual router in the server software, simply add a new router and make the Router Model Type Virtual. Then when the Edit Routes button is selected, the Edit Virtual Router Window appears. In the PathFinderPC 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 be edited from either the Client or Server software, but it may only be completely deleted from the server software.

Virtual Router
Editing Screen.

#	Sources - (Machine Outs)	Description
1	CD 1	CD 1 ON DanLiveAu
2	CD 2	CD 2 ON DanLiveAu
3	SRC 3	SRC 3 ON DanLiveA
4	SRC 4	SRC 4 ON DanLiveA
5	SRC 5	SRC 5 ON DanLiveA
6	SRC 6	SRC 6 ON DanLiveA
7	SRC 7	SRC 7 ON DanLiveA
8	CD 1	CD 1 ON DanLiveAu

#	Destinations - (Machine Ins)	Description
1	DST 1	DST 1 ON DanLiveA
2	DST 2	DST 2 ON DanLiveA
3	DST 3	DST 3 ON DanLiveA
4	DST 4	DST 4 ON DanLiveA
5	DST 5	DST 5 ON DanLiveA
6	DST 6	DST 6 ON DanLiveA
7	DST 7	DST 7 ON DanLiveA
8	DST 1	DST 1 ON DanLiveA

Virtual Router Name:

Buttons: Edit Route, Remove Route, Add Route, Edit Route, Remove Route, Add Route, Import Routes, Create Close, Cancel

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 the server 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.

Import Routes Router Selection Screen

Router Name: TestNonTerm

OK

After selecting OK, a selection list will appear.

Use this list to select the routes you wish to import. Use the Cntrl 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.

Virtual Import Selection

Select All Clear All

Source Name	#	Description	Host IP	Host Name	Host Port	Axis
Studio_1	1	Studio_1 ON DanLiv	172.16.1.201	DanLivewire	1	21
Studio_2	2	Studio_2 ON DanLiv	172.16.1.201	DanLivewire	2	22
Studio_3	3	Studio_3 ON DanLiv	172.16.1.201	DanLivewire	3	23
Studio_4	4	Studio_4 ON DanLiv	172.16.1.201	DanLivewire	4	24
InsertSource_1a	5	InsertSource_1a ON	172.16.1.201	DanLivewire	5	25
InsertSource_2	6	InsertSource_2 ON	172.16.1.201	DanLivewire	6	26
InsertSource_3	7	InsertSource_3 ON	172.16.1.201	DanLivewire	7	27
InsertSource_4	8	InsertSource_4 ON	172.16.1.201	DanLivewire	8	28
Pgm 1	9	Pgm 1 ON Livewire	172.16.1.60	Livewire	1	101

Select All Clear All

Destination Name	#	Description	Host IP	Host Name	Host Port
AirChain_1	1	AirChain_1 ON DanL	172.16.1.201	DanLivewire	1
AirChain_2	2	AirChain_2 ON DanL	172.16.1.201	DanLivewire	2
AirChain_3	3	AirChain_3 ON DanL	172.16.1.201	DanLivewire	3
AirChain_4	4	AirChain_4 ON DanL	172.16.1.201	DanLivewire	4
AirChain_5	5	AirChain_5 ON DanL	172.16.1.201	DanLivewire	5
AirChain_6	6	AirChain_6 ON DanL	172.16.1.201	DanLivewire	6
AirChain_7	7	AirChain_7 ON DanL	172.16.1.201	DanLivewire	7
AirChain_8	8	AirChain_8 ON DanL	172.16.1.201	DanLivewire	8
SAVIAD 1	9	SAVIAD 1 ON SAVIAD	172.16.1.3	SAVIAD	1
SAVIAD 2	10	SAVIAD 2 ON SAVIAD	172.16.1.3	SAVIAD	2

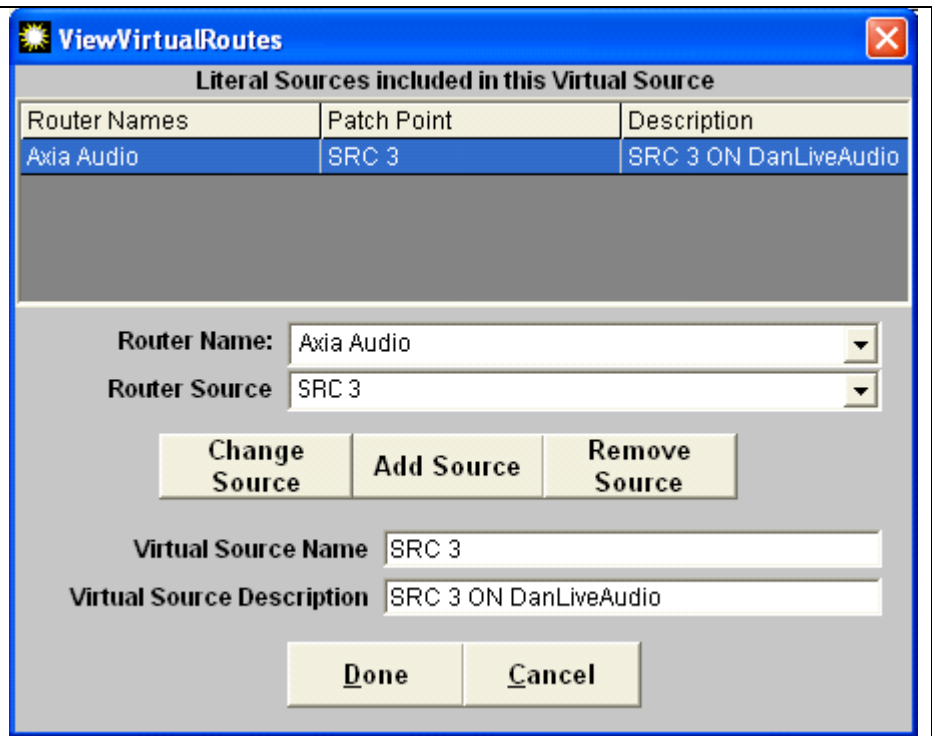
Append
 Replace Existing

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.

This is the window 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 concept of using a Virtual Router 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 is fairly simple. 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.

VirtualRecordingStudio, Inc. has three routers. They have an audio router, a video router, and a machine control router. Using PathFinderPC and PathFinderPC Server they create connections and set up each of the three routers on the server. At this point the users can use the Routers menu in PathFinderPC to switch between each of the routers and quickly make audio, video, or machine control routes. However, the grumbling engineers, after tasting the speed of not having to walk to a central place to make their routes, complain it's still not enough. Because that's just what grumbling engineers do. They want to be able to make one click of a mouse and route audio, video, and machine control from their DAWs (digital audio workstations) to any video deck in the machine room all at once. This is where the Virtual Router excels.

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 "Digital Beta Video Deck Input." Then select the Audio router from the router name drop down combo. Select the correct audio point for the DBeta's input from the router Point drop down combo. Click Add Destination. Next Select the Video Router from the router name drop down combo. Select the correct video input point for the DBeta from the router point list that is now displaying the list of routes for the Video router since that is what is selected above. Click Add Destination. Finally repeat the procedure to add the Machine Control Point. Last click the "Done" button to add the point into the configuration.

Next repeat the same procedure under the Sources List (Device outputs) and make a point for the DAW. You could also create an output point for the DBeta and an input point for the DAW, so that routing could take place in both directions. Finally, once all desired points are created, click the Create/Close button to submit the new router design to the server. 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 DAW1 Out to DBeta IN, Audio, video, and Machine Control are routed simultaneously.

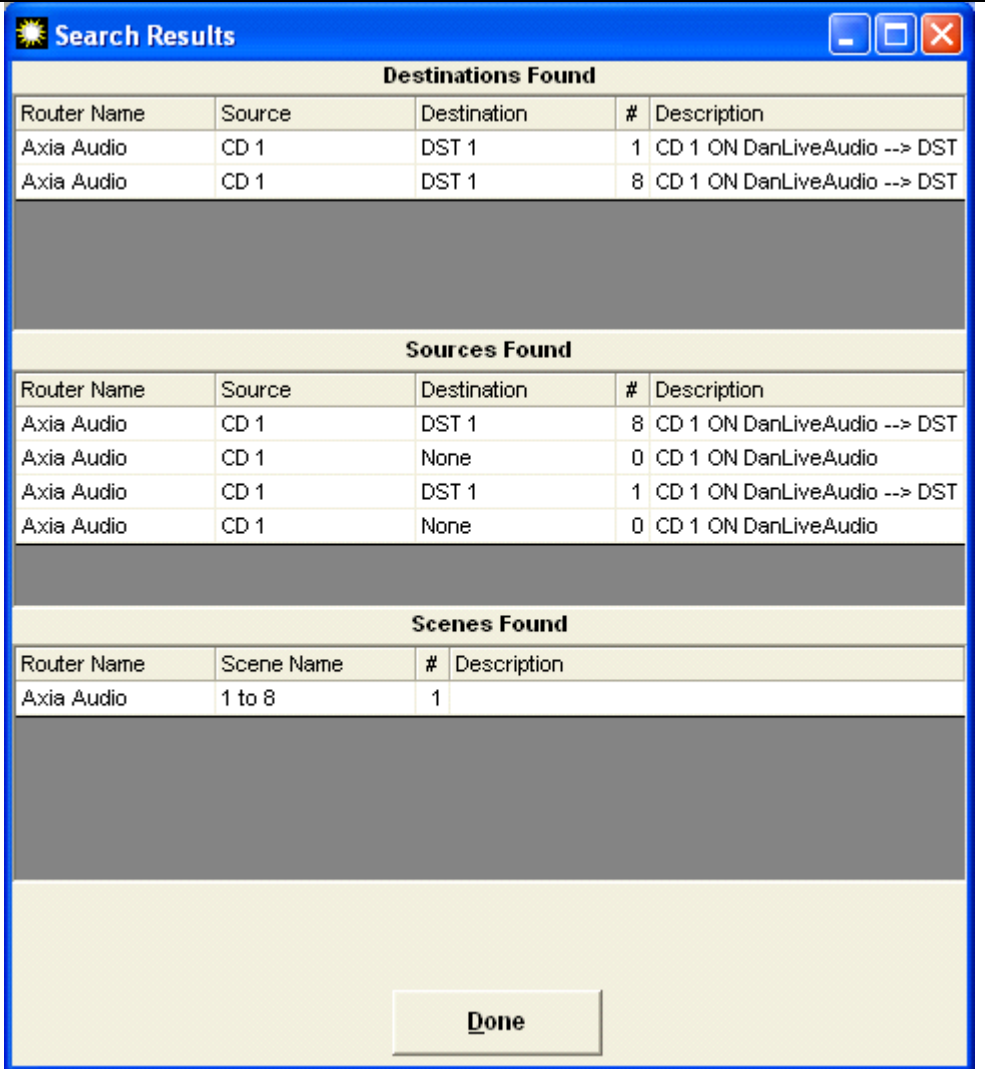
Finally a scene could be created which makes two routes on the virtual router to route both the output of the DAW1 to the Input of the DBeta and the output of the DBeta to the input of the DAW1 with one mouse click, and all of the audio, video, and machine control would follow.

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

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 box and select whether the search scope should be the current router or all routers on the server. The software will then open the search results window.

Search Result Window



The screenshot shows a window titled "Search Results" with a blue header and standard window controls. It contains three tables of search results. The first table, "Destinations Found", lists two entries for "Axia Audio" with source "CD 1" and destination "DST 1". The second table, "Sources Found", lists four entries for "Axia Audio" with source "CD 1" and various destinations. The third table, "Scenes Found", lists one entry for "Axia Audio" with scene name "1 to 8". A "Done" button is located at the bottom of the window.

Destinations Found				
Router Name	Source	Destination	#	Description
Axia Audio	CD 1	DST 1	1	CD 1 ON DanLiveAudio --> DST
Axia Audio	CD 1	DST 1	8	CD 1 ON DanLiveAudio --> DST

Sources Found				
Router Name	Source	Destination	#	Description
Axia Audio	CD 1	DST 1	8	CD 1 ON DanLiveAudio --> DST
Axia Audio	CD 1	None	0	CD 1 ON DanLiveAudio
Axia Audio	CD 1	DST 1	1	CD 1 ON DanLiveAudio --> DST
Axia Audio	CD 1	None	0	CD 1 ON DanLiveAudio

Scenes Found			
Router Name	Scene Name	#	Description
Axia Audio	1 to 8	1	

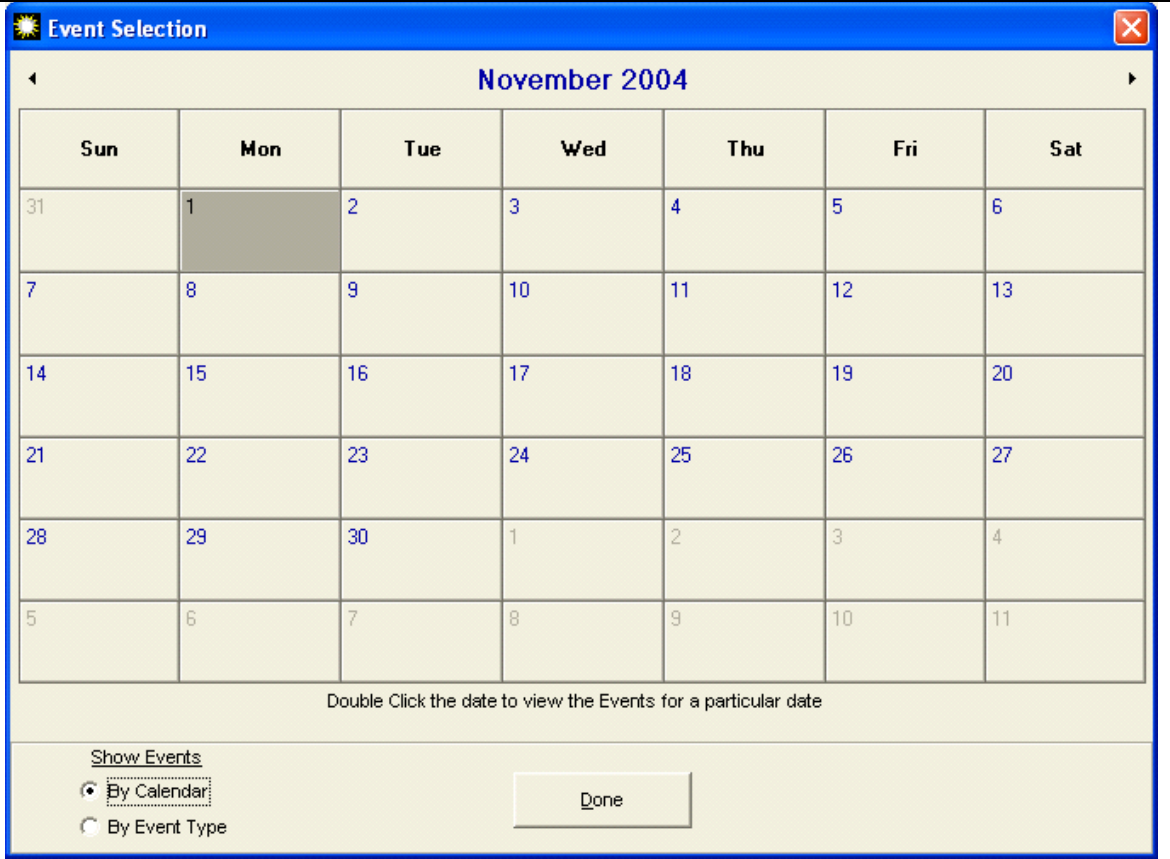
Done

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 if that particular router is open.

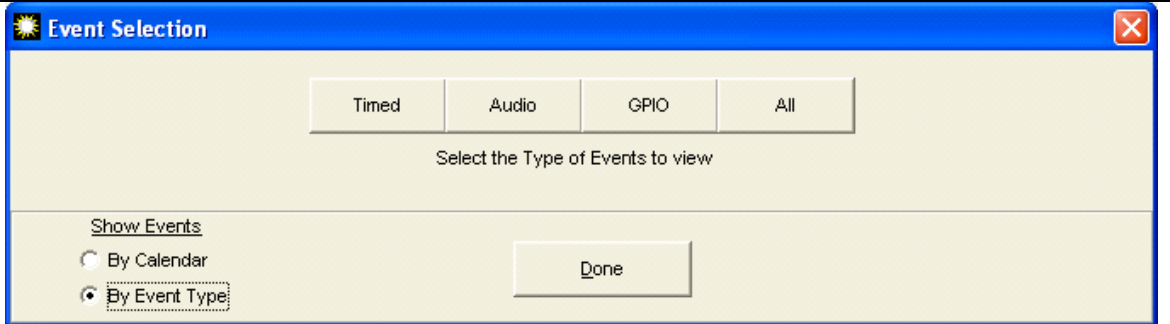
Scheduling

The event scheduling window can show event by date or by type.

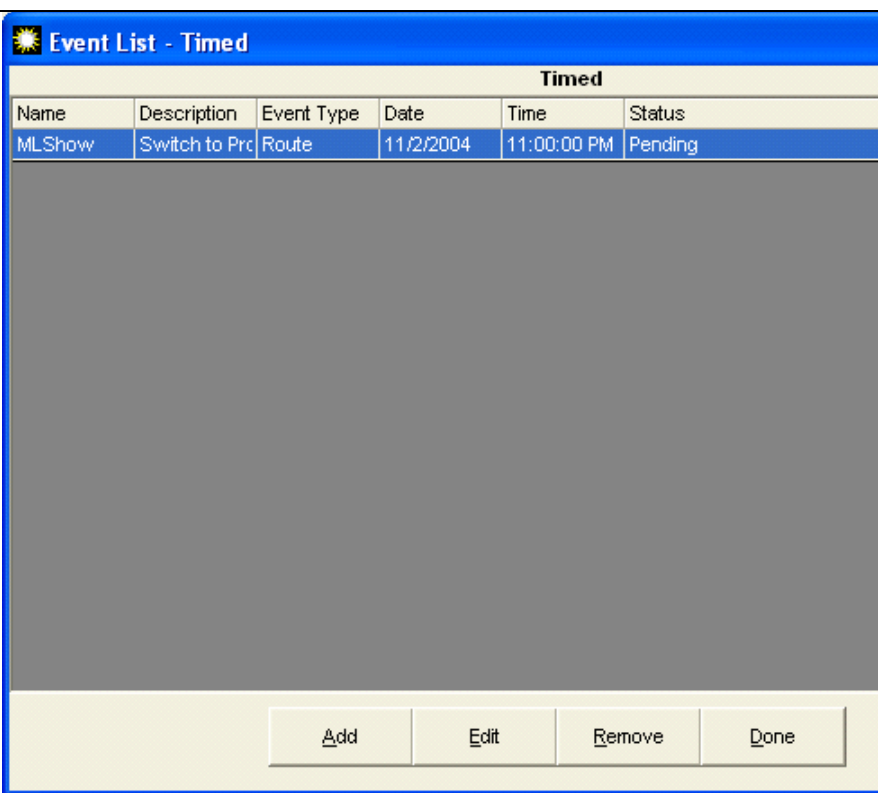
Double click a particular date to show events which pertain to a particular date



The type event window will list the events according to what type of event they are.



Events can also be added, edited, and removed from the event list window. The event wizard screens are the same as for the server application. See the event section under PathfinderPC Server in this document for more details.



User Panel Menu

The User panel menu will only be present if user panels have been created using the Panels Designer application and saved in the PFSPanels folder of the server. See the section in this document on 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.

PathfinderPC Client Registry Settings

We have found that many clients do not want their users to have to login when running PathfinderPC 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

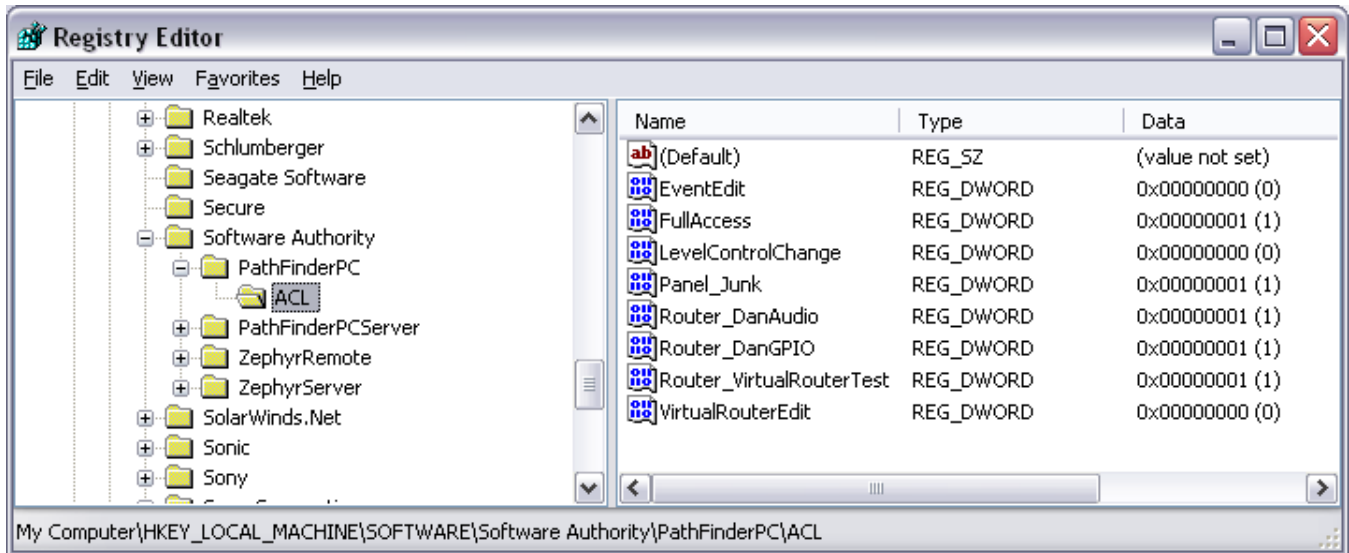
VirtualRouterEdit (0 or 1) for deny or allow

SceneChangeEdit (0 or 1) for deny or allow

LevelControlChange (0 or 1) for deny or allow

EventEdit (0 or 1) for deny or allow

Settings 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.

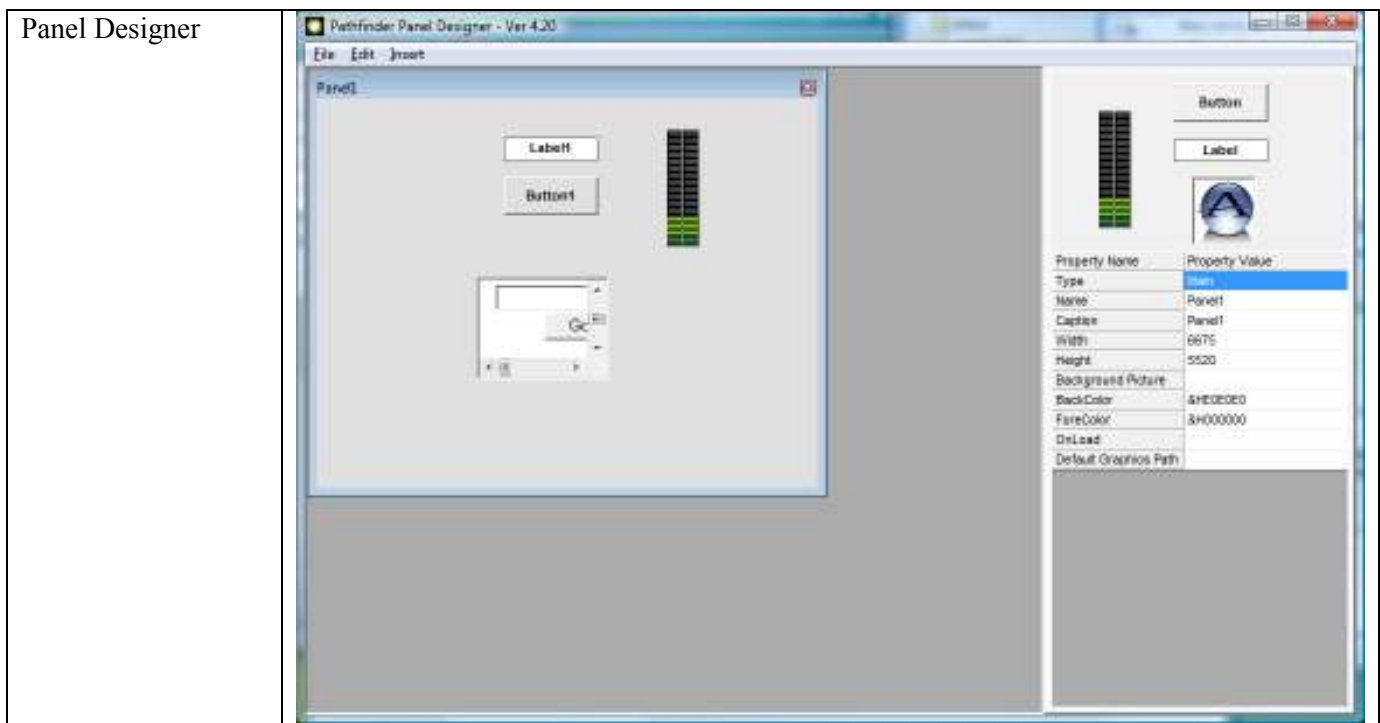


Panel Designer

The PathfinderPC Panel Designer application is a tool for creating custom user panels to be used either in the PathfinderPC or PathfinderPC Mini applications. With it you can create a user panel with background colors and pictures, and drag and drop buttons and labels 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 designer only allows you to design and define the panel. When you are finished designing the panel it should be saved to the PFSPanels Folder within the PathfinderPC Server folder. Then any panel in this folder will become available for use by PathfinderPC Client or PathfinderPC Mini.

Creating a Panel

To begin creating a user panel, launch the panel Designer application. Then from the File Menu select Create Panel.



You can then resize the main panel form to be the correct size. At this point in time there are either three or five objects available in the Panel Designer depending on your license. One is the main Panel Form itself, one is a button, and one is a Label Control. These are available to all PathfinderPC users. In addition, Pathfinder Pro users will have a meter control and a web browser control. If you are PathfinderPC user (not Pro), and you install the panel designer to a machine other than the server, these additional controls may appear as well. But they will only be usable in running panels if the panel is connected to a PathfinderPro server. Each Panel gets stored to a unique file and becomes a single user panel available to the PathfinderPC Client and Server. Buttons and Labels may be added to the panel by dragging the control from the top right hand control pad 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 controls 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. So 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 PathfinderPC System when the panel is opened and run using either PathfinderPC Client or PathfinderPC 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 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 stack events, the button will be identified in the stack event 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.

So at this point we have a panel that has a named button with a nice caption, but the button does not 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 Client and/or PathfinderPC Mini. To determine what numbers to use in these properties, look at the main PathfinderPC Server window and select the Routers Tab. Find the number of the router that has the source and destination to be used. Enter this number into the `MapRouter` property. Then select that router in the list and click on the Route Names button. This will list all of the sources and destinations in the router. 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 into the correct folder on the server, and then the panel should be opened using either PathfinderPC Client or PathfinderPC Mini. So 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. Now save the Panel, giving it a name, and making sure it is being saved to the `PFSPanels` folder within the PathfinderPC Server folder.

Now launch the PathfinderPC Client application. 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 Client called User Panels. Under that menu item you will find your newly created user panel. If you create additional panels and save them to the `PFSPanels` folder on the server, 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 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 Mini for details. Then launching PathfinderPC 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 resave the changes you must close and open the panel again within PathfinderPC Client to get the changes to display. In 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 Stack Event qualifiers to determine the button functions. The section on latching buttons in the Stack Event part of this document is a good example. In addition the detailed property descriptions below will provide more insight on the huge variety of functions these controls can perform.

To continue 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 GPIOButton 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. From the PathfinderPC Server PC open a command prompt and type:

```
telnet ipaddress 4010
```

example:

```
Telnet 172.16.1.3 4010
```

For the IP address use the ipaddress 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. So enter the module number and the button number that should be mapped to the software button. Then save the panel.

When you save the panel it will also create a HardwareMap file in the PFSPanels folder. This file at the time of this writing is only read when the server starts, so to get your hardware maps to work restart the PathfinderPC

Server application. Now 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 the server 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 stack events to define the button behavior. In that case create a stack event group and stack event (or series of stack events). Fill in the UserPanelButton qualifier entering the PanelName.ButtonName for the button name and the properties of the button that will trigger the event such as MouseDown (equivalent to pressing the button). Then in the Action section we could create a route action, and a user panel action that would change the route and change the color and caption of the button on the fly. Or we could do both. We could define mapped routes, but also use stack events on the same buttons to change its captions or flashing state. In this manner the user panels can be designed to accomplish an infinite array of custom tasks in a button package that is simple for the end user to use and understand. See the Stack Event section of this document for more details on using Panel Controls with Stack Events.

You will also notice that some of the properties are picture properties. PathfinderPC Server will not serve out pictures, but the client applications will look in specific places for the graphic files. First in the Panel Designer, if you click on the main panel 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 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.

Meter controls are also available to PathfinderPro users to display dynamic metering for a source or destination. Finally, you can use the Web Browser control with a PathfinderPro license 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



Please 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.

<u>Panel Properties</u>	
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 PanelGraphics folder in the PathfinderPC 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 at this point in time.
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.

<u>Label Control Properties</u>	
Type	Defines the object type as label and cannot be changed by the designer.
Name	Defines the name of the label 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 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.
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	<p>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:</p> <p><i>Open a command prompt on the windows PC and type Telnet IPAddress 4010 - "Telnet 172.168.2.3 4010"</i></p> <p><i>Press Enter. This should open a Telnet session with the CPU.</i></p> <p><i>Now tap several of the user panel buttons. You should see messages that look like:</i></p> <pre><i>EVENT MOD_USER#3.BUT#6 KEY=DOWN</i> <i>EVENT MOD_USER#3.BUT#6 KEY=UP</i></pre> <p><i>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.</i></p>
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.

<u>Button Control Properties</u>	
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 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 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.
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 Client's or PathfinderPC 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 On	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 Client's or PathfinderPC 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.
--	---

<u>Meter Control Properties</u>	
(Meters are only available on panels that are connected to a PathfinderPro server license)	
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 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 Meter. 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.
Source/Dest	Defines whether the IO this meter will be monitoring 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

<u>WebBrowser Control Properties</u>	
(Meters are only available on panels that are connected to a PathfinderPro server license)	
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	Defined the URL to display in the web browser

PathfinderPC Mini

PathfinderPC Mini is a small application that displays a specified user panel as its own application. It should be used in situations where the PathfinderPC Administrator only wants the user to have access to the routes and control offered by a specific user panel and nothing else. To use PathfinderPC Mini the administrator need to complete the details in a small configuration file stored in the PathfinderPC Mini 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 the PathfinderPC Server and a second clustered server if desired. Also enter the name of the Panel to be displayed. If you have changed or created specific user accounts within PathfinderPC Server, you may also need to enter a valid user name and password.

Save the file and launch PathfinderPC Mini. As long as the application can find the server and finds the Panel within the Server's PFSPanels folder, the panel will be displayed and run as its own application.

PathfinderPro VMIX Control

Overview

VMIXControl is a mixer application that is designed to provide the user with a software mixer that can be run on any PC in a network and controls an Axia Engine's virtual mixer. At this point in time, this application does not control any of the actual Element surface itself. Rather, each Axia mix engine also contains a virtual 40 channel mixer that is divided into 8 five channel submixers. This virtual mixer can be used to create background submixes, or to automate mixing processes using PathfinderPC Stack Events. It can also be controlled by the VMIXControl application.

VMIXControl is not designed to replace an Element. The Axia engine VMixer is a simple mixer only. It does not support the automatic backfeed generation, eq, processing, monitoring, and many of the other advanced features required for a true control room environment. However, for environments where all that is required is simple mixing procedures, it becomes an incredibly useful tool, and one that we hope to build on in the future. Furthermore it can work from anyplace that has TCP access to both PathfinderPro and the Axia mix engine. So with proper network security, this application can easily be used for remote mixing over the Internet.

VMIX Control requires TCP access to both PathfinderPro Server and an Axia Mix Engine in order to function. This is slightly different from other Pathfinder client applications which only require a TCP connection to the Pathfinder server. VMIX control requires a TCP connection to both PathfinderPro Server and to the Axia Mix Engine. It will only function when connected to a PathfinderPro server.

Licensing

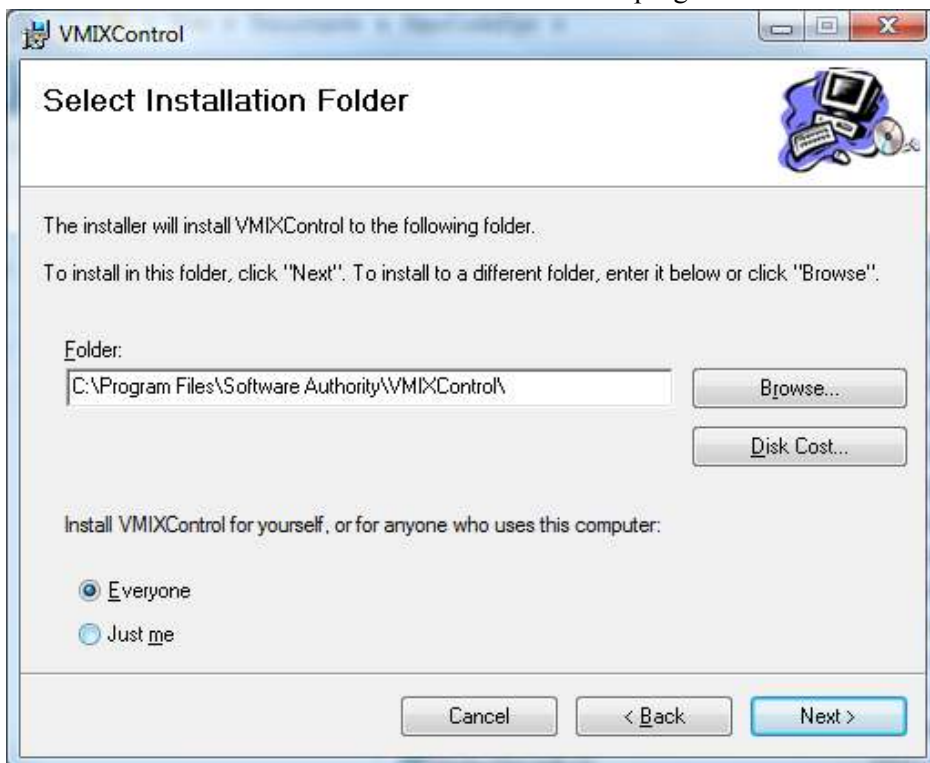
The VMIXControl application requires a connection to a licensed PathfinderPro server running version 4.20 or later to function. Like PathfinderPC and Mini, it can be installed on as many computers in the network as you like as long as it is used with a valid PathfinderPro server.

Installation

To install VMIXControl, simply double click the VMIXControl installer. Please note that this application requires that the computer already have the .Net framework version 2.0 installed. Most XP/Vista/2003 machines will already have this. If the framework is not present, it can be downloaded and installed through the Windows update system.



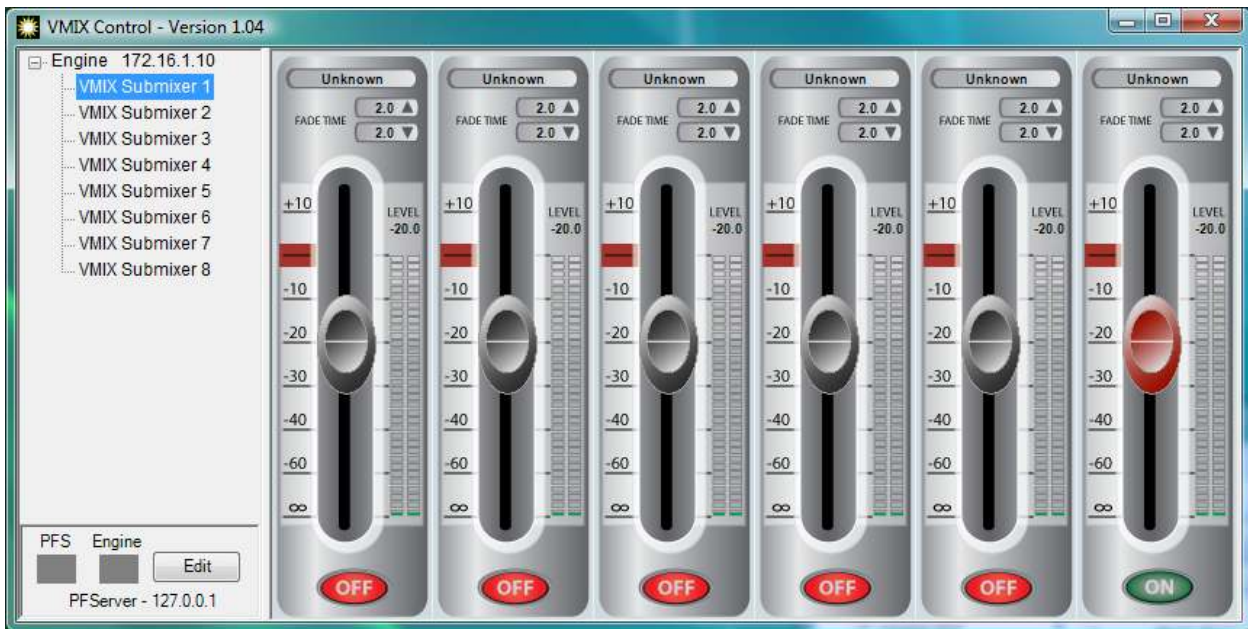
Click Next and select who will be allowed to use the program.



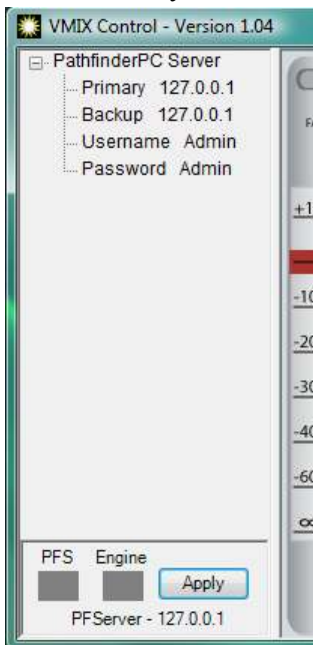
Continue to Click Next and Finish until the application is installed.

Configuring VMIXControl

The first time VMIX control is launched, it will look like this:



In order to use VMIXControl, you first must enter the PathfinderPro Server and the Axia Engine addresses that will be used by VMIXControl. Click the Edit button to begin.

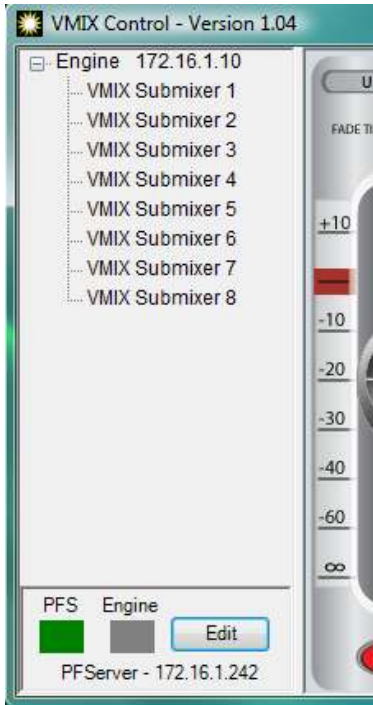


Primary and Secondary IP Address: Until you have an active connection to PathfinderPro, the Engine IP address is not adjustable, and in fact will not be visible in the edit window as shown above. Begin by clicking on and entering the server IP addresses for PathfinderPro. If you are running a single server you can enter the same IP address in both Primary and Secondary. If you have a PathfinderPro Cluster, you can enter both server IP addresses.

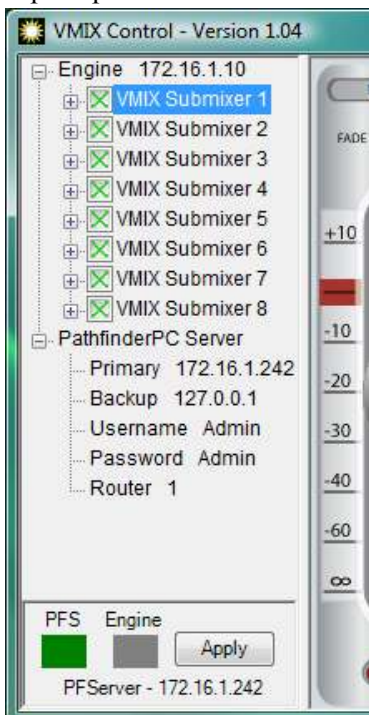
UserName and Password: The username and password fields are very important. By default they are set to the default Pathfinder Administrator account. The username and password you select needs to be a user in the PathfinderPC Server's user database. If you select an Admin user, the user will have the rights to change the Engine they are controlling and the VMIX faders that are visible using the Edit button. If you select a General User, clicking the edit button will ask for an Administrator username and password. If one is not provided, the

only thing the user will be able to change is which PathfinderPC Server the VMIX is connected to. In this way an Administrator can configure VMIXControl on the workstation using a PathfinderPC Server Administrative account, and then after clicking Apply, the user will be locked into using the configured Engine and faders. Therefore, the username and password that is entered in this part of the configuration should be the one that the user will generally be running under.

After configuring these settings, click Apply.



At this point, the green PFS should light up indicating a successful connection to PathfinderPro Server. Now click Edit again and enter an Administrative username and password from the PathfinderPro server user database if prompted. The edit window should now show additional options:

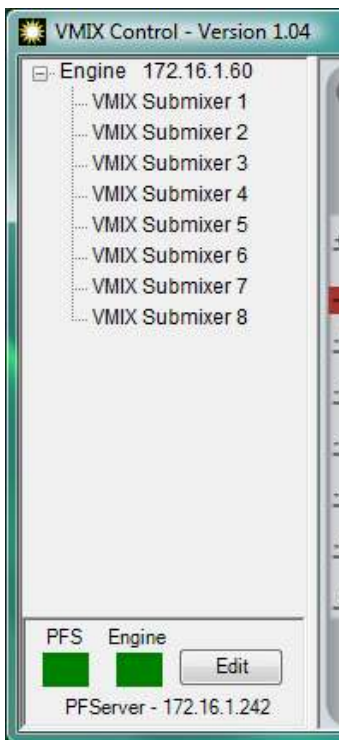


First click on the Engine field and enter the IP address of the Engine whose VMIX you wish to control. Then uncheck any of the 8 VMIXers on this engine you wish to hide from the user. You can also click on the plus sign and disable individual faders within a VMIXer.

Router: Use the Router field to assign a number of a Pathfinder Router that you wish to use for source selection. The router you select here must be a router number in PathfinderPro Server. If you select the number of the main Axia router, all sources on the network will be available to the user to assign to VMIX faders. If you wish to limit this list, create a virtual router in PathfinderPC Server with only the sources you want the user to see. Then assign the router ID number of the virtual router in this configuration screen.

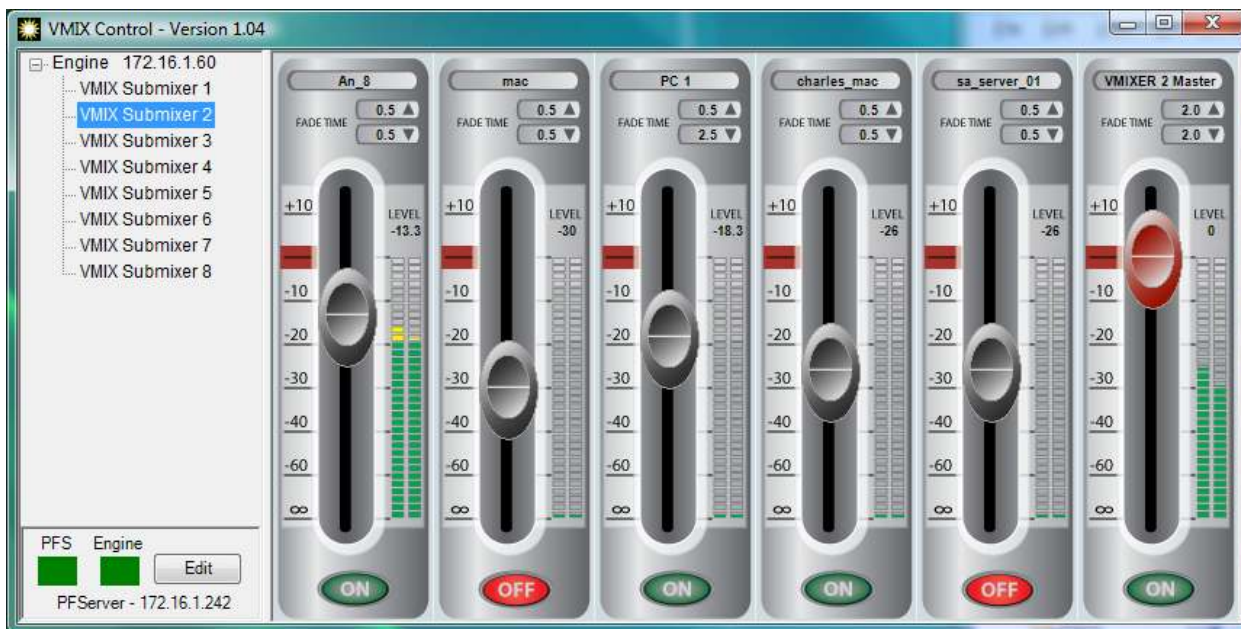
Once you have completed the configuration changes, click Apply again to store the changes.

At this point you should get a green light under the Engine label as well. Finally click on some of the VMIX links on the left hand side of the program, or restart the program to begin using the application.



Using VMIXControl

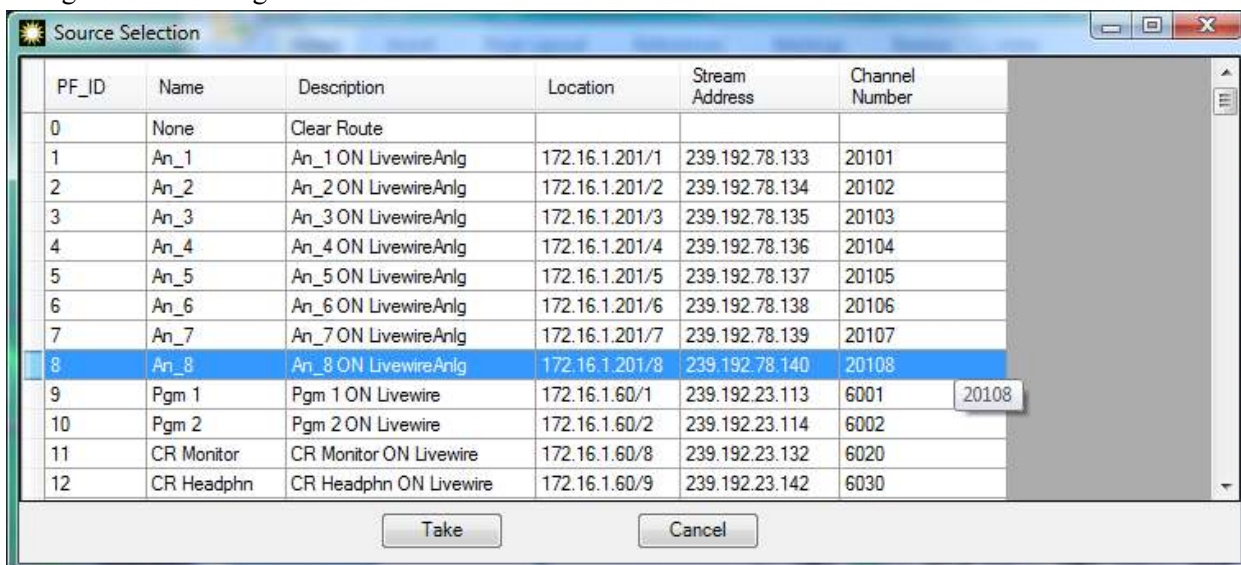
Each Axia Mix engine has a 40 channel virtual mixer in addition to the mixing resources for the Element console. The 40 channel virtual mixer is divided into 8 five channel submixers. Clicking on each submixer in the left side of the control panel will cause the 6 faders (5 channels plus a submix master) to display on the right hand side.



It is possible that you may not see 8 submixers on the left, or all of the faders for a particular submixer on the right. This would be the case if your Administrator has configured VMIXControl to be limited only to the submixers and/or faders you are supposed to use. Master faders are designated by a red slider.

Using a Fader

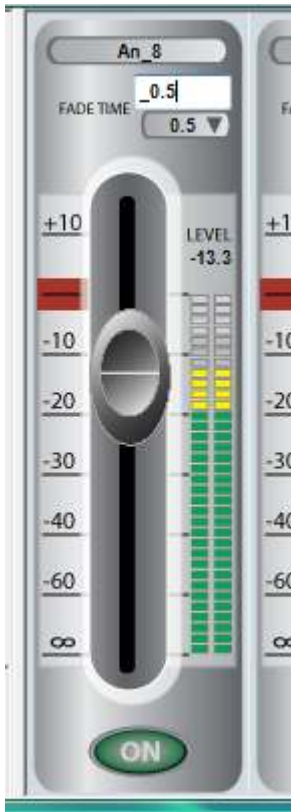
Source Selection: The very top field lists the source that is assigned to each fader. If you click in this field, you will get a list of the sources that are available to be assigned to the fader. In this way you can dynamically change the fader assignments.



Select a new source by double-clicking on it or selecting the source and clicking Take. Click Cancel to exit this screen without making any changes to the fader. Please note that you cannot change the source assignment of a master fader.

Turning a fader on or off and Fade Times: To turn a fader on or off, simply click the large On/Off button at the bottom of the fader. It is important to understand that the fade out or fade in time when turning a fader on or

off is based on the fade up or down time. This means that if there is a fade time for the fader (shown underneath the source assignment label), the audio will fade to the current fader's position over that period of time when being turned on. And it will be faded out over the fade out time when being turned off. This fade time only affects turning a fader on or off and not any real time mixing functions. This allows a softer and more graceful fade in and out when a fader is turned on or off. If you want an instant on, set the fade times to 0. You can modify the fade times by clicking on the fade up or down label on each fader and changing the value. This will change the fade time for that VMIX channel inside the Axia Mix Engine.



Additionally, you cannot turn a master fader on or off. This is not an available option inside the Axia mix engine at this point in time.

Controlling the Fader: The simplest way to control a fader is just to click on the slider and drag it to a new position. As you drag the fader, the Level label will also display the precise level. In addition, if you are using a mouse with a wheel, you can position the mouse over the slider of a fader, and use the wheel to move the fader up and down. Double-clicking the slider will force it to unity gain (0). Finally, you can also click on the Level label and enter a precise value for the fader.

Metering: All of the meters in VMIXControl are peak reading and are based on control information returned from the mix engine. VMIX channel metering is pre-fader, but all master fader metering is post-fader.

SAPortRouter

Overview

SAPortRouter is an application that runs as a service on a computer. It is designed to act as a router of textual data between serial ports and TCP ports. In many ways it is designed to be a software based version of a hardware serial port router, except that it can also route that data to TCP ports as well. This provides a very powerful tool for transferring legacy serial data around a network. This program runs as a service on a computer and can be set to start up automatically at startup. It also includes a configuration application that allows the service to be configured using a simple, intuitive, drag and drop interface. Finally the service has a control port where third party controllers or PathfinderPC Server itself can control the routing of the ports dynamically.

Please Note: At this point in time only ASCII data is supported. We are working on an update which will also support proper binary data but that is not complete yet.

Licensing

The SAPort Router requires a PathfinderPro license to function. This license is entered into the system using the configuration UI. However, SAPortRouter may be installed on multiple computers using that license.

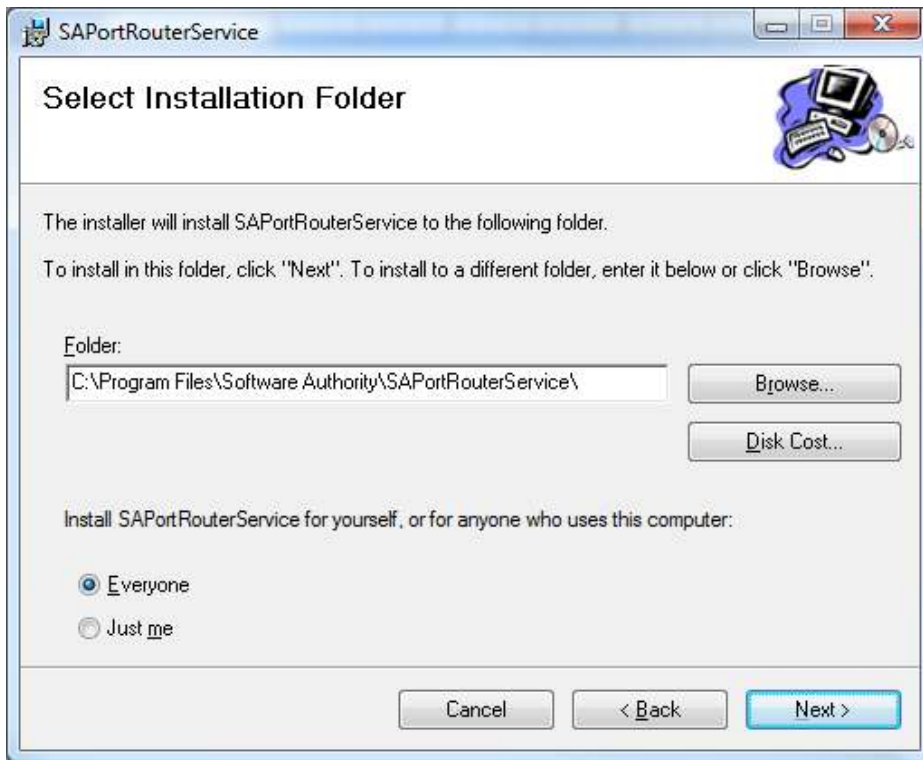
Installation

To install SAPortRouter, simply double click the SAPortRouter installer. Please note that this application requires that the computer already have the .Net framework version 2.0 installed. Most XP/Vista/2003 machines will already have this. If the framework is not present, it can be downloaded and installed through the windows update system.

Vista users - please see the Vista notes at the end of this document about UAC when installing this service.



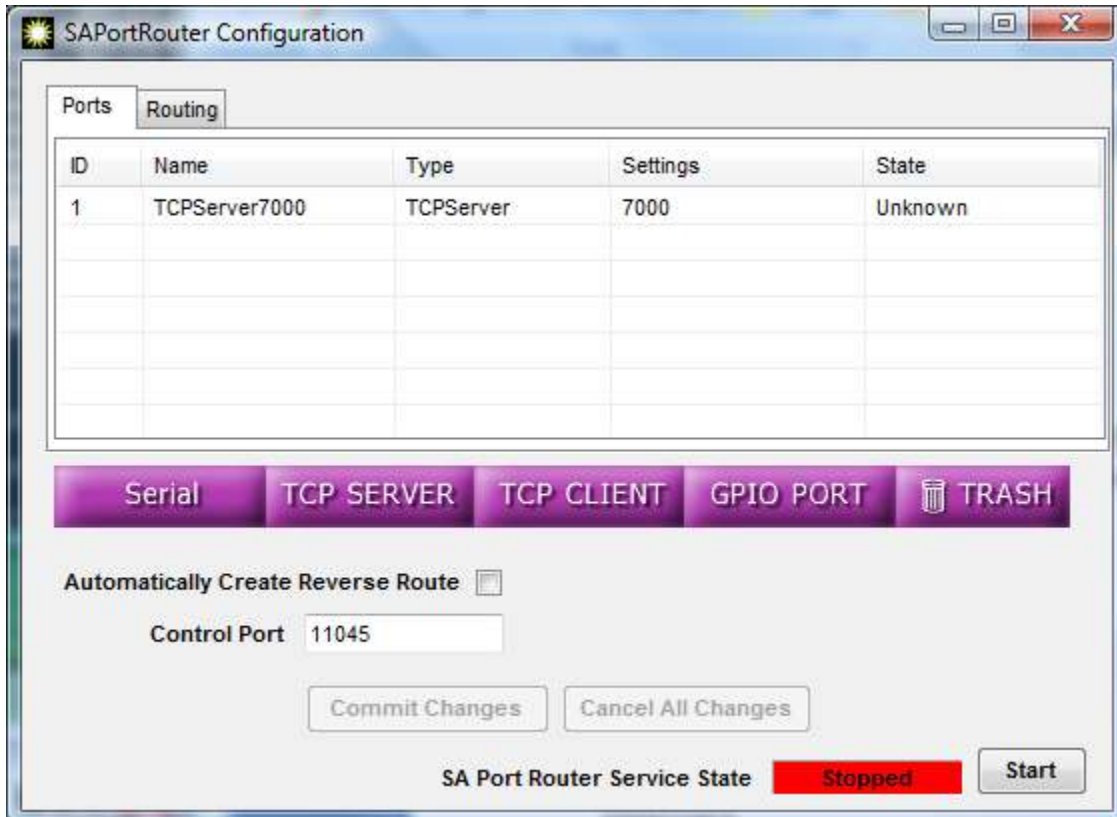
Click Next and select who will be allowed to use the program.



Continue to Click Next and Finish until the application is installed.

Configuring SAPortRouter

To Configure SAPortRouter, Run the SAPortRouterConfig application. Vista users will need to set the application link to Run as Administrator. See the section at the end of this document for details.



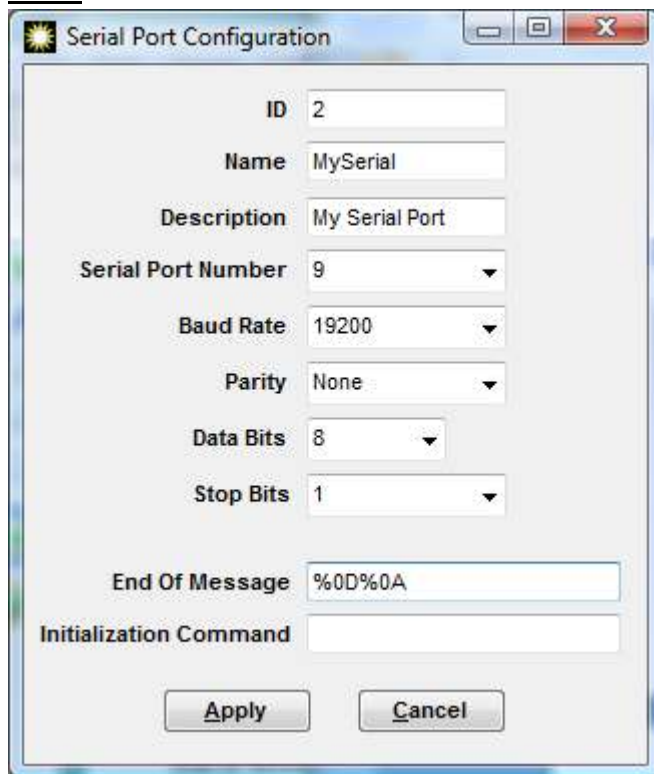
If a licensing window appears, enter your PathfinderPro Request code and license Key.

Creating Ports

To add ports drag the purple button for the type of port you want to create into the grid at the top of the screen. This will cause a configuration window to appear where you can enter the setting information for the port. To delete a port drag it from the grid to the trash. To edit a port that has been created, double-click on it in the grid. When you are done with the configuration, be sure to click on the Commit button to commit the changes. Then use the Start/ Stop button to restart the service.

Port Types

Serial

A screenshot of a 'Serial Port Configuration' dialog box. The dialog has a title bar with a sun icon and standard window controls. It contains several fields and dropdown menus: 'ID' (text box with '2'), 'Name' (text box with 'MySerial'), 'Description' (text box with 'My Serial Port'), 'Serial Port Number' (dropdown menu with '9'), 'Baud Rate' (dropdown menu with '19200'), 'Parity' (dropdown menu with 'None'), 'Data Bits' (dropdown menu with '8'), and 'Stop Bits' (dropdown menu with '1'). At the bottom, there are two text boxes: 'End Of Message' (containing '%0D%0A') and 'Initialization Command' (empty). Below these are 'Apply' and 'Cancel' buttons.

Enter the serial port settings for the port including a name and description if desired.

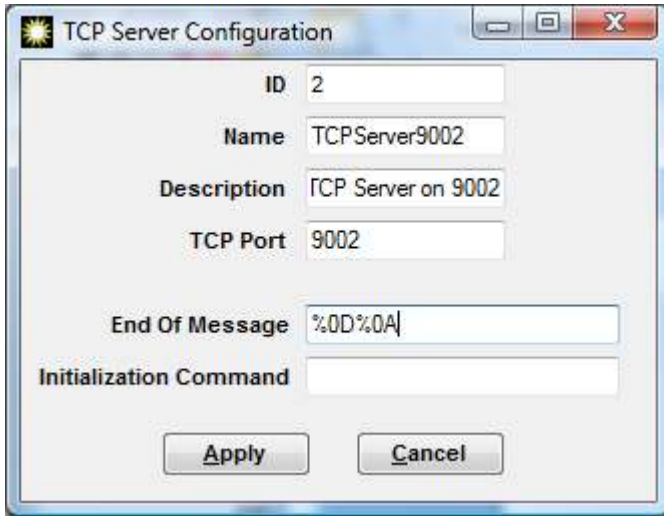
The “End Of Message” field allows you to enter a command which is expected at the end of each message for the protocol that the port will be using. This allows the application to wait until it has a complete message before sending it on to other ports to which this port is routed. You can enter non-typable characters by using % plus a 2 digit ascii hex code. So %0D%0A is equivalent to a carriage return and line feed. If you leave this blank data will be sent as soon as it is received which will often be on a character by character basis.

The initialization command (if present) will be sent each time the port connects or opens.

One trick that many users use with serial ports is to use a null modem cable and a second port to loop the data. For example if you have an application that only understands serial ports, you can take that serial port and loop it

to another port. Then run the SPortRouter application on the second port with a TCP port to convert the serial data to TCP.

TCPServer

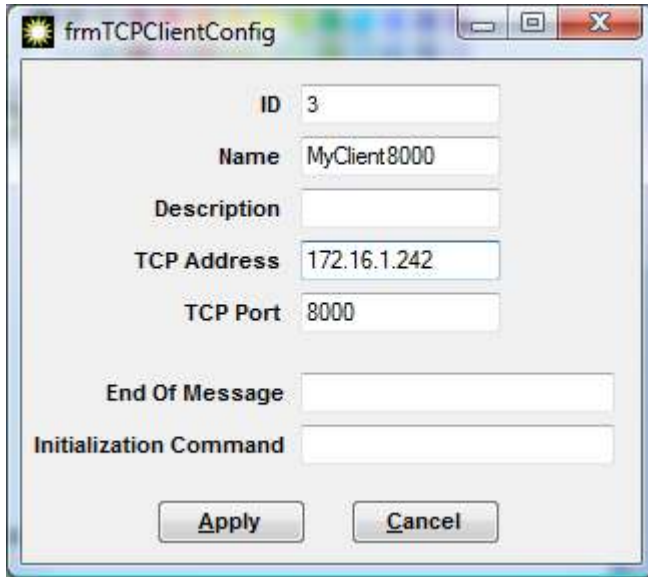


The TCP Server opens a listening TCP connection on this machine on the specified port. It will accept incoming connections on this port from as many TCP clients as you like and accept packets from them to forward to other ports to which this port is routed. If you have multiple TCP clients connected to this port at the same time, some care needs to be taken to make sure each packet is a complete command. Otherwise if two clients try to send data at the same time, it could get spliced in an unforgiving way.

The “End Of Message” field allows you to enter a command which is expected at the end of each message for the protocol that the port will be using. This allows the application to wait until it has a complete message before sending it on to other ports to which this port is routed. This leads to more efficient transmission. You can enter non-typable characters by using % plus a 2 digit ascii hex code. So %0D%0A is equivalent to a carriage return and line feed. If you leave this blank data will be sent as soon as it is received which may be on a character by character basis.

The initialization command (if present) will be sent each time a client port connects.

TCPClient



The screenshot shows a configuration window titled "frmTCPClientConfig". It contains the following fields and values:

Field	Value
ID	3
Name	MyClient8000
Description	
TCP Address	172.16.1.242
TCP Port	8000
End Of Message	
Initialization Command	

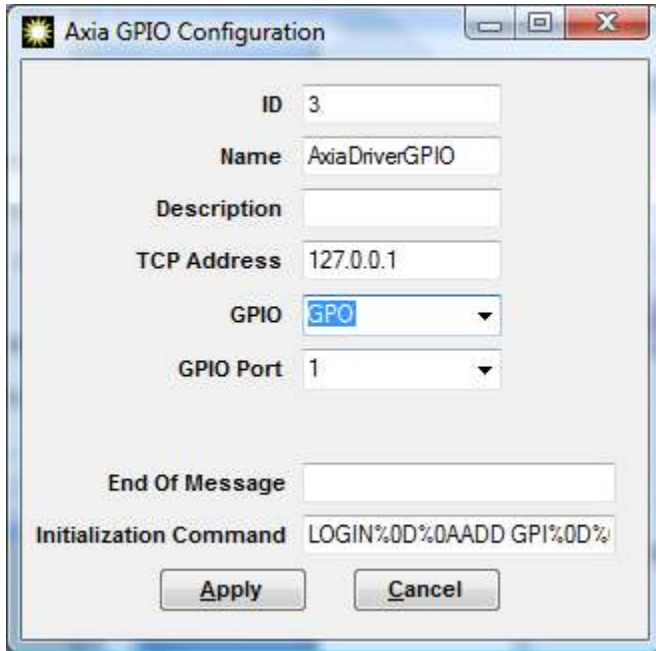
Buttons: Apply, Cancel

This port connects to a remote server at the specified IP address and TCP port. Enter the correct configuration information for this port.

The “End Of Message” field allows you to enter a command which is expected at the end of each message for the protocol that the port will be using. This allows the application to wait until it has a complete message before sending it on to other ports to which this port is routed. This leads to more efficient transmission. You can enter non-typable characters by using % plus a 2 digit ascii hex code. So %0D%0A is equivalent to a carriage return and line feed. If you leave this blank data will be sent as soon as it is received which may be on a character by character basis.

The initialization command (if present) will be sent each time the port connects or opens.

GPIOPort



The screenshot shows a window titled "Axia GPIO Configuration". It contains the following fields and controls:

- ID: 3
- Name: AxiaDriverGPIO
- Description: (empty)
- TCP Address: 127.0.0.1
- GPIO: GPO (dropdown menu)
- GPIO Port: 1 (dropdown menu)
- End Of Message: (empty)
- Initialization Command: LOGIN%0D%0AADD GPI%0D%
- Buttons: Apply, Cancel

The Axia Windows Driver allows textual messages to be sent to it that applications can read if they know the protocol. This allows messages to be sent to a specific Axia driver GPIO port. Fill in the appropriate details for the correct Axia driver. Please note – this is only for communicating with an Axia Windows Driver. This will not work with node or Element GPIO ports. And the computer running the driver must have an application that is aware of these messages.

The “End Of Message” field allows you to enter a command which is expected at the end of each message for the protocol that the port will be using. This allows the application to wait until it has a complete message before sending it on to other ports to which this port is routed. You can enter non-typable characters by using % plus a 2 digit ascii hex code. So %0D%0A is equivalent to a carriage return and line feed. If you leave this blank data will be sent as soon as it is received which will often be on a character by character basis.

The initialization command (if present) will be sent each time the port connects or opens.

Completing the Configuration

Once your configuration is complete, click the CommitChanges button to write the changes to the configuration file, and then Stop and Start the Service.

The “Automatically create reverse route” checkbox means that if you route one port to another, it will automatically also complete the opposite route. This may or may not be desired depending on the application.

Service State

The screenshot shows the 'SAPortRouter Configuration' window with the 'Routing' tab selected. The 'Ports' section contains a table with the following data:

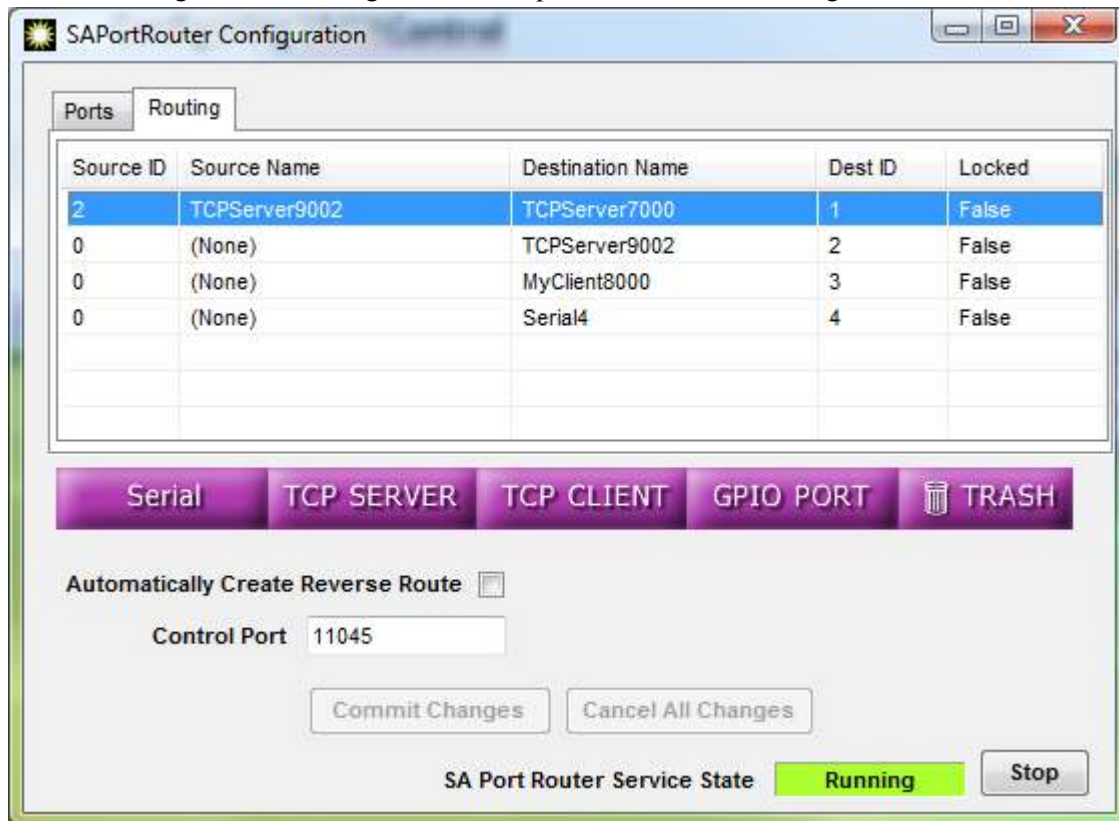
ID	Name	Type	Settings	State
1	TCPServer7000	TCPServer	7000	TCPServer Listenin...
2	TCPServer9002	TCPServer	9002	TCPServer Listenin...
3	MyClient8000	TCPClient	172.16.1.242,8000	TCPClient Connected
4	Serial4	Serial	Com4,9600,N,8,1	Serial Open

Below the table are five buttons: 'Serial', 'TCP SERVER', 'TCP CLIENT', 'GPIO PORT', and 'TRASH'. There is a checkbox for 'Automatically Create Reverse Route' which is unchecked. Below that is a 'Control Port' field with the value '11045'. At the bottom, there are 'Commit Changes' and 'Cancel All Changes' buttons. The 'SA Port Router Service State' is displayed as 'Running' in a green box, with a 'Stop' button next to it.

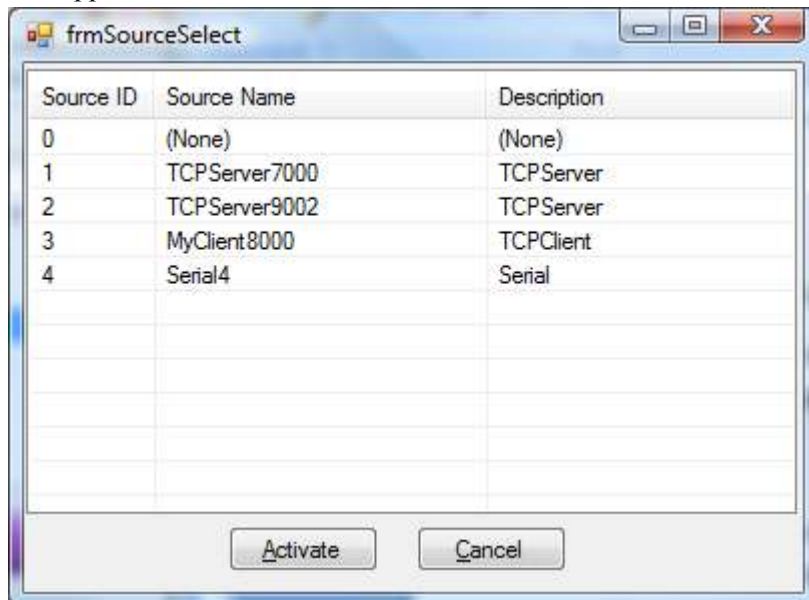
When the service is running, you will be able to see each port, its settings, and its current state. The state will show whether the port is connected, listening, open, etc. This is an excellent troubleshooting tool because you will be able to tell from this screen whether the system is connected to the remote host.

Routing

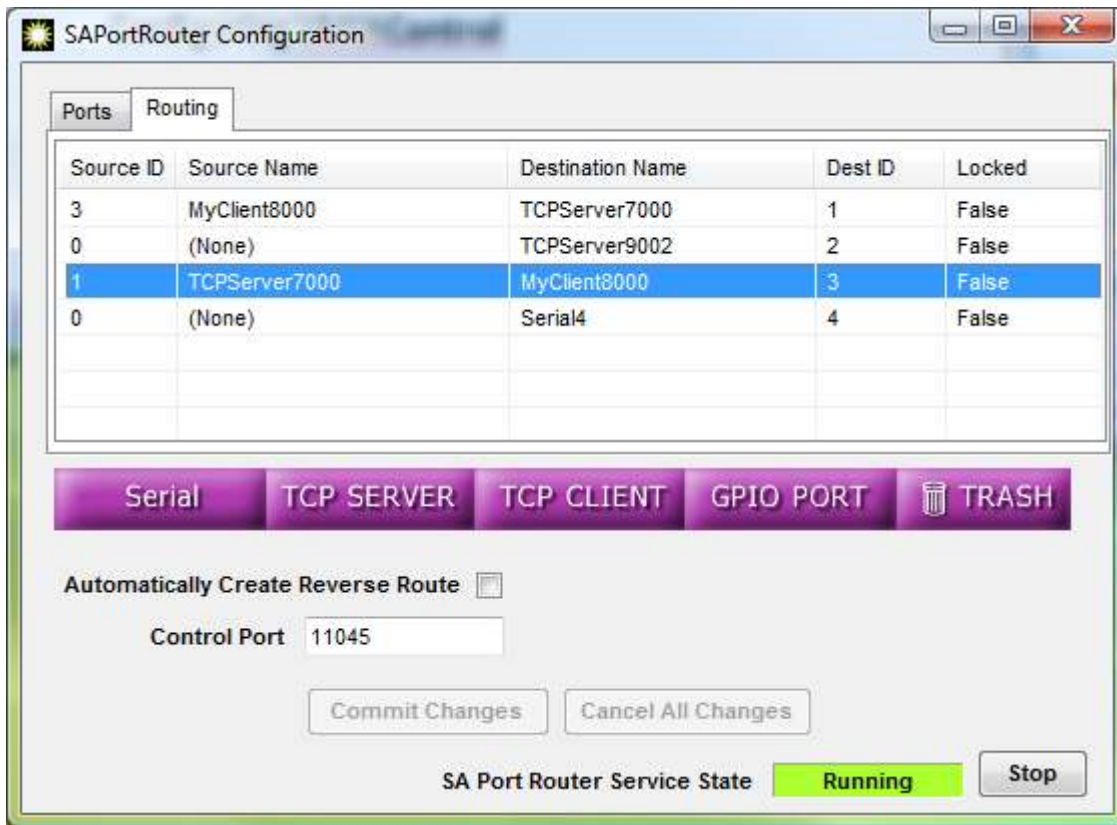
To make changes to the routing between the ports, click on the routing tab.



This tab allows the user to change source to destination port routing. Click on any Destination, and a pop-up box will appear to select the source.



Select the source port and click activate, or double-click the source to complete the route. This only activates the data flow in one direction. To establish bi-directional communication, you need to make the reverse route as well.



Once the route is established, data that comes in one port will be sent out any port that the source is routed to. This happens no matter what style of port it is.

Multiplexing

If you use the control or shift keys to select multiple source ports in the source selection window, you will discover that it is possible to route multiple sources to the same destination. This can be inherently dangerous in an uncontrolled environment, because if multiple source ports are sending data simultaneously, the data could get corrupted when it gets multiplexed together. However, this can be an extremely useful tool in a controlled environment. For example if we have an automation system that needs to send serial port data to Pathfinder, and the facility has a 2 server PathfinderPro cluster, Port Router can be used to send the automation system data to both PathfinderPro servers. Then we multiplex route the return data from both servers back to the single automation system serial port. Finally, in the Pathfinder cluster we set the protocol translator up to only work on the server that currently owns the event system. Then clustering makes sure that only one server at any point in time will be communicating. In this way we can make a single automation system cluster aware.

PathfinderPro Control

SAPortRouter is an actual router. Since this is true, PathfinderPro Server can control it like any other router. Within PathfinderPro Server, add a new router of type SAPortRouter, and configure with the IP address settings for the computer hosting the SAPortRouter application. See the PathfinderPro Server documentation to set this up. Once complete, Pathfinder can dynamically schedule and activate serial routing changes just like any other router.

Third Party Remote Control

SAPortRouter has a TCP port that is dedicated for remote control. This can be used for third party applications which need to make routing changes to SAPortRouter. This protocol can be seen in action by telnetting to the

computer hosting SAPortRouter on port 11045. You can then type a ? to get a list of commands. Type the command name and a ? to get details on how to use the command.

For Example:

```
>>?
```

```
ActivateRoute, RouteStat, PortState, PortName
```

Type the name of a command plus a space and a question mark for detailed help on a command.

Example:

```
ActivateRoute ?
```

```
>>
```

Remote Control Command List

RouteStat – returns the routing status of the system

```
>>RouteStat ?
```

Display Current Route Statistics of a specific destination or all destinations.

If No Specific Destination Point is listed, this command returns Status of all the points.

Return Command is formatted as follows:

```
RouteStat <DestinationNumber> <SourceNumber> <LockCondition>
```

Usage Syntax:

```
RouteStat
```

Or

```
RouteStat <DestinationNumber>
```

```
>>RouteStat
```

```
RouteStat 1 3 False
```

```
RouteStat 2 0 False
```

```
RouteStat 3 1 False
```

```
RouteStat 4 0 False
```

```
>>
```

PortState – returns the state of each port in the system

```
>>PortState ?
```

Displays the Port Type and State for a port or all ports.

If No Specific Port Number is listed this command returns the Port Type and State of all Ports.

Return Command is formatted as follows:

PortState <PortNumber> <PortType> <CurrentState>
Or
PortState <PortNumber> <PortType> <CurrentState><Tab><CurrentSettings>

Usage Syntax:

PortState
Or
PortState <PortNumber>

```
>>PortState
PortState 1 TCPServer Listening - 0 Clients Connected 7000
PortState 2 TCPServer Listening - 0 Clients Connected 9002
PortState 3 TCPClient Connected 172.16.1.242,8000
PortState 4 Serial Open Com4,9600,N,8,1
```

>>

PortName – returns the name and description of each port in the system

```
>>PortName ?
Displays the Port Name and Description for a port or all ports.
```

If No Specific Port Number is listed, this command returns the Port Name and Description of all Ports.

Return Command is formatted as follows:
PortName <PortNumber> <PortName><Tab><PortDescription>

Usage Syntax:
PortName
Or
PortName <PortNumber>

```
>>PortName
PortName 1 TCPServer7000 TCP Port 7000 Server
PortName 2 TCPServer9002 TCP Server on 9002
PortName 3 MyClient8000
PortName 4 Serial4
```

>>

ActivateRoute – Changes a port route

```
>>ActivateRoute ?
Change a particular Route Point.
Usage Syntax:
ActivateRoute <DestinationNumber> <NewSourceNumber>
```

```
>>ActivateRoute 4 2
RouteStat 4 2 False
```

>>

Vista Users

Vista runs under a standard user rights system no matter what user is logged in. However, the service needs to run under Administrative privileges. Therefore, you may need to either run the installer for this application from a command prompt with Administrator privileges or disable UAC to install. Then you can re-enable when you are done. To disable UAC, goto the control panel, click on User Account, and click the link to turn UAC off. You will be warned and asked to reboot the computer. Then install SAPortRouter. Then you can turn UAC back on again and reboot.

Additionally, if you are running on Vista, you will need to run the configuration application as Administrator as well. To do this, make a shortcut to the application on your desktop. Right click on the shortcut, and click properties. Then click Advanced and select Run as Administrator. Click OK. When you launch from this icon, UAC will ask for confirmation, and then you will be able to configure the settings for the service.

Demo Limitations and Licensing

The demo version of this software is functional with the following limitations. The server software shuts down after 120 minutes. It may then be restarted, but it will only run for 120 minutes at a time. In the demo version only three routers may be created on the server. They can be any combination of real or virtual routers. Finally, only the first 5 points of any router may actually be routed with the demo software.

The only thing that is necessary to upgrade the demo version to a fully functional version of the software is a license key for the server application. Simply contact Axia or Software Authority at the number or email listed below to obtain a valid license key. You can either purchase a PathfinderPC or a PathfinderPro license. The differences in these licenses are outlined in a chart at the beginning of this document.

A valid license key must be purchased for each instance of the server application. The client applications may be placed on as many machines as desired.



Contact Information

Any additional questions may be directed to:

Axia Audio
2101 Superior Avenue
Cleveland, OH 44114
USA

Telephone: +1.216.241.7225

E-mail inquiry@AxiaAudio.com



Software Authority, Inc.
6001 West Creek Rd.
Independence, OH 44118
USA

Telephone: +1.216.654.0194

E-mail techsupport@softwareauthority.com



Appendix A

SAProtocol - Software Authority Control Protocol

Overview

The Software Authority control protocol is designed to be a simple to use text protocol for controlling PathfinderPC Server functions. It can be used both for third party control applications and as a simple way to type commands to the system for remote control or debugging purposes.

Each command in the protocol must be terminated with a carriage return and line feed. This means if you are trying to use Hyperterminal to connect to the server, you need to turn on the Send Line Ends with Line Feeds option. Telnet sessions in a Windows terminal window do this by default.

Command List – Client to PathfinderPC Server

Login <UserName> <Password>

Logs in to the PathfinderPC Server system. The Login username and password must match a valid user account in the PathfinderPC Server User database.

Logoff

Logs Off The System

? <Command>

The ? returns the list of commands or help for a specified command. If no command is specified, this returns the list of available commands. Until the user is successfully logged in, the only available command will be Login.

Example:

```
>>?  
RouterNames, SourceNames, DestNames, Snapshots, RouteStat, ActivateSnap, Activat  
eScene, ActivateRoute, SetLock, RunScript, GPIStat, GPOStat, TriggerGPI, Trigger  
GPO, TriggerAllDestGPI, TriggerAllDestGPO, ShowSubscriptions, VMIXOn, VMIXOff, V  
MIXGain, VMIXTimeUp, VMIXTimeDown, VMIXState, Login, Logoff, GetList, GetListVer  
sion
```

Type Help and the name of the command for more information.

```
>>? RouterNames  
Display Names of all routers on the System.  
Usage Syntax:  
RouterNames
```

```
>>
```

RouterNames

Returns Names of all Routers in the system using the following format:

Begin RouterNames

<NUM1> Name1

```

    <NUM2> Name2
    <NUM3> Name3
    ...
End RouterNames

```

Example:

```

>>RouterNames
Begin RouterNames
  1 DanNonTerm
  2 TestGPIO
  3 Dan_Virtual
  4 DanTestRouter
  5 OffsetLockTest
  6 Test256SAPort
  9 BT_A
  10 BT_B
End RouterNames
>>

```

SourceNames <RouterNumber>

Returns Names of all Source Points on a Router. Minimally, the command will return the Number of the point in the router and the name. There may be additional tab delineated fields after the name depending on the type of router. For example Axia audio routers will also return a description field, host ip field, hostname, Host IO number, IO Address, channel number, and stream address. There will be a space between the number and the first field. All fields after the first field are tab delineated.

Begin SourceNames - <RouterNumber>

```

  <NUM1> Name1
  <NUM2> Name2
  <NUM3> Name3

```

...

End SourceNames - <RouterNumber>

Example:

```

>>sourcenames 1
Begin SourceNames - 1
  1 An_1 An_1 ON LivewireAnlg 172.16.1.201 LivewireAnlg 1 20101 239.192.78.133
  2 An_2 An_2 ON LivewireAnlg 172.16.1.201 LivewireAnlg 2 20102 239.192.78.134
  3 An_3 An_3 ON LivewireAnlg 172.16.1.201 LivewireAnlg 3 20103 239.192.78.135
  4 An_4 An_4 ON LivewireAnlg 172.16.1.201 LivewireAnlg 4 20104 239.192.78.136
  5 An_5 An_5 ON LivewireAnlg 172.16.1.201 LivewireAnlg 5 20105 239.192.78.137
  6 An_6 An_6 ON LivewireAnlg 172.16.1.201 LivewireAnlg 6 20106 239.192.78.138
  7 An_7 An_7 ON LivewireAnlg 172.16.1.201 LivewireAnlg 7 20107 239.192.78.139
  8 An_8 An_8 ON LivewireAnlg 172.16.1.201 LivewireAnlg 8 20108 239.192.78.140
  9 Pgm 1 Pgm 1 ON Livewire 172.16.1.60 Livewire 1 6001 239.192.23.113
  10 Pgm 2 Pgm 2 ON Livewire 172.16.1.60 Livewire 2 6002 239.192.23.114End SourceNames
- 1
>>

```

DestNames <RouterNumber>

Returns Names of all Destination Points on a Router. Minimally, the command will return the Number of the point in the router and the name. There may be additional tab delineated fields after the name depending

on the type of router. For example Axia audio routers will also return a description field, host ip field, hostname, and Host IO number. There will be a space between the number and the first field. All fields after the first field are tab delineated.

```
Begin DestNames - <RouterNumber>
  <NUM1> Name1
  <NUM2> Name2
  <NUM3> Name3
  ...
End DestNames - <RouterNumber>
```

Example:

```
>>destnames 1
Begin DestNames - 1
  1 Destination_1 Destination_1 ON LivewireAnlg 172.16.1.201 LivewireAnlg 1
  2 Destination_2 Destination_2 ON LivewireAnlg 172.16.1.201 LivewireAnlg 2
  3 Destination_3 Destination_3 ON LivewireAnlg 172.16.1.201 LivewireAnlg 3
  4 Destination_4 Destination_4 ON LivewireAnlg 172.16.1.201 LivewireAnlg 4
  5 Desitnation_5 Desitnation_5 ON LivewireAnlg 172.16.1.201 LivewireAnlg 5
  6 Destination_6 Destination_6 ON LivewireAnlg 172.16.1.201 LivewireAnlg 6
  7 Destination_7 Destination_7 ON LivewireAnlg 172.16.1.201 LivewireAnlg 7
  8 Destination 8 Destination 8 ON LivewireAnlg 172.16.1.201 LivewireAnlg 8
End DestNames - 1
>>
```

SnapShots <RouterNumber>

Returns Names of all SnapShots on a Router. Snapshots are also referred to as scene changes.

```
Begin SnapshotNames - <RouterNumber>
  <NUM1> Name1
  <NUM2> Name2
  <NUM3> Name3
  ...
End SnapshotNames - <RouterNumber>
```

Example:

```
>>snapshots
snapshots 1
Begin SnapshotNames - 1
  1to4
  Almost1To8
  BadRoute
  Clear1And2
  ForCharles
  ForDan
  Test
End SnapshotNames - 1
>>
```

RouteStat <RouterNumber> <DestNum|Null>

Returns the status of a single route point in a router or all route points in a router. If nothing is sent in the RouteStat Dest Field, the command returns the status of all points. The response listed below is also sent whenever a route point changes to show the change in status.

```
RouteStat <RouterNum> <DestNum> <SourceNum> <Lock (True or False)>
RouteStat <RouterNum> <DestNum> <SourceNum> <Lock (True or False)>
RouteStat <RouterNum> <DestNum> <SourceNum> <Lock (True or False)>
RouteStat <RouterNum> <DestNum> <SourceNum> <Lock (True or False)>
```

Example:

```
>>RouteStat 1
RouteStat 1 1 9 False
RouteStat 1 2 25 False
RouteStat 1 3 62 False
RouteStat 1 4 204 False
RouteStat 1 5 16 False
RouteStat 1 6 41 False
RouteStat 1 7 50 False
RouteStat 1 8 11 False
RouteStat 1 9 1 False
>>
```

ActivateSnap <RouterNumber> <SnapShotName>

This command activates a snapshot also called a scene change on a specific router. The command returns that a snapshot has been initiated. Any routes that change as a result of the snapshot will get printed out as RouteStat responses. This may include route change commands from virtual routers that are built on the base router on which the scene change is being activated.

Example:

```
>>activatesnap 1 Clear1And2
Snapshot Initiated
>>RouteStat 3 1 0 False
>>RouteStat 4 1 0 False
>>RouteStat 9 1 0 False
>>RouteStat 1 1 0 False
>>RouteStat 3 2 0 False
>>RouteStat 4 2 0 False
>>RouteStat 9 2 0 False
>>RouteStat 1 2 0 False
>>
```

ActivateScene <RouterNumber> <SnapShotName>

See ActivateSnap. This is just an alias for that command.

ActivateRoute <RouterNumber> <DestNumber> <SourceNumber>

Tries to initiate a route change. If the change is successful, a RouteStat command will be returned.

Example:

```
>>ActivateRoute 1 1 2
```

```
Route Change Initiated
>>RouteStat 1 1 2 False
>>
```

SetLock <RouterNumber> <DestNumber> <True|False>

Tries to change the lock status of a destination. If successful, a RouteStat command will be returned.

Example:

```
Lock Change Initiated
>>RouteStat 1 1 2 True
>>
```

RunScript <ScriptName> <CallingControlName>, <CommandLineParameters>

This command starts a custom script. This script must exist in the PFSScripts folder.

CallingControlName is optional but offers a way to pass the name of an object that should be identified as originating the script to the script.

CommandLineParameters is also optional and allows any additional information to be passed into the Script. All remaining text on the command line after the command is placed into the command line parameter and is available to the script to use.

```
RunScript MyScript MySourceObject, MyCommandLineText, AndAlsoThisCommandLineText
```

Example:

```
>>RunScript CheckACRow
Starting Script CheckACRow
>>
```

GPIStat <RouterNumber> <IONumber>

This command requests the state of a GPIPoint or all of the GPI points on a GPIO router. It returns a GPIStat Command. If the IO number is not present, the command will return all GPI statuses for the router. The

Router must be a GPIO router or an error will be returned.

Example:

```
>>GPIStat 2
GPIStat 2 1 hhhhh
GPIStat 2 2 hhhhh
GPIStat 2 3 hhhhh
GPIStat 2 4 hhhhh
GPIStat 2 5 hhhhh
GPIStat 2 6 hhhhh
GPIStat 2 7 hhhhh
GPIStat 2 8 hhhhh
GPIStat 2 9 hhhhh
>>
```

GPOStat <RouterNumber> <IONumber>

This command requests the state of a GPOPoint or all of the GPO points on a GPIO router. It returns a GPOStat Command. If the IO number is not present, the command will return all GPI statuses for the router.

The Router must be a GPIO router or an error will be returned.

Example:

```
>>GPOStat 2
GPOStat 2 1 hhhh
GPOStat 2 2 hhhh
GPOStat 2 3 hhhh
GPOStat 2 4 hhhh
GPOStat 2 5 hhhh
GPOStat 2 6 hhhh
GPOStat 2 7 hhhh
GPOStat 2 8 hhhh
GPOStat 2 9 hhhh
>>
```

TriggerGPI <RouterNumber> <DestinationNumber> <State 'llll|hhhh|lhx... . . '> <PulseDuration in Milliseconds>

Trigger a GPI contact closure on a GPIO Router or Virtual router with GPI mappings. It is important to know that in the Axia GPIO Router, only windows driver GPI points can be triggered. While hardware GPO points can be triggered with the next command shown below, GPIs cannot.

Duration is optional and will cause a pulse in milliseconds rather than a steady state change. The State is a list of the pins to set to high or low on the port. Any pins with l are set to high, and any pins with h are set to low. Pins with x are unaltered. Therefore llxxh would set pins 1 and 2 to low, 3 and 4 would not get changed from their current state, and 5 would be set to high.

If the command is successful, a GPIStat command will be returned.

Example:

```
>>TriggerGPI 2 3 lxxxx 500
>>GPIStat 2 3 lhxxx
>>GPIStat 2 3 hhhh
```

TriggerGPO <RouterNumber> <DestinationNumber> <State 'llll|hhhh|lhx... . . '> <PulseDuration in Milliseconds>

Trigger a GPO contact closure on a GPIO Router or Virtual router with GPO mappings. Duration is optional and will cause a pulse in milliseconds rather than a steady state change. The State is a list of the pins to set to high or low on the port. Any pins with l are set to high, and any pins with h are set to low. Pins with x are unaltered. Therefore llxxh would set pins 1 and 2 to low, 3 and 4 would not get changed from their current state, and 5 would be set to high.

If the command is successful, a GPOStat command will be returned.

Example:

```
>>TriggerGPO 2 3 lxxxx 500
>>GPOStat 2 3 lhxxx
```

```
>>GPOStat 2 3 hhhhh
```

TriggerAllDestGPO <RouterNumber> <SourceNumber> <State 'llll|hhhhh|lhx... '> <PulseDuration in Milliseconds>

Trigger GPO contact closures on a GPIO Router or virtual router with GPO mappings. Given a source it will trip the closure on all destinations the source is routed to. Duration is optional and will cause a pulse in milliseconds rather than a steady state change. The State is a list of the pins to set to high or low on the port. Any pins with l are set to high, and any pins with h are set to low. Pins with x are unaltered.

Therefore llxxh would set pins 1 and 2 to low, 3 and 4 would not get changed from their current state, and 5 would be set to high.

Example:

```
>>TriggerAllDestGPO 2 3 lxxxx 500
>>GPOStat 2 3 lhhhh
>>GPOStat 2 3 hhhhh
```

TriggerAllDestGPI <RouterNumber> <SourceNumber> <State 'llll|hhhhh|lhx... '> <PulseDuration in Milliseconds>

Trigger GPI contact closures on a GPIO Router or virtual router with GPI mappings. GPIs can only be triggered on Axia Drivers. Hardware equipment will be unaffected. Given a source it will trip the closure on all GPI destinations the source is routed to. Duration is optional and will cause a pulse in milliseconds rather than a steady state change. The State is a list of the pins to set to high or low on the port. Any pins with l are set to high, and any pins with h are set to low. Pins with x are unaltered.

Therefore llxxh would set pins 1 and 2 to low, 3 and 4 would not get changed from their current state, and 5 would be set to high.

Example:

```
>>TriggerAllDestGPI 2 3 lxxxx 500
>>GPOStat 2 3 lhhhh
>>GPOStat 2 3 hhhhh
```

ShowSubscriptions <RouterNumber>

Shows the Silence Detection subscriptions for a given router. This is only valid for Axia Audio Routers at this point in time. This command is primarily used for debugging. The return value will have a source column, a destination column, a type column, and the name of the object that asked for the subscription. The type column is for future use and will always be 2 for now. The return will list all the subscriptions the server has, and then relist them by device.

Example:

```
>>ShowSubscriptions 1
showsubscriptions 1
-1 9 2 StackEvent_Test_Junk_10
```

```
23 -1 2 ProtTrans_1_ACU1_Prophet
50 -1 2 ProtTrans_1_ACU1_Prophet
51 -1 2 ProtTrans_1_ACU1_Prophet
52 -1 2 ProtTrans_1_ACU1_Prophet
53 -1 2 ProtTrans_1_ACU1_Prophet
54 -1 2 ProtTrans_1_ACU1_Prophet
55 -1 2 ProtTrans_1_ACU1_Prophet
56 -1 2 ProtTrans_1_ACU1_Prophet
-1 3 2 Events_1_EASTest
62 -1 2 Events_1_EASTest
63 -1 2 Events_1_EASTest
```

Device - 172.16.1.201

```
-1 3 2 Events_1_EASTest
```

Device - 172.16.1.60

```
60 -1 2 ProtTrans_1_ACU1_Prophet
61 -1 2 ProtTrans_1_ACU1_Prophet
62 -1 2 ProtTrans_1_ACU1_Prophet
63 -1 2 ProtTrans_1_ACU1_Prophet
64 -1 2 ProtTrans_1_ACU1_Prophet
65 -1 2 ProtTrans_1_ACU1_Prophet
66 -1 2 ProtTrans_1_ACU1_Prophet
67 -1 2 ProtTrans_1_ACU1_Prophet
```

>>

VMIXOn <Element/EngineIP> <SubmixerNumber> <ChannelNumber>

Turns on a VMIX Channel on an Element/Engine. This is only valid for Axia Audio Routers at this point in time.

Example:

```
>>VMIXOn 172.16.1.60 1 5
OK
>>
```

VMIXOff <ElementIP> <SubmixerNumber> <ChannelNumber>

Turns off a VMIX Channel on an Element. This is only valid for Axia Audio Routers at this point in time.

Example:

```
>>VMIXOff 172.16.1.60 1 5
OK
>>
```

VMIXGain <ElementIP> <SubmixerNumber> <ChannelNumber> <Gain>

Changes the gain on a VMIX Channel on an Element. This is only valid for Axia Audio Routers at this point in time.

Example:

```
>>VMIXGain 172.16.1.60 1 5 0.0
OK
>>
```

VMIXTimeUp <ElementIP> <SubmixerNumber> <ChannelNumber> <TimeUp>

Changes the fade up time on a VMIX Channel on an Element in seconds. This is only valid for Axia Audio Routers at this point in time.

Example:

```
>>VMIXTimeUp 172.16.1.60 1 5 0.0
OK
>>
```

VMIXTimeDown <ElementIP> <SubmixerNumber> <ChannelNumber> <TimeDown>

Changes the fade down time on a VMIX Channel on an Element in seconds. This is only valid for Axia Audio Routers at this point in time.

Example:

```
>>VMIXTimeDown 172.16.1.60 1 5 0.0
OK
>>
```

VMIXState <ElementIP> <SubmixerNumber>

Gets the state of all channels of a VMixer. This is only valid for Axia Audio Routers at this point in time. This command is not fully implemented at this point in time.

Example:

```
>>VMIXState 172.16.1.50 1
SUB#1.IN#1   ON   0.0 db 0.5 Secs   0.5 Secs
SUB#1.IN#2   OFF  0.0 db 0.5 Secs   0.5 Secs
SUB#1.IN#3   OFF  0.0 db 0.5 Secs   0.5 Secs
SUB#1.IN#4   OFF  0.0 db 0.5 Secs   0.5 Secs
SUB#1.IN#5   OFF  0.0 db 0.5 Secs   0.5 Secs
```

GetList

This command returns lists of system information. See the GetList function documentation for details on this command and its usage.

GetListVersion

This command returns a version stamp for certain lists of system information. See the GetList function documentation for details on this command and its usage.

Server To Client Commands Only

UserCommand

This allows a script or Stack Event to send custom user information to the protocol translator port.

Example:

```
Begin User Command
<User Information as defined in stack event or script>
End User Command
```

RouteStat <RouterNumber> <DestinationNumber> <SourceNumber> <Locked (True|False)>

This command is issued by the server any time a route state changes.

Example:

>>RouteStat 1 2 0 False

GPIStat <RouterNumber> <DestinationNumber> <GPIState>

Indicates a GPI has changed. See TriggerGPI for a description of the State Field.

Example:

>>GPIStat 2 3 lhhhh

GPOStat <RouterNumber> <DestinationNumber> <GPIState>

Indicates a GPO has changed. See TriggerGPO for a description of the State Field.

Example:

>>GPOStat 2 3 lhhhh

Appendix B - GetList Function for Protocol Translator

Overview

This function will be available through the Software Authority Protocol Translator and will provide lists of available objects of different types to third party developers and outside control and UI applications. It will provide lists of routers, sources, destinations, Elements, etc. This specification is a work in progress. To use this function, see the PathfinderPCServer instructions on how to add a Software Authority Protocol Translator into the system. You be logged into the translator with a valid user name and password before the GetList Function will be available.

Scenarios

- 1) John the sedentary station manager decides he wants to build a new stack event to start his coffee brewing when he loads his profile in the morning. Since the IT department will not let him near the server room, and his sedentary lifestyle makes the flights of stairs abhorrent to get there anyway, it is important that he be able to build this new event from a simple User Interface on his workstation that can then be copied or uploaded to the server. (See stack event UI tool spec in development). In order for this to work the new UI tool must be able to obtain the lists of closures and profiles so that he can select his profile and the GPIO contact closure for the coffee maker. Enter GetList Function.
- 2) A new Automation System developer, Automated Disc Jockeys, wants to support Axia. Since PFS already does all of the hard work of finding and communicating with all of the equipment, it makes sense for the automation system to talk through PFS. But the system designer wants to be able to display the route names in their User Interface for macro development, etc. Enter GetList Function.

Nongoals

- 1) While this function will duplicate some already existing listing functions, we will leave the other functions intact for existing users in the field. We will not phase out those functions.
- 2) Within PFS are already several functions for obtaining lists of routers, sources, Destinations, etc. Where possible these will be reused, but if formatting does not match, we will be willing to make duplicate functions to create the list rather than shoe-horning the existing one into supporting the new format as well or redeveloping the format across the application as that is likely to break more than it will fix.

Usage Notes

In order to reduce server load and make the client application able to appear more responsive when large lists are involved, each call to GetList will also return a Version number based on date, time, and a number. The return from GetList should then be cached by the calling application along with the version number so that it may be used repeatedly without re-downloading the list. The calling

application can check if the list has changed by using the GetListVersion call which will return only the version of the specific List in question. If the version has not changed from what is in the client application's cache, then the client already has up to date information.

Some commands may also list aliases for the list types. These aliases may be used in place of the full list type argument.

Key

- 1) In this document CRLF means a carriage return followed by a linefeed.
- 2) All commands need to be terminated with a CRLF
- 3) All lines in returned commands will be terminated with CRLF

Syntax

GetList <ListType> <Arguments>

Examples:

- 1) GetList Routers
- 2) GetList SourceNames 1

The return from each command will present rows of data separated by CRLF. The command return will always start with a line that says BeginList and a recap of the ListType and Arguments followed by a space and a version number.

```
BeginList SourceNames 1 2007-07-14_14-26-07_001
```

The version number is the date and time followed by a three digit number in the format yyyy-mm-dd_hh-mm-ss_XXX.

It will end with a line that says EndList. Columns will be separated by tabs. The first row after the BeginList line will always carry column names. Each row's first column will be an availability column which will define whether the row's resources are available for use, and if not why. This column will use the following key:

- 0 = Available
- 1 = User Locked
- 2 = System Locked
- 3 = Disabled Stream
- 4 = Host Device is offline
- 5 = Service Not Running

The availability column is important because the application will usually want to skip and not display resources that are not available. But in some cases you will want to display all resources regardless of whether they are available. For example, Engine Destination routing may not be changed by PFS. So a route change action should not display these destinations as available for

making a route change action. However, you may want to make a qualifier that executes some action if the Engine itself changes the route assigned to one of its destinations. So the qualifier section should display all resources whether they are available or not.

TechNotes:
 The XML Configuration for the stack event properties for the new Flow Chart Stack Event UI should include several fields to determine what is displayed from the returned list.

<DisplayAvailability>0,1,4,5</ DisplayAvailability >
 This defines that rows that are system locked or disabled should not be listed in the UI.

<DisplayColumns>2,3</DisplayColumns>
 This defines which columns of returned data will be displayed – Column counts start with 0 which is always the availability column.

<StoredColumn>1</StoredColumn>
 This defines which single column of returned data will be stored as the selected value when the stack event property is stored.

In order to reduce the server load the list and its version number should be cached by the calling application. Use the GetListVersion to see if the list has changed before calling GetList for the same list again.

Return Example:

(Note: The example below has multiple tabs between some columns for readability. The actual implementation will only have 1 tab between columns.)

BeginList	SourceNames	1	2007-07-14_14-26-07_001
Avail	ID	Name	Description
0	1	MySource	This is my Source
0	2	YourSource	This is your Source
EndList			

Required Lists

- 1) Routers – Returns the list of routers in the system (Aliases – rtr)

Command

GetList Routers <RouterType>

If RouterType is absent a list of all routers will be returned.

Available Device Types include:

- ZSystems
- Sigma
- Videoquip
- NTIVEemux
- SAPort
- AxiaAudio
- AxiaGPIO
- Metered (Only returns routers that support metering and therefore silence detection/clipping/audio present)
- GPIO (Only returns GPIO style routers)
- Virtual
- Gateway
- Generic

Example:

GetList Routers AxiaAudio

Return Columns

- Availability
- ID
- Name
- Description
- Type
- SourceCount
- DestinationCount

Example:

BeginList Routers 2007-07-14_13-15-22_001							
Avail	ID	Name	Description	Type	SourceCount	DestinationCount	
0	1	WZZZ Audio	Main Audio Router		AxiaAudio	758	323
0	2	WZZZ GPIO	Main GPIO Router		AxiaGPIO	253	253
EndList							

- 2) SourceNames – Returns the list of source names in a router in the system (Aliases – source, src, sn)

Command

GetList SourceNames <RouterNumber>

Example:

GetList SourceNames 1

Return Columns

- Availability
- ID
- Name
- Description

Example:

BeginList SourceNames 1 2007-07-14_13-15-22_001			
Avail	ID	Name	Description
0	1	MySource	This is my Source
0	2	YourSource	This is your Source
EndList			

- 3) DestinationNames – Returns the list of destination names in a router in the system (Aliases – dest, dst, dn)

Command

GetList DestinationNames <RouterNumber>

Example:

GetList DestinationNames 1

Return Columns

Availability

ID

Name

Description

Example:

```
BeginList DestinationNames 1 2007-07-14_13-15-22_001
Avail    ID      Name      Description
0        1      MyDestination  This is my Destination
0        2      YourDestination This is your Destination
EndList
```

4) RouteStats – Returns the routing state of a router in the system (Aliases – rs)

Command

GetList RouteStats <RouterNumber>

Example:

GetList RouteStats 1

Return Columns

Availability (Follows destination availability and lock state)

SourceID

DestinationID

Lock (T or F)

Example:

```
BeginList RouteStats 1 2007-07-14_13-15-22_001
Avail    SourceID  DestinationID  Lock
0        1          1              T
0        2          99             F
EndList
```

5) Scenes – Returns the Scenes of a router in the system (Aliases – sc)

Command

GetList Scenes <RouterNumber>

Example:

GetList Scenes 1

Return Columns

Availability

Name

Example:

```
BeginList Scenes 1 2007-07-14_13-15-22_001
```

```
Avail    Name
0        DanShowScene
0        MarkShowScene
EndList
```

6) AxiaDevices – Returns a list of Axia Devices in the system according to their type (Aliases – ad)

Command

GetList AxiaDevices <DeviceType>

If DeviceType is blank – returns all Axia devices

Available Device Types include:

```
AudioNode
GPIO
Driver
Element
Engine
```

Example:

```
GetList AxiaDevices Element
```

Return Columns

```
Availability
Name
IP
Type
```

Example:

```
BeginList AxiaDevices Element 2007-07-14_13-15-22_001
Avail    Name        IP        Type
0        DanElement  172.16.1.4  Element
0        Studio3     172.16.1.5  Element
EndList
```

7) ElementProfiles – Returns a list of Profiles on a specific Element (Aliases – ep)

Command

GetList ElementProfiles <ElementIP>

Example:

```
GetList ElementProfiles 172.16.1.4
```

Return Columns

```
Availability
ID
Name
Example:
```

```
BeginList ElementProfiles 172.16.1.4 2007-07-14_13-15-22_001
```

```

Avail    ID    Name
0        1    DanShow
0        2    MarkShow
EndList

```

8) ProtocolTranslators – Returns a list of Protocol Translators in the System (Aliases – pt)

Command

GetList ProtocolTranslators

Example:

```
GetList ProtocolTranslators
```

Return Columns

Availability

ID

Name

Description

Type

ConnectionStyle (Serial/TCP/UDP/UDPSend)

ConnectionPort

Example:

```

BeginList ProtocolTranslators 2007-07-14_13-15-22_001
Avail    ID    Name          Description      Type              ConnectionStyle  ConnectionPort
0        1    My Translator  For Me          SA Protocol Translator  TCP              8000
0        2    ProphetInterface  For Prophet     Probel General Router  Serial           3
EndList

```

9) UserPanels – Returns a list of User Panels in the System (Aliases – up)

Command

GetList UserPanels

Example:

```
GetList UserPanels
```

Return Columns

Availability

PanelName

FileName

Example:

```

BeginList UserPanels 172.16.1.4 2007-07-14_13-15-22_001
Avail    PanelName  FileName
0        DanPanel  DanPanel.xml
0        331      331.xml
EndList

```

10) PanelControls – Returns a list of Panel Controls in a specific Panel by type (Aliases – pc)

Command

GetList PanelControls <PanelName> <ControlType>

If ControlType is blank – returns all Controls on the Panel

Available Control Types include:

Label

Button

Form

Example:

```
GetList PanelControls DanPanel Button
```

Return Columns

Availability

ControlName

ControlType

Example:

```
BeginList PanelControls Button 2007-07-14_13-15-22_001
```

Avail	ControlName	ControlType
0	DanPanel.DanButton	Button
0	DanPanel.HeadLabel	Label

```
EndList
```

11) SourceAddress – Returns the list of source names in a router in the system (Aliases – sourceadd, srcadd, sa)

Command

GetList SourceAddress <RouterNumber> <SourceNumber>

If SourceNumber is blank – returns all Addresses for the Router

Example:

```
GetList SourceAddress 1
```

Return Columns

Availability

ID – Pathfinder ID

Location – For Virtual Routers, this will contain the Source’s mapping to base routers. For Axia Routers this will contain the IP Address of the host device and a slash with the port number (172.16.1.3/5). For all other Routers this will be empty.

StreamAddress - For Axia Audio Routers, this will contain the multicast stream address for the Source. For all others, this will be blank.

ChannelNumber – For Axia Audio Routers, this will contain the Channel number derived from the stream address for the source. For all others, this will be blank.

Example:

```
BeginList SourceAddress 1 2007-07-14_13-15-22_001
Avail      ID      Location      StreamAddress  ChannelNumber
0          1      127.16.1.3/1  239.192.0.1   1
0          2      127.16.1.3/2  239.192.0.2   2
EndList
```

12) DestinationAddress – Returns the list of Destination names in a router in the system (Aliases – destadd, dstadd, da)

Command

GetList DestinationAddress <RouterNumber> <DestinationNumber>

If DestinationNumber is blank – returns all Addresses for the Router

Example:

```
GetList DestinationAddress 1
```

Return Columns

Availability

ID – Pathfinder ID

Location – For Virtual Routers, this will contain the Destination’s mapping to base routers. For Axia Routers this will contain the IP Address of the host device and a slash with the port number (172.16.1.3/5). For all other Routers this will just be empty.

Example:

```
BeginList DestinationAddress 1 2007-07-14_13-15-22_001
Avail      ID      Location
0          1      127.16.1.3/1
0          2      127.16.1.3/2
EndList
```

GetListVersion

For each command above calling GetListVersion and the same parameters will return a list version number command only in the format:

ListVersion <Command> <Arguments> <Version>

Example:

```
GetListVersion DanPanel Button
```

Return:

