

Wheatstone Corporation

Technical Documentation

Scheduler Control Of IP-12/16 Air Sources

Using Event Scheduler To Change Fader Source When Fader Is Normally On – Using The Wheatstone IP-12 As An Example – WheatNet-IP System



600 Industrial Drive
New Bern, NC 28562
252.638.7000
www.wheatstone.com

Rev 1.0 – February, 2015
Dick Webb

Table of Contents

1. The Problem.....	2
2. The Example Scenario.....	2
3. Controlling Fader On/Off Remotely.....	2
3.1 Setting Up The SLIOs.....	4
4. Setting Up The Events In Event Scheduler.....	6
4.1 Create The First Event.....	6
4.1.1 A Few Words About The ACI Command String.....	8
4.2 Create The Second Event.....	9
4.3 Create The Third Event.....	10
4.4 Create The Fourth And Fifth Events.....	10
4.5 Create The Events To Switch The Source Back.....	12
5. Expanding On The Principles.....	12
6. Summary.....	12

1. The Problem

It is often necessary to use some form of automation to change the source of a control surface fader in a studio during periods of unattended operation. The Wheatstone Event Scheduler program is one way of achieving such control.

Unfortunately, in radio applications, the surface normally does not allow the source to be changed while the fader is ON, preventing accidental changing of the On Air source. So what's an Engineer to do when asked to provide automated source switching when the surface fader is on and there's no one in the studio to turn it off?

Some control surfaces can be programmed to allow the source change while the fader is on, and some control surfaces have an Automation Control Interface (ACI) that allows the Event Scheduler to directly control fader on/off status. The Wheatstone IP-12/16 series has neither of these features, and so a different method has to be employed.

This paper will explain how to set up such an operation using the WheatNet-IP Event Scheduler to control a Wheatstone IP-12 Control Surface.

2. The Example Scenario

We use a rather simple example scenario to demonstrate the approach. The task is to switch the source on fader 1 of the IP-12 from source 1 to source 2 at 1:40 PM, and change the source back to source 1 at 1:45 PM. The sequence of events requires turning the fader off, switching the source, and turning the fader back on at 1:40 PM, and a similar sequence of events at 1:45.

3. Controlling Fader On/Off Remotely

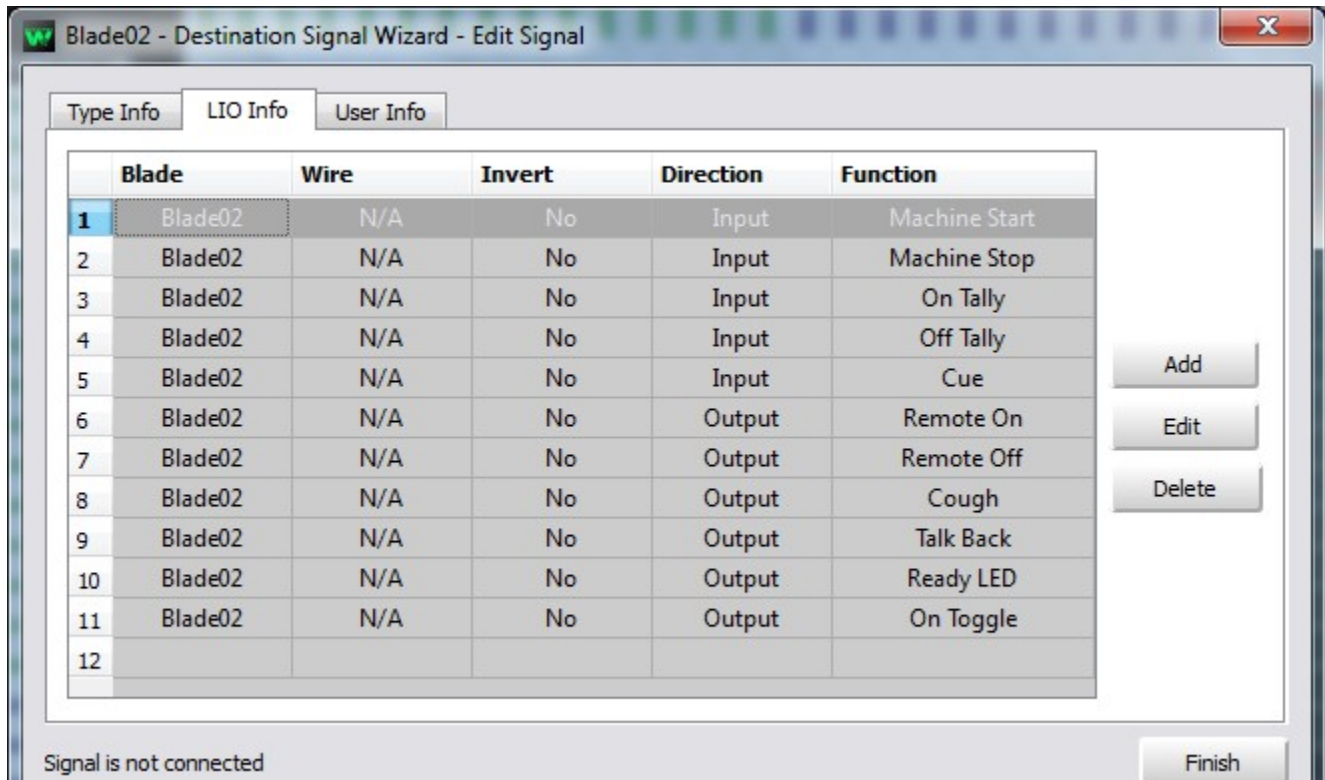
To turn an IP-12 fader on or off via the Scheduler requires the use of virtual logic. The vehicle of virtual logic in the WheatNet-IP system is the Software Logic Input/Output, also known as the SLIO. Each Blade has 64 SLIOs available (128 in the Blade-3 series), so there are plenty to go around.

SLIOs can be “mapped” to source signals, allowing for logic that follows a source. This is very convenient, allowing you to turn the source's fader on and off without needing to know which fader the

source is currently feeding. By using this “associated” logic, on and off control of the fader is accomplished via the source signal. Thus, if we decide to run the sources on a different fader it is not necessary to modify the schedule.

In order to see how a fader can be controlled through logic operating on its audio source it is necessary to have some understanding of the logic that is automatically programmed to each fader on a control surface. Faders have built-in logic for basic function like on, off, cough, talkback, on tally, off tally, etc. This logic can be inspected, but not changed, using Navigator to look at the destination signals representing surface faders.

For example, here is a picture of the built-in fader logic on an E-6 fader destination, as seen on the signal's *LIO Info* tab:

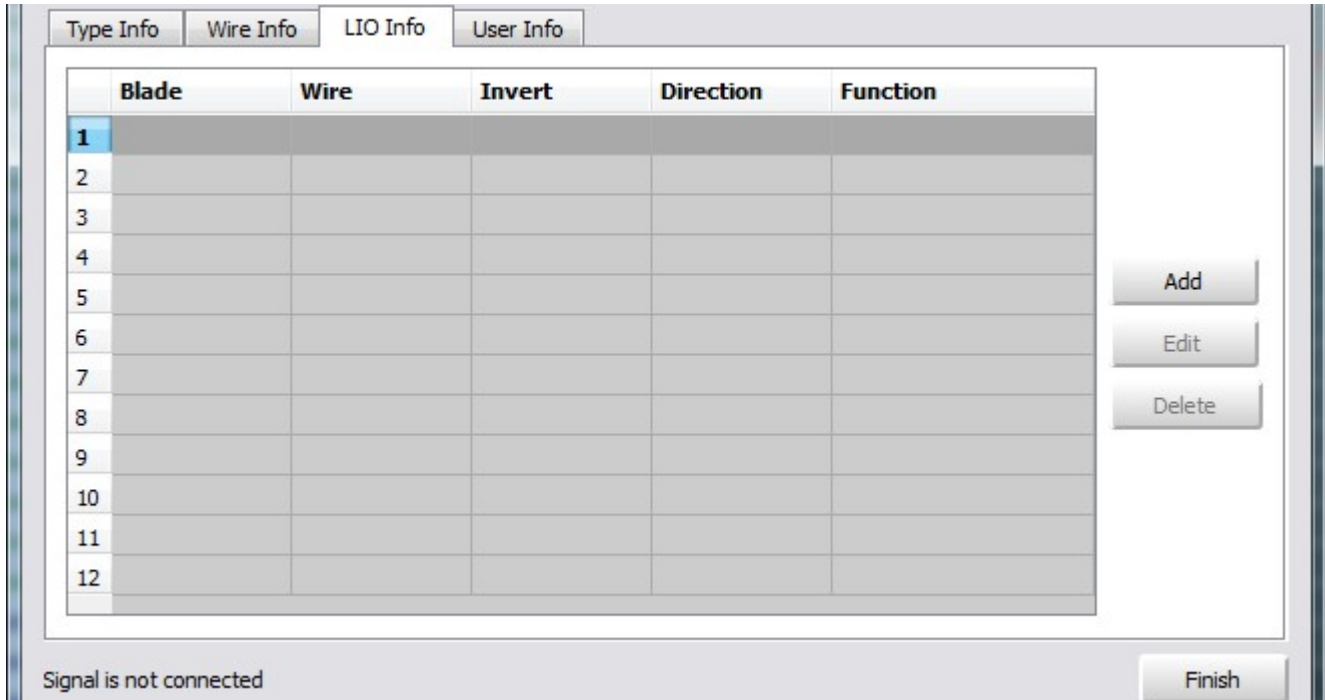


There are two important features of the logic that you need to know when using it: the *Direction* and the *Function*. In order for the logic on an audio source to be able to interface with the built-in fader logic, the *Function* name must match, and the *Direction* must be complimentary. Thus, to turn a fader on by using logic on the audio source, the logic must use the *Function* name **Remote On**, and, since **Remote On** is defined with the *Direction* **Output**, the logic to control it must have a *Direction* of **Input**.

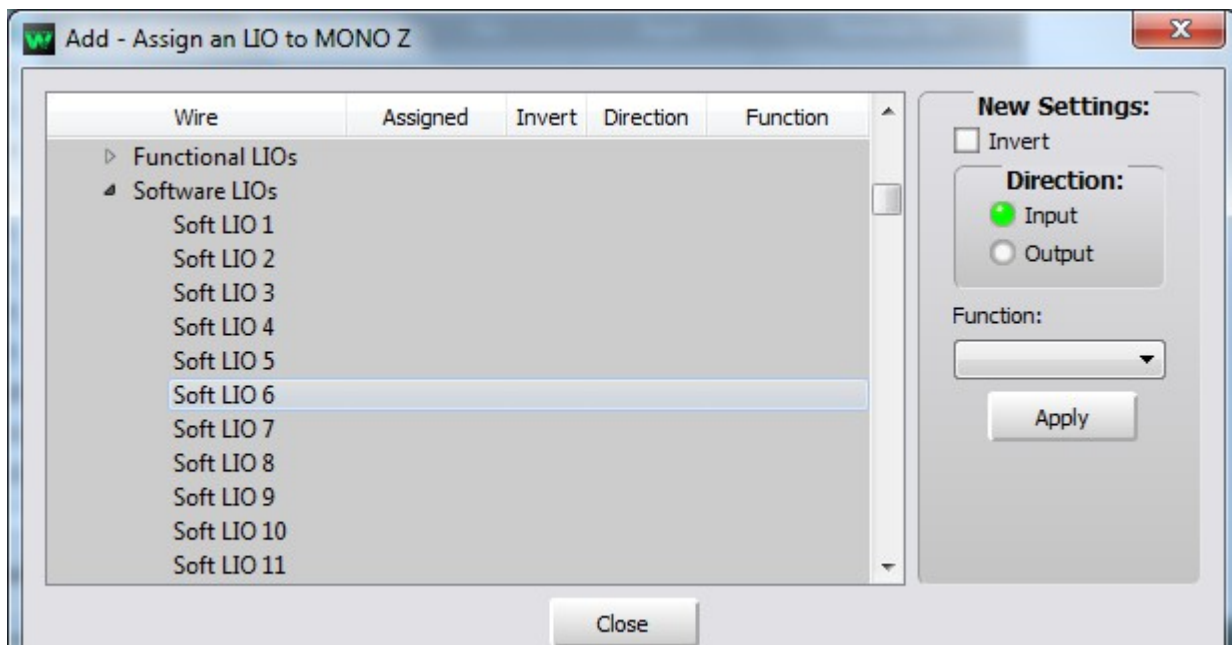
Similarly, to turn a fader off by using logic on the audio source, the logic must use the *Function* name **Remote Off** and a *Direction* of **Input**.

3.1 Setting Up The SLIOs

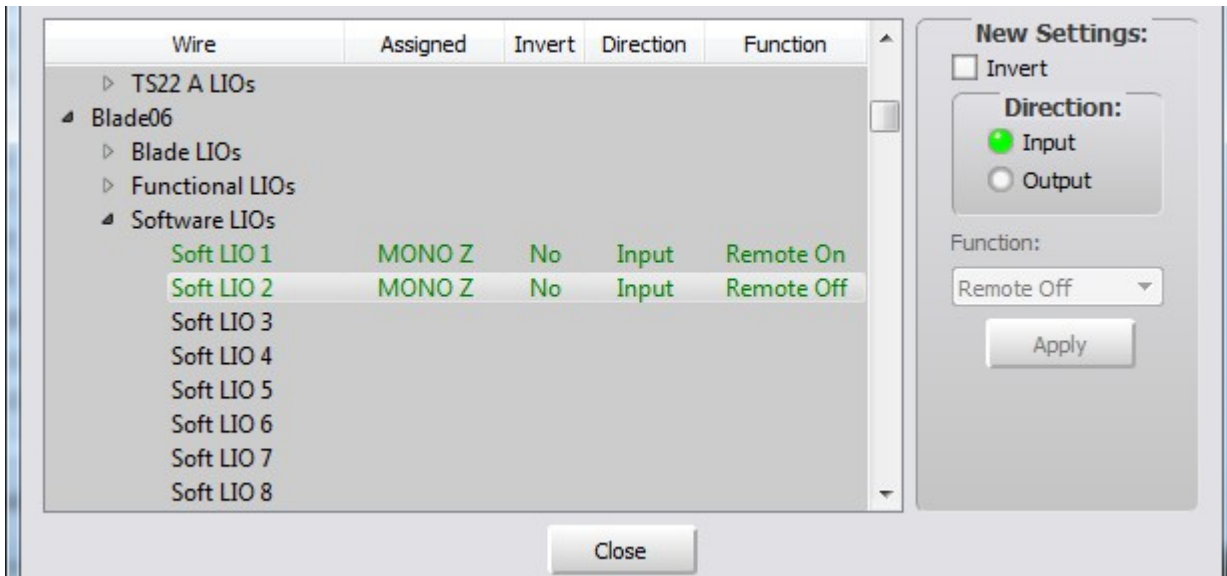
Find the first source that will be assigned to the fader in the Sources area of the signal matrix on the System Crosspoint tab in Navigator. Right click on the name of the source and select *Modify Signal...* from the popup menu to bring up the *Source Signal Wizard – Edit Signal* dialog for that source. Select the *LIO Info* tab.



Click the *Add* button to begin the process of adding logic to the signal.

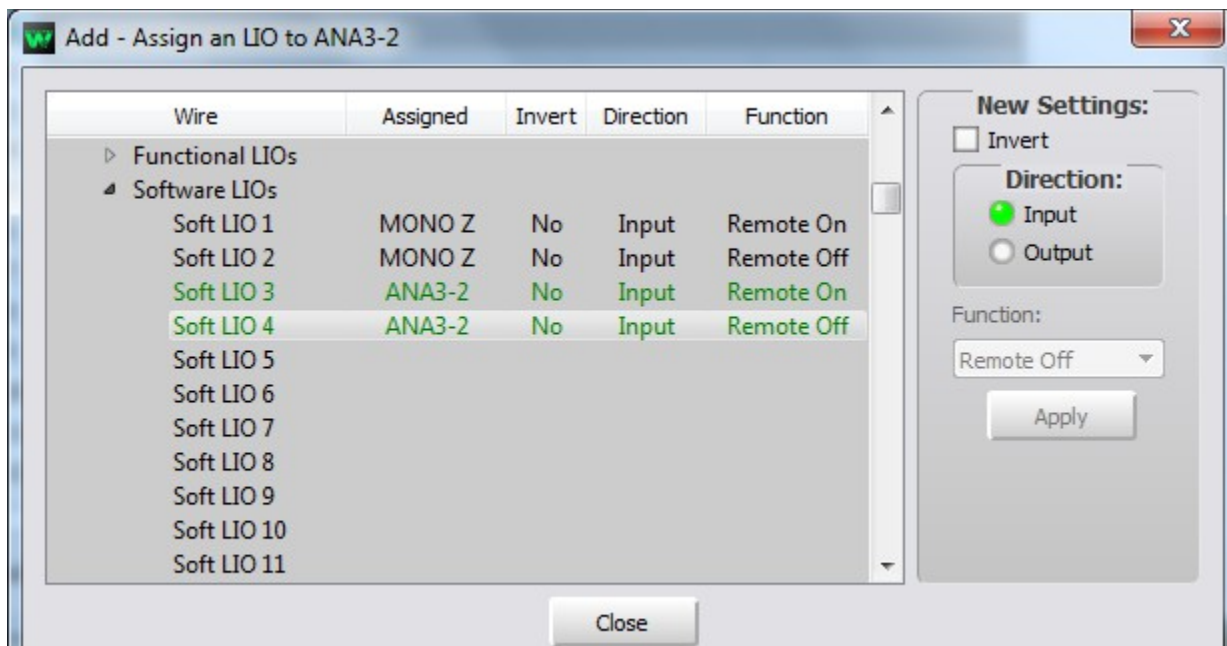


Find the *Software LIOs* list for the Blade you will be programming the SLIOs on. Highlight the first SLIO you will use. Our example uses the first four SLIOs. First, highlight *Soft LIO 1*, find the **Remote On** function in the *Function* drop down list, and select it. The form has selected a *Direction* of **Input** by default. This is the required *Direction*, so all that's left to do with *Soft LIO 1* is to click *Apply*. Since each source will need an SLIO for both on and off, next highlight the *Soft LIO 2* line and set it up for **Input** and **Remote Off**. Here is what the screen should look like after hitting *Apply* to accept these settings for *Soft LIO 2*:



To complete the process for the first signal, click *Close*, then click *Finish*.

Next, go through a similar process to program the logic on the second source. Here is a picture of the *Add – Assign an LIO to [SIGNAL]* screen after the second source has been programmed:



At this point, click *Close*, then click *Finish*, to complete the programming of the second source.

4. Setting Up The Events In Event Scheduler

Next we need to set up the necessary events in the WheatNet-IP Scheduler. For the purpose of this discussion it is assumed you are familiar enough with the Scheduler to know what we events are and how they are used, and that you have already set up Scheduler to talk to a Host Blade so it is able to communicate with the system. If not, a little reading in the WheatNet-IP Scheduler manual is in order.

When setting up the events you may be creating a new schedule, in which case your first event will be number 1. Or you may be adding events to an existing schedule. In our example we will add events to an existing schedule, and our first new event will be number 17.

4.1 Create The First Event

We need a sequence of five events to effect a source change. The first event will work through the current source to generate a logic high on the channel off SLIO for the first source.

In order to have an event control an SLIO we need to know two things – which Blade the SLIO lives on, and what the index number of the SLIO is.

In our example, all of the SLIOs will be on Blade ID 6, located at 192.168.87.106. Before creating the event the Scheduler must be set up to talk to this Blade. Select *File>Define Devices* to open the *ACI Define Properties* dialog.

Current Settings		New Settings - Will Take Effect On Program Restart	
Device Type	IP Address	Select New Device Type	Set New IP Address
surface	192.168.87.202	<input type="radio"/> Blade <input checked="" type="radio"/> Surface <input type="radio"/> Processor	192 . 168 . 87 . 202 <input type="button" value="Delete # 1"/>
blade	192.168.87.106	<input checked="" type="radio"/> Blade <input type="radio"/> Surface <input type="radio"/> Processor	192 . 168 . 87 . 106 <input type="button" value="Delete # 2"/>
blade	. . .	<input checked="" type="radio"/> Blade <input type="radio"/> Surface <input type="radio"/> Processor	. . . <input type="button" value="Delete # 3"/>
blade	. . .	<input checked="" type="radio"/> Blade <input type="radio"/> Surface <input type="radio"/> Processor	. . . <input type="button" value="Delete # 4"/>
blade	. . .	<input checked="" type="radio"/> Blade <input type="radio"/> Surface <input type="radio"/> Processor	. . . <input type="button" value="Delete # 5"/>
blade	. . .	<input checked="" type="radio"/> Blade <input type="radio"/> Surface <input type="radio"/> Processor	. . . <input type="button" value="Delete # 6"/>
blade	. . .	<input checked="" type="radio"/> Blade <input type="radio"/> Surface <input type="radio"/> Processor	. . . <input type="button" value="Delete # 7"/>
blade	. . .	<input checked="" type="radio"/> Blade <input type="radio"/> Surface <input type="radio"/> Processor	. . . <input type="button" value="Delete # 8"/>

Delete # X button sets device X type to Blade and IP address to blank.

Commits Type changes and valid IP address changes without closing form.

Cancels any uncommitted changes without closing form.

Commits Type changes and valid IP address changes, then closes form.

Please note that all currently defined devices, other than the Host Blade through which the Scheduler communicates with the system for Connect, Disconnect, and Salvo events, will be listed on the left. If the Blade that you need to access is not already defined on this dialog, use the right side of the form to set it up. Choose the next available device number and work with the fields on that row. To fire SLIOs we need to talk to a Blade, so select *Blade* as the *Device Type*. Then fill in the *IP Address* of the Blade. Should you choose to use SLIOs on the Host Blade you will need to add that to this form. Once complete, click *Close*, which will commit the device changes and close the form. You will need to restart Scheduler to use this new device.

Now that we are sure the target Blade is accessible to Scheduler's ACI events, we can begin setting up the events.

From the main Scheduler screen, click *New* to bring up the *Event Properties* dialog.

Working your way around the dialog, fill in the required properties. These are:

- *Start Time* – our example uses 1:40:00 PM
- *Repeat Interval* – put a time in here only if you will be firing the exact same event at evenly spaced intervals throughout the day
- *Days* – check at least one *Day* – our example only fires this sequence on **Friday**, so that's the *Day* we have checked
- *I-Time-Only* – check this box if the event is meant to run only this week
- *ID* – this is automatically selected for you – you can dial up a different unused event slot if you have a compelling reason to do so
- *Name* – your event must have a name of at least one character – it is best to make the name be descriptive of what the event will do
- *Event Type* – use the drop down list to select **ACI**
- *Device* – dial up the desired device from the defined devices – should be a blade with the right *IP Address*, and the *OnLine* light should be green
- *Event ACI Command String* – this will show the ACI command string that will be sent when the event is fired
- *Compose String* – this is where you will enter the desired *ACI Command String* – you may type a string or select one from the drop down list – string details are outlined below
- *Save String* – click this button if you want the current *Compose String* entry to be selectable from the drop down list later
- *Enter* – click this button to transfer the current text in the *Compose String* field to the *Event ACI Command String* field
- *Test* – click this button to send the current *Compose String* contents to the selected device to see if it works

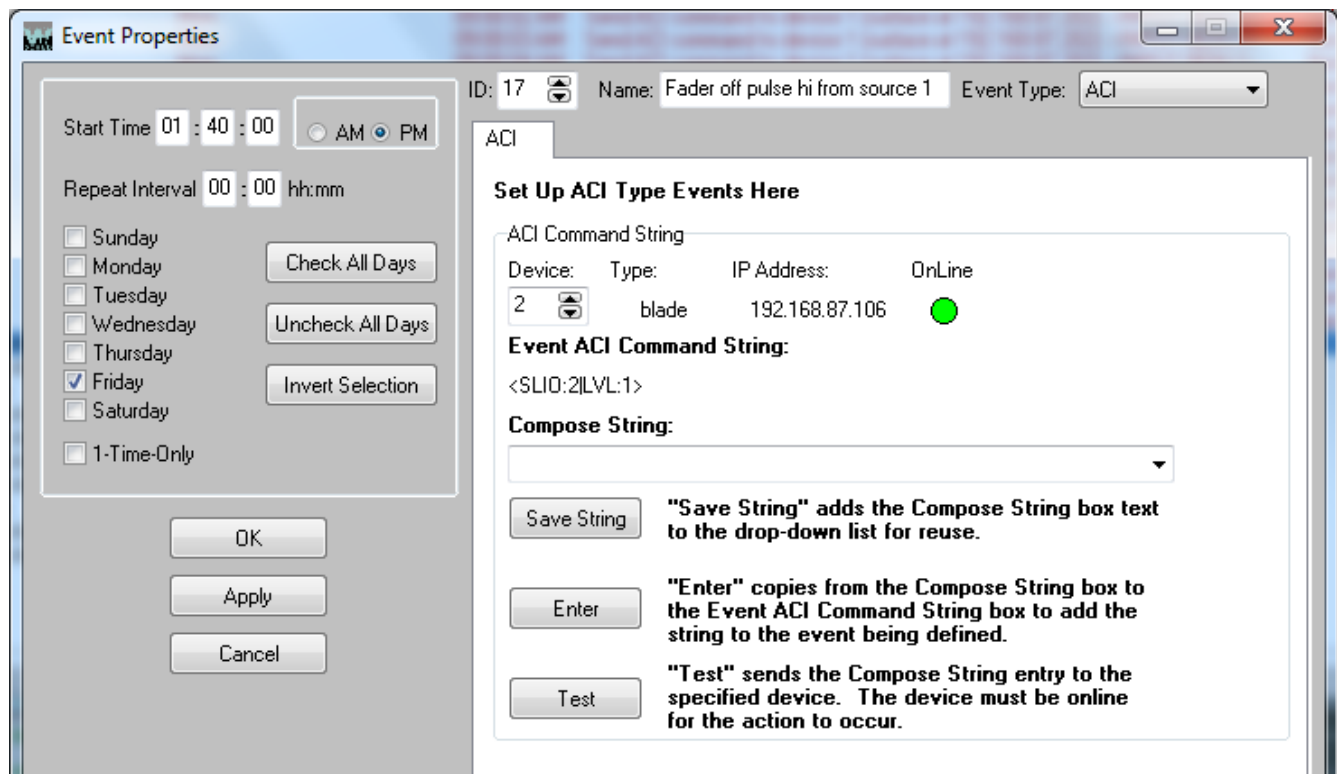
4.1.1 A Few Words About The ACI Command String

The event's ACI Command String must be properly formatted to be understood by the WheatNet-IP system. In this case, we are commanding a particular SLIO to change to a particular state. The selected Device for the command determines where the command is being sent – in the case of an SLIO level command, the device must be the Blade where the SLIO lives. The command itself needs to specify which SLIO is being commanded, and to what level. The possible levels for an SLIO are the numbers **0** and **1**. We can think of level **1** as ON and level **0** as OFF.

The command we need right now is this: **<SLIO:2|LVL:1>**

Here is a brief explanation of the syntax of this command:

- a command always starts with a left-pointing angle bracket - “<”
- all letters in the command must be upper case
- the command target comes next – in this case it is “**SLIO**”
- the name is followed by a colon - “:”
- a parameter associated with the target is next – in this case it is the SLIO number, “**2**”
- the character “|” comes next – it is sometimes called the “pipe” character
- the command action is next – this one is “**LVL**”, short for “level”
- another “:”
- a parameter associated with the action is next – in this case it is “**1**” because that's the value we want to send to the SLIO
- a command always ends with a right-pointing angle bracket - “>”



If at all possible we recommend that you work on ACI command events connected to a live system so that you can test the commands before starting the schedule. But take care, sending the wrong command to the wrong device can cause problems that may be difficult to track down!

4.2 Create The Second Event

Now that we have driven the SLIO high to turn the IP-12 fader off, we must bring that SLIO low to make it ready for the next time it's used. This, by the way, is the reason our event sequence has five events and not three.

Dial to the next free event. It will be set similarly to the first event, with these exceptions:

- the *Start Time* for our example will be 1:40:01 PM – one second after the first command
- the *ID* will be the next one in line
- create a new descriptive *Name* for this event
- the *Event ACI Command String* for this event will be: **<SLIO:2|LVL:0>**

4.3 Create The Third Event

Now that we have turned the channel off we can program the next event in line to be a *Connect* command. This will happen in our example at 1:40:02 PM. Once again, pick a descriptive event *Name*. The event type is *Connect*. Select the desired destination – this will be the fader on the IP-12 – our example uses an E-6 destination of **E6In01**, the first fader on an E-6 surface. Select the desired source – our example is using **ANA3-2**.

ID: 19 Name: source 2 to fader Event Type: Connect

Start Time 01 : 40 : 02 AM PM

Repeat Interval 00 : 00 hh:mm

Sunday Monday Tuesday Wednesday Thursday Friday Saturday
 1-Time-Only

Check All Days Uncheck All Days Invert Selection

OK Apply

Connect

Set Up Connect Type Events Here

Destination (Output Signal)

Name and Location: E6In01 Signal ID: 00810000

Source (Input Signal)

Name and Location: ANA3-2 Signal ID: 00C00002

4.4 Create The Fourth And Fifth Events

Now that we have turned the channel off and changed its source, all we have left to do is turn the fader back on by toggling (changing to 1 and back to 0) the SLIO for channel ON associated with the new source. In our example this is SLIO 3. Here's the event to send SLIO3 high:

ID: 20 Name: Fader on pulse hi from source 2 Event Type: ACI

Start Time 01 : 40 : 03 AM PM

Repeat Interval 00 : 00 hh:mm

Sunday Monday Tuesday Wednesday Thursday Friday Saturday
 1-Time-Only

Check All Days Uncheck All Days Invert Selection

OK Apply Cancel

ACI

Set Up ACI Type Events Here

ACI Command String

Device:	Type:	IP Address:	OnLine
2	blade	192.168.87.106	●

Event ACI Command String:
 <SLIO:3|LVL:1>

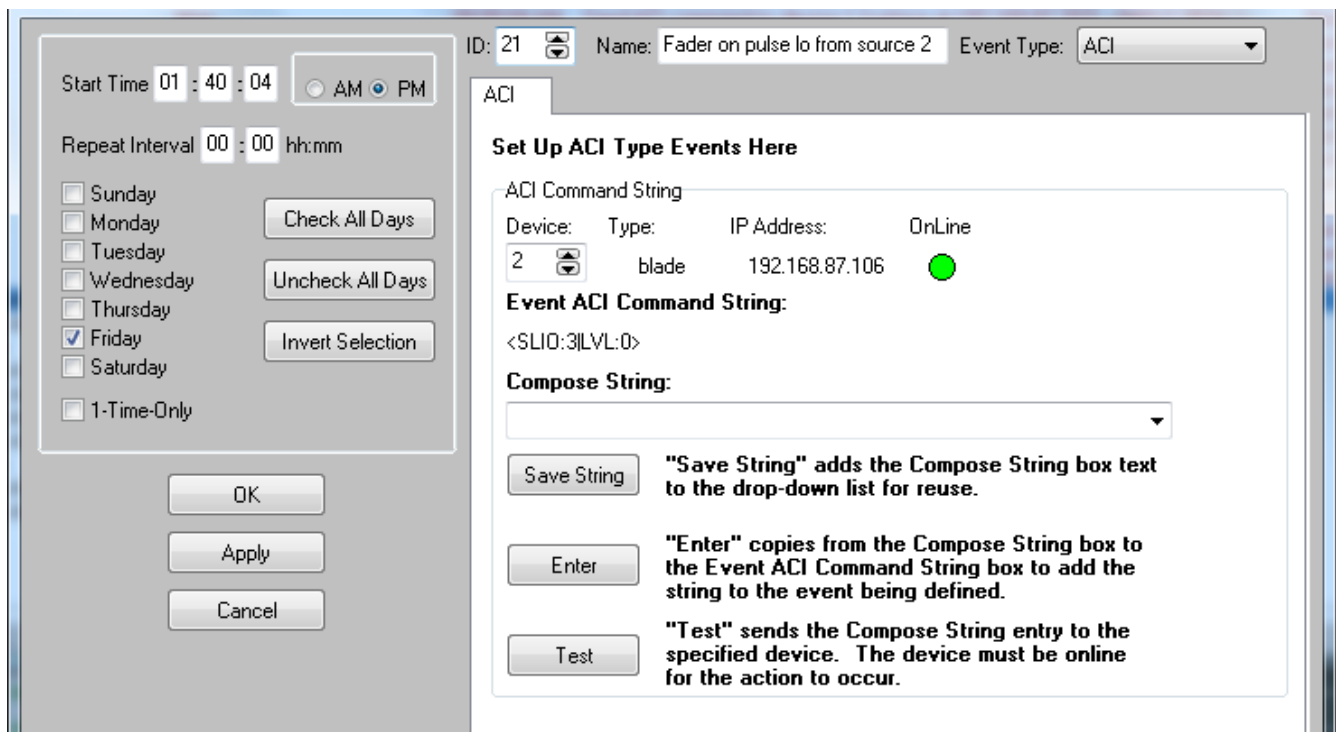
Compose String:

Save String "Save String" adds the Compose String box text to the drop-down list for reuse.

Enter "Enter" copies from the Compose String box to the Event ACI Command String box to add the string to the event being defined.

Test "Test" sends the Compose String entry to the specified device. The device must be online for the action to occur.

And here's the event to take SLIO3 back low:



4.5 Create The Events To Switch The Source Back

By now you should be able to see the pattern and can go ahead and create the sequence of five events that will turn the fader off through toggling SLIO 4 on the second source, switch back to the first source, and turn the fader back on through toggling SLIO 1 on the first source, starting at 1:45:00 PM.

5. Expanding On The Principles

As you can see, it takes a well thought out sequence of events programmed in the Scheduler to accomplish changing an IP-12 fader source without operator assistance when the fader is intended to be on to provide a station's on air audio.

The idea can be expanded to include more than the two sources used in the example. Each source will be required to have two associated SLIO logic functions, one to turn the fader off and another to turn the fader on. Thus you will need to keep close track of which SLIO performs which function for which source. And each source added to the rotation will require a set of five Scheduler events to accomplish the switch.

6. Summary

We have investigated the process of using the WheatNet-IP Event Scheduler to allow on air sources to be changed on an IP-12 Control Surface in the WheatNet-IP system when no surface operator is present to perform the operation. The process involves triggering a fairly complex sequence of events via the Scheduler that will turn a fader off, switch to a new source for the fader, and turn the fader back on.