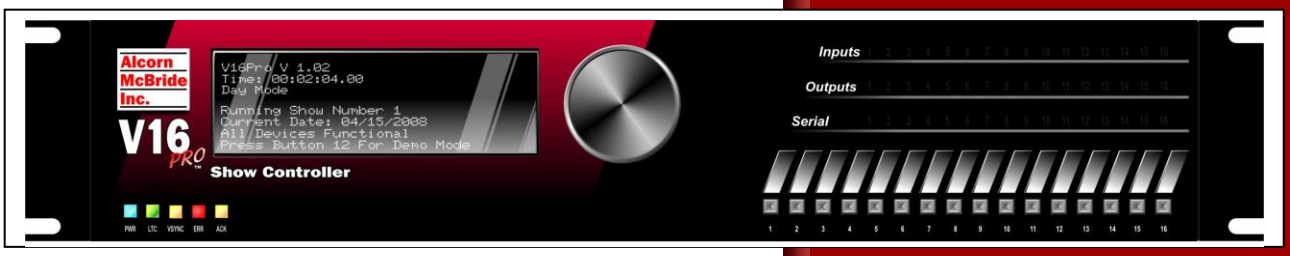
V Series Network Controllers

User's Guide



Alcorn McBride

January 3, 2018

Document Revision 2.5
January 3, 2018

Every effort has been made to assure the accuracy of the information contained in this manual, and the reliability of the Alcorn McBride Show Control hardware and software. Errors can sometimes go undetected, however. If you find one, please bring it to our attention so that we can correct it for others. Alcorn McBride welcomes comments and suggestions on the content and layout of its documentation.

Applications described herein are for illustrative purposes only. Alcorn McBride Inc. assumes no responsibility or liability for the use of these products, and makes no representation or warranty that the use of these products for specific applications will be suitable without further testing or modification. Alcorn McBride products are not intended for use in applications where a malfunction can reasonably be expected to result in personal injury. Customers using or selling Alcorn McBride products for use in such applications do so at their own risk, and agree to fully indemnify Alcorn McBride for any damages resulting from such improper use or sale.
Alcorn McBride Inc. reserves the right to make changes to these products, without notice, in order to improve their design or performance.

Hardware Design:	Jim Carstensen, Scott Harkless, and Joy Burke
Firmware Design:	Joy Burke, Scott Harkless and Adam Rosenberg
Software Design:	Steve Alcorn, Joy Burke, Devin Acker, and Adam Rosenberg
Documentation:	John Conley, Joy Burke, Adam Rosenberg, Kevin Lang, Diego Reano, Jim Carstensen and Steve Alcorn
Mechanical Design:	Martin Chaney

Alcorn McBride Inc.

Alcorn McBride Inc.
3300 S. Hiawassee, Bldg. 105
Orlando, Florida 32835
TEL: (407) 296-5800
FAX: (407) 296-5801
http://www.alcorn.com
info@alcorn.com

Table of Contents

Welcome

The V16Pro, V4Pro and VCore show controllers are the latest versions of our original, most powerful, show controllers. They are ideally suited for the control of theme park attractions, museum displays, retail spaces, casino installations, games shows, or almost any automated venue. They provide more serial ports than any of our other controllers, and offers almost unlimited memory capacity. Other important features include self-healing outputs, software configurable inputs, dual Ethernet ports, MIDI, USB, video sync, and a built-in SMPTE generator and reader. In short, they have just about every control interface you'd ever need, all in one easy to use box.

Product Comparison Chart

Product	V-Core	V4Pro	V16Pro
Recommended # Devices	10	50	50
Time-based Sequences	YES	YES	YES
Serial Ports	1	4	16
Inputs	4	16	16
Buttons	-	16	16
Outputs	4	16	16
Ethernet Ports	1	2	2
Redundant Mode	YES	YES	YES
Remote IO	YES	YES	YES
Graphical Timeline	YES	YES	YES
Display	-	42x8 VFD	42x8 VFD
MIDI	-	YES	YES
V-Sync	-	YES	YES
SMPTE Timecode	-	-	YES
ARTNet	-	-	-
POE	YES	-	-
Purpose	Small systems / sub-systems / kiosks	Medium systems	Large and high-profile attractions

Important Information

Congratulations! You have purchased an extremely fine product that would give you thousands of years of trouble-free service, except that you undoubtedly will destroy it via some typical bonehead consumer maneuver. This is why we ask you to:

Please for God's sake read this manual carefully before you unpack the product.

You already unpacked it, didn't you? You unpacked it and plugged it in and turned it on and randomly punched the buttons, and now your tech, the same tech who only has a fleeting understanding of the difference between 24VDC and 240VAC, this tech is also punching the buttons with his screw driver even as you read this, right? We might as well just break these products right at the factory before we ship them out, you know that?!?

We're sorry. We just get a little crazy sometimes because we're always getting back "defective" merchandise where it turns out that the consumer inadvertently bathed the product in battery acid for six days. So, in writing these instructions, we naturally tend to assume that your skull is filled with dead insects, but we mean nothing by it. OK?

But we digress...

Thank you for purchasing this wonderful product. It will serve you for its entire lifetime, however long that may be. Since no one ever reads this section of the manual, we're going to take this opportunity to get a few things off our collective chests and out of our collective drawers, as it were.

As always, we welcome calls to our technical support department. Unlike many companies, our technical support personnel all speak at least one language. That's in addition to any talking they do to themselves. And they're not located in some far off backwater, but right near our engineering staff. In fact, according to their probation officers, they'll be here quite a while. This is more than I can say for our tools, which have been disappearing at an alarming rate.

Speaking of engineers, did you know that you're always welcome to speak directly to the engineer who designed your product? It's where they get most of their ideas, because Lord knows they don't come up with much on their own.

Of course, the people you really want to speak to are our sales department, because once you experience the orgasmic joy of owning this product, you're going to want lots, lots more of them. And there's no group better at dispensing orgasmic joy than our sales staff. And if you believe that, you've obviously never met them.

While I'm on the subject of that plastic sheeting your product came wrapped in (and potential uses for same), here's a friendly reminder to dispose of all packing materials in an environmentally friendly (and hygienic) manner. Also, please inspect all packaging carefully before discarding it, as we're still looking for Quality Assurance Manager Shirley Peltwater's prosthetic toe.

Finally, we'd like to once again thank you for purchasing this spectacular product. You have no idea how much we depend on our design challenges to reduce the amount of time we spend surfing the web for pictures of… oh my God, is that our sales staff?!

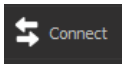
Quick Start Guide

Download WinScriptLive (www.alcorn.com/support/software) and get online with controller examples

1. Open Example Script

Open WinScriptLive and click on "New" and select one of the "Starter Scripts" of interest. Click "Open".

2. Connect

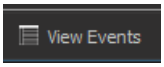
- Connect an ethernet cable from your controller to network or PC.
- Click 

to start connection. (Save Script when prompted)
- Find and click on your show controller in the "Find Your Show Controller" window.*

- When prompted, "Send the Script" to the controller and wait for a restart

** For VCore, if your controller is unable to be reached, you can assign an IP address here or switch DIP switch 1 to "ON" to use DHCP (Automatic IP assignment). For more, see the "Setting VCore IP Address."*

3. Watch and Run!

Click View Events 

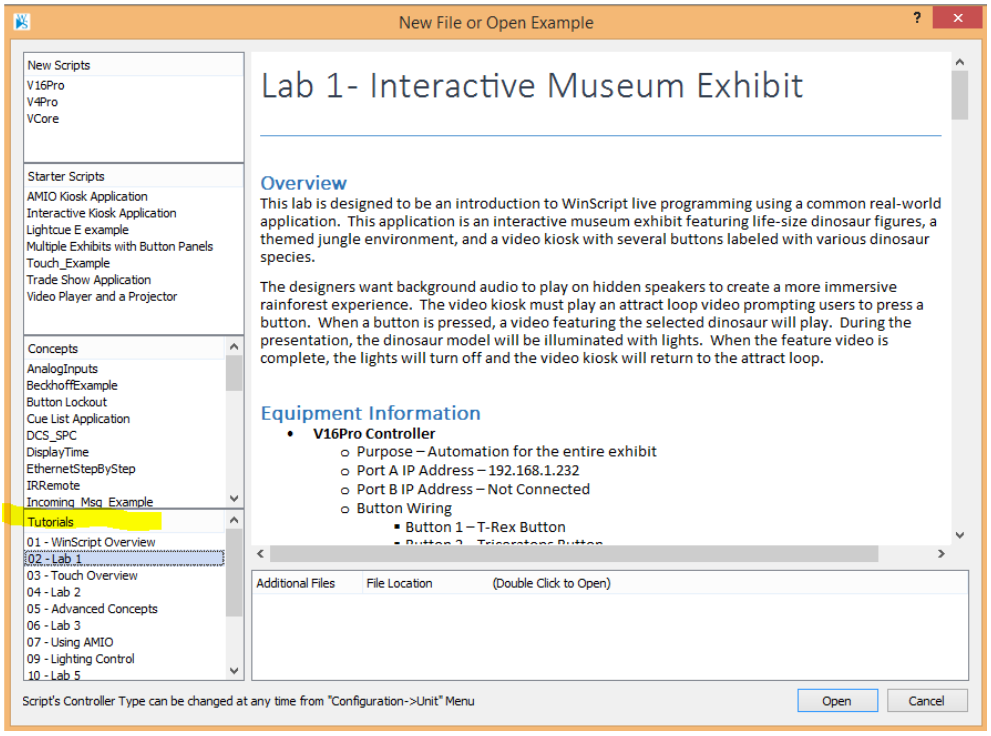
view or edit contents of each sequence.

Click the status buttons to start or stop sequences 

WinScriptLive Tutorial

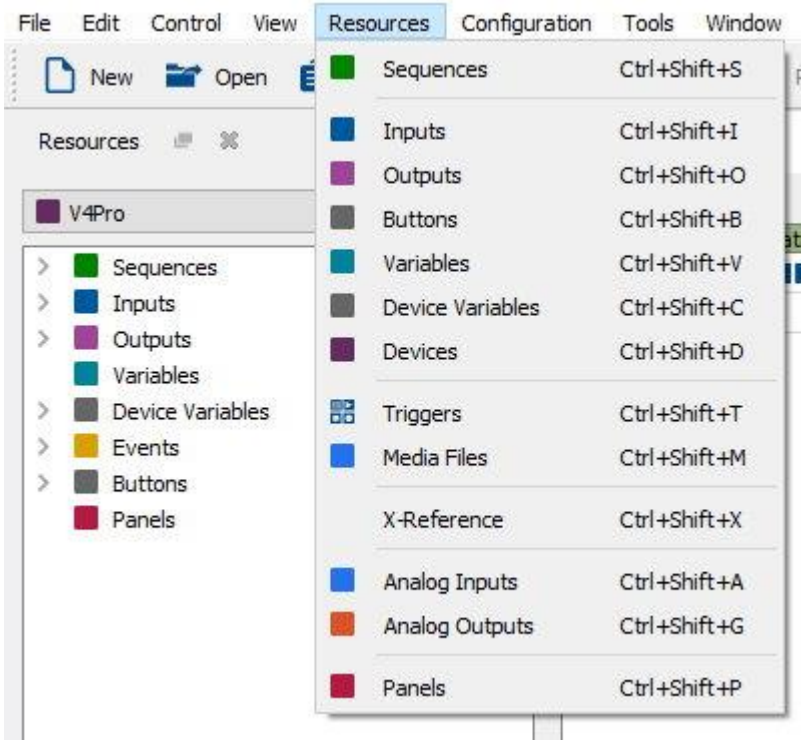
To view the latest WinScriptLive Tutorials and Examples, launch WinScriptLive and click the “Examples” button.

Tutorial “Labs” and corresponding documentation can be found in the screen that appears.

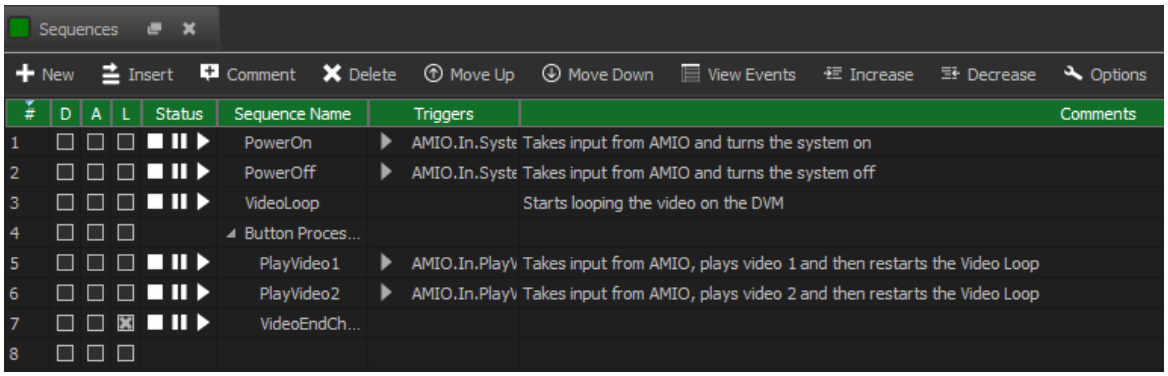


WinScript Live Resources

The actions performed by the show controller are created by using the show controller's "resources." These resources can be accessed using the side toolbar, or from the "Resources" menu.



Sequences



Scripts are made up of sequences that are groups of events.

Sequence Columns

Notice the columns labeled D, A, L and Status just before the Sequence Name column. Clicking on the box places a check mark in the column for that sequence.

D is disable, the sequence will not be checked for errors or run. This allows you to remove that line from the show without deleting it from your script.

A is autostart and will run the sequence when the script is started.

L is for looping the sequence.

Status/Control is a real time event indication of what is running in the show controller when in "Live" mode. Also you can control the sequence as you wish outside the normal script flow.

Sequence is stopped if highlighted, will stop it the sequence if it is running

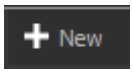
Sequence is paused if highlighted, will pause the sequence if it is running.

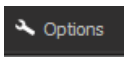
Sequence is running when highlighted, will start the sequence if it is stopped.

Each sequence will need a name so that it can be call if needed and to identify its function.
The sequence may require a trigger to start and is defined by the trigger column.

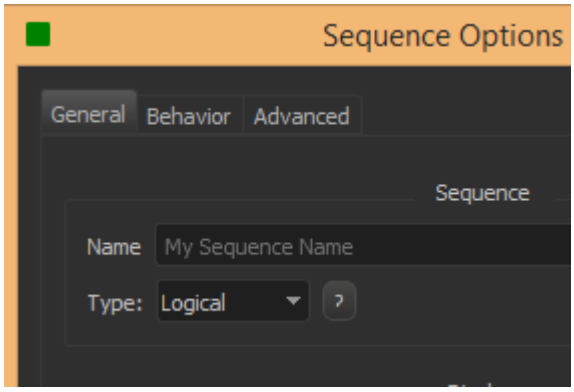
Sequence Clock

Each sequence runs according to its own "Sequence Clock." This clock keeps track of the current frame for that sequence. This sequence clock can be generated using the show **controller's internal timecode**, using the show controller's **SMPTE/EBU generated timecode**, or using an **external SMPTE/EBU timecode** source.

Creating a New Sequence 

or clicking on the 'Sequence Options' button 

allows configuration of the sequence clock.



Selecting this option will bring up a dialog where the options can be changed. These options can also be changed from the "Events" screen (see next section).

- **Logical** In this mode, the events in a sequence are executed as fast as possible when the sequence is started. If the sequence is looping, the events may execute more than once within a single frame.

- **Timed** This uses the show controller's internal timecode. This internal timecode can be synced to an external Blackburst/C-Sync source.

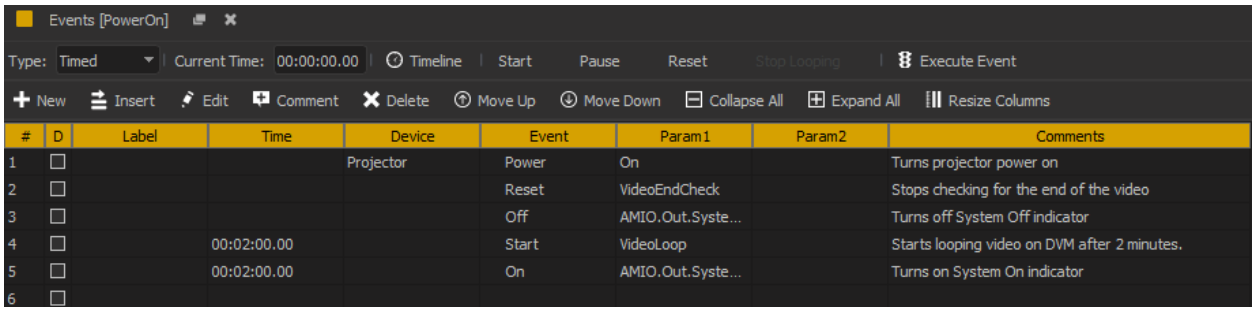
- **SMPTE/EBU** This timecode chasing mode causes the sequence to adjust its location (or scrub) in the event that the timecode skips backwards or forwards.

Events

Each sequence is made up of lines called "Events". An event is a single step that is taken to perform the function of the sequence. Events interact with all the show control hardware and devices.

Event Grid View

A sequence can be viewed as either a Timeline or a Grid view. The Grid View is shown below.



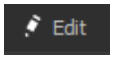
Event Columns:

D is for Disable. Checking this box will cause the system to skip over the event and not execute it.

Label When working with events, sometimes it is necessary to skip over an event or even groups of events. This is accomplished by using the "Goto" event that requires operator to give the event a place to go. So you might want to "Goto There" where the label is "There"

The **Time** column gives the user the option to delay the start time of events to a specific time. It is important to remember that putting a value in the Time column of an event will cause the series of events that follow it to wait until the event is completed. All events that follow will be delayed even if the time fields for those events are less that the time given in the previous event. All events that have start times earlier than the previously executed event will be run sequentially as fast as possible after the executed event.

The **Device** column identifies the device to be controlled by the event. A list is given for each event under the Device column. A drop-down menu of available devices will open when the operator double-clicks in the event line under the device column. Refer to the Devices section for adding additional devices. Events are what actions the show controller can perform. Event commands in the drop down list are only listed for the device selected in the previous column.

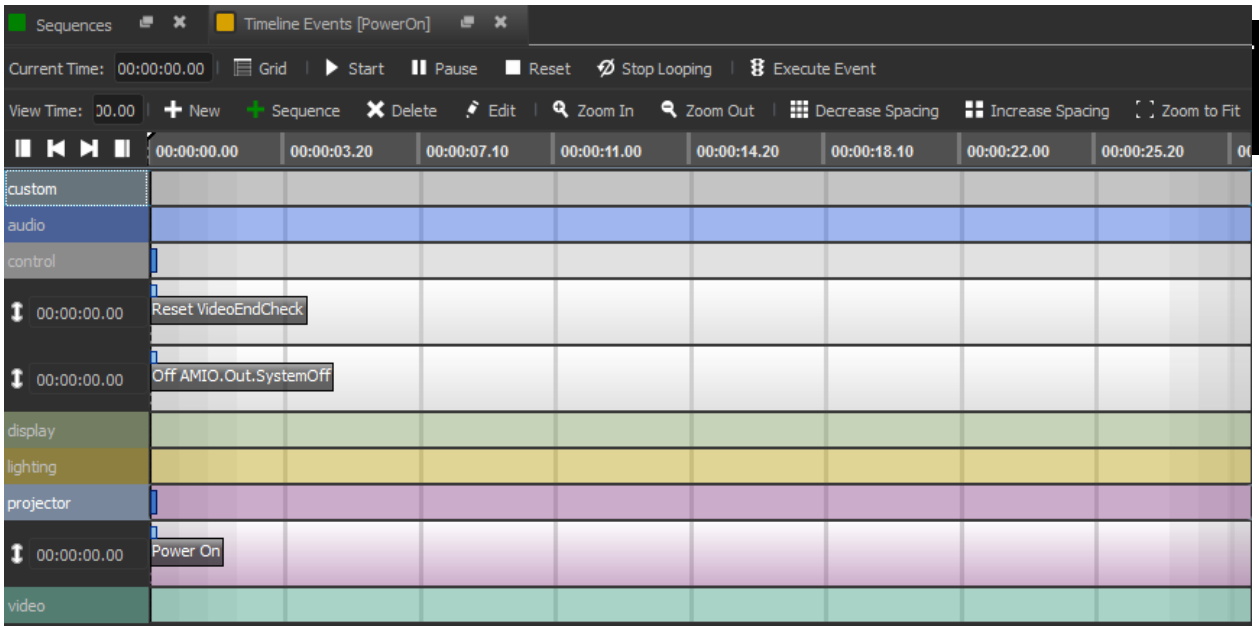
The **Edit** 

option will walk you through the selections available and the required data fields required for the proper operation of the event.

Data1,2,3,4 fields are to hold the event options as an example the "Goto" event requires a label to go to, this label is placed in "Data1" which will be "There".

Event Timeline View

A sequence can be viewed as either a Timeline or a Grid view. The Timeline View is shown below.

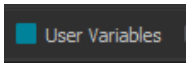


Timeline allows you to view and edit events in a more intuitive manner. Events can be dragged from the resources list on the left, and dropped on top of the timeline at the precise time the event needs to happen.

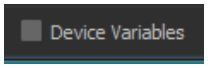
All the editing for a respective event can be done on the **Properties** window on the right side of the screen. For a more detailed description on Timeline and all of its features, please refer to the Timeline section of this manual.

Variables

Two types of variables are available for use in WinScriptLive.



User variables are created by the script writer for a custom purpose.



Device variables are created automatically after adding a device.

User Variables

Selecting a **specific type** of variable defines how it can be used in events, and how it is displayed.

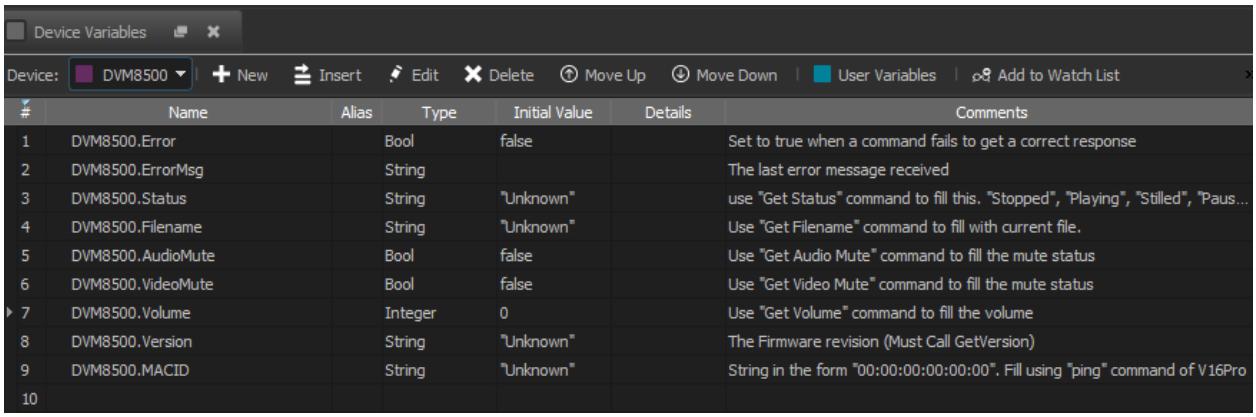
Variable Type	Description
Boolean	possible states: On/Off, True/False, 1/0. In previous versions of WinScript, these were referred to as flags.
Integer	An integer in the range of –2,147,483,647 to 2,147,483,647.
Percent	Decimal entry using a % sign. 0-100%.
Timecode	SMPTE/EBU timecode style of 00:00:00.00

Display String	String formatted for use on the VFD display. Lines of the display are separated by commas. To clear a line, use "clr" outside of the quotes. Example: "Hello","world",clr,"line4",, Would print: Hello World Line4
Date/Time	The month/day/year followed by time in military style : ie: 10/15/09 13:45.
Decimal	A decimal number accurate up to four decimal places with the same possible range as Integer.

Device Variables

These variables are created automatically after adding a new device. They are usually read-only, but in some cases they can be set to an initial value in the "device wizard" during device setup.

After clicking the "Device Variables" button in the "Variables" screen, a list of the show controller's device variables will automatically appear. This list will change based on the family member selected.



#	Name	Alias	Type	Initial Value	Details	Comments
1	DVM8500.Error		Bool	false		Set to true when a command fails to get a correct response
2	DVM8500.ErrorMsg		String			The last error message received
3	DVM8500.Status		String	"Unknown"		use "Get Status" command to fill this. "Stopped", "Playing", "Stilled", "Paus...
4	DVM8500.Filename		String	"Unknown"		Use "Get Filename" command to fill with current file.
5	DVM8500.AudioMute		Bool	false		Use "Get Audio Mute" command to fill the mute status
6	DVM8500.VideoMute		Bool	false		Use "Get Video Mute" command to fill the mute status
7	DVM8500.Volume		Integer	0		Use "Get Volume" command to fill the volume
8	DVM8500.Version		String	"Unknown"		The Firmware revision (Must Call GetVersion)
9	DVM8500.MACID		String	"Unknown"		String in the form "00:00:00:00:00:00". Fill using "ping" command of V16Pro
10						

Any of these variables can be referenced in the 'Events' by using the device name followed by a period '.'. For example, to access the automatically created "V16Pro" device variables of "LTC", type "V16Pro.LTC". Device variables may have different family members.

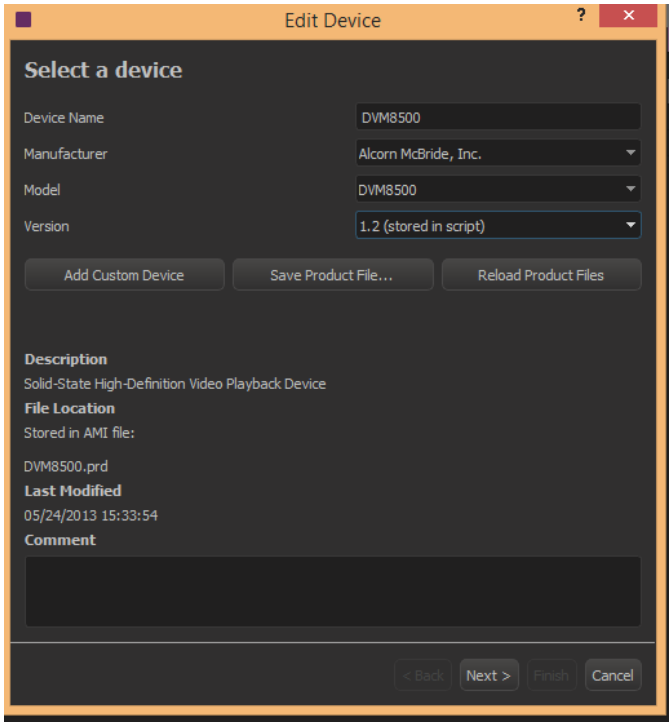
Other device's variables can be viewed from this screen by selecting the device's name in the drop down list on the toolbar.

Devices

All the machines needed to complete your show are called devices. The connections to the show controller are through the serial ports or by the Ethernet network. Adding devices to your show will add "Device Variables" and additional possible "Events" to your show.

Clicking on the 

button in the devices tab will bring up a wizard that will guide you through setting up communication from the show controller to your device. Alternatively, you can right click on a blank row and click “Edit”.



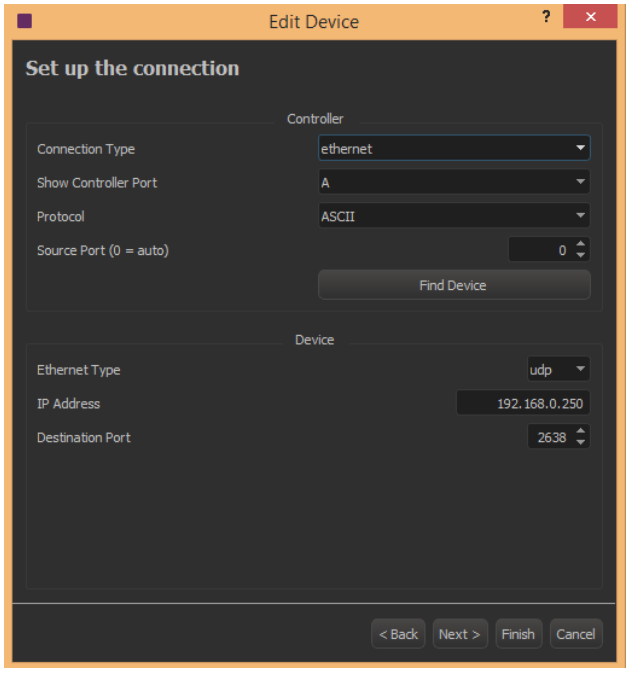
This form will configure the show controller for the kind of device to be connected such as DVM, DMX, Audio or any other kind of equipment needed. The information about the device selected is displayed and where the product file is located.

Note: Product files are often shown with the Resulting File description of "**Stored .ami file.**" This occurs after a file has been saved with a particular device configured. Every .ami file contains all of the product files necessary for the script to run and be edited. If you wish to specifically refresh the product file to a later version after a product file has been saved to the .ami file, you can do so from this screen.

The next step is to configure the hardware communications link.

If serial is selected, the user will be prompted for the port number, protocol format, baud rate and other serial control information. All the serial ports may be configured for RS232 or RS422.

If Ethernet is selected, the user will be prompted for the network port A or B, protocol format, IP address, Ethernet Type, and Ethernet port numbers for the device and show controller.



Ethernet Types:

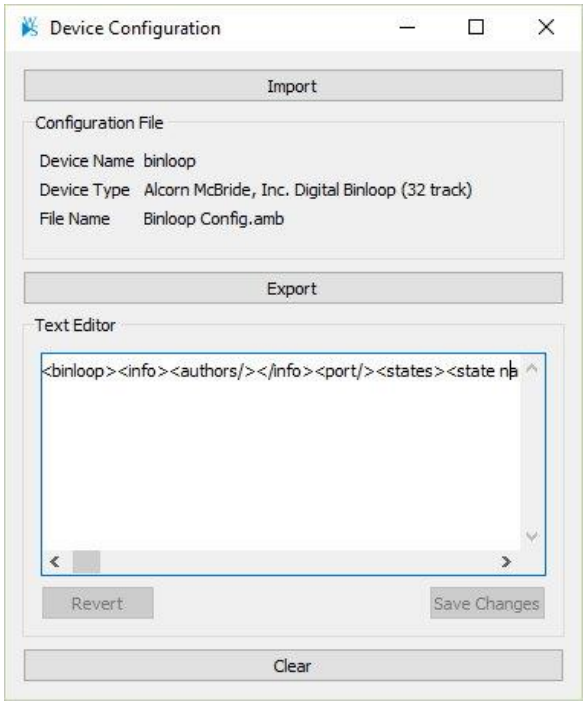
- **"UDP"** – a protocol with no "handshaking." The Show Controller's port number will be used to receive data. Basically, the show controller will "listen" to messages coming to the Show Controller's port from the specified IP address. The device's port number is where the show controller will attempt to send the any command messages.

- **"TCP/Telnet" or "TCP_Client"** – This protocol requires a "connection" between the two specified ports. The show controller will initiate the connection to the specified Device's IP address and Device's port. If the device does not respond, the controller will attempt to make a new connection whenever a an event involving that device is executed.

- **"TCP_Server"** – This protocol requires a "connection" between the two specified ports. The show controller will "listen" for connections and messages on the specified Show Controller port. If the controller uses a "message out" event, data will be sent to all devices that have made a connection to that port.

Device Configuration File

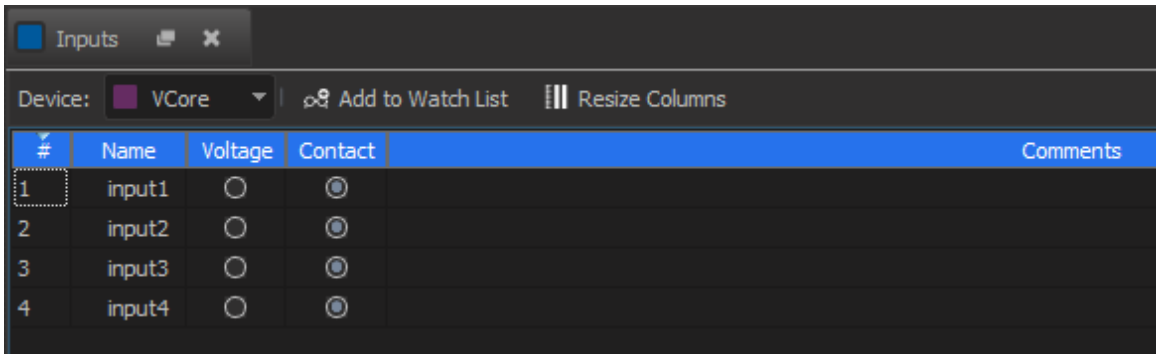
- To help keep your data organized, it is possible to save a configuration file with the device. The file itself will be stored within WinScript and can be saved at any time should it be needed to reconfigure a device or to configure a replacement device.
- Import – bring a configuration file into your WinScript document
- Export – save a stored configuration file to your computer
- Clear – remove a stored configuration file



- If the configuration file is a text document. You can edit the file and Save Changes directly from this dialog

Inputs

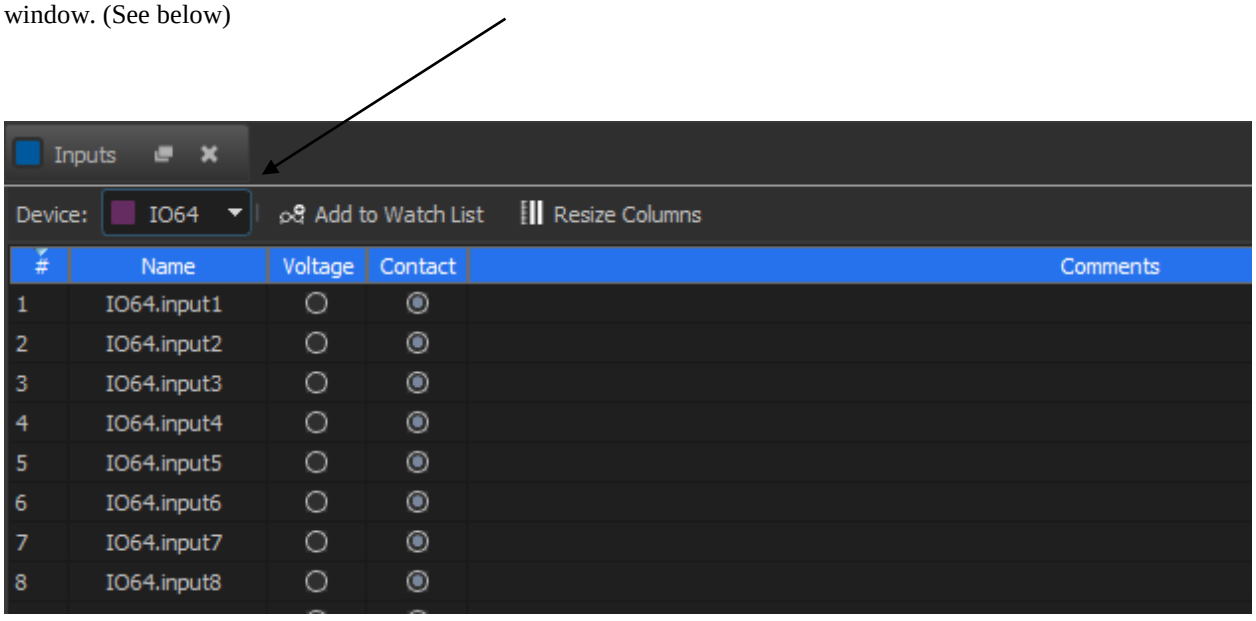
Rear inputs can be configured in WinScript Live as either Voltage or Contact Closure. They can also have their names changed to more easily reference them within the script.



IO64 Slave Inputs

The Alcorn McBride IO64 can be setup as a "Slave IO" protocol. In this mode, inputs from the IO64 are placed directly into the "Inputs" window. Once in this window, these inputs can be referenced just like other internal show controller inputs. The only difference is a 1-frame delay from the time the input status is received at the IO64 to the time it is updated in the show controller.

To view a specific device's input, select the device from the drop down list at the top of the "inputs" window. (See below)



If this list does not appear, make sure that "Slave IO" is listed as the "protocol" for the device in the "Devices" screen. If it is not, edit the device and select "Slave IO" as the protocol.

This feature can be used with the IO64, V16+, V4+, V2+ and the Interactivator. However, other types of IO can be integrated in this fashion by creating the appropriate protocol file.

Note: In order to get the "on" or "off" status of the Slave IO, an .amw script file must be sent to show controller using WinScript (Standard, not Live). The corresponding .amw script files can be found under the "Scripts" directory of the C:\Program Files\Alcorn McBride Inc\WinScriptLive\Scripts.

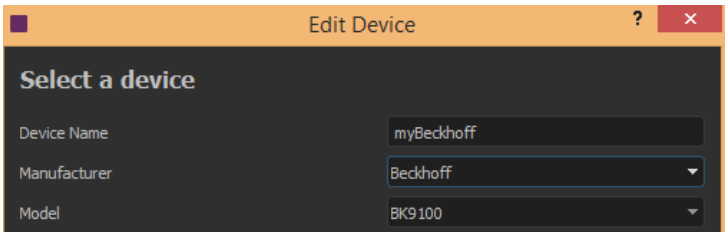
Modbus TCP Slave Inputs

Modbus TCP is a standard protocol used for many IO device modules. Any IO device capable of using ModbusTCP can have its IO controlled as if it was IO internal to the V16/V4Pro. So far, Beckhoff IO and Avantech Adam-6000 series IO have protocol files available. Please contact support if you have a ModbusTCP IO device you would like to control.

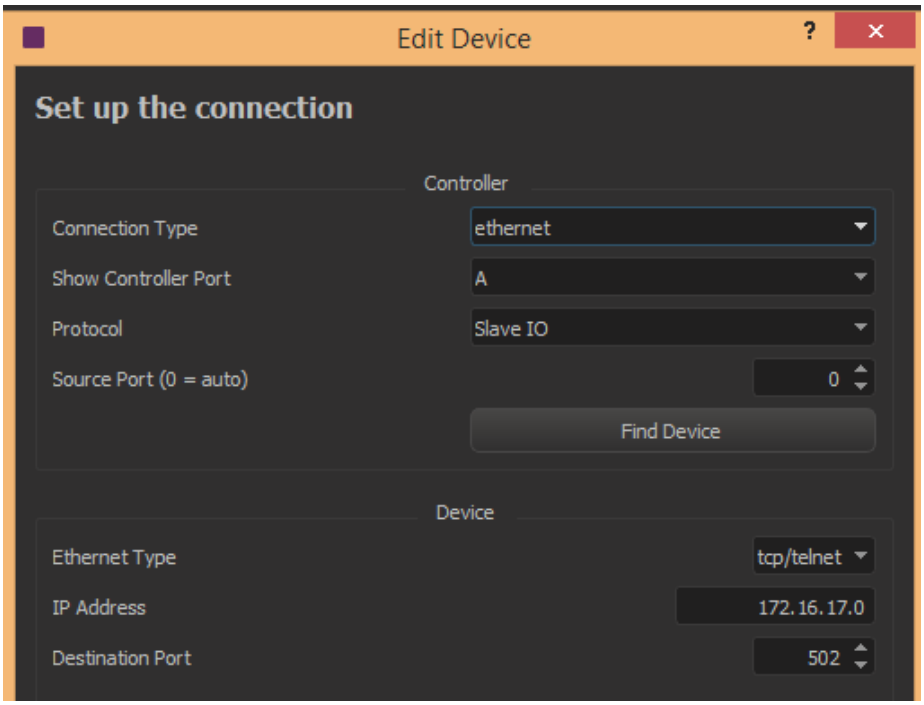
The following example will demonstrate how to setup a Beckhoff BK9100 for control by the V16/V4Pro.

1. Add the IO module to the "Devices" list by clicking the "Add" button in the "Devices" window.

2. Enter a name such as "MyBeckhoff"
3. Select the Make, Model and version of the IO



4. Enter the IP Address information, the V16/V4Pro Ethernet port you would like to use, and make sure the protocol is set to "Slave IO"

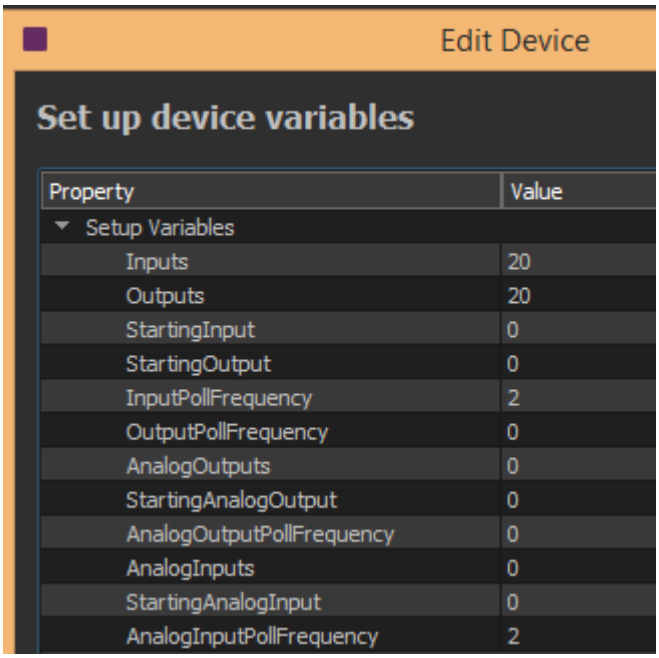


5. Enter the number of "**Inputs**" and "**Outputs**" you will be controlling into the boxes.

The "**StartingInput**" and "**StartingOutput**" are typically 0, but may be a greater number if you're only controlling a sub-section of the IO on a particular control module. For example, if you only wanted to watch inputs 3-12 on a module that had 0-12 available.

The "**InputPollFrequency"** can be set to as little as 1 frame. If you have greater than nine devices that are setup to poll inputs, you must decrease the polling to 2 frames or more.

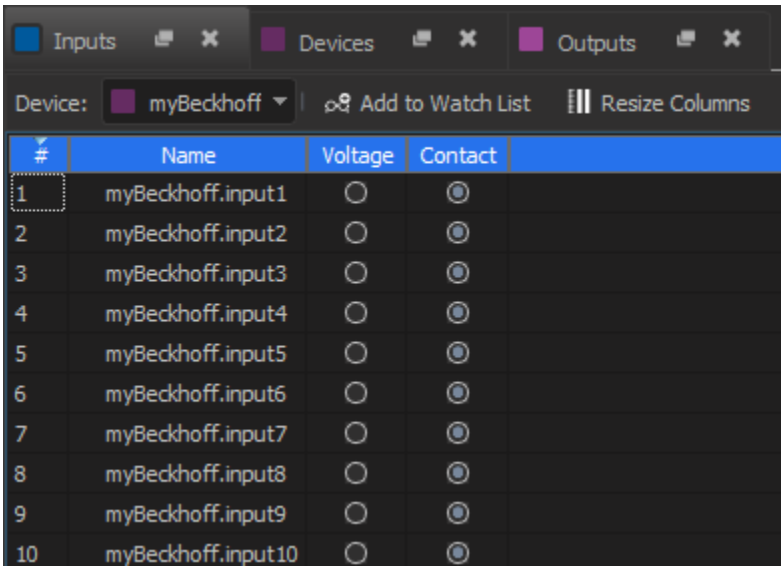
The "**OutputPollFrequency**" is typically set to zero. This does not mean that the outputs will never be polled. They will be polled on startup and after any command is send to change the output's status. (ie: after an "On", "Off" or "Toggle" command). If you prefer more constant polling, a recommended value would be 15 frames.



6. Click "Finish"

Your IO will now show up in the "Inputs" and "Outputs" lists. To View your Beckhoff IO inputs:

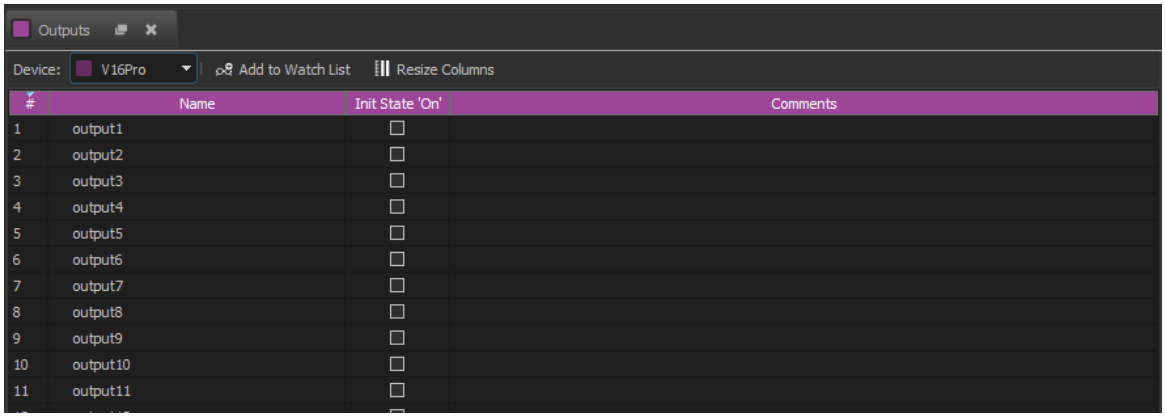
1. Go to "Resources"→"Inputs" to view the inputs list.
2. Select the name of your device (in our example: "MyBeckhoff") from the toolbar



You may now use your inputs as triggers, or directly in internal "events" such as "If On". See the "Triggers" section for more information about triggers.

Outputs

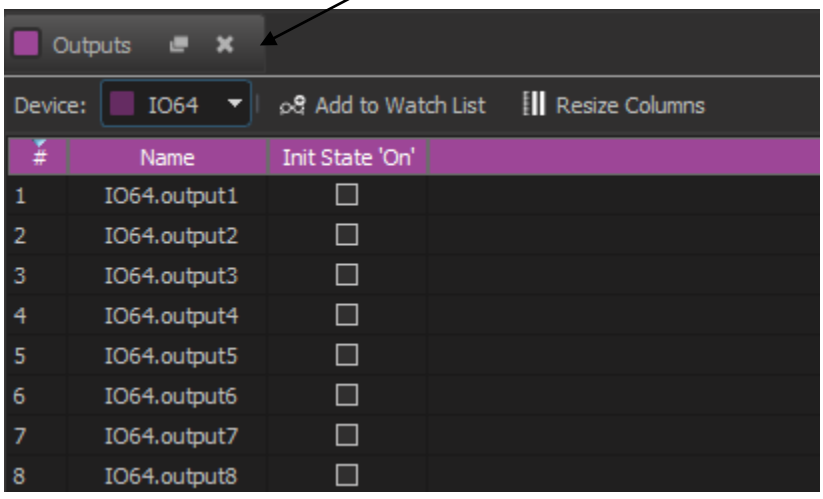
Outputs can be renamed to more easily reference them in the Event's "Data" columns during programming. From the output screen, you can also set the initial state of the outputs after a show controller has loaded the show.



IO64 Slave Outputs

The Alcorn McBride IO64 can be setup as a "Slave IO" protocol. In this mode, inputs from the IO64 are placed directly into the "Outputs" window. Once in this window, these outputs can be referenced just like other show controller outputs.

To view a specific device's output, select the device from the drop down list at the top of the "outputs" window. (See below)



If this list does not appear, make sure that "Slave IO" is listed as the "protocol" for the device in the "Devices" screen. If it is not, edit the device and select "Slave IO" as the protocol.

This feature can be used with the IO64, V16+, V4+, V2+ and the Interactivator. However, other types of IO can be integrated in this fashion by creating the appropriate protocol file.

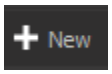
Note: In order to get the "on" or "off" status of the Slave IO, an .amw script file must be sent using WinScript (Standard, not WinScript Live). The corresponding .amw script files can be found under the "Scripts" directory of the C:\Program Files\Alcorn McBride Inc\WinScriptLive\Scripts.

Modbus TCP Slave Inputs

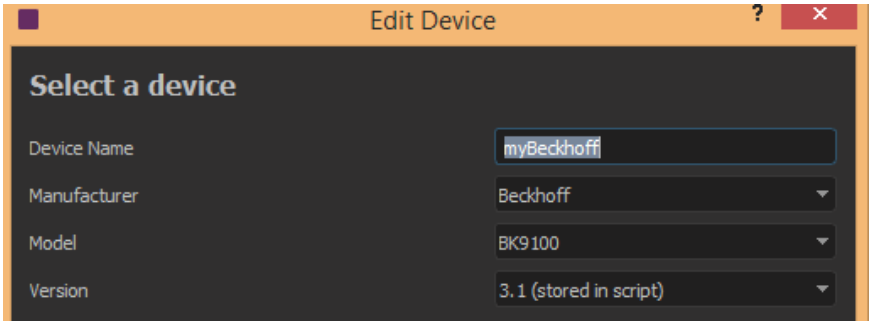
Modbus TCP is a standard protocol used for many IO device modules. Any IO device capable of using ModbusTCP can have its IO controlled as if it was IO internal to the V16/V4Pro. So far, Beckhoff IO and Avantech Adam-6000 series IO have protocol files available. Please contact support if you have a ModbusTCP IO device you would like to control.

The following example will demonstrate how to setup a Beckhoff BK9100 for control by the V16/V4Pro.

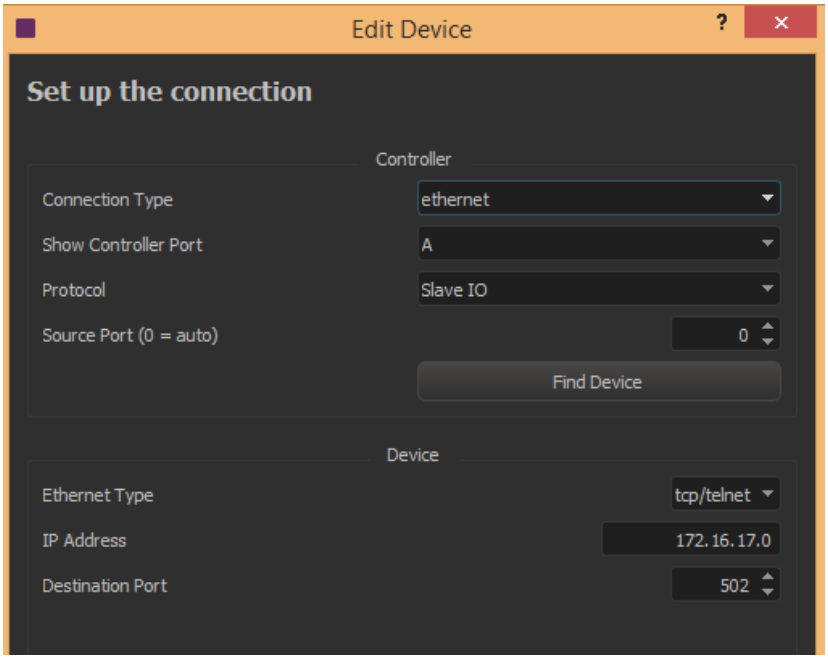
1. Add the IO module to the "Devices" list by clicking the "New" button in the "Devices".



2. Enter a name such as "MyBeckhoff
3. Select the Make, Model and version of the IO



4. Enter the IP Address information, the V16/V4Pro Ethernet port you would like to use, and make sure the protocol is set to "Slave IO"

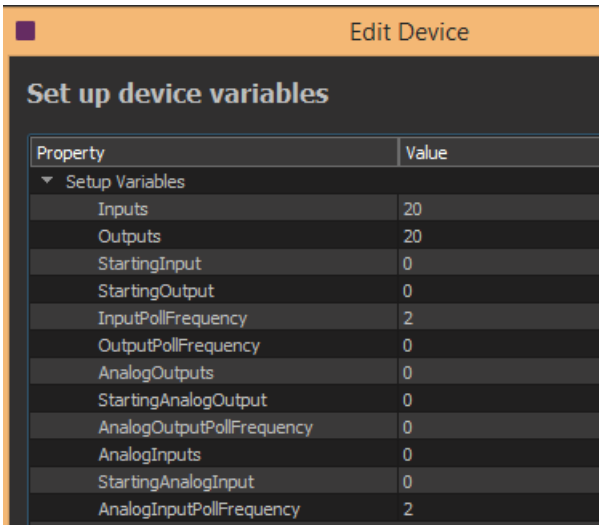


5. Enter the number of "**Inputs**" and "**Outputs**" you will be controlling into the boxes.

The "**StartingInput**" and "**StartingOutput**" are typically 0, but may be a greater number if you're only controlling a sub-section of the IO on a particular control module. For example, if you only wanted to watch inputs 3-12 on a module that had 0-12 available.

The "**InputPollFrequency**" can be set to as little as 1 frame. If you have greater than nine devices that are setup to poll inputs, you must decrease the polling to 2 frames or more.

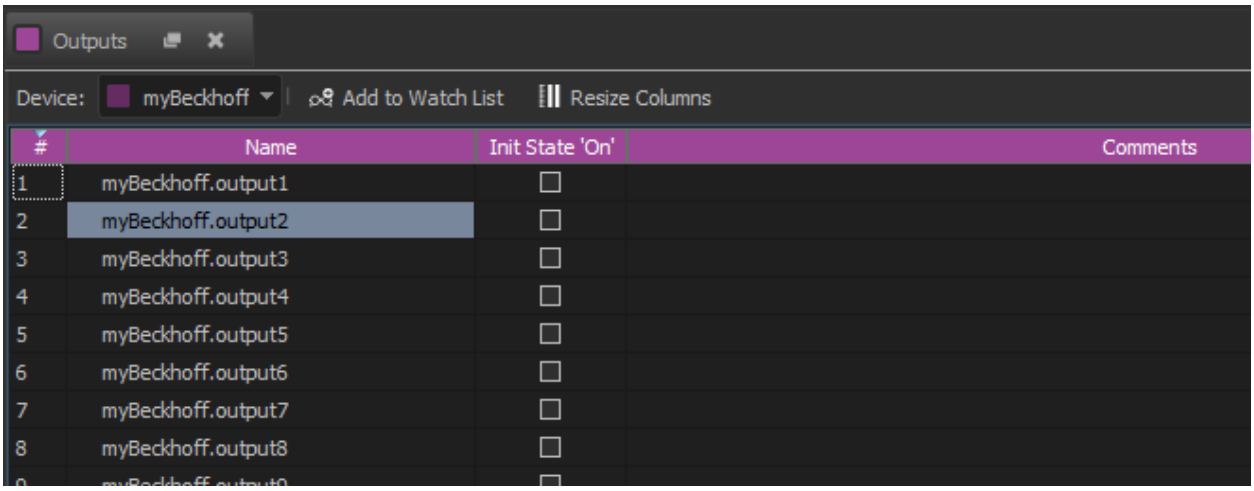
The "**OutputPollFrequency**" is typically set to zero. This does not mean that the outputs will never be polled. They will be polled on startup and after any command is send to change the output's status. (ie: after an "On", "Off" or "Toggle" command). If you prefer more constant polling, a recommended value would be 15 frames.



6. Click "Finish"

Your IO will now show up in the "Inputs" and "Outputs" lists. To view your Beckhoff IO inputs:

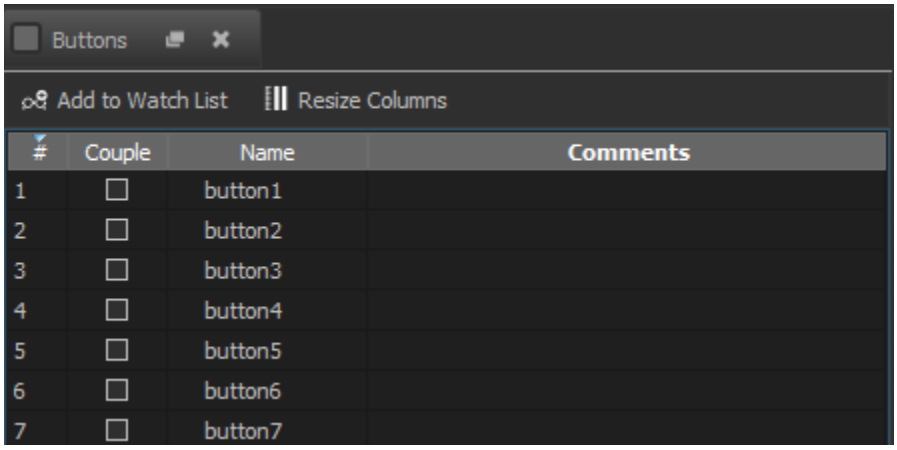
1. Go to "Resources"→"Outputs" to view the output list.
2. Select the name of your device (in our example: "MyBeckhoff") from the toolbar



You may now use your outputs just like any other internal output. Use commands such as "Off", "On" or "Toggle" to control the output state. Commands such as "If On" or "If Off" may be used as well.

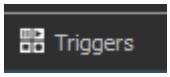
Buttons

Buttons refer to the front panel buttons found on the show controller. By default, these buttons are **not the same** as the rear inputs. Checking the "Couple" box causes a press of the button to have the same effect as a rear pulse on the corresponding input. For example, checking "Couple" on "Button1" will cause any triggers relating to "Input1" to occur when "Button1" is pressed.

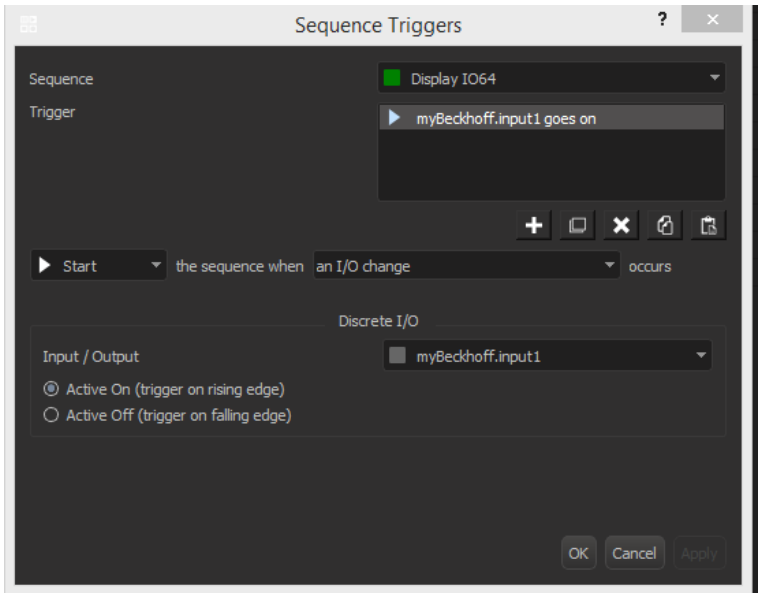


Triggers

Triggers are a method to start or stop a particular sequence. Multiple triggers can be added for a single sequence.

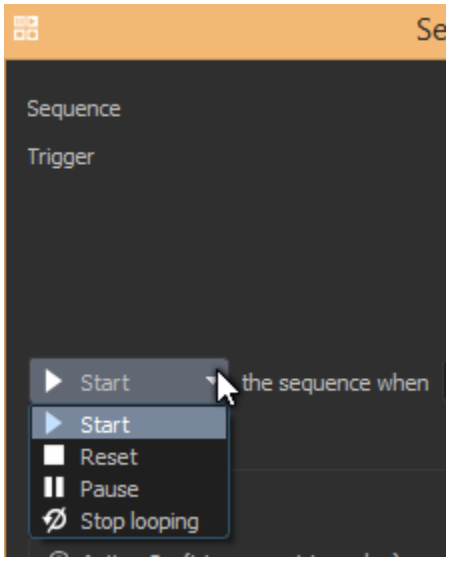
Clicking on the Triggers 

button on any toolbar will allow you to view or edit **all triggers** in your show.



Trigger Types

Triggers can start, reset (stop), pause or stop looping a particular sequence.

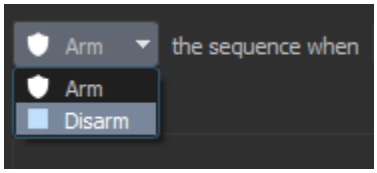


Note: **pause** does **not** have the same meaning as previous versions of WinScript.

Start	Start a sequence running
Reset	Stop a sequence, and start the sequence at the beginning if it is started again
Pause	Stop a sequence at its current location, and resume from that point if it is started again
Stop Looping	Stop the sequence as soon as it reaches the end of the sequence (if it is looping)

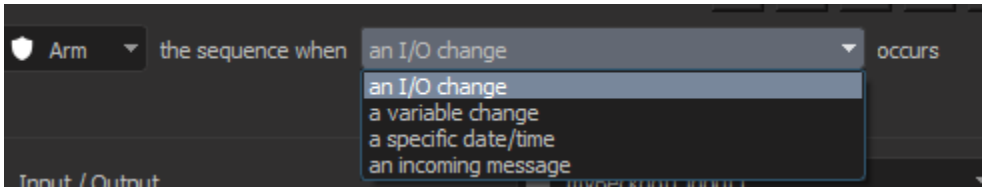
Trigger Definitions

In the case of sequences based on SMPTE/EBU timecode (LTC), a sequence can either be "armed" or "disarmed." Basically, a sequence is allowed to be chased to the timecode or to ignore the timecode.



Trigger Cause

There are four reasons a trigger might activate: I/O change, Variable change, Date/Time or Incoming Message.



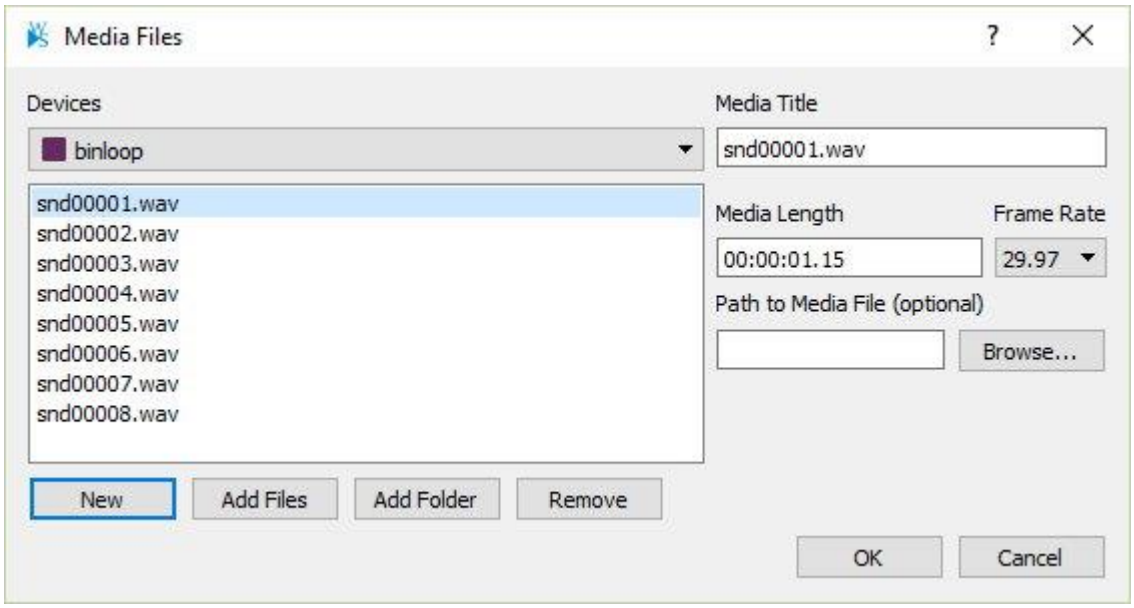
- **IO:** Any input, output, button going "on" or "off"
- **Variable**: A Variable matching a specific value or matching another Variable.
- **Date/Time:** Any time of day with an optional repeating pattern
- **Incoming Messages:** Any message coming in on a specific device's port. This message may already be defined in the protocol file or set as a "custom" message.

Note on Incoming messages: If an incoming, unsolicited message is already defined in the product file (.prd), it will be checked BEFORE any "custom" incoming message triggers in a script. If the incoming message that is defined in the protocol file is found, that string will thus not be checked against the "custom" trigger. The same applies for incoming responses to a product-file defined commands.

Media Files

Media Files stores a list of media used by each Device. The actual content data is not stored in the WinScript file. Only information about the media is stored. The information can be used as a convenient way to organize, access, and visualize the media within the script.

Each Media File entry can track the Title, Length, and Path to the file if it exists on the computer. The Media File entries can also store more specific data like video frame rate.

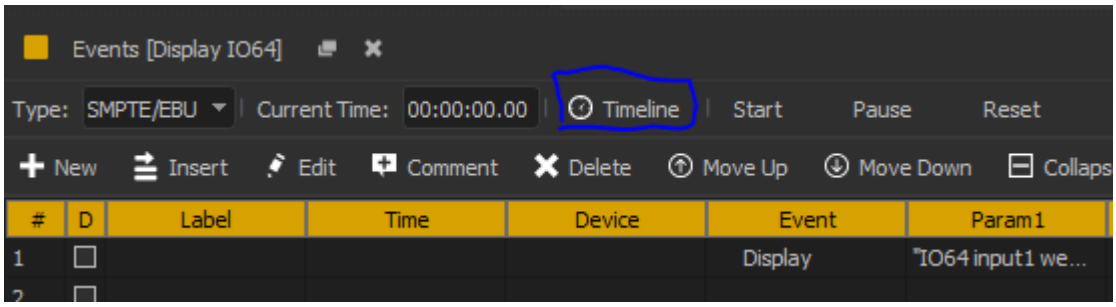


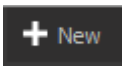
Timeline

Timeline allows for the intuitive editing of sequence events. Elements are displayed in the order they happen on a time line. The following explains all the different options and tools for Timeline.

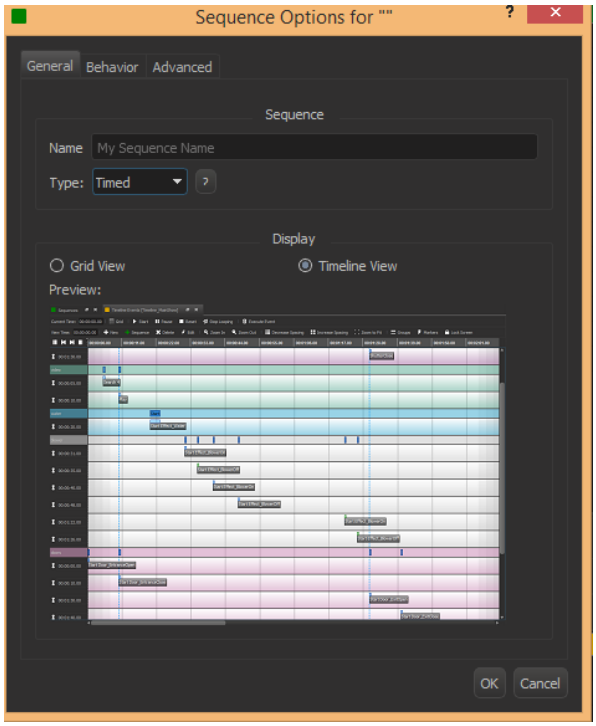
Display Timeline

To view the as a Timeline, click on the Timeline button after opening a specific sequence.

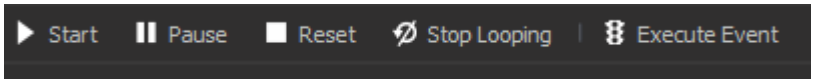


Alternatively, you can create a new Sequence as a timeline by clicking the 

button.



Play, Pause, Stop, and Execute



Play: This button will play the entire sequence.*

Pause: This button will pause the current sequence at the time it is pushed.*

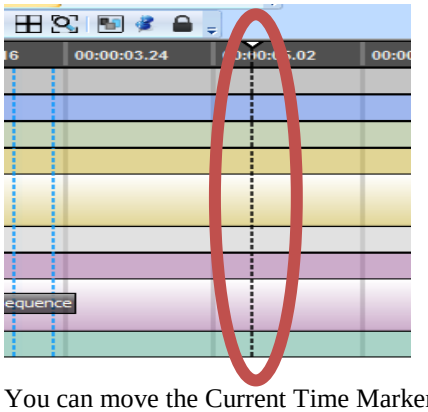
Stop: This button will pause the current sequence and set the play marker at the beginning.*

Execute: This button will executed only the selected event.

** Live Mode Only*

Current Time Marker

In Live Mode, the Current Time Marker marks the current time as a sequence is playing. It automatically moves through the Timeline as the sequence progresses.

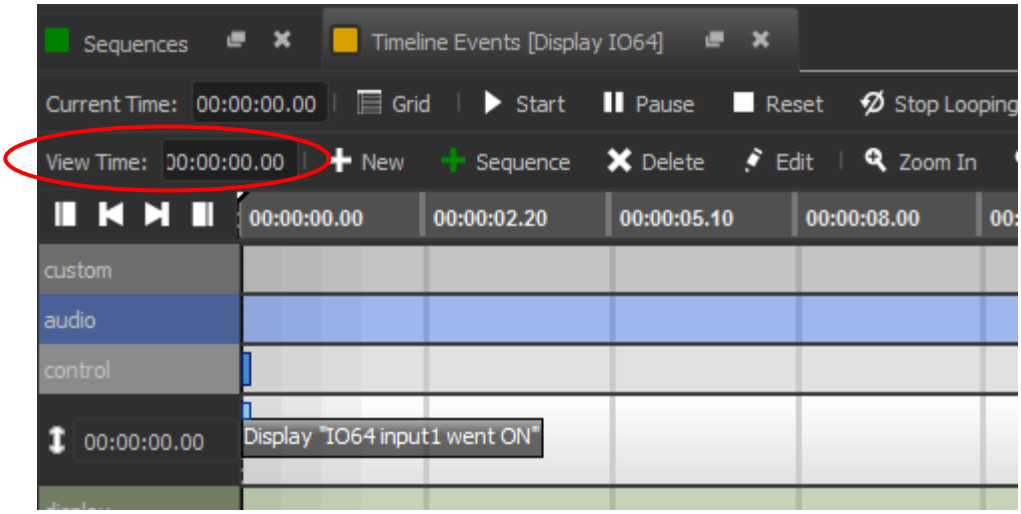


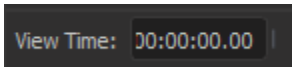
You can move the Current Time Marker to dynamically navigate through the sequence by dragging the arrow on the header part of the marker. 

Timeline Specific Functions

This section explains the different Timeline-specific buttons on the timeline toolbar. To read the name of any specific button, hover your mouse pointer over the button to display the tooltip.

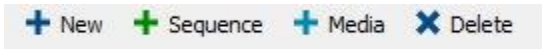
View Time



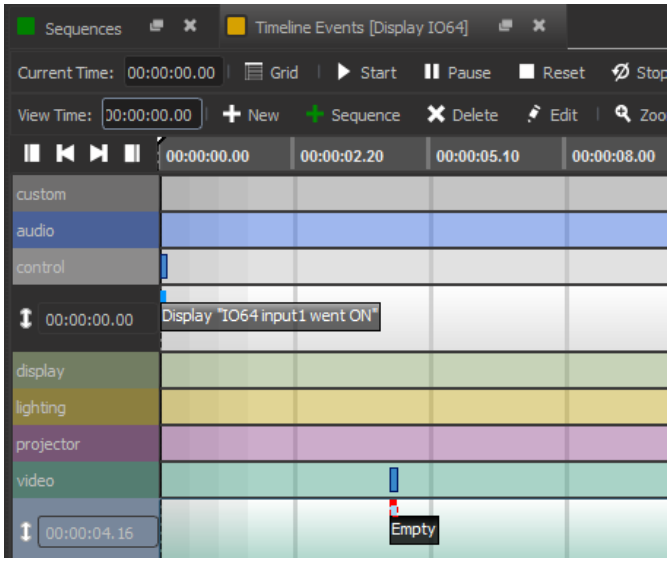


This field represents the current time that is viewed on the Timeline (hours : minutes : seconds : frames). You can edit the field to view any specific time on the Timeline. In addition, you can change the time viewed by using the scroll bar located at the bottom of the window.

New Event, New Sequence, New Media, Delete

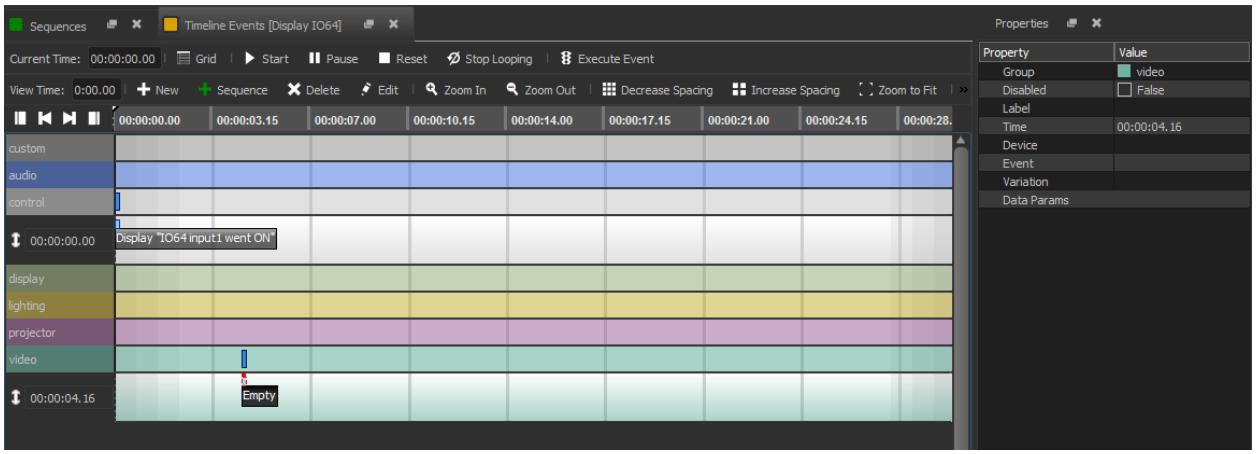


New: This button creates a new event. The event is by default named "Empty" (no operation) and it is placed on the group (row) that was last selected at the time that was last clicked.

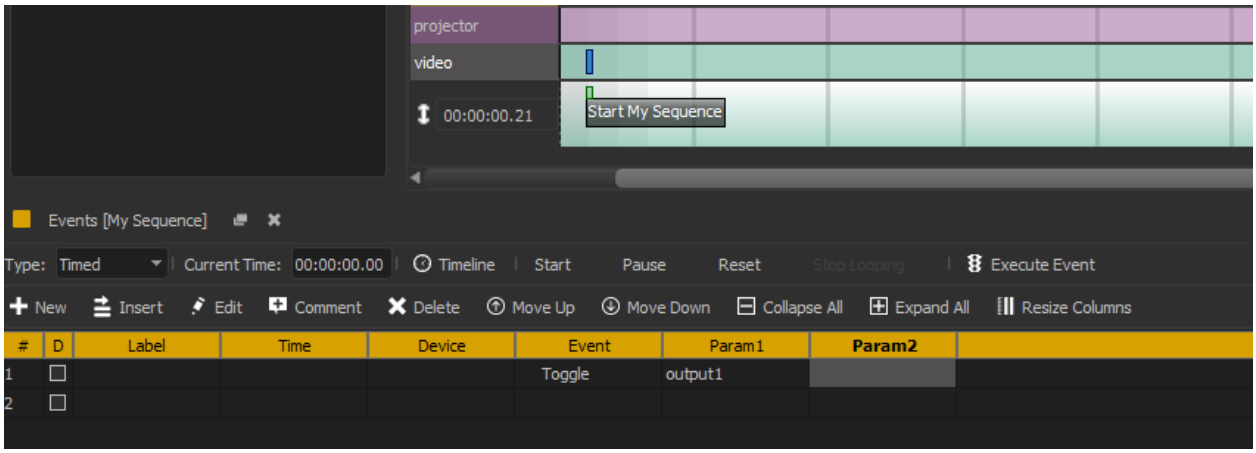


New Sequence: This button creates a new sequence. The program allows you to name the sequence as soon as the button is clicked. Also, an event to execute this sequence is created automatically within the

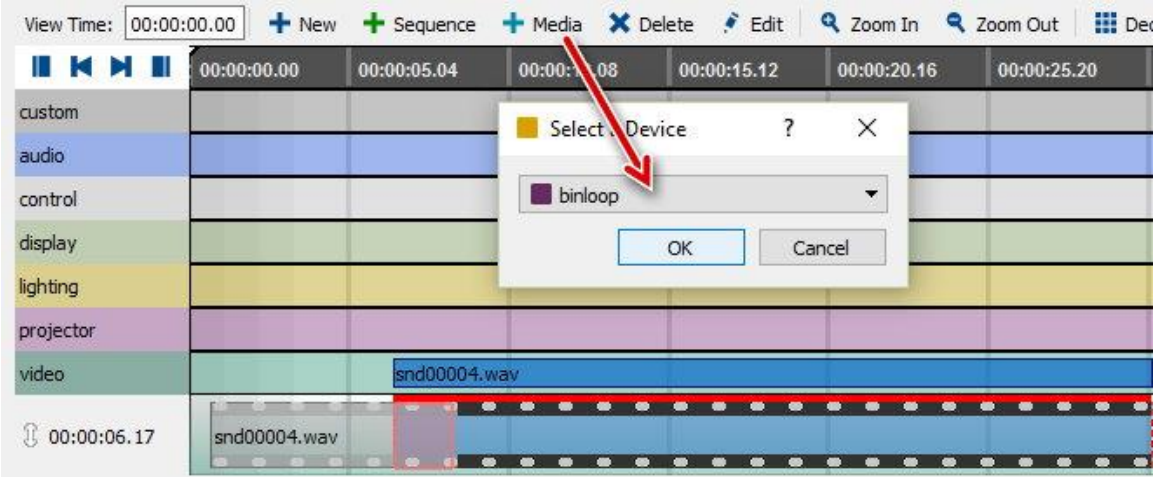
sequence in which you are currently working. This event is then placed on the group (row) that was selected at the moment the button was clicked. This event, like with any other, can be edited with the *Properties* window at the right of the screen.



The contents of the newly created sequence can be edited in the Grid window that appears below the Timeline when the event is double clicked. An event "toggle" is shown in the sequence below. More events can be added to the newly created sequence by double clicking on the sequence in Timeline or editing the events in the Grid window.



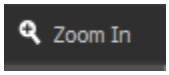
New Media: This button creates a new media event for any existing device that has a command with media file support. Once a device is selected, the media files dialog will allow a file to be added or selected to be placed onto the timeline.



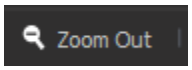
Delete: This button deletes the selected event.

Display Options

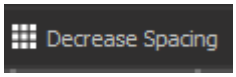
These options allow you to customize the Timeline view so that it is more comfortable to use.



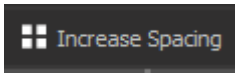
Zoom In: This button allows you to zoom into the Timeline to configure any small details. The maximum zoom-in level is one frame.



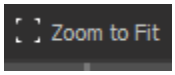
Zoom Out: This button allows you to zoom out of the Timeline to see the bigger picture. The maximum zoom-out level is twenty two hours.



Decrease Spacing: This button allows you to decrease the spacing between the different groups (rows). This is useful for when multiple rows and events are added as more rows will fit on the screen.

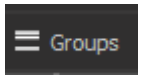


Increase Spacing: This button allows you to increase the spacing between the different groups (rows). This is useful for having a clearer view of the Timeline.

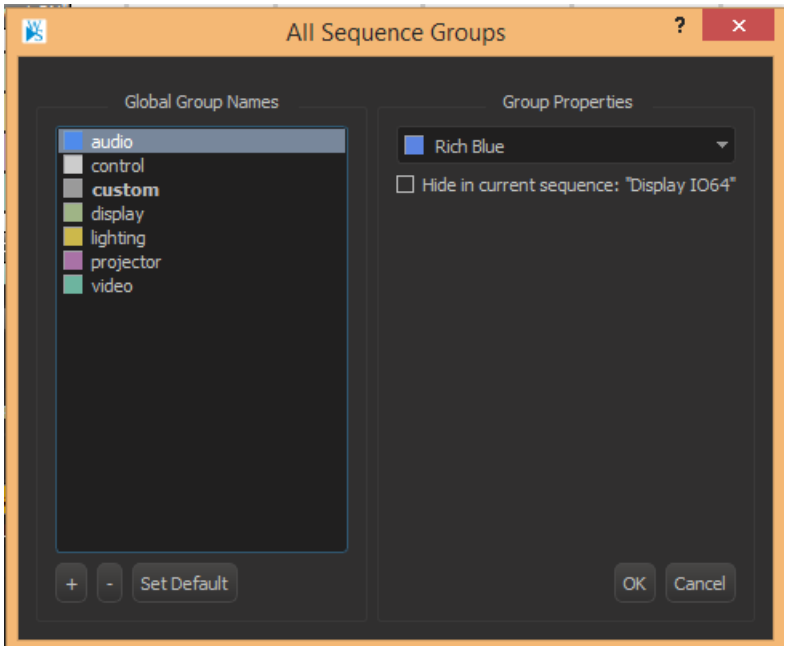


Zoom to Fit: This button allows you to fit all of the events on the screen. Zooms-in or out automatically to allow you to see all the events. The page will now start displaying at the time when the first event occurs and end when the last event is done.

Groups

In Timeline each group is represented by a row. These can be customized using the **Groups** button on the tool bar. This menu can also be accessed by clicking the **View** menu and clicking on **Groups**. 

In this new window, shows the list of all the groups. From this window, you can add new groups, delete old groups, change their colors, hide a specific group for a sequence, and set a default group.



Add New Group: To add a new group, click the "+" symbol. This will add a new group to the global list with a default name. To change the name, double click on the new group from the list, and it will allow you to edit it. Any groups you add will be included in any other sequences; to hide a group from a sequence, click the **Hide** checkbox.

Delete Group: To delete a group, select a group from the list then click on the "-" symbol. This will delete the currently selected group from the list.

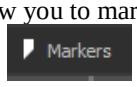
Set Default: This button will set the selected group to be the default. When a new event is created in grid view, if the group is not specified, the default group will be selected, and the newly created events will be added to it.

Change Color: By clicking on this dropdown list, you can change the color of a group by selecting a specific color from the list, or click **More…** to pick a color from the palette.

Hide in Current Sequence: Mark this checkbox if you wish to hide the selected group from the current sequence.

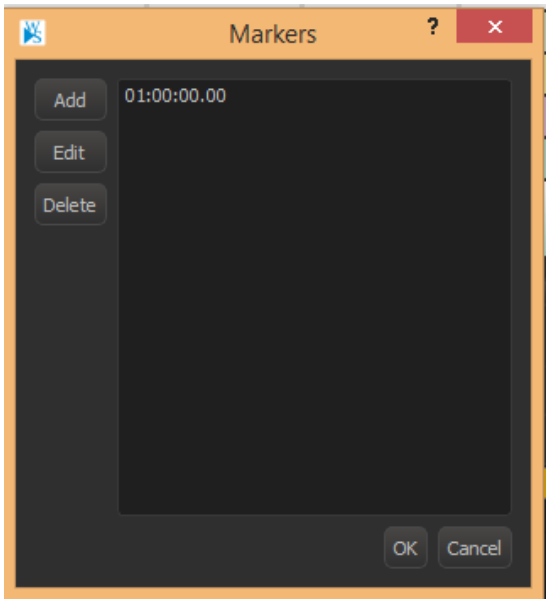
Once you are done editing your groups click **OK** to save your changes.

Markers

Markers allow you to mark a specific time on the Timeline. They can be accessed and edited by clicking on the **Markers** 

button or, alternatively, in live mode, they can be placed at the current frame by pressing **M** on your keyboard as the current time marker advances.

Markers allow you to easily place events next to them by automatically pulling the event next to it as you drag it close to the marker. Alternatively, you can select both the event and the marker, then **right click**, and select **Snap to Selected Marker** to accomplish the same as described before.



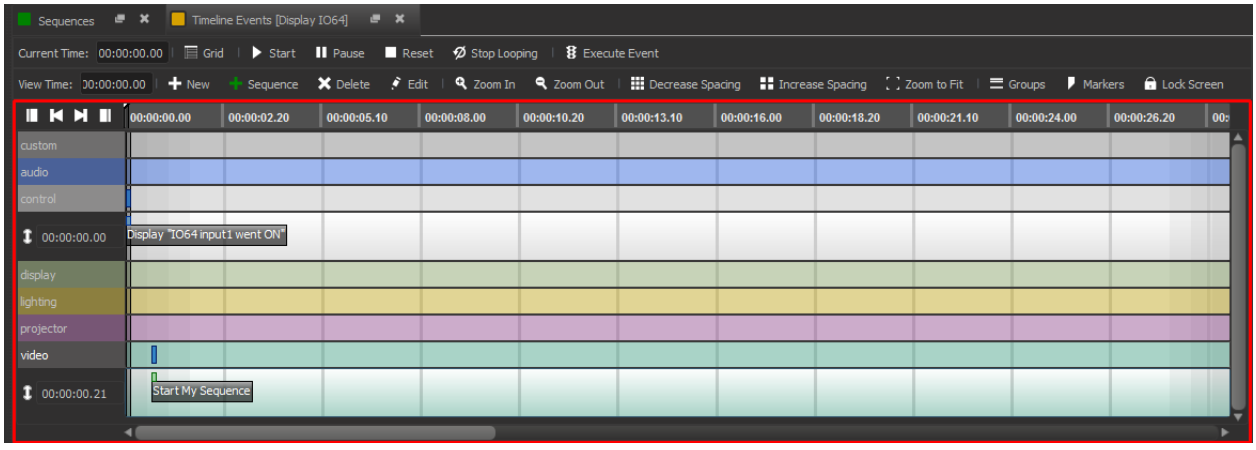
Add: Click this button to add a new marker. Once clicked, you can type the specific time in which to place the marker.

Edit: This button allows you to edit the time for the selected marker.

Delete: This button deletes the selected marker.

Lock the Screen

When locked and in Live Mode, as the current time marker progresses through the sequence, the view will automatically follow the marker. Clicking the lock button again, will allow you to scroll through the sequence again. Clicking this button creates a red margin around the timeline.



Event Buttons

 These buttons help you to easily navigate your events.

Select First Event: This button will select your first event, taking you to it on the timeline.

Select Previous Event: This button will select your previous event, taking you to it on the timeline.

Select Next Event: This button will select your next event, taking you to it on the timeline.

Select Last Event: This button will select your last event, taking you to it on the timeline.

Other Functionalities

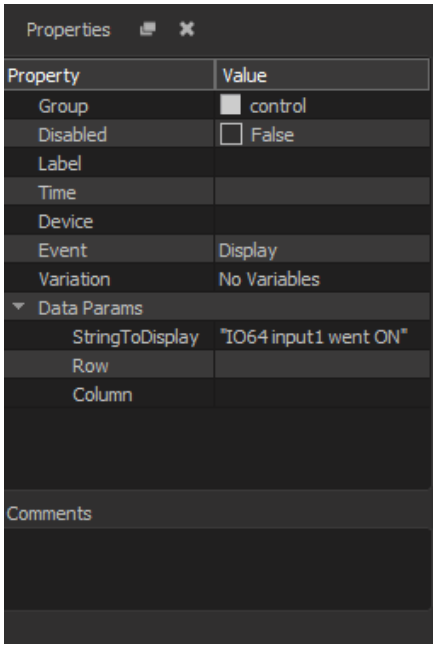
Creating Events and Dragging: You can create a new event by dragging it from the resources list on the left, and dropping it on the desired spot. When dragging an event to a time that is outside of the current view, the timeline will automatically expand as you approach the margin of the Timeline while dragging an element. This functionality has two different speeds depending on the distance to the margin.

Changing Groups: After an event has been created, you can change the group by dragging and dropping, or changing it from the **Properties** window.

Other Way to Access Options: Most of the Timeline options can be accessed via the **View** top menu.

Properties Window

The properties window allows you to configure an event. It provides the same options that are found in Grid view, and the parameters that are specific to an event. All of the values shown can be edited as needed.



Data Params: The list of parameters changes depending on the type of the event that has been selected.

Explanations on what each parameter does can be found at the bottom of the window.

The **Properties** window can be moved around the screen and be docked to the side of the application itself. Also, you can put two panes (Properties and Resources) on top of each other if needed. The application will save this customization for the next time the program is opened. If you need to return to the default layout, just go to the **View** menu, and click on the **Default Layout** button.

Note: An event will often times have multiple "Variations." After selecting an Event, be sure to select the appropriate Variation. Changing the Variation will change the Data Params.

Working with Media Files

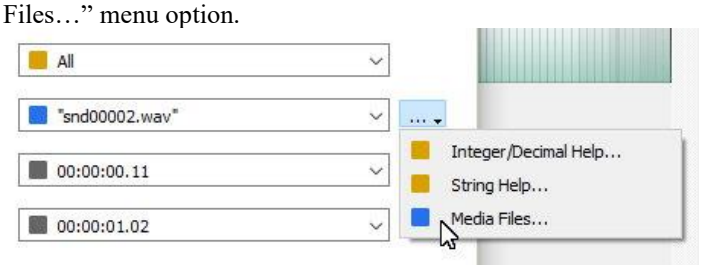
The following sections describe how to work with devices that play audio, video, and other time-based media formats. Before you begin, add a device that has product file support for media file playback (such as the Alcorn McBride Inc. Uncompressed Binloop). See the section on WinScript Live Resources Devices for how to add new devices to your show. Once a device with media file support is added, all media file features will become available.

IMPORTANT: WinScript Live will not store your Media Files in the .ws4 file or on your device. Management of media file data is the user's responsibility. Media File Resources in WinScript Live are simply references to actual media, which is used to simplify adding and synchronizing show events.

Adding Media

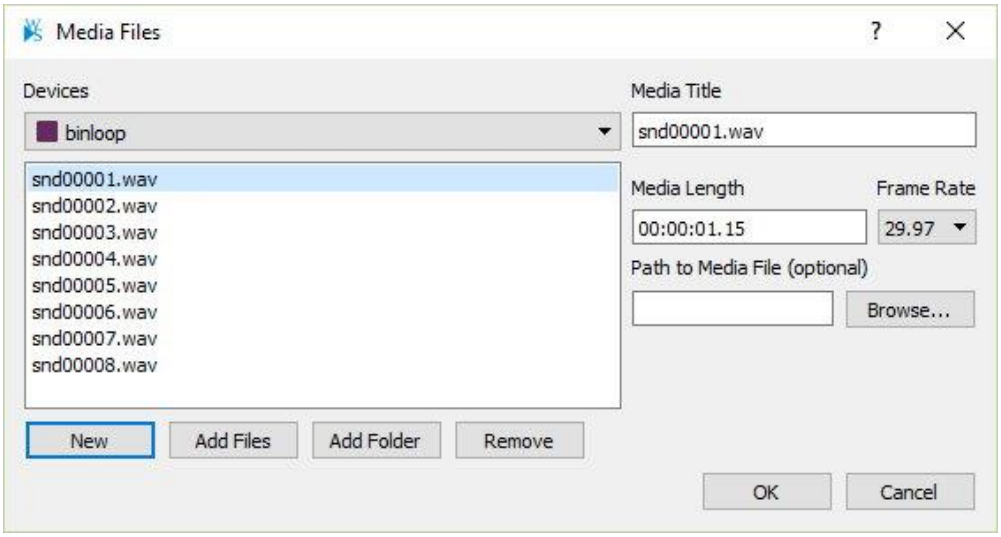
There are several ways to access the Media Files dialog that allows Media Files to be added and removed from a device resource.

- Resource Menu – From the menu bar, select Resources->Media Files and then select the device you would like to work with from the drop down menu
- Resource Tree – From the menu bar, select View->Resources if you do not already have the tree view open. In the Resources tree view, select the device you would like to work with and then double click on the Media Files resources to open the Media Files dialog for that device
- Timeline – When a Timeline Sequence is open, select the Media button to choose a device to work with. The selected media file will be added to the Timeline automatically when OK is pressed.
- Event Wizard – Add an event to any Sequence and open the Event Wizard. Choose a media playback command that supports Media Files and click the "…" button to access the "Media Files…" menu option.



Media Files Dialog

The Media Files Dialog allows the user to add, edit, and remove Media File information for a device.

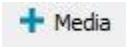


- Devices – Select the device to work with
- New – Add a new Media File entry to enter manually
- Add Files – Select one or more files from the local file system. The file name will be used as the Media Title. This is a very fast way to add multiple Media Files.
- Add Folder – Select a folder from the local file system. The folder name will be used as the Media Title. This is used for devices that play an entire folder instead of a file.
- Media Title – The name that is used when generating the play command for the device. If the device uses file numbers and the name contains the number, it can be automatically converted to the proper value depending on how the product file was written.
- Media Length – The entire length of the media in hours:minutes:seconds:frames. The frames field is in the Frame Rate specified for the media file.
- Frame Rate – The frame rate of the media file. This can be different than the frame rate of the sequence and the controller will perform the calculations necessary to display the media properly.
- Path to Media File – If the Add Files or Add Folder button is used, this is a reference to the location that was selected. This is an optional field.

New Media Event

Once Media Files have been added to a device, they may be used with any media event. The easiest way to add a media event is to drag a Media File from the Resource tree and into a Sequence. This will work with Sequences in the Grid view, but it is recommended to use a Timeline Sequence because it is easy to visualize when the media will play in relation to other events in the show.

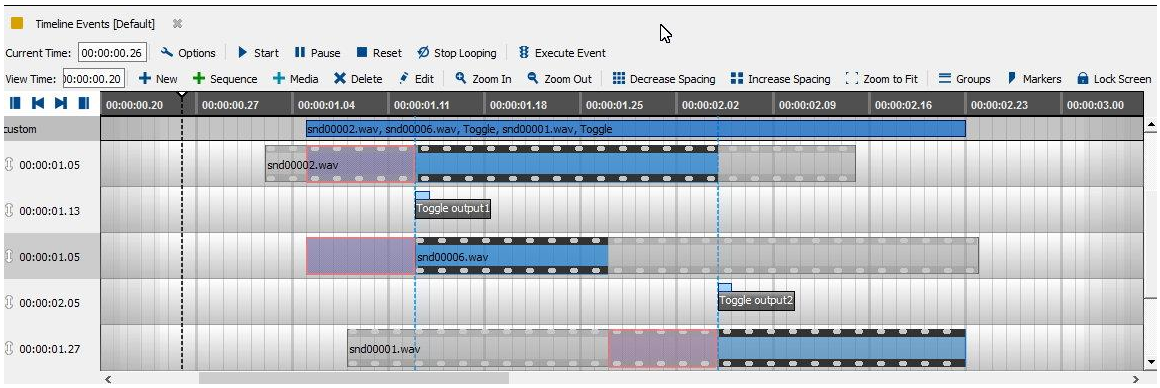
Media in a Timeline Sequence

Either drag a Media File from the Resources tree or add a Media File using the 

toolbar button.

A Timeline Sequence will show a "film-strip" of the media length. In addition to the media length, the following optional visuals will be displayed:

- Start Offset – Shown as a grey portion at the start of the media
- Sync Delay – Shown as a pink highlighted portion at the start of the event
- Media Playback – Shown as a blue portion of the media
- End Offset – Shown as a grey portion at the end of the media. The controller will send a stop command to the device at the first frame of the end offset.



Synchronizing Media with other Events

Media Events will automatically snap to the frame where playback will begin. This is different from normal Events that, when snapped to a marker, will move relative to each other.

To synchronize multiple events:

1. Add a marker to the timeline or place the current time marker at the desired start location
2. Hold shift and select each event that you would like to synchronize. You may select any combination of normal events and media file events.
3. Drag the selected events to the marker and all selected events will align themselves to begin playback at that exact time. You may notice that events with a pink "sync delay" begin earlier than the marker location. This is because WinScript has calculated when the command to play must be sent to the device in order for playback to begin at the desired location.

Scrubbing (jump within) Media Files

Some devices, such as the Alcorn McBride Inc. Uncompressed Binloop, have commands that can begin playback at any offset from the start of the media. When a sequence is running, Devices that support a start offset will automatically scrub (jump to) the current sequence time. The sequence time can be changed using the "Set Seq Time" event, by clicking the Timeline header while connected to Live Mode, or by any changes to SMPTE timecode whether the controller is generating or reading from an external source.

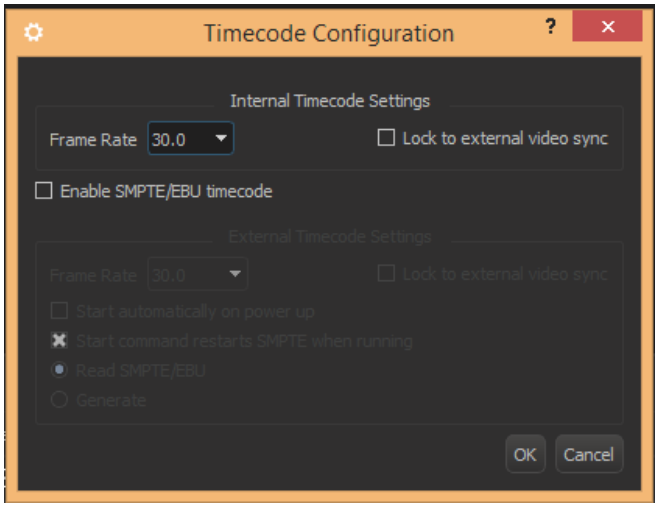
WinScript Live Timecode (SMPTE/EBU)

The following section will explain the different ways to configure global timecode (SMPTE/EBU) settings for the show controller using WinScript Live. For information on timecode settings for a particular sequence, see Sequence Clock on page 14.

Display the Timecode Configuration Dialog

From the menu, go to “Configuration” and select “Timecode”

The following dialog appears.



Internal Timecode Settings

The internal clock frame rate settings determine what rate a sequence’s timecode will increment. The following frame rates can be used: 23.976, 24, 25, 29.97, 30D (drop), and 30fps.

Lock to External Video Sync

Locking to an external sync source can help keep your show perfectly timed with external devices such as a video player. The Alcorn McBride’s A/V Binloop HD is an example of a device that can also accept a Video Sync input from a Blackburst generator.

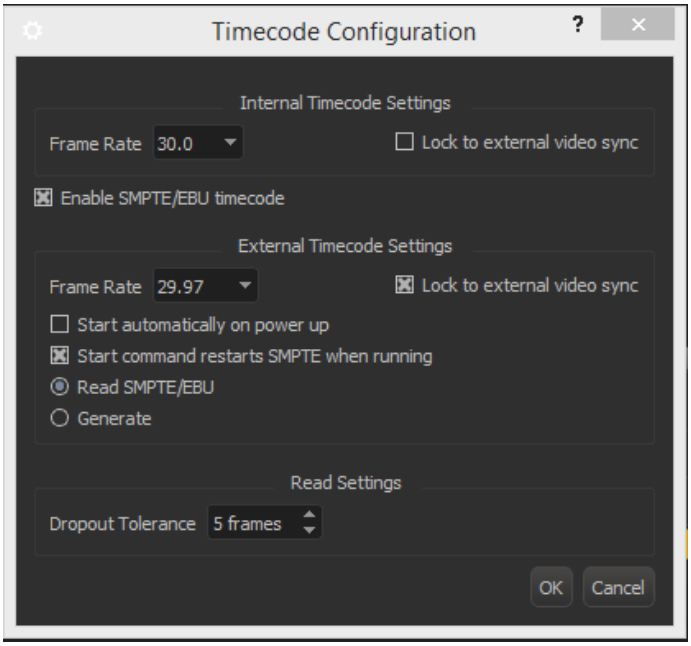
When using an external sync source such as a Blackburst or C-Sync generator, the show controller’s internal clock can be matched to that generator’s frame rate. Consequently, only 23.976, 25, and 29.97fps are available when this mode is selected. A yellow LED on the front and rear of the show controller indicates when the unit is receiving an incoming sync signal.

External (SMPTE/EBU) Timecode Settings

Linear Timecode (LTC) in the form of SMPTE/EBU can be used to synchronize multiple devices to a single running timecode. The Show Controllers contain a LTC reader or generator. Once configured, sequences can be locked to this LTC instead of the above mentioned "Internal Clock". For more information on configuring an individual sequence's clock, see Sequence Clock on page 14.

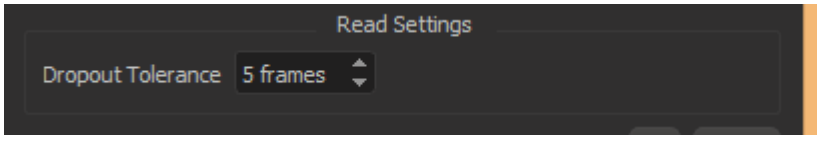
General Settings

- **Enable SMPTE/EBU Timecode:** Allow the LTC to be configured for any usage.
- **Start Automatically on Power Up:** In generate mode, start the timecode running as soon as the show is loaded. In read mode, allow timecode to be read as soon as the show is loaded.
- **Start Command Restarts SMPTE when Running:** When checked, a "Start Timecode" command sent either through external ASCII control, front panel press or sequence command will cause the timecode to start at the beginning.



The frame rates available for LTC are 23.976, 24, 25, 29.97, 30D (drop frame), and 30fps. When "**Lock to External Video Sync**" is selected, 23.976, 25, and 29.97 fps are available. A yellow LED on the front and rear of the unit indicates when the show controller is receiving an incoming sync signal.

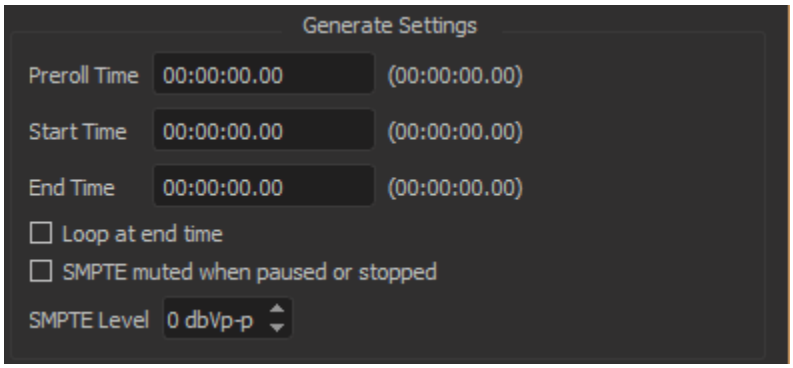
Read Settings



When reading external LTC, it is possible for the timecode to skip or "dropout" a few frames. This "Tolerance" level indicates at what point the show controller will register a "dropout".

A good level for this is usually 3-5 frames. If the timecode skips ahead (or behind) a number that is **less** than the tolerance, the sequence will continue to execute all events normally. If the timecode skips ahead (or behind) a number of frames that is **more** than the tolerance, a "dropout" will be registered and the sequence will either jump ahead (jam-sync mode) or reset (reset mode). For more information on configuring an individual sequence's clock, see Sequence Clock on page 14.

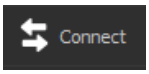
Generate Settings



- **Preroll Time**: Occurs once, on initial start, before reaching the "Start Time." Any further loop will go back to the Start Time.
- **Start Time:** The initial running time
- **End Time:** The time that the timecode stops or loops
- **Loop at End Time:** Returns to Start Time when End Time is reached
- **SMPTE Muted When Paused or Stopped:** No SMPTE time signal is generated on LTC output when Paused or Stopped.
- **SMPTE Level:** The output level of the SMPTE/EBU signal in dbVp-p

WinScript Live "Live Mode"

When operating in "Live Mode", all modifications to the script take immediate effect within the show controller. In addition, resources such as “Watches”, "Live Log" and "Live Display" will provide additional debugging and status information.

Connect to live mode using the 

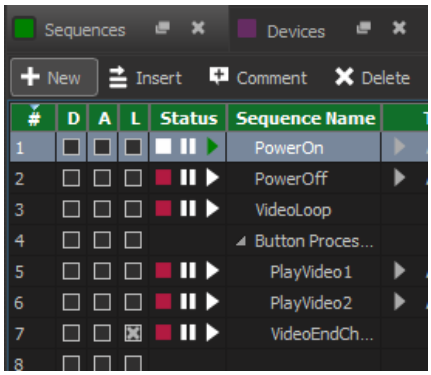
button located on the toolbar. Scripts must be saved and checked for errors before they can be sent to show controller.

After the connect button is clicked, the script's timestamp will be compared to the timestamp of the active file in the show controller. If the timestamps match, live mode will connect immediately. If not, you will be prompted to send your active script or upload the active script from the show controller.

For more details on the exact syntax of the Live Mode protocol, see Live Mode Protocol on page 174.

Sequence Status

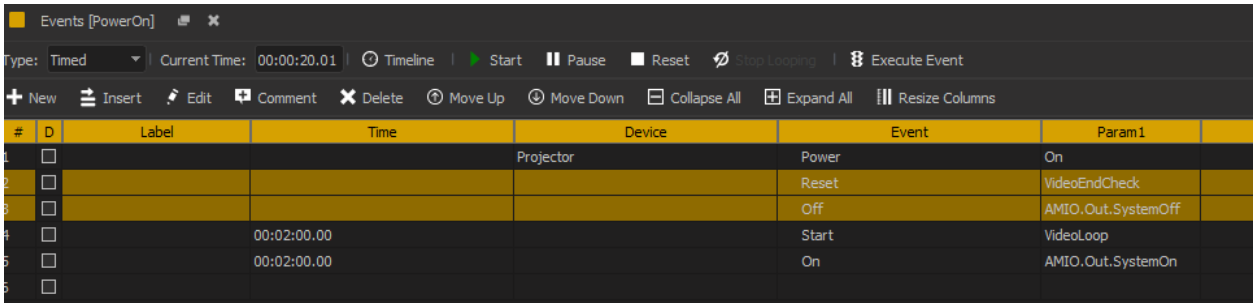
The status column in the **sequence view** will display the current status of the sequence when connected in live mode. The buttons will “light up” to indicate the status.



Event Status

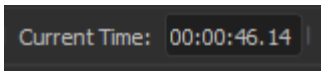
Highlighted Events

After connecting to “Live Mode,” events will appear in yellow **after** they have recently executed. If events are within a few frames of each other, you may not see certain events become highlighted due to screen refresh times.



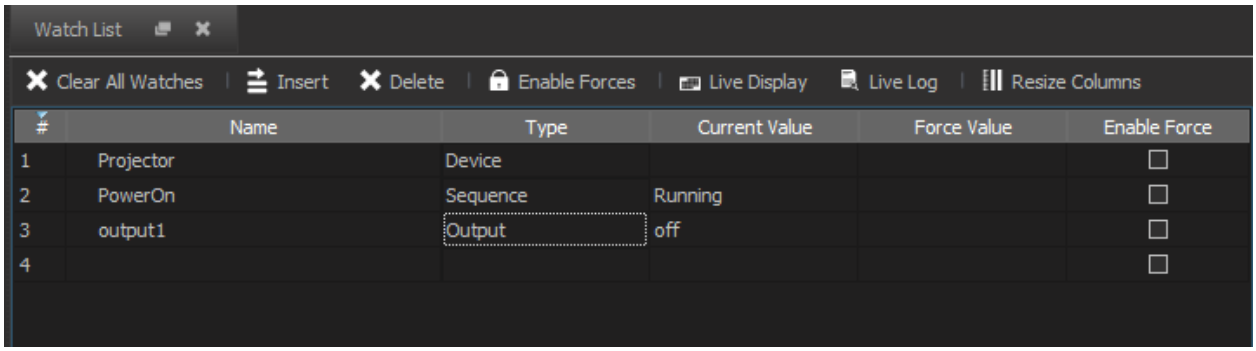
Current Time

The current running time of the sequence can be viewed in the upper right corner of the "Events" window for that sequence.



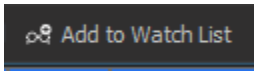
Watches

All resources can be added to the watch list. This list allows for easy viewing of inputs, outputs, and variables, sequences and devices as they change.



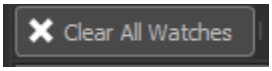
Adding Watches

You can add items to the watch list by typing in the "name" column or by clicking on



from the resource (inputs, outputs, etc) window.

Adding a device into the list allows all communication to and from that device to be displayed into the live log.

The 

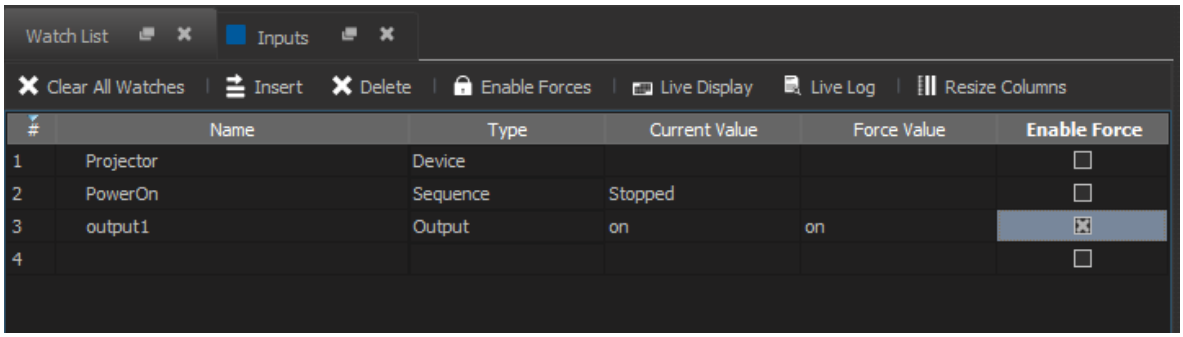
button will delete all watches from the list. This does not affect the resource (input, output, button, etc) in any way, it only the removes the resource from the watches list.

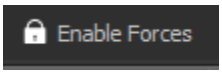
Viewing/Changing Value

The **Current Value** column shows the current value of the resource while the script is running. Typing into the Current Value column sends a command to the show controller to **change the current value**.

Forces

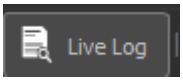
Typing in the **Force Value** column then clicking in the **Enable Force** column to check the box. When the box is checked, a variable's or output's value will not change as instructed by the sequences in the script. Instead, it will retain the constant value of whatever is placed in the Force Value column. When the Enable Force box becomes unchecked, the value will return to whatever value the script has instructed it to be.



The 

button will check all of the "Enable Force" column rows at once.

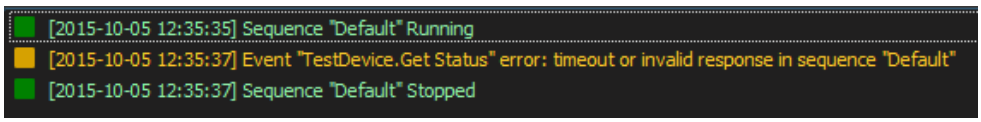
Live Log

Clicking on the 

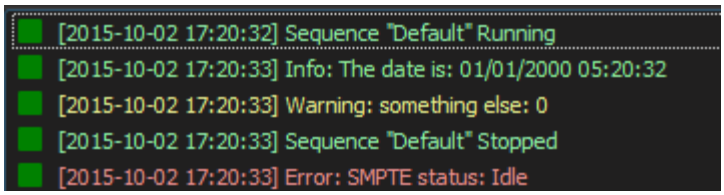
button in the watch window shows a list of sequences as they are started and stopped. Communication to devices listed in the "watches" list is also recorded in the log.

Note: incoming messages will always appear before outgoing messages. If an outgoing message occurs, and an incoming comes in a few frames later, the incoming message may still appear before the outgoing message.

Error Messages that cause the V16Pro/V4Pro "ERR" light to blink on the front of the device are also automatically sent to the log. This message indicates what event and device caused the front error light to blink.



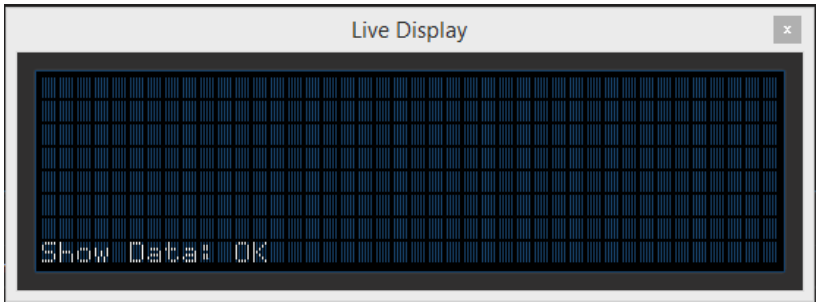
Custom log messages can also be sent to the Live Log by using the "Log Message" event. See "Log Message" on page 84.



Messages using "Log Message" event

Live Display

The live display shows represents the front panel display of the show controller. This is only available when in Ethernet Live Mode. Connections for USB or serial will display a blank screen.

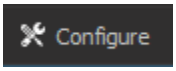


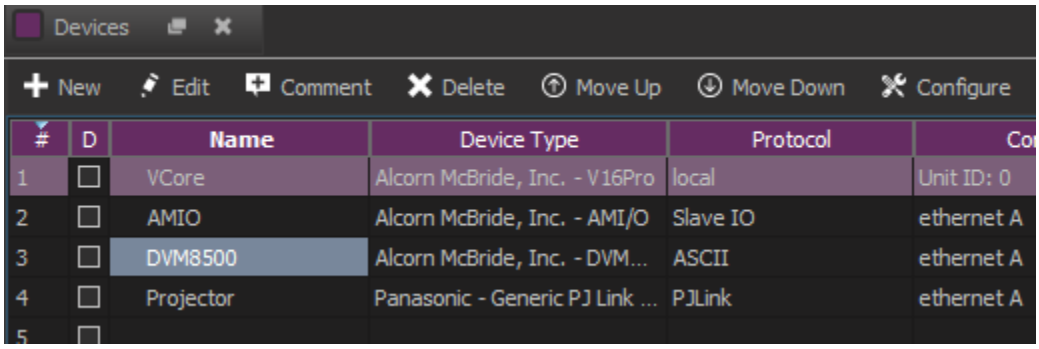
Live Config

The “Live Config” option is used to ping external devices through the show controller itself, and can also be used to set IP Addresses of an AMI/O product.

Pinging Devices

In this section, we will go over how to ping an A/V Binloop HD through the “Live Config” menu. This is useful in troubleshooting the connection between the show controller and its devices.

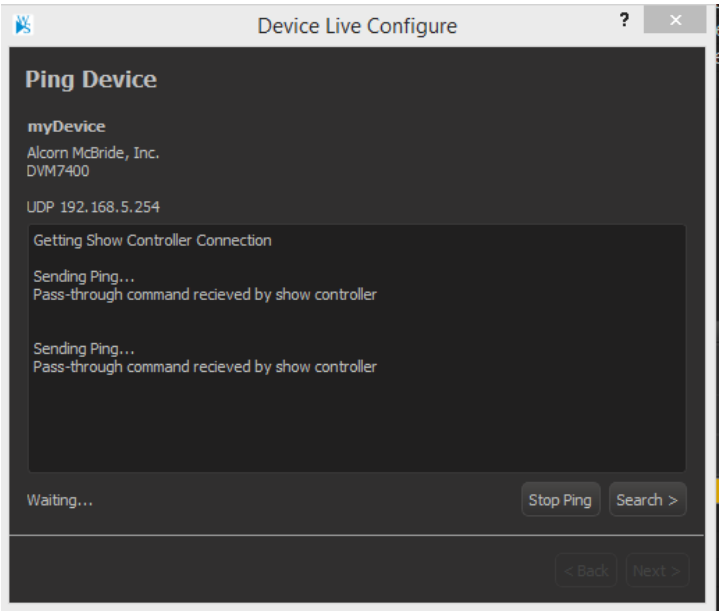
Click  from the “Devices” screen.



Follow the subsequent prompts to successfully connect in Live Mode if you weren’t already.

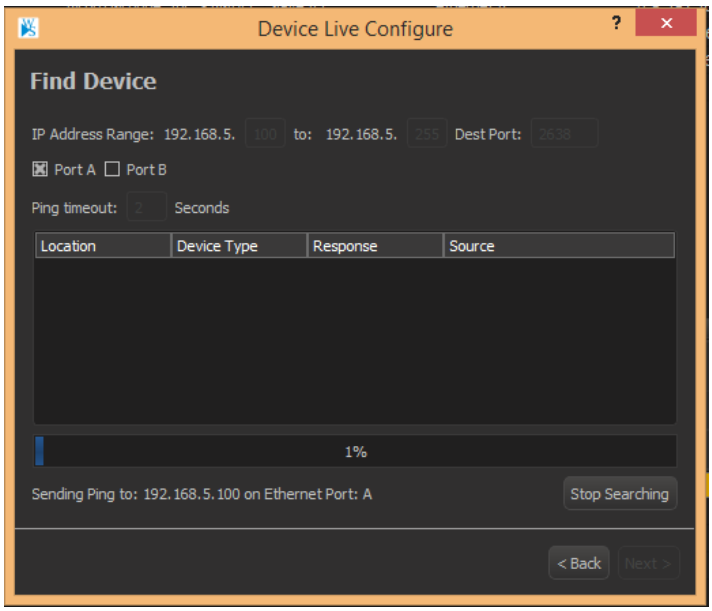
Once connected, it should bring up a “Ping Device” window. After clicking **Send Ping** the window should send some pings and end with “Device Found” if the connection info for the respective device you are pinging was properly configured, and the show controller is able to communicate with it.

If you were unable to successfully ping it, you may have put incorrect connection info for the device, and should read the next section, “Finding Devices”.



Finding Devices

Assuming you were unable to successfully ping it, press “Search”. This will bring up the “Find Device” dialog. Select the **port range** you want to search, the **destination port** you want to ping, and the **ping timeout** time, and the **show controller port** you want to scan through. If the device is not found at all regardless of the settings you use, it may be set to an invalid address and should be configured via the device itself. The process for doing this for an AMI/O product is within the section titled “Resetting IP Addresses – AMI/O”. Otherwise, select your device, click “Next”, and proceed to the “Setting Device Addresses” section.



Setting Device Addresses

After the scan, once you select the device you wish to connect to and click “Next”, a “Change Location” dialog will appear, offering the following options:

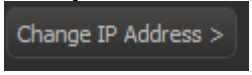
Option 1: Change the device to use WinScriptLive location
This option should be used if you want to change the AMI/O's IP address to match the address you have already specified in WinScriptLive

Option 2: Change Winscript to use found device location
This option is only available if you have already set the IP address of the AMI/O, and have found the device on the same network as your show controller. This will only change the WinScriptLive script and not the AMI/O.

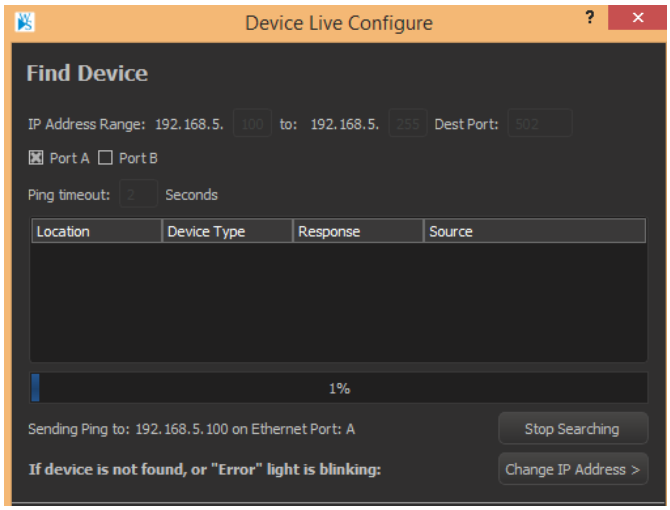
Option 3: Change WinScript and device to be location
The WinScriptLive's script is changed to the specified IP address. The AMI/O device is also set to the IP address specified.

After clicking “Next”, a new window will appear detailing the progress of the previous selected action. Once it finishes the configuring (100%), click “Finish” to close the dialog. Your device connection configuration should now be complete.

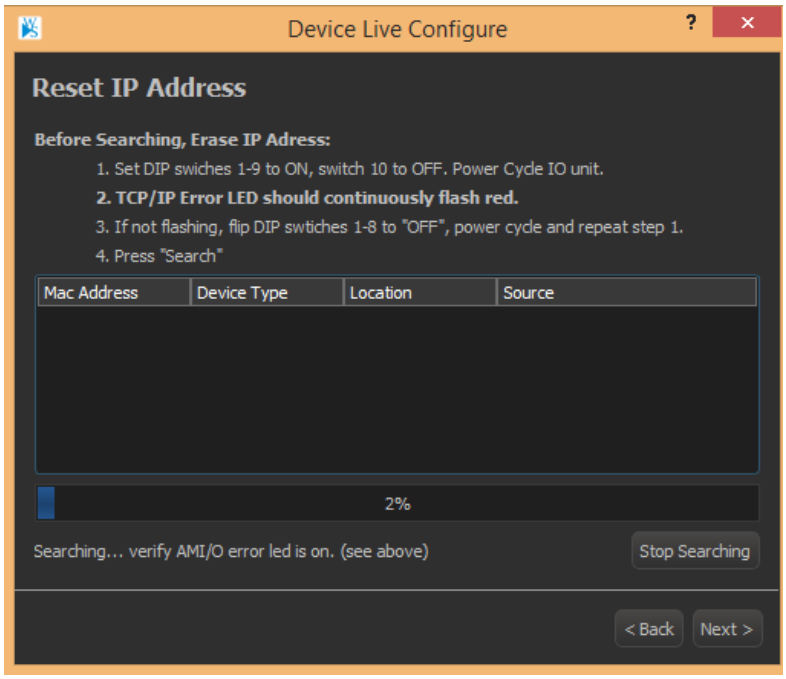
Resetting IP Addresses – AMI/O

The “Live Config” functionality can be used to reset then set up the IP Address of an AMI/O product. If during the “Pinging Devices” tutorial, you were unable to find the address of your AMI/O product, you can select the “Change IP Address” 

option within the “Find Device” dialog



To change address back to 0.0.0.0, the DIP switches must be toggled and at least one cold boot must take place.



To do this, follow the on screen instructions (printed here)
1. Set the DIP switches 1-8 to "ON" and power cycle
2. The "ERROR" led should continuouly flash red
3. If not flashing, flip DIP switches 1-8 to "OFF", power cycle and repeat step 1.
4. Click "Search"

Once the desired AMI/O device appears in the list, select it from the list and click "Next" to move to the "IP Set" screen. Go to the preceding “Setting Device Addresses” section for assistance on setting the IP.

Note: if the device does not appear, make sure you are connected on the same local network and that the "ERROR" led is blinking red

Show Controller External Control

Many methods exist for sending commands to your show controller to start sequences. Several common methods are listed below.

"ShowTouch" Software (For PC, iPhone, iPad)

ShowTouch is a rugged touch-screen interface designed to work exclusively with with our Ethernet show controllers. The script running on the show controller is read by the Touch software so that buttons, toggles, and text displays can be linked to the show controller quickly and easily. The Touch software that runs on ShowTouch will also run on any Windows PC, and it's free!

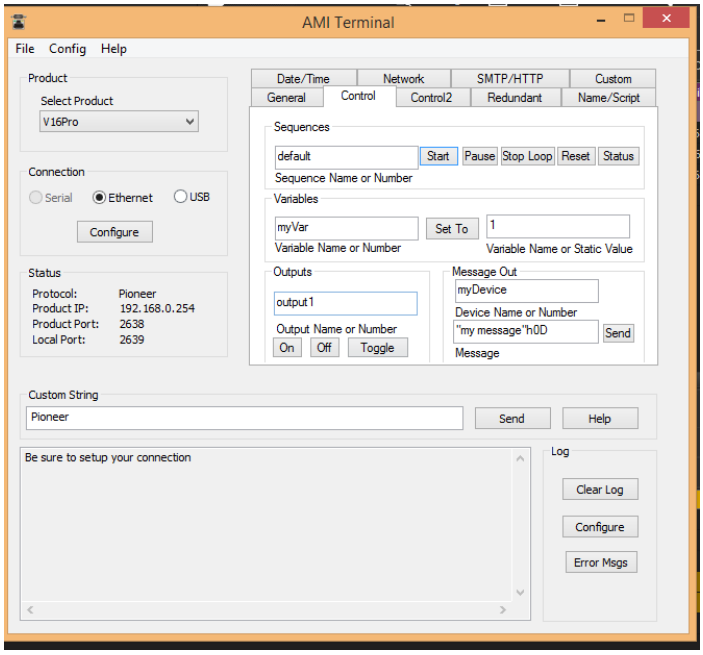
When using the "ShowTouch" software, no additional setup is required in WinScriptLive. Simply run the touch software and retrieve the script from the V16/V4Pro. Buttons can be easily added to start and monitor sequences.

Ami-Terminal Control

Ami-Terminal is a serial, USB or Ethernet client software that can connect and send simple commands to the V16Pro.

It can be launched from the "Tools" menu in WinScript Live.

All commands available to control the V16Pro are displayed with easy to use buttons. No additional setup in the Script is required.



Webpage Control

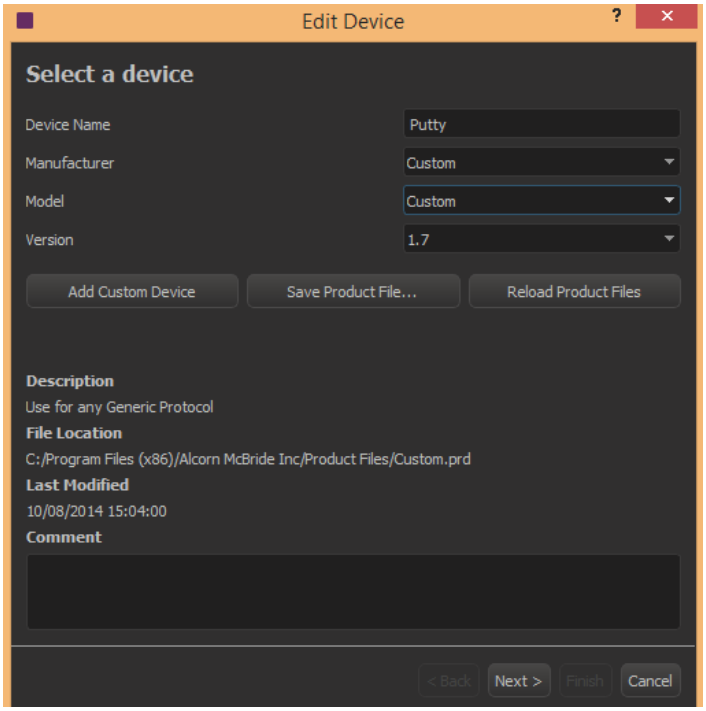
Sequences can be started using form buttons in HTML. Variable values can also be pulled into webpages. See the "WEB Server Quick Start" section of this manual for more details.

TCP Client (Telnet) Control

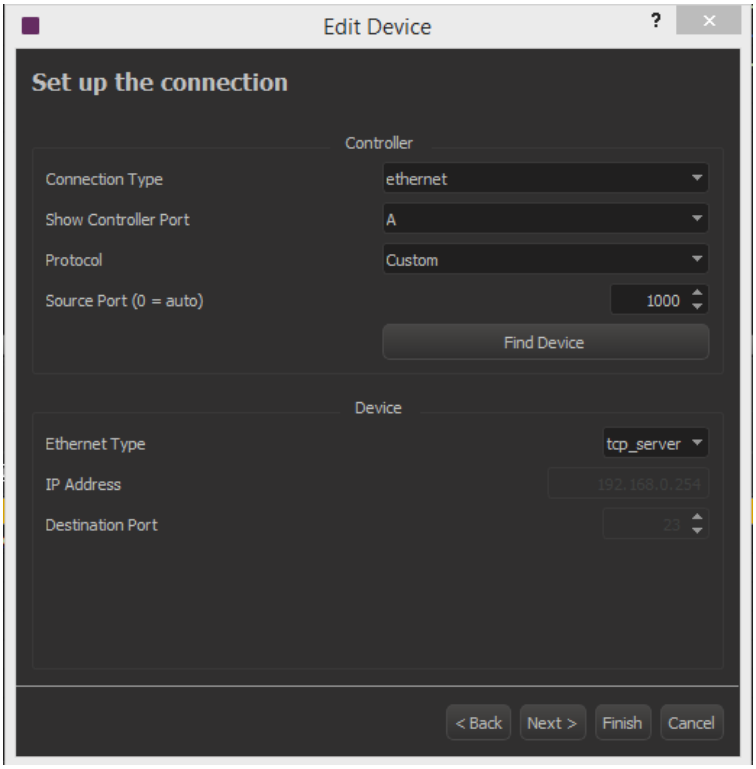
Any iPhone app capable of sending TCP or UDP messages can control the V16/V4Pro. For TCP, a server port must be setup in your devices list.

1: Setup TCP Server on Show Controller

1. Go to "Resources"→"Devices" from the menu bar and click "New"
2. Name the Device anything you'd like. In this example, we'll use "Putty"
3. Select the make, model and version as “Custom” and “Custom”



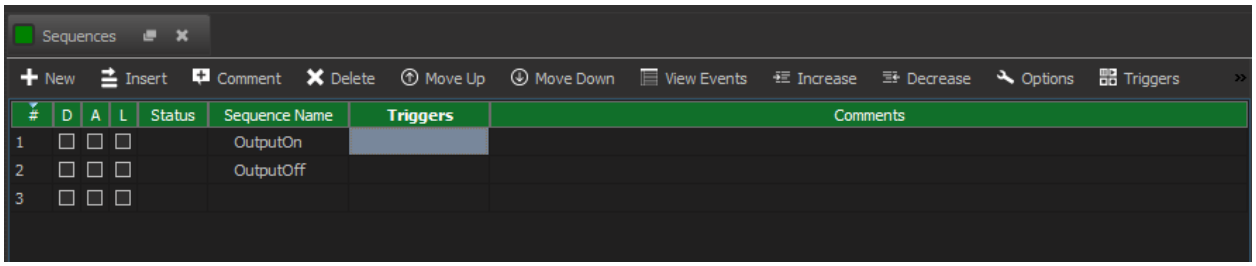
4. Choose the Ethernet connector (A or B) that you will be connecting do. Select the "TCP/IP Remote" protocol. Port "1000" is usually ok, but if you want a different port number, you can change it to anything you'd like. Just make sure it matches what you set up in the Putty app.



2: Add Incoming Message Triggers

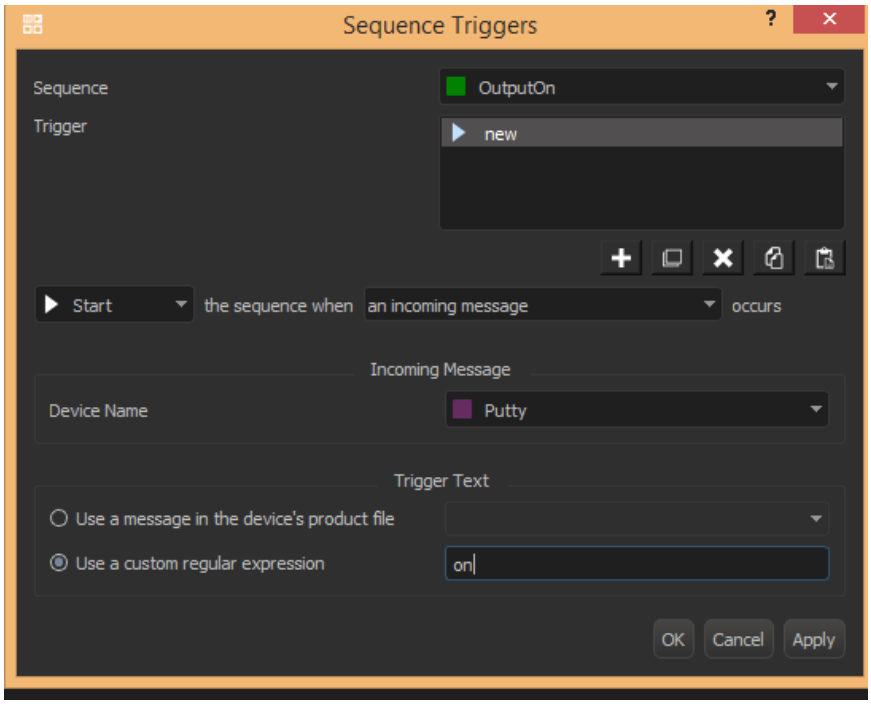
After you've created sequences that do what you'd like, you can add triggers to those sequences that allow the Putty to start the sequence.

In this example, I have two sequences named "OutputOn" and "OutputOff".



To allow these sequences to be started with an iPhone app:

1. Click on the "Triggers"
2. Click "yes" to open the triggers window.
3. Create a Trigger as shown below



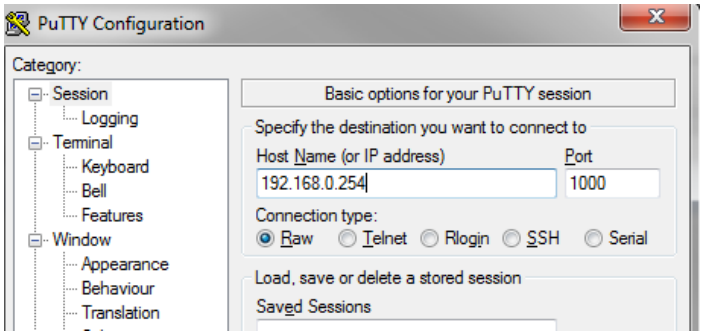
4. Repeat steps 1-8 for the "OutputOff" sequence. Use the text "off" for step 7.
5. Send the script to the V16Pro.

3: Control with Putty or TCP Client

Any TCP client can now sent the "triggers" strings that were setup in the above example.

Launch Putty.exe or other TCP Client and connect to the V16/V4Pro's IP Address using the port specified in the Device Setup. In our example, we used 1000.

Type "on" or "off" into the terminal window and press enter.



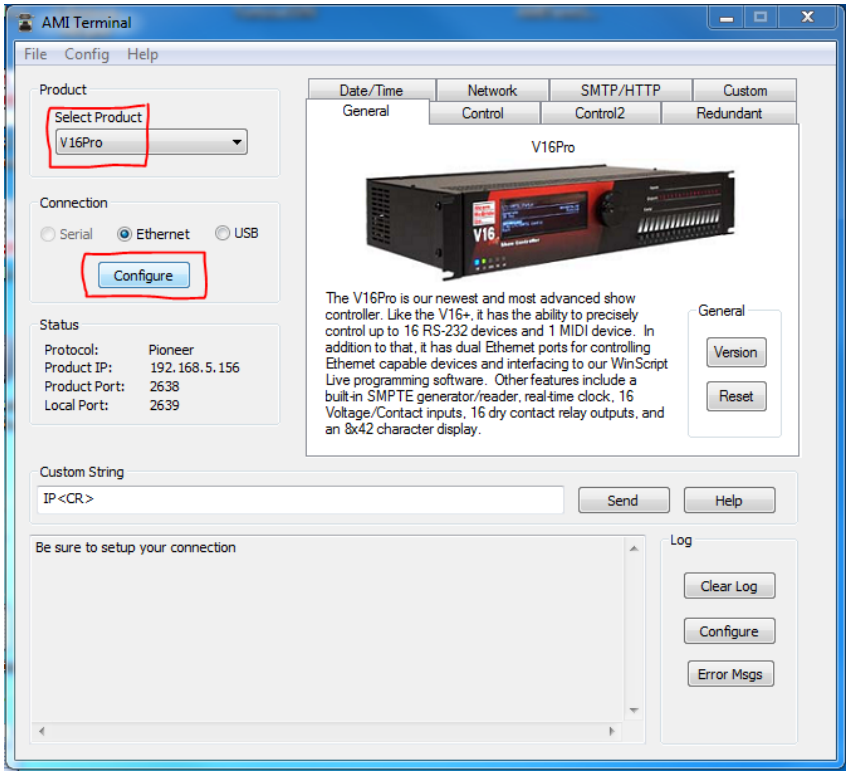
Redundant Mode

Redundant mode allows a 2nd "slave" controller to monitor the sequence status and follow along with the "master" controller should the "master" controller go offline. While we design our hardware to operate for years without a power cycle, and run our own operating system that is not susceptible to viruses, we understand that sometimes a redundant system is a requirement for our customers.

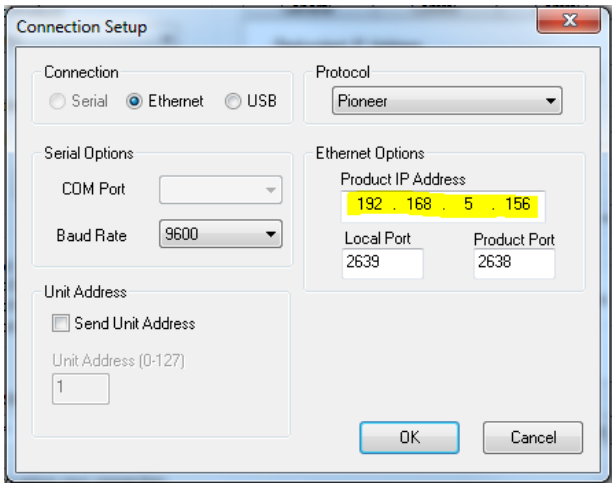
Redundant Mode Setup

To enable redundant mode, use AMI-Terminal to configure both V16Pro units:

1. Open AMI-Terminal from the Tools menu in WinScriptLive
2. Select "V16Pro" from the product list.

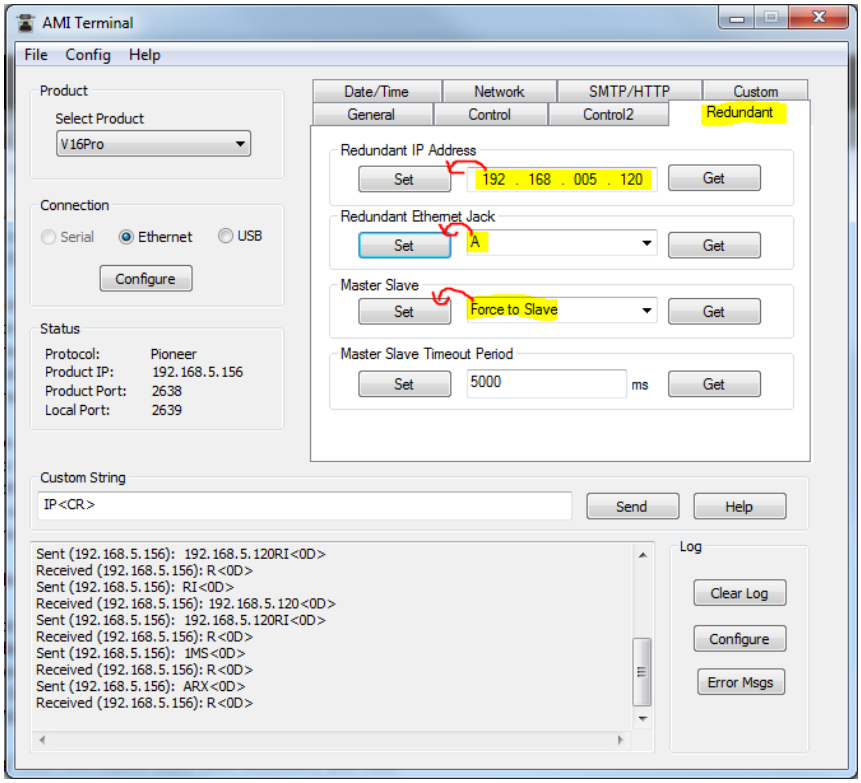


3. Click "Configure" and enter the IP address of the first V16Pro



4. Edit the "Redudant IP Address" to enter the 2nd V16Pro's address. Click "Set".
5. Set the Redundant Ethernet Jack to "A" or "B" (as connected)
6. In the "Master/Slave" box, select either "Force to be Slave" or “Force to be Master”

Note: in the end it will not matter if you "Force to be Slave" or "Force to be Master". Which ever V16Pro is first box to be powered on will be the master. The slave will automatically convert to master if the master "goes away."



7. Repeat steps 2-6 for the 2nd V16Pro

8. All sequence status is "shared" by default between master and slave. If certain variable status is critical, you must allow those variables to be "Watched" by checking the "Redundant Watched" checkbox in the variable's edit wizard.

   Notes: Make sure the script is the same in both V16Pro units. You can still send a script to either master or slave unit but you can only enter "Live Mode" while the unit is in "master" mode.

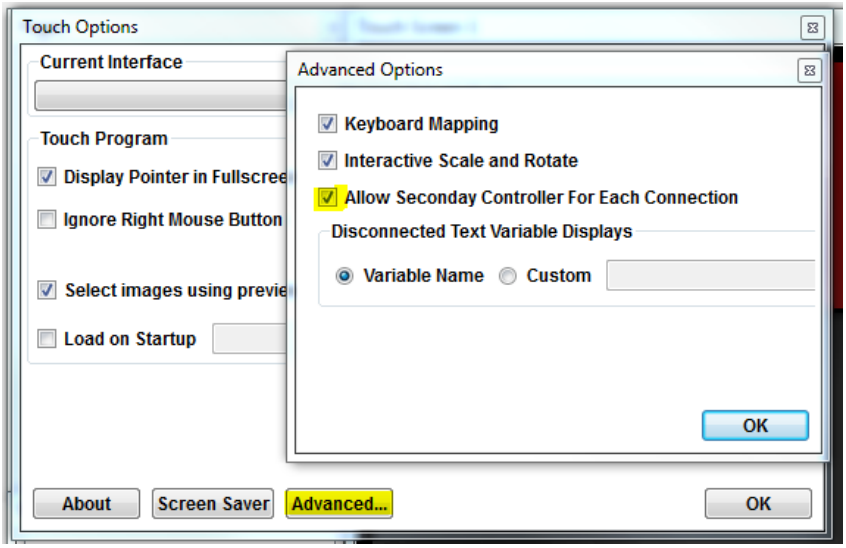
The front panel of the master will display the master/slave status and IP Address automatically on boot, but this can be overridden with any "Display" command in WinScriptLive. The slave will always show "Slave" on the front panel and will not take into account any WinScriptLive "Display" commands.

You can use the command "Set Slave" in WinScriptLive to try to force a particular box to become the slave controller by default. (For example on boot or on other external input). However, if a master does not exist, the slave will revert back to master after the timeout period. The timeout period can be adjusted in AMI-Terminal.

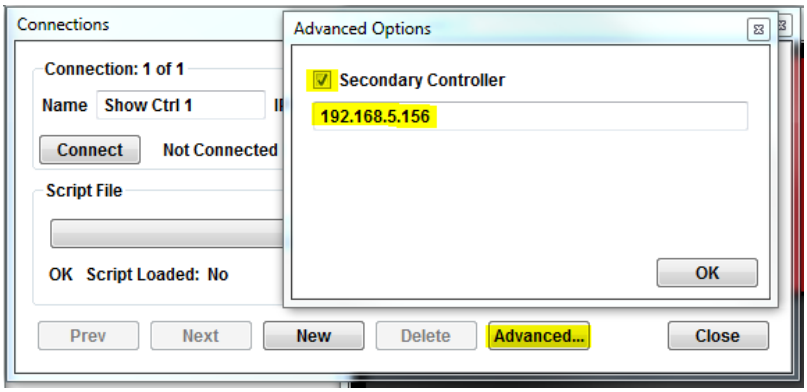
Redundant Mode in Touch Panels

Touch Panels cannot connect to the Slave. This is by design. The V16Pro.RedundantStatus variable will always show "Master" in Touch because otherwise it won't be connected. You can still use this variable in your Script if needed; just remember that Touch can only connect to the Master.

Before using Touch with Master/Slave show controllers, you must open the "Options" dialog and select "Advanced..." then check the box for "Allow Secondary Controller for Each Connection". You only need to do this one time and only on the computer you are using to create the Touch file. This is just to prevent other customers from seeing the secondary controller option without first understanding what it means.



To enable Touch to automatically attempt a connection to the secondary controller, open the "Show Controller" dialog and select "Advanced..." for each Connection you create. Since each Connection represents one Master/Slave pair, you will probably only have one Connection.



Check the box for "Secondary Controller" and enter the IP address. It doesn't matter which controller is the master or slave at the time, Touch will automatically switch between the two if it fails to communicate more than twice. You will see a message in the Touch Log that states when a switch is made between the Primary and Secondary controllers. All earlier versions of Touch will completely ignore this setting and will only use the primary IP address so make sure you are running Touch 2.06 or greater.

WinScript Live Command Reference

The events available within the show controller before adding any additional external devices are listed below.

Discrete Events

Discrete Events utilize discrete relay contact closures.

To Do This…	Use This Event…
Turn on an Output	***On***
Turn off an Output	***Off***
Toggle the state of an Output	***Toggle***
Continuously blink an Output at a constant rate	***Blink***
Pulse an Output for a user-defined length of time	***Pulse***
Set a group of eight Outputs to a binary value	***Out Port***
Read a group of eight Inputs to a Variable	***In Port***

On

Turns on an Output. The Output remains on until another event modifies its state.

Event Syntax

Event	Data1
On	Name of Output

Off

Turns off an Output. The Output remains off until another event modifies its state.

Event Syntax

Event	Data1
Off	Name of Output

Toggle

Toggles the state of an Output. If the Output is currently on, it will be turned off. If the Output is currently off, it will be turned on.

Event Syntax

Event	Data1
Toggle	Name of Output

Blink

Blinks an Output. Blinking an Output causes it to turn on (for the specified Blink Time) and off (for the specified Blink Time) continuously until reset by an **Off**, **On**, **Pulse**, **OutPort**, or **Toggle** event.

Event Syntax

Event	Data1	Data2
Blink	Name of Output	Blink Time*

**The Blink Time should be in Hours:Minutes:Seconds.Frames (e.g. 00:00.4.15) or a Timecode type variable*

Example

Event	Data1	Data2
Blink	*Output1*	*00:00.01.15*

Blinks Output1 with a Blink Time of 00:00:01.15 (one second, fifteen frames). This means that if Output1 is currently "off", it will turn on for 1.15 and then off for 1.15 repeatedly until reset by another Discrete Control event.

Pulse

Pulses an Output. If the Output is currently on, it will be turned off for the specified Pulse Length and then on again. If the Output is currently off, it will be turned on for the specified Pulse Length and then off again.

Event Syntax

Event	Data1	Data2
Pulse	Name of Output	Pulse Length*

**The Pulse Length should be in Hours:Minutes:Seconds.Frames (e.g. 00:00.4.15) or a Timecode type variable*

Example

Event	Data1	Data2
Pulse	*Output3*	*2.00*

Pulses Output3 (assume it is currently "off") with a Pulse Length of 2.00 (two seconds). This means that Output3 will turn on for 2.00 and then off again.

Out Port

Sets a group of eight Outputs to a single binary value. The lowest number Output becomes the Least Significant Bit (or LSB), the highest becomes the Most Significant Bit (or MSB).

Event Syntax

Event	Data1	Data2
OutPort	Output Bank*	Desired Literal Value (0-255)

**Bank1 = Outputs 1-8; Bank2 = Outputs 9-16; All= Outputs 1-16. If specifying "All" as the bank, the Literal Value range is 0-65535.*

Example

Event	Data1	Data2
OutPort	*Bank1*	*157*

Sets output bank 1 (Outputs 1-8) to the binary representation of 157 (or **10011101**). After the **Out Port** event is executed, the following outputs are actuated:

1	2	3	4	5	6	7	8
■		■	■	■			■

In Port

Reads a group of eight inputs to a Variable. The lowest number input becomes the LSB, etc.

Event Syntax

Event	Data1	Data2
InPort	Input Bank*	Name of Variable

**Bank1 = Inputs 1-8; Bank2 = Inputs 9-16; All = Inputs 1-16*

Example

Event	Data1	Data2
InPort	*Bank1*	*Var7*

Sets var7 to the value of input bank 1 (inputs 1-8). Assuming inputs of **10011101** (where 1 is on), after the **InPort** event is executed, Var7 will contain the value 157.

Logical Events

Logical Events perform operations on Variables in the Show Controller.

To Do This…	Use This Event…
Turn on a Boolean Type Variable	***On***
Turn off a Boolean Type Variable	***Off***
Toggle the state of a Boolean Type Variable	***Toggle***
Add a value to a Variable	***Add***
Subtract a value from a Variable	***Subtract***
Multiply Variable by a value	***Multiply***
Divide a Variable by a static value	***Divide***
Bitwise And a variable by a value	***BitAnd***
Bitwise Or a variable by a value	***BitOr***
Get the modulus value of a variable (remainder of division)	***Mod***
Concatenate (put together) two String Variables	***Concat***
Combine multiple variables of different types into a string variable	***Format***
Set the value of a Variable	***Set Variable =***
Save a Variable to non-volatile memory	***Save Variable***
Recover a Variable from non-volatile memory	***Restore Variable***

On

Turns on a Boolean Type Variable. The Boolean Type Variable remains on until another event modifies its state.

Event Syntax

Event	Data1
On	Name of Boolean Type Variable

Off

Turns off a Boolean Type Variable. The Boolean Type Variable remains off until another event modifies its state.

Event Syntax

Event	Data1
Off	Name of Boolean Type Variable

Toggle

Toggles the state of a Boolean Type Variable. If the Boolean Type Variable is currently on, it will be turned off. If the Boolean Type Variable is currently off, it will be turned on.

Event Syntax

Event	Data1
Toggle	Name of Boolean Type Variable

Add

Adds a value to a Variable. This value can be a constant value or another Variable. Acceptable variable types are Integer, Decimal, and Timecode. Timecode variables will be converted into number of frames.

Event Syntax

Event	Data1	Data2
Add	Name of Variable	Constant value <or> Name of another Variable

Subtract

Subtracts a value from a Variable. This value can be a constant value or another Variable. Acceptable variable types are Integer, Decimal, and Timecode. Timecode variables will be converted into number of frames.

Event Syntax

Event	Data1	Data2
Subtract	Name of Variable	Constant value <or> Name of another Variable

Divide

Divide a Variable by a value. This value can be a constant value or another Variable. Acceptable variable types are Integer, Decimal, and Timecode. Timecode variables will be converted into number of frames.

Event Syntax

Event	Data1	Data2
Divide	Name of Variable	Constant value <or> Name of another Variable

Multiply

Multiply a Variable by a value. This value can be a constant value or another Variable. Acceptable variable types are Integer, Decimal, and Timecode. Timecode variables will be converted into number of frames.

Event Syntax

Event	Data1	Data2
Multiply	Name of Variable	Constant value <or> Name of another Variable

BitAnd

Take the bitwise AND of a Variable and a value. This value can be a constant value or another Variable. Acceptable variable types are Integer and Timecode. Timecode variables will be converted into number of frames. For example, a Variable with a value of 3 (0011) and a contestant with a value of 1 (0001) would give a result of 1 (0001).

Event Syntax

Event	Data1	Data2
BitAnd	Name of Variable	Constant value <or> Name of another Variable

BitOr

Take the bitwise OR of a Variable and a value. This value can be a constant value or another Variable. Acceptable variable types are Integer and Timecode. Timecode variables will be converted into number of frames. . For example, a Variable with a value of 8 (1000) and a contestant with a value of 1 (0001) would give a result of 9 (1001).

Event Syntax

Event	Data1	Data2
BitOr	Name of Variable	Constant value <or> Name of another Variable

Mod

Multiply a Variable by a value. This value can be a constant value or another Variable. Acceptable variable types are Integer, Decimal, and Timecode. Timecode variables will be converted into number of frames.

Event Syntax

Event	Data1	Data2
Multiply	Name of Variable	Constant value <or> Name of another Variable

Concat

Concatenate a string-type variable with another variable or constant string value. The result is stored in the first variable.

Event Syntax

Event	Data1	Data2
Concat	Name of Variable	Constant value ie: "Hello" <or> Name of another Variable

Format

Create an ASCII string using other variables as inputs to the string. This is similar to the sprintf function found in C.

Event Syntax

Event	Data1	Data2	Data3…. DataN
Format	Name of Variable to store result	Format String: "hello %s %d"	Variable Name

Data2's "Format String" is made up of placeholders that indicate where and how the variables in Data3-DataN should be inserted. The table below shows what % character should be used to format the string. In general, %s will work for most cases.

Formatted String

Variable Type to Insert	%	Description
String, Display String	%s	Inserts the text in string format. Hex characters are converted into ASCII printable characters like "h0D".
Timecode	%s	Inserts time in the format "00:00:02.01"
Timecode	%d	Inserts the number of frames – ie: 61
Integer	%d	Inserts number without leading zeros or spaces. To add leading zeros, use %0nD where "0" indicates to pad with zeros and "n" is number of zeros to pad. For example: %03d will insert the number 2 as 002.
Integer	%X	Inserts the number as a hex string. For example, the number 11 would be inserted as 0B.
Integer	%p	No conversion to ASCII will be made, and the number will be placed in the string. For example, the number 13 (hex 0D) will be inserted as a Hard Return.
Decimal	%f	Inserts one decimal place OR number of places specified after "." preceding "f". For number 1.12345," %.1f" inserts 1.1. "%.2f" inserts 1.12. "%.4f" inserts 1.1234.
Boolean	%s	"true" or "false" will be printed
Boolean	%d	"1" or "0" will be printed
Date/Time	%s	Inserts in the format: month/day/year hours:minutes
Percent	%s	Inserts string including % character. ie: 100%
Percent	%f	Inserts decimal percentage, shows 25% as .25

Set Variable =

Sets the value of a Variable to a constant value or to the value of another Variable. If they are different variable types, the will be converted. Decimal values are truncated when set to Integer types. Timecode variables will convert to number of frames for Integer types.

Event Syntax

Event	Data1	Data2
Set Variable =	Name of Variable	Constant value <or> Name of another Variable

Save Variable

Stores the value of a Variable in non-volatile memory so that it can be recovered, even after power cycling, using Restore Variable.

Note: A maximum of 256 variables with a maximum string length of 256 characters for each variable can be stored. Variables are stored using their string name.

Event	Data1
SaveVar	Name of Variable

Restore Variable

Recovers the value of a Variable from non-volatile memory.

Event	Data1
RestoreVar	Name of Variable

Program Control Events

Program Control Events can be used to control the flow of your show. Program Control Events include events for controlling Sequences in a local or remote Show Controller. Events are also included for performing conditional branching within a sequence based on Variable values and/or Input/Output/Boolean Type Variable states.

To Do This…	Use This Event…
Start a Sequence	***Start***
Stop a Sequence	***Reset***
Pause a Sequence at the current event	***Pause***
Stop a looping Sequence after the last event	***Stop Loop***
Unconditionally jump over events	***Goto***
Perform specific events if an Input, Output, or Boolean Variable is "on"	***If On***
Perform specific events if an Input, Output, or Boolean Variable is "off"	***If Off***
Perform specific events if a Variable is equal to a constant value or the value of another Variable	***If =***
Perform specific events if a Variable is greater than a constant value or the value of another Variable	***If >***
Perform specific events if a Variable is greater than or equal to a constant value or the value of another Variable	***If >=***
Perform specific events if a Variable is less than a constant value or the value of another Variable	***If <***
Perform specific events if a Variable is less than or equal to a constant value or the value of another Variable	***If <=***
Perform specific events if a Variable is not equal to a constant value or the value of another Variable	***If not =***
Complete "If" sections. (If not using labels)	***End If***
Specify "False" events if statement is false	***Else***
No operation - Set a dummy placeholder for a branch event	***Empty***

Start

Starts a sequence. If the sequence is not currently running and was never paused in the middle by a **Reset** event, the sequence will begin execution at the first event. If the sequence started *was* running and is now paused by some other sequence, the sequence started will resume execution at the event. If the sequence started is currently running and the setup for the sequence has Restart Enabled, the sequence will stop event execution and restart execution from the first event. If the sequence started is currently running and does not have Restart Enabled, it will continue running as it was and the start event will be ignored.

Event Syntax

Event	Data1
Start	Sequence Name

Pause

Pauses a sequence at the current event. A Start event will cause the sequence to resume from the point at which it was stopped.

Event Syntax

Event	Data1
Pause	Sequence Name

Stop Loop

Causes a looping sequence to stop looping after the last event. If the sequence is restarted, it starts execution from the first event.

Event Syntax

Event	Data1
Stop Loop	*Sequence Name*

Reset

Stops a sequence immediately. If the sequence is restarted, it starts execution from the first event.

Event Syntax

Event	Data1
Reset	Sequence Name

Goto

Unconditionally jumps over events. Forward AND backward jumps are allowed. In the case of backward jumps, the sequence will wait 1 frame after jumping backward before executing event.

Note A branch event causes no change in time within the sequence; all events occur based on time from sequence start.

Event Syntax

Label	Time	Event	Data1	Data2
	00:00.00	*Goto*	Event Label	

	00:00.00	*Skipped*	*Events*	
Event Label	*00:00.00*	Some Event		

Example

Label	Time	Event	Data1	Data2	Data3
	00:00.00	*If =*	*ShowVar*	*1*	*RunShow1*
	00:00.00	*If =*	*ShowVar*	*2*	*RunShow2*
	00:00.00	***Goto***	*End*		
RunShow1	*00:00.00*	*Start*	*GoShow1*		
	00:00.00	***Goto***	*End*		
RunShow2	*00:00.00*	*Start*	*GoShow2*		
End	*00:00.00*	*Empty*			

If ShowVar is not a valid number, the first Goto is reached and the sequence jumps to the end and performs no action. If ShowVar is equal to 1, "Show 1" is started, then the second Goto event causes the sequence to jump over the "Show 2" events.

If On, If Off

Conditionally performs specific events based on the state of an Input, Output, or Boolean Type Variable. If an *Event Label* is used, events are jumped over if the condition is true. If an *Else* or *End If* event is used, events are executed inside the area between the *If* and the *End If* if the condition is true.

- **If On** – Is true if an Input, Output, or Boolean Type Variable is "on".
- **If Off** – Is true if an Input, Output, or Boolean Type Variable is "off".

Note A branch event causes no change in time within the sequence; all events occur based on time from sequence start.

Event Syntax Using "End If"

Time	Event	Data1
00:00.00	Event Name	Name of Input, Output, or Boolean Type Variable
00:00.00	*Executed if condition is true*	*Events*
00:00.00	End If	

Event Syntax Using "Label"

Label	Time	Event	Data1	Data2
	00:00.00	Event Name	Name of Input, Output, or Boolean Type Variable	Event Label
	00:00.00	*Skipped if condition is true*	*Events*	
Event Label	*00:00.00*	Some Event		

Example using "End If"

Time	Event	Data1
00:00.00	***If Off***	*NightMode Boolean Type Variable*
00:00.00	*Play*	*Ldp1*
00:00.00	*End If*	

The Play event is skipped if the system is in Night Mode.

Example using "Label"

Label	Time	Event	Data1	Data2
	00:00.00	***If On***	*NightMode Boolean Type Variable*	*End*
	00:00.00	*Play*	*Ldp1*	
End	*00:00.00*	*Empty*		

The Play event is skipped if the system is in Night Mode.

If =, If not =, If >, If >=, If <, If <=

Conditionally executes events based on the value of a Variable. . If an *Event Label* is used, events are jumped over if the condition is true. If an *Else* or *End If* event is used, events are executed inside the area between the *If* and the *End If* if the condition is true.

- **If =** -- Is true if the value of a Variable is equal to a constant value (0-255) or the value of another Variable.
- **If not =** -- Is true if the value of a Variable is not equal to a constant value (0-255) or the value of another Variable.
- **If >** -- Is true if the value of a Variable is greater than a constant value (0-255) or the value of another Variable.
- **If >=** -- Is true if the value of a Variable is greater than or equal to a constant value (0-255) or the value of another Variable.
- **If <=** -- Is true if the value of a Variable is less than or equal to a constant value (0-255) or the value of another Variable.
- **If <** -- Is true if the value of a Variable is less than equal to a constant value (0-255) or the value of another Variable.

Event Syntax using "End If"

Time	Event	Data1	Data2
00:00.00	Event Name	Variable	Constant value (0-255) <or> another Variable
00:00.00	*Executed Events*	*Events*	
00:00.00	End If		

Example #1

Time	Event	Data1	Data2
00:00.00	***If >=***	*ShowVar*	*5*
00:00.00	*Play*	*Ldp1*	

00:00.00	***End If***		

The Play event is executed if ShowVar ≥ 5.

Example #2

Time	Event	Data1	Data2
00:00.00	*Add*	*ShowVar*	*1*
00:00.00	***If <=***	*ShowVar*	*100*
00:00.00	*Set Variable =*	*ShowVar*	*0*
00:00.00	***End If***		

This sequence adds one to ShowVar and then sets it back to 0 if it greater than 100.

Event Syntax using "Label"

Label	Time	Event	Data1	Data2	Data3
	00:00.00	Event Name	Variable	Constant value (0-255) <or> another Variable	Event Label
	00:00.00	*Skipped*	*Events*		
Event Label	*00:00.00*	*SomeEvent*			

End If

Used to mark the End of any of the conditional "If" commands listed on the previous pages. This is only required when **not** using **Labels**.

Event Syntax Using "End If"

Time	Event	Data1
00:00.00	Conditional Event Name	Conditional Variable or Input
00:00.00	*Executed if condition is true*	*Events*
00:00.00	End If	

Else

Used to mark the **False** case of any of the conditional "If" commands listed on the previous pages. This can only be used when **not** using **Labels**.

Event Syntax

Time	Event	Data1
00:00.00	Conditional Event Name	Conditional Variable or Input
00:00.00	*Executed if condition is true*	*Events*
00:00.00	**Else**	
00:00.00	*Executed if condition is false*	*Events*
00:00.00	End If	

Empty (formerly Nop)

Used as a branch placeholder.

Event Syntax

Event	Data1
Empty	

Example

Label	Time	Event	Data1	Data2	Data3
	00:00.00	*IfVarEQ*	*ShowVar*	*5*	*End*
	00:00.00	*Play*	*Ldp1*		
End	*00:00.00*	***Empty***			

Display Events

Display Events display custom text messages as well as Boolean Type Variable and Variable states on the LCD.

To Do This…	Use This Event…
Display a custom message on the VFD	***Display***
Store the currently displayed message	***Store Display***
Retrieve and display a previously stored VFD message	***Recover Display***

Display

Displays a custom message on the Display.

Event Syntax – With Variables

Event	Data1	Data2	Data3
Display	Name of Display String <or> Literal Message*	Row (optional)	Column (optional)

Event Syntax – With Variables

Event	Data1	Data2	Data3-DataN
Display	Name of Display String <or> Literal Message*	Variable to Insert (optional)	Variable to Insert (optional)

Tip Using the "Display Wizard" located under the "Event Wizard" or "Variable Wizard" can help with formulating display strings.

Display Strings can be formulated with variable "placeholders." The correct % placeholder to use depends on the type of variable being inserted. The table below shows the placeholders and their description.

Formatting a Display String

Variable Type to Insert	%	Description
String, Display String	%s	Inserts the text in string format. Hex characters are converted into ASCII printable characters like "h0D".
Timecode	%s	Inserts time in the format "00:00:02.01"
Timecode	%d	Inserts the number of frames – ie: 61
Integer	%d	Inserts number without leading zeros or spaces. To add leading zeros, use %0nD where "0" indicates to pad with zeros and "n" is number of zeros to pad. For example: %03d will insert the number 2 as 002.
Integer	%X	Inserts the number as a hex string. For example, the number 11 would be inserted as 0B.
Integer	%p	No conversion to ASCII will be made, and the number will be placed in the string. For example, the number 13 (hex 0D) will be inserted as a Hard Return.

Decimal	%f	Inserts one decimal place OR number of places specified after "." preceding "f". For number 1.12345," %.1f" inserts 1.1. "%.2f" inserts 1.12. "%.4f" inserts 1.1234.
Boolean	%s	"true" or "false" will be printed
Boolean	%d	"1" or "0" will be printed
Date/Time	%s	Inserts in the format: month/day/year hours:minutes
Percent	%s	Inserts string including % character. ie: 100%
Percent	%f	Inserts decimal percentage, shows 25% as .25

Display strings can also indicate lines to print by using commas outside of the quotes. For example:

"1","2","3","4" will print on the Display:

1
2
3
4

Using they keyword "clr" without quotes has a special meaning. It will clear the line. For example:

"1",clr,"3",clr will print on the Display:

1
3

Where the 2nd and 4th lines are cleared.

Store Display

Stores both lines of text currently displayed on the LCD. Text may be recovered at any time by using Recover Display.

Event Syntax

Event	Data1
StoreLCD	

Recover Display

Re-displays both lines of text previously stored by Store Display. If no text was previously stored, the Show Controller version number is displayed.

Event Syntax

Event	Data1
RecoverLCD	

Timecode (LTC, SMPTE, EBU) and Internal Time Events

Timecode (LTC, SMPTE, EBU) related events perform function on the single, global timecode source for unit. Other time related events such as "Get Seq Time" and "Delay" reference a specific sequence clock.

To Do This…	Use This Event…
Wait a specific amount of time in a non-timed sequence	***Delay***
Sett SMPTE/EBU Timecode to a specific time	***Timecode Set***
Start SMPTE/EBU Timecode Running	***Timecode Start***
Stop SMPTE/EBU Timecode	***Timecode Stop***
Pause SMPTE/EBU Timecode	***Timecode Pause***
Stop SMPTE/EBU Timecode at next loop point	***Timecode Stop Loop***
Get a current sequence time (internal clock) in a variable	***Get Seq Time***
Allow SMPTE/EBU Timecode jam-synced sequence to be run.	***Arm***
Prevent a SMPTE/EBU Timecode jam-synced sequence from being run	***Disarm***

Delay

Causes the running sequence to delay of a specific amount of time. For use sequences where the timecode is set to be "None."

Event Syntax

Event	Data1
Delay	Name of Timecode Variable <or> Timecode in the form 00:00:00.00

Timecode Set

Set the SMPTE/EBU Timecode to a specific value.

Event Syntax

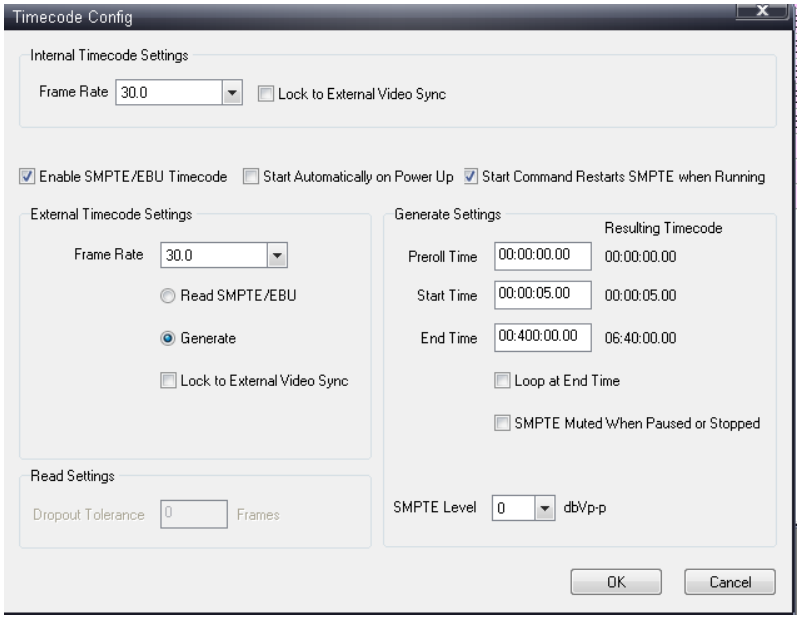
Event	Data1
Timecode Set	Name of Timecode Variable <or> Timecode in the form 00:00:00.00

Timecode Pause

Pauses the SMPTE/EBU Timecode. Can be resumed with Timecode Start.

Timecode Start

Starts the timecode running at its current location if it's paused. If it was previously stopped, the timecode starts running at its Start Time specified in the Timecode Config window (shown below). This window can be accessed from the "Devices" screen.



Timecode Stop

Stop and Reset the current SMPTE/EBU Timecode.

Timecode Stop Loop

Stop the SMPTE/EBU Timecode at the next loop point if looping. Loop settings can be specified using the "Timecode Config" dialog (see above).

Get Seq Time

Places the current sequence's internal clock time into a timecode-type variable defined by the user.

Event Syntax

Event	Data1	Data2
Get Seq Time	Name of Sequence	Name of Timecode-Type variable

Arm

Arm is similar to a sequence "Start" command. Setting a sequence "armed" allows current SMPTE/EBU timecode to execute a sequence's events as the timecode runs. This is used when the Clock Source is set to SMPTE/EBU (see image below).

Disarm

Disarm is similar to a sequence "Reset" command. The Sequence will ignore current SMPTE/EBU timecode when in disarmed state. This is used when the Clock Source is set to SMPTE/EBU (see image above).

Network Events

Send Mail

Send e-mail to a specific address using the SMTP settings defined using front panel or Terminal interface. Currently SSL is not supported.

Event Syntax

Event	Data1	Data2	Data3
Send Mail	The email address to send the email to. This value must be enclosed with < and > such as <nameto@email.com>	The subject of the email message in quotes	The body of the email message in quotes

Number Generation

Get Random

Get a random integer number and save it to a variable

Event Syntax

Event	Data1	Data2	Data3
Get Random	A variable name	The minimum value for the random number (optional: default is variable's minimum)	The maximum value for the random number (optional: default is variable's maximum)

External Communication Events

Built-In Events send a custom message to a specific Ethernet or serial device.

To Do This…	Use This Event…
Control a custom device	***Message Out***
Create a custom string to send	***Format***
Send a message to the "Live Log"	***Log Message***

Message Out

Sends a custom serial message out one of the serial ports. The message is sent in the protocol defined for that port, but the Show Controller will not wait for an ACK or other response unless a TCP connection is required.

Event Syntax

Event	Data1	Data2
Message Out	Name of Port	Name of String Variable or Literal Message

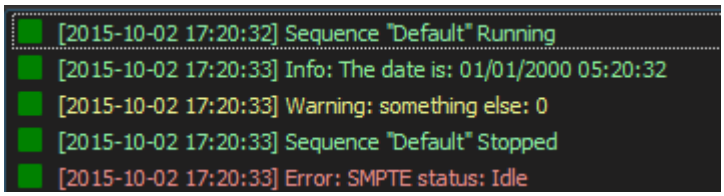
Example #1

Event	Data1	Data2
Message Out	Port3	DataMsg

Sends (where h represents hex character):
Hello h0D
If DataMsg is:
"Hello\r"

Log Message

Send a message to appear in the "Live Log" in WinScriptLive.



A log message has a priority of "Information", "Warning" or "Error".

Optionally, variables may be used as part of the string the string. This is similar to the sprintf function found in C.

Event Syntax

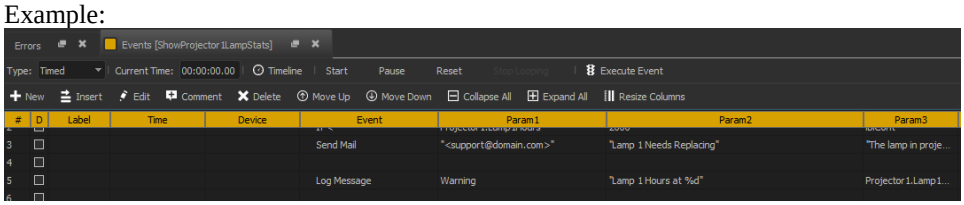
Event	Data1	Data2	Data3…. DataN
Log Message	Priority of the message	Format String: "hello %s %d"	Variable Name

Data2's "Format String" is made up of placeholders that indicate where and how the variables in Data3-DataN should be inserted. The table below shows what % character should be used to format the string. In general, %s will work for most cases.

Formatted String

Variable Type to Insert	%	Description
String, Display String	%s	Inserts the text in string format. Hex characters are converted into ASCII printable characters like "h0D".
Timecode	%s	Inserts time in the format "00:00:02.01"
Timecode	%d	Inserts the number of frames – ie: 61
Integer	%d	Inserts number without leading zeros or spaces. To add leading zeros, use %0nD where "0" indicates to pad with zeros and "n" is number of zeros to pad. For example: %03d will insert the number 2 as 002.
Integer	%X	Inserts the number as a hex string. For example, the number 11 would be inserted as 0B.
Integer	%p	No conversion to ASCII will be made, and the number will be placed in the string. For example, the number 13 (hex 0D) will be inserted as a Hard Return.
Decimal	%f	Inserts one decimal place OR number of places specified after "." preceding "f". For number 1.12345," %.1f" inserts 1.1. "%.2f" inserts 1.12. "%.4f" inserts 1.1234.
Boolean	%s	"true" or "false" will be printed
Boolean	%d	"1" or "0" will be printed
Date/Time	%s	Inserts in the format: month/day/year hours:minutes
Percent	%s	Inserts string including % character. ie: 100%
Percent	%f	Inserts decimal percentage, shows 25% as .25

Example:



Displays a warning message including lamp hours

V16Pro

Figure 1- Front View

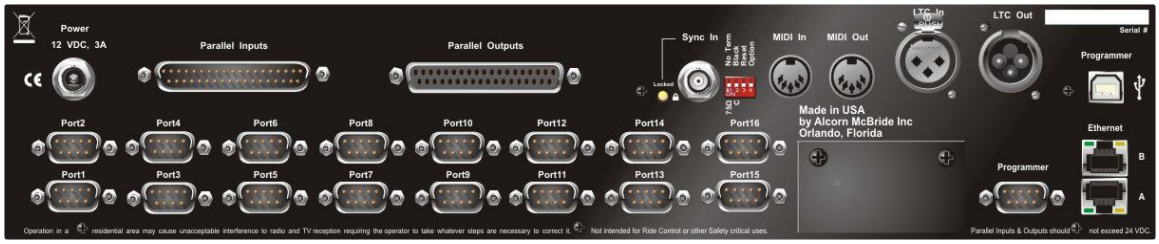


Figure 2 - Rear View

Specifications

Size and Weight:	Standard 2U rack mount (3.5" x 17" x 10.5"), 10 lbs
Power:	12 VDC at 3 amps. An external power supply is included with the V16Pro and will provide the required input power. The power supply is rating is 100 to 250 VAC, 50 to 60 Hz, 25-watts maximum. UL listed Class 2 power adapter
Environment:	0 to 38 C (32 to 100 F), 0 to 90% relative humidity, non-condensing
Front Panel:	8x40 VFD Display Power LED LTC (SMPTE) LED Vsync LED Error LED Acknowledge (ACK) LED 16 Serial Activity LEDs 16 Input Status LEDs 16 Output Status LEDs 16 Pushbuttons
Rear Panel:	Programming Port DB-9M 16 Serial Ports DB-9M MIDI Input 5-pin DIN Female MIDI Output 5-pin DIN Female Discrete Inputs DB-37M

	Discrete Outputs DB-37F NTSC or PAL Sync Input BNC Power barrel jack LTC Input 3-pin XLR Female LTC Output 3-pin XLR Male Ethernet Jack A Ethernet Jack B
Serial Ports:	(16) RS-232C, RS-422/485, individually software configurable 300 baud – 115.2 Kbaud 7, 8, or 9 Data Bits 1 or 2 Stop Bits All parity types
MIDI	MIDI input and output ports
Opto Inputs:	(16) Each input is software configurable for voltage or contact-closure operation. Input voltage range is 5-24VDC. Misconfiguration or reverse polarities will not damage inputs. Trigger latency < 1 frame (33.3ms @ 30 fps).
Relay Outputs:	(16) Contact Closures limited internally to 900 mA with self-restoring polymer fuses.
Show Memory:	Removable Compact Flash card allows scripts with thousands of events. Maximum size per show is 5MB. Multiple shows can be loaded per card.

Certifications

EMC Compliance: US, Canada and Europe (CE Mark)

Emissions Compliance:

EN 55103-1:2009, Electromagnetic compatibility (emissions). Product family standard for audio, video, audio-visual and entertainment lighting control apparatus for professional use.

Formal Emissions Compliance, Information Technology Equipment, EN 55022:2010 (EU/AUST), FCC CFR 47

Part 15 (US), ICES-003 (Canada), VCCI V-3 (Japan) Class B Emissions.

- Radiated and Conducted emissions

- Include Telecommunications Port

Formal Emissions Compliance, Radiated Magnetic requirements. - 100mm, 50 Hz to 50KHz

In rush Current : Annex B

EN 61000-3-2: Limits for Harmonic Current Emissions

EN 61000-3-3, Limitation of Voltage Fluctuations and Flicker

Immunity Compliance:

EN 55103-2:2009, Electromagnetic compatibility (Imuunity). Product family standard for audio, video, audio-visual and entertainment lighting control apparatus for professional use.

EN 61000-4-2, Electrostatic Discharge, Immunity Compliance

EN 61000-4-3, Radiated Electromagnetic Fields, Immunity Compliance - 80 MHz to 2.7 GHz

EN 61000-4-4, Electrical Fast Transient / Burst, Immunity Compliance

EN 61000-4-5, Surge, Immunity Compliance

EN 61000-4-6, Conducted Immunity Compliance

EN 61000-4-8, Magnetic Field Immunity Compliance - Annex A: 50 Hz to 10 kHz

EN 61000-4-11, Voltage Dips and Variations - Audio Frequency: Annex B

LTC Ports

The LTC ports provide SMPTE/EBU timecode IN for reading and timecode OUT while generating.

Male XLR (OUT)

Pin	
1	*GND*
2	+
3	-

Female XLR (IN)

Pin	
1	*GND*
2	+
3	-

Serial Ports

The V16Pro provides many serial ports used for programming and controlling show related machines. The serial interfaces are RS-232, RS422/485, MIDI, USB and Ethernet giving the V16Pro greater control flexibility for all system and show control functions.

RS-232/422/485 Ports

A few words on the RS-232C port differences. As shown in the table below, take note of the differences between the programmer port and the show control ports 1-16. Programmer cables cannot be used as show control cables and visa versa even if they are RS-232 only.

1. The programmer port is RS-232C only while the show control ports 1-16 have RS-232C as well as RS422/485.

2. Also pins 2 and 3 are reversed between the programmer port and the show control ports 1-16.
3. There are additional pins on the show control ports 1-16 that are not present on the programmer port. The additional pins on these ports are for supporting the RS-422/485 functions. Note: The V16Pro RS422 TXD signals are reversed from the V16+
4. Finally, the +9-volt pull up supply is on pin 8 of the programmer port and on pins 4 and 7 on the show control ports 1-16.

Differences at a Glance

	(2 and 3 reversed)	(Same pin out as the PC)	
Pin	**Programmer port**	**Ports 1-16 RS-232**	**Ports 1-16 RS-485**
2	*RS-232 TXD*	*RS-232 RXD*	*RS-422/485 RX-*
3	*RS-232 RXD*	*RS-232 TXD*	*RS-422/485 TX-*
4		*+9V Pull Up*	*+9V Pull Up*
5	*GND*	*GND*	*GND*
6		*Do Not Connect*	*RS-422/485 RX+*
7		*+9V Pull Up*	*+9V Pull Up*
8	*+9V Pull Up*		
9		*Do Not Connect*	*RS-422/485 TX+*

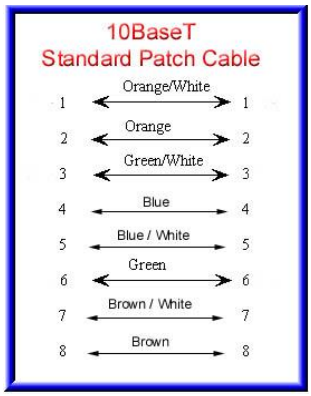
Ethernet Ports

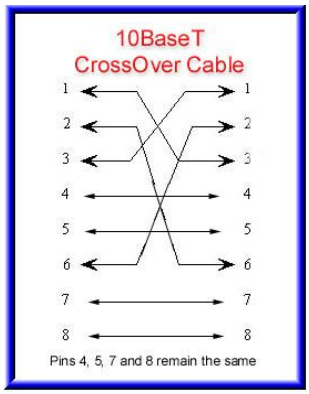
The 2 Ethernet ports on the V16Pro do not have data pass-thru capability and cannot be used as hubs, switches or routers. Both ports can be show or programmer ports. If your control is assigned to one of the other programmer ports, you may have two independent show control network ports if needed. For additional information on configuring and using Ethernet see the sections on Ethernet and Networking Primer at the end of this manual.

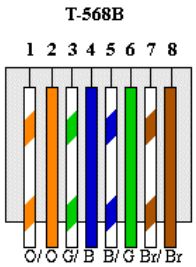
There are many protocols associated with networking, as applied to the V16Pro; the ones of interest are UDP, SNMP, NTP, TCP, SMTP, HTTP and FTP. Because these ports are true Ethernet ports in every way, there is nothing in the hardware that prevents these ports to be swapped.

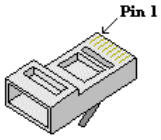
Ethernet Cables

There are two types of Ethernet cables, they are known as patch cables and crossover cables. The connector is auto sensing so either one will work with the V16Pro.


10BaseT
Standard Patch Cable
Orange/White
Orange
Green/White
Blue
Blue / White
Green
Brown / White
Brown
1
2
3
4
5
6
7
8


10BaseT
CrossOver Cable
1
2
3
4
5
6
7
8
Pins 4, 5, 7 and 8 remain the same


T-568B
1 2 3 4 5 6 7 8
O/ O G/ B B/ G Br/ Br


Pin 1

RJ-45 Plug

Programmer Ports

The V16Pro has three ports that can be used to programming. RS-232 Programmer port, USB or Ethernet port may be used for programming, control and monitoring.

RS-232C

The V16Pro can be connected to a PC using a cable that is wired as one-to-one. This means pin 2 is wired to pin 2, 3 to 3 and 5 to 5 thru the cable. The connector type is a 9-pin D-sub with female pins at both ends of the cable. The V16Pro port pin out is listed below:

Pin	Connection
2	*RS-232 TXD*
3	*RS-232 RXD*
5	*GND*
8	*+9V Pull Up*

Table 2 – Programmer port connections.

USB

The USB Port is used for programming and Live Mode only. The driver can be installed with WinScriptLive setup file.

Ethernet Ports A and B

The optimal way to program the V16Pro is to use the Ethernet ports. There are two Ethernet ports labeled A and B. Ethernet port A or B may be used for V16Pro programming control and status monitoring.

Show Control Ports

The Show Control ports are used to control other devices used to control show devices. These devices include, for example, video and audio playback machines, lighting controls and gates, doors and curtains and any other show related machines.

Ports 1-16: RS-232 or RS-422/485

Ports 1-16 are configured as RS-232 or RS422/485 by software. The V16PRO requires an external 220-Ohm termination for RS-485. When using RS-232C, take note of the "Do Not Connect" pins.

Pin	RS-232 Connection	RS-485 Connection
2	*RS-232 RXD*	*RS-422/485 RX-*
3	*RS-232 TXD*	*RS-422/485 TX-*
4	*+9V Pull up*	*+9V Pull up*
5	*GND*	*GND*
6	*Do not connect*	*RS-422/485 RX+*
7	*+9V Pull up*	*+9V Pull up*
9	*Do not connect*	*RS-422/485 TX+*

Table 3 – Ports 1-16 connections for RS-232 or RS-422/485 operation.

MIDI Ports

There are two MIDI ports on the V16Pro MIDI-IN and MIDI-OUT. The MIDI IN port receives MIDI Input, and MIDI Output is sent out the MIDI OUT port.

MIDI IN

Pin	Connection
4	*MIDI RX+*
5	*MIDI RX-*

Table 5 – MIDI IN connections.

MIDI OUT

Pin	Connection
2	*GND*
4	*MIDI TX+*
5	*MIDI TX-*

Table 6 – MIDI OUT connections.

Ethernet Ports A and B

There are two Ethernet ports as mentioned before and are named A and B. If the Ethernet ports are not being used to configure the V16Pro, you have the option of controlling two isolated show networks. This can be an advantage when IP

address conflicts arise or when testing show configurations apart from an operating show.

You may control a networks using one of the Ethernet ports located on the backside of the unit in the lower right corner as viewed from the back. Each port has a different IP address and may be changed by the front panel menu wheel control.

Display

The V16Pro includes a standard 8x40 Backlit VFD Display. When the V16Pro is in startup or configuration mode, the display will show setup parameter menus and feedback controlled by the menu wheel. When in the show control mode, the display may be configured to give up to 8 lines of text as needed. WinScript Live will provide the show producer with simple but powerful control of what is shown in the display.

Menu Wheel

The Menu Wheel provides easy access to all of the configurable parameters of the V16Pro. By rotating the wheel to the right or to left will cause menus will scroll forward or backward. Pressing the menu wheel will select the displayed item.
The menu map shown below shows how to get to each menu item

Menu Map

The Menu Map is provided so an understanding of the
The menu map will change with updates as we add functions, with that in mind use the map as a general guide and consult the update documents for details of additions and changes.
The Menu Wheel gives the user access to many of the configurable operating parameters. The highlighted parameter may be selected by pressing the wheel. A lower level menu will be visible or the line of the parameter to be altered. The Exit item, at the end of every menu, will pop the map up one level or out of the menu system and back to normal operation when at the top menu.

Main Menu
- System
- Firmware Version:
- SMPTE Version:
- Hardware Version:
- Percent of Frame used by Process:
- Serial Number:
- Exit
- Real-Time Clock
- Time:
- Date:
- NTP Enabled:
- NTP Errors:
- RTC Config:
- Exit

LTC/SMPTE

- Mode . . . SMPTE Time Display
- Current Frame Time . . . SMPTE Status
- Stop, Start SMPTE:
- LTC/SMPTE Config:
- Exit
- Network
  - Network Adapter A
    - IP Address: 192.168.000.254
    - Subnet Mask: 255.255.255.000
    - Gateway: 192.168.000.001
  - Network Adapter B
    - IP Address: 192.168.000.253
    - Subnet Mask: 255.255.255.000
    - Gateway: 192.168.000.001
  - Exit
- Password
  - Enter Password
- Script Configuration
  - Default Script, Reload and View Watches.
- Exit

Function Description

The menu definitions are described here. Some items are informational only and are configured by WinScriptLive as part of the script. From time to time as updates are made the function may change. See the update documents for the details of the changes and additions.
At the end of each menu is the "Exit", this item will pop to the preceding upper level menu. If the menu is at the top the Exit item will exit the menu system and take the V16Pro back to viewing the operation of the script in one is running. The menu system will operate in parallel with all other operations and will not affect the script or current operations.

System Sub-menu

System

Firmware Version: – This item cannot be changed and is the version level of the operating system called the OS in the V16Pro.

SMPTE Version: – This item cannot be changed and is the version level of the code running on the SMPTE controller on the main board of the V16Pro. This processor is a separate subsystem within the unit.

Hardware Version: – This item cannot be changed and is the version level of the main board within the V16Pro.

Percent of Frame used by Process: – This item is a real-time indication of the current workload of the V16Pro. This number represents the amount of available time used in the currently selected frame time. If the number is below 100% all functions of the script are being executed as written. If the number is higher than 100% then some items are being leftover to be run on the next frame start.

This is not a problem if it happens once in a while because many of the system functions do not get processed on every frame edge. Multiple long messages on many serial ports may cause Percent of frame used to go high and then settle back down again. Pulsing workloads are normal when complex serial communications are taking place between older serial protocol devices.

If you're interested in knowing the amount of time available for processing based on the frame rate setting, they are listed below.

Frames Per Second	Time Available for Processing (ms)
30	33.33ms
29.97	33.36ms
25	40ms
24	41.66
23.976	41.708

Serial Number: – This item cannot be changed It is the units serial number. This number is part of the unique identification tag assigned to the unit. This number is also embedded into the Ethernet MAC address as network identifier.

Real-Time Clock

Time: – This is the real time of day and is used along with the date variable to trigger events in the show script.

Date: – This is the real calendar date and is used along with the time variable to trigger events in the user script.

NTP Enabled: – This is a way to keep the date and time synchronized with the Internet Network Time Protocol (NTP). This requires access to the Internet and a timeserver address. On power up the NTP server is accessed to set the time and date, if the NTP Error field remains set to zero this is an indication of a good NTP connection. The timeserver is accessed twice a day every day the unit is in operation.

NTP Errors: – This item cannot be changed and is reset to zero on power up. The NTP errors are counted and displayed, in this field, as an indication of a valid connection to the NTP server. Some errors are to be expected from time to time. As mentioned before, the timeserver is access twice a day every day, so if the error count is increasing there is a problem with the network connection. The date time update needs to be synchronized bi-weekly to maintain good time.

RTC Config: – This is where the operator may set the parameters for the RTC functions. Many are easy to understand while others may not.

Time, date and time zones are set by highlighting the item and pressing the menu-wheel and spinning the wheel to dial in the required information. Pressing the wheel again will move the cursor to the next field.

The daylight savings time type and enable selects how each country handles the time change as to the dates and if it is done at all.

NTP Ethernet Jack: – Selects the port on the back of the unit that is connected to the Internet and directs the V16Pro to search for the server thru that port, A or B.

NTP IP Address: – is the Internet IP address of the NTP server.

NTP Enabled: – This parameter permits the V16Pro to access the Internet connection to get NTP information.

LTC/SMPTE

Mode . . . SMPTE Time Display

Generate/Read shows the current mode of the SMPTE processor. The Display shows the active time code clock.

Current Frame Time . . . SMPTE Status

The frame per second setting is displayed in this location. Status Idle/Running.

Stop, Start SMPTE:

This is the control for the SMPTE processor.

LTC/SMPTE Config:

This field will open the SMPTE configuration menu and the user can view the setting such as the mode, frame rate, (preroll, start and end times), (loop, powerup, restart and idle modes), and output levels.

Exit

Network

There are two independent network controllers in the V16Pro and each of them has separate configuration options as follows

Network Adapter A (Defaults)

IP Address: 192.168.000.254

Subnet Mask: 255.255.255.000

Gateway: 192.168.000.001

Network Adapter B (Defaults)

IP Address: 192.168.000.253

Subnet Mask: 255.255.255.000

Gateway: 192.168.000.001

Exit

Password

Enter Password

Script Configuration

Script Filename is displayed in the first field. A directory of all the files found on the CF card may be displayed by turning the menu-wheel. Pressing the menu-wheel will select the file and cause it to begin flashing; pressing the wheel again will select that script to be loaded at the next restart.

Reload will start the default script selected from the field above.

The View Watches permits the monitoring of script variables. The user may select a number of variables to watch while the script is running. Selecting View Watches and pressing the menu-wheel will display the watch list.

Digital Inputs

Input Connector

The V16Pro has 16 opto-isolated inputs that can control the show operation. If desired, the software may be configured to allow the front panel buttons to mimic these inputs. Otherwise, the front panel buttons operate independently, as an additional set of sixteen inputs. We'll describe your configuration operation in a couple of pages.

Pin	Connection	Pin	Connection
1	*Input 1*	*20*	*Input 1 Return*
2	*Input 2*	*21*	*Input 2 Return*
3	*Input 3*	*22*	*Input 3 Return*
4	*Input 4*	*23*	*Input 4 Return*
5	*Input 5*	*24*	*Input 5 Return*
6	*Input 6*	*25*	*Input 6 Return*
7	*Input 7*	*26*	*Input 7 Return*
8	*Input 8*	*27*	*Input 8 Return*
9	*Input 9*	*28*	*Input 9 Return*
10	*Input 10*	*29*	*Input 10 Return*
11	*Input 11*	*30*	*Input 11 Return*
12	*Input 12*	*31*	*Input 12 Return*
13	*Input 13*	*32*	*Input 13 Return*
14	*Input 14*	*33*	*Input 14 Return*
15	*Input 15*	*34*	*Input 15 Return*
16	*Input 16*	*35*	*Input 16 Return*
17	*N/C*	*36*	*N/C*
18	*N/C*	*37*	*N/C*
19	*N/C*		

Table 7 – Parallel Input connections.

Two forms of inputs can be applied to the Parallel Inputs connector: Voltage Inputs, and Contact Closures. When a specific input on the V16Pro is software configured for Voltage Inputs, power for the connection is provided by an external source (in-rack power supply etc.), but when the input is configured as a Contact Closure, power is taken internally from the V16Pro.

Voltage Inputs vs. Contact Closures

The main reason for selecting one type in input over the other comes down to the distance the contact closure is from the unit. There are other reasons as well and the pros and cons are listed below.

Using Voltage Inputs over Contact Closures (Switches) will add additional complexity to the installation but provide greater distance.

1. PRO – The installer can overcome long distances when connecting contact closures. By using higher voltage sources, installers can compensate for resistance in wiring.

   CON – The installer must provide a power source for the contact closure(s).

2. PRO – Inputs can be completely isolated from one another.

   CON –An external power supply is needed

Using Contact Closures over Voltage Inputs provides a simple installation but is limited in distance.

1. PRO – Contact closure installations require only wiring and contacts

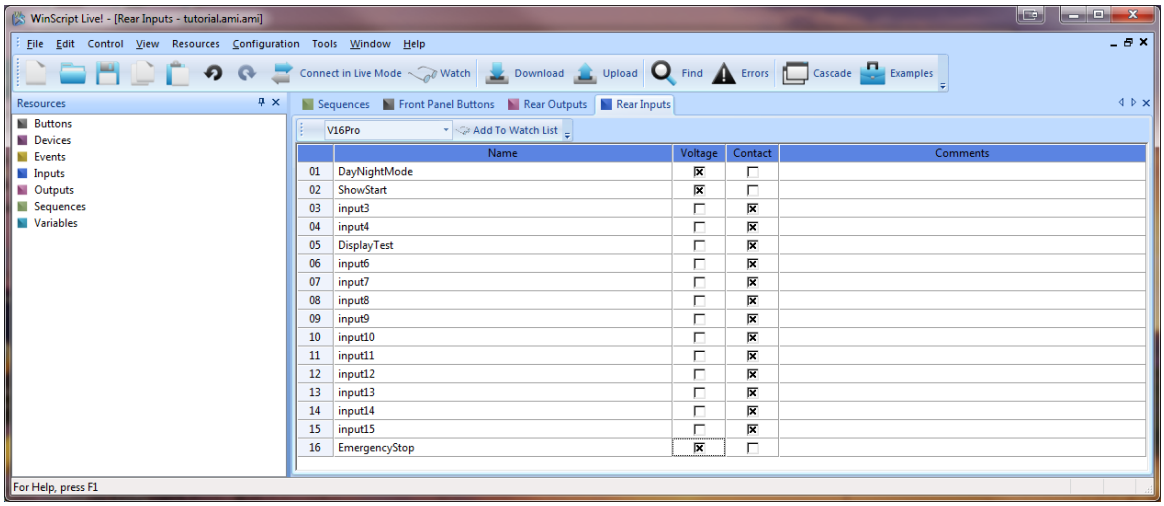
   CON – The contact closure must be located close (10-20 ft) from the unit.

2. PRO – No external power supply is needed.

   CON – The wiring will not be isolated; therefore, errors and problems in any circuit could affect all contact closures. A high voltage short to this wiring could damage the V16Pro.

Input Configuration

The inputs are configured by WinScriptLive software. Select Resources command from the main menu bar, and then select Inputs. The window shown below lets you select the input type for each input. Notice the input name may be changed, and a comment describing the input's use may be added. This makes it easier to remember what you were trying to do when you look at the script again later!



Connecting a Voltage Input

1. Using a Female DB37 connector, attach the appropriate wire from the Input signal pin (pin 1 for Input1, pin 2 for Input2, etc.) to the positive terminal of the external power supply.
2. Connect the negative terminal of the external power supply to one of the terminals of the contact closure or push button.
3. Connect the appropriate Input Return pin to the other terminal of the contact closure (pin 20 for Input1, Pin 21 for Input2, etc.)

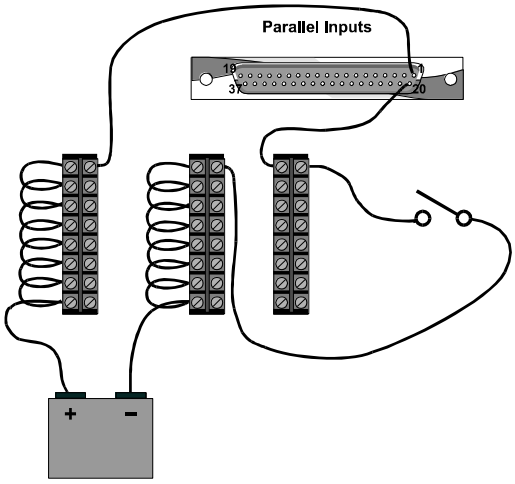


Figure 3 - Sample connection for a Voltage Input to Input1 of the Parallel Inputs connector. The terminal blocks are used for power bussing and modularization of the input signals.

Connecting a Contact Closure

1. Using a Female DB37, attach the appropriate wire from the Input signal pin (pin 1 for Input1, pin 2 for Input2, etc.) to one of the terminals of the external contact.
2. Connect the appropriate Input Return pin to the other terminal of the external contact (pin 20 for Input1, Pin 21 for Input2, etc.)

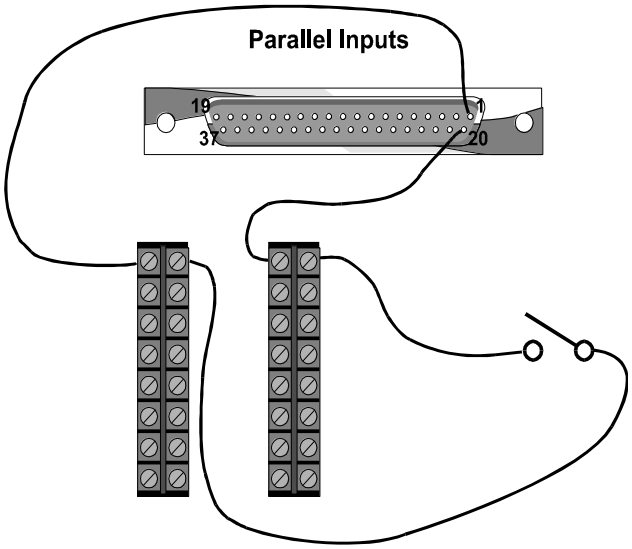


Figure 4 - Sample connection for a Contact Closure Input to Input1 of the Parallel Inputs connector.

Using Front Panel Buttons

The front panel buttons are configured by WinScript Live software. Select Resources command from the main menu bar then select Buttons. The window shown below will open and the user has the option of editing button parameters. Notice the Button name may be changed as well as adding comment to describe the function.

The front panel buttons on the V16Pro are not, by default, connected to the associated back panel input with the same number. In fact the front panel buttons are completely independent inputs. If the programmer needs to connect the front panel button to the back panel input with the same number the programmer need only select the "Couple" option in the "Front Panel Buttons" form shown below. This will make the V16Pro input hardware automatically behave the same way as the V16, allowing for drop-in replacement. WinScript Live gives the programmer control to program the front panel buttons as needed.

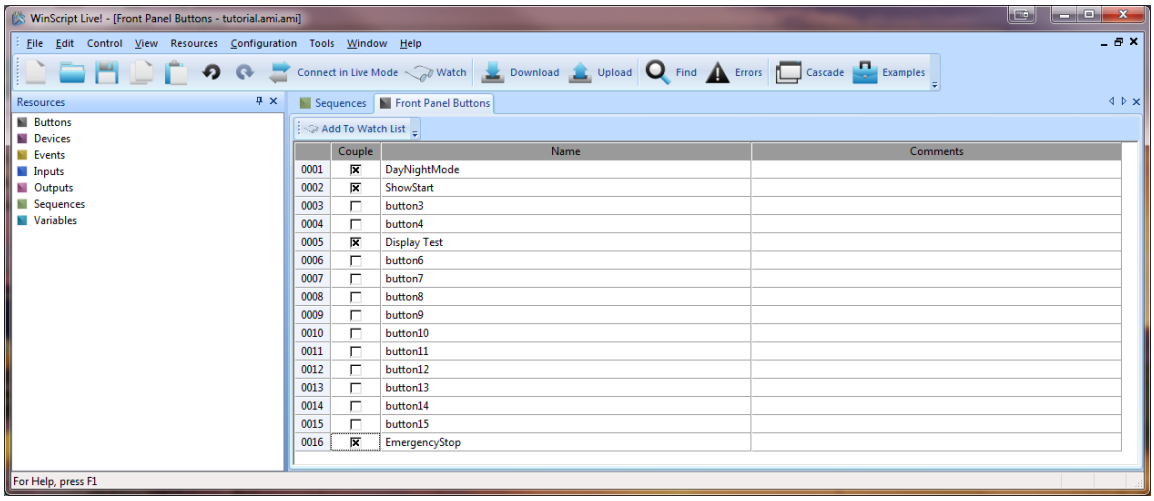


Figure 5 - Buttons View

Digital Outputs

Configuring Outputs

The V16Pro provides 16 Dry-Contact Relay Outputs for discrete control. The initial state of each output may be configured by WinScript Live to be open or closed when the script is started.

The back panel outputs are configured by WinScript Live software. Select Resources command from the main menu bar then select Outputs. The window shown below opens. You may change the name, define the initial state of the output, and add a comment to describe the output's use.

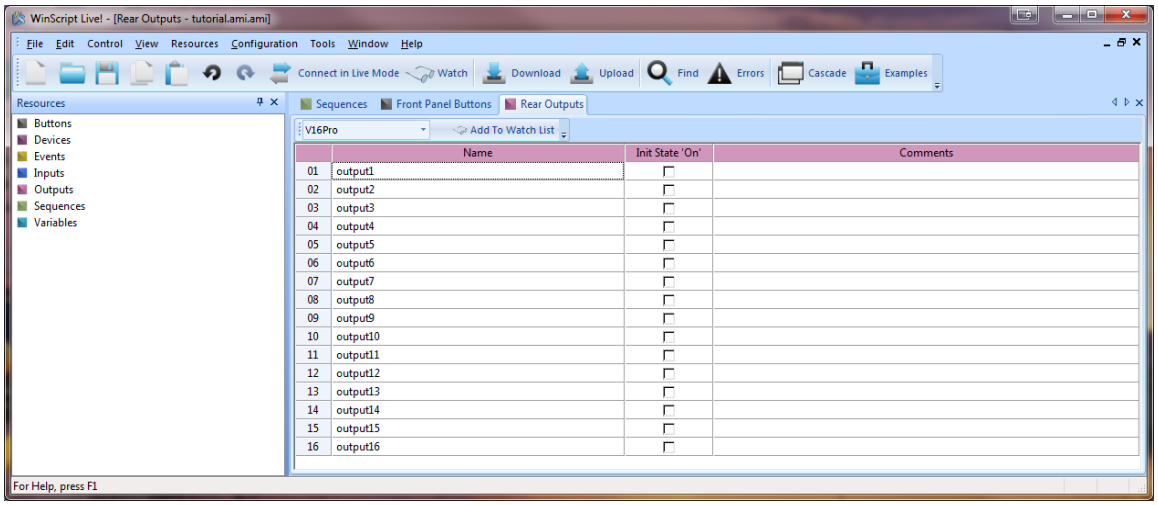


Figure 6 - Outputs View

Output Connector

Note The Relay Outputs are fused at 900mA using self-restoring polymer fuses. If an overload occurs, the fuse will open until the problem is corrected; then it will heal itself.

Pin	Connection	Pin	Connection
1	*Output 1*	*20*	*Output 1 Return*
2	*Output 2*	*21*	*Output 2 Return*
3	*Output 3*	*22*	*Output 3 Return*
4	*Output 4*	*23*	*Output 4 Return*
5	*Output 5*	*24*	*Output 5 Return*
6	*Output 6*	*25*	*Output 6 Return*
7	*Output 7*	*26*	*Output 7 Return*
8	*Output 8*	*27*	*Output 8 Return*
9	*Output 9*	*28*	*Output 9 Return*
10	*Output 10*	*29*	*Output 10 Return*
11	*Output 11*	*30*	*Output 11 Return*
12	*Output 12*	*31*	*Output 12 Return*

13	*Output 13*	*32*	*Output 13 Return*
14	*Output 14*	*33*	*Output 14 Return*
15	*Output 15*	*34*	*Output 15 Return*
16	*Output 16*	*35*	*Output 16 Return*
17	*N/C*	*36*	*N/C*
18	*N/C*	*37*	*N/C*
19	*N/C*		

Table 10 – Parallel Output connections.

Non-inductive load

Non-inductive loads are resistive. Incandescent bulbs, LEDs and filament lamps do not require additional hardware. Loads that do not have inductors, coils or transformers are non-inductive loads.

1. Using a DB37 Male connector, attach the appropriate Output pin (pin 1 for Output1, pin 2 for Output2, etc.) on the Parallel Outputs connector to the positive terminal of the external power supply.
2. Using the same DB37 Male connector, connect the corresponding Output Return pin (pin 20 for Output1, Pin 21 for Output2, etc.) to the positive terminal of the device that is receiving the output signal.
3. Connect the negative terminal of the device that is receiving the output signal to the negative terminal of the external power supply.

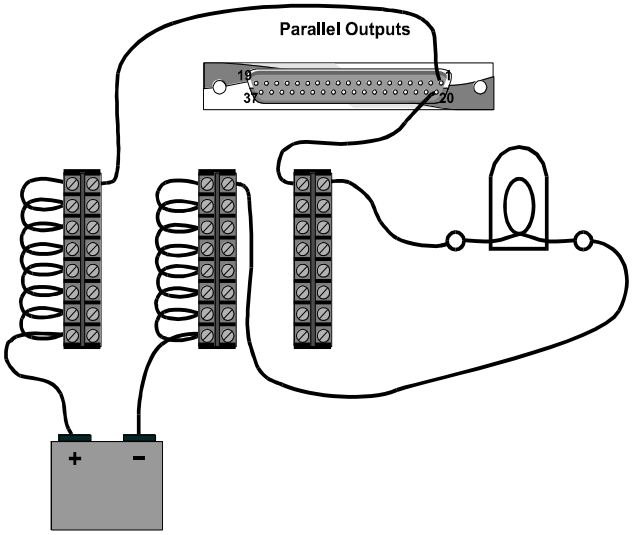


Figure 7 – An indicator lamp is a common example of a non-inductive load

.

Inductive loads

Inductive loads have inductors, coils or transformers as part of the load or may be the load. Relays, motors and mechanical actuators such as door latches, curtain controllers and other such devices are all inductive loads. These devices store electromagnetic energy to do work. When turned off, the energy stored within the device must be returned to a ground state or damage could occur to other devices in the system. Note the diode across the load in figure 7 below.

1. Using a DB37 Male connector, connect the appropriate Output pin (pin 1 for Output1, pin 2 for Output2, etc.) on the Parallel Outputs connector to the positive terminal of the external power supply.
2. Using the same DB37 Male connector, connect the corresponding Output Return pin (pin 20 for Output1, Pin 21 for Output2, etc.) to the positive terminal of the device that is receiving the output signal.
3. Connect the negative terminal of the device that is receiving the output signal to the negative terminal of the external power supply.
4. Connect an appropriate 1N4000-series (1N4001-1N4007) diode across the load. Note the polarity of the diode in reference to the supply.

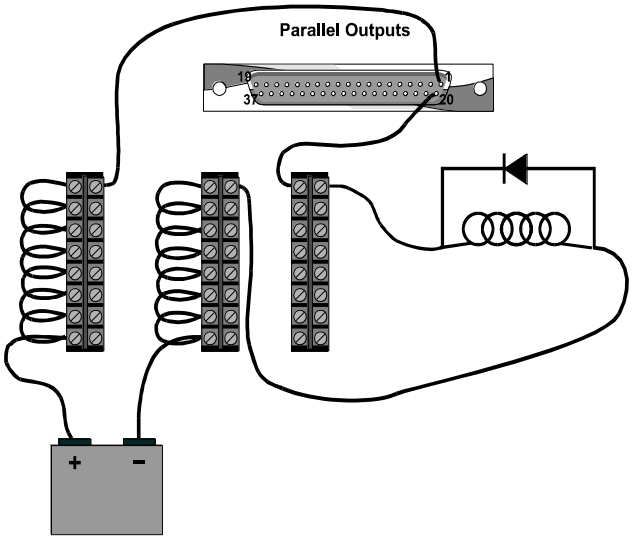


Figure 8 - A relay coil or solenoid is a common example of an inductive load and must have a 1N4000-Series snubber diode placed across it. Be sure to observe proper polarity (anode to negative side).

Video Sync Input

The V16Pro is designed to extract the vertical frame clock from an external video sync signal. This signal should be NTSC or PAL composite video at the standard sync level of 4-volts peak-to-peak.

The V16Pro also works with "Black Burst Sync", if its level is high enough. Black burst sync is generally well below the C-Sync level, which is approximately 1-volt peak-to-peak. Signals at this level should ***not*** be terminated with the 75-Ohm terminator.

The sync signal is connected to the V16Pro via a rear panel BNC connector. If additional devices are to be wired to the same sync signal, a BNC "T" connector may be used to daisy chain the signal. If the V16Pro the only device connected or is the last device in the chain, terminate the line by setting the "75 Ohm" rear-panel dipswitch to "on" position. Otherwise leave it off.

SMPTE Reader/Generator

The V16Pro has a built in SMPTE read/generator. The unit will accept differential SMPTE in and use it to synchronize the selected script. The V16Pro may be programmed to output SMPTE to other systems. The SMPTE time code may be in following frame rates:

23.976
24
25
29.97
30D drop format skips a frame to remain in sync with the 29.97 rate.
30

Power Supply

The V16Pro includes an external universal power supply that allows connection too many domestic as well as international wall voltages (110VAC, 220VAC, 200VAC) without special configuration. The V16Pro uses a threaded 5.5 mm barrel connector as its power input.
The DC power requirements are 12-18 VDC at 3.3 Amps

The power supply that comes with the V16Pro has the following specifications:

Input: 100-250VAC, 50-60Hz, 0.7-0.3A

Output: 18 VDC @ 3.3 Amps

Rear DIP Switches

75Ω Termination

The first switch is used to terminate the sync input connector. Down (ON) will apply 75Ω termination. Up (OFF) will remove this termination.

Terminated

No Termination

Sync

The second switch configures the V16Pro to receive either Blackburst or Composite Sync (C-Sync). Down (ON) will configure for C-Sync. Up (OFF) will configure for Blackburst.

C-Sync

Blackburst

Reset

will restore certain settings back to factory defaults. These settings include: IP Address, Front Panel Password, Date/Time and related time zone configuration, NTP, SMTP, E-mail Settings, and Script Variables stored using "Save Variable."

To apply the reset, flip the switch up into the "OFF" position. Leave in this position for about 1 minute. Flip the switch down again and power-cycle the V16Pro.

Note: Script Variables take the longest to clear. After a few seconds, most settings will be cleared to defaults.

Normal Operation

In Reset

Option

For future use only. No effect.

Firmware

The V16Pro's operating system is called ScriptOS. It is stored in internal memory. Occasionally we publish updates, which are available for free download on our website. The procedure for updating the OS is as follows:

1. Unplug the power cable from the back of the unit.
2. Obtain the OS.NEW file from our web site or from customer service.
3. Copy the OS.NEW to a flash card. This file should be the only file on the card.
4. Insert the flash card into the slot in the back of the unit.
5. Reconnect the power cable.
6. The unit will begin the procedure of updating the OS automatically. The display will show "Updating Firmware" followed by a series of numbers showing the progress.
7. The update finishes with the "Update Complete" and will rename the OS.NEW to OS.SAV. To re-flash the unit the OS.SAV must be renamed back to OS.NEW to restart the process.

Show Memory

When scripts are compiled and sent to the V16Pro, the data is stored in the rear-panel accessible Compact Flash card. The smallest Compact Flash card made will accommodate about a hundred copies of the largest script ever written, so it won't be necessary to upgrade this memory! Finally something that's not bigger, costs more or you have to order.

V16+ or V4+ Compatibility

One of our goals in designing these show was to make then a drop-in hardware replacement for the earlier V Plus series. So if you're already familiar with the V Plus family (or are replacing one in an existing installation), you'll be pleased to find that you already know a lot about these show controllers. Although you'll need to import your scripts to take advantage of the more advanced WinScript Live! Programming environment, your hardware should be good to go.

Hardware Compatibility

If you are using ports 1 – 4 of the V16 Plus in RS422 mode of operation, please take note of the following changes. The buffers used in the show controllers use a technology that provides both single ended (RS232) and differential ended (RS422/485) connections to be used without changing circuit components.

The TXD+ and TXD- signals needed to be reversed when using the show controllers. Pin 3 is now TXD- and pin 9 is TXD+ in RS422/485 mode only. RS232 operations remain unchanged.

If you were previously using a serial port as "MIDI", you will need to use the "MIDI out" connector instead of the 9 pin connector. You will also need to select the "MIDI" port instead of the serial port number you were previously using.

Importing .amw files (WinScript scripts)

You will need to import your script into WinScriptLive by going to "File-->**New**". Then, after selecting your controller, select **" Import" from the "File" menu**.

You may need to re-select a "Device" type used for a particular serial port. This is under the "Resources-->Devices" (Previously "Ports") Click "Edit" on the line of the device you wish to assign to a particular protocol.

If you have custom protocols (.pcl files) for your .amw script, you will need to convert them to .prd files using the "Product File Converter" under the "Tools" menu. Then, place the .prd file under the "Alcorn McBride Inc\WinScriptLive\My Product Files" directory under "My Documents" directory.

V4Pro

Figure 9- Front View

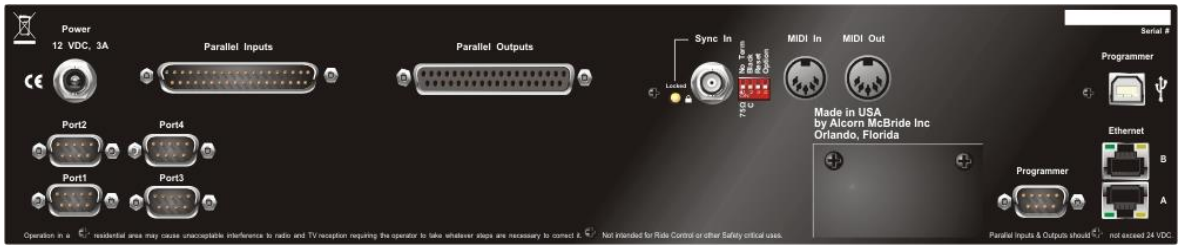


Figure 10 - Rear View

Specifications

Size and Weight:	Standard 2U rack mount (3.5" x 17" x 10.5"), 10 lbs
Power:	12 VDC at 3 amps. An external power supply is included with the V4Pro and will provide the required input power. The power supply is rating is 100 to 250 VAC, 50 to 60 Hz, 25-watts maximum. UL listed Class 2 power adapter
Environment:	0 to 38 C (32 to 100 F) 0 to 90% relative humidity, non-condensing
Front Panel:	8x40 VFD Display Power LED Vsync LED Error LED Acknowledge (ACK) LED 4 Serial Activity LEDs 16 Input Status LEDs 16 Output Status LEDs 16 Pushbuttons
Rear Panel:	Programming Port DB-9M 4 Serial Ports DB-9M MIDI Input 5-pin DIN Female MIDI Output 5-pin DIN Female Discrete Inputs DB-37M Discrete Outputs DB-37F NTSC or PAL Sync Input BNC

	Power barrel jack Ethernet Jack A Ethernet Jack B
Serial Ports:	(4) RS-232C, RS-422/485, individually software configurable 300 baud – 115.2 Kbaud 7, 8, or 9 Data Bits 1 or 2 Stop Bits All parity types
MIDI	MIDI input and output ports
Opto Inputs:	(16) Each input is software configurable for voltage or contact-closure operation. Input voltage range is 5-24VDC. Misconfiguration or reverse polarities will not damage inputs. Trigger latency < 1 frame (33.3ms @ 30 fps).
Relay Outputs:	(16) Contact Closures limited internally to 900 mA with self-restoring polymer fuses.
Show Memory:	Removable Compact Flash card allows scripts with millions of events. Maximum size per show is 5MB. Multiple shows can be loaded per card.

Certifications

EMC Compliance: US, Canada and Europe (CE Mark)

Emissions Compliance:

EN 55103-1:2009, Electromagnetic compatibility (emissions). Product family standard for audio, video, audio-visual and entertainment lighting control apparatus for professional use.

Formal Emissions Compliance, Information Technology Equipment, EN 55022:2010 (EU/AUST), FCC CFR 47

Part 15 (US), ICES-003 (Canada), VCCI V-3 (Japan) Class B Emissions.

- Radiated and Conducted emissions

- Include Telecommunications Port

Formal Emissions Compliance, Radiated Magnetic requirements. - 100mm, 50 Hz to 50KHz

In rush Current : Annex B

EN 61000-3-2: Limits for Harmonic Current Emissions

EN 61000-3-3, Limitation of Voltage Fluctuations and Flicker

Immunity Compliance:

EN 55103-2:2009, Electromagnetic compatibility (Imuunity). Product family standard for audio, video, audio-visual and entertainment lighting control apparatus for professional use.

EN 61000-4-2, Electrostatic Discharge, Immunity Compliance

EN 61000-4-3, Radiated Electromagnetic Fields, Immunity Compliance - 80 MHz to 2.7 GHz

EN 61000-4-4, Electrical Fast Transient / Burst, Immunity Compliance

EN 61000-4-5, Surge, Immunity Compliance

EN 61000-4-6, Conducted Immunity Compliance

EN 61000-4-8, Magnetic Field Immunity Compliance - Annex A: 50 Hz to 10 kHz

EN 61000-4-11, Voltage Dips and Variations - Audio Frequency: Annex B

Serial, USB, Ethernet, Inputs and Outputs

Refer to the V16Pro section for hardware information about serial ports, usb, ethernet and IO. Any information in the V16Pro hardware section regarding SMPTE or Ports 5-16 should be ignored for the V4Pro.

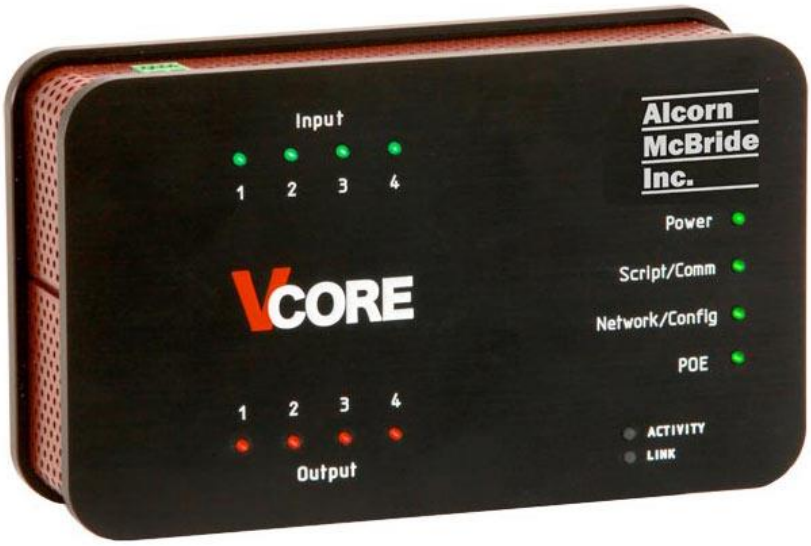


Figure 11 - Front View

Figure 12 - Rear View

Figure 13 - Top View

Figure 14 - Bottom View

Specifications

Size and Weight:	8” W x 4.75” H x 2” D (20.32 cm W x 12.07 cm H x 5.08 cm D), 2RU 3 lbs. (1.4 Kg)
Power:	Power Options: Power-over-Ethernet (POE) capable; or AC Adapter AC Adapter Option: 100 to 250 VAC, 50 to 60 Hz, 20W maximum (CE, UL, CSA, WEEE, RoHS Compliant)
Environment:	0 to 38 C (32 to 100 F) 0 to 90% relative humidity, non-condensing
Front Panel:	Power LED Script Status LED (SD read, external device status) IP/Configuration LED (DHCP, valid IP; other config) Power-over-Ethernet (POE) LED (2) Ethernet Status LED’s (Link, Activity) (4) Input Status LEDs (4) Output Status LEDs
Rear:	Mounting for standard 35mm Din Rail
Side Panel(s):	USB 2.0 Type B (Programming Port) RJ-45 (Ethernet) 4 Position DIP Switch: DHCP, CC/Voltage, Settings Reset, Backlight on/off Removable SD Card for Script Storage (currently 4GB) (1) Serial Port 3 pin (phoenix style screw terminal)
Serial Ports:	(1) RS-232C 300 baud – 115.2 Kbaud 7, 8, or 9 Data Bits 1 or 2 Stop Bits All parity types
Network:	(1)10/100 Base-T Ethernet Supports hundreds of networked device protocols UDP Ethernet IP Client, native CIP TCP Client, TCP Server, ModBus TCP HTTP; Custom web pages, FTP server DHCP, NTP, SMTP Client Compatible with ShowTouch 7”, 10”, 17”, and ShowTouch for iOS
Digital Inputs:	Input voltage range 5-24 VDC, 5 mA maximum Hardware protected against misconfiguration Trigger latency < 1 frame
Relay Outputs:	Contact Closures limited internally to 900 mA with self-restoring polymer fuses.
Show Memory:	Removable SDHC card allows scripts. Multiple shows can be loaded per card.

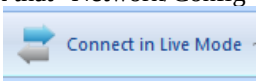
Setting VCore IP Address

The three different ways to set the IP address are listed below.

DHCP (Automatic Assignment)

Connect to your existing network. Make sure DIP switch position 1 is "ON"
The "Network/Config" LED will turn green when a valid IP has been assigned.

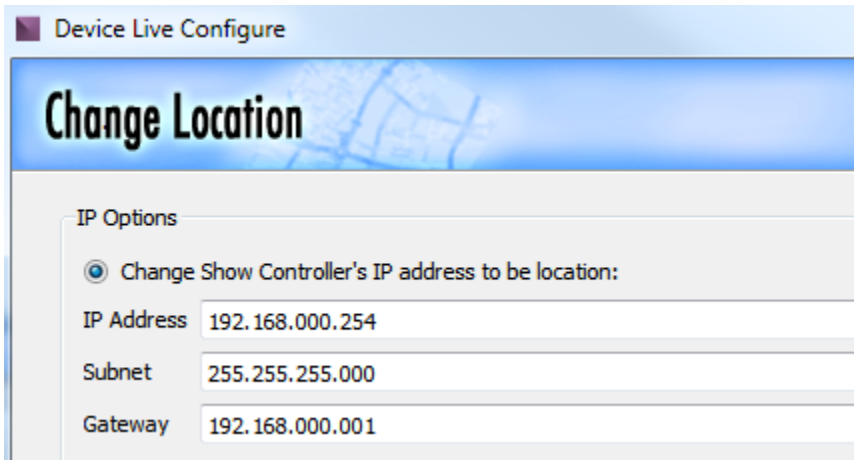
Manual IP Set

1. Set DIP switch 1 to "OFF"
2. Check that "Network/Config" LED is yellow. If it is not, toggle DIP switch 1 "ON" then "OFF".
3. Click 

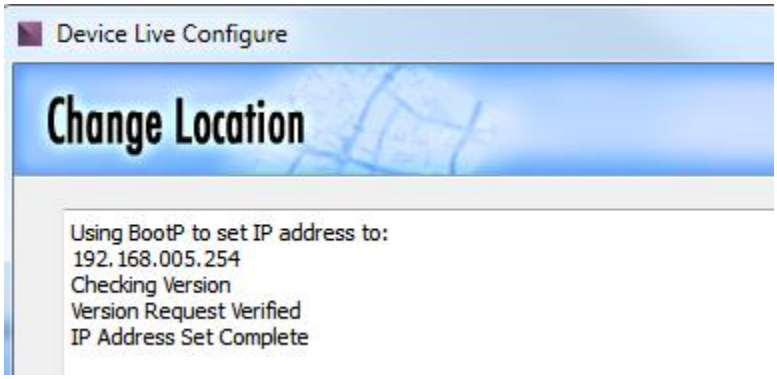
in WinScriptLive

4. Click on the line for the "Broadcast" from the VCore. If this does not appear, try clicking "Clear History"

5. Click "OK" to change the IP address
6. Enter an IP address that is on the same subnet as your PC



7. Once the set is completed, your PC will send a version request to verify that it can reach the device.



8. If your device cannot be reached, try clicking "Send Again."
9. If it still cannot be reached, try a different IP address by starting again with step 2.

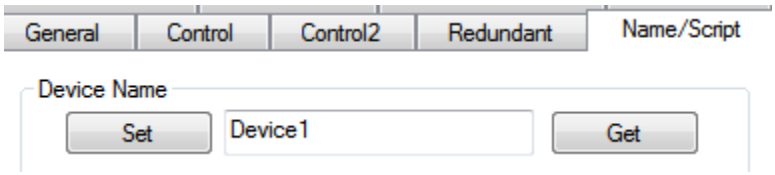
USB or Serial Set

1. Set DIP switch 1 to "OFF"
2. Open "AMI-Terminal" from the "Tools" menu in WinScriptLive or from start menu.
3. Click on "USB" or "Serial" radio button
4. Click on the "Network" tab and select the "Set" radio button
5. Type in desired IP address and click on the "IP" button.
6. Type in desired subnet and gateway and click on corresponding buttons.

Naming VCore

We took the liberty of giving your VCore a name based on it's MacID. This "name" appears when connecting in WinScriptLive. But, if you're tired of the name "Victor Corben", "Victoria Corell" or other "VC" name, you can name your VCore using AMI-Terminal.

1. Select "AMI-Terminal" from the "Tools" menu
2. Select either Serial, Ethernet, or USB
3. Click the "Name/Script" tab
4. Enter the desired name and click "Set"



Serial Port (RS-232)

The VCore provides a serial port for controlling show related devices.

The 3pin serial port on the VCore can be used as either a “programming port” to connect and receive scripts, or a standard “device” port to control external devices.

Serial port wiring is “GND, RX, TX” from left to right.
(When facing connector with Outputs connector on the left and Ethernet jack on the right)

Ethernet Ports

There is one Ethernet port available on the VCore. For additional information on configuring and using Ethernet see the sections on Ethernet and Networking Primer at the end of this manual. For Cable pin-out, see the Ethernet section of the V16Pro manual.

The protocols available on the VCore are currently: Ethernet IP Client (CIP), TCP Client, TCP Server, ModBus TCP, HTTP (Custom web pages), FTP server, DHCP, NTP, and SMTP Client.

These protocols can be accessed using the custom product files for a particular device, or sometimes by using script commands directly in the V16Pro.

USB

The USB Port is used for programming and Live Mode only. The driver can be installed with WinScriptLive.

Digital Inputs

Input Connector

The VCore has 4 inputs that can control the show operation

Two forms of inputs can be applied to Inputs connector: Voltage Inputs, and Contact Closures. When a specific input on the VCore is software configured for Voltage Inputs, power for the connection is provided by an external source (in-rack power supply etc.), but when the input is configured as a Contact Closure, power is taken internally from the VCore.
The **DIP switch position 2** determines Contact Closure or Voltage Mode.
For the pros and cons of each, see the section titled "Voltage Inputs vs. Contact Closures" in the V16Pro section of this manual.

Input Configuration

The input names are configured by WinScript Live software. Select Resources command from the main menu bar, and then select Inputs. Notice the input name may be changed, and a comment describing the input's use may be added. This makes it easier to remember what you were trying to do when you look at the script again later!

Connecting a Voltage Input

1. Attach the wire from the Input signal pin to the + terminal of the external power supply.
2. Connect the negative terminal of the external power supply to one of the terminals of the contact closure or push button.
3. Connect the appropriate Input Return pin to the other terminal of the contact closure

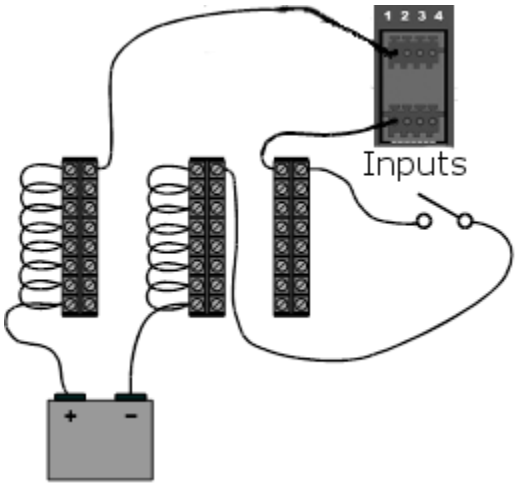


Figure 15 - Sample connection for a Voltage Input to Input1 of the Parallel Inputs connector. The terminal blocks are used for power bussing and modularization of the input signals.

The **positive** (+) terminal for inputs is located next to the **"1 2 3 4"**

The **return (-)** terminal is located next to the word "**Inputs**"

Connecting a Contact Closure

1. Attach the appropriate wire from the Input signal pin (pin 1 for Input1, pin 2 for Input2, etc.) to one of the terminals of the external contact.
2. Connect the appropriate Input Return pin to the other terminal of the external contact closest to the word "inputs"

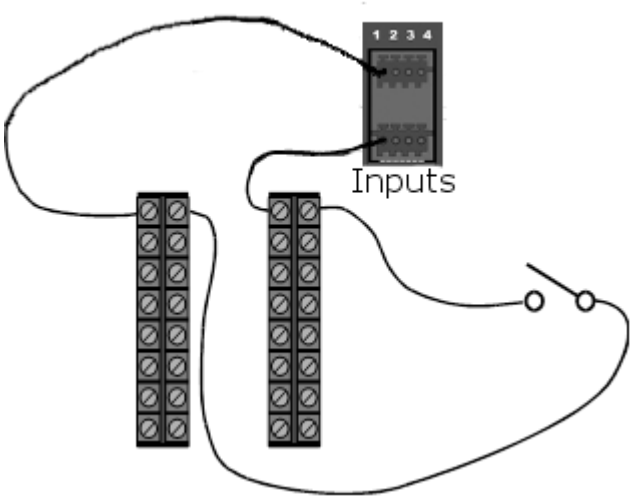


Figure 16 - Sample connection for a Contact Closure Input to Input1 of the Parallel Inputs connector.

Digital Outputs

Configuring Outputs

The VCore provides 4 Dry-Contact Relay Outputs for discrete control. The initial state of each output may be configured by WinScript Live to be open or closed when the script is started.

The back panel outputs are configured by WinScript Live software. Select Resources command from the main menu bar then select Outputs. The window shown below opens. You may change the name, define the initial state of the output, and add a comment to describe the output's use.

Output Connector

Note The Relay Outputs are fused at 900mA using self-restoring polymer fuses. If an overload occurs, the fuse will open until the problem is corrected; then it will heal itself.

Pin	Connection	Pin	Connection
1	*Output 1*	*5*	*Output 1 Return*
2	*Output 2*	*6*	*Output 2 Return*
3	*Output 3*	*7*	*Output 3 Return*
4	*Output 4*	*8*	*Output 4 Return*

Table 10 – Parallel Output connections.

Non-inductive load

Non-inductive loads are resistive. Incandescent bulbs, LEDs and filament lamps do not require additional hardware. Loads that do not have inductors, coils or transformers are non-inductive loads.

1. Attach the appropriate Output pin (pin 1 for Output1, pin 2 for Output2, etc.) on the Parallel Outputs connector to the positive terminal of the external power supply.
2. Connect the corresponding Output Return pin (pin 5 for Output1, Pin 6 for Output2, etc.) to the positive terminal of the device that is receiving the output signal.
3. Connect the negative terminal of the device that is receiving the output signal to the negative terminal of the external power supply.

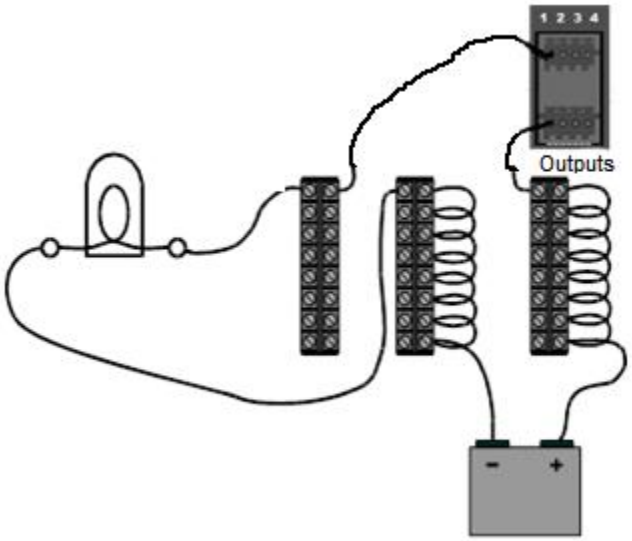


Figure 17- An indicator lamp is a common example of a non-inductive load.

Inductive loads

Inductive loads have inductors, coils or transformers as part of the load or may be the load. Relays, motors and mechanical actuators such as door latches, curtain controllers and other such devices are all inductive loads. These devices store electromagnetic energy to do work. When turned off, the energy stored within the device must be returned to a ground state or damage could occur to other devices in the system. Note the diode across the load in figure 7 below.

1. Cconnect the appropriate Output pin (pin 1 for Output1, pin 2 for Output2, etc.) on the Parallel Outputs connector to the positive terminal of the external power supply.
2. Connect the corresponding Output Return pin (pin 5 for Output1, Pin 6 for Output2, etc.) to the positive terminal of the device that is receiving the output signal.
3. Connect the negative terminal of the device that is receiving the output signal to the negative terminal of the external power supply.
4. Connect an appropriate 1N4000-series (1N4001-1N4007) diode across the load. Note the polarity of the diode in reference to the supply.

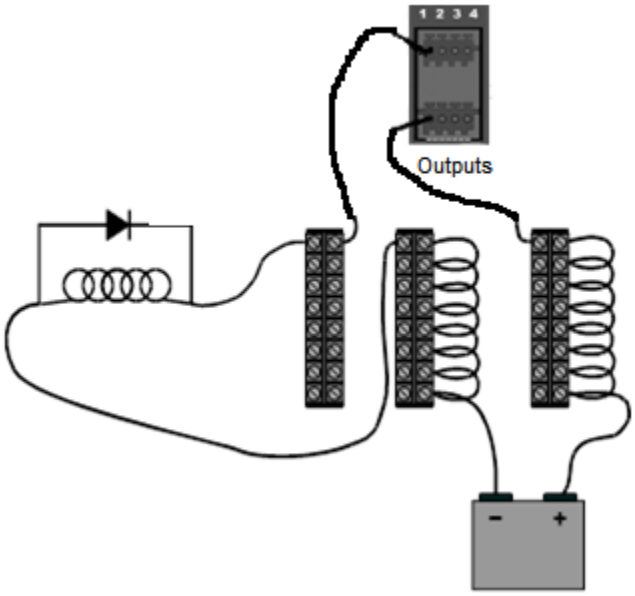


Figure 18 - A relay coil or solenoid is a common example of an inductive load and must have a 1N4000-Series snubber diode placed across it. Be sure to observe proper polarity (anode to negative side).

LED Indicators

Script/Comm - usually solid green, frequently solid yellow

LED	Connection
Off	No script is loaded, Script failed to load
Solid Green	Script is loaded and running
Fast Blinking Red	SD card for script removed or not detected
Red Single Blink or Solid Red	Invalid Communication from device or no response from device
Yellow Single Blink or Solid Yellow	Valid communication to external devices

Network/Config - usually off, frequently green

LED	Connection
Solid Green	Valid IP address is in place
Slow Blinking Green	Live Mode connected
Solid Yellow	IP Address is 0.0.0.0 and requesting IP using Alcorn's BOOTP Method
Solid Red	IP Address is 0.0.0.0 and using standard DHCP (unable to reach DHCP server)

LEDs on Boot

Memory Init:
Script/Comm : yellow in process, red is error 4 seconds
IP/Config: yellow in process, red is error 4 seconds

RTC Init:
Script/Comm: off
IP/Config: red is error 4 seconds

CF Card Init:
Script/Comm: green in process, red is error 4 seconds
IP/Config: off

Network Init:
Script/Comm : off
IP/Config: green in process, red is error 4 seconds

Power Supply

The VCore includes an external universal power supply that allows connection too many domestic as well as international wall voltages (110VAC, 220VAC, 200VAC) without special configuration. The VCore uses a 2-position terminal block plug (3.5mm spacing) as its power input.
The DC power requirements are 9-36 VDC at 1 Amp.

The power supply that comes with the VCore has the following specifications:

Input: 100-250VAC, 50-60Hz, 0.7-0.3A

Output: 18 VDC @ 3.3 Amps

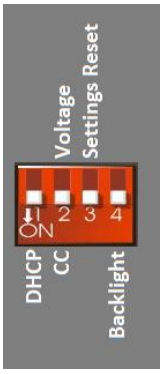
Power over Ethernet

The VCore can also be powered via PoE(Power over Ethernet) class 3 (up to 13W), which is a standard that passes both power and data along the same Ethernet cable. This permits the installer to not have to wire the DC Power cable, if the network and wiring in place supports the standard.

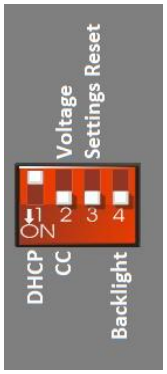
Rear DIP Switches

DHCP

The first switch is used to toggle DHCP. Down (ON) will enable DHCP. Up (OFF) will disable this feature.



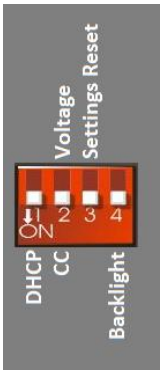
DHCP



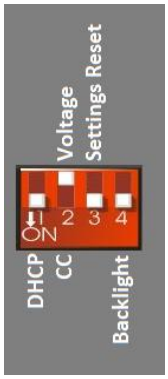
No DHCP

Contact Closure / Voltage Input

The second switch configures the VCore to receive either Contact Closure or Voltage Input. Down (ON) will configure for Contact Closure. Up (OFF) will configure for Voltage Input.



Contact



Voltage Input

Reset

Will restore certain settings back to factory defaults. These settings include: IP Address, Date/Time and related time zone configuration, NTP, SMTP, E-mail Settings, and Script Variables stored using "Save Variable."

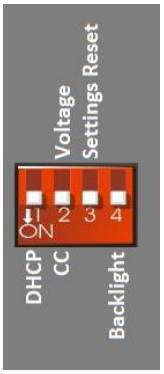
To apply the reset, flip the switch up into the "OFF" position. Leave in this position for about 1 minute. Flip the switch down again and power-cycle the VCore.

Note: Script Variables take the longest to clear. After a few seconds, most settings will be cleared to defaults.

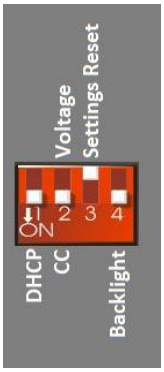
Backlight on / off

This switch will be used to enable or disable the backlight feature. Down (ON) will turn on the backlight. Up (OFF) will turn it off.

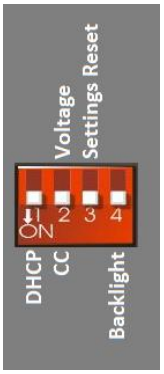
Closure



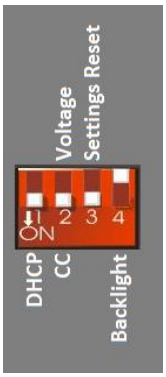
Normal Operation



In Reset



Backlight ON



Backlight OFF

Show Memory

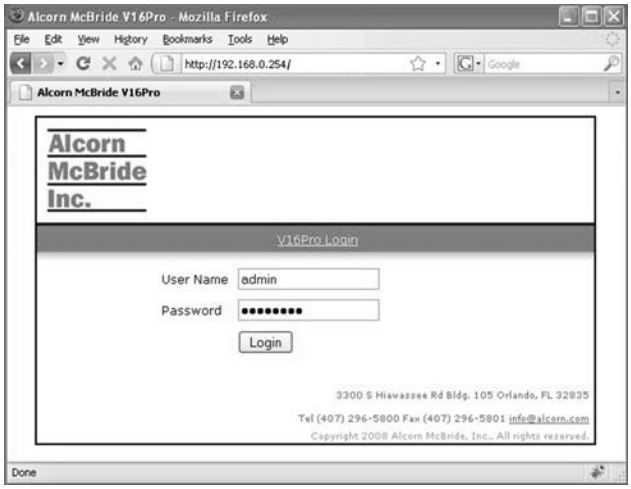
When scripts are compiled and sent to the VCore, the data is stored in SD card. The smallest Compact Flash card made will accommodate about a hundred copies of the a script, so it won't be necessary to upgrade this memory.

Scheduler (Web-based)

Schedule files are text files with a .xml extension. Schedule files can be created in any editor and placed on the compact flash card. The following section describes using the web-based interface to create a schedule file. The example that follows will use the V16Pro and will apply equally well to the entire show controller family.

Getting Started

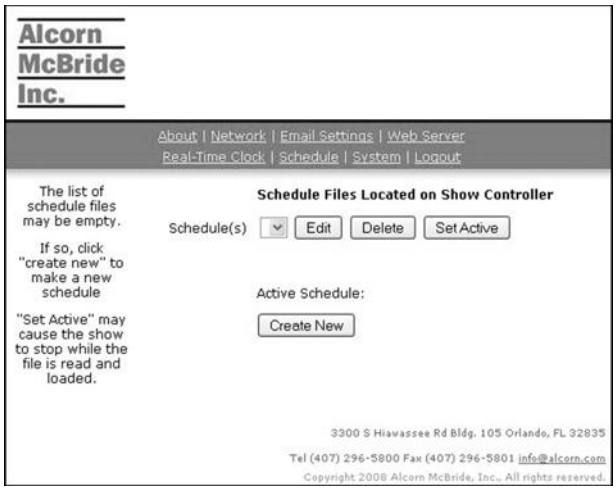
Using a web browser, enter the IP address of the V16Pro into the address bar. The admin user must be logged-in to edit schedule files.



Enter the admin user name and password then click “Login”.

If successful, the V16Pro serial number and firmware version will be displayed.

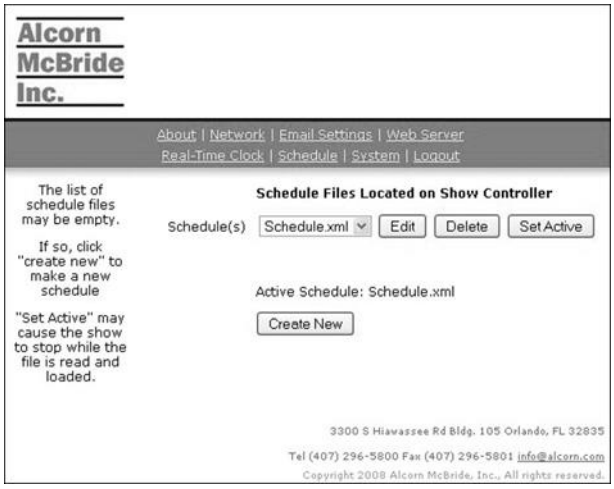
Creating a New Schedule File



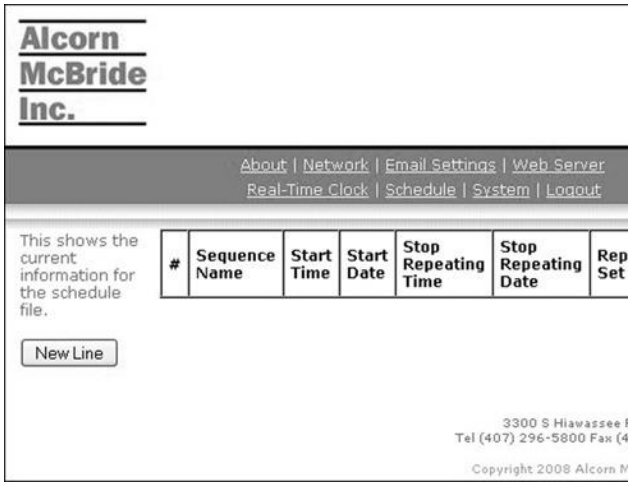
Click “Schedule” from the links at the top to display the schedule selection menu.

Click “Create New” to make a new schedule.

Enter the name of the schedule and click "Create"

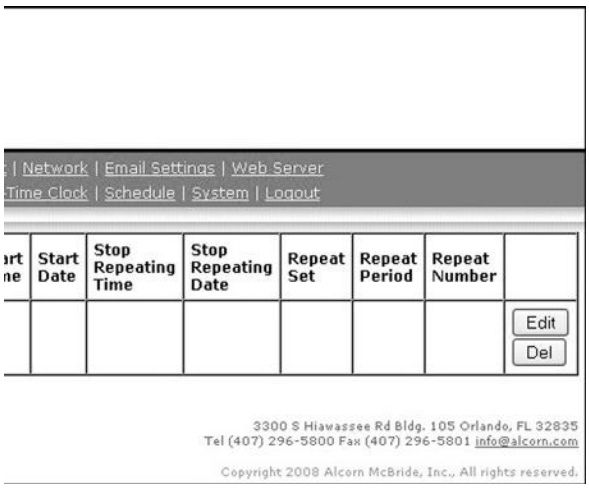


Select the schedule file to edit and click "Edit"

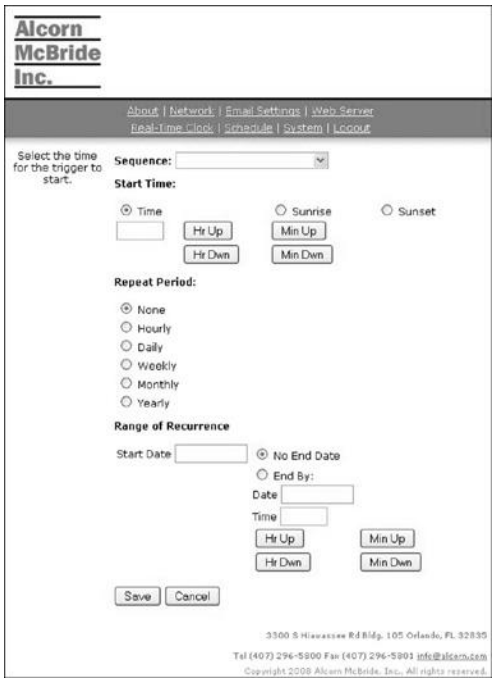


New schedule files have no entries. Click "New Line" to create a new entry in the schedule.

Editing Schedule Entries



Click “Edit” to change the new entry created.



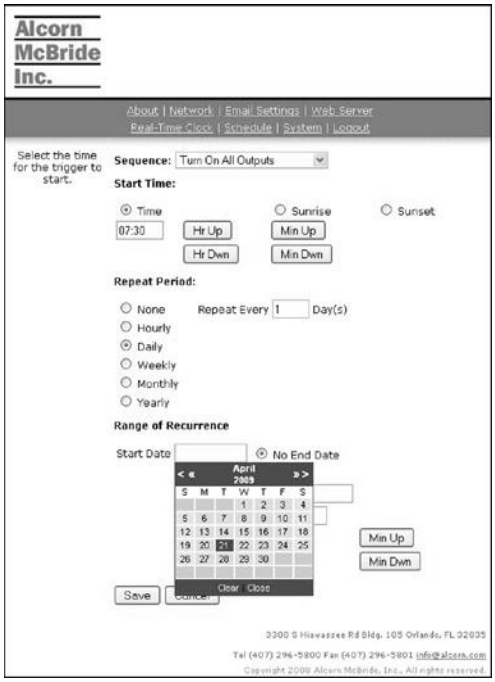
The entry is empty when it is first created. Enter the following information to complete entry:

Sequence: The name of the sequence to start

Start time: The time from 00:00 to 23:59 to start the selected sequence

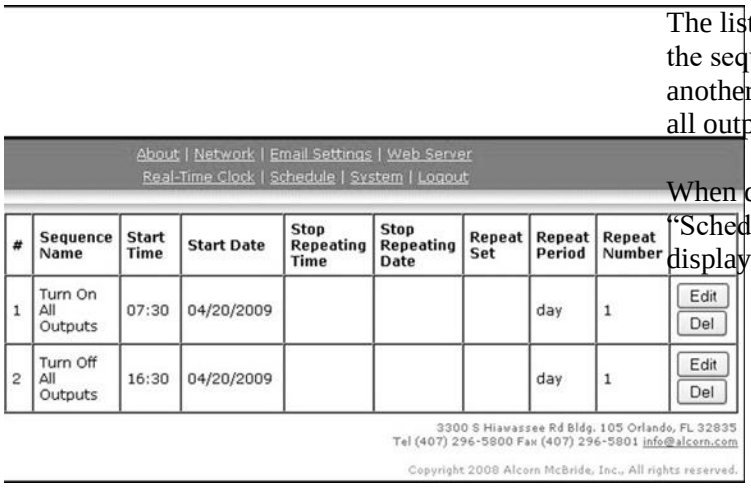
Repeat Period: Select whether the entry should repeat, and how often

Start Date: The date to start the selected sequence. If this entry is set to repeat, this is the day it will begin on.



In this example, the sequence "Turn On All Outputs" will start at 7:30am every day starting on April 21st, 2009.

Click "Save" to save changes to the entry.



The list now shows the entry that starts the sequence "Turn On All Outputs" and another entry that was created to turn off all outputs every day at 4:30pm (16:30).

When done with this schedule. Click "Schedule" from the links at the top to display the schedule selection menu.

Select the schedule to use and click “Set Active”. The active schedule will always be shown on this page.

WEB Server Quick Start

This quick start will demonstrate how to connect to the web server from your computer, use the configuration web pages provided, and walk-through how to setup a web page for your customer. This quick start assumes that theV16Pro is used and is configured using factory settings and the original Compact Flash card provided. This example applies equally well across the entire show controller family.

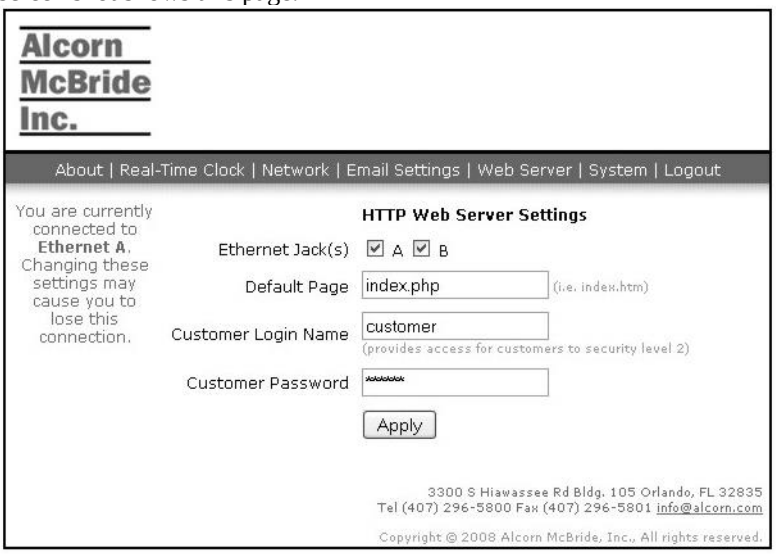
Step 1: Connecting to the Web Server

- Open a web browser client such as Internet Explorer or Firefox
- Enter the IP address of your show controller into the location bar. For example: http://192.168.0.254/
- The default web page will load as shown in the screen shot below.

- Enter the administrator user name and password and click "Login". The default user name is "admin" and the password is "password".
- If the login is successful, information about the show controller will be displayed including the serial number and firmware version.

Step 2: Configuration

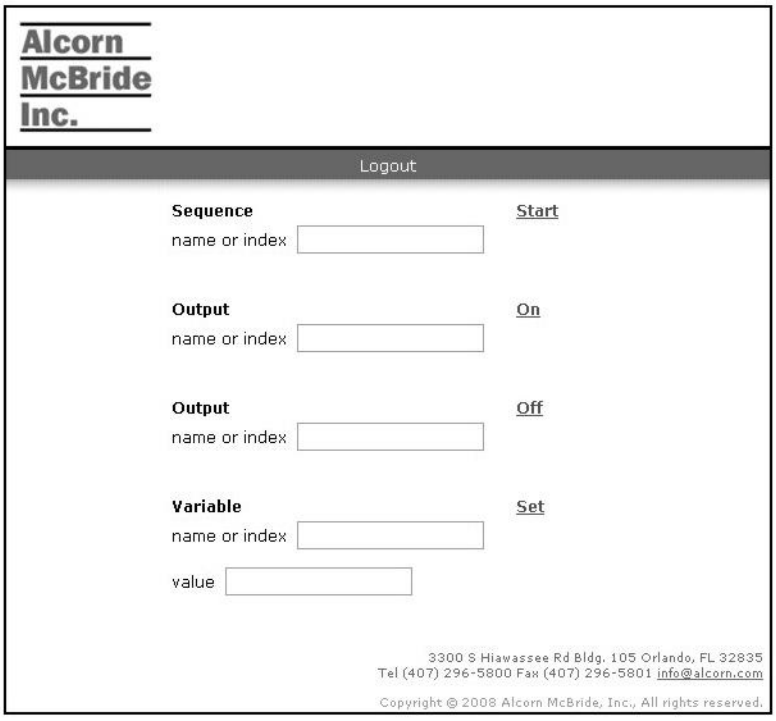
- Select the "Web Server" link to display web server settings. The following screen shot shows this page.



- Set the customer login name to "customer" or another name of your choice. Set the customer password. Leave the other settings as they are and click "Apply". These changes will occur immediately and there is no restart required.
- Select the "Logout" link to return to the login page.

Step 3: Customer Web Page

- Enter the customer user name and password and click "Login". The customer home page will load as shown below.



- This page demonstrates how to start sequences, turn on and off outputs, and set the value of variables in the currently running show. You can modify this page to make an easy to use interface for the customer.
- To modify this page, the file "home.php" on the Compact Flash card must be changed. You can do this by removing the card from your show controller and placing it into your computer's card reader.
- Copy the file "home.php" to your computer and open it in your favorite HTML editor.

Step 4: Understanding home.php

- The first few lines of home.php contain show controller web-script that is used to restrict access to this page to only the customer or administrator (see the section on Show Controller Web-Script). To remove this restriction, simply remove this:
  <?
  RequireLogin("2");
  ?>
- In the HTML head section, the script "ami.js" is included. This is required to use background processing of HTTP POST requests (see the section on Variables).
- The first form on this page is used to start sequences. It has a single text input item but a hidden item could be used instead with a preset value. This way, a link can be created that will always start a specific sequence. To perform the POST request in the background, the previously mentioned JavaScript is used for the action:
  javascript:AmiHttpPostForm('cmdstart.php', document.form1);

  This tells the web browser to post the form named "form1" to the file named "cmdstart.php". This file comes preloaded on the Compact Flash card and accepts a POST variable named "index", which is the name of the form item in the form.

  The link to:
  javascript:document.form1.submit()
  will submit the form and cause the sequence specified by the index to start. In this case, the index can also be the name of the sequence.

This ends the Quick Start guide to the show controller web server. The following sections describe the features of the web server in detail.

Hypertext Transfer Protocol

The show controller's have a built-in HTTP server for serving web pages to remote web clients. By default, the server is available on both Ethernet ports using TCP port 80. The server can be disabled from one or both ports to prevent unauthorized access to this service (see the section on Web Server Configuration).

File Names and Types

The show controller's HTTP server was designed to respond quickly and efficiently without interrupting the normal operation of the show controller. To ensure this, limits have been placed on file names and types.

All file names used with the HTTP server must be in 8.3 format meaning 8 characters followed by a '.' then followed by up to 3 characters for a file extension.

Only the following file types may be used and must end with the appropriate file extension as listed.

Type	Extension
Hypertext Markup Language (HTML)	htm
Joint Photographic Experts Group (JPEG)	jpg
Graphics Interchange Format (GIF)	gif
Portable Network Graphics (PNG)	png
JavaScript	js
Cascading Style Sheets (CSS)	css
Flash	swf
Web-Script	php

Show Controller Web-Script

The show controller's HTTP server supports a scripting language with a similar syntax to the widely used PHP language. This syntax is recognized by many modern HTML editors and therefore will not interfere with the design of a web page. Although the syntax is similar, the show controllers do not support the PHP language.

Web-Script Blocks

A single web-script file may have many web-script blocks but no single block may contain over 350 characters. This limit has been established to ensure that normal operation of the show controller is not interrupted by a web-script. The following table shows the characters used to define a web-script block.

Characters	Description
<?	start a script block
?>	end a script block

If Statements

If statements can be used to control whether commands within the web-script will be executed. In the examples below, the value of A and B may be a string, number, or variable.

Usage	Description

if(A == B)
{
}
	Compare two values and execute the code between the braces only if they are equal

if(A != B)
{
}
	Compare two values and execute the code between the braces only if they are not equal

Variables

There are three global variables: $_POST, $_ENV, and $_SERVER. These variables are used in a similar way to their PHP counterparts. The show controller web-script does not support local variables and does not allow the value of a variable to be changed directly (see the section on Functions).

The $_POST variable is used to access the value of data submitted to the server by an HTTP POST request. This is typically done using an HTML form, but can also be accomplished using AJAX (Asynchronous JavaScript and XML) or Flash.

The $_ENV variable is used to access the value of variables from the currently running show. These values can be accessed by the variable name or index.

The $_SERVER variable provides access to values stored within the show controller. A complete table of the values available from the server can be found below.

To access a value within any of these variables, the proper index must be used. If a value is posted using an HTML form, the index will be the name of the form item. For example, the value of a submit button named "submit" can be retrieved as $_POST["submit"]. An open and closed bracket always surrounds the index. The $_SERVER variable uses similar index names as follows:

$_SERVER index name	Description
jack	The Ethernet jack that is being used to connect to the HTTP server. This value is either "A" or "B"
level	The access level of the user that is currently logged in. 0 = guest 1 = administrator 2 = customer
Port A	
ipa	The IP address of port A (i.e. "192.168.0.254")
suba	The Subnet mask of port A (i.e. "255.255.255.0")
gwa	The Gateway address of port A (i.e. "192.168.0.1")
dnsa	The DNS server address of port A
dhcpa	The state of the DHCP client for port A "checked" = enabled "" = disabled
Port B	
ipb	The IP address of port B (i.e. "192.168.0.254")
subb	The Subnet mask of port B (i.e. "255.255.255.0")
gwb	The Gateway address of port B (i.e. "192.168.0.1")

$_SERVER index name	Description
dnsb	The DNS server address of port B
dhcpb	The state of the DHCP client for port B "checked" = enabled "" = disabled
NTP	
ntpa	The state of the NTP client for port A "checked" = enabled "" = disabled
ntpb	The state of the NTP client for port B "checked" = enabled "" = disabled
ntpserver	The address of the NTP server (i.e. "pool.ntp.org")
ntpdisabled	The state of the NTP client "checked" = disabled "" = enabled
Versions	
v16sn	The serial number of this V16Pro
v16ver	The firmware version of this V16Pro
smptever	The firmware version of the SMPTE module in this V16Pro
Date/Time	
date	The current date in the form "m/d/Y" m = month from 1 to 12 d = day from 1 to 31 Y = year (i.e. 2008)
time	The current time in the form "H:m:s" H = hours from 1 to 23 m = minutes from 0 to 59 s = seconds from 0 to 59
SMTP	
smtpa	The state of the SMTP client for port A "checked" = enabled "" = disabled
smtpb	The state of the SMTP client for port B "checked" = enabled "" = disabled
smtpserver	The address of the SMTP server (i.e. "smtp.example.com")
smtpport	The SMTP server port (default is 25)
smtpuser	The SMTP user name
smtppass	The SMTP user password
smtpfrom	The email address to send from

$_SERVER index name	Description
	example: <V16Pro> "email@address.com"
HTTP	
httpa	The state of the HTTP server for port A "checked" = enabled "" = disabled
httpb	The state of the HTTP server for port B "checked" = enabled "" = disabled
httppage	The name of the default HTTP server page
custname	The customer user name
custpass	The customer user password

Functions

The show controller server script has built-in functions that are used to configure settings, control the show, and display values. The following table lists all functions and their usage.

Name	Params	Description
Web-Script		
startseq	1. The name or index of the sequence to start	Start the specified sequence
setvareq	1. The name or index of the variable to set 2. The new value	Set the value of the specified variable
on	1. The name or index of the output to turn on	Immediately turn on the specified output
off	1. The name or index of the output to turn off	Immediately turn off the specified output
Display		
print	*	Output any number of params.
printhtml	*	Output the same as the print function but replace special characters with their HTML entities
HTTP Processing		
exit	None	Immediately stop processing the script
location	1. Name of the file to change location to	Immediately stop processing the script and tell the client to load the specified file. This only works when no data has been sent yet.
header	1. Complete header including line endings	Send the specified header immediately. This only

Name	Params	Description
		works if no data has been sent.
Access		
requirelogin	1. Minimum user level that is required to view the page	Use this function before any data is sent to ensure the user has the proper access level. See $_SERVER["level"] for access level values and their meanings.
login	1. user name 2. password	Use this function before any data is sent to allow a user to login. The login function requires cookies to be enabled in the user's browser. The $_SERVER["level"] value will be set immediately if login is successful.
logout	None	Use this function before any data is sent to logout. The $_SERVER["level"] value will be reset immediately and any login cookies that were previously set will be cleared.
SMTP Settings		
setsmtpjack	1. The jack to select for sending email. Either "A" or "B"	Set the SMTP jack to send email from.
setsmtpserver	1. The server name or ip address 2. The server port (use 25 if unsure)	Set the SMTP server to use for sending email.
setsmtpauth	1. The user name 2. The password	Set the SMTP Auth user name and password
setsmtpfrom	1. The email address to send from	Set the email address to send from
HTTP Settings		
sethttpjacks	1. Jack A. Set to "enabled" or "disabled" 2. Jack B. Set to "enabled" or "disabled"	Set which Ethernet jacks, if any, should be enabled for HTTP.
sethttppage	1. The file name	Set the file name of the

Name	Params	Description
		page to load when no page is specified by a request
setcustauth	1. The customer user name 2. The customer password	Set the customer user name and password for the HTTP server login
Date/Time Settings		
settime	1. The time. See $_SERVER["time"] for formatting.	Set the current time
setdate	1. The date. See $_SERVER["date"] for formatting.	Set the current date
setntpjack	1. The Ethernet jack. Set to "A", "B", or "disabled"	Set the Ethernet jack to use for NTP
setntpserver	1. The ntp server name or IP address	Set the NTP server name or IP address
Network Settings		
setdns	1. The Ethernet port. Set to "A" or "B" 2. IP address of the DNS server	Set the DNS server for the specified Ethernet port
setnetwork	1. The Ethernet port. Set to "A" or "B" 2. IP address 3. Subnet Mask 4. Gateway	Set the IP Address, Subnet Mask, and Gateway for the specified Ethernet port
setdhcp	1. The Ethernet port. Set to "A" or "B" 2. Set to "enabled" or "disabled"	Set whether to use DHCP for the network and DNS settings instead of static values for the specified Ethernet port
System		
restart	None	Immediately restart the show controller
savesettings	None	Save all changes to Settings so that they will remain the next time the show controller is restarted.

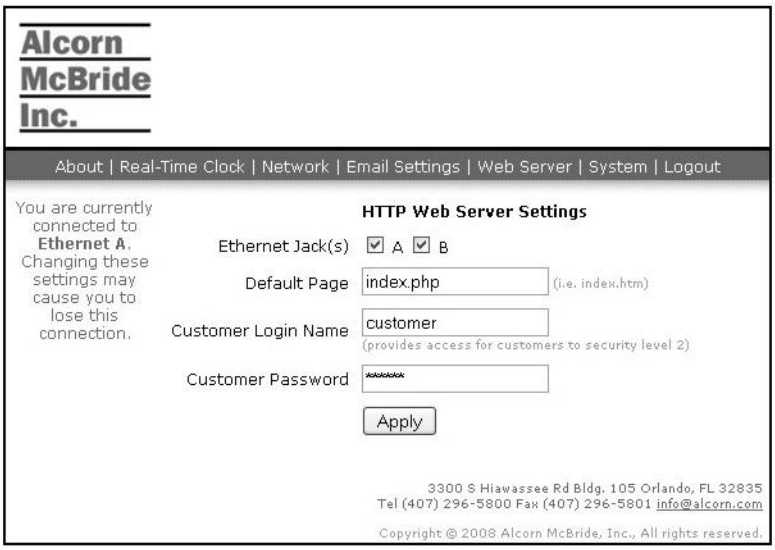
Function Params

A function parameter can be a string, number, variable, or combination thereof. The following table defines these parameter types

Type	Usage
String	A string is any combination of double-quoted values and hex values. Double-quoted values may include escape characters using a backslash \ Valid escape characters are: \r – carriage return \n – new line \t – tab \" – double-quotation mark \\ – backslash \x – hex where the x is followed by two ASCII hex characters. For example, \x35 represents the ASCII character '5'.
Number	Numbers include any whole number
Variable	See the section on Variables
Combination	To combine two or more values together for a single parameter, use the concatenation operator. The concatenation operator is a single period "." For example, to output a link to a page on the web server using Ethernet port A, use the print function as follows: <a href="<? print("http://" . $_SERVER["ipa"] . "/newpage.htm"); ?>">New Page Using the default IP address, this will result in a link to: http://192.168.0.254/newpage,htm

Web Server Configuration

The web server can be enabled or disabled on any of the Ethernet ports. Follow the Quick Start section to login to the administrator web pages and display the Web Server Settings as shown:



Ethernet Jack(s) – use these checkboxes to enable or disable the web server on the specified ports.

Default Page – this value specifies the page that will load when no page is specified by an HTTP request

Customer Login Name – a login that only provides access to web pages for the customer. This name and password cannot access show controller settings unless web pages are created specifically for this purpose.

Customer Password – the password for the customer login

Serial and Ethernet Control

The show controllers can be controlled through the RS232 programmer port, USB port or Ethernet ports A or B. Any controller that is capable of sending ASCII characters is capable of controlling the show controller using the set of serial commands described below.

Command set

All the commands sent to the show controllers are two characters, ending with a carriage return <0D>. The commands may have the Get/Set function and in those cases the user supplies the optional information. In this case the additional information is placed before the command followed by the two-character command then a carriage return. If the optional information is not supplied, the command will return the current data in the form of a string of ASCII characters as a response.

For the most part the command set is not usually needed unless the WinScript Live application is not going to be used. These commands give the user access to the remote control aspect of the show controller. The command structure is a terminal like interface, where the terminal sends a command and the show controller will respond with the required data. It is not generally recommended as a generic interface but hooks into some other environment.

Some of the commands will setup a data stream that will continue until told to stop, such as Live Mode command when told to monitor a variable.

?V Get Firmware Version

Description: This command will return the controller firmware version number.

Command: **?V<0D>**

Message Response: **V16Pro Vx.xx<0D>**

Example: Send Command: **?V<0D>**

Response: **V16Pro V1.23<0D>**

?S Get SMPTE Firmware Version

Description: This command will return the firmware version of the SMPTE module.

Comments: It is possible for the SMPTE process not to send status . If this occurs, this command will return a hardware error (E01).

Command: **?S<0D>**

Message Response: **SMPTE vx.xx<0D>**

Example: Send Command: **?S<0D>**

Response: **v1.23<0D>**

ES Enable SMPTE

Description: This command will enable SMPTE module.

Comments: If the SMPTE module is configured to Generate, this command will cause the clock to start generating at the configured Preroll time. If the SMPTE module is configured to Read, it will start listening for SMPTE time code on the SMPTE Input. If the SMPTE clock is in a paused state, this command will cause it to resume from its current position.

Get command: **ES<0D>**

Message Response: **R<0D>**

Example: Send Command: **ES<0D>**
Response: **R<0D>**

DS Disable SMPTE

Description: This command will disable the SMPTE module

Comments: If the SMPTE module is configured to Generate, this command will cause the SMPTE clock to stop at its current time. If the SMPTE module is configured to Read, this command will cause the SMPTE clock to stop running and ignore any incoming time code.

Get command: **DS<0D>**

Message Response: **R<0D>**

Example: Send Command: **DS<0D>**
Response: **R<0D>**

PS Pause SMPTE (Next Loop Point)

Description: This command will STOP the SMPTE at the next loop point.

Comments: This command only applies when the SMPTE module is in Generate mode and is also configured to loop. When paused, the SMPTE clock can be resumed by sending an Enable SMPTE command.

Get command: **PS<0D>**

Message Response: **R<0D>**

Example: Send Command: **PS<0D>**
Response: **R<0D>**

IS Pause SMPTE (Immediately)

Description: This command will pause the SMPTE clock immediately.

Comments: When paused, the SMPTE clock can be resumed by sending an Enable SMPTE command.

Get command: **IS<0D>**

Message Response: **R<0D>**

Example: Send Command: **IS<0D>**
Response: **R<0D>**

CT Get/Set SMPTE Time

Description: This command will get or set the current SMPTE time. If the parameter is excluded the command is executed as a get command.

Comments:

hh	= hours	mm	= minutes
ss	= seconds	ff	= frames

For now, this function must use two digits in each field even if it is zero(0) and the delimiters between each field must be followed. hh:mm:ss.ff

Get command: **CT<0D>**

Message Response: **hh:mm:ss.ff<0D>**

Example: Send Command: **CT<0D>**
Response: **00:01:59.29<0D>**

Set command: **hh:mm:ss.ff CT<0D>**

Message Response: **R<0D>**

Example: Send Command: **00:01:59.29CT<0D>**
Response: **R<0D>**

ID Get/Set Unit ID

Description: This command will get or set the unit ID number.

Comments: The ID is used to identify the unit in a shared serial multi-drop line configuration. Where xx is the unit ID in the range 0-49.

Get command: **ID<0D>**

Message Response: **(0-49)<0D>**

Example: Send Command: **ID<0D>**
Response: **0<0D>**

Set command: (**0-49)ID<0D>**

Message Response: **R<0D>**

Example: Send Command: **1ID<0D>**
Response: **R<0D>**

IP Get/Set IP address

Description: This command will get or set the selected port IP address. Port A or B may be selected

Comments: Where xxx is a decimal number in the range of 0 – 255.

Get command: **(A or B)IP<0D>**

Message Response: **xxx.xxx.xxx.xxx<0D>**

Example: Send Command: **AIP<0D>**
Response: **192.168.0.254<0D>**

Set command: **xxx.xxx.xxx.xxx(A or B)IP<0D>**

Message Response: **R<0D>**

Example: Send Command: **192.168.0.254AIP<0D>**
Response: **R<0D>**

SM Get/Set Subnet Mask number

Description: This command will get or set the number used to isolate the subnet.

Comments: xxx is a decimal number in the range of 0 – 255.

Get command: **SM<0D>**

Message Response: **xxx.xxx.xxx.xxx<0D>**

Example: Send Command: **SM<0D>**
Response: **255.255.255.0<0D>**

Set command: **xxx.xxx.xxx.xxxSM<0D>**

Message Response: **R<0D>**

Example: Send Command: **255.255.255.0SM<0D>**
Response: **R<0D>**

GW Get/Set Gateway IP Address

Description: This command will get or set the IP address of the gateway the unit will connect through.

Comments: xxx is a decimal number in the range of 0 – 255.

Get command: **GW<0D>**

Message Response: **xxx.xxx.xxx.xxx<0D>**

Example: Send Command: **GW<0D>**
Response: **192.168.0.1<0D>**

Set command: **xxx.xxx.xxx.xxxGW<0D>**

Message Response: **R<0D>**

Example: Send Command: **192.168.0.1GW<0D>**
Response: **R<0D>**

DA Get/Set Date

Description: This command will get or set the calendar date.

Comments: mm/dd/yyyy Month/Day/Year.

Get command: **DA<0D>**

Message Response: **mm/dd/yyyy<0D>**

Example: Send Command: **DA<0D>**
Response: **11/15/2008<0D>**

Set command: **mm/rr/yyyyDA<0D>**

Message Response: **R<0D>**

Example: Send Command: **12/15/2008DA<0D>**
Response: **R<0D>**

TI Get/Set Time

Description: This command will get or set the time of day.

Comments: hh:mm:ss Hours/Minutes/Seconds.

Get command: **TI<0D>**

Message Response: **hh:mm:ss<0D>**

Example: Send Command: **TI<0D>**
Response: **11:59:59<0D>**

Set command: **hh:mm:ssTI<0D>**

Message Response: **R<0D>**

Example: Send Command: **11:59:59TI<0D>**
Response: **R<0D>**

US Get/Set User Name

Description: This command will get or set the user login name.

Comments: The default name is admin.

Get command: **US<0D>**

Message Response: **(current login)<0D>**

Example: Send Command: **US<0D>**
Response: **admin<0D>**

Set command: **(new login)US<0D>**

Message Response: **R<0D>**

Example: Send Command: **adminUS<0D>**
Response: **R<0D>**

PW Get/Set Password

Description: This command will get or set the password.

Comments: The default password is password.

Get command: **PW<0D>**

Message Response: **password<0D>**

Example: Send Command: **PW<0D>**
Response: **currentpassword<0D>**

Set command: **(new password)PW<0D>**

Message Response: **R<0D>**

Example: Send Command: **passwordPW<0D>**
Response: **R<0D>**

SD Get/Set DST Enable

Description: This command will get or set the status of the daylight saving time parameter.

Comments: 0 = Standard Time, 1 = using DST.

Get command: **SD<0D>**

Message Response: **current setting<0D>**

Example: Send Command: **SD<0D>**
Response: **1<0D>**

Set command: **1SD<0D>**

Message Response: **R<0D>**

Example: Send Command: **1SD<0D>**
Response: **R<0D>**

DT Get/Set DST Type

Description: This command will get or set the status of the daylight saving time Type.

Comments: Example are: 1 = US, 2 = Universal, 3 = Australia, 4 = Europe.

Get command: **DT<0D>**

Message Response: **current setting<0D>**

Example: Send Command: **DT<0D>**
Response: **1<0D>**

Set command: **1DT<0D>**

Message Response: **R<0D>**

Example: Send Command: **1DT<0D>**
Response: **R<0D>**

TZ Get/Set Time Zone

Description: This command will get or set the Time Zone.

Comments: The time zone 0 is the Greenwich Mean Time GMT. Time zones going west are given –1 numbers and +1 going east from GMT 0 until the International Date Line is reached. Eastern TZ in the USA is -5

Get command: **TZ<0D>**

Message Response: **current timezone<0D>**

Example: Send Command: **TZ<0D>**
Response: **-5<0D>**

Set command: **-5TZ<0D>**

Message Response: **R<0D>**

Example: Send Command: **-5TZ<0D>**
Response: **R<0D>**

DI Display Text

Description: This command will place user text on the display screen at the row and column specified in the command.

Comments: r = 1 to 8, cc = 1 to 42.

Command: **"Display Text"|r|ccDI<0D>**

Message Response: **R<0D>**

Example: Send Command: **"Display Text"|4|10DI<0D>**
Response: **R<0D>**

LO Get/Set Longitude Coordinates

Description: This command will get or set the longitude coordinates.

Get command: **LO<0D>**

Message Response: **current value<0D>**

Example: Send Command: **LO<0D>**
Response: **81.0<0D>**

Set command: **81.0LO<0D>**

Message Response: **R<0D>**

Example: Send Command: **81.0LO<0D>**
Response: **R<0D>**

LA Get/Set Latitude Coordinates

Description: This command will get or set the latitude coordinates.

Get command: **LA<0D>**

Message Response: **current value<0D>**

Example: Send Command: **LA<0D>**
Response: **28.0<0D>**

Set command: **28.0LA<0D>**

Message Response: **R<0D>**

Example: Send Command: **28.0LA<0D>**
Response: **R<0D>**

VA Get/Set a Variable

Description: This command will get or set the variable.

Comments: The variable name located at the far left of the sequence form. The user assigns the label name.

Get command: **var1VA<0D>**

Message Response: **(value of var1)<0D>**

Example: Send Command: **var1VA<0D>**
Response: **Hello World<0D>**

Set command: **var1|"new value for var1"VA<0D>**

Message Response: **R<0D>**

Example: Send Command: **var1|"Hello World"VA<0D>** Response: **R<0D>**

VT Toggle a Boolean Variable

Description: This command will change the variable state to the opposite state

Comments: 0 to 1, 1 to 0, Off to On, On to Off.

Command: **var1VT<0D>**

Message Response: **R<0D>**

Example: Send Command: **var1VT<0D>**
Response: **R<0D>**

RJ Reset Sequence

Description: This command will reset a sequence to the start.

Comments: If the sequence is not triggered or setup to loop the sequence will be in a waiting condition.

Command: (s**equence number)RJ<0D>**

Message Response: **R<0D>**

Example: Send Command: **1RJ<0D>**
Response: **R<0D>**

PA Pause a Sequence

Description: This command will cause the executing sequence to pause. The sequence is identified by its number.

Comment: The sequence number is the first field in the form. WinScript Live assigns the number.

Command: **(sequence number)SD<0D>**

Message Response: **R<0D>**

Example: Send Command: **1SD<0D>**
Response: **R<0D>**

SL Stop a Looping Sequence

Description: This command will stop a looping sequence. The sequence is identified by its number.

Comments: The sequence number is the first field in the form. WinScript Live assigns the number.

Command: **(sequence name or number)SL<0D>**

Message Response: **R<0D>**

Example: Send Command: **mySeqSL<0D>**
Response: **R<0D>**

PL Run a Sequence

Description: This command will run a sequence from the currently selected script. The sequence is identified by its number.

Comments: The sequence number is the first field in the form. WinScript Live assigns the number.

Command: **(sequence name or number)PL<0D>**

Message Response: **R<0D>**

Example: Send Command: **mySequencePL<0D>**
Response: **R<0D>**

SQ Get Sequence Status

Description: This command will get the status of the selected sequence. The sequence is identified by its number.

Comments: The sequence number is the first field in the form. WinScript Live assigns the number. Running Stopped or Paused are the responses.

Get command: **(sequence name or number)SQ<0D>**

Message Response: **status<0D>**

Example: Send Command: **1SQ<0D>**
Response: **Running<0D>**

OU Output Control

Description: This command will control one of the outputs off, on and toggle.

Comments: 0 = OFF, 1 = ON, 2 = toggle.

Command: **(channel or output Name)(command)OU<0D>**

Message Response: **R<0D>**

Example: Send Command: **myOutput1OU<0D>** (turns myOutput ON)

Response: **R<0D>**

SS Send Message

Description: This command will send a message to the selected port.

Comments: Any valid port may be used.

Command: **(port)|"message test"SS<0D>**

Message Response: **R<0D>**

Example: Send Command: **Sport1|"Hello World"SS<0D>**
Response: **R<0D>**

XX Reboot

Description: This command will perform a hard reboot of the system.

Comments: The show controller will reload and run the selected script. The normal power on response will apply

Command: **XX<0D>**

Message Response: **K<0D>**

Example: Send Command: **XX<0D>**
Response: **K<0D> (after a rebooting only)**

NI Get/Set NTP IP Address

Description: This command will get or set the NTP IP address needed to contact the timeserver.

Comments: The default IP address is 068.216.79.113. Other examples are Boulder Colorado US is 132.163.4.101 Europe is 213.251.169.205

Get command: **NI<0D>**

Message Response: **(current NTP address)<0D>**

Example: Send Command: **NI<0D>**
Response: **68.216.79.113<0D>**

Set command: **"NTP IP address"NI<0D>**

Message Response: **R<0D>**

Example: Send Command: **"68.216.79.113"NI <0D>**
Response: **R<0D>**

NE Enable/Disable the NTP Function

Description: This command will enable or disable the NTP function and will get the current status.

Comments: 0 = disable, 1 = enable

Get command: **NE<0D>**

Message Response: **(current status)<0D>**

Example: Send Command: **NE<0D>**
Response: **1<0D>**

Set command: **(0 or 1)NE<0D>**

Message Response: **R<0D>**

Example: Send Command: **1NE<0D>**
Response: **R<0D>**

NJ Get/Set the Ethernet Port for NTP

Description: This command will get or set the show controller Ethernet port used to contact the NTP server.

Comments: A = port A, B = port B. The default port is A

Get command: **NJ<0D>**

Message Response: **(current port)<0D>**

Example: Send Command: **NJ<0D>**
Response: **A<0D>**

Set command: **(port A or B)NJ<0D>**

Message Response: **R<0D>**

Example: Send Command: **ANJ<0D>**
Response: **R<0D>**

TS Time Stamp

Description: This command will attach the time to the active script.

Comments: Used in live mode to compare scripts.

Command: **TS<0D>**

Message Response: **-5<0D>**

Example: Send Command: **TS<0D>**
Response: **-5<0D>**

SF Get/Set Active Script file

Description: This command will get the current script file or set the script to be used by the show controller.

Comments: If there are multiple scripts on the CF card the user may select one to become active.

Get command: **SF<0D>**

Message Response: **(current file name) <0D>**

Example: Send Command: **SF<0D>**
Response: **sequences1.ami<0D>**

Set command: **"scriptname.ami"SF<0D>**

Message Response: **R<0D>**

Example: Send Command: **"sequences1.ami"SF<0D>**
Response: **R<0D>**

NM Get/Set Device Name

Description: This command will get the current device name or set the name of the device.

Comments: This can be used to "Name" your V16Pro for reference when you're looking at the device list in Live Mode.

Get command: **NM<0D>**

Message Response: **(current device name) <0D>**

Example: Send Command: **NM<0D>**
Response: **deviceName<0D>**

Set command: **"deviceName"SF<0D>**

Message Response: **R<0D>**

Example: Send Command: **"Device1"SF<0D>**
Response: **R<0D>**

FT Get Script Edit Date

Description: This command will get the 'last edited' date of the active script on the device, or another non-active script on the device.

Comments: This can be useful to see when the script on the show controller was last updated.

Get command: **FT<0D>**

Message Response: **yyyy-mm-dd hh:mm:ss<0D>**

Example: Send Command: **FT<0D>**
Response: **2009-02-13 23:31:30<0D>**

Get command: **"scriptname.ami"FT<0D>**

Message Response: **yyyy-mm-dd hh:mm:ss<0D>**

Example: Send Command: **"sequences1.ami"FT<0D>**
Response: **2013-02-14 15:10:54<0D>**

DH Enable/Disable the DHCP Function

Description: This command will enable or disable the DHCP function and will get the current status.

Comments: x is 0 = disable, 1 = enable

p is A = port A, B = port B

Get command: **pDH<0D>**

Message Response: **x<0D>**

Example: Send Command: **ADH<0D>** (returns port A status)
Response: **1<0D>**

Set command: **xpDH<0D>**

Message Response: **R<0D>**

Example: Send Command: **1ADH<0D> (**enables port A)
Response: **R<0D>**

SJ Get/Set the Ethernet Port for SMTP

Description: This command will get or set the show controller Ethernet port used to contact the SMTP server.

Comments: A = port A, B = port B. The default port is A

Get command: **SJ<0D>**

Message Response: **(current port)<0D>**

Example: Send Command:**SJ<0D>**
Response: **A<0D>**

Set command: **(port A or B)SJ<0D>**

Message Response: **R<0D>**

Example: Send Command: **ASJ<0D>**
Response: **R<0D>**

SA Get/Set the SMTP Address

Description: This command will get or set the SMTP address.

Comments: A text string such as mail.alcorn.com

Get command: **SA<0D>**

Message Response: **(current address)<0D>**

Example: Send Command: **SA<0D>**
Response: **mail.alcorn.com<0D>**

Set command: (new address**)SA<0D>**

Message Response: **R<0D>**

Example: Send Command: **mail.alcorn.comSA<0D>**
Response: **R<0D>**

SP Get/Set the SMTP Port

Description: This command will get or set the SMTP port number needed to contact the SMTP server.

Comments: 0 to 65535. The default port number is 578

Get command: **SP<0D>**

Message Response: **(current port)<0D>**

Example: Send Command:**SP<0D>**
Response: **A<0D>**

Set command: **(new port)SP<0D>**

Message Response: **R<0D>**

Example: Send Command: **587SP<0D>**
Response: **R<0D>**

SU Get/Set the SMTP User Login Name

Description: This command will get or set the SMTP name used when logging into the account.

Comments: A text string such as mylogin

Get command: **SU<0D>**

Message Response: **(current name)<0D>**

Example: Send Command: **SU<0D>**
Response: **mylogin<0D>**

Set command: **(new name)SU<0D>**

Message Response: **R<0D>**

Example: Send Command: **mynewloginSU<0D>**
Response: **R<0D>**

SW Get/Set the SMTP Password

Description: This command will get or set the SMTP password needed to log onto the SMTP server.

Comments: The default is password. Get returns E current password is not returned.

Get command: **SW<0D>**

Message Response: **E<0D>**

Example: Send Command:**SW<0D>**
Response: **E<0D>**

Set command: **(NewPassword)SW<0D>**

Message Response: **R<0D>**

Example: Send Command: **NewPasswordSW<0D>**
Response: **R<0D>**

FR Get/Set the SMTP From Name

Description: This command will get or set the SMTP name used in the from-field of the E-Mail.

Comments: A text string such as mylogin

Get command: **FR<0D>**

Message Response: **(current name)<0D>**

Example: Send Command: **FR<0D>**
Response: **mylogin<0D>**

Set command: (**current name)FR<0D>**

Message Response: **R<0D>**

Example: Send Command: **current nameFR<0D>**
Response: **R<0D>**

MA Send E-Mail

Description: This command will send the e-mail

Comments: <to>|<subject>|<textmessage>

Command: <To e-mailaddress>|<Subject>|<message>**MA<0D>**

Message Response: **R<0D>**

Example:

Send Command: **control@MyShow.com|Show Status|Main show went to day modeMA<0D**

Response: **R<0D>**

HJ Get/Set the HTTP Ethernet Port

Description: This command will get or set the Ethernet port that will be used in connecting to the Internet

Comments: p = A/B Ethernet jack locations on the show controller

Get command: **HJ<0D>**

Message Response: **x<0D>**

Example: Send Command: **HJ<0D>**
Response: **A<0D>**

Set command: **(A or B)HJ<0D>**

Message Response: **R<0D>**

Example: Send Command: **AHJ<0D>** (enables port A)
Response: **R<0D>**

HP Get/Set the HTTP WEB Page

Description: This command will get or set the default HTTP WEB address.

Comments: The default WEB page is index.html

Get command: **HP<0D>**

Message Response: **(currentwebpage)<0D>**

Example: Send Command: **HP<0D>**
Response: **myWebPage.html<0D>**

Set command: **(myWebPage.html)HP<0D>**

Message Response: **R<0D>**

Example: Send Command: **myWebPage.htmlHP<0D>**
Response: **R<0D>**

RI Get/Set Redundant IP Address

Description: This command will get or set the Redundant IP address.

Comments: "x" represents a number in the IP.

Get command: **RI<0D>**

Message Response: **(xxx.xxx.xxx.xxx)<0D>**

Example: Send Command: **RI<0D>**
Response: **xxx.xxx.xxx.xxx<0D>**

Set command: **(xxx.xxx.xxx.xxx)RI<0D>**

Message Response: **R<0D>**

Example: Send Command: **xxx.xxx.xxx.xxxRI<0D>**
Response: **R<0D>**

RX Get/Set Redundant Ethernet Jack

Description: This command will get or set the Redundant Ethernet Jack.

Comments: "n" represents either A or B

Get command: **RX<0D>**

Message Response: **(n)<0D>**

Example: Send Command: **RX<0D>**
Response: **A<0D>**

Set command: **(n)RX<0D>**

Message Response: **R<0D>**

Example: Send Command: **ARX<0D>**
Response: **R<0D>**

MS Get/Set Master Slave Message

Description: This command will get or set the Master Slave Message.

Comments: "1" represents force to slave, "2" represents force to master, and "0" represents force to be stand alone (disabled).

Get command: **MS<0D>**

Message Response: **(0|1|2)<0D>**

Example: Send Command: **MS <0D>**
Response: **2<0D>**

Set command: **(0|1|2)MS <0D>**

Message Response: **R<0D>**

Example: Send Command: **2MS <0D>**
Response: **R<0D>**

TMS Get/Set Master Slave Timeout Period

Description: This command will get or set the Master Slave Timeout Period. Time delay before assuming master is gone.

Comments: "nnn" represents the time in milliseconds.

Get command: **TMS<0D>**

Message Response: **(nnn)<0D>**

Example: Send Command: T**MS <0D>**
Response: **5000<0D>**

Set command: **(nnn)TMS <0D>**

Message Response: **R<0D>**

Example: Send Command: **5000TMS <0D>**
Response: **R<0D>**

JP Jump to Timecode Message.

Description: This command jumps a sequence to a specific point in time.

Comments: Does not change SMPTE synced sequences

Command: **<SequenceName>|<timecode>JP<0D>**

Message Response: R**<0D>**

Example: Send Command: T**MS <0D>**
Response: **5000<0D>**

Set command: **(nnn)TMS <0D>**

Message Response: **R<0D>**

Example: Description: **Jump sequence "MySequence" to timecode 00:00:01.00**

Send Command: **MySequence|00:00:01.00JP<0D>**

Response: **R<0D>**

LV Live Mode

Description: This command is the Live mode initiator. The live mode starts a real time interface between the host system and the show controller. Information is requested by the following protocol.

Comments: Live mode communication records take the following format. b|t|i|dLV

b = 0/1, Don't watch or turn off / Watch or turn on

t = an item from the list below

i = index of the item in the list example input7 = 7|6 (0 as the first item)

d = device index connected to the show controller denoted by an index number assigned by the device table in WinScript Live list of devices (0 is the show controller)

Command: bool|type|index|device|**LV<0D>**

Message Response: **R<0D>**

Example: Send Command: **1|7|6|0LV<0D>** (watch input7)
Response: **R<0D>**

boolean	2
integer	3
decimal	4
string	5
variable	6
input	7
output	8
button	9
sequence	10
percent	11
label	12
timecode	13
lcdstring	14
date/time	15

For more detail, see "Live Mode Protocol" section in the appendix of this manual.

EX Execute a command

Description: This command will run any of the commands that can be used in a sequence.

Comments: All command parameters must be supplied as if in the WinScript Live form and separated with the pipe "|" character. If your unsure if the fields needed enter the command in WinScript Live and verify the fields needed

Command: **device|event|data1|data2EX<0D>**

Message Response: **R<0D>**

Example: Send Command: **V16Pro|On|output1EX<0D>**

Response: **R<0D>**

Additional Notes: If WinScript Live is not used as the script editor, there is a way to speed up the interpretation of the data fields needed to execute the sequence commands. The above command line is written as: V16Pro|On|output1[8]EX<0D> [x] points the show controller to use the parallel output1 instead of something else called by the same label. Parameter types for the [x] data fields are as follows:

boolean	2
integer	3
decimal	4
string	5
variable	6
input	7
output	8
button	9
sequence	10
percent	11
label	12
timecode	13
lcdstring	14
date/time	15

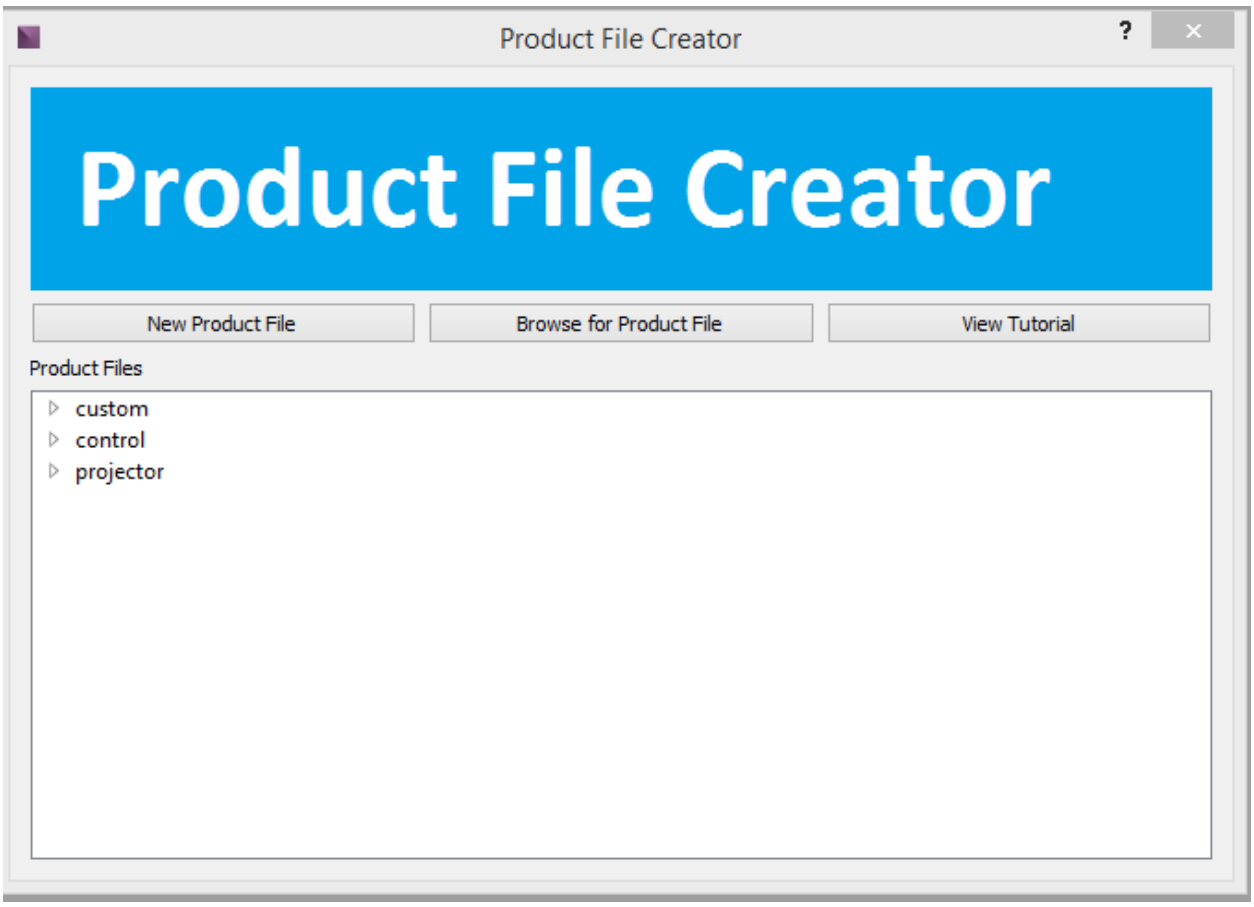
Product File Creator

The "**Product File Creator**" tool found under the **"Tools" menu of WinScriptLive** can be used to get started creating a product file. This can be used for adding new product files to WinScriptLive.

The following tutorial is only in reference to creating a product file through the use of this tool built into WinScriptLive. For more advanced product file creation and editing in XML, go to http://www.alcorn.com/library/manuals/man_v16proXMLProdFiles.pdf for the document **"Creating/Editing Your Own Product File via XML."**

Product File Creator Tutorial

Go to **Tools→Product File Creator** and on the opening window click "**View Tutorial**".



Live Mode Protocol

Live Mode communication is used by the show controllers to give updated information on the state of variables, inputs, outputs and other resources. This information is not polled, but instead requested once and then the show controller will send updated information as that resource changes. This method is used by Touch to retrieve status information and can also be used by third party drivers.

Note: All messages described in this section (such as "1LV") will be printable ascii messages and will always have a carriage return (hex 0D) at the end of the message.

Timing Information

Live information will be sent as often as possible, but there is no guarantee of frame accuracy. Live mode communication is considered a "Low Priority", and can be delayed by several frames if there are higher priority items to be done (like processing incoming ethernet messages, running a sequence, etc). Typical delay will be between 1 and 5 frames. (1 frame ~33ms).

Connection Information

The connection is on UDP port 2638 or 2639. To initiate a Live Mode connection, the show controller must be fully booted and a script must be running. All messages end in a carriage return (hex 0D).

To start the connection, send: 1LV
To stop the live connection, send: 0LV

A "heartbeat" packet must be sent to the controller every 2 seconds or less.

Heartbeat packet: 3LV

Heartbeat packets will not be acknowledged by the show controller.

After a connection is established using the above method, the show controller will send a heartbeat response packet of: LV

This "LV" heartbeat response packet will be sent approximately every 1 second. This packet is only sent if no other resource response packets are being sent from the show controller. In other words, heartbeat response packets will be sent if no watched resources have changes that need to be sent.

Sequence Status

The sequence status (stopped, paused, running) is always sent when a live connection is established.
It is sent in the following format:

s|<index of sequence>|<sequence state>|<frames>|<event1>|<exent2>.....|<eventN>|LV

s - the ASCII letter 's'
<index of sequence> - the zero indexed number of the sequence in the "sequences" view of WinScriptLive
<sequence state> - The sequence running state: Stopped =0, Running = 1, Paused = 2
<frames> - The sequence's current time (in frames)
<event1>....<eventN> - The event indexes most recently executed by the sequence.

Note: <frames> and <event> items are sent if the show controller has received a "resource request" for that particular sequence. Otherwise, only the sequence state is sent along with "0" for <frames>.

Resource Requests

To request the status of a variable, input, output, or sequence, use the following format. (All messages end in a carriage return 0x0D).

<type>|<bool>|<index>|<device>|<listIndex>LV

- **<type>** = single ascii character the item type (see Resource Type Lookup)
- **<bool>** = 0/1, Don't watch or turn off / Watch or turn on
- **<index>** = zero indexed resource number: Example input7 = 6
- **<device>** = zero indexed device number (for device variables, inputs, etc.) from the "devices" table in WinScript
- **<listIndex>** = if this variable is a "list" type variable (array), the position in the array to retrieve. Zero indexed only if the variable itself uses zero indexing, otherwise 1 indexed.

If <listindex> is unused, use "-1" as a placeholder. If <listIndex> does not apply (in the case of inputs, outputs, etc), remove from command.

Currently there is no ability to watch only a specific bit of an integer variable. If watching a specific bit, watch the entire integer variable and filter on the application/PC side.

Resource Type Lookup

Sequence	s
Variable	v
Input	i
Output	o
Button	b
Display	d
Event	e
Device	c
Watch	w
Trigger	t
Analog Input	a
Analog Output	g
Error (see Error Section for more details)	z
User Log Message (Log Message Event)	u

Example Resource Requests

Watch "Input5"

Command: i|1|4|0LV
Message Ack Response: R

Watch Sequence #100 (To get frame counter and events recently executed)
Command: s|1|99|-1|-1LV
Message Ack Response: R

Watch V16Pro's integer variable V16Pro.Hours (variable #12 in V16pro's "Device Variables" list)

Command: v|1|11|0|-1LV
Message Ack Response: R

Watch V16Pro's Timecode variable V16Pro.LTC (variable #1 in V16pro's "Device Variables" list)

Command: v|1|0|0|-1LV
Message Ack Response: R

Watch a string variable named "myVar" (variable #8 in "User Variables")

Command: v|1|7|-1|-1LV
Message Ack Response: R

Stop Watching a string variable named "myVar" (variable #8 in "User Variables")

Command: v|0|5|-1|-1LV
Message Ack Response: R

Resource Status

Resource status is returned in the following format:

<type>|<index>|<device index>|<value>LV

- **<type>** = single ascii character the item type (see Resource Type Lookup)
- **<index>** = zero indexed resource number: Example input7 = 6
- **<device>** = zero indexed device number (for device variables, inputs, etc.) from the "devices" table in WinScript

For **Variables**, the status is returned as:

<type>|<index>|<device index>|<value>|<listIndex>LV

- **<listIndex>** = if this variable is a "list" type variable (array), the position in the array to retrieve. Zero indexed only if the variable itself uses zero indexing, otherwise 1 indexed.

Resource Status Examples

V16Pro's LTC Timecode Variable #1 is 00:00:00.02:
v|0|0|00:00:00.02|-1LV

Device #8 variable number #100 is 1234
v|99|7|1234|-1LV

User variable number #100, list (array) position 8 is 456
v|99|-1|456|7LV

Input #5 status is "Off"
i|4|-1|0LV

Input #5 status is "On"
i|4|-1|1LV

Resource Status Frequency

Resource status is sent immediately following a "resource request" for that resource, or whenever that resource's value has changed.

In the case of sequences, the sequence status is always sent when the state (paused, stopped, running) has changed.

Error Reporting

Error status is returned in the following format:

<type>|<error type>|<command index>|<device index>|<sequence name>LV

- **<type>** = single ascii character the item type (see Resource Type Lookup)
- **<error type>** = single ascii character the error type (see Error Type Lookup)
- **<command index>** = the index in the device's protocol file
- **<device index>** = the index of the device in the script
- **<sequence name>** = if applicable, the sequence name this command was used in

Error Type Lookup

Unsolicited Message Error	u
Timeout	t
Connection Timeout (tcp)	c
Invalid Response to Command	i
Error Response Received From Device	e

User Log Message

Error status is returned in the following format:

<type>|<level>|<message>LV

- **<type>** = single ascii character the item type (see Resource Type Table)
- **<level>** = "information", "warning" or "error (see Log Type Table)
- **<message>** = the actual message information

Log Type

Information	i
Warning	w
Error	e

Name Lookup

Note: Resource indexes can be queried by using the "Name Lookup" command (NL):

<resource name>NL<h0d>

The results returned in the form:
<type>|<index>|<device>|<listIndex>NL<h0D>

- **<type>** = single ascii character the item type (see Resource Type Lookup)
- **<index>** = zero indexed resource number: Example input7 = 6
- **<device>** = zero indexed device number (for device variables, inputs, etc.) from the "devices" table in WinScript
- **<listIndex>** = if this variable is a "list" type variable (array), the position in the array to retrieve. Zero indexed only if the variable itself uses zero indexing, otherwise 1 indexed.

If the index is not relevant (ie: no device index) then "-1" is returned.

Example where "default" is the first sequence in the script:

Received:

defaultNL<h0d>

Returned:

s|0|-1|-1NL<h0d>

If the value is not found, the following string is returned:

Error: Resource name not found<h0D>

Troubleshooting Tips

The following table provides some possible reasons for behavior of the show controller or WinScriptLive. Please see our website for more frequently asked questions and knowledgebase.

WinScriptLive

Symptom	Possible Cause	Possible Solution
Can't send a script – WinScriptLive stays stuck on "verify"	Compact Flash card failure	Remove compact flash card from rear of unit. Place in CF reader connected to a PC. Save any script data or webpages you need to the PC. Format as FAT32. In WinScriptLive, go to "Tools" and click "Create default CF Card" to get default webpages if desired.
	Firewall blocking port 2638 or 2639 if using Ethernet	Connect directly from PC to Show Controller (without router or switch), or change router/switch settings.
Can't paste/insert in a new row in WinScriptLive. It copies over existing row.	Row is highlighted	Make sure before pasting that no other complete rows are highlighted
"Live Mode" keeps getting disconnected	Touch and WinScriptLive are running on the same PC	It is not recommended to run the touch software and WinScriptLive both in Live Mode at the same time. This may lead to disconnections from Live Mode in WinScriptLive.
	Wireless connection intermittent	If you are using a wireless Ethernet connection, try connecting using a wired connection.
Product file changes don't take effect	Re-selection of product and Restart of WinScriptLive Required	After modifying a product file, WinScriptLive must be re-started to re-read the file. In addition, for your script file, you must re-select the "version" of the product in the "devices" screen window using "edit device" in WinScriptLive.
NTP doesn't update	Firewall block	Make sure that port 123 is allowed to pass through any routers or switches to reach your destination ip or dns address for NTP.
SMPTE timed sequence doesn't run as timecode is running	Sequence isn't "armed"	Make sure that the sequence is "armed". This can be done using an "arm" command from another sequence, the "autostart" checkbox, or a trigger.