

BlueLite X1 Automation API

March 21, 2005

Introduction

Versions 1.1 (and newer) of the BlueLite X1 Show Control System include an Automation API. This API is intended to allow 3rd Party Applications to control the BlueLite X1 software for specialized purposes, for example, complex show scheduling or responding to non-DMX user input.

The API is implemented as an ActiveX control which is automatically registered when the BlueLite software is installed. For robustness, the X1 application is not a so-called 'Server Application'. This means that the BlueLite X1 Windows application will not automatically launch when the ActiveX control is instantiated. The program must either be launched manually, or must be launched programmatically (WinExec(), ShellExec(), CreateProcess(), etc.) before the X1 Automation Services can be used.

Basic Services Provided

The Automation API lets an external program do anything that can be done from the BlueLite X1's DMX-512 input. This includes:

- Set Submaster Levels
- Load Presets
- Set the Grandmaster Level
- Set XY Crossfader Levels
- Play, Stop, Record
- Set Transport Mode
- Play a specific event list section (DMX-512 input's 'Automate' feature)

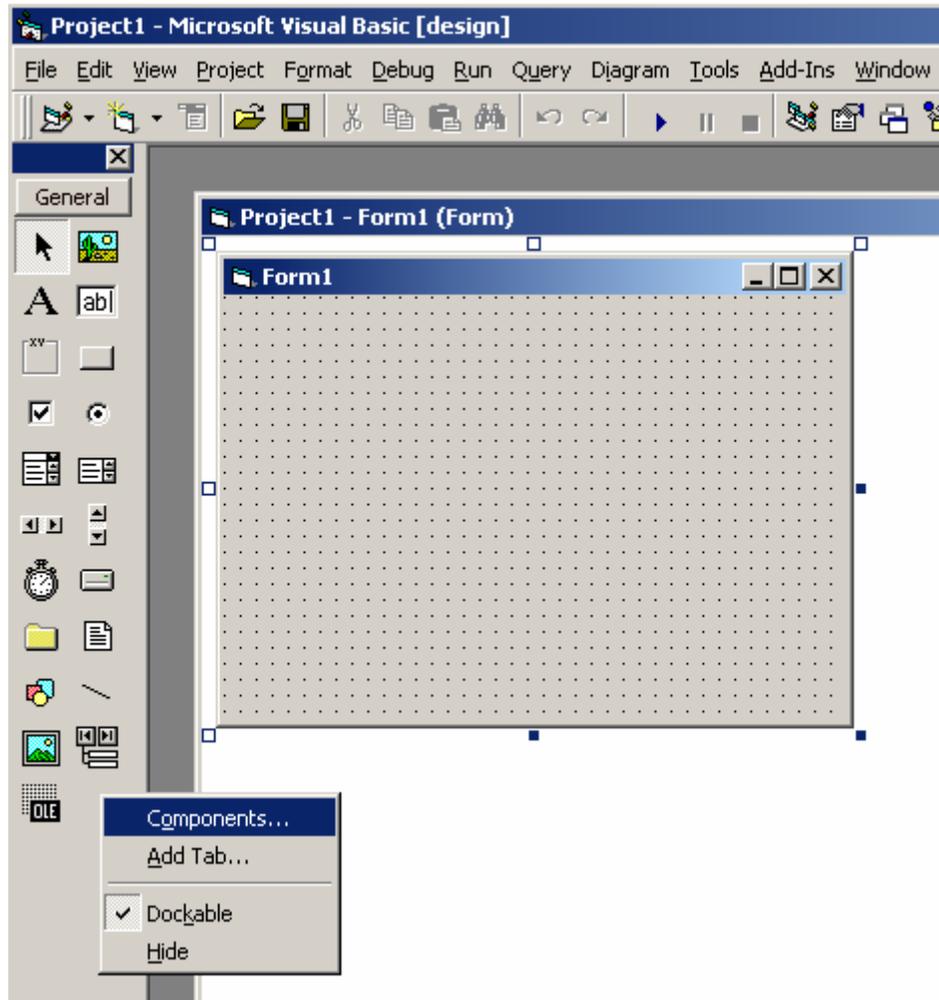
In addition, the Automation API provides support for the following

- Assigning Cues to either a Submaster or an XY Crossfader
- 'Quietly' loading a Show file (no prompts for changes, etc.)
- Monitor X1 State (stopped, playing, recording), Mode (Internal, Timecode, Step), Current Time, and, if applicable, Current Timecode Status (waiting, present, lost)

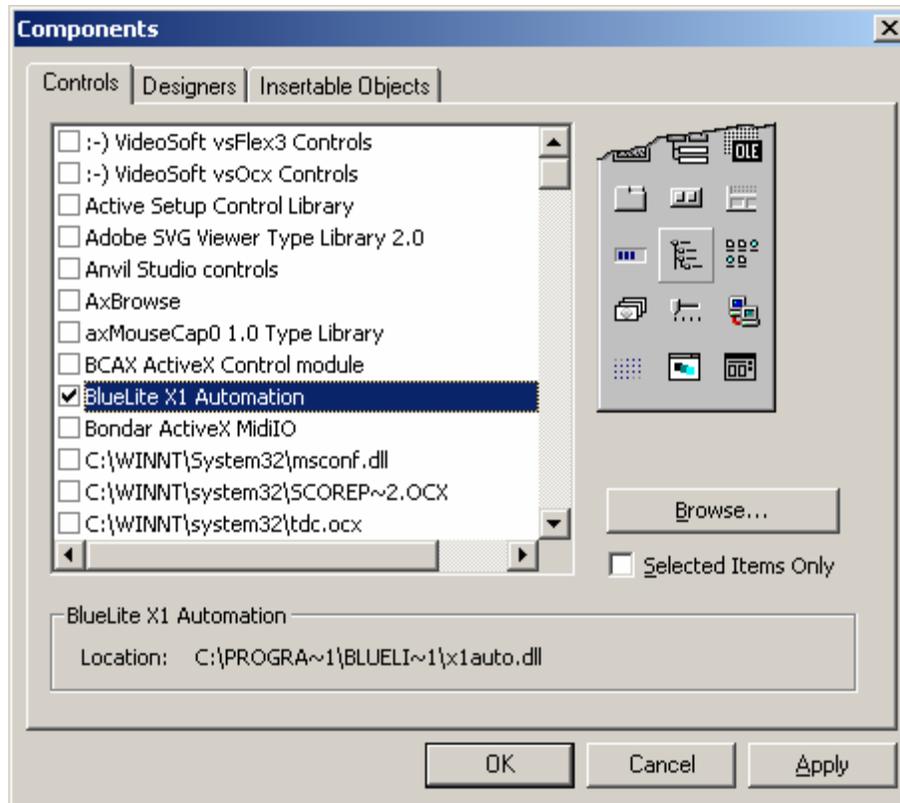
Simple Example

To demonstrate the API, let's begin with a simple example using Microsoft's Visual Basic. With the X1 Software already installed, launch the Visual Basic environment and select "Standard EXE" from the "New" project Tab.

Next, right click on the Control toolbar and select 'Components':



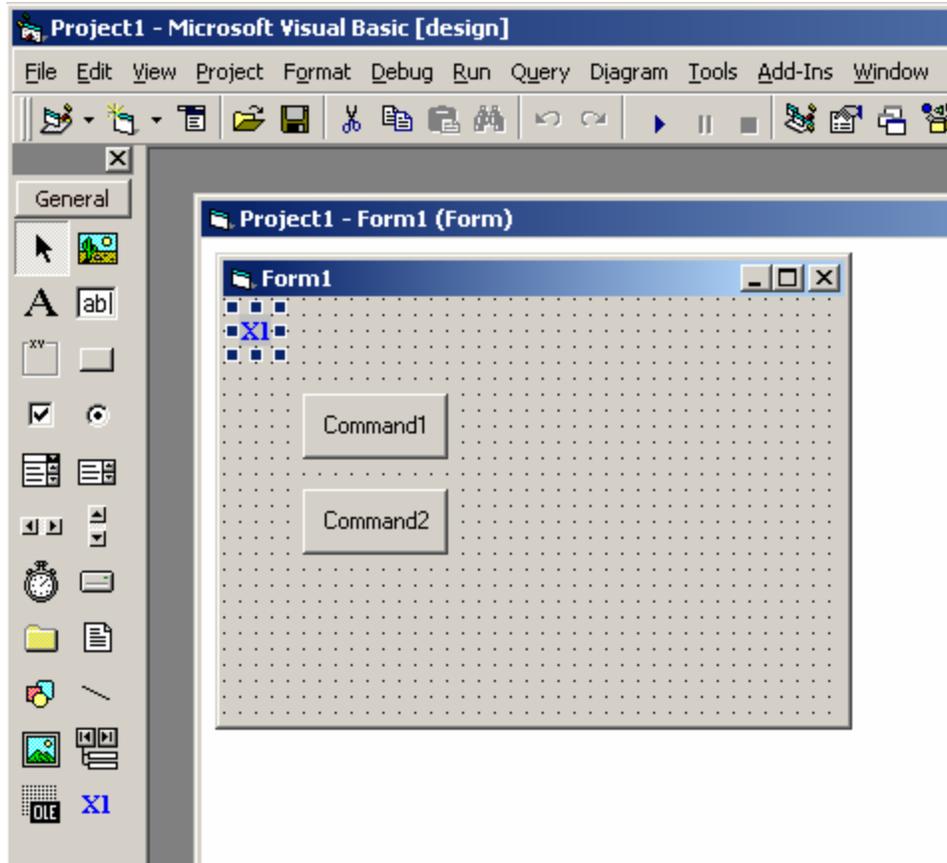
Select "BlueLite X1 Automation" from the Control List:



You can also 'Browse...' for x1auto.dll, located in the BlueLite X1 Installation Directory. Either way, the BlueLite control will appear in the toolbar:



We can add an instance of this control to the new Form for our application. Since the control is not visible at run time, it can be placed anywhere on the form.



In the picture above, we have also added two 'Command Buttons'. Next we will add three subroutines:

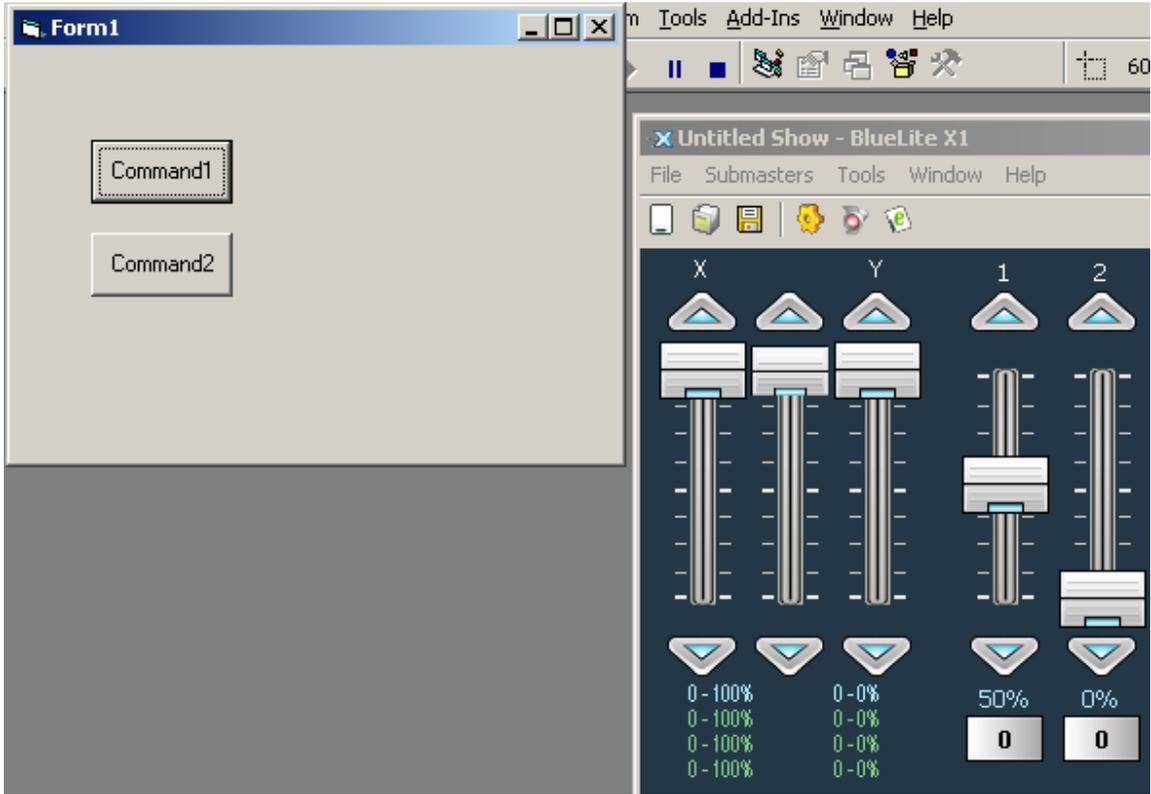
```
Private Sub Command1_Click()
X1Automation1.SetSubmasterLevel 0, 4096
End Sub
```

```
Private Sub Command2_Click()
X1Automation1.Load "c:\test.x1"
X1Automation1.Play 0
End Sub
```

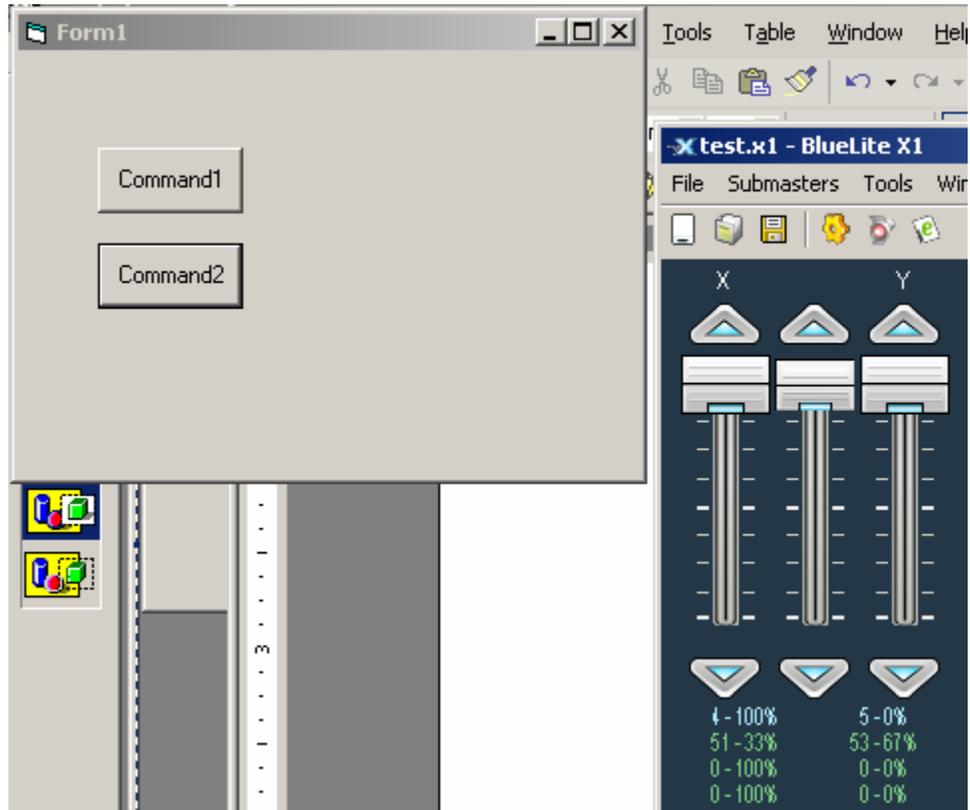
```
Private Sub Form_Load()
If X1Automation1.IsConnected = 0 Then
MsgBox "The buttons will have no effect until the X1 Software is
started"
End If
End Sub
```

The last one is invoked when our Form first loads. It checks if we have an active connection with the X1 software. No connection means that the X1 application is not running, so we put up a message.

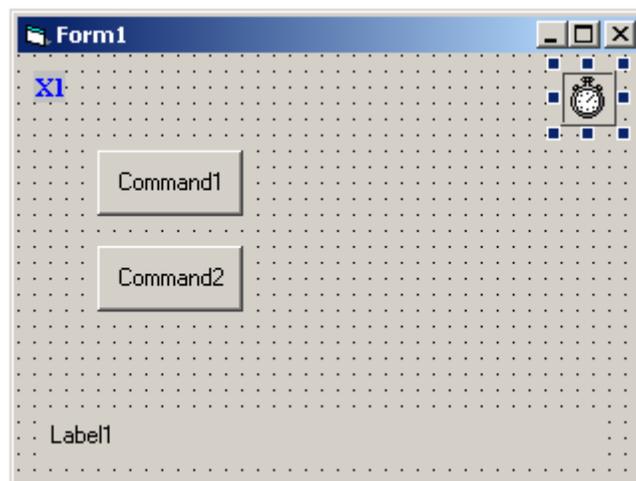
The other two subroutines handle mouse clicks on the two buttons. Button one will set Submaster 1 (the API counts 0-23) to 50% (the API accepts 0 to 8192 for levels):



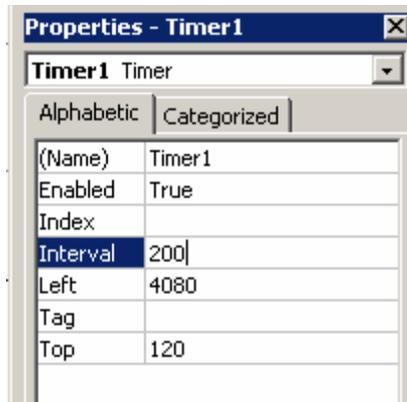
Button two loads a specific Show File (c:\test.x1), and starts it playing at 00:00:00:00 (times are in milliseconds):



Now that we have some very basic control over the X1, we will add a Timer control and a Label control to our form:



The interval on the time will be set to 200 (the value is in milliseconds, so our timer will trigger 5 times a second).



Our Timer subroutine is a bit more complicated than the other three:

```

Private Sub Timer1_Timer()
' Build Status String
StatusString = "X1: "

' Are we connected?
If X1Automation1.IsConnected <> 0 Then
' Add Mode
If X1Automation1.X1Mode = 0 Then
StatusString = StatusString + "Internal Mode, "
ElseIf X1Automation1.X1Mode = 1 Then
StatusString = StatusString + "Timecode Mode, "
Else
StatusString = StatusString + "Step Mode, "
End If

' Add State
If X1Automation1.X1State = 0 Then
StatusString = StatusString + "Stopped: "
ElseIf X1Automation1.X1State = 1 Then
StatusString = StatusString + "Playing: "
Else
StatusString = StatusString + "Recording: "
End If

' Convert MS to H, M, S, F
X1Time = X1Automation1.X1Time
Hours = Int(X1Time / 3600000)
X1Time = X1Time Mod 3600000
Mins = Int(X1Time / 60000)
X1Time = X1Time Mod 60000
Secs = Int(X1Time / 1000)
X1Time = X1Time Mod 1000
Frames = Int(X1Time / 33)

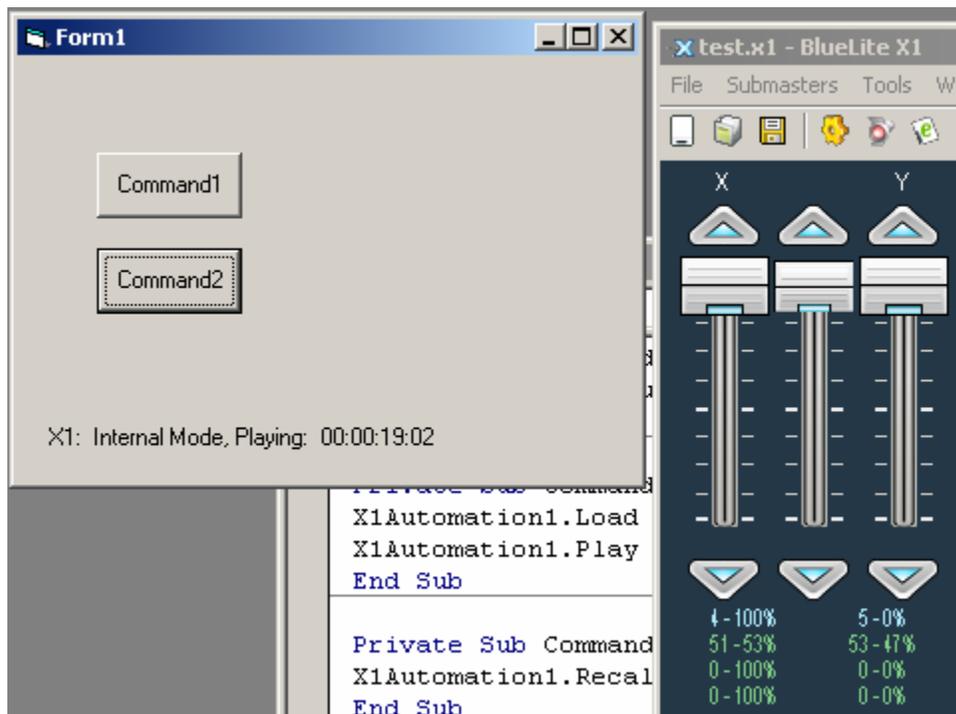
' Format and Add Time
StatusString = StatusString + Format(Hours, "00:") + Format(Mins,
"00:") + Format(Secs, "00:") + Format(Frames, "00")
Else
' Not connected
StatusString = StatusString + "?"

```

```
End If
```

```
' Update our Label  
Label1.Caption = StatusString  
End Sub
```

It checks the X1 Status properties and constructs a string to display as our new label. Now, when we selected Button two, we get a running display:



API Reference

The following is a list of all ActiveX methods and properties provided by the Automation API

Properties

Name	Access	Description
IsConnected	Read Only	Connection State 0=No Connection, 1=Connection
X1Mode	Read Only	X1 Transport Mode 0=Internal, 1=Timecode, 2=Step
X1State	Read Only	X1 Transport State 0=Stopped, 1=Playing, 2=Recording
X1TimecodeState	Read Only	X1 Timecode State 0=Waiting for Timecode, 1=Timecode Present, 2=Timecode Lost Note: Only valid when X1Mode = 1 (see above)
X1Time	Read Only	X1 System Time Current X1 time (HH:MM:SS:FF) in milliseconds

Methods

The following is a list of all X1 Automation Methods in alphabetical order.

`HRESULT Automate(long starttime, long stoptime)`

Stop the X1 (if playing or recording), switch to Internal Mode, then play the event list from *starttime* to *stoptime* (both in milliseconds).

The *stoptime* parameter can be -1 if not needed (ex. loop or stop flags set in list)

`HRESULT Load(BSTR filename)`

Quietly load *filename*. No prompts will be provided from the X1 application unless an error occurs reading the file.

`HRESULT Play(long starttime)`

Equivalent to entering a time (in milliseconds) and pressing the 'Play' button on the X1 Event window. The *starttime* can be -1 if the current System Time should be used.

`HRESULT RecallPreset(long preset)`

Recall an X1 Preset. The value can be 0 to 23.

HRESULT Record(long starttime)

Equivalent to entering a time (in milliseconds) and pressing the 'Record' button on the X1 Event window. The *starttime* can be -1 if the current System Time should be used.

HRESULT SetGrandmasterLevel(long level)

Set the X1 Grandmaster to a specific level. The *level* parameter can be 0 to 8192.

HRESULT SetSubmasterCue(long sub, long cue)

Assign a Cue to an X1 Submaster. The *sub* can be 0 to 23, *cue* can be any valid Cue number. If the Cue does not exist, Cue 0 will be assigned.

HRESULT SetSubmasterLevel(long sub, long level)

Set an X1 Submaster to a specific level. The *sub* can be 0 to 23, *level* can be 0 to 8192.

HRESULT SetTransportMode(long mode)

Select the current X1 Transport mode. Set *mode* to 0 for Internal, 1 for Timecode, and 2 for Step. This call is ignored if the X1 is currently Playing or Recording.

HRESULT SetXCue(long xfader, long cue)

Assign a Cue to an X1 X Fader. The *xfader* can be 0 to 3, *cue* can be any valid Cue number. If the Cue does not exist, Cue 0 will be assigned.

HRESULT SetXLevel(long xfader, long level)

Set an X1 X Fader to a specific level. The *xfader* can be 0 to 3, *level* can be 0 to 8192.

HRESULT SetYCue(long yfader, long cue)

Assign a Cue to an X1 Y Fader. The *yfader* can be 0 to 3, *cue* can be any valid Cue number. If the Cue does not exist, Cue 0 will be assigned.

HRESULT SetYLevel(long yfader, long level)

Set an X1 Y Fader to a specific level. The *yfader* can be 0 to 3, *level* can be 0 to 8192.

HRESULT Stop()

Equivalent to pressing the Stop button on the X1 Event window.

Additional Resources

Additional samples for Visual Basic and MFC can be found, with source code at www.innovateshowcontrols.com/x1automation. Questions regarding the Automation API can also be posted in the BlueLite Forum at www.innovateshowcontrols.com.