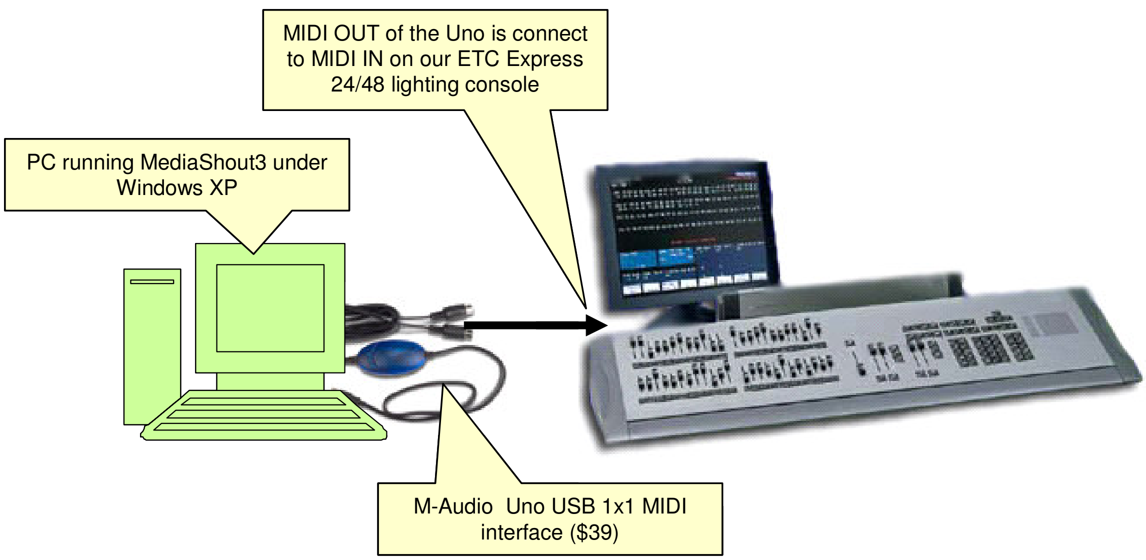
How To Run Your Lighting Console From MediaShout3 using MSC (MIDI Show Control)

Mike Oliver (mikeolivercgp@yahoo.com)
Capital Area Christian Church
http://www.capitalareachurch.com/

Over the past few years, I've been increasingly interested in having MediaShout execute lighting cues on our lighting console. We pre-program all of our lighting cues before Sunday. On Sunday morning, our operator executes these cues sequentially by hitting the [GO] command button on our ETC 24/48 Express lighting console. Since these cues are tied to elements in our worship services (like song transitions, video transitions, etc.), I thought it might be nice to integrate the execution of these cues from within MediaShout.

I've known for the past five years that our console supports MSC (MIDI Show Control) and can either send or receive MIDI commands via its MIDI IN and OUT ports. With the release of MediaShout3, I can now take advantage of MSC since MediaShout3 can play MIDI song files as one of several sound cue file types it supports.

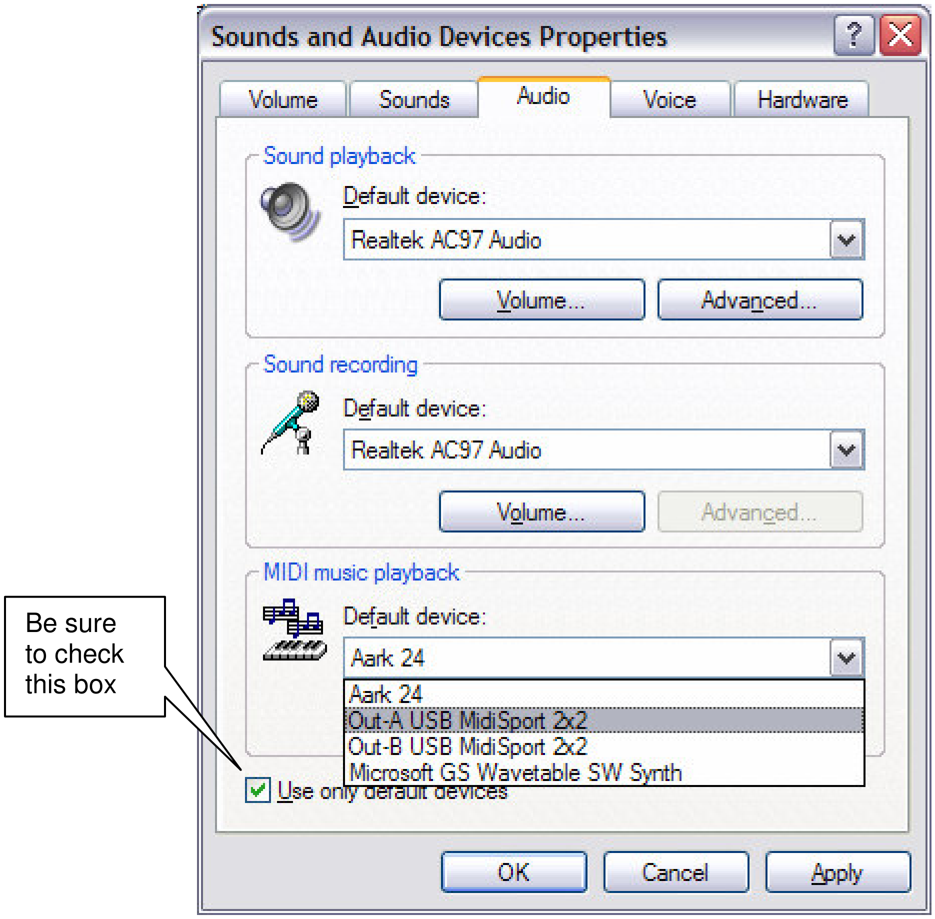
The basic configuration I used was very simple:



For M-Audio Uno USB 1x1 MIDI port see: http://www.americanmusical.com/item--i-MII-UNO.html

After you install your MIDI port, you need to instruct Windows XP that it should always use this as the default playback device for MIDI. The reason this is important is because unlike other MIDI applications which allow you to choose which MIDI ports they should use, MediaShout3 assumes it will always use Microsoft Windows' designated MIDI port. Since I always want MIDI information from MediaShout to be transmitted to my USB MIDI port, I must designate it as the Windows default.

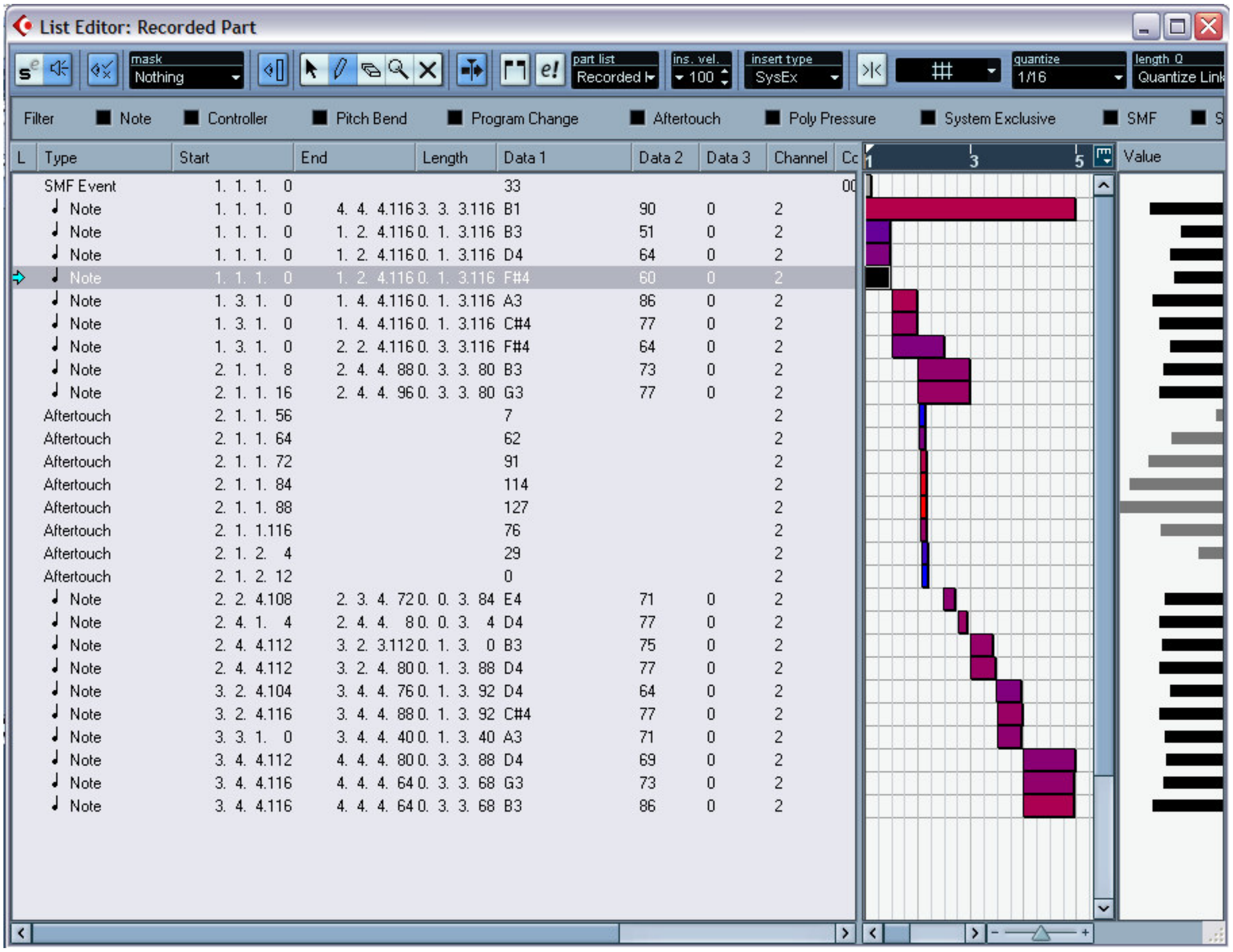
To specify which port Windows should use, go to the Windows ***Start*** menu (usually in the lower left corner of your screen). Go to ***Control Panel*** and select ***Sounds and Audio Devices***. You'll be presented with the following screen:



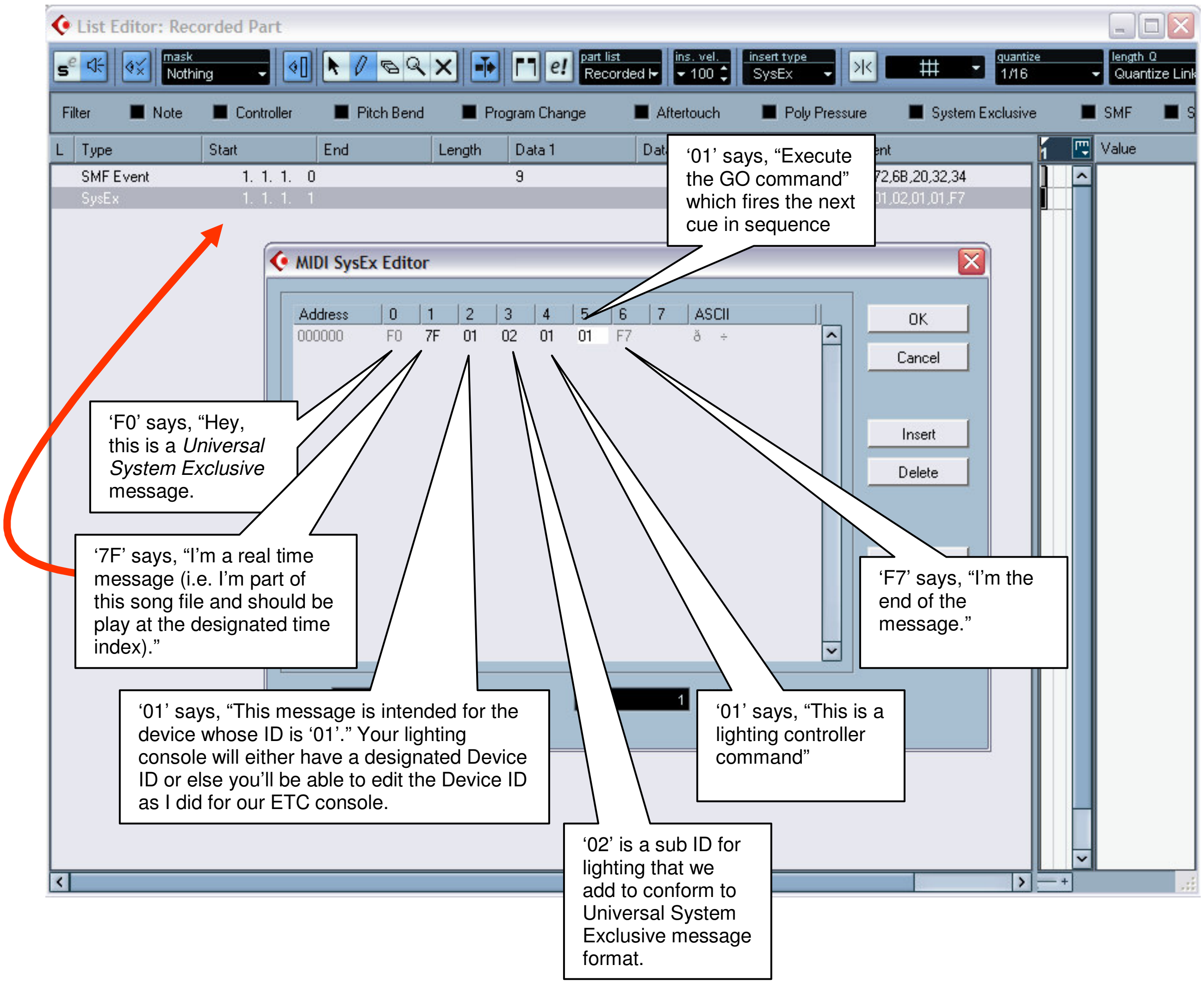
Please ignore the fact that you don't see my Uno 1x1 MIDI device listed. I'm writing this document on a different PC.

With the MIDI port installed, configured, and attached to your lighting console, you now need a MIDI song file which contains what command(s) you want to send to your lighting console. For the moment, I'm using a single MSC command which simply tells my lighting console to "GO". The GO command fires the next consecutive lighting cue in an active cue list.

To create this file, I took a MIDI song file that I had and opened it in a MIDI List Editor, which lets me see every command in that file:



After I removed all the notes and other controller instructions, I entered a Sysex (System Exclusive) command as illustrated below:



This MIDI file containing the GO command is available at the same website from which you downloaded this document.

Setting Up For A Worship Service

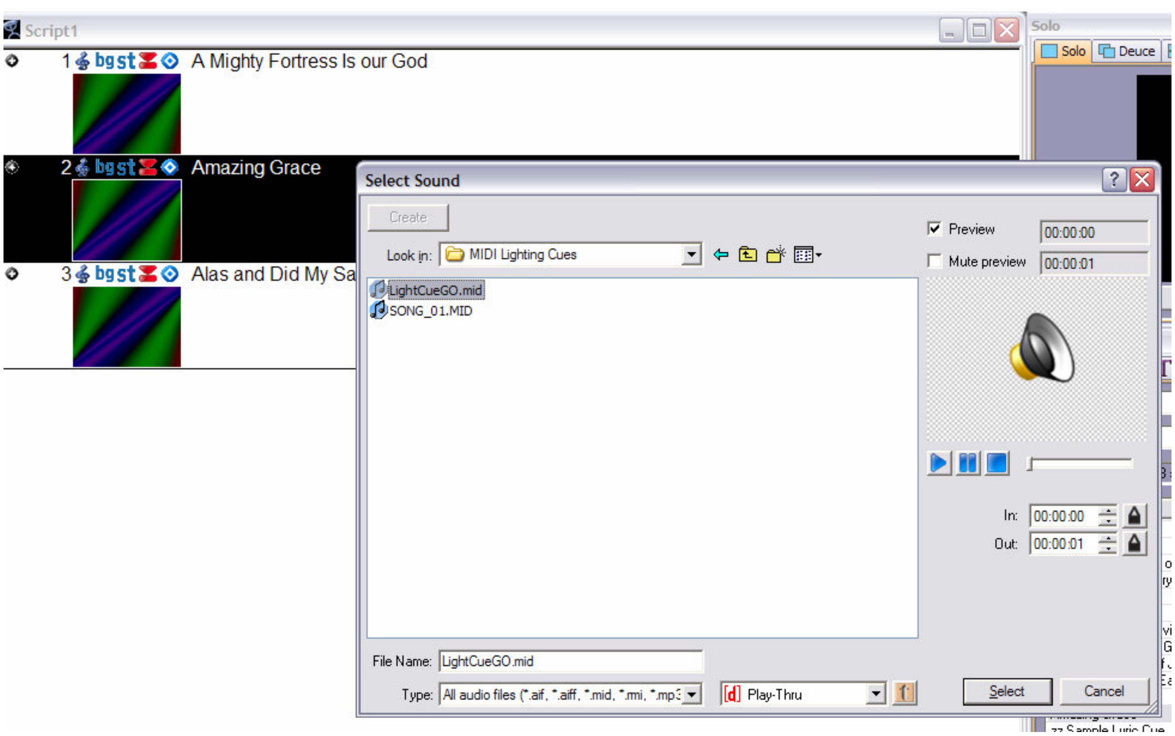
To bring this solution to bear for a worship service, do the following:

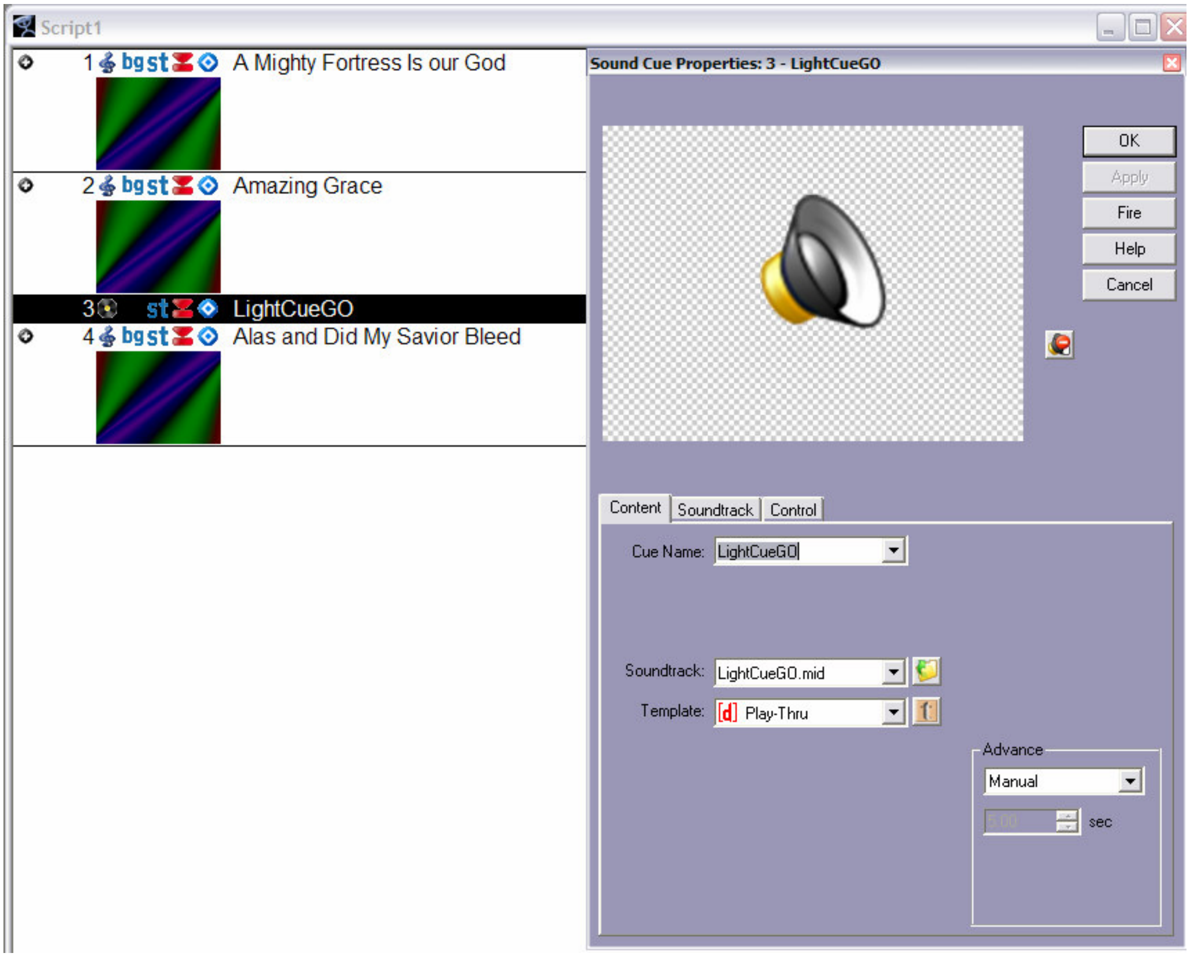
1. **Program your lighting console so that every lighting cue you'll use is in consecutive order.** We rarely execute manual cues. Programmed cues where you can control the timing of a fade are far smoother than someone manually moving a fader, especially when you're using DMX-512 controlled intelligent lighting whereby you may be using follow cues to re-index lighting instrument parameters while the lighting level is dimmed - it's less distracting (I was going to say, "It's more professional…", but the reason we want that professional feel is that we don't want to distract our worshippers with technical performance errors.

   Historically, we've routinely programmed our cues in consecutive order so that "in a pinch," our FOH audio tech could reach over and hit our lighting console's GO command button to fire the next lighting cue for the service. Most weeks, we would have an operator who would sit there with a script for a respective service and hit the GO command button. They would usually be bored to tears, since they all have to do is hit a single button to advance to the next lighting cue in the service (now you understand 'why' I want MediaShout to fire these cues).

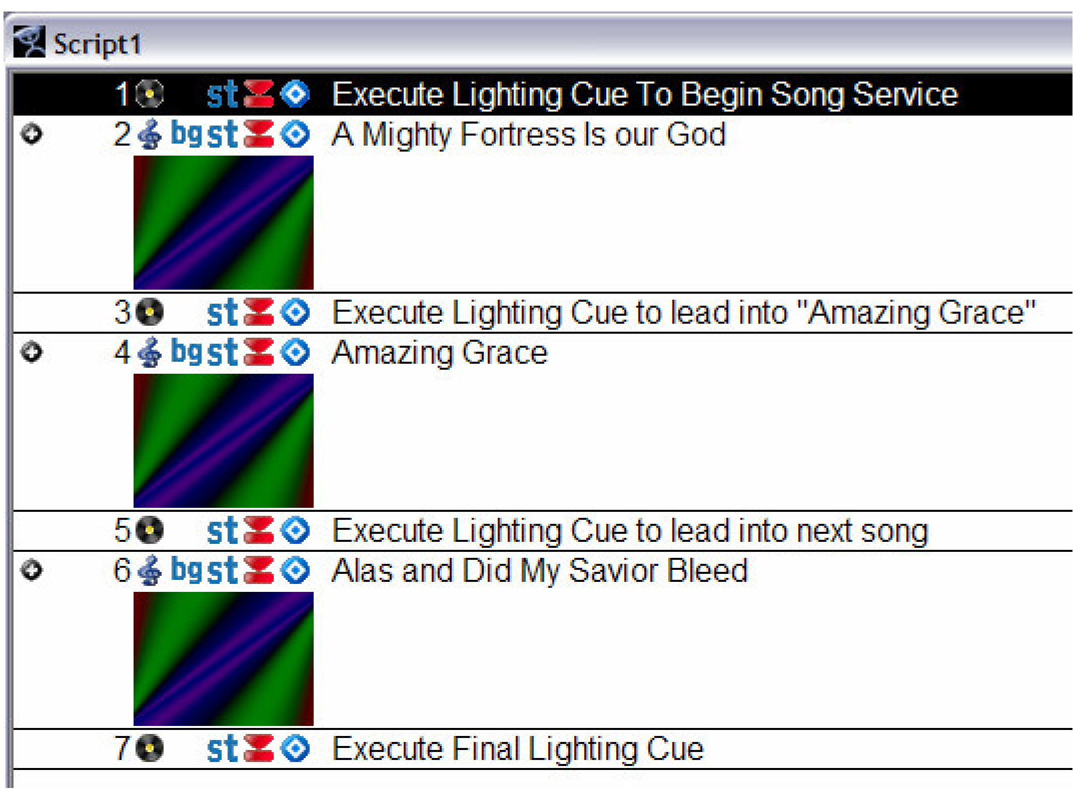
2. **Cue up your initial lighting cue on your lighting console**.

3. **Within MediaShout, load this MIDI file to be played as a sound cue for each place in the service you would like to advance to the next lighting cue** (between songs, perhaps between dramatic elements in the song like the start of a final 'high intensity' chorus, etc.). Here's an example of the how to load this MIDI file into MediaShout3:





4. Copy this cue to every other location at which you should be advancing to your next lighting cue:



5. **Do a trial run-through of your service to ensure you don't have any "GO" commands in the wrong places** (or you don't have more "GO" commands than you have lighting cues)

When the service is underway, each time your MediaShout operator executes one of these "GO" command cues, the MIDI file containing the SysEx command plays through the designated default MIDI port. This sends a message to the lighting console, "Execute the next lighting cue in sequence". Since all of my lighting cues have designated UpFade and DownFade times, once the lighting console is told "GO (to the next cue in sequence)", it follows it's own programming for the execution of the next cue.

Other Things Within The "Art of the Possible"

The items I've discussed above are just the 'tip of the iceberg'. I've offered just one basic command – "GO". The MIDI Show Command language is pretty robust. Within the 'art of the possible' are things like:

- Issuing MSC commands to execute specific cues or macros for moving intelligent lighting

- Synching lighting cues to specific places in a song – Since we're issuing these commands within a MIDI song file, you could choose to add a GM synth module like Yamaha's MU15. With a MIDI interface which offers more than 1 set of ports (or by using the MIDI through port on your sound module or lighting console), you could sequence an entire song, feeding just a click track to the band (or just the drummer) in their monitors. Since we no the time index of every note in that song, we can add SysEx lighting commands to execute at the same time index in the song.

The documentation for your lighting console will usually contain an appendix documenting all the MIDI command syntax your console understands. For more information about formatting MIDI Show Control commands, I would encourage you to pick up a copy of a book called **Control Systems For Live Entertainment**. In Chapter 19, author John Huntington goes into great detail with a number of examples of formatting SysEx commands to control stage lighting.

Available at Amazon:
http://www.amazon.com/gp/product/0240803485/102-2969855-1943364?v=glance&n=283155&n=507846&s=books&v=glance