

CobraNet Applications Introduction:

Rane makes two CobraNet products:

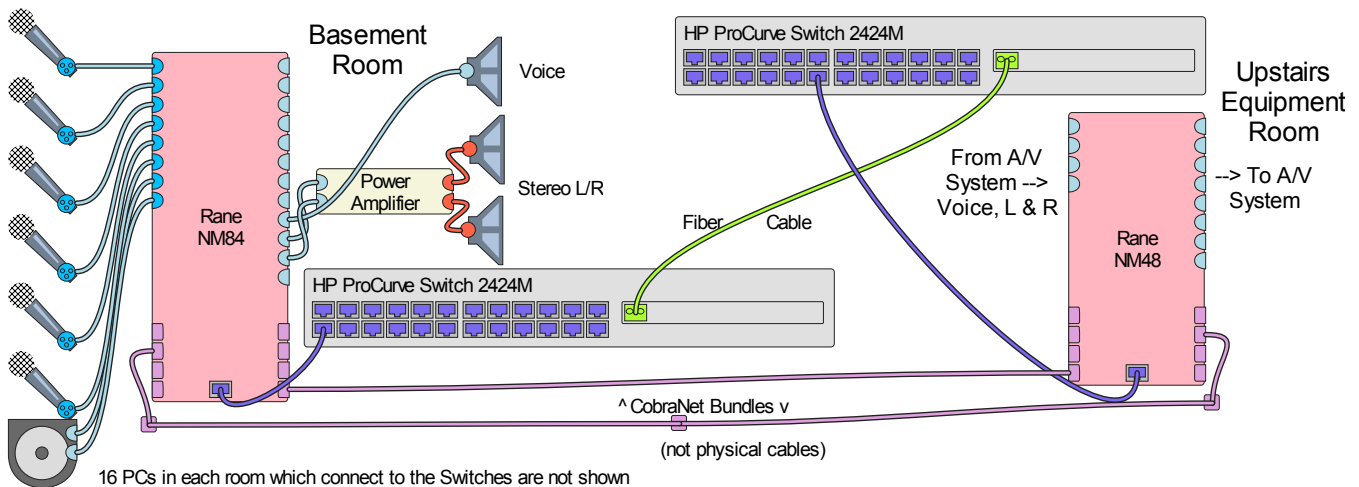
NM 84 Network Microphone Preamp & the NM 48 Network Module, which utilize CobraNet technology to run uncompressed, 20- or 24-bit, 48 kHz digital audio over standard Ethernet networks. The NM 84 offers 8 mic or line audio channels *onto* the net work plus 4 line channels off of the network. The NM 84 also has line-level Direct Outputs for each of the 8 mic channels. Mostly conversely, the NM 48 offers 4 line-level onto and 8 line-level off of the network. An LCD front panel interface makes system setup easy for installers.

CobraNet Application #1 – Historic building audio renovation

Problems: In a nationally registered historic building, get 6 microphone channels and a stereo, line-level audio source from the basement up to the equipment room on the second floor that’s also across the building 6 rooms down the hall. The caveat is that besides the existing CAT5 and fiber computer network cabling, no other cables are currently available or run, no more space is available in the existing conduit and no additional holes or external cabling or conduit can be added – otherwise Uncle Sam will be very upset about his old, old building.

Solution: Use CobraNet and run the audio over the existing computer network.

System Block Diagram:



Description:

Six mics and stereo audio are fed into a Rane NM 84 in the basement closet with the proper gains set by the installer using the NM 84 front panel. The Rane device is in a locked closet in a locked cabinet, but the NM 84’s front panel is also locked out to keep the 137 “audio experts” in the building out. Two new Ethernet computer network switches are required to replace the old repeater hubs of yore, one in the basement and one in the equipment room. Since this solution requires sharing computer data and CobraNet audio data over the same network cables, managed switches must be used. Hewlett Packard ProCurve 2424M switches are used in this case since more than a dozen computers are connected across the network to other PCs and the servers in the equipment room. Since these switches also have a very noisy fan, special acoustic dampening must be incorporated in the basement switch location to avoid the excess noise while simultaneously allowing the proper heat dissipation.

In both hp switches, setup separate VLANs (Virtual LANs) – one VLAN for the computer traffic, and a separate VLAN for the CobraNet traffic. Since no PC control of the NM 84 mic stages is required in this

application, there is no need for a PC on the network to “talk” across the VLANs for control purposes – although, this is certainly possible if needed.

Upstairs at the other end of the fiber optic cable in the equipment room, the managed switch connects to a Rane NM 48 to retrieve the 8 audio channels at line-level and feed them into the A/V equipment rack for mixing, routing, level adjustments and control. Once next year’s budget comes through, audio teleconferencing and/or video conferencing capabilities will be available in the equipment room. Up to four, line-level audio feeds back to the basement are available on the existing Rane devices. But since only three audio channels will be required, one stereo feed plus a mono voice “center” channel, all is right with the world.

CobraNet Application #2 – Seattle’s Key Arena (basketball/hockey; home of the Seattle Supersonics)

Problems: The conduit running from courtside and the North bucket microphone to the 4th floor equipment room, then to the broadcast truck area, then to the center-hung clusters is still filled with water since the total redesign of the arena years ago. Radio and TV broadcasters, building renters (Disney on Ice; Ringling Brothers Circus, Celine Dion and Ozzy Osbourne etc) have complained about the intermittent quality of the audio feeds. Some days there’s no noise and no EMI (radio) interference in the audio feeds, other days they’re not as lucky. Noise and inconsistency is only one problem, the distances involved is a second issue. Plus, Ozzy and Celine really need at least the same pristine audio quality in the private boxes that Kevin Calabro (voice of the Sonics) achieves on his stereo broadcasts (we won’t consider the AM sports radio audio quality). The distances involved are:

Courtside to the equipment room = 1,200 feet
Equipment room to broadcast feeds = 600 feet
North court bucket to equipment room = 400 feet
Equipment room to center clusters = 300 feet

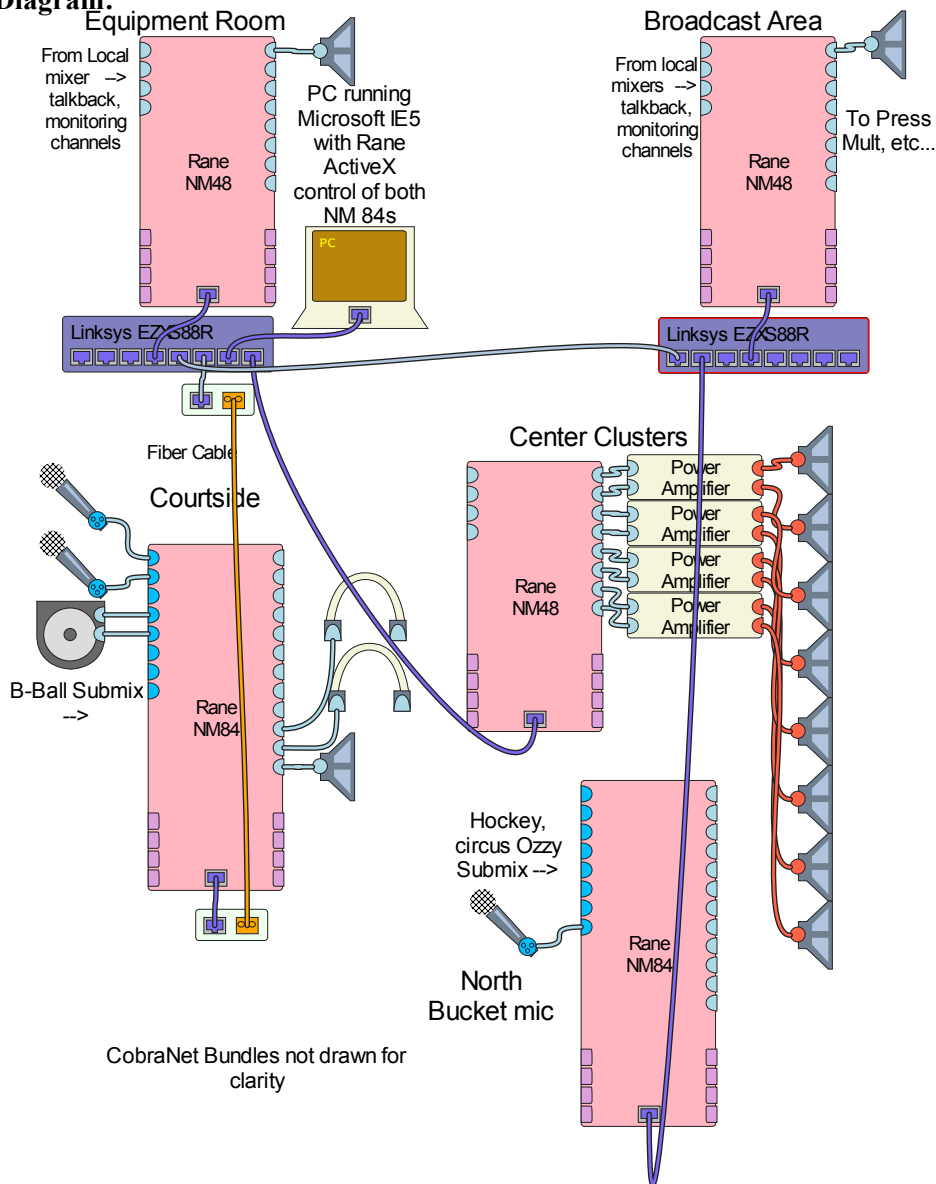
Solution: Use CobraNet and run the audio over CAT5 for runs under 1000 feet and use fiber for the 1,200 foot run. (See system on next page)

Description:

Place a Rane NM 84 at courtside with a media converter to convert to fiber, then run the 1,200 feet to the equipment room through the watery conduit. Run “sits-in-water-just-fine” CAT5 cable through the other runs. In the equipment room, in addition to the North bucket microphone that’s plugged into a second NM 84, use a custom web page on a PC running Microsoft Internet Explorer and utilize Rane’s ActiveX controls to link the web page’s screen objects to the device parameters (mic gains) in both the NM 84s. Place NM 48s at both the broadcast area and hung in the center cluster for the multiple audio feeds involved at these locations.

Sub mix audio as needed at either courtside, equipment room, broadcast location or all of the above as required. Re-route the audio as required for the various uses: the basketball announcer location is across the floor from the hockey/circus announcer location. Route the concert audio to the proper places when the touring rigs bring *almost* all their own equipment.

System Block Diagram:



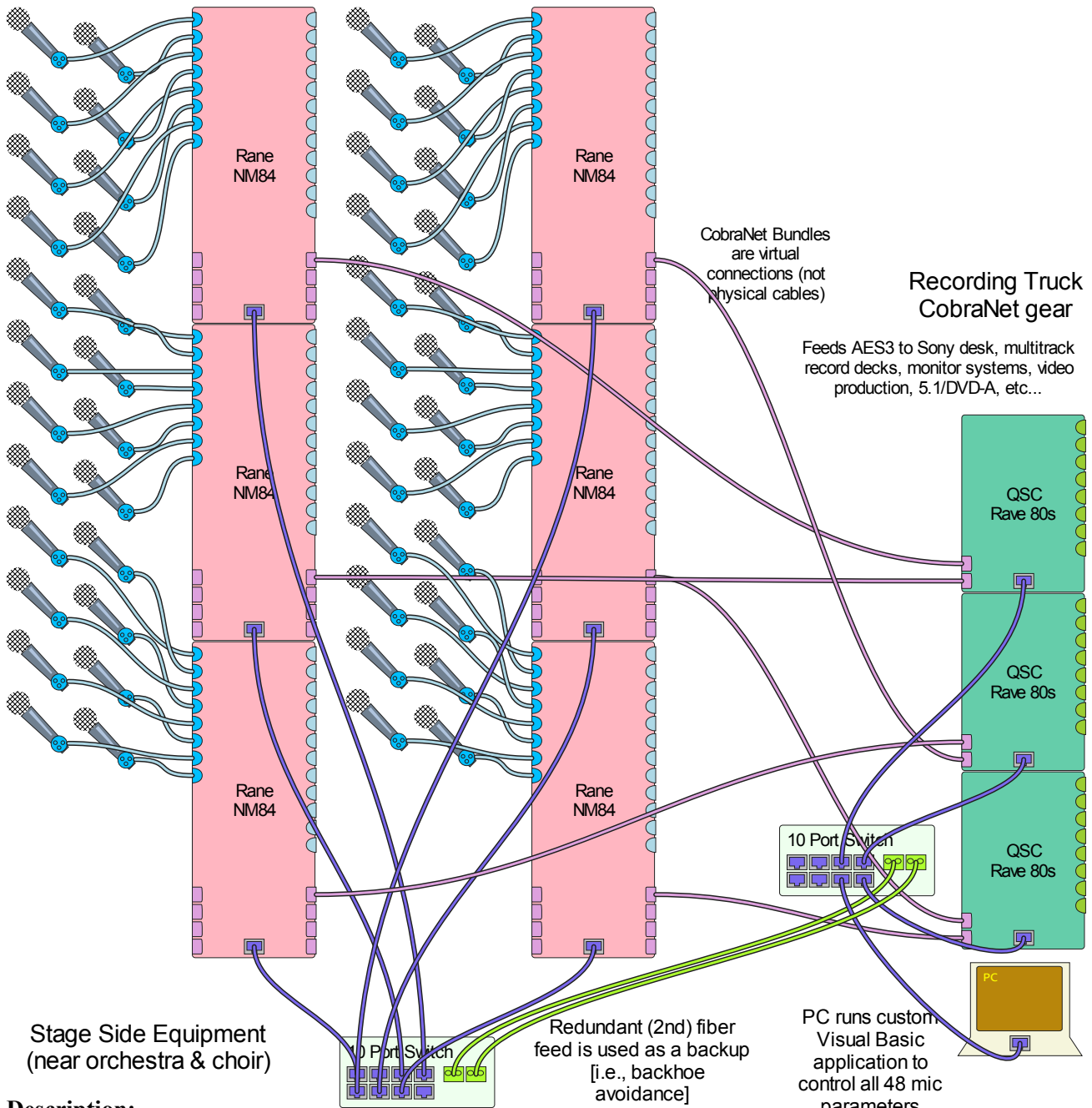
CobraNet Application #3 – Classical Remote Recording (design

and install by Mark Harrison at *Floating Earth* www.floatingearth.demon.co.uk)

Problems: Provide a classical music, remote recording company with these capabilities: allow engineers to minimize the distance between the microphone and the mic preamp and the A/D converters (i.e., convert to digital at the bottom of the mic stand), while providing full control of the preamp from a remotely-located computer. Maintain audio quality so the remote company's long-time customers, EMI, Sony, blah blah blah are provided with the best audio quality available for the discerning classical CD/DVD buying ear.

Solution: Use CobraNet and run the audio over fiber using standard Ethernet hardware devices.

System Block Diagram:



Description:

Place Rane NM 84s on the stage next to the dozens of Neumann and other high end microphones. Connect the Rane devices into a quiet Ethernet switch, run fiber out of the switch to the remote truck in the street. The switch in the recording truck connects to QSC Audio RAVE 188s devices providing digital, AES3 audio directly into the Sony console for monitoring, individual recording feeds, multi-channel 5.1 and 7.1 live mixing for broadcast, The NM 84 microphone settings control interface can be any software application which supports Microsoft ActiveX controls – even a web page. But in this example, Floating Earth developed their own interface using Microsoft Visual Basic since it runs more reliably than a web page. We won't use the words Microsoft and reliability in the same sentence.

CobraNet Application # 4 – Multi-point digital mic snake (Design

by Dave Dugdale, Acoustical Design Group. Installed in a variety of locations; this example is in Wichita, KS with a smaller implementation in Lenexa, KS.)

Problems: A 64 channel microphone snake is required at multiple locations in a church:

- (1) in four different locations on the stage for monitor mixing,
- (2) in the backstage equipment room where the stage-side mic boxes all come out,
- (3) at the front of house location,
- (4) in the on-site multitrack recording studio,
- (5) at the broadcast truck parking area.

Luckily, the lobby amp room and the video production suite in the balcony don't need all 64 channels, just sub mixes for left, center, right and the stereo submix for pro video tape, FM broadcast, and the VHS, audio cassette tape and CD-R recordings for the after-service sermon sales. The collection for the surround sound production equipment (DVD sales) take place at the June, 2002 services (just kidding).

As if this wasn't enough, every Sunday, the traditional service with the choir and orchestra is quickly followed by the contemporary service where the choir and orchestra is replaced with the full band: drums, bass, guitars, three keyboards, horn section, lead and background vocalists, thus requiring complete reconfiguration of the microphone snake and its gain structure – all within 45 minutes. The space is also rented for band and orchestra recording when worship services aren't taking place, thus the snake must often re-route audio to and from the recording studio.

The monitor mix desk location moves depending on the application needs. For traditional services, monitors are mixed in the orchestra pit; for contemporary services monitors are mixed either at the side of the stage or sometimes at the front of house location; for recording sessions, monitors are mixed sometimes in the recording studio since the monitor desk's fans are too noisy for orchestral recording.

Solution: Use CobraNet and run the audio over CAT5 using standard Ethernet hardware devices.

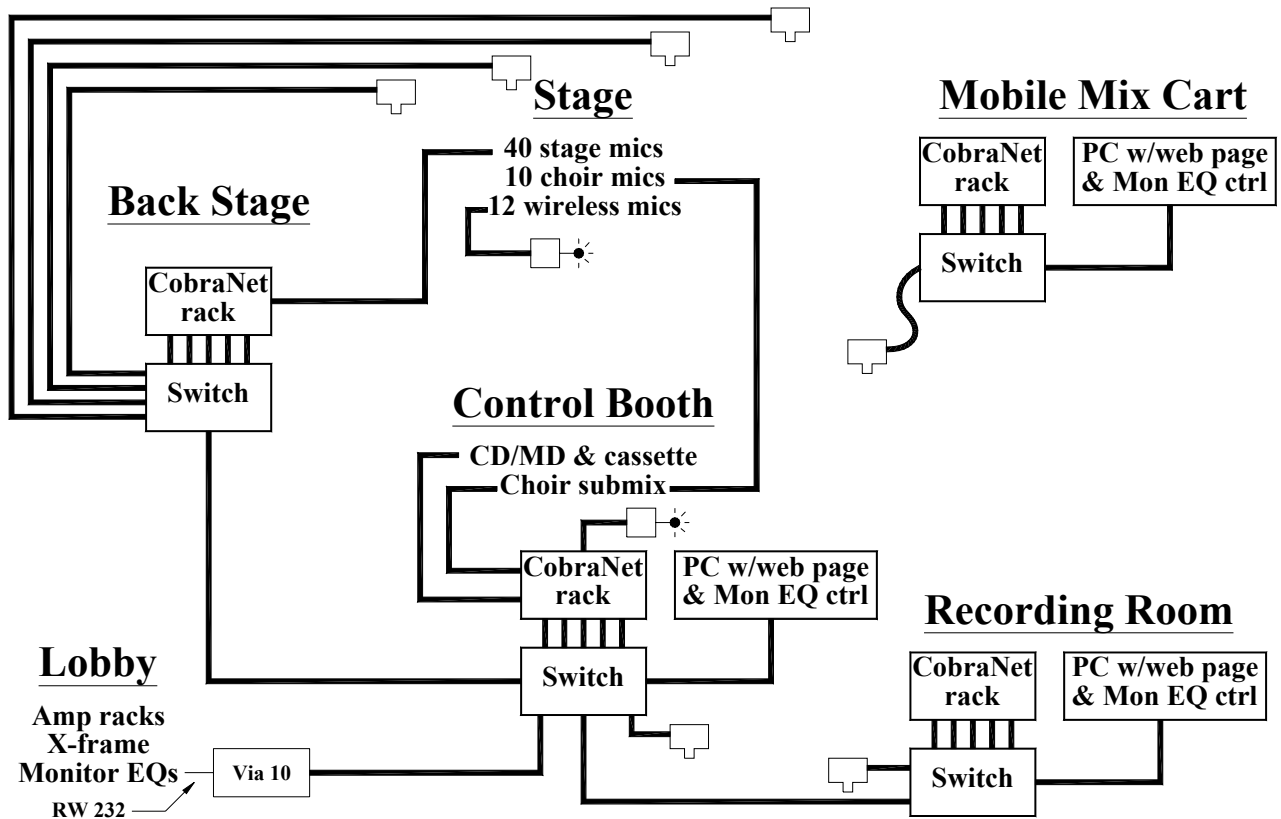
Background

Dave Dugdale, a consultant at Acoustical Design Group, Inc. in Mission, Kansas (www.adgkc.com), designed this church system (whose AutoCAD system drawings are available from Steve Macatee at Rane). The audio contractor was Audio Architects in Groveport, Ohio (www.audioarchitects.com). The video was installed by Ford AV; but, as any audio person knows, since video is always secondary to audio, we'll skip the video details. Yes, this is a joke. This is the Central Christian Church in Wichita, Kansas which opened June 18, 2000.

Basic System

The basic system can be thought of as a digital snake since this is the function of the CobraNet system for this application. (See the block diagram below.) There are 40 microphone feeds from the stage via an analog snake to a backstage room (lavatory casework south). The Rane NM 84 mic stages are backstage where the analog audio is brought up to line level, converted to 24-bit digital audio in the NM 84s and transported through the back stage network switch to both the front of house (FOH) and the recording studio. The switches used here are hp 2424M managed switches. These initial audio channels are designated digital snake channels 1 through 40. At the FOH location, 10 choir mics, cassette & CD sources, as well as 12 wireless microphones are also fed onto the CobraNet network as channels 41 through 56 using more NM 84 units. Additionally, a rolling, mobile cart is used at various locations in the building as a monitor mix station. Roll the cart to any CAT 5 wall plate, plug it into 120 VAC and into the network and mix. The cart is equipped with many QSC Audio RAVE devices, an hp 2424M switch and a Yamaha O2R mixer and the required outboard effects and a CD/mini disc player.

Central Christian Church Block Diagram



Problems and solutions

Dave and the owner of the church decided that remote control of the mic stages would solve two system problems: (1) the church has a traditional service with orchestra and choir, then in less than an hour the snake gain structure must be reconfigured for the contemporary service with full “rock” band with horns, drums, keys, etc. (2) They require multiple operators in different areas of the church to simultaneously trim the mic stage gains when needed. Two or three locations wanted mic gain control: front of house (FOH), recording studio and rarely, the mobile monitor mix cart. To control the mic preamplifier gain between and during sometimes services, CobraNet technology with ActiveX control was utilized because it solved the problem.

After discussions Dave utilized Rane CobraNet devices in this case, not because it offered significant cost advantages over other approaches, but because it solved the remote control need. Comparatively, the new approach’s cost was about the same, but it solved the problem and provided much more flexible re-configuration and future expansion capabilities. This is another advantage to CobraNet-based and network approaches in general – install the system and figure out later the details of the routing and configuration.

Stage monitor EQ must be controllable from either FOH or from the mobile monitor mix cart. Many CobraNet devices offer the ability to transport RS-232 data over the Ethernet network from one CobraNet node to another. In this case, no CobraNet devices were required in the location of the EQs. So, control of the monitor EQs was provided over Ethernet to three, two-channel Rane RW 232, one-third octave equalizers using a Rane Via 10 Ethernet Bridge. The Via 10 enables Rane RW 232 devices to be controlled from the standard RaneWare software screen (Win 95/98 only) on any PC on the network. Simply select Ethernet communications to the proper Via 10 IP address rather than the usual RS-232 communications path.

ActiveX web pages

The web pages pictured below were created so any computer on the network can pull up the mic stages and view and/or edit the mic preamplifier settings. A two-level approach to the web pages was incorporated

due to the large quantity of controls required (40 channels with 7 controls each = 280 controls). The denser Main page is the most often used and displays only 3 of the 7 possible controls for each of the 40 channels. Clicking on the “Mic #” text above each cell provides access to the remaining controls for the selected mic – including duplicated controls from the initial page.

For this system, with 120 ActiveX controls on a single web page, the response timing was about one second, which was approaching the comfort inflection point. The slower timing is not an issue for this application; the sound system users intend to only occasionally utilize the web page controls during performances. Most often, changes will be made between performances while no audio is running through the system. Additionally, with a bit more software programming know-how, one could setup the snake from the web page, then save versions of the snake’s gain structure as an HTML web file to be later recalled into the equipment. This involves writing the VB Script code with the HTML file so it dumps settings *to* the devices rather than loading the web page *from* the devices which is usually the initialization procedure for this application.

Creating the web pages can involve a learning curve if the ActiveX approach is new to you. So plan on this. See the discussions in the ActiveX write up “*Controlling Audio Systems with ActiveX Controls Over CobraNet and other Ethernet-based Networks*” found on Rane’s web site. It covers many of the issues.

The slightly slow ActiveX timing was not an issue for this application, but may be for others. So, choose your control implementation wisely.

Suggestions

Until more audio contractors gain experience in network systems, good network installation and debugging skills will be important traits to look for in audio contractors.

A wise practice that would be good to adopt when designing or installing a CobraNet system was incorporated by Dave: run a couple extra CAT 5 cables – particularly for longer runs. While this at first may seem like a waste, it can literally save your _____. This cable is inexpensive, particularly if you compare the savings versus running many individual cables or multi-channel snakes. Having extra cable runs can overcome several potential problems. For example, while pulling cable, one or more of the cables can be damaged. Keep in mind that continuity may be intact, but damage to even one of the twisted pairs can cause intermittent or not-easily-found network errors. Having an extra cable may provide a valuable debugging tool during setup or at the very least provides future expansion capabilities.

For this church system, having two extra cable runs proved invaluable since it allowed us to verify the proper operation of the FOH CobraNet rack by bypassing the FOH switch. This church design calls for two crossover cables between the back stage switch and the FOH switch. These two CAT 5 cables were Trunked. While debugging the system, being able to use the extra two runs which were wired as standard CAT 5 cables (not crossover wiring), pointed us toward a network setup error – namely, we hadn’t yet configured the Trunks in the switches. We were able to bypass the FOH switch to verify that the FOH CobraNet rack was all working fine, thus, this pointed us to the switch setup being the culprit. Otherwise, we could have spent many hours thinking that the problem was with the CobraNet rack – which was easily verified as “working” thanks to the extra cable runs.

Drawing Suggestions

Those designing systems can solve many in-field problems and make everyone’s job of troubleshooting CobraNet systems easier by including any or all of the following details on the system drawings or elsewhere in the specification. Include on the drawing:

- IP addresses of all CobraNet devices. Note that CobraNet devices do not require IP addresses for audio, but they do require IP addresses if you want to *control* a parameter in a CobraNet device.
- Include Bundle transmit and receive values on each CobraNet device. Also include the converted Bundle hex values to use for each QSC Audio RAVE device. There are convenient tables available from QSC and Peak Audio’s web site.
- Clearly label the crossover CAT 5 cables and the standard wiring CAT 5 cables both in drawings and on the cables themselves. [The following are the only times a crossover cable is needed: when connecting two CobraNet devices directly together with no repeater hub or switch in-between; when connecting a

computer's Ethernet port directly to a single CobraNet device; when connecting between two repeater hubs which are not equipped with Uplink ports; when connecting from one switch to another switch.]

- IP addresses and Subnet Masks of managed switches if you assign IP addresses to them – which is wise for troubleshooting.
- IP addresses and Subnet Masks of all computers in the system used for control
- For switch-based networks, label CAT 5 paths with the total number of bundles entering and leaving switches. This aids in calculating if and whether your design has exceeded the available bandwidth. Peak Audio's CobraCAD software is very useful for this bandwidth calculation task.
- Label the CAT 5 cables that must be Trunked both on the drawings and on the cables themselves.
- If embarking on a system with ActiveX web page control, include the IP addresses of the controlled CobraNet devices in the specification sent to the web page/user interface software provider. They'll need to build the CobraNet devices' IP addresses into the web page(s). This is similar to providing the 232 device addresses to AMX/Crestron programmers.
- Some systems might want the received Bundle numbers to be memory sensitive which is possible with the Rane CobraNet devices. Include a memory chart that maps memories to the Bundle-received values.

Troubleshooting Tips

Networking for audio dudes. Troubleshooting computer networks is not a skill often found in audio people – not yet anyway. Peak Audio's web site has many useful documents to help you design and troubleshoot networks. But, a few brief points here couldn't hurt. Here's the Cliffs Notes version of the list.

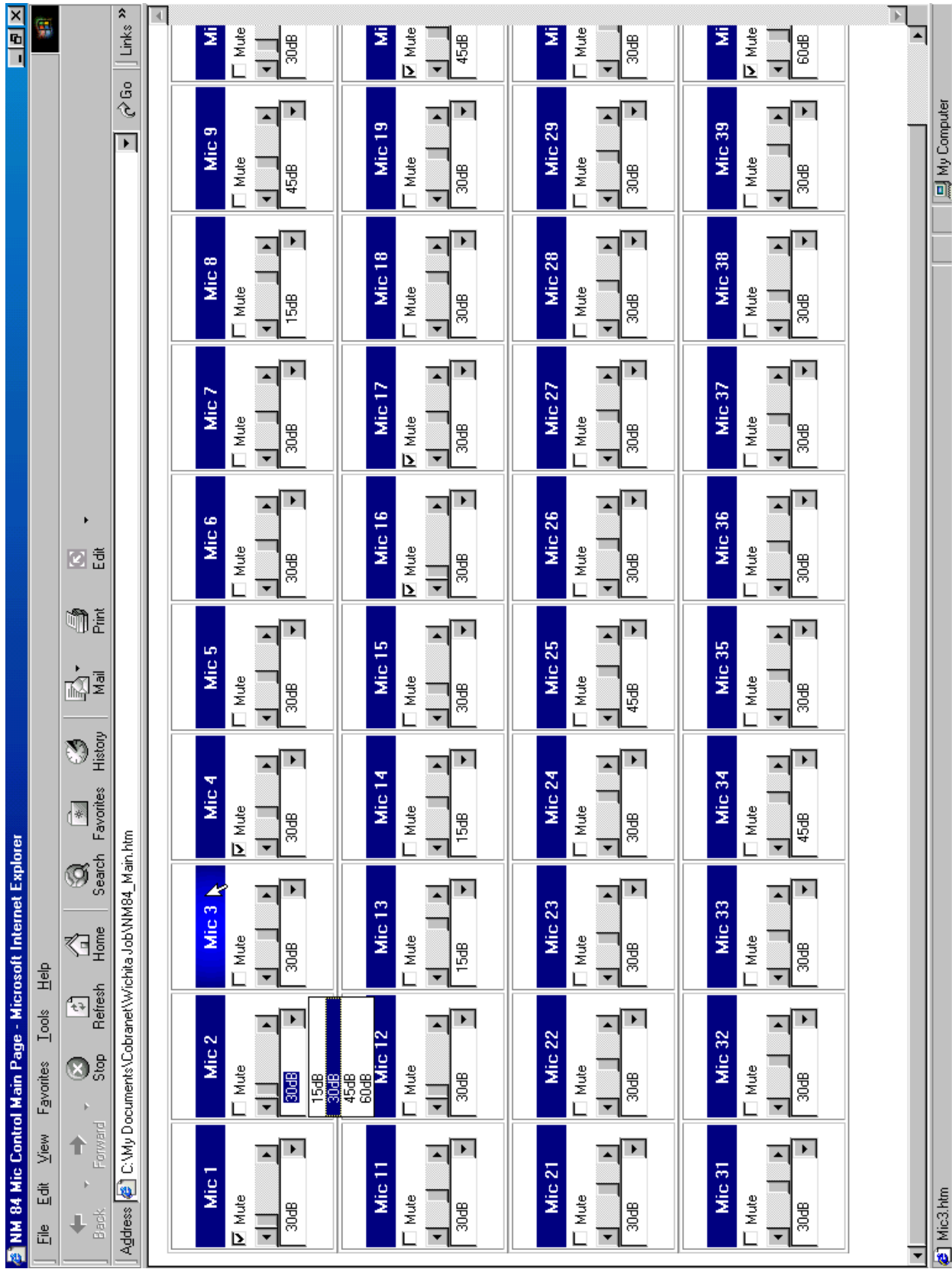
1. Make sure the network cables are *really* wired correctly. Just because the continuity cable tester tells you they're wired properly doesn't mean that they are. [I've trademarked the sayings: *Twist Matters*TM and *Physics Works*TM because, particularly at 100 MHz – they *really* do. Get the twist wrong, and physics makes it not work. Don't you love Mother Nature?]
2. Understand the troubleshooting differences between unicast and multicast networks. Things like, in unicast networks, Bundles are only transmitted when a receiver is "asking" for them. So, in this case, don't assume that all data is present on the network.
3. If possible, spend the extra money for managed switches. Their troubleshooting and monitoring capabilities really will save you time and their extra cost if you run into problems/unknowns.
4. Network bandwidth errors tend to sound like dropouts and clicks and pops – not distortion.
5. Don't automatically assume digital clipping sounds are network related.
6. Obtain compatible CobraNet Firmware for all devices. I've never had audio problems when using differing versions of CobraNet firmware only control bugs. Although very unlikely, network errors that sound like distortion *may* be a CobraNet firmware incompatibility issue.
7. Ask questions in this order:
 - A. Is the transmitting device sending clean audio?
 - B. Is the audio being transported over the network cleanly through each switch? Clean audio and control data transport is very likely through the network unless there's a cabling issue.
 - C. Is the audio being received clean at the other end?
8. When you're really having trouble, take one step or Bundle at a time. Divide and conquer. Troubleshoot by taking one sending device at a time, then add a second, and so on. Check the transmitter, the network transport, then the receiver. Temporarily eliminate the network if need be by bypassing switches. Also try different ports on the same switch since I've made the mistake of assuming that the switch was 100% working – when it turned out that one of its ports was bad.
9. The Rane NM 84 CobraNet device offers a convenient troubleshooting method to check the following:
 - Mic stage clip verification: Listen to the audio on the Direct Output. This output spews an analog version of what the A/D converter in the device is fed.

- Local A/D, D/A & CobraNet interface error verification: Listen to the Monitor Output on the NM 84 sending device and set it up to receive Local Mic Input audio. If this Monitor audio is clean, it tells you that all A/D and D/A converters and the CobraNet interface in the NM 84 are fine.
- Distant D/A & CobraNet interface errors verification: Move to any other NM or analog output CobraNet device and listen to the transmitted Bundle from the sending device to verify that the network devices and cabling are working properly. You might want to do this one device or Bundle at a time.

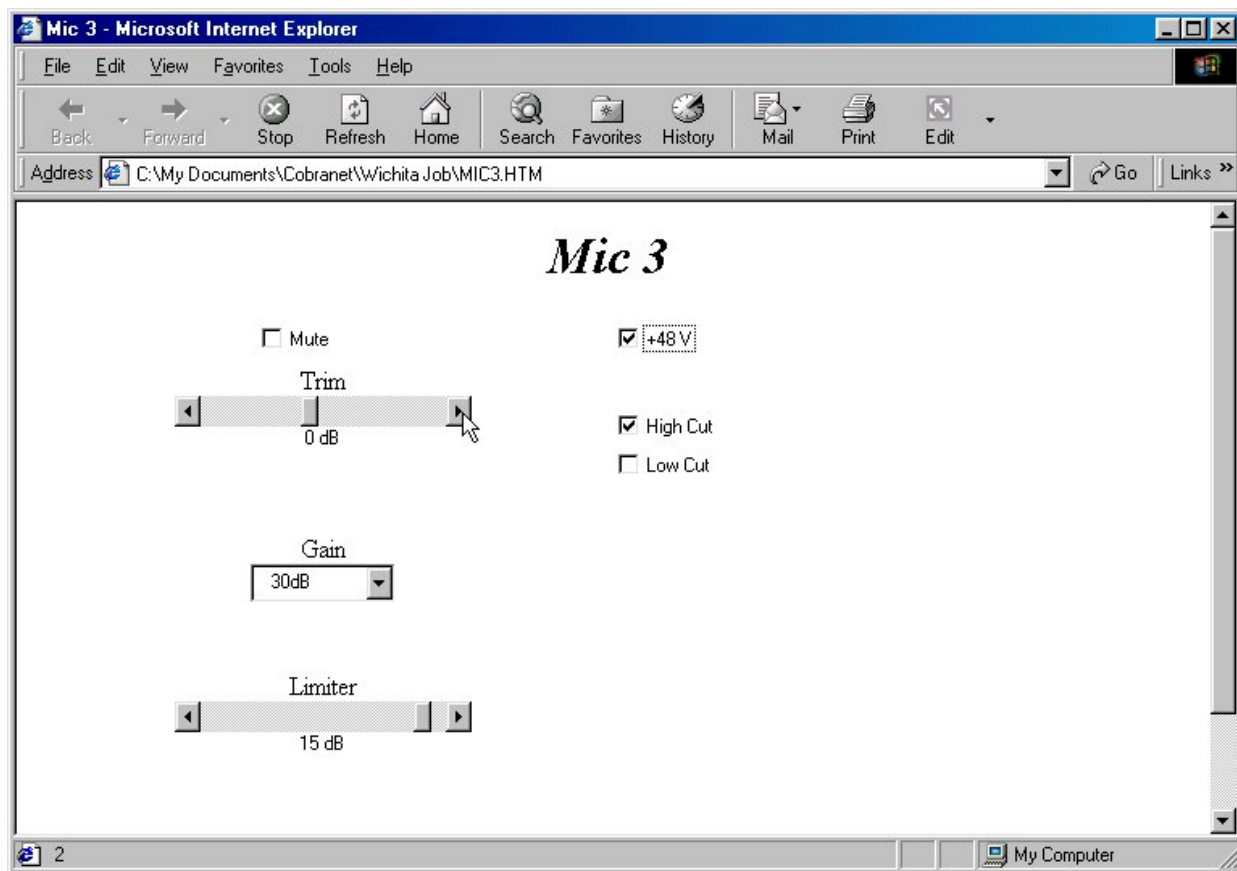
CobraNet Firmware

An important issue when combining CobraNet devices is being sure you're using compatible CobraNet firmware in all devices. I've never seen audio problems when using differing CobraNet firmware versions in different devices, only when controlling devices over the network have minor bugs been found. Luckily, all CobraNet devices support downloadable CobraNet firmware using Peak Audio's Disco (Discovery) software. So updates are easy once you have the proper files from the manufacturer (not Peak Audio).

When combining units from more than one CobraNet licensee and particularly if you're upgrading a current system where "older" and "newer" CobraNet devices are to be simultaneously used, check with the CobraNet licensees involved to be sure all devices do or will contain CobraNet firmware that works together. Firmware incompatibilities do not necessarily cause drastic problems by any means, but I like to say: *all bets are off until everything has compatible firmware.*



Main Page: The Main web page offers quick access to all 40 mic channels, but only the most often used parameters.



Detail Page: Each mic has its own detailed web page. All adjustable parameters for mic channel 3 are found here.

Here's an example of the software code pasted directly from the Microsoft Script Editor within Microsoft FrontPage 2000 for this church project. The comments are all preceded by the ' (apostrophe) symbol and are not part of the code. These comments attempt to explain the purpose of each line of code. This code is Microsoft's Visual Basic Script language which gets embedded into the web page's HTML code.

This is the code required for the above church system's Detail Page for Mic 3. Therefore, there is a similar chunk of code for each of the 40 Detail web pages – the only difference between each is the references to "mic 3" which need to be changed to reflect the required mic channel: 1 through 40. You can see why even a one-letter typing error in software code can generate a lot of time debugging. The code for the Main web page is quite a bit longer than this, but equally as repetitive in nature.

This: <SCRIPT ID...> delimiter marks the start of the Script code. When viewing this code in the Script Editor, you'll see a bunch of HTML code before this point.

```
<SCRIPT ID=clientEventHandlersVBS LANGUAGE=vbscript>
<!--
Option Explicit                'Tells VB Script that any variable it sees that is not
                                'Declared are Errors. This can save hours!

'Declarations of Global "Constants", like Pi, they can't be changed (easily!)
Const Mic = 3                  'Mic is #3 - Mics are 1 .. 8
Const Idx = 2                  'Idx goes from 0 .. 7, corresponding to Mic 1 .. 8
```

```

'Declare Global "Variables". These can and will change
Dim IgnoreChange

'*****
'*                               Subroutine window_onload                               *
'*                               *                                                       *
'* This "Subroutine" is called whenever the web page is loaded                       *
'* thus window_onload                                                         *
'*****
Sub window_onload
'Setup the Mic Gain's List Box (LB) by adding the selections of 15, 30, 45 and 60 dB
    Mic_Gain_LB.AddItem("15dB")
    Mic_Gain_LB.AddItem("30dB")
    Mic_Gain_LB.AddItem("45dB")
    Mic_Gain_LB.AddItem("60dB")

'Tell the NM84 ActiveX control to start talking to the unit
    NM84.Connect
End Sub

'*****
'*                               Subroutine NM84_Settings_Updated                               *
'*                               *                                                       *
'* Called whenever the NM84 ActiveX control detects that the device's             *
'* settings have changed.                                                       *
'*                                                                               *
'* It will automatically be called when communication is establish.             *
'*****
Sub NM84_Settings_Updated
'Ignore changes if the unit is not communicating properly
    if NM84.Online = FALSE then exit sub

'We are updating the web page controls, so generated events from the web page should be
'ignored to avoid conflicts
    IgnoreChange = TRUE

'Declare variables to hold the Mic settings we are interested in
    dim mute
    dim trim
    dim gain
    dim limiter
    dim PhantomPower
    dim filter

'Fill the variables with the settings from the unit
    mute =          NM84.Mic_Mute(Idx)
    gain =          NM84.Mic_Mic_Gain(Idx)
    limiter =       NM84.Mic_Limiter(Idx)
    PhantomPower =  NM84.Mic_Phantom_Power(Idx)
    filter =        NM84.Mic_Filter(Idx)

'The Trim needs to be adjusted to change its range from "-16 to + 20" -to- "36 - the setting"
    trim = 36 -          NM84.Mic_Trim(Idx)

'Sanity check the values to make sure the unit is communicating ok
    if (mute<0) or (mute>1) then
'Value is illegal, write a debug message
        Window.Status = "Bad Mic Mute value for Mic " & Mic & " = " & mute
    else
'Make the Mute Check Box match the Mic's Mute status
        Mic_Mute_CB.value = mute
    end if

'Sanity check the values to make sure the unit is communicating ok
    if (trim<0) or (trim>36) then
'Value is illegal, write a debug message
        Window.Status = "Bad Mic Trim value for Mic " & Mic & " = " & trim

```

```

        else
'Updated the Trim ScrollBar to match the Mic's Trim setting
    Mic_Trim_SB.value = trim
'Update the Text to reflect the setting
    Trim_Label.Caption = -20 + Trim & " dB"
    end if

'Sanity check the values to make sure the unit is communicating ok
    if (gain<0) or (gain>3) then
'Value is illegal, write a debug message
        Window.Status = "Bad Mic Gain value for Mic " & Mic & " = " & gain
    else
'Updated the Gain ListBox to match the Mic's Gain setting
        Mic_Gain_LB.ListIndex = gain
    end if

'Sanity check the values to make sure the unit is communicating ok
    if (limiter<0) or (limiter>48) then
'Value is illegal, write a debug message
        Window.Status = "Bad Mic Limiter value for Mic " & Mic & " = " & limiter
    else
'Updated the Limiter ScrollBar to match the Mic's Limiter setting
        Mic_Limiter.value = 48 - limiter
'Update the Text to reflect the setting
        Limiter_Label.Caption = 18 - limiter & " dB"
    end if

'Sanity check the values to make sure the unit is communicating ok
    if (PhantomPower<0) or (PhantomPower>1) then
'Value is illegal, write a debug message
        Window.Status = "Bad Mic Phantom Power value for Mic " & Mic & " = " & PhantomPower
    else
'Make the Phantom Power Check Box match the Mic's Phantom Power status
        Mic_PP_CB.value = PhantomPower
    end if

'Sanity check the values to make sure the unit is communicating ok
    if (Filter<0) or (Filter>3) then
'Value is illegal, write a debug message
        Window.Status = "Bad Mic Filter value for Mic " & Mic & " = " & Filter
    else
'Values for the Filter are      0 - Off
'                               1 - Low Cut
'                               2 - High Cut
'                               3 - Both
        if Filter > 1 then
            Mic_Hi_Cut.Value = 1
        else
            Mic_Hi_Cut.Value = 0
        end if
        if Filter AND 1 then
            Mic_Lo_Cut.Value = 1
        else
            Mic_Lo_Cut.Value = 0
        end if
    end if

'Ok, we are done updating the controls, so any further events are
'really caused by the User
    IgnoreChange = FALSE
End Sub

'*****
'*                               Subroutine Mic_Mute_CB_Change                               *
'*                                                                           *
'* Called whenever the Mic Mute CheckBox changes.                               *
'*****
Sub Mic_Mute_CB_Change
'First, did the User make the change from this web page, or did someone else on the network make it?
    if IgnoreChange then

```

```

        Exit Sub      ' Someone else made the change, ignore it
    End if

'Set the Mic's Mute to match the Check Box's value
    NM84.Mic_Mute(Idx) = Mic_Mute_Cb.Value
End Sub

'*****
'*
'          Subroutine Mic_Gain_LB_Change
'*
'*
'* Called whenever the Mic Gain ListBox changes.
'*
'*****
Sub Mic_Gain_LB_Change
'First, did the User make the change from this web page, or did someone else on the network make it?
    if IgnoreChange then
        Exit Sub      ' Someone else made the change, ignore it
    End if

'Set the Mic's Gain to match the Gain List Box's value
    NM84.Mic_Gain(Idx) = Mic_Gain_LB.ListIndex
End Sub

'*****
'*
'          Subroutine Mic_Hi_Cut_Change
'*
'*
'* Called whenever the High Cut CheckBox changes.
'*
'*****
Sub Mic_Hi_Cut_Change
'First, did the User make the change from this web page, or did someone else on the network make it?
    if IgnoreChange then
        Exit Sub      ' Someone else made the change, ignore it
    End if

'The Filter variable holds the calculate value for the filter setting
'this value is the combination of both the high and low cut Check Boxes
    Dim Filter
    Filter = 0

'Values for the Filter are      0 - Off
'                               1 - Low Cut
'                               2 - High Cut
'                               3 - Both
        if Mic_Lo_Cut.Value then Filter = Filter + 1
        if Mic_Hi_Cut.Value then Filter = Filter + 2

'Set the Mic's Filter to match the calculated filter value
    NM84.Mic_Filter(Idx) = Filter
End Sub

'*****
'*
'          Subroutine Mic_Lo_Cut_Change
'*
'*
'* Called whenever the Low Cut CheckBox changes.
'*
'*
'* Note: The High Cut's handler already does the filter calculation so just
'*       Call its handler.
'*
'*****
Sub Mic_Lo_Cut_Change
    call Mic_Hi_Cut_Change
End Sub

'*****
'*
'          Subroutine Mic_Limiter_Change
'*
'*
'* Called whenever the Limiter Scroll Bar changes.
'*
'*****
Sub Mic_Limiter_Change
'First, did the User make the change from this web page, or did someone else on the network make it?
    if IgnoreChange then
        Exit Sub      ' Someone else made the change, ignore it
    End if

```

```

        End if

'Calculate the new limiter value
    Dim Limiter
    Limiter = 48 - Mic_Limiter.value

'Update the Mic's Limiter setting
    NM84.Mic_Limiter(Idx) = Limiter
'Update the Limiter Text setting
    Limiter_Label.Caption = 18 - limiter & " dB"
End Sub

'*****
'*                               Subroutine Mic_PP_CB_Change                               *
'*                               *                                                       *
'* Called whenever the Phantom Power Check Box changes.                               *
'*****
Sub Mic_PP_CB_Change
'First, did the User make the change from this web page, or did someone else on the network make it?
    if IgnoreChange then
        Exit Sub      ' Someone else made the change, ignore it
    End if

'Update the Mic's Phantom Power setting
    NM84.Mic_Phantom_Power(Idx) = Mic_PP_CB.Value
End Sub

Sub Mic_Trim_SB_Change
'First, did the User make the change from this web page, or did someone else on the network make it?
    if IgnoreChange then
        Exit Sub      ' Someone else made the change, ignore it
    End if

'Calculate the new Trim value
    Dim Trim
    Trim = 36 - Mic_Trim_SB.value

'Update the Mic's Trim setting
    NM84.Mic_Trim(Idx) = Trim
'Update the Trim's Text setting
    Trim_Label.Caption = -20 + Trim & " dB"
End Sub

-->
</SCRIPT>
</head>

```

This: </SCRIPT> delimiter marks the end of the Script code. When viewing this code in the Script Editor, you'll see a bunch of HTML code after this point.