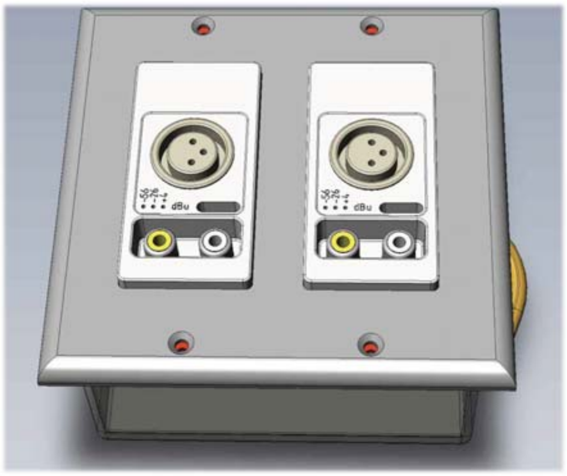
MEDIA**TECHNOLOGY**SYSTEMS INC.



PROGRAMMING MANUAL

MTS products with CobraNet™™ Interfaces

5818 CALVIN AVENUE, TARZANA, CALIFORNIA 91356 U.S.A. www.mediatechnologysystems.com
Part # MAN-0308-MCA-RevB

Table of Contents

1 ION2.0/ION0.2 Simple Configuration (No DSP)

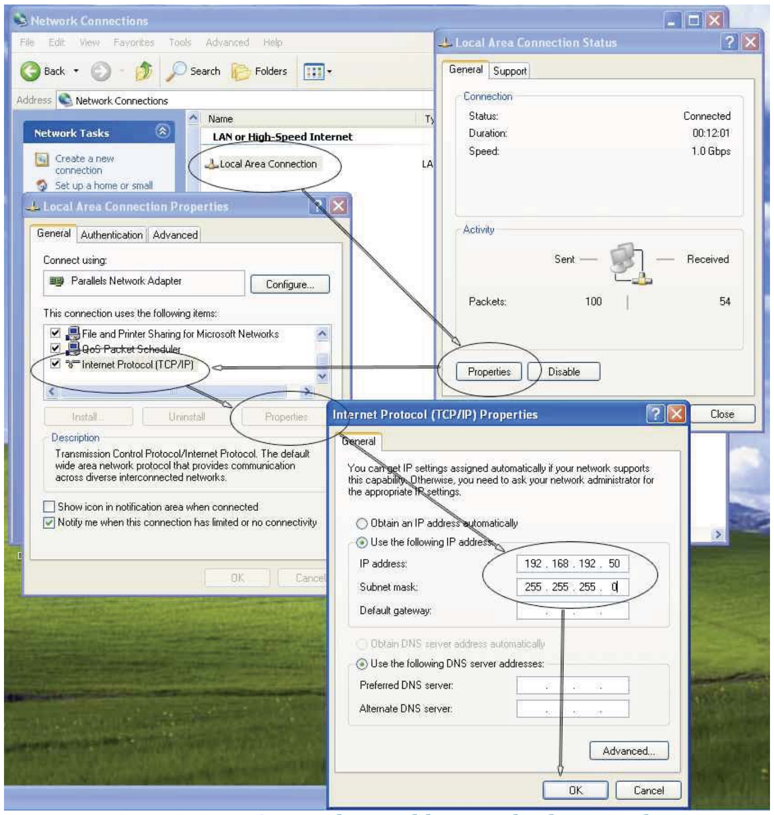
In many applications, the ION2.0 and ION0.2 will be used as a simple 2 channel Cobranet interface, providing local analog inputs or outputs (respectively) and connecting to a central DSP processor. In that situation, the DSP will not be used. This Section covers the use of the ION2.0/0.2 as a simple 2 channel interface and Section 6 cover the more complex DSP configuration and control

The standard CobraNet™ tools, including CobraNet™ Discovery (CNDISCO) and CobraCAD are available for use with the MTS CobraNet™ enabled IONs. These tools are available as a free download from the Cirrus Logic Website.

The utility CNDISCO is the simplest method of configuring the ION2.0 and ION0.2 interfaces. Simply download from the Cirrus Logic website and run the exe file on the PC or laptop.

1.1 CNDISCO - Setup

To use CNDISCO, the host PC or laptop must be set to the default IP subnet in order to talk to the ION. Figure 1-1 below shows the method of setting up a Windows computer.


Network Connections
Local Area Connection
Local Area Connection Properties
Internet Protocol (TCP/IP)
Properties
Local Area Connection Status
Properties
Internet Protocol (TCP/IP) Properties
OK
Cancel

Local Area Connection Status: Status: Connected; Duration: 00:12:01; Speed: 1.0 Gbps; Packets: Sent 100, Received 54

Local Area Connection Properties: Connect using: Parallels Network Adapter. This connection uses the following items: File and Printer Sharing for Microsoft Networks; QoS Packet Scheduler; Internet Protocol (TCP/IP)

Description: Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.

Internet Protocol (TCP/IP) Properties: You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

Use the following IP address:
IP address: 192 . 168 . 192 . 50
Subnet mask: 255 . 255 . 255 . 0

Figure 1-1: Setting up the IP address and subnet mask.

- Go to Control Panel and then open "Network connections".
- Click on the General tab and open "Properties".
- Select "Internet Protocol (TCP/IP)" and click on properties.
- Finally, change the selection from "Obtain an IP address automatically" to "Use the following IP address" and set to your desired IP domain, eg:-
    - IP address: 192.168.192.50
    - Subnet mask: 255.255.255.0
- After finishing using the CNDISCO application, return to the Control Panel and reset the selection back to "Obtain an IP address automatically".

If an intelligent or managed switch/router is in use, then the switch address will need to be set to the same subnet, usually 192.168.1.1 or 192.168.1.254 are the most common default addresses.

1.2 Advanced settings

In order to use CNDISCO effectively, it will be necessary to enable the configuration and advanced features. This will also allows you to put any version of firmware on any hardware-compatible CobraNet™ module. CNDISCO needs to have the particular firmware version of a device in its firmware directory in order to properly identify the device for compatible firmware upgrades. Should the situation arise where you know the device is a specific model but CNDISCO says there are no compatible firmware upgrades, using the advanced feature, you'll be able to update the firmware anyway.

How to enable the advanced feature:

Open cndisco.ini (WinXP) or the config file (WinVista/Win7) the in Notepad

Its usually in a directory like this: C:\Program Files\Peak Audio\CobraNet™ Discovery.

Find the Configuration section. It usually looks something like this:

```
[Configuration]
Adapter Index=[10] [10] Broadcom NetXtreme 57xx Gigabit Controller
Firmware Location=C:\Program Files\Peak Audio\CobraNet™ Discovery\firmware
```

Start a new line after one of the lines in that section and type in Advanced Feature=1.
Add CC_Enable=1 under Advanced Feature=1

It should look something like this when you're done:

```
[Configuration]
Adapter Index=[10] [10] Broadcom NetXtreme 57xx Gigabit Controller
Firmware Location=C:\Program Files\Peak Audio\CobraNet™ Discovery\firmware
Advanced Feature=1
CC_Enable=1
```

For WinVista/Win7, the line is slightly different, ie, change…

```
<add key "Advanced Feature" value="0" />
```

to...

```
<add key "Advanced Feature" value="1" />
```

Save the changed .ini/config file and exit Notepad. The advanced features are now enabled.

Now when you update the firmware you'll see a check box in the "Select Firmware Version" dialog box marked "Show All Firmware Versions". Check the box and you'll be able to choose from all the firmware versions stored in the firmware directory.

1.3 Configuration

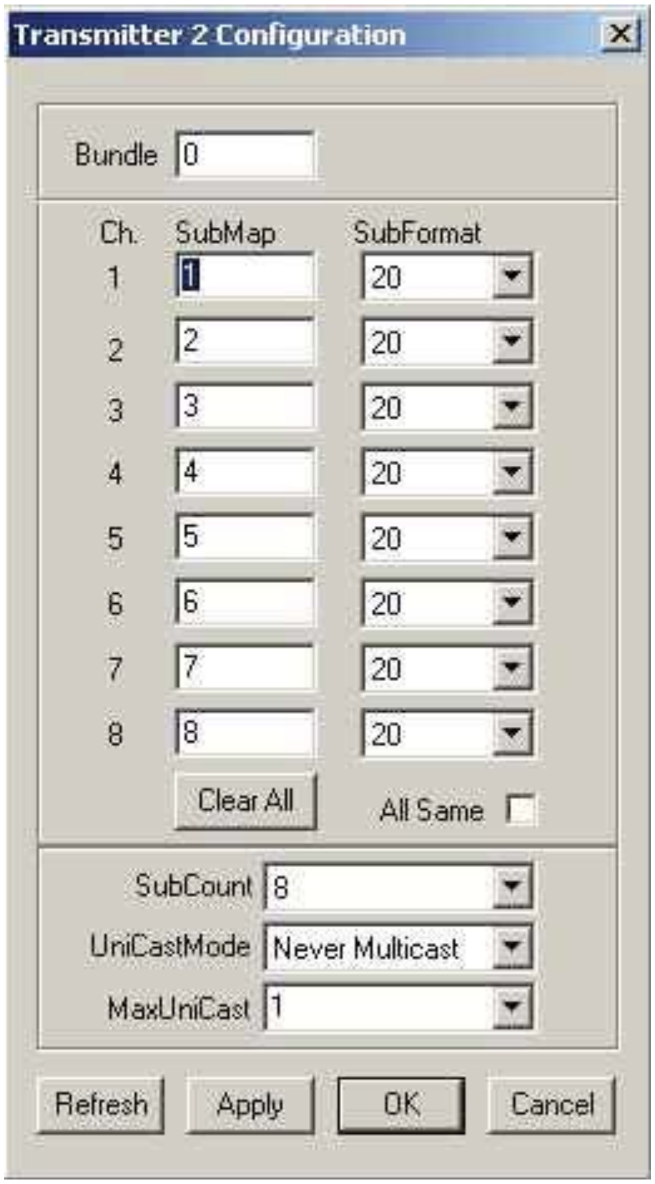
The CNDISCO manual (found in the C:\Program Files\Cirrus Logic\CobraNet™ Discovery folder) will explain in detail most of the configuration processes, so these have not been repeated here. However, there are some useful features of the CobraNet™ protocol that are not covered explicitly, ie…

One of the key features of the ION product is the ability to set up to 4 CobraNet™ audio transmitters (ION2.0) and 8 CobraNet™ receivers (ION0.2). In addition, MTS has provided the ability to set each bundle subchannel configuration.

The settings are:-

- Transmitter setup: This section covers the CobraNet™ transmitters (see Figure 1-2). The CS496112 chipset allows for up to 4 transmitters, each of up to 8 channels, subject to an overall channel count of 2 analog input channels and 8 network audio streaming channels. The settings are:-
    - Bundle number: This sets the bundle address of each transmitter. The bundle numbers are 0 (off, ie no transmission), 1-255 are multicast, 256-65279 are unicast and 65280-65535 are private.
    - Unicast mode: If the transmitter bundle address is normally unicast (>255), but more than one receiver is available for that bundle address, then the bundle can be transmitted either multicast or multi-unicast.
    - Max Unicast: Depending on unicast mode, the maximum number of multi-unicast bundles can be set between 1 and 4.
    - Transmitter1…Transmiter4: This lists the four transmitters associated with the bundle address and allows the user to set the audio subchannels associated with that bundle. The subchannel mapping allows the user to decide which of the 8 audio channels are mapped to each bundle and in which order they are transmitted.
    - Subformat Resolution: This sets the word length of the transmitted audio to 16, 20, or 24 bit. Note: if the word depth is set to 24bit, then only 7 audio channels can fit in one bundle.

- UnicastMode: This value can be used to override or modify the normal unicast vs. multicast implications of the assigned bundle number. The normal default value is 'Never Multicast'. The available options are:
  - Always Multicast – All bundles are sent multicast regardless of Bundle number.
  - Multicast over 1 – If more than one receiver is set to receive this bundle, it will be multicast, else it will be Unicast
  - Multicast over 2 – If more than two receivers are set to receive this bundle, then it will be multicast, else it will be unicast or multi-unicast
  - Multicast over 3 – If more than three receivers are set to receive this bundle, then it will be multicast, else it will be unicast or multi-unicast
  - Multicast over 4 – If more than four receivers are set to receive this bundle, then it will be multicast, else it will be unicast or multi-unicast
  - Never Multicast – Only a single bundle will be sent unicast


Transmitter 2 Configuration

Bundle: 0

Ch.	SubMap	SubFormat
1	1	20
2	2	20
3	3	20
4	4	20
5	5	20
6	6	20
7	7	20
8	8	20

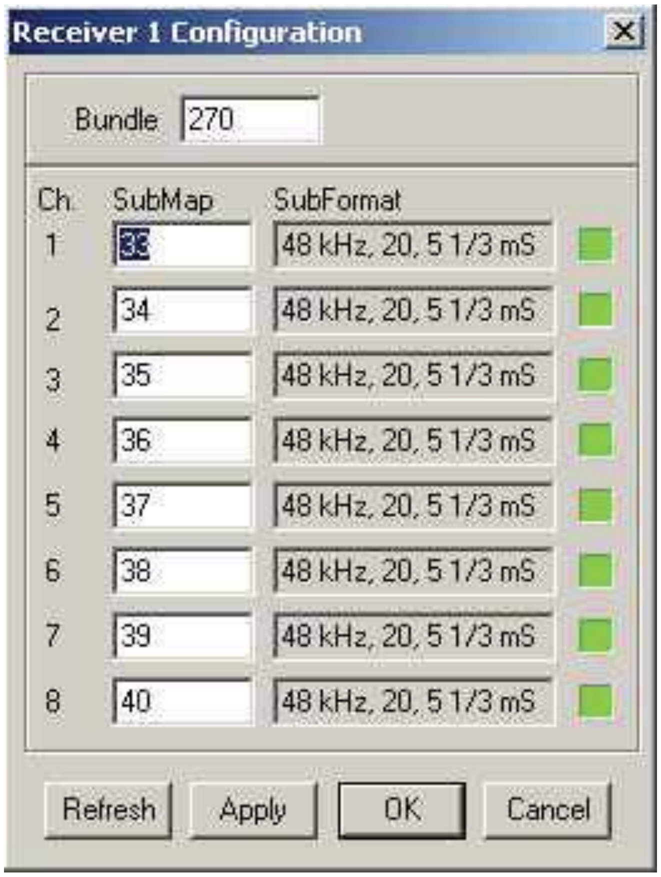
Clear All   All Same ☐

SubCount: 8
UniCastMode: Never Multicast
MaxUniCast: 1

Refresh   Apply   OK   Cancel

Figure 1-2: CobraNet™ Transmitter settings page

- Receiver setup: This section covers the CobraNet™ receivers (see Figure 1-3). The CS496112 chipset allows for up to 8 receivers, each of up to 8 channels, subject to an overall channel count of 2 analog output channels and 8 network audio streaming channels.. The settings are:-
    - Bundle number: Same process and limitations as described in the transmitter section
    - Receiver active: This LED only lights if there is a valid transmitter sending audio on that bundle address and channel.
    - RX1…RX8: Same process and limitations as described in the transmitter section


Receiver 1 Configuration

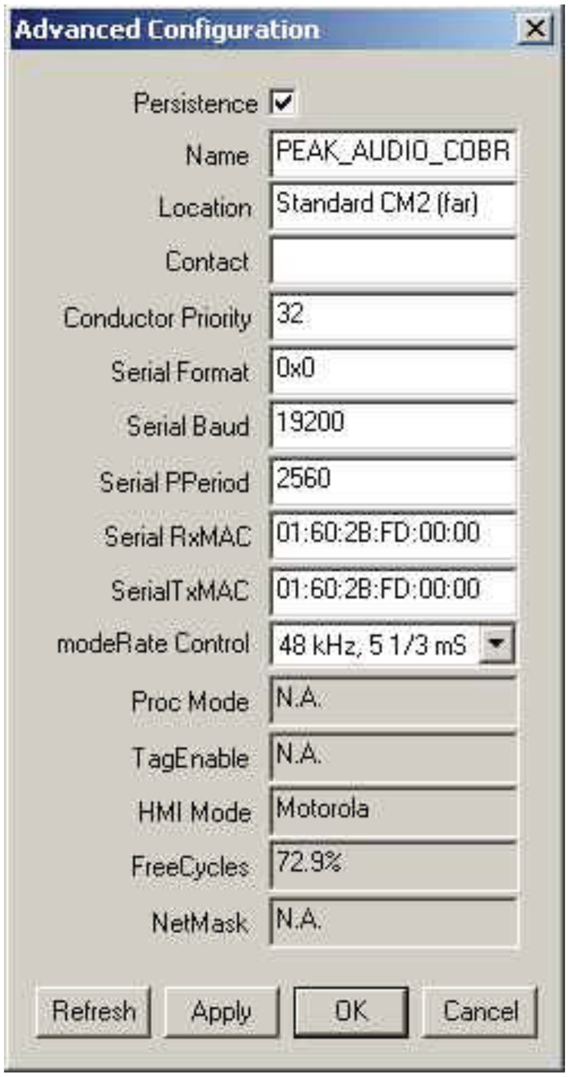
Bundle 270

Ch.	SubMap	SubFormat	
1	33	48 kHz, 20, 5 1/3 mS	🟩
2	34	48 kHz, 20, 5 1/3 mS	🟩
3	35	48 kHz, 20, 5 1/3 mS	🟩
4	36	48 kHz, 20, 5 1/3 mS	🟩
5	37	48 kHz, 20, 5 1/3 mS	🟩
6	38	48 kHz, 20, 5 1/3 mS	🟩
7	39	48 kHz, 20, 5 1/3 mS	🟩
8	40	48 kHz, 20, 5 1/3 mS	🟩

Figure 1-3: CobraNet™ Receiver settings page

- Main Interface settings: This section covers the more advanced variables not usually associated with bundle management and which apply to the CobraNet™ device globally (see Figure 1-4). These are explained in detail in the CNDISCO manual, but 2 are of particular importance to the ION series interfaces, ie…
    - Persistance: The ION2.0 and ION0.2 is a very simple interface product and does not have a preset memory, so "persistence" is used to store the last settings in case of power down. Please note…
        - All CobraNet™ settings need up to 1 minute to establish persistence, as they are stored in the CobraNet™ flash. If the ION power is cycled before the settings are stored to flash, then the settings will be lost.

        - The persistence only applies to Cobranet settings and not the DSP settings. DSP settings will need to be stored using MTS Control and a Preset device (see 4.5).
        - If the persistence tick box is off, then no settings will be saved.
    - Mode Rate Control: The options are 1.33mS, 2.66mS or 5.33mS latency. Note: there are significant trade-offs if changes are made to the 5.33mS default settings (see PM25), particularly in terms of the number of switch hops that can be used. If the ION interface is being used with a simple local network with a single Ethernet switch, then 1.33mS can be safely used. If more than 1 switch, then use 2.67mS. If more than 3 switches, then use 5.33mS.
    - The "Location" is a useful way of uniquely naming the ION2.0 or ION0.2 interface. Up to 60 characters, eg "Ballroom 3: Stage left, Mics 7/8". For more detailed naming information in a large project, both the "Location" and "Contact" fields can be used.


Advanced Configuration

Persistence	✓
Name	PEAK_AUDIO_COBR
Location	Standard CM2 (far)
Contact	
Conductor Priority	32
Serial Format	0x0
Serial Baud	19200
Serial PPeriod	2560
Serial RxMAC	01:60:2B:FD:00:00
SerialTxMAC	01:60:2B:FD:00:00
modeRate Control	48 kHz, 5 1/3 mS
Proc Mode	N.A.
TagEnable	N.A.
HMI Mode	Motorola
FreeCycles	72.9%
NetMask	N.A.

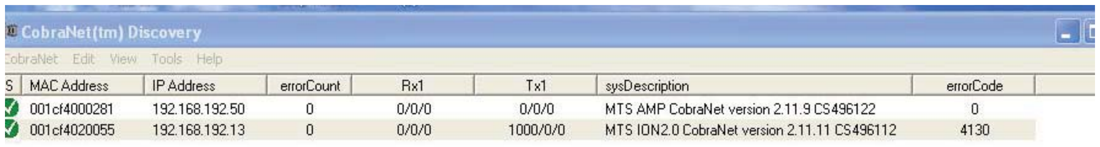
Figure 1-4: Global Interface settings

1.4 Presets

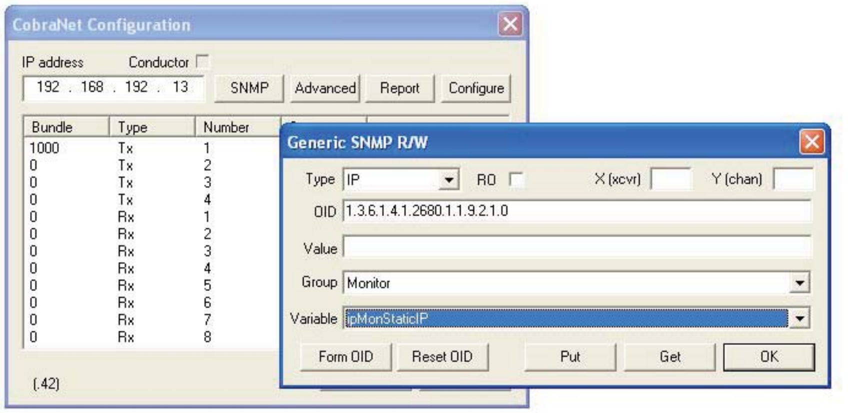
The ION2.0 and ION0.2 are simple devices and do not contain a host processor and no access to multiple preset functionality. However, the Cobranet flash can be used to save Cobranet interface settings via "persistence" (see previous section and Figure 1-4). If the persistence option is enabled, then the last set of values/settings can be stored into the Cobranet flash and will be restored on power up. Note that these settings can take up to 1 minute to save, as they are stored in between other processes. Also note that DSP settings are not stored-only Cobranet interface settings.

1.5 Setting a static IP address

First set persistence on (See Advanced in Section 5.3). Then double click on the device in the main CNDISCO window to open the configuration menu (see below). In the configuration menu select the "SNMP" button. An SNMP window will open and select the "Monitor" Group and the "ipMonStaticIP" variable.



S	MAC Address	IP Address	errorCount	Rx1	Tx1	sysDescription	errorCode
✓	001cf4000281	192.168.192.50	0	0/0/0	0/0/0	MTS AMP CobraNet version 2.11.9 CS496122	0
✓	001cf4020055	192.168.192.13	0	0/0/0	1000/0/0	MTS ION2.0 CobraNet version 2.11.11 CS496112	4130



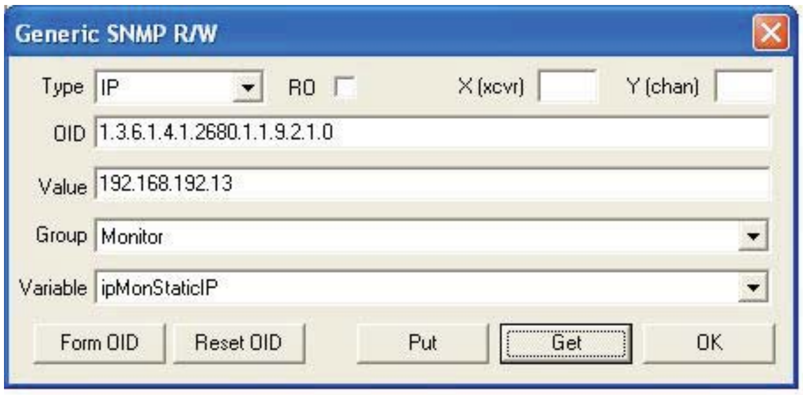
IP address 192 . 168 . 192 . 13

Bundle	Type	Number
1000	Tx	1
0	Tx	2
0	Tx	3
0	Tx	4
0	Rx	1
0	Rx	2
0	Rx	3
0	Rx	4
0	Rx	5
0	Rx	6
0	Rx	7
0	Rx	8

(.42)

Type IP
OID 1.3.6.1.4.1.2680.1.1.9.2.1.0
Value
Group Monitor
Variable ipMonStaticIP

In the value section, type the desired IP address in AAA.BBB.CCC.DDD format and then press "PUT". Confirm the setting by pressing "GET" See below for an example.



Type IP
OID 1.3.6.1.4.1.2680.1.1.9.2.1.0
Value 192.168.192.13
Group Monitor
Variable ipMonStaticIP

2 DSP Configuration

The DSP control and monitoring parameters are available via SNMP OID’s. As these OID’s are 32 bit registers with non-intuitive numbers (eg, +12dB of gain on a mixer is represented by the 32 bit value 534330399), it is recommended that the MTS version of Stardraw Control (SDC) is used for control and monitoring application. Both MTS SDC application and programmers manual are available as a download from the MTS website.

For those intrepid explorers who wish to create a custom driver or script for the ION2.0 or ION0.2, the information on how to access and use the OID’s are given in below

NOTE: The manual refers to examples of the ION2.0 or ION0.2, but the principles are applicable to all MTS Cobranet products

2.1 Interpreting the OID’s.

All SNMP variables are referenced by the object identifier (OID). An OID is a dot-separated string of integers representing the path from the root of the SNMP management information base (MIB) to the variable. Firmware supporting DSP Conductor features a dspExtensions SNMP extension agent rooted at…

iso(1).org(3).dod(6).internet(1).private(4).enterprises(1).peakAudio(2680).cobraNet(1).dspExtensions(4).control(2).

Then a signature is added. The signature is used to generate a 6 digit section of the OID string and ensures that the version of the DSP schematic is valid and true. Any change to the schematic (but not individual parameter values) will generate a new signature shown as follows…

iso(1).org(3).dod(6).internet(1).private(4).enterprises(1).peakAudio(2680).cobraNet(1).dspExtensions(4).control(2).signature(aa.bb.cc.dd.ee.ff).

DSP Conductor parameters under this extension are separated into two tables of 32-bit integers. The first table, rooted at…

dspExtensions(4).control(2).signature(aa.bb.cc.dd.ee.ff).controlRWTable(2)

…holds read-write parameters. The second table, rooted at…

dspExtensions(4).control(2).signature(aa.bb.cc.dd.ee.ff).controlROTable(4)

…holds read-only parameters.

Note that access to SNMP tables is achieved using 1-based indices. The word offsets specified in the configuration header file are 0 based.

To construct an OID for a read/write variable, append the parameter's word offset (wo) plus one to the OID for the controlRWTable values:

iso(1).org(3).dod(6).internet(1).private(4).enterprises(1).peakAudio(2680).cobraNet(1).dspExtensions(4).control(2).signature(aa.bb.cc.dd.ee.ff).controlRWTable(2).controlRWEntry(1).controlRWValue(2).wordoffset+1(wo+1)

For example, the OID for a read/write parameter with word offset 5 is…

1.3.6.1.4.1.2680.1.4.2.aa.bb.cc.dd.ee.ff.2.1.2.6

To construct an OID for a read-only variable, append the parameter's word offset (wo) plus one to the OID for the controlROTable values:

iso(1).org(3).dod(6).internet(1).private(4).enterprises(1).peakAudio(2680).cobraNet(1).dspExtensions(4).control(2).signature(aa.bb.cc.dd.ee.ff).controlROTable(4).controlROEntry(1).controlROValue(2).wordoffset+1(wo+1)

For example, the OID for a read-only parameter with word offset 13 is…

1.3.6.1.4.1.2680.1.4.2.aa.bb.cc.dd.ee.4.1.2.14

The full list of SNMP OID's for the ION2.0 and ION0.2 are given in the xml file in the CDROM. The OID's represent a table of 32 bit registers in the memory map of the Cobranet chipset. Each 32 bit register provides either a read only (RO) register for monitoring DSP values (usually signal or status information), or a read/write (RW) register for controlling DSP values (gain, threshold, delay, etc).

Although the xml file is an intimidating list of possible parameters, much of the information will be irrelevant to the needs of a particular application.

2.2 Crunch Values

All OID's are 32-bit values. How the value is interpreted depends on the parameter. The way values are interpreted can be discerned by examining the Crunch Functions section in the element's implementation XML file. For details on the implementation XML file refer to Cirrus application note AN277, "Creating DSP Conductor Primitives".

Note: A doubling of level is 6dB. 1 bit in the digital domain doubles a value (ie shifting a number 1 bit to the left is the same as multiplying by 2). Thus having (for example) a signed 29 bit fractional value, allows the use of the 32nd bit for sign, the 31st and 30th bits for +12dB of gain and the 29th to 0th bits for greater than -100dB of attenuation.

An example of how this interprets to Mixer gain is given in Figure 2-1 below.

Mixer NxM Input gain (dB)	Non-inverted OID value 2^29	Inverted OID value 2^29
24	8508830537	-8508830537
12	2137321597	-2137321597

0	536870912	-536870912
-12	134855876	-134855876
-24	33874264	-33874264
-36	8508831	-8508831
-48	2137322	-2137322
-64	338743	-338743
-100	5369	-5369
Mute	0	0

Figure 2-1: Example of Mixer Gain OID values

The algorithm used above is POWER(10, (Target dB value)/20)*2^29. This will give the needed OID value.

Integer values (eg Bypass and Router) are usually defined as fractional bits = 0. These values are simple, as they are represented by a conventional integer, as shown in Figure 2-2 and Figure 2-3 below.

Bypass	OID value
OFF	0
ON	1

Figure 2-2: Bypass OID value

Router (OUT)	OID value
OFF/MUTE	0
IN 1 to OUT	1
IN 2 to OUT	2
IN 3 to OUT	3
IN 4 to OUT	4
…	…
Inn to OUT	n

Figure 2-3: Router OID value

Simple timing values are defined by using the sample rate divided by the block size. The Hold Time shown in Figure 2-4 is calculated by the algorithm HT = (Desired Time in Seconds)*48000/16, where the sample rate is usually 48kHz and the block size is 16.

Hold time (S)	OID value Sample/Block
10	30000
1	3000
0.1	300

Figure 2-4: Hold time OID

Other dB values (eg, Threshold, Depth and Knee) are calculated by the algorithm

(Target dB value)*2^23. See Figure 2-5 for example values.

Threshold/ Depth/Knee (dB)	OID value = dB x 2^23
24	201326592
12	100663296
0	0
-12	-100663296
-24	-201326592
-36	-301989888
-48	-402653184
-64	-536870912
-100	-838860800

Figure 2-5: Depth, Threshold & Knee OID values

Other values are calculated from the Sample rate. For example the Frequency of a sine wave generator is calculated by F = 2*(Desired Frequency in Hertz)/48000 and then the result multiplied by 2^31 to get a 2 bit value – see Figure 2-6 below

Sine wave Freq	2*F/Sample Rate OID value 2^31
20	1789570
100	8947849
500	44739243
1000	89478485
10000	894784853
20000	1789569707

Figure 2-6: Frequency OID values

While the sign (sg) and fraction bits (fb) attributes give some indication of how the DSP might interpret the values (as shown in the examples above), it is the specifics of the DSP implementation that determine exactly how values are interpreted. The full suite of crunch values are found in the DSP Conductor folder shown below. DSP conductor can be downloaded from the Cirrus Logic website.

C:\Program Files\Cirrus Logic\DSPConductor\plugins\coyote\devices

The "imp" xml file contains details of the crunch functions. These can be opened in the xml editor embedded in Internet Explorer. For example the crunch functions for a sine wave generator are found in the file "generator_sine_wave.imp.xml". The file contains the text for a crunch function written in python, as shown below…

```
scale_factor = float( property[ "scale_factor" ] )
```

```
sample_rate = float( property[ "sample_rate" ] ) * scale_factor
block_size = float( property[ "block_size" ] ) * scale_factor
ramp_time_constant = float( property[ "ramp_time_constant" ] )

block_rate = sample_rate / block_size
ramp = 1 - math.exp( -1 / ( ramp_time_constant * block_rate ) )

omega = 2 * frequency / sample_rate;
gain = -math.pow( 10, ( level - 20 ) / 20 ) / 16

if mute :
gain = 0
```

The crunch function can easily be converted to C# or C++ code for use in other applications.

2.3 OID List for all MTS Cobranet products

The *.xml files on the CDROM has a complete list of all the OID's and offsets for the amplifier read/write and red only values, where the devices relate to the schematic in Section 4 above.

2.4 Metering values

The DSP domain metering and headroom has some slight differences compared to the analog domain, primarily due to the differences in internal word length compared to the conversion word length (usually 20bit for Cobranet). Also the polling speed will affect the real time accuracy. However, the ION2.0/ION0.2 inputs and outputs should approximately correspond (within a dB or so)with the DSP meters as follows…

2.4.1 ION0.2

Assuming that the input and output level shifters (DAI/DAO) are set to unity gain, then applying an external sine wave (+3dB crest factor) will have the following results…

- -8dBu (0.32 Volts RMS) at the XLR (minimum gain) = -8dB (DSP Meter)
- 0dBu (0.775 Volts RMS) at the XLR (minimum gain) = 0dB (DSP Meter)
- -8dBu (0.32 Volts RMS) at the RCA = +4dB (DSP Meter)

NOTE: Applying 30dB and 60dB of gain at the XLR will reduce the input signal required to obtain the same DSP meter reading.

2.4.2 ION0.2

Assuming that the input and output level shifters (DAI/DAO) are set to unity gain, then setting an internal sine wave (+3dB crest factor) will have the following results…

- 0dBu (DSP Meter) = 0dBu (0.775 Volts RMS) at the XLR (minimum sensitivity) and -8dBu at the RCA

- +18dB (DSP Meter) = +18dBu (6 Volts RMS) at the XLR (minimum sensitivity) and +10dBu at the RCA

NOTE: The circuit is designed so that when you Pin 2 or Pin 3 of the XLR is shorted to pin 1 it will turn off the op-amp that is driving that pin and increase the level at the other pin 6dB, (+/-0.2dB), so that the gain says about the same as in balanced mode. However, the output will lose 6dB of headroom and clip at +18dBu.

3 Control & Monitoring software – Quickstart executable

The CDROM contains a number of executable files that will show the basic features and configuration of the ION2 series products.

Each executable will run on a standard Windows PC (XP, Vista, Win7) with .net3.5 or higher.

NOTE: In order to use the executables, it will be necessary to use CNDISCO to set an IP address into the ION device (see section 1.3 above). Also, if the settings are to be saved to the ION, then use CNDISCO to set "persistence" ON.

3.1 ION2.0

Launching ION20_rev1_1.exe will show the Main Screen, as shown in Figure 3-1 below.

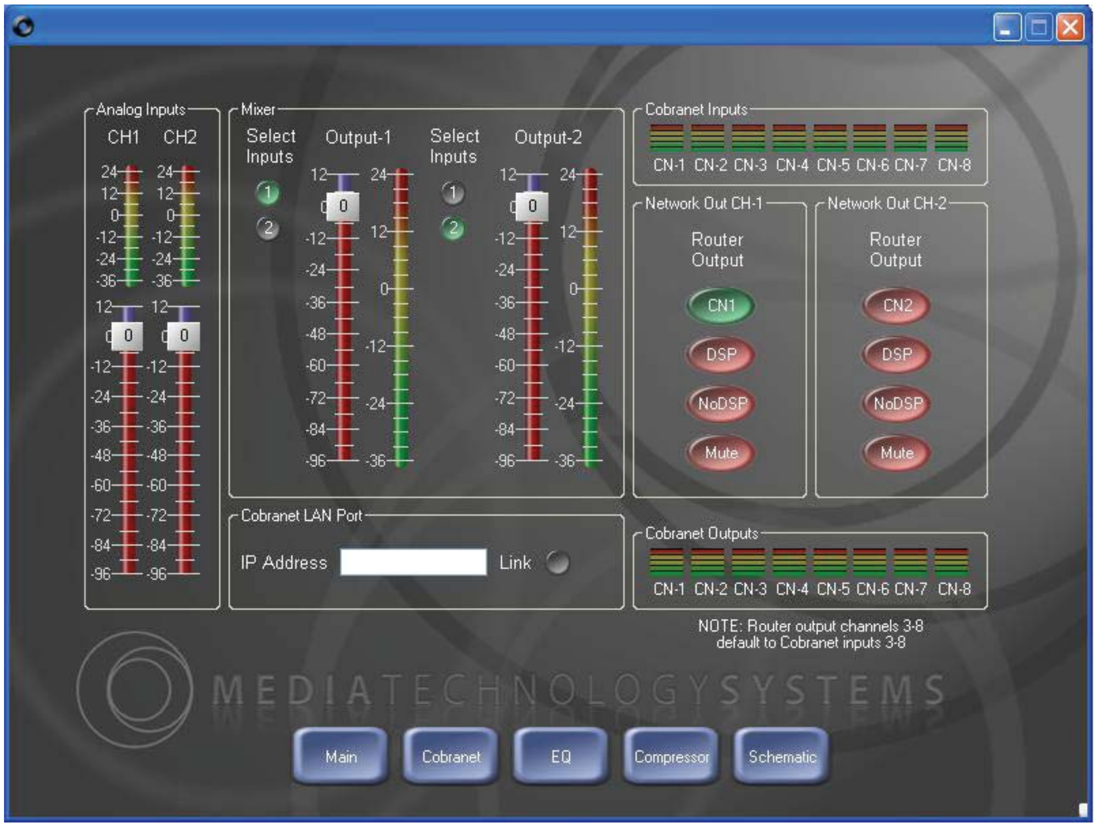


Figure 3-1: ION2.0 Main Screen

The Main Screen is reasonably self-explanatory, but the DSP schematic is available the sequence of the controls (see Figure 3-2 below). The "Schematic" button will open a graphic in a child window, allowing it can be left on the screen.

The first function will be to set the IP address and if it has connected successfully to the ION, the Link indicator will turn green.

The Analog Inputs section shows the DAI input sliders (prior to the Analog In Meters) and the "Analog In Meters". The Mixer section allows either of the 2 input channels to be routed to each output, along with a slider and the "Analog In Mixer Meters".

The 8 channels of meters in the Cobranet Inputs and Outputs refers to the "Network In Meters" and "Network Out Meters" respectively in the schematic.

The Network Out CH1 and CH2 routing refers to the 12x8 "Network Out Router" in the schematic and selects between Cobranet input 1/2 (CN1/2), Audio Input 1/2 direct (No DSP), Audio Input 1/2 through the DSP side chain (DSP) or Router Channel Mute (Mute).

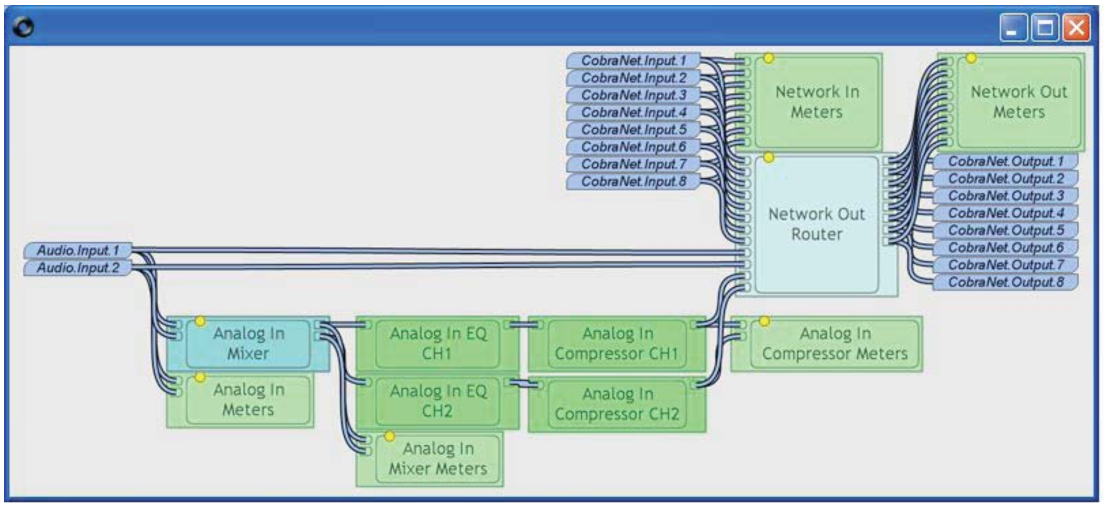


Figure 3-2: ION2.0 DSP Schematic

Pressing the Blue "Cobranet" button will open the Cobranet settings as shown in Figure 3-3 below.

The only settings required for a "quickstart" would be…

- TX1: This is the first Cobranet transmitter and the bundle address is 1-255 for multicast and 256 and above for unicast. Setting the bundle address to 0 will switch the transmitter off.
- txSubCount: As the ION is only 2 channels, txSubCount can be set to 2.
- Resolution: Usually set to 20bit.

Once the transmitter (TX1) bundle address has been set, then as long as there is a corresponding receiver on the same bundle address, then audio will pass over the network and the TX1 green LED will light up showing that it is active.

Unicast mode, Max Unicast, Subchannel mapping, etc are advanced settings and are explained in (see section 1.3 above). Full details are available in the Cirrus Logic PM25 programmers manual, which is a free download from the Cirrus website.

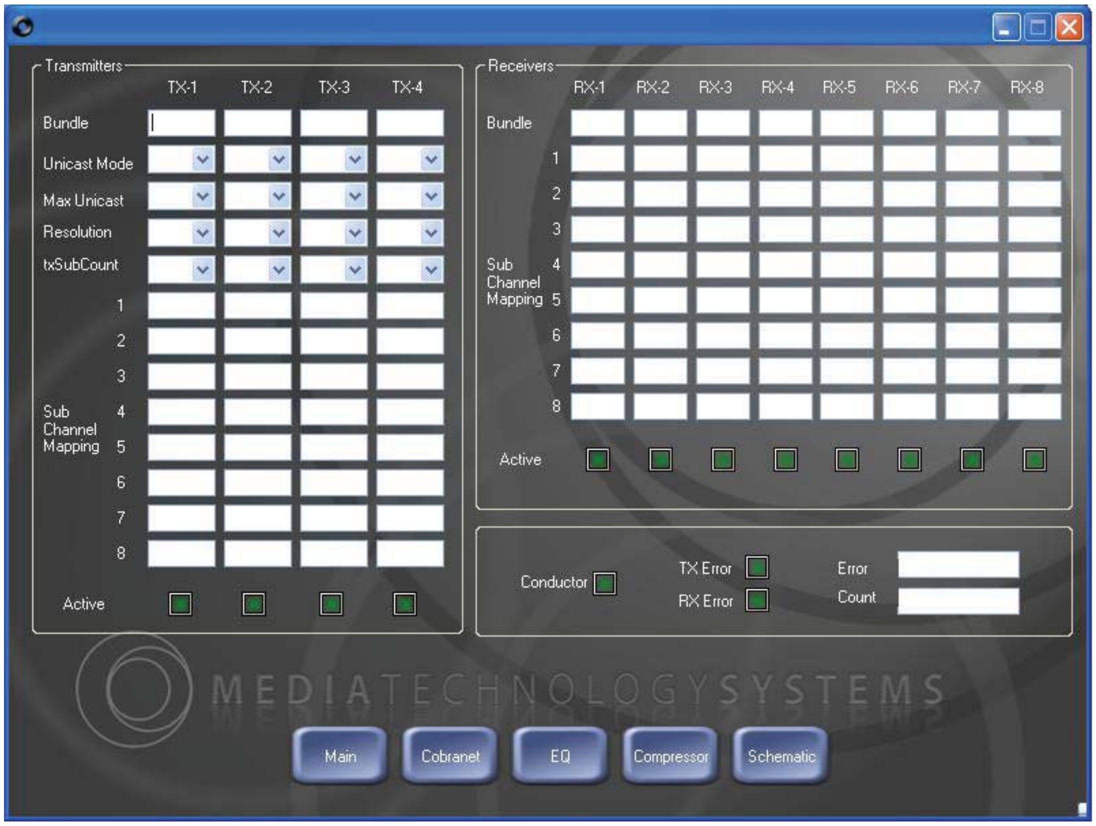
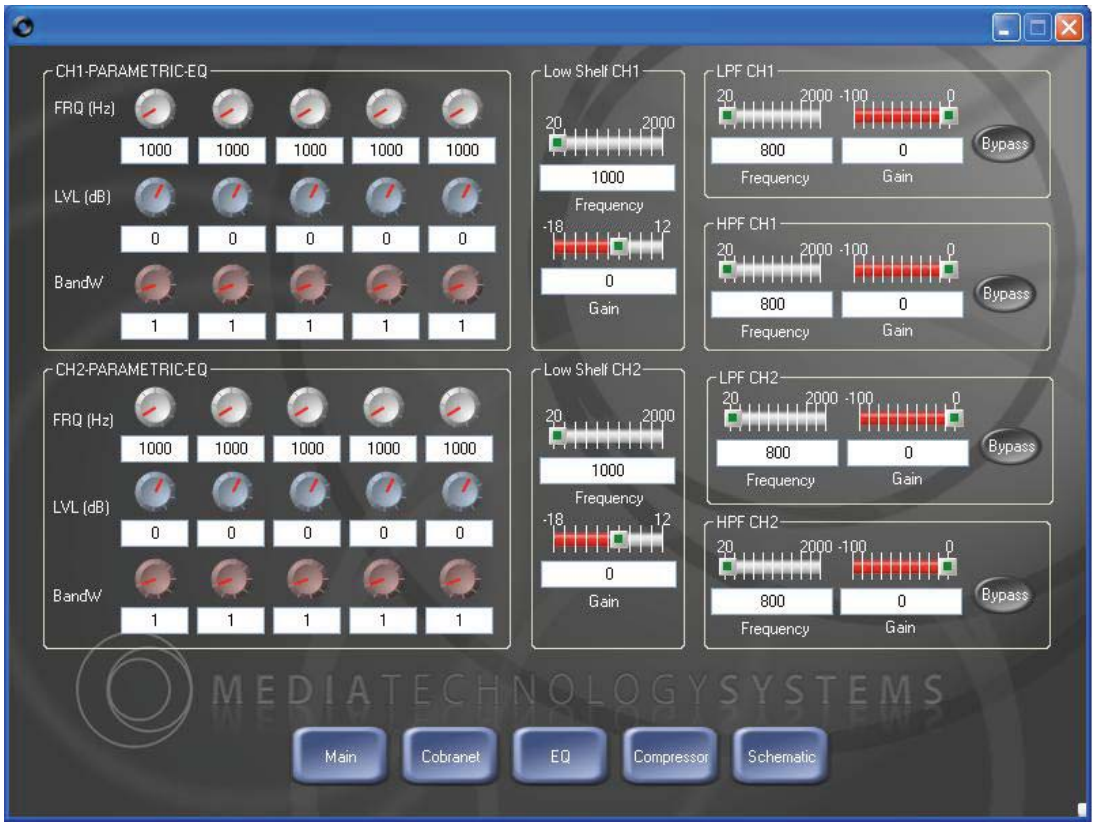


Figure 3-3: Cobranet settings



CH1-PARAMETRIC-EQ

FRQ (Hz)	1000	1000	1000	1000	1000
LVL (dB)	0	0	0	0	0
BandW	1	1	1	1	1

Low Shelf CH1: Frequency 1000, Gain 0

LPF CH1: Frequency 800, Gain 0

HPF CH1: Frequency 800, Gain 0

CH2-PARAMETRIC-EQ

FRQ (Hz)	1000	1000	1000	1000	1000
LVL (dB)	0	0	0	0	0
BandW	1	1	1	1	1

Low Shelf CH2: Frequency 1000, Gain 0

LPF CH2: Frequency 800, Gain 0

HPF CH2: Frequency 800, Gain 0

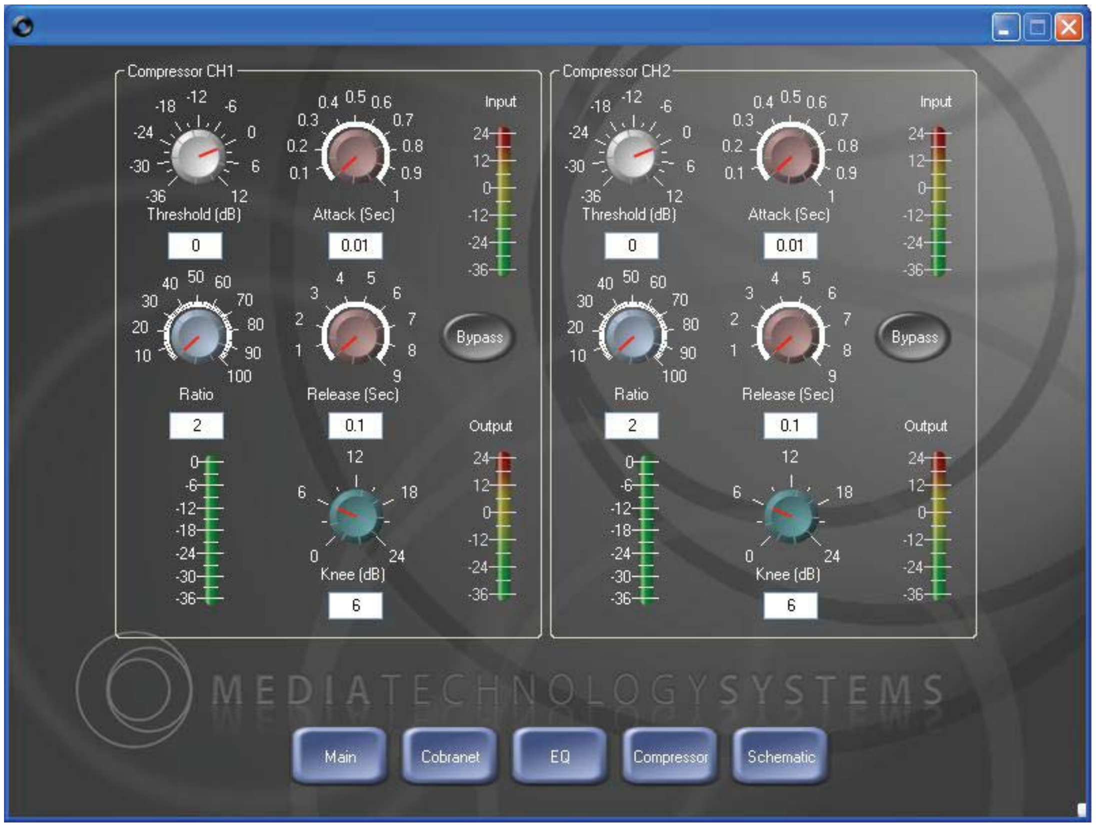
Figure 3-4: ION2.0 Equaliser settings

Pressing the “EQ” button will reveal the Equalizer settings page (see Figure 3-4 above).

The settings are reasonably straightforward… 5 bands of paramatric EQ, High Pass, Low Pass and Low shelf on each input channel. This provides basic input signal conditioning before the audio is put on to the network. The settings are…

- Parametric: Frequency (20-20kHz), Level (-18dB to +12dB) and Bandwidth (0.1-10).
- Low Shelf: Frequency (20-20kHz) and Level (-15dB to +12dB)
- Low Pass/High Pass: Frequency (20-20kHz), Level (-100dB to +0dB) and Bypass. Note: The HPF and LPF filters have a Butterworth characteristic and are set at 24dB/Oct rolloff.

Pressing the “Compressor” button will show the Compressor settings (see Figure 3-5 below).


Compressor CH1

Threshold (dB): 0
Attack (Sec): 0.01
Ratio: 2
Release (Sec): 0.1
Knee (dB): 6

Compressor CH2

Threshold (dB): 0
Attack (Sec): 0.01
Ratio: 2
Release (Sec): 0.1
Knee (dB): 6

Figure 3-5: Compressor settings

The Compressor uses an RMS detector and is primarily intended to provide dynamic range control “riding the gain” for microphone inputs. The settings will be very familiar to an audio engineer and are…

- Threshold: -36dB to +12dB
- Ratio: 1 to 100 in 0.1 steps
- Attack/Release: 10ms to 1 second/100mS to 9 seconds respectively
- Knee: 0db to +24dB

3.2 ION0.2

Launching ION02_rev1_1.exe will show the Main Screen, as shown in Figure 3-6 below.

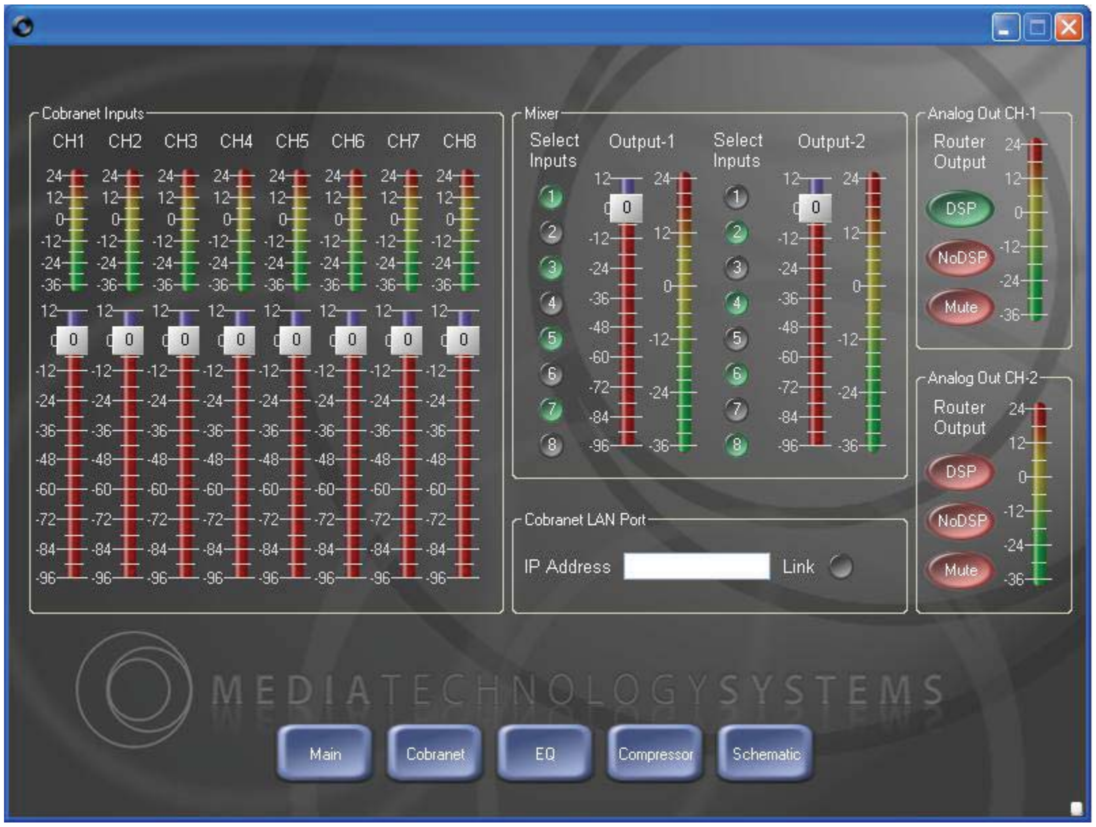


Figure 3-6: ION0.2 Main Screen

Again, the Main Screen is reasonably self-explanatory, but reference to the Schematic should explain the sequence of the controls (see Figure 3-7 below). The Schematic will open in a child window, so that it can be left on the screen.

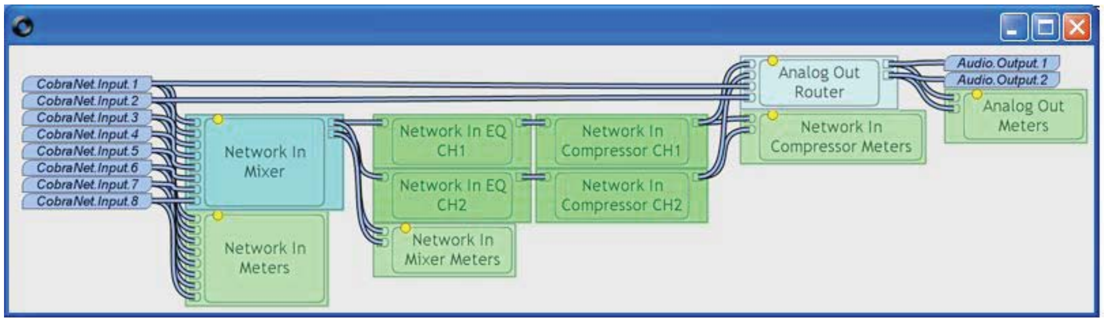


Figure 3-7 ION0.2 Schematic

The first function will be to set the IP address and if it has connected successfully to the ION, the Link indicator will turn green.

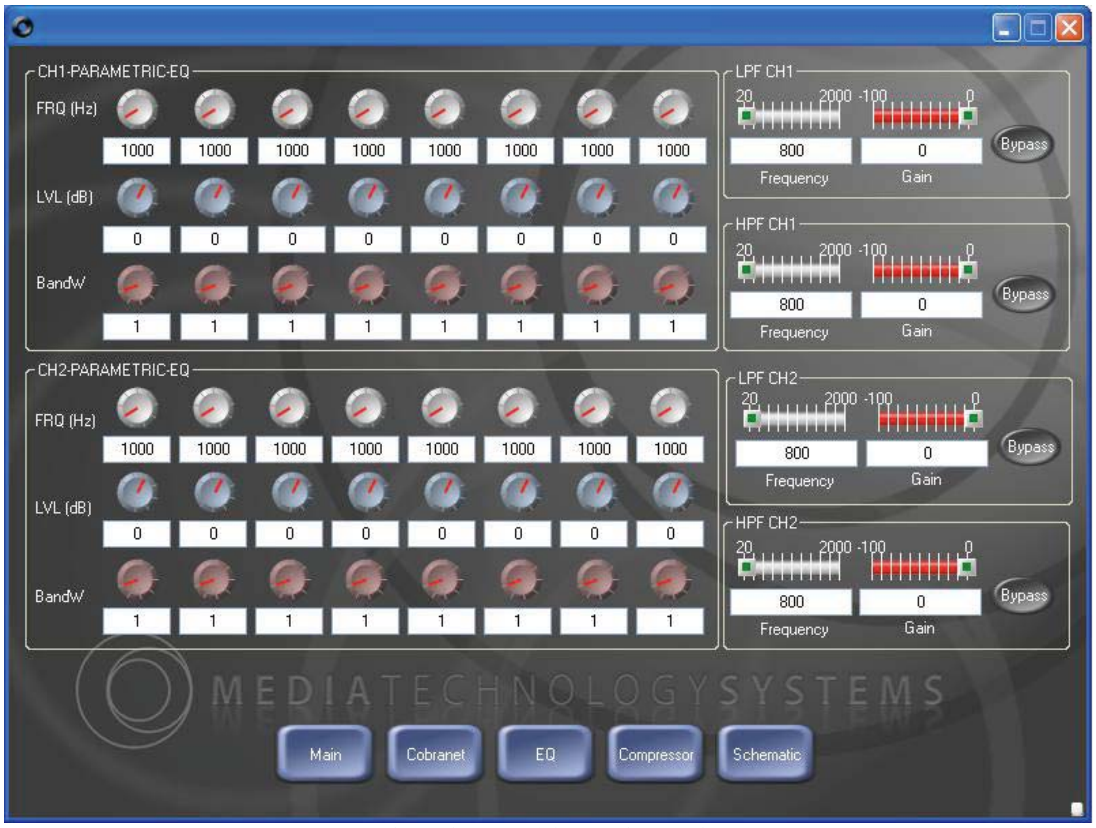
The Cobranet Inputs section shows the NetRx input sliders (prior to the Network In In Meters) and the “Network In Meters”. The Mixer section allows any of up to 8 input channels from any one or more of up to 8 Cobranet Receivers to be routed to each output, along with a slider gain control for each channel and the “Network In Mixer Meters”.

The Analog Out CH1 and CH2 routing refers to the 4x2 “Analog Out Router” in the schematic and selects between Cobranet input 1/2 direct (NoDSP), Cobranet Input 1/2 through the DSP side chain (DSP) or Router Channel Mute (Mute).

The ION0.2 shares the Cobranet and Compressor screens with the ION2.0 and these screens are covered in Figure 3-3 and Figure 3-5 above.

In the case of the ION0.2, it is RX1 and not TX1 that would be configured to receive Cobranet audio.

Pressing the “EQ” button will reveal the Equalizer settings page (see Figure 3-8 below).



CH1-PARAMETRIC-EQ

FRQ (Hz)	1000	1000	1000	1000	1000	1000	1000	1000
LVL (dB)	0	0	0	0	0	0	0	0
BandW	1	1	1	1	1	1	1	1

CH2-PARAMETRIC-EQ

FRQ (Hz)	1000	1000	1000	1000	1000	1000	1000	1000
LVL (dB)	0	0	0	0	0	0	0	0
BandW	1	1	1	1	1	1	1	1

	Frequency (20–2000)	Gain (-100–0)
LPF CH1	800	0
HPF CH1	800	0
LPF CH2	800	0
HPF CH2	800	0

Figure 3-8: ION0.2 Equaliser settings

The settings are reasonably straightforward… 8 bands of paramatric EQ, High Pass and Low Pass on each output channel. This provides basic input signal conditioning before the audio is put on to the network. The settings are…

- Parametric: Frequency (20-20kHz), Level (-18dB to +12dB) and Bandwidth (0.1-10).

- Low Pass/High Pass: Frequency (20-20kHz), Level (-100dB to +0dB) and Bypass. Note: The HPF and LPF filters have a Butterworth characteristic and are set at 24dB/Oct rolloff.

3.3 Using ION2.0 and ION0.2 together

It is possible to have 4 x ION2.0 (8 channels) connect to 1 x ION0.2 and have a simple 8x2 Network audio solution. The Cobranet Configuration is simple, as follows…

- Set TX1 of each of four ION2.0's to bundle address 256, 257, 258, 259 respectively
- Set txSubCount of each ION2.0 TX1 to 2
- Set ION0.2 RX1 to 256, RX2 to 257, RX3 to 258 and RX4 to 259
- Set ION0.2 sub channel mapping as follows…
  - RX1: 33, 34
  - RX2: 35, 36
  - RX3: 37, 38
  - RX4: 39, 40

Eight channels of audio should now be available at the input to the Main Screen and the mix can be set up accordingly.

4 Control & Monitoring software - Programming

MTS provides a Control and Monitoring application called "MTS Control", which is a stripped down version of Stardraw Control 2010.

In some installations, there will be 3rd party control systems (such as AMX and Crestron) and these will be used to provide control and monitoring functionality for the MTS products. MTS Control is intended for installations where there is no 3rd party control system.

The full version of Stardraw Control is a licensed software-based universal control platform designed to create custom User Interfaces that can control any remotely-controlled or monitored hardware. See website below for full details.

http://www.stardraw.com/products/stardrawcontrol/

The MTS Control version is provided by MTS at no cost, but is subject to the same copyright and intellectual property rights as the main Stardraw Control. The primary difference is that MTS Control will ONLY control and monitor MTS products. If the user would like to expand the capability of the application, then a full license can be purchased from Stardraw Control.

The MTS Control software is included in the CDROM, along with this manual

MTS Control is a ~20 Meg download and requires .net3.5. If .net3.5 is not present on the host computer, then it will attempt to download from the internet and install it automatically.

The full download of MTS2010 is a 256Mb as it includes .net 3.5 and SQL Server CE and will install and run as a native 64 bit application as well as 32bit.

Once the application has been downloaded and executed, the MTS Control icon should appear on the desktop – see Figure 4-1 below.

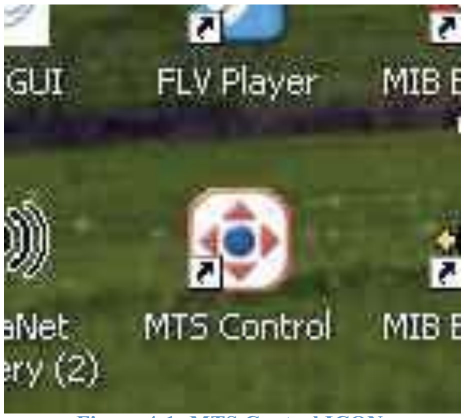
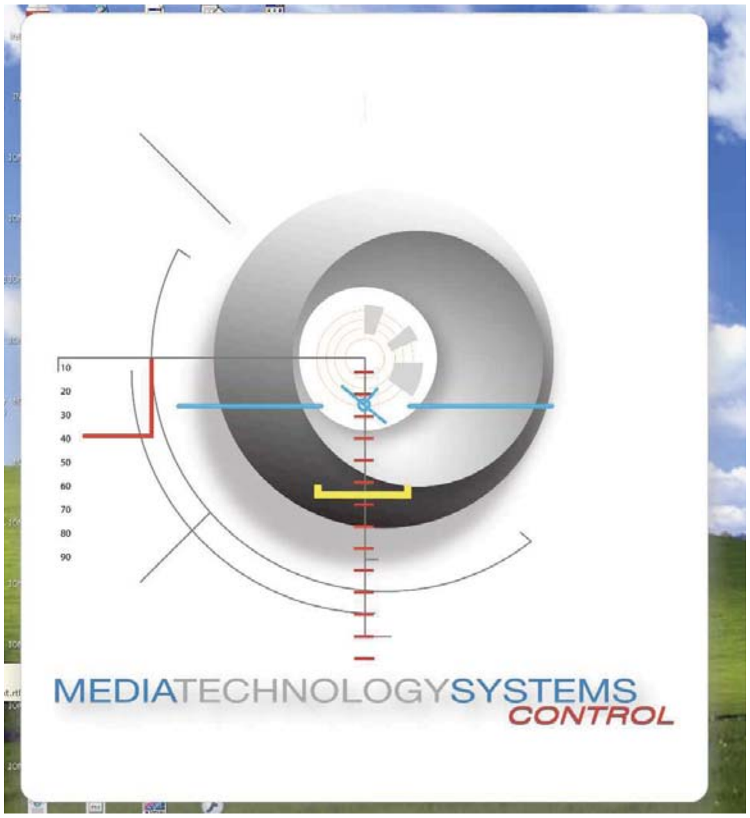


Figure 4-1: MTS Control ICON

Launching MTS Control will show the following splash screen and then the initial opening menu – see Figure 4-2



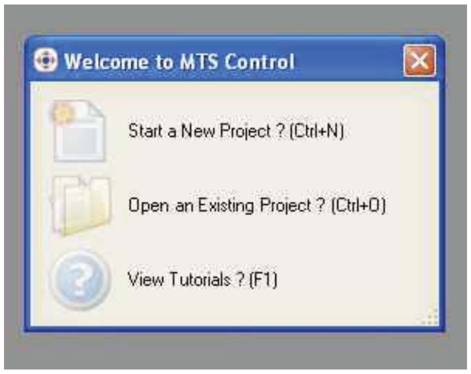
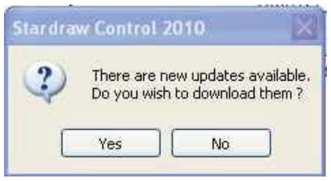

Welcome to MTS Control
Start a New Project ? (Ctrl+N)
Open an Existing Project ? (Ctrl+O)
View Tutorials ? (F1)

Figure 4-2: Opening menu

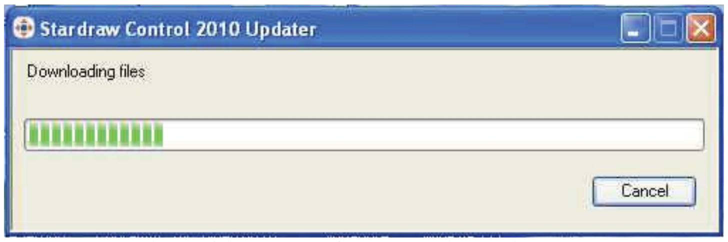
4.1 MTS Control Updates

At this point a new installation should connect to the internet and update the MTS Control software. There are two types of update… authoring tool update (see Figure 4-3 below) and devices update (see Figure 4-4 below). The update process may include one or both. Also, Windows should always be checked for critical updates. MTS Control uses Windows .net3.5 and may have inconsistencies if Windows is not current.



Stardraw Control 2010

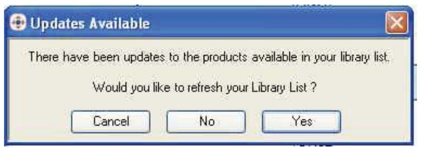
There are new updates available.
Do you wish to download them ?



Stardraw Control 2010 Updater

Downloading files

Figure 4-3: Request for update-Authoring tool



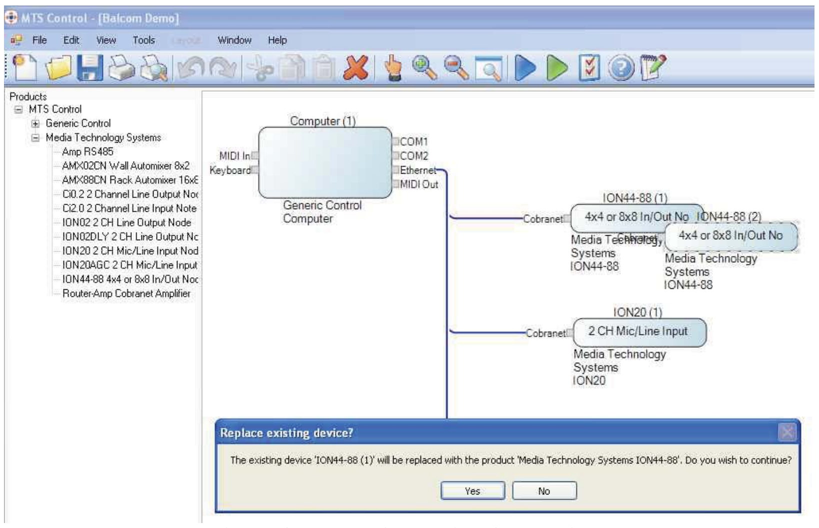
Updates Available

There have been updates to the products available in your library list.

Would you like to refresh your Library List ?

Figure 4-4: Request for update-Device Library

NOTE: If devices have been updated, then the devices in a project (or the provided sample s03 files) may also need updating. This process is simple: open the project and drag a device to an existing device in topology view and drop it on top. The existing device will highlight in green and a request to replace menu will appear (see Figure 4-5 below). Each device may need to be updated.



Replace existing device?

The existing device 'ION44-88 (1)' will be replaced with the product 'Media Technology Systems ION44-88'. Do you wish to continue?

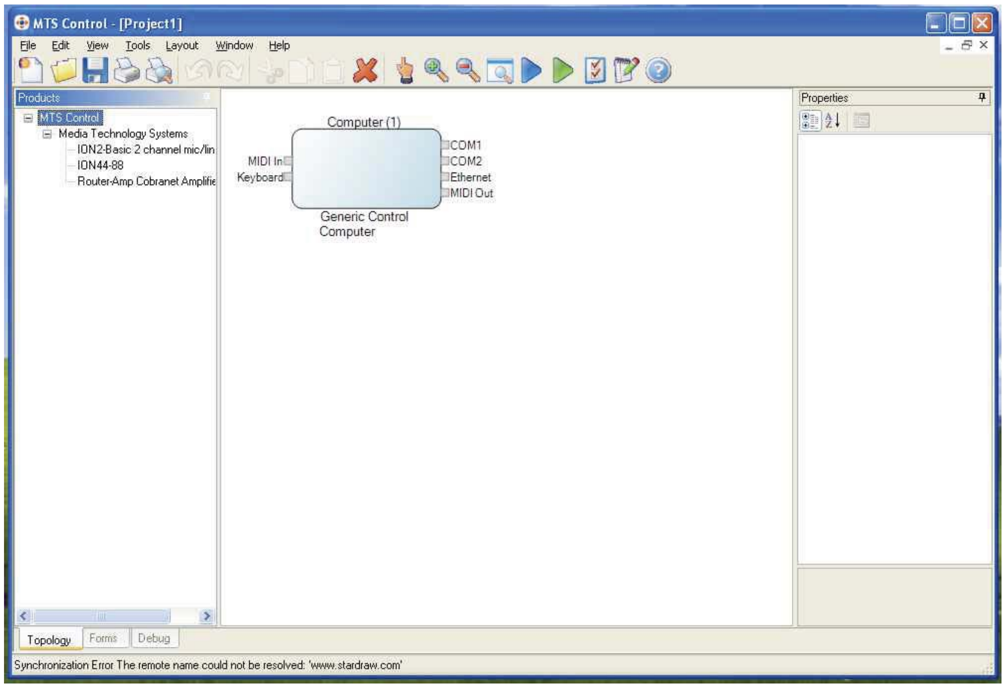
Figure 4-5: Updating devices in a project

4.2 MTS Control - Getting Started

This section is intended to get non-programmers started on an MTS Control Application. It is also intended to cover the most frequently asked questions, as well as some hints and tips.

However, a new user should choose "View Tutorials", as it is beyond the scope of this manual to provide in depth training on either Stardraw Control or C# (the language underpinning Stardraw Control). Some basic movies in /swf format have been included on the CDROM. In addition, the CDROM contains some sample demo files (*.so3) that can be imported into MTS Control as well as the compiled executable versions of the samples. The sample executables can be run on any windows compliant machine (WinXP and above) running .net3.51 and above.

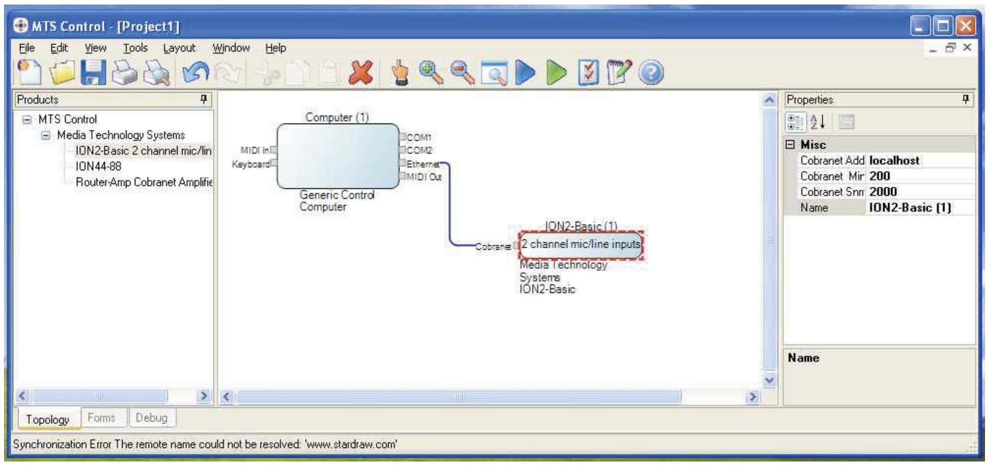
A new project will show the opening Topology View screen given in Figure 4-6 below. Topology View shows the default Computer and allows the user to select the devices in the system.



Synchronization Error The remote name could not be resolved: 'www.stardraw.com'

Figure 4-6: MTS Control – Main Screen

The devices in MTS Control are the Serial and Network Amplifiers, as well as the AMX and ION interfaces. This manual covers the ION2.0 and ION0.2, but the techniques are equally applicable to all devices. Adding a device is simple, drag the ION2.0/ION0.2 device icon across to the main panel and then drag a wire from the Ethernet port of the Computer to the Cobranet port of the ION2.0/ION0.2 – see Figure 4-7 below.



Cobranet Add	localhost
Cobranet Mir	200
Cobranet Snr	2000
Name	ION2-Basic (1)

Synchronization Error The remote name could not be resolved: 'www.stardraw.com'

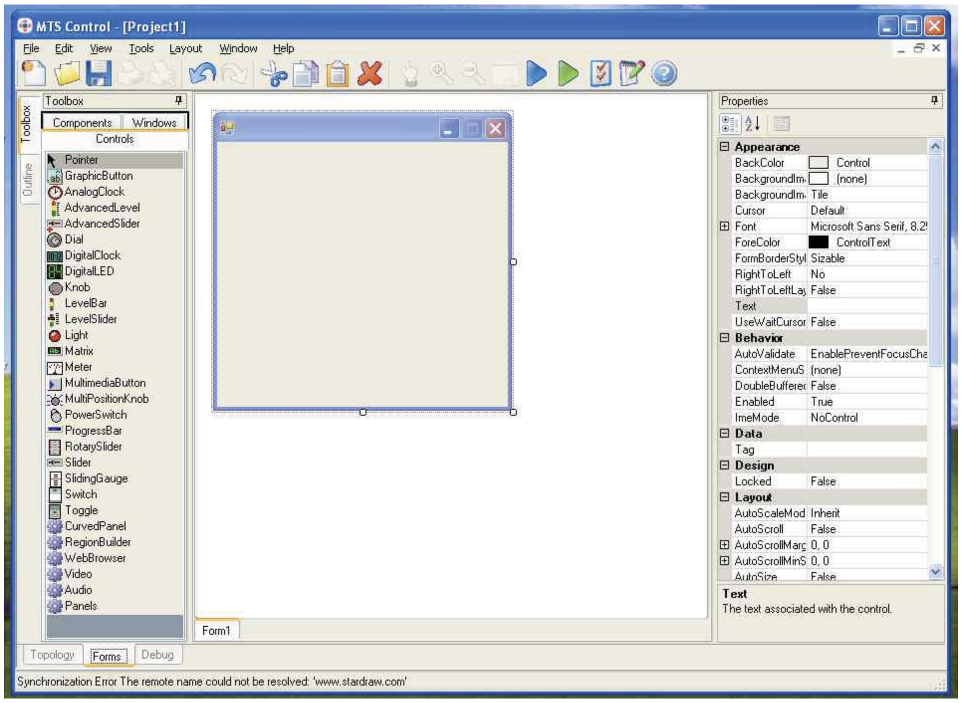
Figure 4-7: Adding a device

As many devices as needed can be added to the panel and connected to the Ethernet port of the Computer.

Note: The properties menu at the RHS. If a static IP address is to be used for the device, then replace “localhost” with the usual 192.168.1.100 type of IP address. This IP address will be fixed, as it will be compiled with the executable. If the IP address needs to change (say one UI is addressing many devices), then leave the address as localhost and set the IP address in the UI.

4.2.1 Forms View

Once the devices are chosen, the User selects the Forms View (see Figure 4-8 below) and can start programming the Graphical User Interface.



Appearance	
BackColor	Control
BackgroundIm	(none)
BackgroundIm	Tile
Cursor	Default
Font	Microsoft Sans Serif, 8.2
ForeColor	ControlText
FormBorderStyl	Sizable
RightToLeft	No
RightToLeftLay	False
Text	
UseWaitCursor	False
Behavior	
AutoValidate	EnablePreventFocusCha
ContextMenuS	(none)
DoubleBufferec	False
Enabled	True
ImeMode	NoControl
Data	
Tag	
Design	
Locked	False
Layout	
AutoScaleMod	Inherit
AutoScroll	False
AutoScrollMarc	0, 0
AutoScrollMinS	0, 0
AutoSize	False

Text
The text associated with the control.

Synchronization Error The remote name could not be resolved: 'www.stardraw.com'

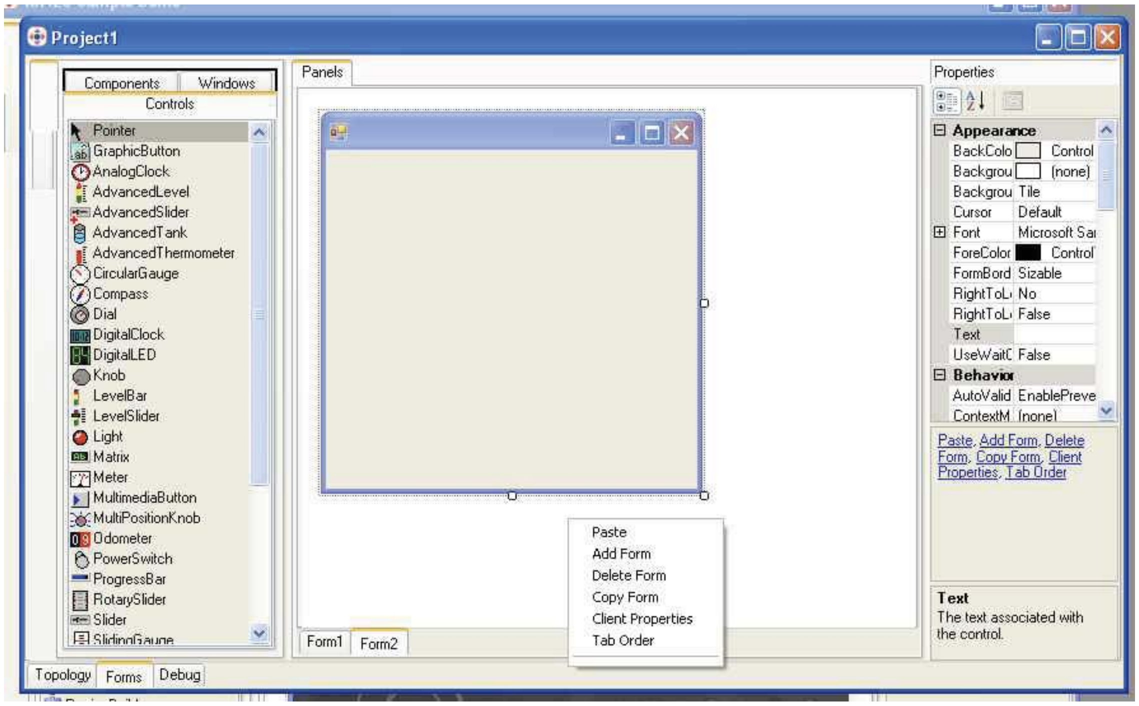
Figure 4-8: Forms View

The Forms View is where the UI is configured. A "Form" can be considered as a 'page' in a typical AV control system touch panel design. There can be a main Form and many sub-Forms (usually called from the main Form) for a complex UI, or a simple UI may only need one main Form.

To add more forms, simply right click in the forms view and elect "Add Form". A new blank form will be added and a "Form2" tab will appear in the lower left of the Form view (see Figure 4-9 below). Clicking on these tabs will allow the user to swap between forms. The Forms can also be re-named in the properties dialog box (Misc/Name: "Form1").

At the left hand side of the Forms view is the Controls menu. The user will usually drag a control (or windows component) from the Menu into the Form. At the right hand side of the Forms view is the properties window. This is where the various settings for the form (or controls) are adjusted to suit the application.

At the top of the Forms view is the control bar. Apart from the usual editing and file control functions, there are two triangles (play buttons), one blue and one green (see Figure 4-10 and Figure 4-11 below). The Blue triangle (or 'play' button) allows the user to run the project from the Forms Window. This is a testing and debugging tool. Once the project is finished, then the user presses the green triangle to create a windows executable. The Windows executable (*.exe) can run on any .net3.51 (or above) compatible PC.



Appearance	
BackColo	Control
Backgrou	(none)
Backgrou	Tile
Cursor	Default
Font	Microsoft Sa
ForeColor	Control
FormBord	Sizable
RightToL	No
RightToL	False
Text	
UseWaitC	False
Behavior	
AutoValid	EnablePreve
ContextM	(none)

Text
The text associated with the control.

Figure 4-9: Adding a new Form

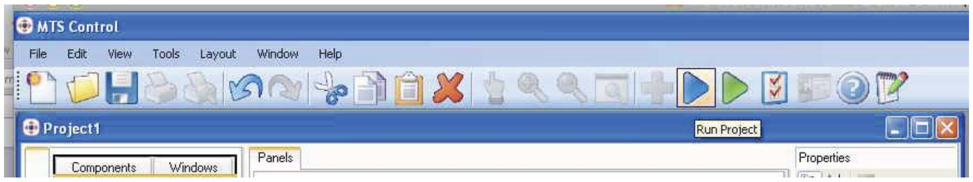


Figure 4-10: Run Project, ie real time testing and debugging

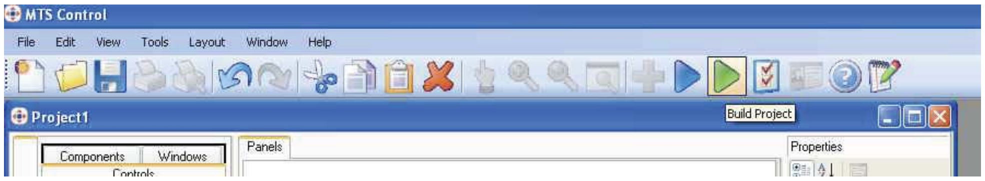
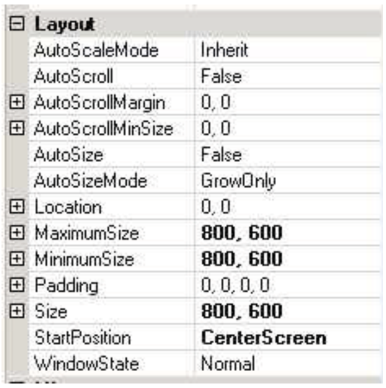


Figure 4-11: Build Project, ie create a Windows Executable

4.2.1.1 Forms: Key properties.

The first decision is size. Usually the control surface is a PC, often a Tablet PC or ultra mobile PC (UMPC). In this situation, the main Form will usually be sized to fill the screen.

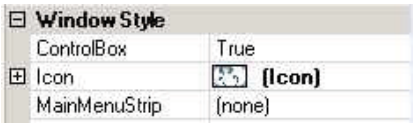
The Form size property (see Figure 4-12 below) can set the start size, start position, as well as the maximum and minimum sizes the operator can set the screen. In the case of the main Form, the size is usually set to the size of the screen (800x600 in the example below). In addition, the maximum and minimum sizes have been fixed, so the user cannot make changes. For sub-Forms, the same 'full screen size' policy can be used, or they can be made smaller, so that multiple sub forms can be open and visible at the same time.



Layout	
AutoScaleMode	Inherit
AutoScroll	False
AutoScrollMargin	0, 0
AutoScrollMinSize	0, 0
AutoSize	False
AutoSizeMode	GrowOnly
Location	0, 0
MaximumSize	**800, 600**
MinimumSize	**800, 600**
Padding	0, 0, 0, 0
Size	**800, 600**
StartPosition	**CenterScreen**
WindowState	Normal

Figure 4-12

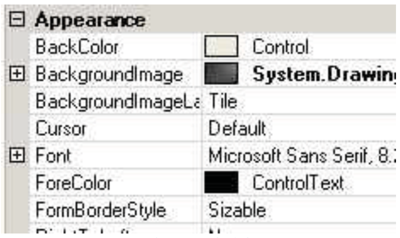
The next property is the Form icon. Windows allows an icon to be loaded at the top left of the Form. The icon can be changed from the default to any icon image in the users HDD (see Figure 4-13 below).



Window Style	
ControlBox	True
Icon	**(Icon)**
MainMenuStrip	(none)

Figure 4-13

Another important graphical property is the background image of the Form. This is set under appearance (see Figure 4-14 below) and can be any standard format of graphical file (bmp, jpg, png, etc) on the users HDD. Note: it is best to size the image to suit the size of the form before adopting as the background image. Also note that if the Form has the ControlBox enabled (see below), then the background image size will need to be smaller than the screen size to accommodate the ControlBox



⊟ **Appearance**	
BackColor	□ Control
⊞ BackgroundImage	■ **System.Drawin**
BackgroundImageLa	Tile
Cursor	Default
⊞ Font	Microsoft Sans Serif, 8.
ForeColor	■ ControlText
FormBorderStyle	Sizable

Figure 4-14

A useful feature for the main Form is to disable the Minimize, Maximize and ControlBox properties. This can be done by setting the ControlBox property of the Form to False. This will also remove the Minimize and Maximize buttons as well (although these can be done individually). If a title bar is still needed (so that you can move the form), then specify the Text property for the Form. This will result in a form with a blank control box (see Figure 4-15 below). Note: it is often better to have no ControlBox in the main Form and a blank ControlBox in the sub-Forms. This will allow the main From to be the full size graphical image and the sub-Forms to be moved (assuming they are smaller then the main Form).

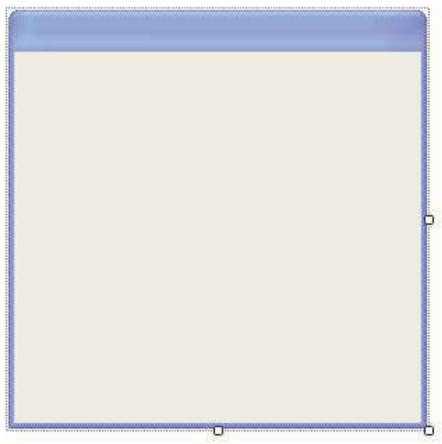


Figure 4-15

4.2.1.2 Forms: Controls

Once the Form and sub-Form properties have been configured, it is time to add some controls. As a simple start, a windows button will be dragged to the main and sub Forms (see Figure 4-16 below)

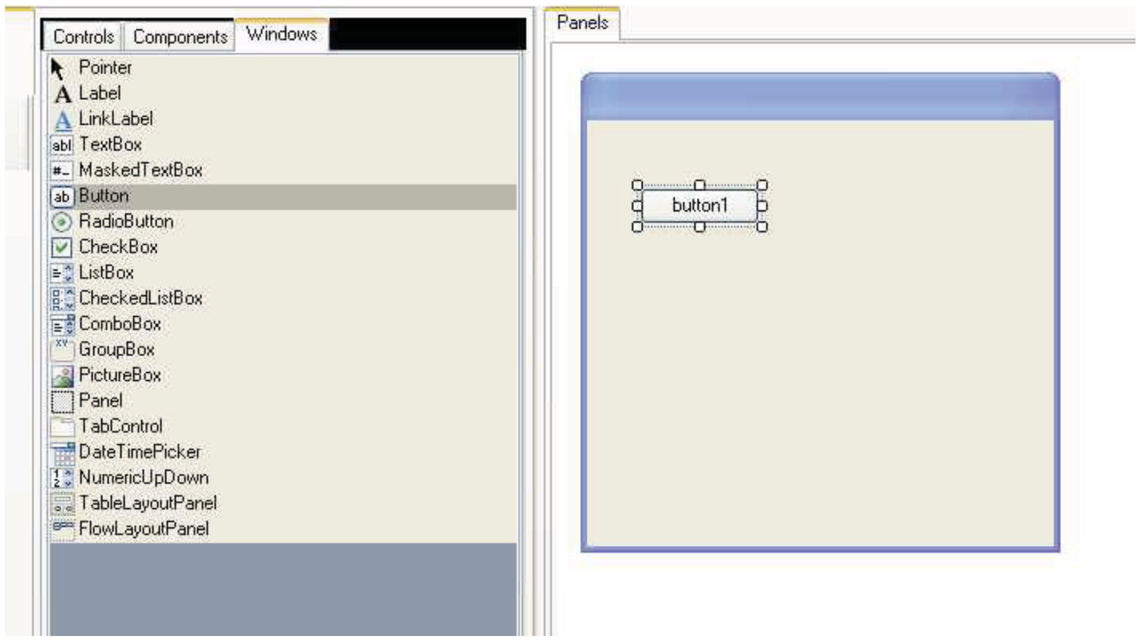


Figure 4-16

When the button is highlighted, the properties menu at the right will now show the properties for the button. The button can be named, sized, colored, have text added/deleted, etc. The Forms view properties menu will always refer to the item that is highlighted.

4.2.1.3 Forms: Actions & Events

Once the graphical properties for the control has been set, it is time to make the control do something. This is configured through the Events/Actions dialog box.

This is the most important concept in MTS Control. All the controls and graphics loaded into the Form UI will have no effect until the UI control (or indicator) is associated with an "Event" and then uses that Event as a trigger for one or more "Actions".

For example, the simple button in Figure 4-16 above, can be programmed to open the sub-Form. First, the button is double clicked to open the Events –Actions dialog box (see Figure 4-17 below).

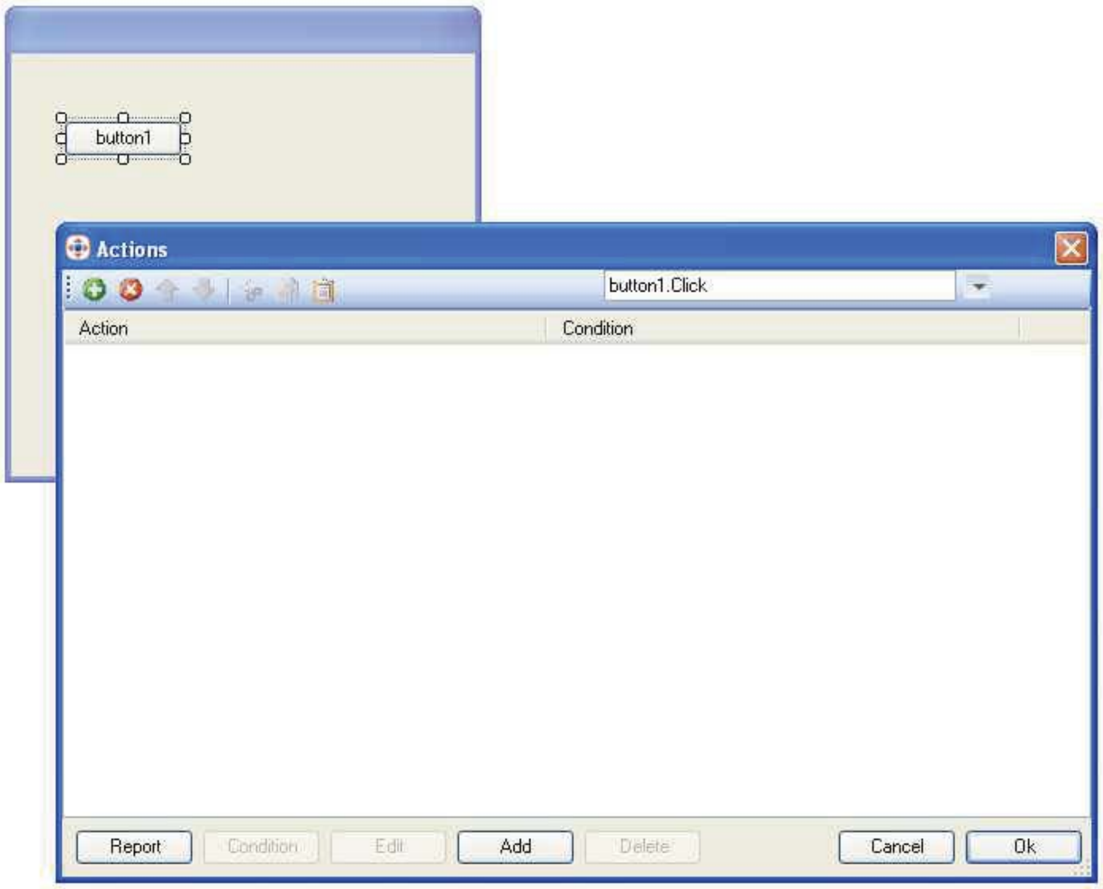


Figure 4-17

The default event can be seen in the dropdownbox at the top right of the dialog box, ie "button1.click". If a different event is required, then the dropdownbox can be used to select any available event, either for the button, or any other device or control in the Form.

As button1.click is the required event to change forms on clicking the button, no change is required.

The next step is to associate one or more actions with that event. This is done by clicking on the green "+" button in the top left corner. A dialog box will allow the user to select an action from those available. In this example "Forms", then "Form2", then "Show" should be selected (see Figure 4-18 below).

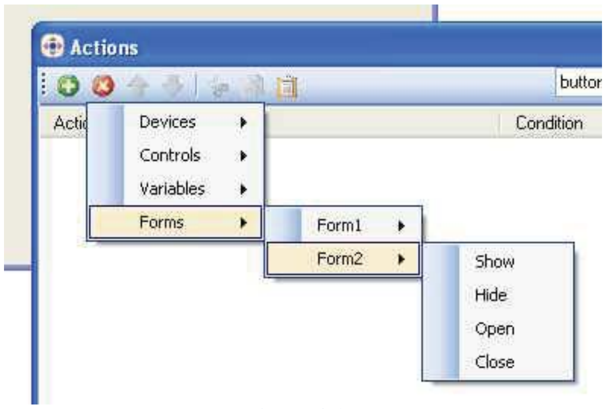
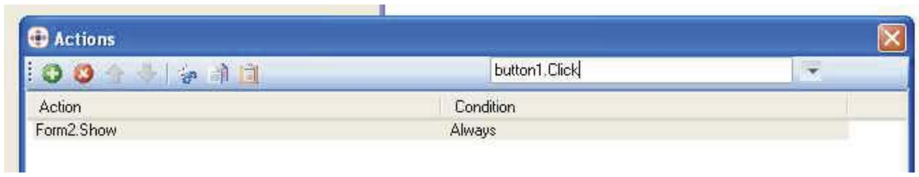


Figure 4-18

Once the action has been accepted, then the action will appear in the list associated with that event (see Figure 4-19 below).

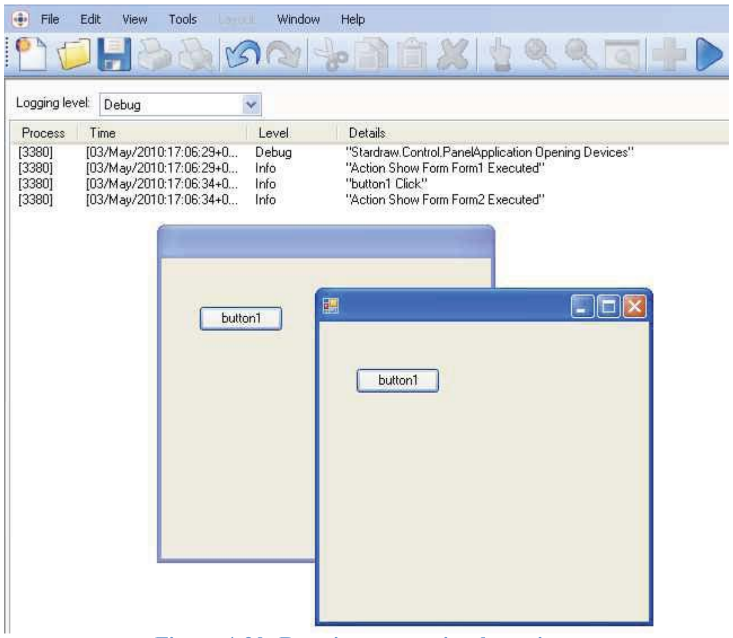


button1.Click

Action	Condition
Form2.Show	Always

Figure 4-19

Running the Project (blue triangle) will start the debug window and pressing button 1 will open the sub-Form "Form2" (see Figure 4-20 below).



Logging level: Debug

```
Process  Time                      Level   Details
[3380]   [03/May/2010:17:06:29+0...  Debug   "Stardraw.Control.PanelApplication Opening Devices"
[3380]   [03/May/2010:17:06:29+0...  Info    "Action Show Form Form1 Executed"
[3380]   [03/May/2010:17:06:34+0...  Info    "button1 Click"
[3380]   [03/May/2010:17:06:34+0...  Info    "Action Show Form Form2 Executed"
```

Figure 4-20: Running a very simple project

The most common examples of events and actions are:

- Using a control: In this situation, the "Event" will be the UI control changing value as the user adjusts the control. The "Action" will be to apply the change in

UI control value to a parameter in the target device. Figure 4-21 below shows a slider dragged into the Form. Double clicking on the slider will open the Events/Actions dialog box.

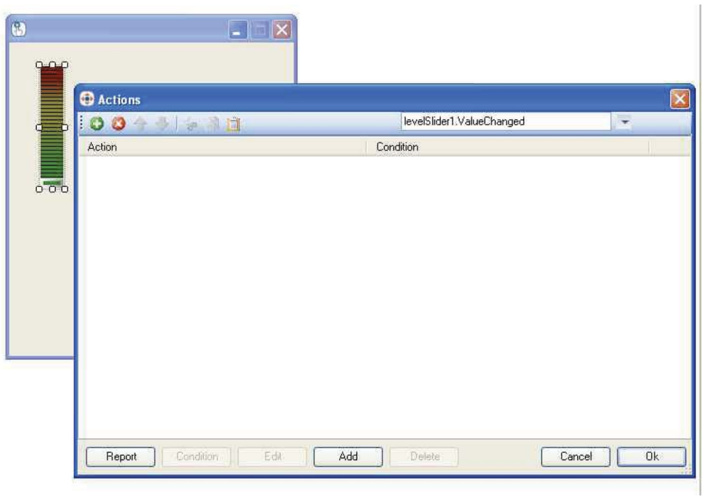


Figure 4-21

The default Event shown is “levelSlider1.valueChanged”. The ION2.0 has a number of parameters that can be controlled. In this example, the desired control is input channel 1 gain. The action will be set by clicking on the green “+” in the top left corner, selecting “Device”, then scrolling to “ION2.0 AGC” will show a list of all the available elements in the device - see Figure 4-22 below.

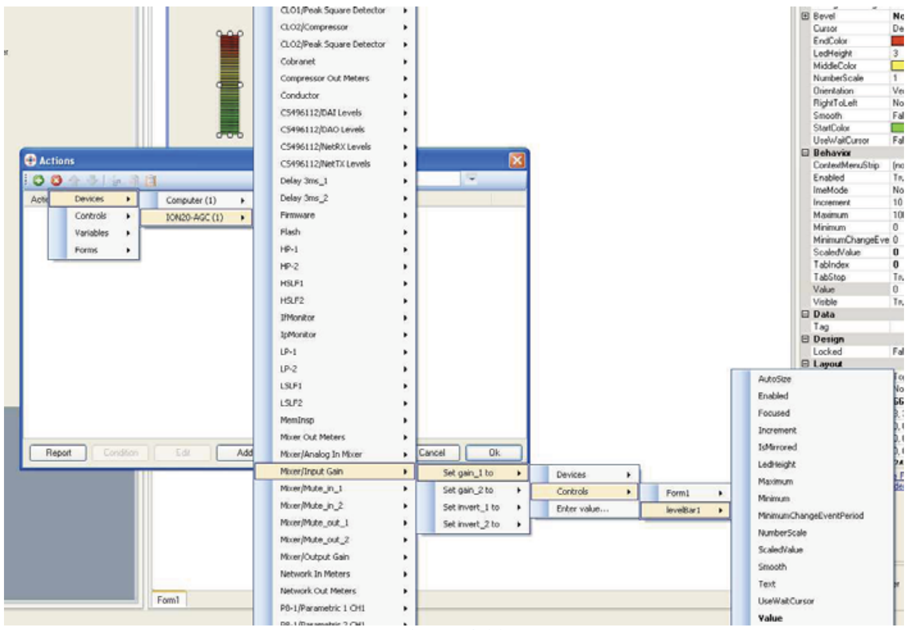


Figure 4-22

From the list of elements, “Mixer/Input gain” is chosen and “gain_1” is the desired control. The Action to be chosen is “Set “gain_1 to” and then we associate the Action with the UI controller, ie “Controls”, “levelSlider1” and finally “value”.

In the above example, every time a change is made to "levelSlider1" in the UI, the "levelSlider1.valueChanged" event will trigger and the new value of levelSlider1 will be sent to the device parameter "gain_1" in the Mixer/Input gain element in the ION2.0 device.

- Setting an indicator: In this situation, the Event and Action is reversed compared to a control. Instead of setting a device parameter value to the UI control value, the need is to set the UI indicator value to a device parameter value. Figure 4-23 shows an advanced level indicator dragged into the form and the default Event is "advancedLevel1.ValueChanged".

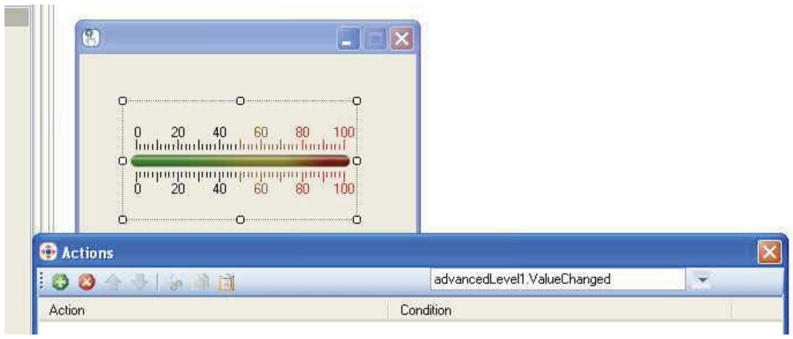


Figure 4-23: Default Event is incorrect

However, the desired Event is the ION2.0 input channel 1 level, so the Event is changed to "Device/ION2.0", element "Analog In meters" and parameter "peak_1Changed", as seen in Figure 4-24 & Figure 4-25 below.

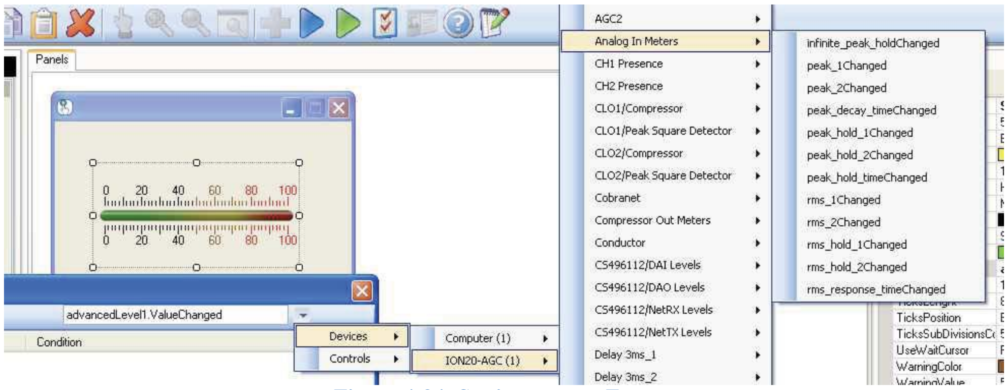


Figure 4-24: Setting correct Event.

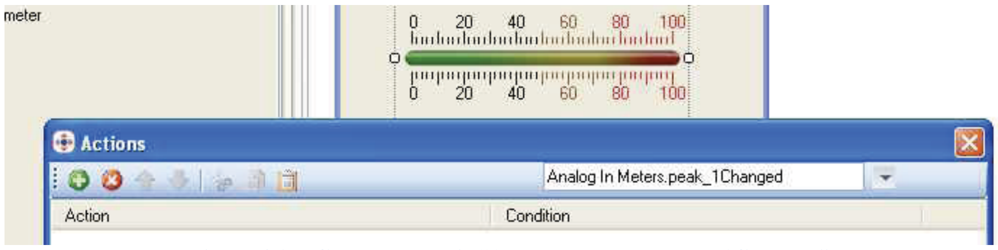


Figure 4-25: Correct Event is "Analog In Meters peak_1Changed"

Now that the correct Event has been selected, the UI level meter value needs to be associated with the Device meter value, as shown in Actions dialog box in

Figure 4-26 below. The UI “advancedLevel1” value will be set to the device parameter Analog In/Meters “peak_1” value, whenever the Event (“peak_1” value changes) is triggered.

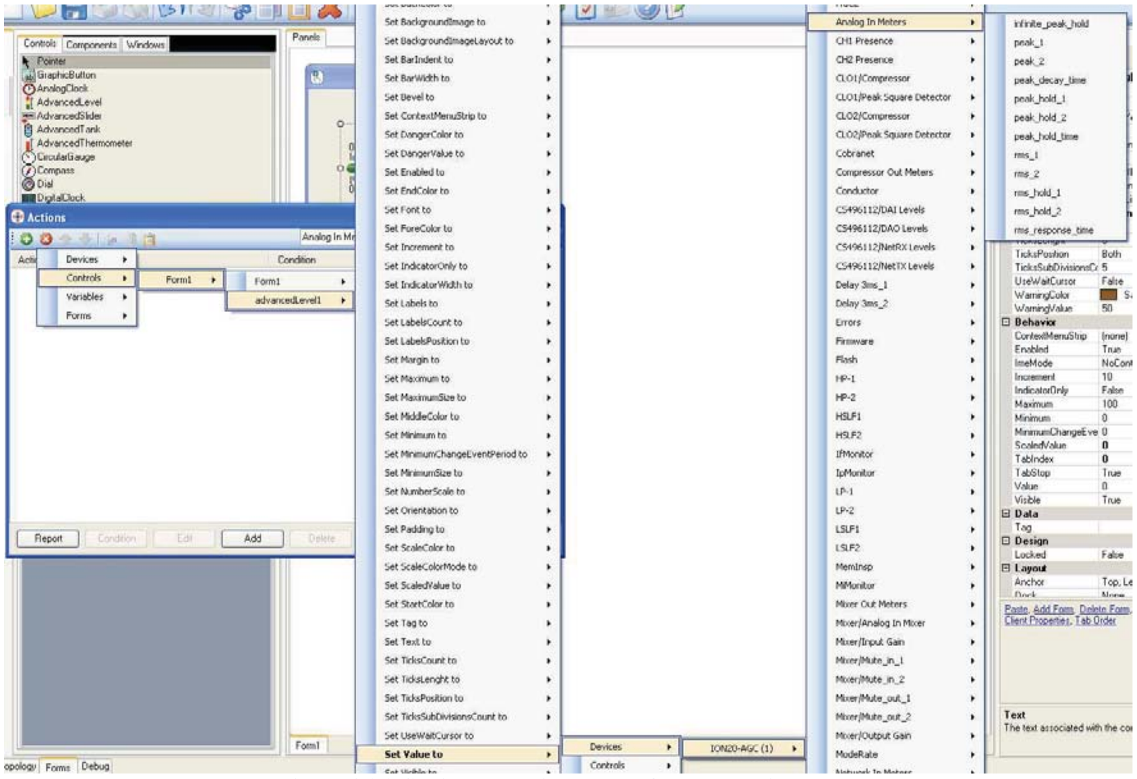


Figure 4-26: Setting UI Indicator to reflect the Device value

4.2.1.4 Formload

There is one other important element associated with a Form and that is the “Formload”. When a form opens it is often necessary to import a set of default values, or to read the current settings in the device under control. This is done through the Formload function.

The Formload is simply an event that is triggered when a Form is opened at runtime (ie when the Form or sub-Form is opened when running the executable). Double clicking the ControlBox (the bar-usually blue-at the top of the Form). This will open an event dialog box as shown in Figure 4-27 below.

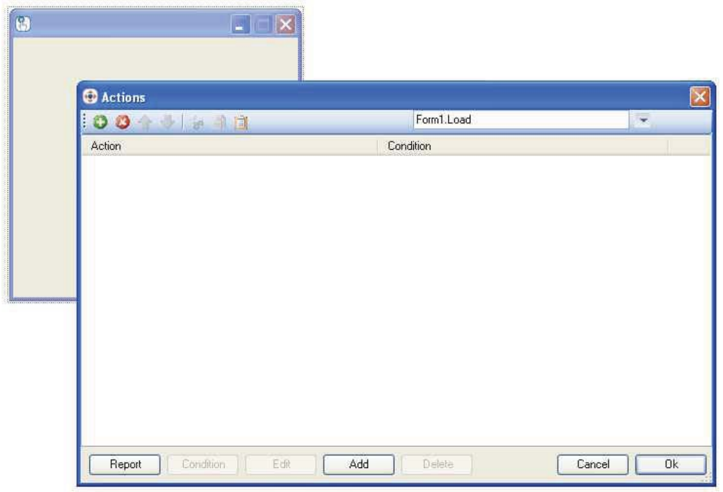
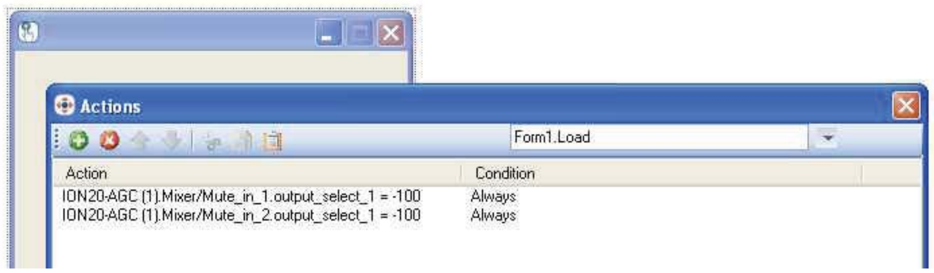


Figure 4-27: Formload

The event dialog box opens with form.load already setup as the event. Actions can now be set to happen as part of that Formload event. For example, Figure 4-28 below shows 2 mute actions. When the Form is opened, the input channels will be muted, ensuring that loud audio is not accidently sent to the loudspeakers on startup.



Action	Condition
ION20-AGC (1).Mixer/Mute_in_1.output_select_1 = -100	Always
ION20-AGC (1).Mixer/Mute_in_2.output_select_1 = -100	Always

Figure 4-28

NOTE: Please remember that MTS Control applies Actions to Events. An Event is usually a parameter in the device changing value, which is why Events are usually shown as “something.changed”, when scrolling through the Events menu. This places a limitation on uploading initial values during the mainform.load, as many parameters have yet to change, so no Event has been fired and thus no current value has been read by the device driver in MTS Control.

This problem is overcome by forcing an Event to fire during the mainform.load. The simplest method is to use mainform.load to read the initial value in the device driver into the UI.

When running an executable for the first time, the following will happen…

- The device driver values are cleared and are initially set to a default which varies for each property, either zero, or the default value assigned by Cirrus Logic to the DSP settings

- The mainform.load will have an action to set every parameter to the default value. This will force the device driver to fire an Event, if the value of the parameter is different to the default.
- The normal action of

4.2.1.5 Forms vs Panels

Forms are an excellent way of creating touch panel UI pages in large control systems. However, they have 1 primary advantage and one primary disadvantage…

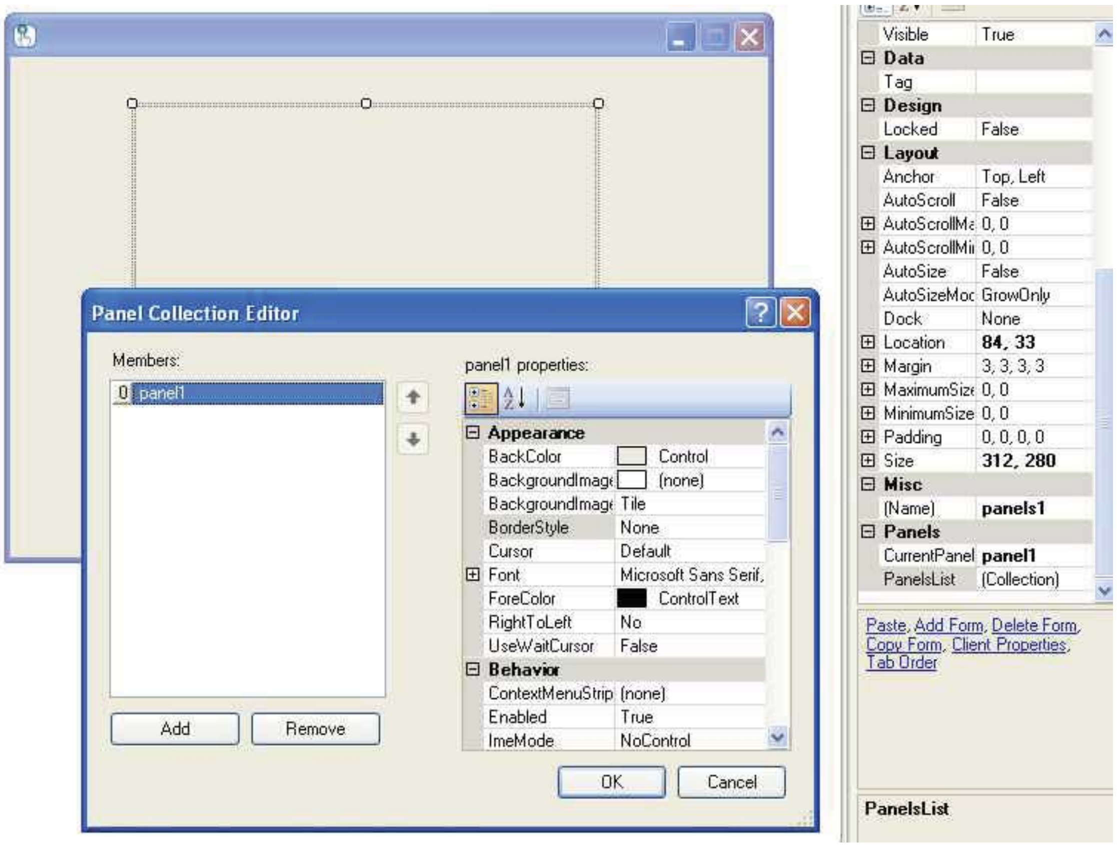
- The primary advantage is that sub-Forms are independent child windows with all the control features of child windows. This will allow many child windows to be open at the same time, be resizeable and pushed around a screen and not one line of code was written. This apparently simple feature is hard to achieve in conventional AV touch panel solutions without some experience in programming.

  The benefit of sub-Forms is that large touch screens can be used to create complex control surfaces

- The primary disadvantage of sub-Forms is that constantly opening and closing sub-Forms requires the tracking of all parameters used between sub-Forms. This will require a sub-Form to load in current values when it is opened (see Formload above).

  The solution is reasonably straightforward-instead of setting values directly to device parameters, values are set to "Variables" and the variables used in turn to control device parameters. In this way, it does not matter which sub-Form was used to control a parameter or how many times it was opened or closed.

However, using variables adds a second layer of actions and is an overkill for simple UI's. In this situation, it is better to use a small number of Forms and use the "Panels" control. The panels control is simply dragged into the form and square outline will be seen on the form (see Figure 4-29 below).



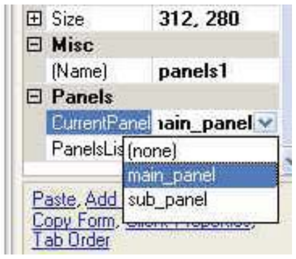
Appearance	
BackColor	Control
BackgroundImage	(none)
BackgroundImage	Tile
BorderStyle	None
Cursor	Default
Font	Microsoft Sans Serif,
ForeColor	ControlText
RightToLeft	No
UseWaitCursor	False
Behavior	
ContextMenuStrip	(none)
Enabled	True
ImeMode	NoControl

Visible	True
Data	
Tag	
Design	
Locked	False
Layout	
Anchor	Top, Left
AutoScroll	False
AutoScrollMa	0, 0
AutoScrollMi	0, 0
AutoSize	False
AutoSizeMoc	GrowOnly
Dock	None
Location	**84, 33**
Margin	3, 3, 3, 3
MaximumSize	0, 0
MinimumSize	0, 0
Padding	0, 0, 0, 0
Size	**312, 280**
Misc	
(Name)	**panels1**
Panels	
CurrentPanel	**panel1**
PanelsList	(Collection)

Figure 4-29: Panels properties

Clicking on the panels "collection" property will open a dialog box (see Figure 4-29). Panels can now be added and named, as well as having size and graphical properties set for each panel. Panels should be considered as a simple version of sub-Forms. The main Form page can have a number of panels and a series of buttons to call those panels-the same principle as sub-Forms without the need to track parameters/values or configure child windows.

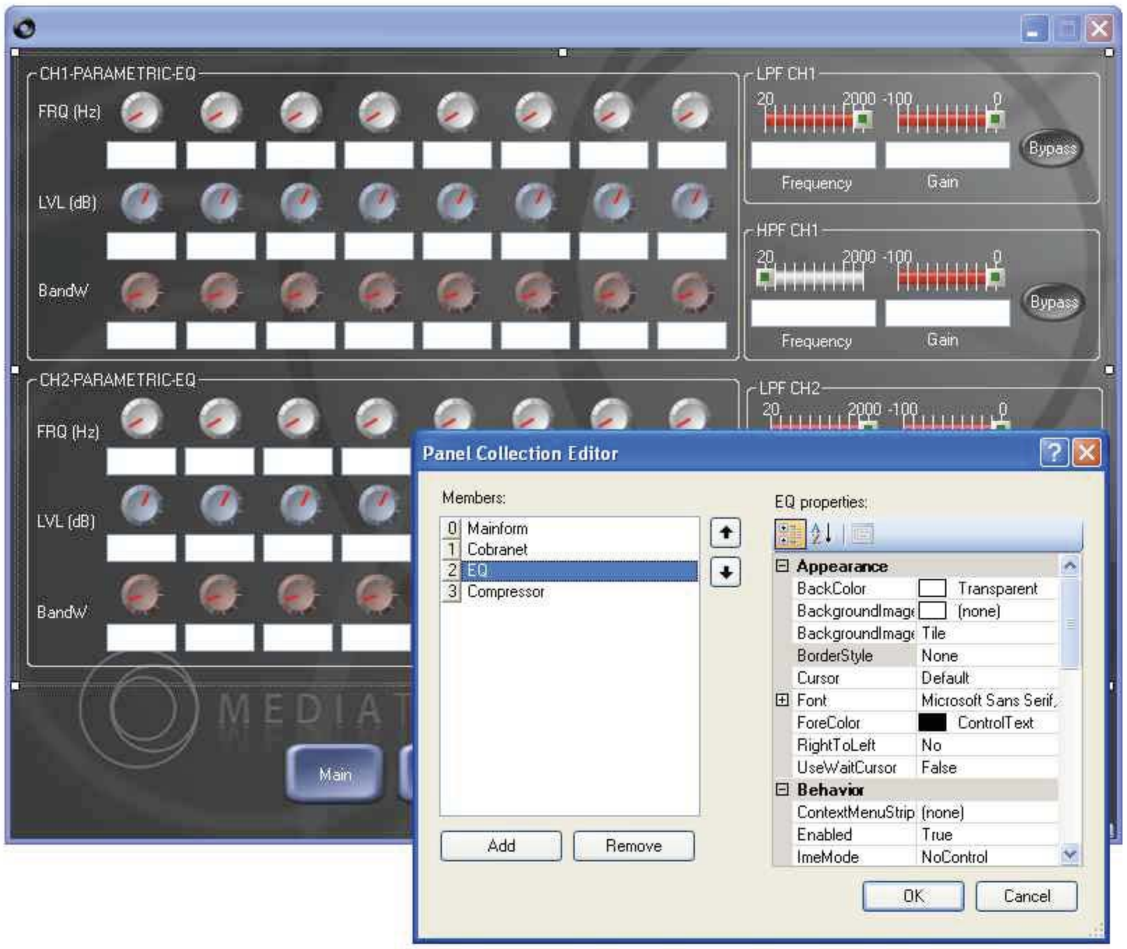
The various panels are selected in the "CurrentPanel" section in the panels properties (see Figure 4-30 below).



Size	**312, 280**
Misc	
(Name)	**panels1**
Panels	
CurrentPanel	**main_panel**
PanelsList	

Figure 4-30

Controls and indicators can be dropped into different panels in exactly the same way as sub-Forms. The ION2.0 and ION0.2 sample demo so3 files use panels to switch between the controls for EQ, Cobranet, Compressor, etc. See Figure 4-31 and Figure 4-32 below

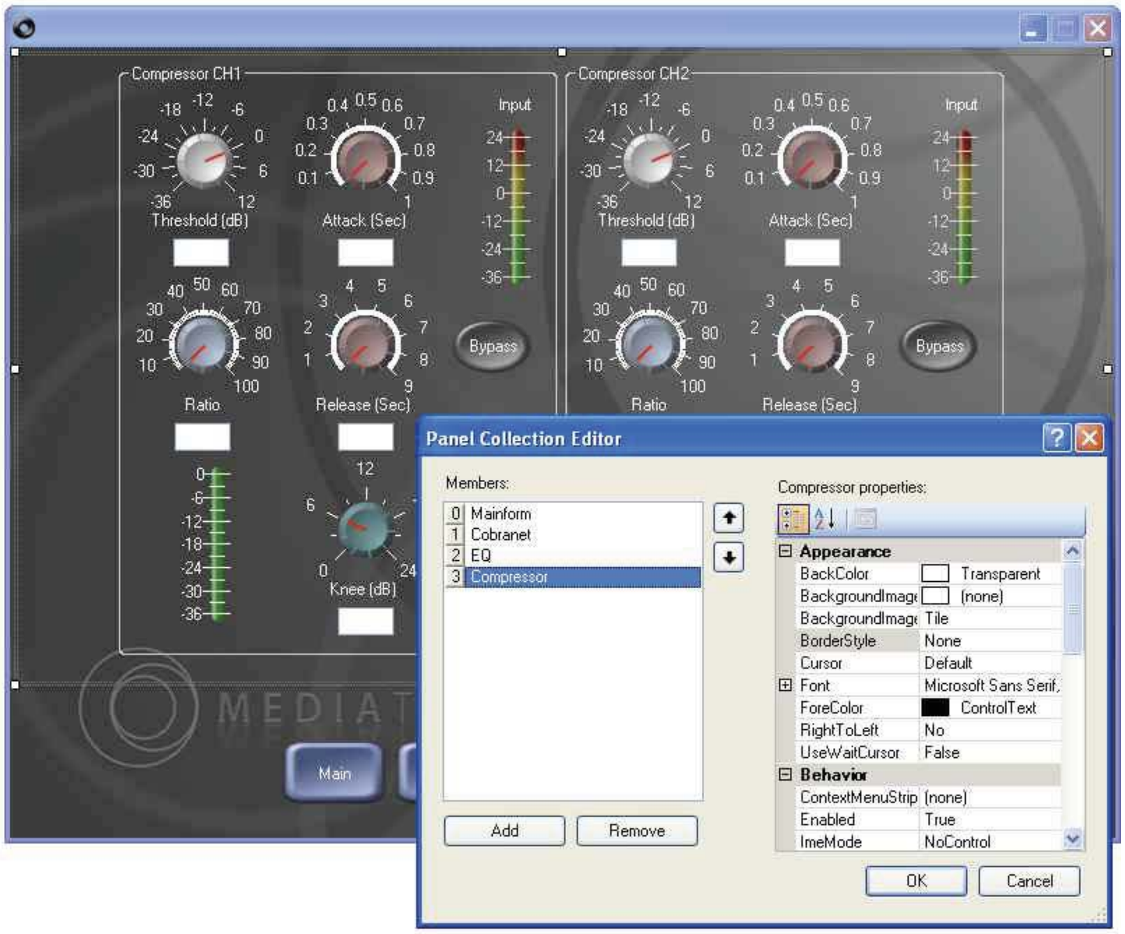


0	Mainform
1	Cobranet
2	EQ
3	Compressor

Appearance	
BackColor	Transparent
BackgroundImage	(none)
BackgroundImage	Tile
BorderStyle	None
Cursor	Default
Font	Microsoft Sans Serif,
ForeColor	ControlText
RightToLeft	No
UseWaitCursor	False
Behavior	
ContextMenuStrip	(none)
Enabled	True
ImeMode	NoControl

Figure 4-31: EQ Panel

Note: In these examples, the Panel background was set to transparent, so that the main Form background image was always visible.



Members:
0 Mainform
1 Cobranet
2 EQ
3 Compressor

Compressor properties:	
Appearance	
BackColor	Transparent
BackgroundImage	(none)
BackgroundImage	Tile
BorderStyle	None
Cursor	Default
Font	Microsoft Sans Serif,
ForeColor	ControlText
RightToLeft	No
UseWaitCursor	False
Behavior	
ContextMenuStrip	(none)
Enabled	True
ImeMode	NoControl

Figure 4-32: Compressor Panel

4.2.2 Names

MTS Control is a windows compliant application and all names must not contain spaces, or any special characters (? / \ * etc). Names should be alphanumeric characters and if a separator is needed, then use the underscore character, eg, "button_1", or "mixer_slider_3".

4.2.3 Graphical Controls: tips and hints

This section covers some simple techniques for configuring the properties of controls and indicators.

4.2.3.1 Setting up Meters and Sliders

Dragging an advanced slider into the Form will result in the example in Figure 4-33 below. The slider is shown horizontal, with a maximum value of 100 and a minimum value of zero.

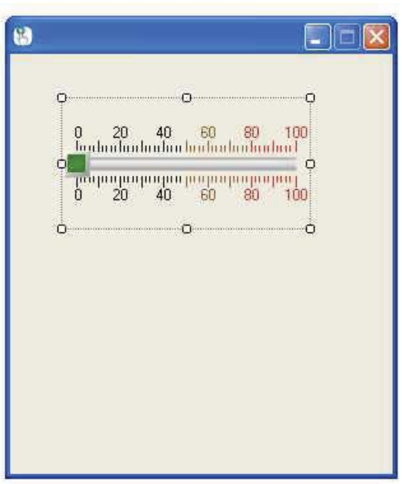


Figure 4-33

The following changes will be made to convert this slider into a useful audio slider with a minimum value of -36dB and a maximum value of +12dB…

- Change orientation from horizontal to vertical
- Make maximum +12 and minimum -36
- Change labels count from 5 above minimum ie 20-40-60-80-100) to 4 above minimum (ie -24, -12, 0, +12)
- Make TicksCount (ie the large scale subdivisions) to 8, ie at 6dB intervals and the TicksSubDivision to 2, ie 3dB intervals
- Change TicksPosition from both (sides) to internal. This will reduce the slider width and allow more audio sliders side by side
- Change LabelsPosition from both to Top left. Again this will help reduce slider size.
- Change slider style from an LED to Value. This will allow the user to easily adjust to an exact value.
- Finally grab one corner of the slider and adjust the size to suit the UI.

This will result in a slider as shown in Figure 4-34 below. Once a slider has been configured, it can be copied and pasted many times either in a project or across projects.

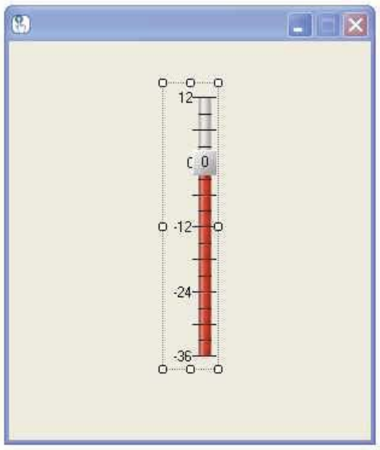


Figure 4-34

Many other properties can be set, such as transparency (change BackColor to Web/Transparency) if the slider is to be set on a graphical background. Similarly the ScaleColor can be set to white, if the background image is a dark color.

Graphical indicators can be setup in exactly the same way as the slider above.

Note: It is good practice to name all controls and indicators, otherwise it is difficult to keep track of many similar controls in a single project.

4.2.3.2 Using Scaled Values

The default Event when opening any knob or slider Control is "valueChanged". The value generated by any control in MTS Control is an integer. However, many applications require a non-integer (decimal) value. For example in a compressor, ratio needs to vary from 1 to 100, but the important numbers are between 1 and 4 and can be 1.5 or 2.5.

For the compressor ratio example, the maximum would be changed to 1000 and minimum to 10, ie increased by a factor of 10 and the "NumberScale" property would be set to 10. The "Value" will now vary from 10 to 1000 (still an integer), but the "ScaledValue" will vary from 1.0 through 1.1, 1.2, 1.3 etc to 100.0, ie there is now a decimal value.

Instead of setting the compressor ratio parameter to "Value", it would now be set to "ScaledValue".

Similarly, compressor attack times vary from 0.001 second to 1 second. Now the maximum would be changed to 1000, the minimum to 1 and the NumberScale to 1000.

Although not strictly necessary, it is good practice to use ScaledValue instead of Value for all controls to avoid mistakenly using Value when a decimal number is needed.

4.2.3.3 'Tick' settings on knobs and sliders

MTS Control generally assumes that maximum and minimum are part of the same evenly spaced sequence of steps, eg 0-100 or -100 to 0, etc. In those examples, the large ticks would probably be set to 0-10-20-30 etc and the small ticks to every 1 or 2 divisions.

In some cases, there is a need to have a maximum or minimum that is out of step with the sequence, eg a compressor ratio will vary from 1 to 100 instead of 0-100. As the scales and ticks are derived from the maximum and minimum values, this can cause some problems with tick settings.

For example, a compressor ratio knob varying from 0 to 100 with the need for a decimal point would have maximum = 1000, minimum = 0, large ticks set at 100 and small ticks at 5. However, if the minimum is changed to 10, then the ticks and scale values are lost, as they no longer match up – see examples in Figure 4-35 below… the left hs a minimum value of 0 and the right has a minimum value of 10.

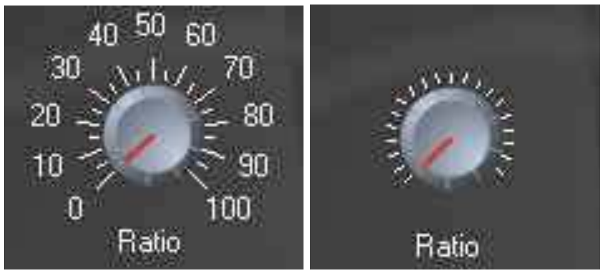


Figure 4-35

There is a simple solution… always set the small ticks value to the same as the minimum value.

4.2.3.4 Setting Vertical or Horizontal action for a Knob

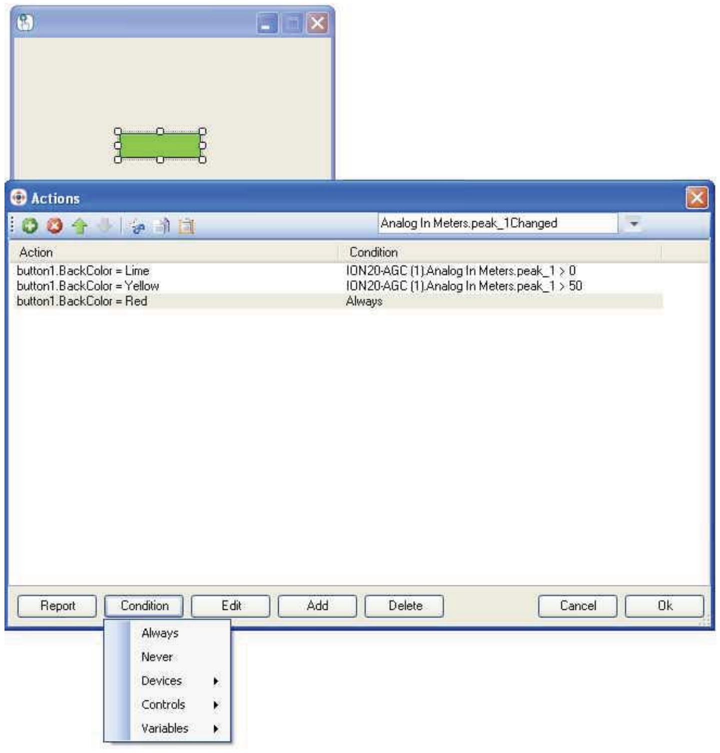
Knobs have a property called "ControlMode" under the Behaviour section. This can be set to Horizontal or Vertical and will determine if a Horizontal or Vertical movement of the mouse will control the knob. Default is Vertical, as it is assumed that the mouse will be used in the same way as a slider.

4.2.3.5 Setting up indicators & LED's and using Conditions

One useful ability of MTS Control is to use an Action to change the Background color or Image of a control or indicator and show current status.

For example, the Event could be a meter "peak_1Changed" and the desired action to set an indicator to emulate a tricolor LED (green, yellow, red). This can be achieved by dragging a simple button into the Form and setting the Actions and Conditions, as follows…

• Action: button1.BackColor = Green	Condition: value >= 0
• Action: button1.BackColor = Yellow	Condition: value > 50
• Action: button1.BackColor = Red	Condition: value > 80



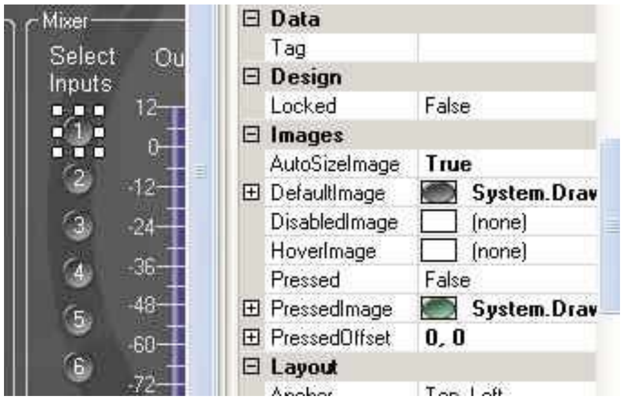
Action	Condition
button1.BackColor = Lime	ION20-AGC (1).Analog In Meters.peak_1 > 0
button1.BackColor = Yellow	ION20-AGC (1).Analog In Meters.peak_1 > 50
button1.BackColor = Red	Always

Figure 4-36: Setting a Condition

Figure 4-36 above shows the Actions dialog box for "Analog In/Meters peakChanged". There are 3 actions, each sets a different color as the button "backcolor". In order to differentiate the actions, a condition is set whereby each color is selected depending on the value of the peak_1 meter.

Editing Note: Clicking on an Action will highlight the line. This can now be copy and pasted into the actions window and will give a second instance of the action. Double clicking on the second Action or Condition will allow direct editing of the action/condition.

Alternatively, a toggle button can have a grey default image and a bright green or red pressed image – see Figure 4-37 below.



Data	
Tag	
Design	
Locked	False
Images	
AutoSizeImage	**True**
DefaultImage	**System.Drav**
DisabledImage	(none)
HoverImage	(none)
Pressed	False
PressedImage	**System.Drav**
PressedOffset	**0, 0**
Layout	

Figure 4-37

In the above example, the “OFF” state is dark grey and the “ON”state is bright green to indicate status.

4.2.3.6 DSP Mixers

Some of the Mixers in the DSP domain use gain as mute controls. For example, the ION0.2 has an 8 channel mute device and an 8 channel gain device on the input to the mixer. Both mute and gain devices have -100dB to +12dB of gain control. When used as a mute, the gain should be switched between -100dB and 0dB.

4.2.3.7 Buttons: Dual state

Be careful with toggle buttons with status indication. These can be confusing to the user. A famous example is pressing the “START” button in MS Windows to switch the machine off.

In touch panel based control systems, a common error is to label a toggle button (eg “POWER”) and then use colors (eg red/green) to indicate the state. Unless the user is familiar with the system, it will be confusing (eg, does red mean alert-power is on, or alert-power is off?). A better solution is to separate the indicator from the button.

However space constraints often require the button to also indicate status. In that situation, the status must be made unambiguous. A POWER button with a dark grey background when off and a bright color background when on is less confusing.

4.2.3.8 Buttons: PressedOffset

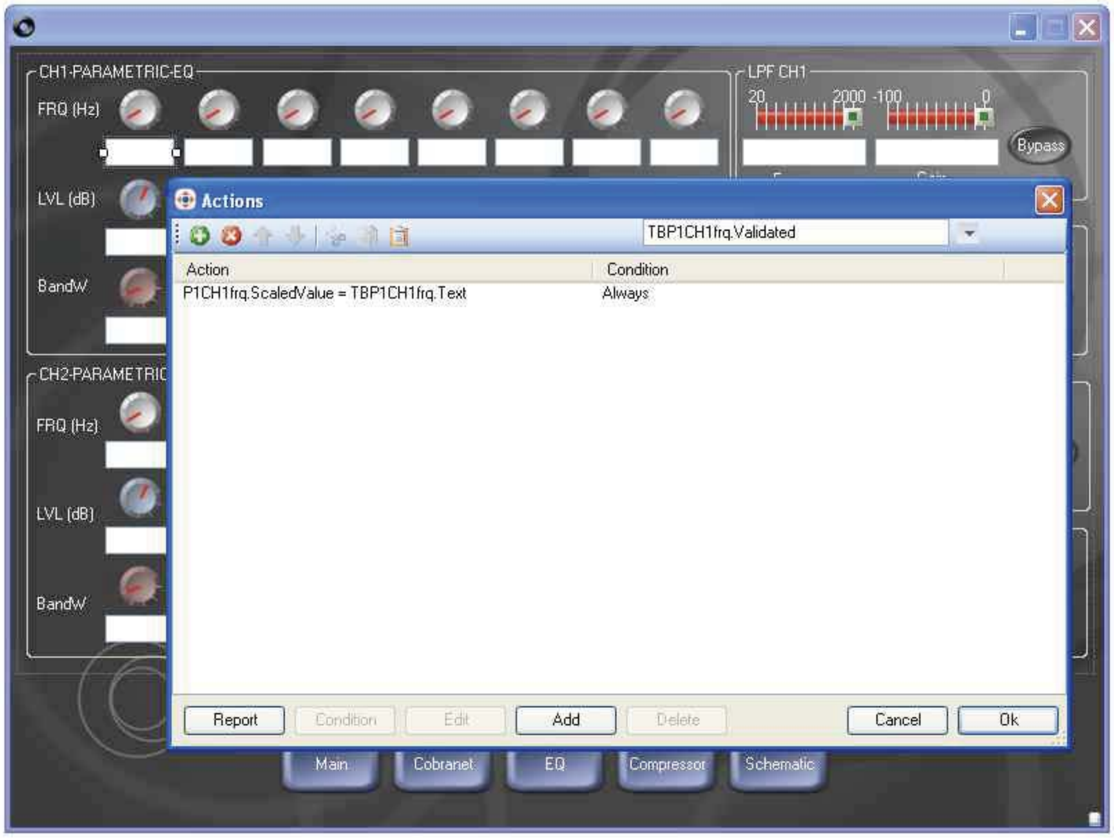
Alternatively, increasing “PressedOffset” in the button properties can indicate dual status. The PressedOffset property moves the pressed image slightly sideways and downwards to show status. However, if the button has a label, then PressedOffset will have an odd appearance. In that situation, set the PressedOffset to 0,0.

4.2.4 Text Boxes

4.2.4.1 Associating Text Boxes and Graphical controls (sliders and knobs)

It is sometimes necessary to have 2 methods of value entry… typically a slider and textbox (or a knob and text box). In that situation, the following process should be observed…

- First use the textbox value to set the slider/knob value. The textbox should not set the device parameter directly. See example in Figure 4-39 below – the textbox used to set the CH1 band 1 Parametric frequency only sets the scaledValue of the knob.

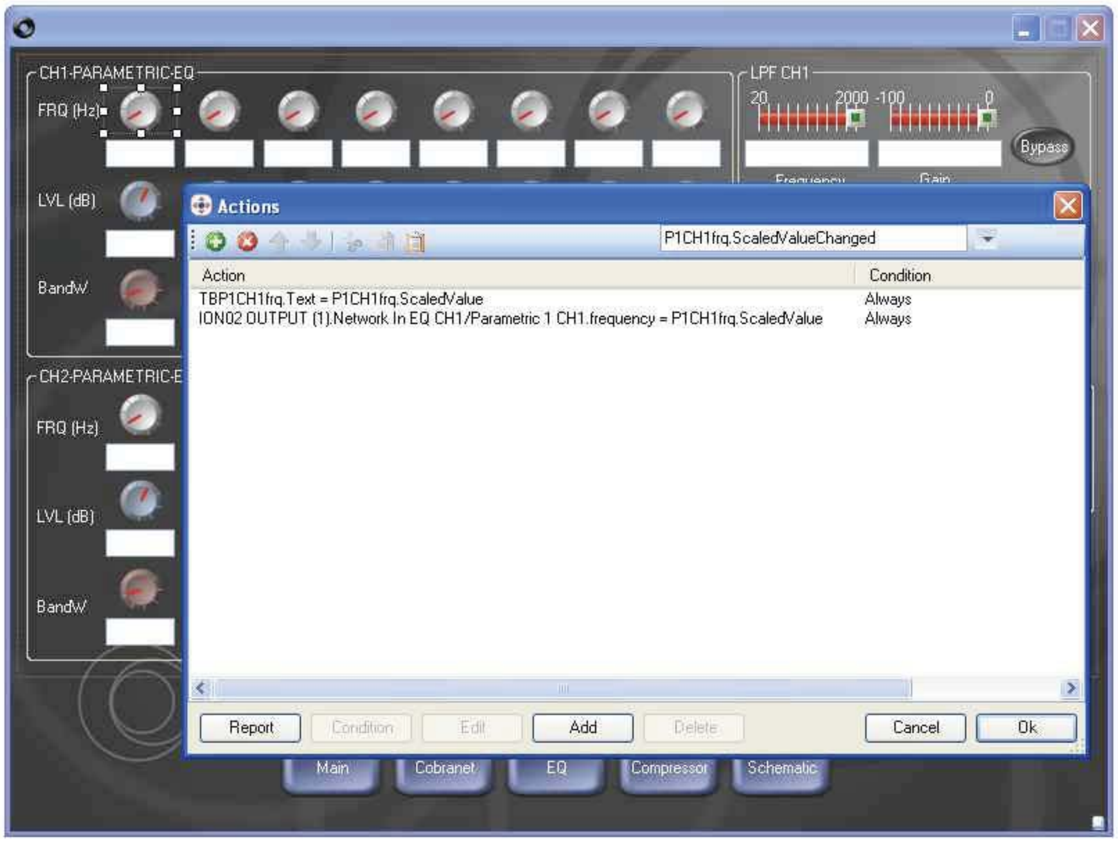


TBP1CH1frq.Validated

Action	Condition
P1CH1frq.ScaledValue = TBP1CH1frq.Text	Always

Figure 4-38

- Then set user limits on the slider/knob-this will also stop invalid data entry at the textbox.
- The slider/knob valueChanged Event should be used to set the device parameter value AND the textbox value. See example in Figure 4-39 below – the scaledValue of the knob is used to set both the textbox AND device parameter value.



P1CH1frq.ScaledValueChanged

Action	Condition
TBP1CH1frq.Text = P1CH1frq.ScaledValue	Always
ION02 OUTPUT (1).Network In EQ CH1/Parametric 1 CH1.frequency = P1CH1frq.ScaledValue	Always

Figure 4-39:

A useful alternative to a slider/knob is the NumericUpDown control in the Windows Tab in the Toolbox. Numeric Up/Down allows a very dense collection of controls (eg EQ) in one sub-Form or Panel.

Note: to change the height of the textbox or NumericUpDown control, then change the Font Height.

4.2.4.2 Text boxes and passwords

It is possible to create a simple embedded password to lock users from certain sub-Forms or Panels. A textbox is placed next to the button used to access the sub-Form or panel. The user enters a password in text box. When the user clicks the button to access the sub-Form or Panel, the “show form” has a condition that checks the text in the text box against a value in the condition.

4.2.4.3 Text Box validation

In the textbox examples used in 4.2.4, “textboxValidated” was used as the Event trigger, rather than the default “textboxChanged” Event for a textbox. The problem with textboxChanged is that the Event will ‘fire’ with every character entered. For example, entering 12 into a gain text box will first set the gain to 1 and then to 12.

To avoid this problem, the textboxValidated Event is used. This will only fire the textbox Event once the enter key is pressed or the mouse is moved away from the textbox.

This will work well except on "Formload", where the default values are being loaded on first opening the form. As no textbox has been validated, no "textboxValidated" Event will fire. There is a simple solution:

- Add a tiny button to the Form. This can be 1x1 pixel and the same background color as the form, ie it becomes invisible. Name the button "invisible_button"
- Set the "AcceptButton" properties for the Form to "invisible_button".
- Add an Action to the Formload to read the device parameter value into the textbox. This will fire an event changed action and therefore pickup the necessary values.

Now textboxValidation will work for Formload.

4.2.5 Rounding errors in Conditions

The numbers used in DSP are 32bit, usually a 31bit number and a sign bit. These are extremely large numbers and will result in tiny rounding errors when converted to the familiar -100dB to +24dB audio values.

Normally, these rounding errors are insignificant and can be ignored. However, there is one situation where the rounding errors can cause problems and that is setting conditions. For example, it may be necessary to determine if the gain has been set to mute, ie -100dB. Normally, -100dB would be used in the condition, but the actual value may be -99.999996dB due to rounding errors.

The solution is simple, set the condition for less than or equal (<=) to -99dB, rather than equal to -100dB

4.2.6 Radio Buttons (Mutually Exclusive)

It is often necessary to set up a group of "Mutually Exclusive" buttons, ie only one button of the group can be on at one time. These are often called "Radio" buttons.

There are two ways of doing this…

4.2.6.1 Windows Radio buttons

Windows provides Radio buttons with the mutually exclusive logic built in. However all Radio buttons loaded into a Form will become part of the same mutually exclusive group.

Creating separate mutually exclusive groups in the same Form requires the use of the Windows GroupBox. Each GroupBox will contain the Radio buttons for the particular mutually exclusive group. See Figure 4-40 below for an example of two GroupBoxes, each containing two Radio Buttons.

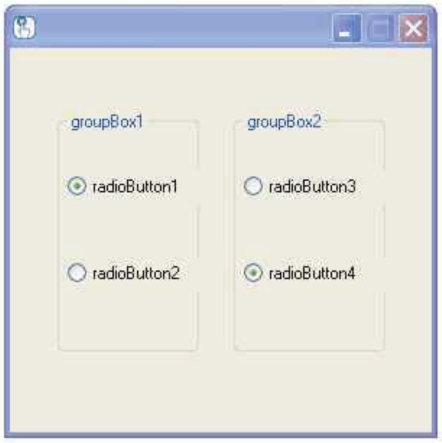
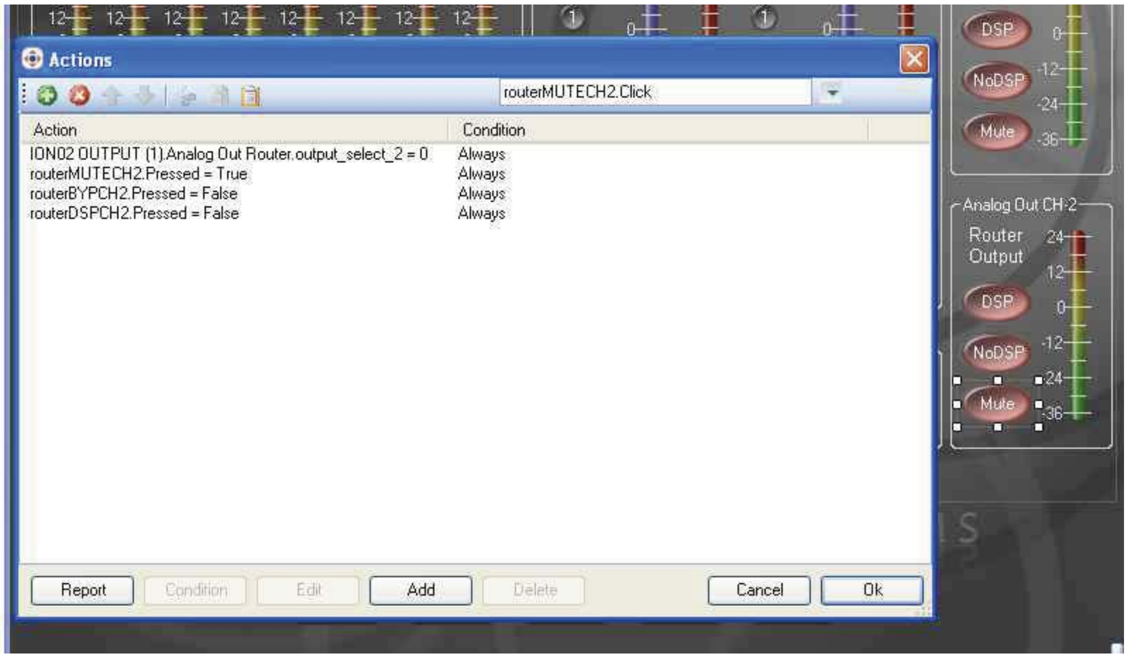


Figure 4-40

4.2.6.2 Programmed Radio buttons

If the visual appearance or sizes of the standard Windows Radio buttons are unacceptable, then any buttons can have their actions grouped mutually exclusively.

For example, the "DSP", NoDSP and Mute" Radio buttons in the ION2.0 sample executable (see Figure 4-41 below) will set the output router to either select the audio with DSP processing, without DSP processing or mute the audio respectively.



Action	Condition
ION02 OUTPUT (1).Analog Out Router.output_select_2 = 0	Always
routerMUTECH2.Pressed = True	Always
routerBYPCH2.Pressed = False	Always
routerDSPCH2.Pressed = False	Always

Figure 4-41

To make the buttons mutually exclusive, the Actions for each button will first set the output router to the required value and it's own pressed state to "TRUE" and then set the other buttons pressed state to "FALSE".

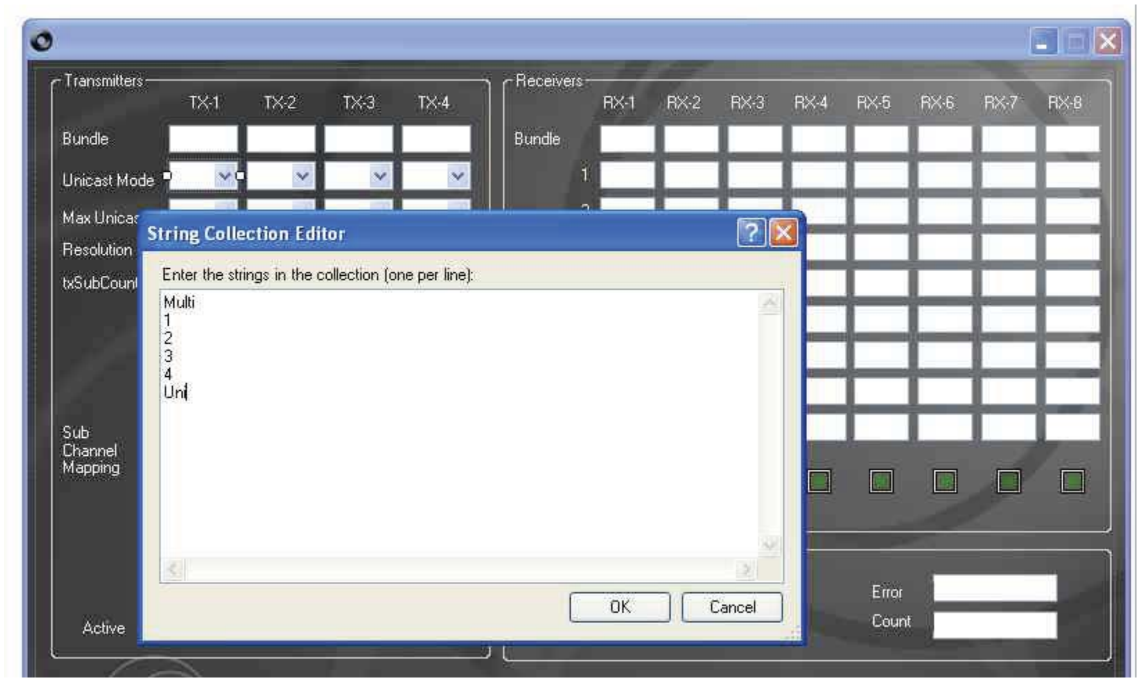
4.2.7 Combo/Dropdown boxes

Drop down list (Combo) boxes are a very useful method for allowing the user to make a choice from a limited number of complicated and non-numeric options.

The Combo box is dropped into the Form and the 3 key properties are…

- DropDownStyle: Should be set to “DropDownList”. That way the user cannot make text entries.
- Data Items (Collection) is where the Options are listed
- MaxDropDownEntries: set to the required quantity

Figure 4-42 below shows the Data Items Collection for the “UniCastMode” Combo box in the ION2.0 sample executable. UniCastMode has 6 options… Always Multicast, Multi-Unicast 1-2-3-4 and Always Unicast. The COMBO box provides a convenient method to allow the user to select from a complex list.



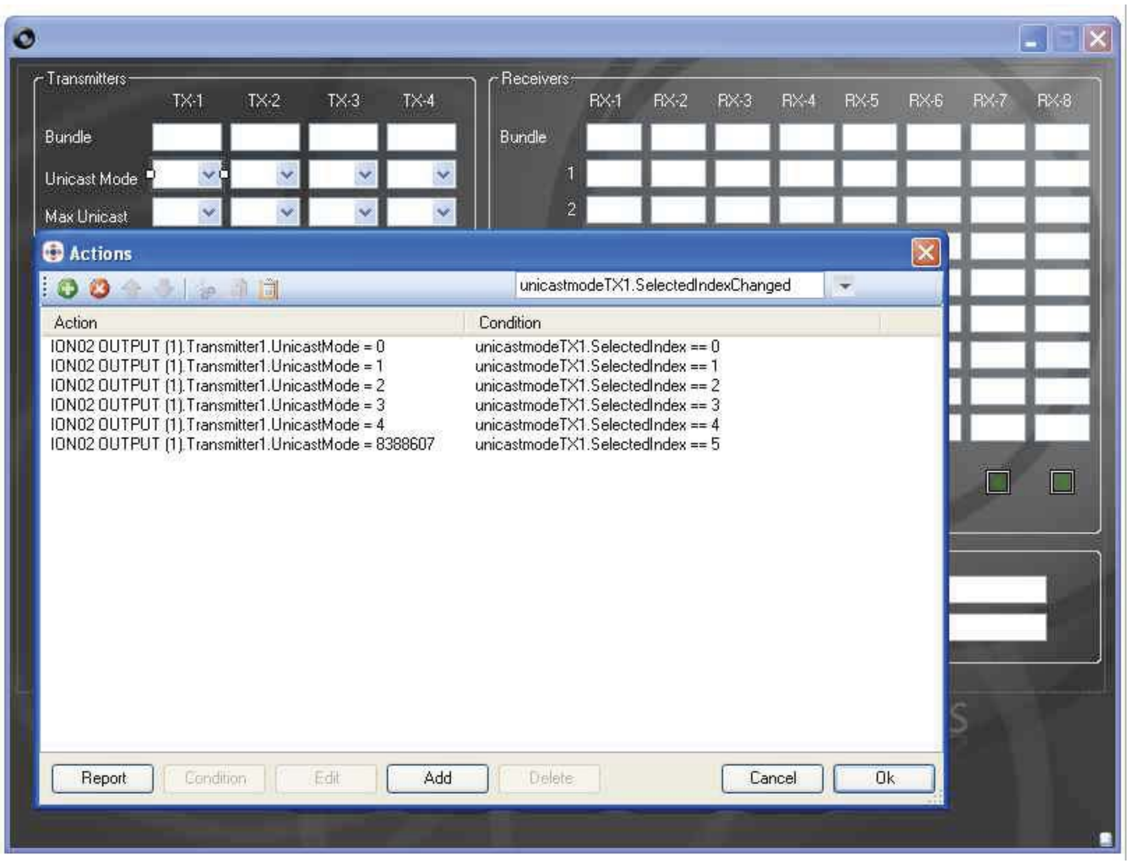
Enter the strings in the collection (one per line):

```
Multi
1
2
3
4
Uni
```

Figure 4-42

Figure 4-43 below shows the Event/Actions dialog window for the COMBO box. The COMBO box is named “unicastmodeTX1” and the Event is “SelectedIndexChanged”. Each selection listed in the “Collection” represents an Index number.

When the index in the COMBO box is changed from its current setting, the SelectedIndexChanged event is fired and the action taken is dependent on the index of the selected item. For example, if the user selects “Uni” from the Collection list, the index is changed to “5” and the device cobranet transmitter 1 has unicast mode set to “8388607”. Conditions in the Actions list ensure that only one of the 6 actions is taken.



unicastmodeTX1.SelectedIndexChanged

Action	Condition
ION02 OUTPUT (1).Transmitter1.UnicastMode = 0	unicastmodeTX1.SelectedIndex == 0
ION02 OUTPUT (1).Transmitter1.UnicastMode = 1	unicastmodeTX1.SelectedIndex == 1
ION02 OUTPUT (1).Transmitter1.UnicastMode = 2	unicastmodeTX1.SelectedIndex == 2
ION02 OUTPUT (1).Transmitter1.UnicastMode = 3	unicastmodeTX1.SelectedIndex == 3
ION02 OUTPUT (1).Transmitter1.UnicastMode = 4	unicastmodeTX1.SelectedIndex == 4
ION02 OUTPUT (1).Transmitter1.UnicastMode = 8388607	unicastmodeTX1.SelectedIndex == 5

Figure 4-43

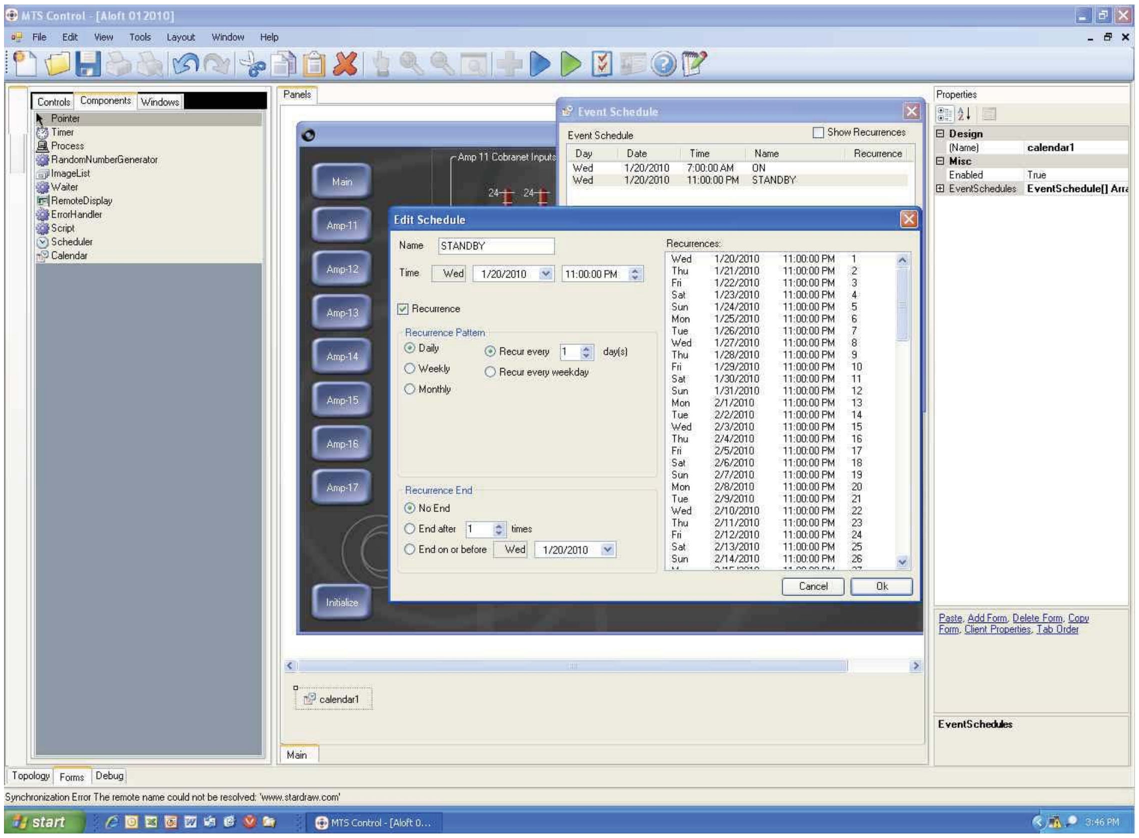
4.3 Calendar Device

The Calendar is a very useful device and simple to set up. The screen shot below (Figure 4-44) shows a calendar device dragged from the Components list and dropped into the form view (components such as calendar and scheduler appear in the lower bar of the form view).

Clicking once on the Calendar device will show the properties. The Calendar can be named (there can be more than one calendar device) and the Event Schedule for the calendar can be opened. The Event Schedule will list all the Events that have been configured. In the screenshot, two events have been set up “ON” and “STANDBY”.

The configuration for “STANDBY” is shown, where the event recurs daily at 11.00pm. This calendar could be used to switch all the MTS Router-amplifiers into standby at 11.00pm and bring them out of standby (ie “ON”) at 7.00am every day.

Once the scheduler events have been configured, they are available within the forms view Event/Actions window and can have one or more actions allocated to the event. Figure 4-45 below shows the scheduled event being selected in an Event/Actions window and Figure 4-46 shows a sequence of amplifiers switched out of standby, plus screen indicators showing System ON.


Event Schedule

Day	Date	Time	Name	Recurrence
Wed	1/20/2010	7:00:00 AM	ON	
Wed	1/20/2010	11:00:00 PM	STANDBY	

Edit Schedule

Name: STANDBY

Time: Wed 1/20/2010 11:00:00 PM

[✓] Recurrence

Recurrence Pattern: [✓ Daily] Weekly Monthly — [✓ Recur every 1 day(s)] Recur every weekday

Recurrence End: [✓ No End] End after 1 times — End on or before Wed 1/20/2010

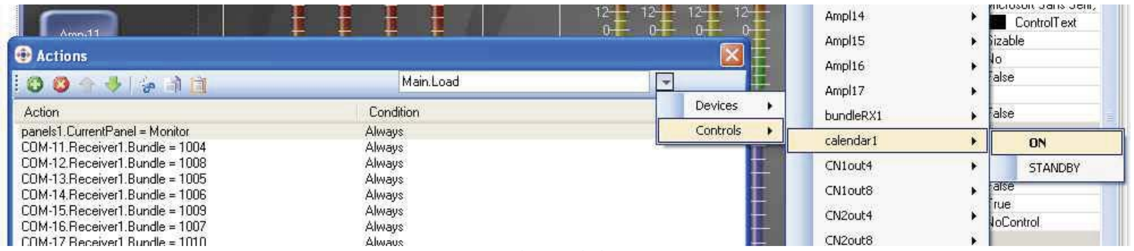
Recurrences:

Wed	1/20/2010	11:00:00 PM	1
Thu	1/21/2010	11:00:00 PM	2
Fri	1/22/2010	11:00:00 PM	3
Sat	1/23/2010	11:00:00 PM	4
Sun	1/24/2010	11:00:00 PM	5
Mon	1/25/2010	11:00:00 PM	6
Tue	1/26/2010	11:00:00 PM	7
Wed	1/27/2010	11:00:00 PM	8
Thu	1/28/2010	11:00:00 PM	9
Fri	1/29/2010	11:00:00 PM	10
Sat	1/30/2010	11:00:00 PM	11
Sun	1/31/2010	11:00:00 PM	12
Mon	2/1/2010	11:00:00 PM	13
Tue	2/2/2010	11:00:00 PM	14
Wed	2/3/2010	11:00:00 PM	15
Thu	2/4/2010	11:00:00 PM	16
Fri	2/5/2010	11:00:00 PM	17
Sat	2/6/2010	11:00:00 PM	18
Sun	2/7/2010	11:00:00 PM	19
Mon	2/8/2010	11:00:00 PM	20
Tue	2/9/2010	11:00:00 PM	21
Wed	2/10/2010	11:00:00 PM	22
Thu	2/11/2010	11:00:00 PM	23
Fri	2/12/2010	11:00:00 PM	24
Sat	2/13/2010	11:00:00 PM	25
Sun	2/14/2010	11:00:00 PM	26

Properties: (Name) calendar1; Enabled True; EventSchedules EventSchedule[] Array

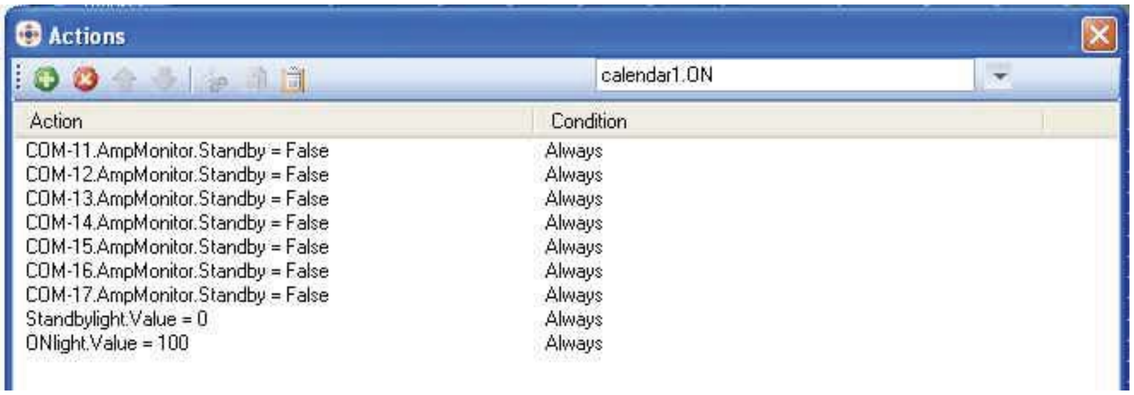
Synchronization Error The remote name could not be resolved: 'www.stardraw.com'

Figure 4-44: Calendar


Actions — Main.Load

Action	Condition
panels1.CurrentPanel = Monitor	Always
COM-11.Receiver1.Bundle = 1004	Always
COM-12.Receiver1.Bundle = 1008	Always
COM-13.Receiver1.Bundle = 1005	Always
COM-14.Receiver1.Bundle = 1006	Always
COM-15.Receiver1.Bundle = 1009	Always
COM-16.Receiver1.Bundle = 1007	Always
COM-17.Receiver1.Bundle = 1010	Always

Figure 4-45


Actions — calendar1.ON

Action	Condition
COM-11.AmpMonitor.Standby = False	Always
COM-12.AmpMonitor.Standby = False	Always
COM-13.AmpMonitor.Standby = False	Always
COM-14.AmpMonitor.Standby = False	Always
COM-15.AmpMonitor.Standby = False	Always
COM-16.AmpMonitor.Standby = False	Always
COM-17.AmpMonitor.Standby = False	Always
Standbylight.Value = 0	Always
ONlight.Value = 100	Always

Figure 4-46

The Calendar will allow daily events to be scheduled. If an event is to be triggered more than once a day, then the Calendar can be used to trigger a Scheduler. The Scheduler device (see Figure 4-47 below) allows multiple events at intervals even down to millisecond time intervals.

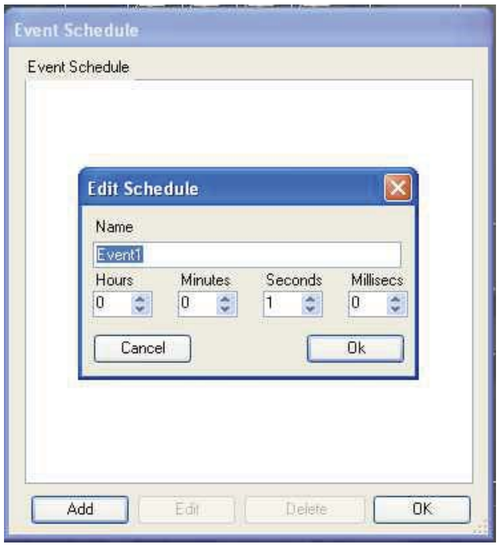
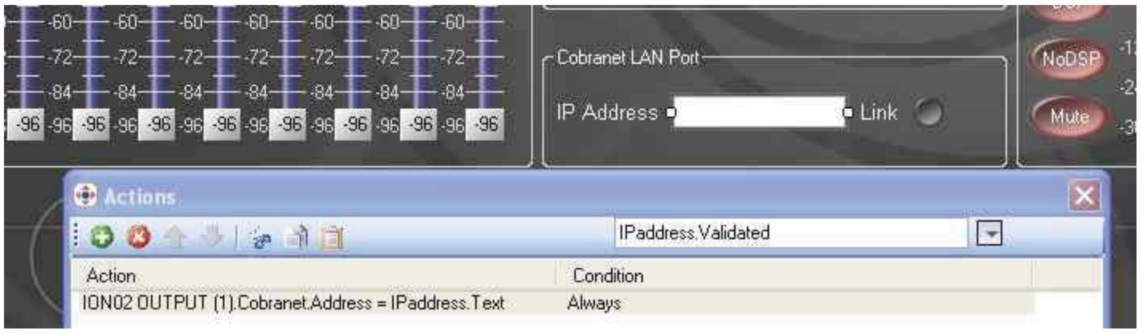


Figure 4-47

4.4 IP Address – Single client configuration

4.4.1 Setting IP address

It is possible to change the IP address of the target device. However, care must be taken to avoid issues in client-server applications and it would only be recommended for single client projects.



Action	Condition
ION02 OUTPUT (1).Cobranet.Address = IPaddress.Text	Always

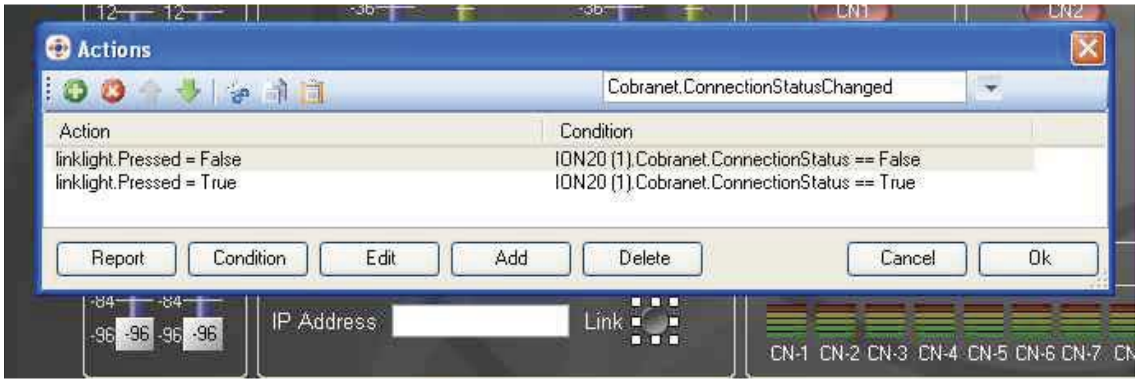
Figure 4-48

The ION2.0 sample demo has a textbox set up to read an IP address-see Figure 4-48 above, where the action simply sets the device Cobranet address to the text box value.

Note: the IP address needs to be entered in 192.168.1.100 format.

4.4.2 Monitoring Connection status

It is often important to monitor connection status. The screenshot in Figure 4-49 below shows that the Cobranet section of the driver has “ConnectionStatusChanged” and this can be used to set the link light to grey (pressed image = false) and green (pressed image = true) according to connection status of the port.


Cobranet.ConnectionStatusChanged

Action	Condition
linklight.Pressed = False	ION20 (1).Cobranet.ConnectionStatus == False
linklight.Pressed = True	ION20 (1).Cobranet.ConnectionStatus == True

Figure 4-49

4.5 Preset Device

MTS Control will allow the store/recall of device parameter values to/from one or more presets. The preset values are stored to the PC hosting MTS Control as files, so it is necessary for the programmer to choose a location to store the presets-usually the same folder as the executable.

Setting up presets is very simple. In the Topology view, open the “Generic Controls” tab (see Figure 4-50 below) and drag a Preset device to the form.

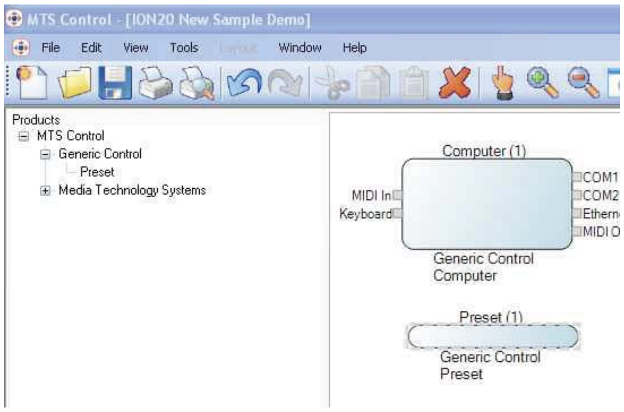
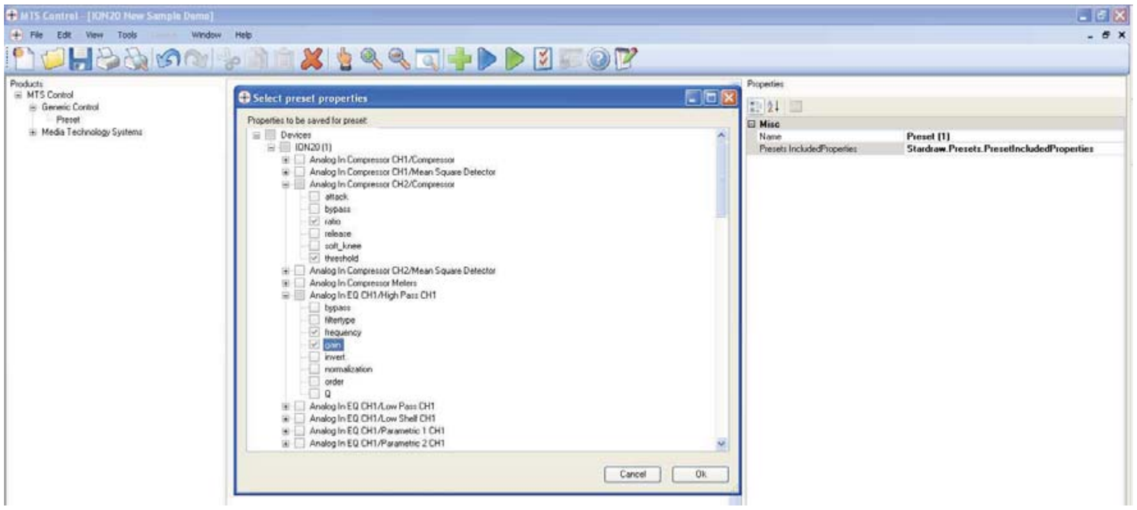


Figure 4-50

A preset device is associated with the parameters available in the MTS ION driver. The association is made by clicking on the “PresetIncluded” property in the Topology window (see Figure 4-51 below) and then selecting the desired parameters. The particular list of parameters associated with a particular preset device forms a unique ‘set’. As soon as one parameter is added or deleted from the ‘set’, then a new preset device would be needed.

For example, if a ‘set’ of equalizer values is needed in a preset, then only the desired EQ parameters for each ION device would be selected in the PresetIncludedProperties list.



Properties to be saved for preset:

- [ ] Devices
  - [ ] ION20 (1)
    - [ ] Analog In Compressor CH1/Compressor
    - [ ] Analog In Compressor CH1/Mean Square Detector
    - [ ] Analog In Compressor CH2/Compressor
      - [ ] attack
      - [ ] bypass
      - [✓] ratio
      - [ ] release
      - [ ] soft_knee
      - [✓] threshold
    - [ ] Analog In Compressor CH2/Mean Square Detector
    - [ ] Analog In Compressor Meters
    - [ ] Analog In EQ CH1/High Pass CH1
      - [ ] bypass
      - [ ] filtertype
      - [✓] frequency
      - [✓] gain
      - [ ] invert
      - [ ] normalization
      - [ ] order
      - [ ] Q
    - [ ] Analog In EQ CH1/Low Pass CH1
    - [ ] Analog In EQ CH1/Low Shelf CH1
    - [ ] Analog In EQ CH1/Parametric 1 CH1
    - [ ] Analog In EQ CH1/Parametric 2 CH1

Misc	
Name	Preset (1)
Presets IncludedProperties	Stardraw.Presets.PresetIncludedProperties

Figure 4-51

Only one preset device is needed for a particular 'set', as many presets (ie different files) can be saved for each preset device. For example, a group of 5 different EQ settings would be stored using one "EQ_Preset" device and 5 files, labeled "EQ_Preset_1", "EQ_Preset_2"… "EQ_Preset_5". Figure 4-52 and Figure 4-53 below show a graphic button click event triggering a device action to set "StorePreset to" "Enter value", where the value entered will be the Preset file name. In this example, it is…

C:\Documents and Settings\Administrator\My Documents\MTS Control\EQ_Preset_1

As presets are stored to the PC HDD, there is no practical limit to the number of preset files that can be created for a particular preset device.

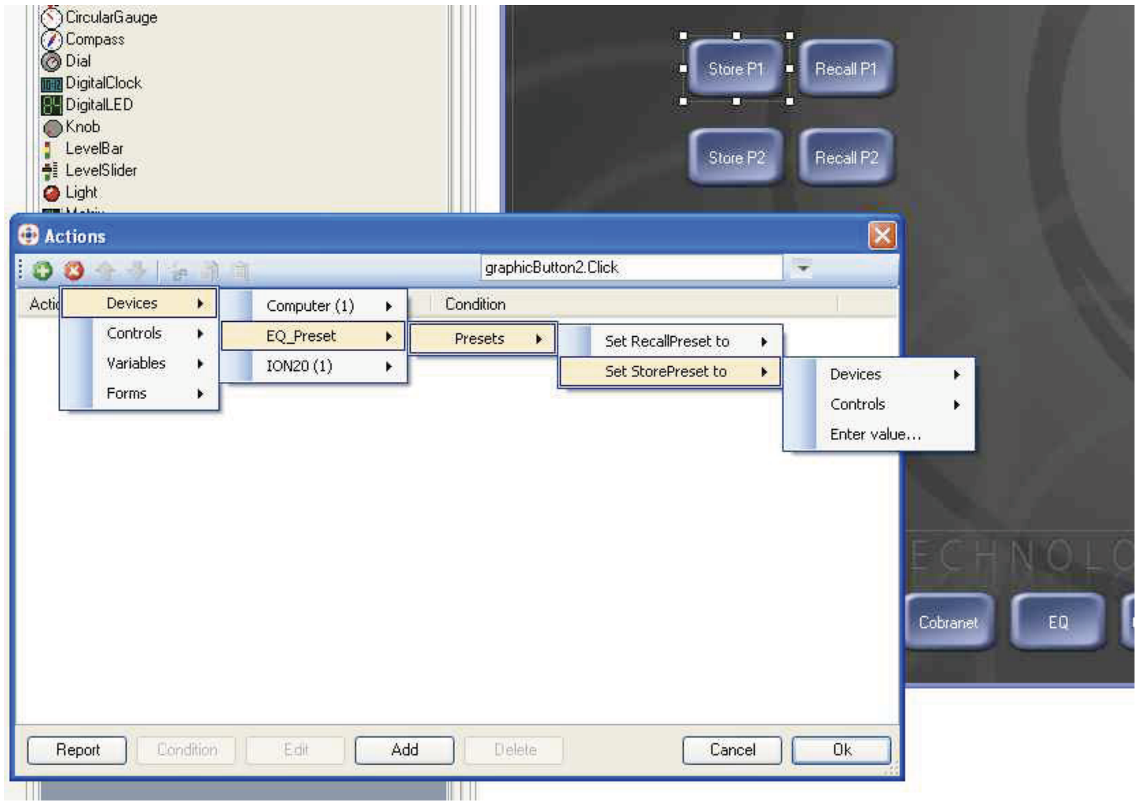
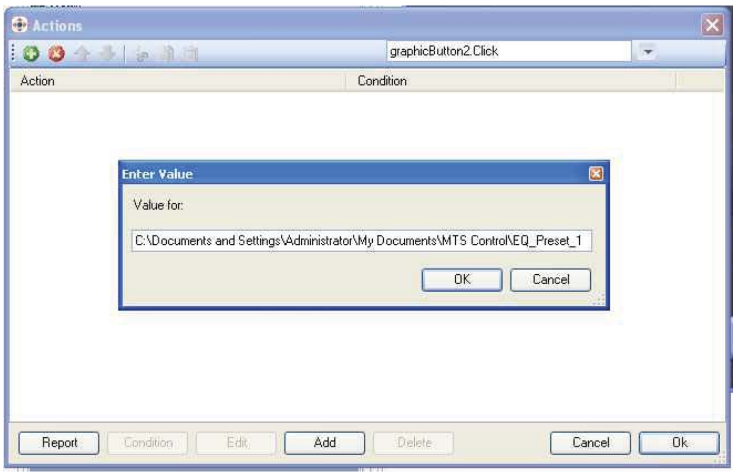


Figure 4-52



C:\Documents and Settings\Administrator\My Documents\MTS Control\EQ_Preset_1

Figure 4-53

Presets devices should ONLY be configured once the Topology view is confirmed, otherwise the 'set' of parameters will need to be frequently adjusted.

NOTE: All sample Project files (*.s03 file type – see Section 4.7) on the CDROM have the default preset file location as…

"C:\Documents and Settings\Administrator\My Documents\MTS Control"

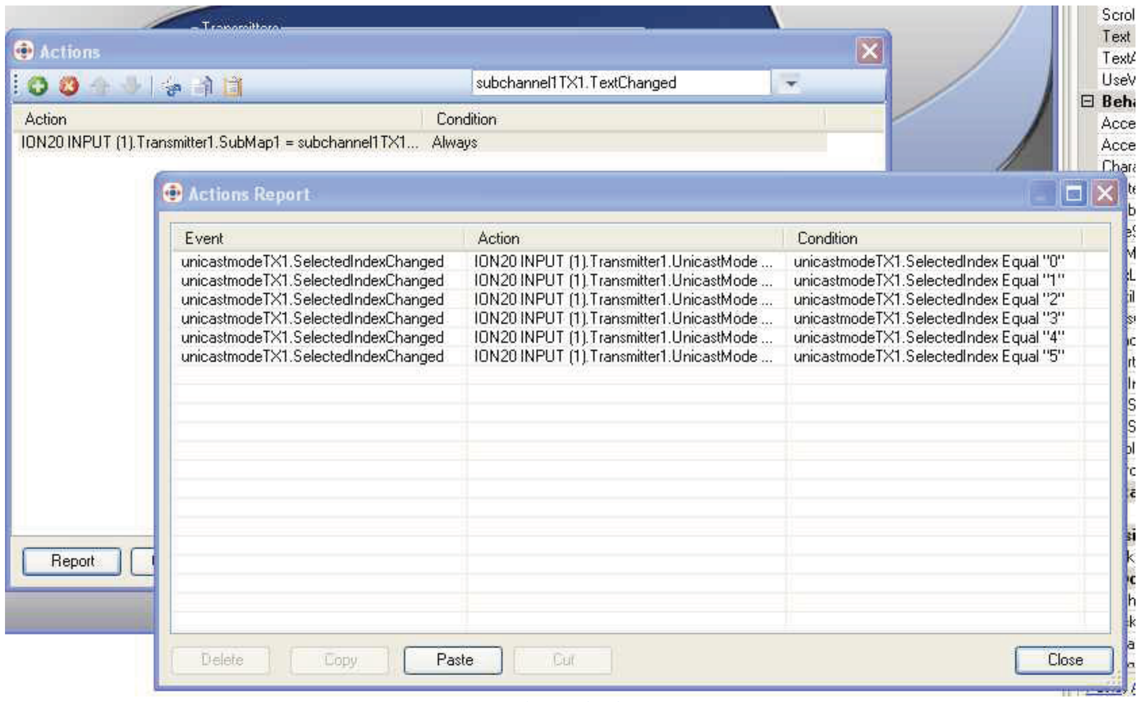
This will need to be edited in the Forms view and changed to the target PC folder location.

4.6 Editing Events and Actions (using MS Excel as an editor)

Simple editing can be done in the Event/Actions dialog window, where clicking on an Action will highlight the line (for copy/paste) and double clicking on the Action or Condition will allow direct editing.

However, an MTS Control application can result in a large number of similar actions (ie “slider_1” controls” gain_1”, “slider_2” controls “gain_2”, etc). Instead of the Event/Actions menu, the fastest way to create a large number of similar actions is to use the report window and copy and paste the basic action into an excel spreadsheet

The report window button is shown at the lower left corner of the Event/Actions dialog window (see Figure 4-54 below)



Action	Condition
ION20 INPUT (1).Transmitter1.SubMap1 = subchannel1TX1...	Always

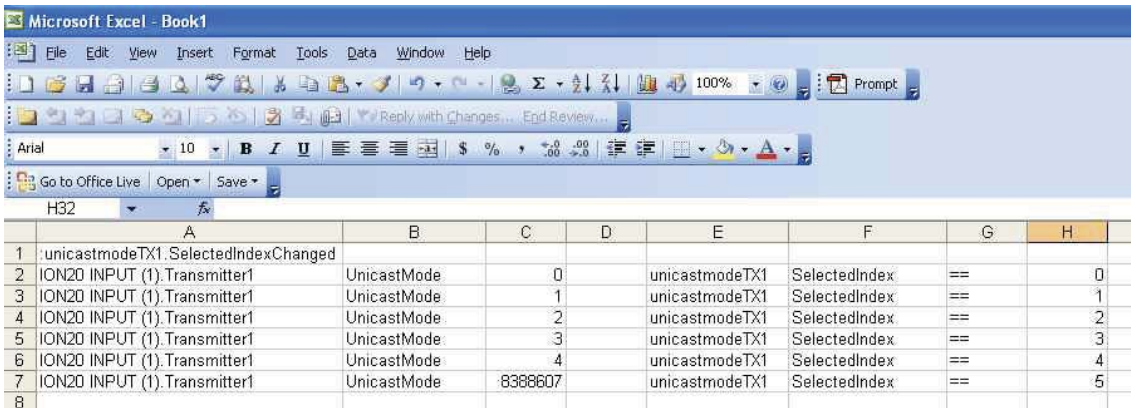
Event	Action	Condition
unicastmodeTX1.SelectedIndexChanged	ION20 INPUT (1).Transmitter1.UnicastMode ...	unicastmodeTX1.SelectedIndex Equal "0"
unicastmodeTX1.SelectedIndexChanged	ION20 INPUT (1).Transmitter1.UnicastMode ...	unicastmodeTX1.SelectedIndex Equal "1"
unicastmodeTX1.SelectedIndexChanged	ION20 INPUT (1).Transmitter1.UnicastMode ...	unicastmodeTX1.SelectedIndex Equal "2"
unicastmodeTX1.SelectedIndexChanged	ION20 INPUT (1).Transmitter1.UnicastMode ...	unicastmodeTX1.SelectedIndex Equal "3"
unicastmodeTX1.SelectedIndexChanged	ION20 INPUT (1).Transmitter1.UnicastMode ...	unicastmodeTX1.SelectedIndex Equal "4"
unicastmodeTX1.SelectedIndexChanged	ION20 INPUT (1).Transmitter1.UnicastMode ...	unicastmodeTX1.SelectedIndex Equal "5"

Figure 4-54

It is only possible to copy/paste entire actions from/to the report window, ie it is not possible to click on an individual event, action or condition..

However, part or all of the events in the report window can be copied into MS Excel (or Word), edited and then the results copied back into the report window.

For example, the unicastmodeTX1 events shown in Figure 4-54 above have been copied into MS Excel below (Figure 4-55). The Event is always at the top of the list and preceeded by a colon (:). Now it is possible to copy those events, actions and conditions and use the text editor to change TX1 to TX2, then TX3 and so on.



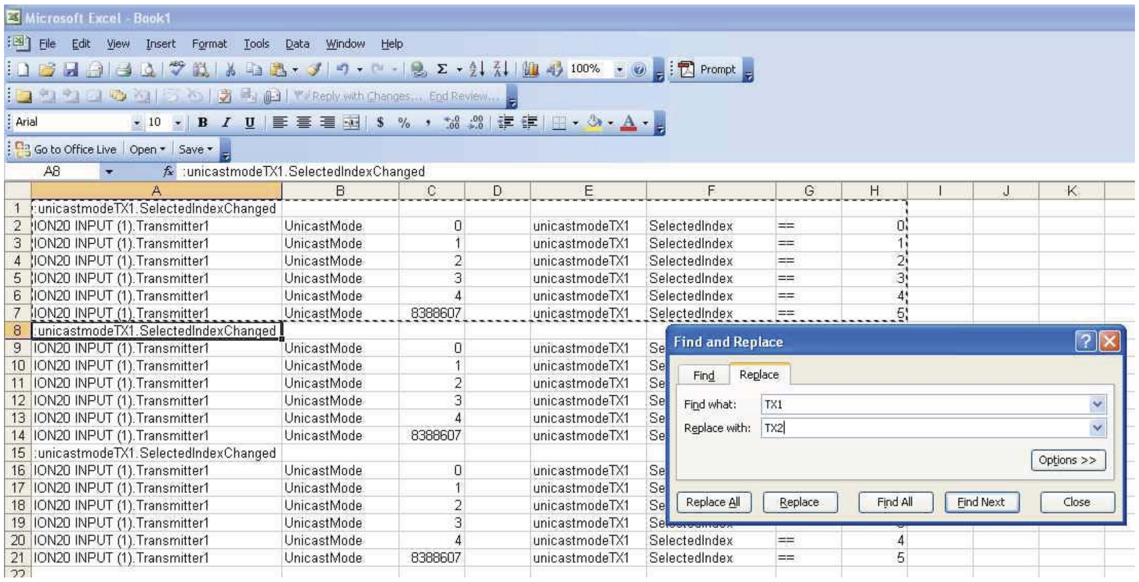
Microsoft Excel - Book1

H32

	A	B	C	D	E	F	G	H
1	:unicastmodeTX1.SelectedIndexChanged							
2	ION20 INPUT (1).Transmitter1	UnicastMode	0		unicastmodeTX1	SelectedIndex	==	0
3	ION20 INPUT (1).Transmitter1	UnicastMode	1		unicastmodeTX1	SelectedIndex	==	1
4	ION20 INPUT (1).Transmitter1	UnicastMode	2		unicastmodeTX1	SelectedIndex	==	2
5	ION20 INPUT (1).Transmitter1	UnicastMode	3		unicastmodeTX1	SelectedIndex	==	3
6	ION20 INPUT (1).Transmitter1	UnicastMode	4		unicastmodeTX1	SelectedIndex	==	4
7	ION20 INPUT (1).Transmitter1	UnicastMode	8388607		unicastmodeTX1	SelectedIndex	==	5
8								

Figure 4-55

Figure 4-56 Below shows the TX1 event and associated actions copied twice. The "Find and Replace" function is used to change TX1 to TX2 and then TX2 to TX3, etc.



Microsoft Excel - Book1

A8 :unicastmodeTX1.SelectedIndexChanged

	A	B	C	D	E	F	G	H
1	:unicastmodeTX1.SelectedIndexChanged							
2	ION20 INPUT (1).Transmitter1	UnicastMode	0		unicastmodeTX1	SelectedIndex	==	0
3	ION20 INPUT (1).Transmitter1	UnicastMode	1		unicastmodeTX1	SelectedIndex	==	1
4	ION20 INPUT (1).Transmitter1	UnicastMode	2		unicastmodeTX1	SelectedIndex	==	2
5	ION20 INPUT (1).Transmitter1	UnicastMode	3		unicastmodeTX1	SelectedIndex	==	3
6	ION20 INPUT (1).Transmitter1	UnicastMode	4		unicastmodeTX1	SelectedIndex	==	4
7	ION20 INPUT (1).Transmitter1	UnicastMode	8388607		unicastmodeTX1	SelectedIndex	==	5
8	:unicastmodeTX1.SelectedIndexChanged							
9	ION20 INPUT (1).Transmitter1	UnicastMode	0		unicastmodeTX1	Se		
10	ION20 INPUT (1).Transmitter1	UnicastMode	1		unicastmodeTX1	Se		
11	ION20 INPUT (1).Transmitter1	UnicastMode	2		unicastmodeTX1	Se		
12	ION20 INPUT (1).Transmitter1	UnicastMode	3		unicastmodeTX1	Se		
13	ION20 INPUT (1).Transmitter1	UnicastMode	4		unicastmodeTX1	Se		
14	ION20 INPUT (1).Transmitter1	UnicastMode	8388607		unicastmodeTX1	Se		
15	:unicastmodeTX1.SelectedIndexChanged							
16	ION20 INPUT (1).Transmitter1	UnicastMode	0		unicastmodeTX1	Se		
17	ION20 INPUT (1).Transmitter1	UnicastMode	1		unicastmodeTX1	Se		
18	ION20 INPUT (1).Transmitter1	UnicastMode	2		unicastmodeTX1	Se		
19	ION20 INPUT (1).Transmitter1	UnicastMode	3		unicastmodeTX1	Se		
20	ION20 INPUT (1).Transmitter1	UnicastMode	4		unicastmodeTX1	SelectedIndex	==	4
21	ION20 INPUT (1).Transmitter1	UnicastMode	8388607		unicastmodeTX1	SelectedIndex	==	5
22								

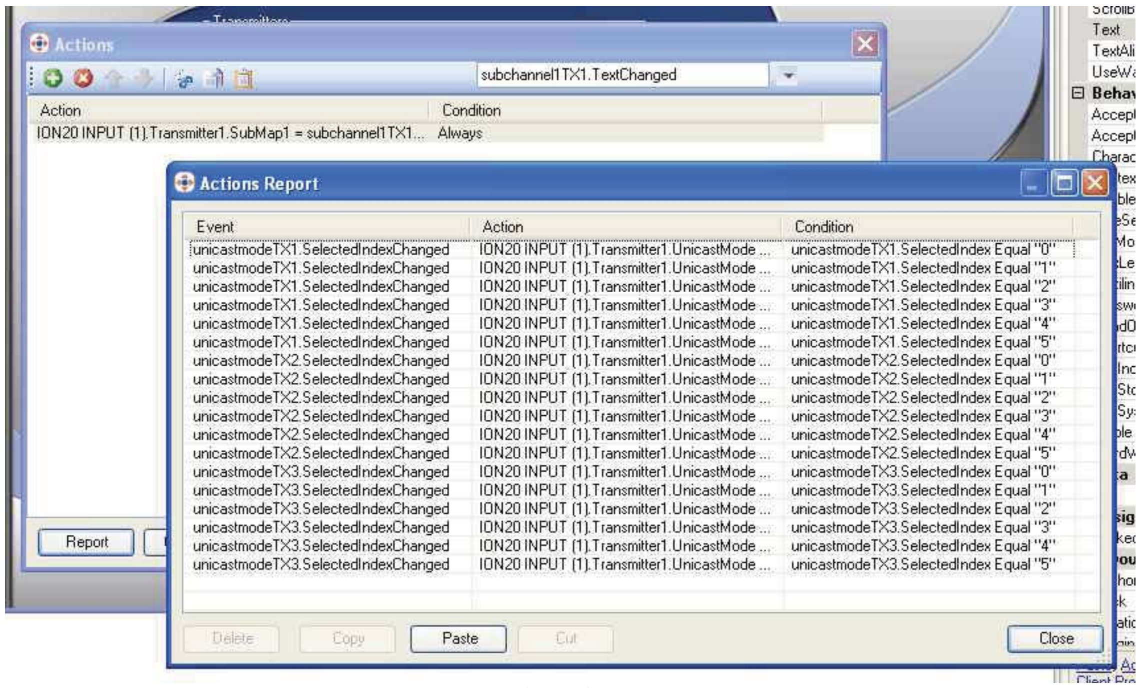
Find and Replace

Find what: TX1

Replace with: TX2

Figure 4-56

Once the editing process has been completed, the spreadsheet cells arte copied and pasted into the reports window (see Figure 4-57 below).



subchannel1TX1.TextChanged

Action	Condition
ION20 INPUT (1).Transmitter1.SubMap1 = subchannel1TX1...	Always

Event	Action	Condition
unicastmodeTX1.SelectedIndexChanged	ION20 INPUT (1).Transmitter1.UnicastMode ...	unicastmodeTX1.SelectedIndex Equal "0"
unicastmodeTX1.SelectedIndexChanged	ION20 INPUT (1).Transmitter1.UnicastMode ...	unicastmodeTX1.SelectedIndex Equal "1"
unicastmodeTX1.SelectedIndexChanged	ION20 INPUT (1).Transmitter1.UnicastMode ...	unicastmodeTX1.SelectedIndex Equal "2"
unicastmodeTX1.SelectedIndexChanged	ION20 INPUT (1).Transmitter1.UnicastMode ...	unicastmodeTX1.SelectedIndex Equal "3"
unicastmodeTX1.SelectedIndexChanged	ION20 INPUT (1).Transmitter1.UnicastMode ...	unicastmodeTX1.SelectedIndex Equal "4"
unicastmodeTX1.SelectedIndexChanged	ION20 INPUT (1).Transmitter1.UnicastMode ...	unicastmodeTX1.SelectedIndex Equal "5"
unicastmodeTX2.SelectedIndexChanged	ION20 INPUT (1).Transmitter1.UnicastMode ...	unicastmodeTX2.SelectedIndex Equal "0"
unicastmodeTX2.SelectedIndexChanged	ION20 INPUT (1).Transmitter1.UnicastMode ...	unicastmodeTX2.SelectedIndex Equal "1"
unicastmodeTX2.SelectedIndexChanged	ION20 INPUT (1).Transmitter1.UnicastMode ...	unicastmodeTX2.SelectedIndex Equal "2"
unicastmodeTX2.SelectedIndexChanged	ION20 INPUT (1).Transmitter1.UnicastMode ...	unicastmodeTX2.SelectedIndex Equal "3"
unicastmodeTX2.SelectedIndexChanged	ION20 INPUT (1).Transmitter1.UnicastMode ...	unicastmodeTX2.SelectedIndex Equal "4"
unicastmodeTX2.SelectedIndexChanged	ION20 INPUT (1).Transmitter1.UnicastMode ...	unicastmodeTX2.SelectedIndex Equal "5"
unicastmodeTX3.SelectedIndexChanged	ION20 INPUT (1).Transmitter1.UnicastMode ...	unicastmodeTX3.SelectedIndex Equal "0"
unicastmodeTX3.SelectedIndexChanged	ION20 INPUT (1).Transmitter1.UnicastMode ...	unicastmodeTX3.SelectedIndex Equal "1"
unicastmodeTX3.SelectedIndexChanged	ION20 INPUT (1).Transmitter1.UnicastMode ...	unicastmodeTX3.SelectedIndex Equal "2"
unicastmodeTX3.SelectedIndexChanged	ION20 INPUT (1).Transmitter1.UnicastMode ...	unicastmodeTX3.SelectedIndex Equal "3"
unicastmodeTX3.SelectedIndexChanged	ION20 INPUT (1).Transmitter1.UnicastMode ...	unicastmodeTX3.SelectedIndex Equal "4"
unicastmodeTX3.SelectedIndexChanged	ION20 INPUT (1).Transmitter1.UnicastMode ...	unicastmodeTX3.SelectedIndex Equal "5"

Figure 4-57

4.7 Sample MTS Control Projects

The CDROM has a number of MTS Control sample Project source code files (*.s03). These are not opened using MTS Control, but must be imported using the File/Import Project drop down menu. Similarly, if a Project source code (rather then a working executable) is to be sent to a jobsite, then the File/Export Project menu must be used.

5 Index