NetPlay Control guide

Revision 8.0
5/17/22

Overview

NetPlay Audio and NetPlay video expose the following functionality to external APIs

- NetPlay general control socket
- **NetPlay virtual matrix controller socket** **(most drivers will only use this API)**
- Squeezebox control socket
- XBMC JSON control (NetPlay video only)
- CEC output
- IR output
- RS232 IO

Hardware specifics

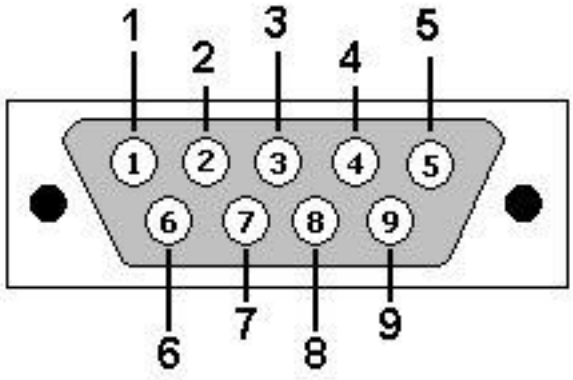
RS232
The NBX100 and VRX010 use a 3.5mm stereo female connector for connection.

Connector pinout:

TIP: RX (to NBX)
Ring: TX (from NBX)
Sleeve: Ground

Please use the DB9 female to 3.5mm stereo cable which can be purchased from Video Storm. The cable pin order is TIP => pin 2 of female DB9, Ring => pin 3 of female DB9, Sleeve => pin 5 of female DB9. We also provide a Male2Male DB9 adapter with the NBX cable. When this adapter is used with the stereo => DB9 cable, the resulting output is a standard DB9 male controller interface (shown below).

The NAB board inside a CMX switch uses the XP male db9 output. It also has the same pinout as shown below.



DB9: View looking into male connector

Only 3 pins on the connector are needed:

Pin 2: RX (to NBX)
Pin 3: TX (from NBX)
Pin 5: Ground

This output can be connected directly to the female db9 CONTROL port on any CMX switch.

NBX010 uses a standard USB to RS232 adapter for RS232. The adapter has the same db9 pinout as above.

IR

Output
3.5mm mono or stereo plug
For connection to 5V emitter
Tip positive, Sleeve negative

Input (not currently support by CLI, only XBMC)
3.5mm stereo plug
For connection to 5V detector
Tip signal, Ring ground, sleeve supply

CEC

On HDMI output of VRX010 only

Protocol settings

RS232 output
Baud rate : 9600
Data bits: 8
Stop bits: 1
Parity: None
Flow control: None

NetPlay TCP control port
Protocol: TCP/Telnet
Port: 23 (single connection only)
Port: 9091 (multiple connection capable)

NetPlay virtual matrix control port (one per NetPlay Video system on designated device)
Protocol: TCP/Telnet
Port: 9092 (multiple connection capable)

Squeezebox CLI server control port (one per system on designated device)
Protocol: TCP/Telnet
Port: 9090
Documentation at http://wiki.slimdevices.com/index.php/Logitech_Media_Server_CLI

XBMC JSON control
Protocol: TCP/Telnet
Port: 9090 *(can be changed, but not recommend to run both Squeeze server and XBMC on one platform)*
Documentation at http://wiki.xbmc.org/?title=JSON-RPC_API

Cover art for the currently playing song is available at
Http://<NBA-IP:80>/coverart<#> where # is the output channel #. Coverart will be JPG format.

Virtual matrix control via HTTP GET
http://<VMM-IP>/vm_request.php?cmd=PROTOCOLCMD
Note PROTOCOLCMD must be URL encoded properly

Commands

NBX/VRX uses the following algorithm to translate the Telnet socket interface to RS232:

1. If the command is a valid NetPlay command (see command set), it will process it locally and provide the echo/response directly back to the Telnet socket.
2. Otherwise, all characters will be directly forwarded to the RS232 TX output unaltered. Note that the output is implicitly buffered by <cr>, since step 1 requires a <cr> termination
3. All RX data received from RS232 port is forwarded directly to the Telnet socket unaltered. Note there is no buffering in this direction.

NAB uses the following algorithm to translate the Telnet socket interface to RS232:

1. If the command is a valid NAB command (see command set), it will process it locally and provide the echo/response directly back to the Telnet socket.
2. If the command begins with the forwarding directive (/F, 2 characters), it will strip these 2 characters and forward the rest of the command (up to and including the next <cr>) to the RS232 output port (XP). Note that the output is implicitly buffered by <cr>, since step 1 requires a <cr> termination.
3. Otherwise, all characters will be directly forwarded to the CMX microprocessor with a '&' character pre pended. This character lets CMX know that this is a command to be processed locally. Note that the output is implicitly buffered by <cr>, since step 1 requires a <cr> termination.
4. All RX data received from RS232 port is forwarded directly to the Telnet socket unaltered. Note there is no buffering in this direction.

NBX/VRX will echo back all characters sent to it. This is the easiest way to verify if your cable connection is correct. The NBX does not add <lf> after any <cr> received, so if you are using windows Hyper-terminal you should change the default settings to allow line feed on carriage return. NBX will only send the echo for valid commands terminated by <cr>.

All commands are terminated by <cr> (carriage return, ascii code 0xD). **NOTE: all references to <cr> in this document mean the single ascii character NOT the four characters "<cr>"**.

The following are the valid command sets for all NBX/NAB/VRX/VTX devices. Different models can be differentiated by the controller by the status readback.

All NetPlay commands start with a Q and end with a <cr>

NetPlay command set:

QCFabbcccz+<cr> : Config control

a is one of (C/R/S) selects the file (C is config, S server, R renderer)
bb is the row index to modify
ccc is the column index to modify (set to 000 for single columns)
z+ is the data to store, terminated by the <cr>

See the config files section at the end for details on the fields

QSETIPaaa.aaa.aaa.aaa<cr> : Set static IP address to aaa field
QSETMKaaa.aaa.aaa.aaa<cr> : Set subnet mask to aaa field
QSETGWaaa.aaa.aaa.aaa<cr> : Set gateway address to aaa field
QSETDNaaa.aaa.aaa.aaa<cr> : Set dns address to aaa field
QSETBCaaa.aaa.aaa.aaa<cr> : Set broadcast address to aaa field
QSETDHCPx<cr>: x=0 => STATIC, x=1 => DHCP
QSETWLANx<cr>: x=0 => wireless off, x=1 => wireless on (adapter required)

QSETSSIDaaa<cr> : Set SSID to aaa field
QSETPWDaaa<cr> : Set WPA PSK password to aaa field
QSTATETH0>cr>: Request details on Ethernet interface.
    Output will be the ifconfig details on eth0 framed by
    QSTATETH0<cr>
    OK<cr>
    Data

QRESTART<cr>: Reboot NBX
QHALT<cr>: Halt NBX OS so power can be removed safely
QUPDATEFW<cr>: Updates the firmware from web server and reboots
QSDDPI<cr>: Send SDDP identity event

QSTATVER<cr> : Request device version
    Output will be the version id framed by
    QSTATVER<cr>
    OK<cr>
    Version string
QSTATCONFabbb<cr> : Request config data
    a is one of (C/R/S) selects the file (C is config, S server, R renderer)
    bbb is the row index (255 to return all rows with header)
    Output will all data columns of that row framed by
    QSTATCONFabbb <cr>
    OK<cr>
    Data
QSTATI2Sa<cr> : Request audio channel information and Metadata
    a is the optional argument to return only one specific of the output
    Metadata includes Title, Artist, Album, Server ID, Player ID
    Output will be the I2S status, one row per output framed by
    QSTATI2S<cr>
    OK<cr>
    Header with column labels
    8 rows of data
QSTATI2Daa<cr> : Request virtual channel information and Metadata
    aa is the requested virtual channel (01-48)
    will return 1 line of data in the same format as above
QSAVEa<cr>: Saves the metadata for the current song on audio channel a
QSTATTOP<cr> : Request cpu utilization
    Output will be the output of TOP header framed by
    QSTATTOP<cr>
    OK<cr>
    Data
QCOVERART#<cr>: Request coverart JPG for channel #
    Output will be
    QCOVERART# <cr>

OK<cr>
width height size JPGDATA<cr>
JPGDATA is base64 encoded

QDELP###<cr>: Delete source profile
if ### is omitted, all source profiles deleted

QREADP###<cr>: Read source profile
if ### is omitted, all profiles sent
Profile_number num_streams video_res video_aspect audio0_channels audio1_channels

NetPlay transport controls

These controls are only valid for all active connections.
These controls only work if the server currently allows the operation.
NOTE: Stopping the service will cause disconnection, so you will not be able to restart using this interface. Pausing will keep the connection active for up to 5 minutes before disconnecting. Note that live streams have limited transport control.

QPLAY#<cr>
QSTOP#<cr>
QDSCN#<cr> (same as stop expect NDCN will not be issued)
QPAUSE#<cr>
QPLPAUSE#<cr> (toggle play/pause)
QSKIPFW#<cr>
QSKIPBK#<cr>
QSIGPS#<cr> (start pause timer but do not pause)
is always the output channel number (1-8) NBX010/VRX010 is always 1

(video only commands)
QSEEKFWxxx<cr> (seek forward X sec)
QSEEKBWxxx<cr> (seek backward X sec)
QLOWLAT<cr> (set low latency mode)
QHIGHLAT<cr> (set high latency mode (default))

NetPlay decoder volume controls

These allow setting the output volume, EQ, audio delay of the decoder directly.
QASETVOLxxx<cr> Volume 0-256 (192 nom)
QASETBALxxx<cr> Balance 0-256 (128 nom)
QASETBASSxxx<cr> Bass 0-256 (128 nom)
QASETTREBxxx<cr> Treble 0-256 (128 nom)
QASETDELxxx<cr> Delay 0-256 (1/4 frame per tick)

NetPlay Audio transmit controls

These controls are to control audio server capabilities.
On NAB, this allows forwarding matrix inputs to remote zones/speakers
On NBX/VRX, this allows audio return channel and audio rebroadcasting
Multiple connections to the same channel are possible (via unicast)

QATPL# rtsp_url<cr> (new stream from RTSP_url)
QATSP#<cr> (stop all streaming on channel #)
is the output I2S channel number

Server RTSP_urls are defined as follows
VRX: rtsp://ip_addr:8554/input1 is the audio return channel
(analog, coax, opt selection in VRX general setup config)
NAB: rtsp://ip_addr:8554/input1 through input8 are the physical network output channels from CMX-A2 (N01 - N08)
rtsp://ip_addr:8554/inputv01 through inputv48 are the virtual server channels defined in setup.

rtsp://ip_addr:8554/test is a speaker test signal on both platforms
Default encoding is L16-44100
Use URL /aac_x for AAC encoding
Use URL /raw_x for raw bitstream

NetPlay Video decode controls

These controls are to control video renderer/decode services. The normal transport controls above can be used on the active stream. Note that NetPlay decoders can play multiple video streams simultaneously. In this case, the transport controls apply to ALL current streams. Normally these controls are not used directly (the virtual matrix driver uses these).

QVMPF options url<cr> Play first url with options:
this command STOPS any current streams
QVMPL options url<cr> Play url with options:
this command starts a new stream
QXBMCON<cr> turn on xbmc gui
QXBMCOFF<cr> hide xbmc gui
QSTATOSD<cr> toggle on/off the debug OSD
QSTATVID<cr> : Request info on currently playing video
One line returned for each active video stream
url options latency state media_time duration

NetPlay Video encode controls

These controls are to control video encoder services. Normally these controls are not used directly (the virtual matrix driver uses these).

QVENCPL #a DEST_IP options <cr> Start streams
#a is the input number + stream#*2
DESP_IP is the target udp://IP:Port or rtp://IP:port/ttl
VTX can support multiple streams, each one identified by the destination IP address

QVENCSTOP #a DEST_IP<cr> Stop streams
If DEST_IP is present, stop stream from input #a to that IP
If DEST_IP not present, stop all streams from input #a
If a not present, stop all streams

QSTATENC<cr> : Request info on current streams
One line returned for each active video stream
Source instance *DEST_IP details*

NetPlay Video encode config

QVENCEDIDMax<cr>
Set to mask certain formats from the encoder EDID
a Input (0 or 1)
x 0 (default) = nothing masked (x is a 4 bit bitfield)
bit 0 = Mask AC3 (Dolby)
bit 1 = Mask DDP (Dolby digital plus)
bit 2 = Mask DTS
bit 3 = Mask MCA (multichannel PCM)

QVENCAOUTMab<cr>
Sets the mode of the audio outputs (setting is persistent)
Analog and optical output are controlled here
a 0 (default) = analog output is active
1 = optical output is active
b 0 (default) = audio output same as hdmi out audio
Decoder output selectable from any network source
1 = audio output is delayed feed through from audio input
Not valid when using optical output
2 = audio output is delayed audio from both hdmi inputs
L = ch1 mono, R = ch2 mono
3 = audio output is delay audio from hdmi 1
4 = audio output is delay audio from hdmi 2
Delay value is set by QASETDELxxx command (persistent)

NetPlay RS232 controls

Any non recognized data will normally be forwarded to RS232 if present.
However, we also have a command to explicitly encode the data for transport

QSRSbase64_ENCODED_DATA<cr>
Sends base64 encoded data to output (decodes it first)

NetPlay IR controls

Decoder control
QSIRPULSE R=## hex_code <cr> pulse proto format hex code via IR ## times
QSIRSTART hex_code <cr> start sending proto format hex code via IR
QSIRSTOP hex_code <cr> stop sending proto format hex code

Source control (a is either 0 or 1)
QSIRPULSEa R=## hex_code <cr> pulse proto format hex code via IR ## times
QSIRSTAREa hex_code <cr> start sending proto format hex code via IR
QSIRSTOPa hex_code <cr> stop sending proto format hex code

NetPlay CEC controls

QCECON<cr>	turn tv on
QCECOFF<cr>	turn tv off

If ON/OFF IR or rs232 codes are defined on the decoder, these are also sent by these commands

QCECSEL<cr>	select this hdmi input
QCECVOLU<cr>	Amp volume up (if supported)
QCECVOLD<cr>	Amp volume down (if supported)
QCECMUTE<cr>	Amp mute toggle (if supported)
QCECTX CMD<cr>	Send generic CEC command

NetPlay OSD engine

QSETOSD command<cr>	Command is -d duration in seconds (0 for always) -t "text to display"

NetPlay power controls

QLOWPOWx<cr>	Sets device in low power mode Normal mode restored by any switching command if x is present, only powerdown subdevice x

NetPlay Video virtual matrix command set

The virtual matrix driver is an application running on one designated device in a NetPlay Video system. This is the interface that the external control system will talk to in order to control NetPlay video. It handles:

Web page config

- Identification of encoders and decoders with input / output assignment
- Manual entry of generic sources like IP cams
- Manual entry of POP sources
- Designation of video wall groups and decoder config

Dynamic control via VM control socket

- Matrix switching of predefined sources to decoders
- IGMP multicasting setup when supported, else unicasting
- Forwarding of IR / RS232 / CEC packets to decoders
- XBMC GUI selection

NetPlay VM controls

Note: using aaa=000 means control all enabled decoders

Vaaabbb<cr> :Video output control

aaa is a number 001-999 selects the desired sink to control
bbb is a number 001-999 selects the source to switch the output to
NOTE: VTX sinks can use this to select source audio

Aaaabbb<cr> :Audio return channel output control

aaa is a number 001-999 selects the desired audio sink to control
bbb is a number 001-999 selects the input (decoder) to switch the output to
NOTE: aaa&bbb are SINK bindings on this command. To use the VTX audio outputs it must also be bound as a SINK

MACROaa b c e d f g<cr> :Macro control

aa is a number 01-99 selects the desired macro to start
Paramaters b through g are optional parameters that will be passed to the macro itself. Inside macros, any number can use "bbb" to indicate a passed parameter (3 digit in this case).
By convention, parameter b is used for sink number
NetPlayTV will populate this when sending a macro

SWaaa x bbb ccc ddd eee ….<cr> : Video output control, matrix display

aaa is a number 001-999 selects the desired output to control
x is the display layout to use
bbb, ccc, …..
are numbers 001-999 selects inputs to switch the output to
These inputs will display according to the layout number X
X layout is defined in web config

SWSaaa x bb ccc<cr> : Video output control, matrix display

aaa is a number 001-999 selects the desired output to control
x is the display layout to use
bb is the layout position subwindow to switch (01-16)
ccc is the number 001-999 selects inputs to switch the subwindow to

Gaaabbb<cr> : Audio output gain control

aaa is a number 001-999 selects the desired output to control
bbb is a number 00-255 selects the output gain
For bbb = 000 output is muted
Else gain(dB) = 31.5 – [0.5*(255-bbb)]
NOTE: on firmware versions 1.5 and up bbb may also be
bbb = U Volume steps up
bbb = D Volume steps down
bbb = MT Mute on
bbb = MF Mute false
bbb = M Mute toggle

Eaaabbb<cr> : Audio input gain control
aaa is a number 001-MAX selects the desired input zone to control
bbb is a number 000-255 selects the input gain
For bbb = 128 input is nominal
Else signal = signal * bbb/128
Maaabbb<cr> : Audio output balance control
aaa is a number 001-MAX selects the desired input zone to control
bbb is a number 000-255 selects the balance setting
For bbb = 128 input is nominal
if (bbb>128) then right_vol=right_vol*(256-bbb)/128
else left_vol=left_vol*(bbb)/128
Baaabbb<cr> : Audio bass control
aa is a number 001-MAX selects the desired input zone to control
bbb is a number 000-255 selects the input gain
For bbb = 128 input is nominal
Else signal = signal * bbb/128
Taaabbb<cr> : Audio treble control
aaa is a number 001-MAX selects the desired input zone to control
bbb is a number 000-255 selects the input gain
For bbb = 128 input is nominal
Else signal = signal * bbb/128
Xaaabbb<cr> : Audio input delay control
aaa is a number 001-MAX selects the desired input to control
bbb is a number 000-255 selects the input delay
Each delay step is 0.25 video frames (60Hz), or 4.1666ms

bbb = U Delay steps up
bbb = D Delay steps down
Daaabbb<cr> : Audio output delay control
aaa is a number 001-MAX selects the desired output to control
bbb is a number 00-255 selects the output delay
Each delay step is 0.25 video frames (60Hz), or 4.1666ms
bbb = U Delay steps up
bbb = D Delay steps down

SETCHANaaabbb<cr>: Set HDHomeRun channel
aaa is a number 001-MAX selects the desired output to control
bbb is a number 001-999 selects the channel to tune
bbb can be UP,DN,UPF,or DNF for tuning up/down (F is favorites)
(HDHomeRun source must also be selected with V/SW/etc)

STAT<cr> : Request device status   (Vaaabbb for each output)

STATL<cr> : Request device status of audio settings   (all audio settings for each input and output)

Faab<cr>: Flash modes control
aa is a number 01-99 selects the flash slot to control
b is either S or R (S saves current config, R recalls saved config)

QSINK<cr>: Request sinks status
Returns the status of all sinks in order separated by <cr>

QSRC<cr>: Request sources status
Returns the status of all sources in order separated by <cr>

QREADP<cr>: Get profile information from all sinks
Return data can take up to 1 minute
QREDOP<cr>: Regenerate source profiles on all sinks
This command takes 30seconds *sources *sinks to complete

QSGETIPxxx<cr>: Get current ip of source xxx
QDGETIPxxx<cr>: Get current ip of decoder xxx

QSETDID<cr>: Sets the OSD on each decoder to display the decoder ID

QSTATHDHRLUP<cr>: Request HDHomeRun channel lineup
Returns the channel lineup num/name separated by <cr>

xxx is the decoder or source # in all below forwarded commands. If 000 is used, it will be sent to ALL enabled decoders/sources.

All transport commands can be forwarded to decoders.

NetPlay VM XBMC controls

QXBMCONxxx<cr> turn on xbmc gui

QXBMCOFFxxx<cr> hide xbmc gui (automatic when V command used)

NetPlay VM IR controls

Send to decoders
QDIRPULSExxx R=## hex_code <cr> pulse proto format hex code via IR ## times
QDIRSTARTxxx hex_code <cr> start sending proto format hex code via IR
QDIRSTOPxxx<cr> stop sending all IR codes

Send to sources
QSIRPULSExxx R=## hex_code <cr> pulse proto format hex code via IR ## times
QSIRSTARTxxx hex_code <cr> start sending proto format hex code via IR
QSIRSTOPxxx<cr> stop sending all IR codes
QFIRPULSExxx R=## hex_code <cr> pulse proto format hex code via IR ## times to source currently selected by zone xxx

NetPlay VM Ethernet controls
URL is encoded Ethernet control url. Please see online documentation.

Send to decoders
QDETHxxx URL <cr> send URL to sink

Send to sources
QSETHxxx URL <cr> send URL to source

NetPlay VM IR/ETH database controls

Send to decoders
QDIRCODExxxyyy R=##<cr> send predefined code yyy, repeat ##

Send to sources
QSIRCODExxxyyy R=##<cr> send predefined code yyy, repeat ##
QFIRCODExxxyyy R=##<cr> send predefined code yyy, repeat ## to source currently selected by zone xxx

NetPlay Delay controls (MACROs/SCRIPTS only)

Wait for X seconds
WAIT x<cr>

Pause until given UTC time
TIME h m s<cr>

Pause until given day of week (Sunday is day 1, UTC time)
DAY d<cr>

NetPlay VM RS232 controls

QSRSxxxBASE64_ENCODED_DATA<cr>
Sends base64 encoded data to sink xxx

NetPlay VM CEC controls

QCECONxxx<cr>	turn tv on
QCECOFFxxx<cr>	turn tv off

If ON/OFF IR or rs232 codes are defined on the decoder, these are also sent by these commands

QCECSELxxx<cr>	select this hdmi input
QCECVOLUxxx<cr>	Amp volume up (if supported)
QCECVOLDxxx<cr>	Amp volume down (if supported)
QCECMUTExxx<cr>	Amp mute toggle (if supported)
QCECTXxxx CMD<cr>	Send generic CEC command

NetPlay VM OSD engine

QSETOSDxxx command<cr> xxx is sink (000 for all sinks)
Command is -d duration in seconds (0 for always)
-t "text to display"
-w "x1 y1 x2 y2" screen window to display in

NetPlay VM Video OSD engine

QVOSDxxxyyy z d o<cr> Start video OSD (PIP)
xxx is sink , yyy is source
z is format (1-8)
d is duration in seconds (0 for always)
o is overlay mode
0 (default) use Netplay or Android PIP
1 use passive overlay (if available, else 0)
2 use interactive overlay (if available, else 0)

QVOSSPxxx<cr> Stop video OSD (PIP)
xxx is sink
Using V switch command will also end Video OSD

QPIPSWAPxxx<cr> Swap current PIP and fullscreen video playback
xxx is sink

QPIPFULLxxx<cr> PIP video => fullscreen playback
xxx is sink

NetPlay VM low latency controls

QLOWLATxxx<cr> Set low latency mode
xxx is sink

QHIGHLATxxx<cr> Cancel low latency mode
xxx is sink

QSIRPULSExxx<cr> Source IR commands increment the lowlat counter. You can send this as a dummy command without hex code.

NetPlay VM profiling controls

QPROFILExxx<cr> Re-profile all inputs on output xxx
If xxx = 999, then all outputs

NetPlay VM power controls

QLOWPOW<cr> Sets all devices in low power mode
Normal mode restored by any switching command

NetPlay encoder settings

QVENCEDIDMaaax<cr> Sets edid mask on source aaa

QVENCAOUTMaaabc<cr> Sets audio output mode on source aaa

QASETDELaaaxxx<cr> Sets audio output delay on source aaa

NetPlay Audio delay device settings

QDDDEVS<cr> return audio delay device settings

QDDSETxx Y Z H<cr> Set config for audio delay device xx
Y = delay in ms
Z = 0 for analog, 1 for digital
H = 0 for 48Khz, 1 for 44.1Khz

NetPlay dynamic url settings

QSETTMPaaa url<cr>
aaa = source number URL to overwrite

Dynamic URLs are to support directly passing a URL which will be displayed. Usually this is used with an external file browser (select a file, send NVMM the URL). In your source setup, create a Video URL, Image URL, or Android APP source type. You can use the QSETTMP command to dynamically provide the URL for this source. If the URL provided ends in (.png, .jpg, or .jpeg) the source TYPE will also be changed to Image URL. If the URL provided end in (.mp4, .mkv, or .ts) the source TYPE will also be changed to Video URL. Dynamic URLs are reset by reload, reboot, or the next QSETTMP command (to that source).

NBX config files

Note that the easiest way to view and modify the config files is through the web interface. Just type the NBX/NAB IP address into a web browser to get started.

Below are examples of the configuration settings. These are not updated for each firmware revision, so it is best to use the web configuration instead!

Config:

The renderer config file is a table with 8 rows of 1 column.
Each row corresponds to a config setting, explained below.

0: Device type
- 0: NBX
- 2: CMX1616A2 (NAB)
- 3: CMX3838A1 (NAB)

1-4: Reserved
5: Logitech Media Server enable
6-7: Reserved

Renderer:

The renderer config file is a table with 48 rows of 8 columns.
Each row corresponds to a renderer, with NAB capable of publishing a maximum of 48 renderers.

Name:	Configurable name of the channel the user sees.
Enabled:	Boolean value indicating if this row will be used.
Type:	4 value Boolean bit field Bit 0: Publish this channel via Bonjour Bit 1: Publish this channel via Upnp Bit 3: Publish this channel as Squeeze Player Bit 3: Reserved
Fixed:	Boolean value indicating if this row will be assigned a fixed I2S channel If not set, I2S channel will be allocated dynamically
OutVector:	48 value Boolean bit field Each bit corresponds to an output of CMX that will be triggered when Connection is made to this renderer channel (NAB only)
Connect:	Optional RS232 string to send on connection
Disconnect:	Optional RS232 string to send on disconnection
Assigned:	Read only value which gives the currently assigned I2S channel

The format of the Connect and Disconnect strings is printf style. You can use %d argument, which will translate to the assigned I2S channel.

Server:

The server config file is a table with 48 rows of 8 columns.
Each row corresponds to a renderer, with NAB capable of publishing a maximum of 48 servers. NBX100 does not use the server file since it does not support audio input.

Name:	Configurable name of the channel the user sees.
Enabled:	Boolean value indicating if this row will be used.
Type:	4 value Boolean bit field Bit 0: Publish this channel via Bonjour Bit 1: Publish this channel via Upnp Bits 2-3: Reserved
Fixed:	Boolean value indicating if this row will be assigned a fixed I2S channel If not set, I2S channel will be allocated dynamically
InVector:	48 value Boolean bit field Each bit corresponds to an input of CMX that will be routed to the server Channel when connection is made. Only 1 bit per row can be set. (NAB only)
Connect:	Optional RS232 string to send on connection
Disconnect:	Optional RS232 string to send on disconnection
Assigned:	Read only value which gives the currently assigned I2S channel

The format of the Connect and Disconnect strings is printf style. You can use %d argument, which will translate to the assigned I2S channel.

Audio players:

Audio players generally have their own configuration files. These are not handled by this control socket. Most audio players have their own control sockets/ports and web config interfaces.