



Video Storm LLC
www.video-storm.com

NBX/NAB Control guide

Revision 1.7
8/9/13

Overview

The RS-232 output on the NBX/NAB series audio players is a flexible interface which can be used to control external devices, such as CMX series video switches. The interface acts as an Ethernet to RS232 bridge.

Hardware specifics

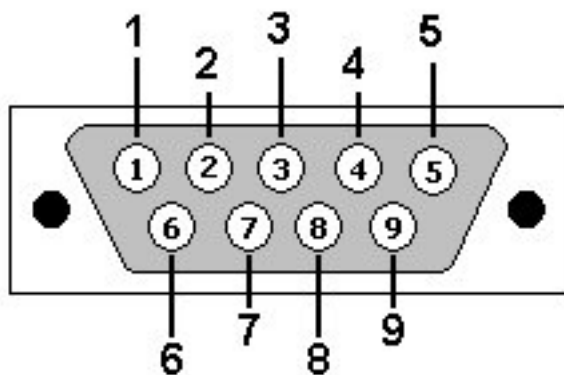
The NBX switch uses a 3.5mm stereo female connector for connection.

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.

Protocol settings

RS232 output

Baud rate : 9600

Data bits: 8

Stop bits: 1

Parity: None

Flow control: None

TCP control port

Protocol: TCP/Telnet

Port: 23

TCP secondary control port (multiple connections capable)

Protocol: TCP/Telnet

Port: 9091

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.

Commands

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

1. If the command is a valid NBX 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 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 devices. Different models can be differentiated by the controller by the status readback.

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

NBX/NAB 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
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 I2S channel information and Metadata
 a is the optional argument to return only one specific of the I2S channel
 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 I2S 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

NBX 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.

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)

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.

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)
1-3: Reserved
4: Pianobar enable
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.