

ZEN NDI Router Protocol

Using blocks with a typical length of 8+2 ASCII characters, in the form

ABxNNxNN[␣], where A & B = text characters, NN = decimal number (with leading zero), x = don't care (often blank), [␣] = Return/Linefeed (= CR + LF, i.e. hex 0d + 0a)

Router Outputs are represented by decimal numbers, even though they are shown as letters in the Router GUI, so A = 1, B = 2, etc.

Commands/Requests accepted by the TCP server (e.g. port 9779)

AB xx yy[␣] *Used to perform "AB" switching between two outputs on the router <v1.41>*

AB Switch Command affecting the sources on router outputs xx & yy (where 1=A, 2=B, etc)

e.g. If Input 5 was routed to Output C, and Input 7 was routed to Output F, then a command string of **AB 03 06** would result in Input 7 being routed to Output C, and Input 5 to Output F.

A command of **AB 00 00** will switch the inputs on the A & B outputs.

No response (except for ACK).

BR xx yy[␣] *Used to request the "base name" used for a router's NDI outputs <v1.4>*

Base-name Request, xx & yy = don't care (for the request).

Response in the form BN XX YY Text[␣], XX = Router number, YY = number of characters in the following Text string which contains the "base name" used for the NDI outputs. (e.g. "RouterOut")

CR xx yy[␣] *Used to request the I/O config of a router*

Config Request, xx & yy = don't care (for the request). Response in the form CR XX YY[␣] where XX = total inputs, YY = total outputs. E.g. CR 16x08[␣] for a 16-input, 8-output configuration.

DD xx yy[␣] *Used to request a "data dump" of multiple parameters*

Data Dump Request, xx & yy = don't care (for the request). This is equivalent to sending the following requests:

CR 00 00[␣] *(Config Request)*

RN 00 00[␣] *(Router Number) <v1.4>*

IV 00 00[␣] *(Validate Inputs)*

IN 00 00[␣] *(get Input Names)*

OR 00 00[␣] *(request Output Status)*

PV 00 00[␣] *(Validate Presets)*

PR 00 00[␣] *(Request current Preset)*

ON 00 00[␣] *(get Output Names) <v1.4>*

IN XX YY *Used to obtain the names of the NDI sources used by the input channels*

Input Name, XX = input number, YY = Option Code, 0 = full name (network name & device name), 1 = network name, 2 = device name

Response in the form IV XX YY Text, XX = input number (See Note below), YY = number of characters in the following Text string

NOTE:- XX is a decimal version of a 7-bit number, where the lower 5 bits indicate the input number, and the 2 highest bits the Option. If the requested Option is zero, then the inputs in the response will be numbered as normal, from 1 upwards. For Option 1, the return values of XX will be numbered from 33, and for Option 2 they will be numbered from 65

e.g. A response of IN 03 14 MSI (image 01) = NDI source "MSI (Image 01)" on Input 3, response of IN 37 03 MSI = NDI source with network name "MSI" on Input 4

A request with a zero input number (XX = 0) would return a list of all inputs (with the Option specified in YY)

IV XX yy *Used to validate input channels that are connected to NDI sources*

Input Validate, XX = input number, yy = don't care (for the request)

Response in the form IV XX YY, XX = input number, YY = 1 for NDI source connected, 0 for no source
e.g. response of IV 03 01 = valid NDI source on Input 3, IV 04 00 = no NDI source for Input 4

A request with a zero input number (XX = 0) would return a list of all inputs

ON XX YY *Used to obtain the output channel names (if not using the "basename") <v1.41>*

Output Name, XX = output number (1 = A, 2 = B, etc), YY = don't care

Response in the form ON XX YY Text, XX = output number, YY = number of characters in the following Text string

A request with a zero input number (XX = 0) would return a list of all output names (as multiple ON XX YY Text responses)

OR XX yy *Used to obtain the status of one (or all) outputs*

Output Request, XX = output number (from 1 to max outputs), YY = don't care

e.g. OR 03*** requests the output status of Output 3. The response will be in the form as above,
e.g. OS 03 05 for Input 5 routed to Output 3

A request with a zero output number (i.e. XX = 0) would return a status list of all outputs

e.g. OR 00***, for a config with 4x Outputs, might return

OS 01 06

OS 02 03

OS 03 00 (no input routed)

OS 04 01

OS XX YY *Used to switch output routing assignments*

Output Set, XX = output number (from 1 to max outputs), YY = input number (1 to max inputs)

e.g. OS 04 10 requests that Output 4 uses Input 10. If successful, the same string will be returned, or else the returned string will show which input is being used by that output.

A response of OS 04 05 would indicate that there was no NDI source connected to Input 10, and that Input 5 was the last selected.

A response of OS 04 00 would indicate that no input is currently routed to Output 4.

PN xx yy *Used to obtain the name assigned to the specified Preset <v1.4>*

Preset Name request, xx = Preset number, yy = don't care (for the request).

Response in the form PN XX YY Text, XX = Preset number, YY = number of characters in the following Text string which contains the name assigned to the preset.

PS XX yy *Used to select a Preset routing configuration for one or more outputs*

PreSet, XX = Preset number, yy = don't care (for the request)

Response in the form PS XX YY where XX = currently selected Preset (or 00 if no Preset is active). If a PreSet has no associated data, then the response will be the currently selected Preset, if any are active. YY = 1 if current routing exactly matches the stored Preset (*GUI button red*), or 0 if one or more of the outputs have been changed from the values in the stored Preset (*GUI button amber*).

PR xx yy *Used to request whether any Presets are currently active/selected*

Preset Request, xx = don't care, yy = don't care (for the request)

Response in the form PR XX YY where XX = currently selected Preset (or 00 if no Preset is active). YY = 1 if current routing exactly matches the stored Preset (*GUI button red*), or 0 if one or more of the outputs have been changed from the values in the stored Preset (*GUI button amber*).

PV XX yy *Used to validate Presets to check whether they have associated routing data*

Preset Validate, XX = Preset number, yy = don't care

Response in the form PV XX YY where XX = Preset number, YY = 1 (or a non-zero value) for valid data (*GUI button coloured*), 0 if no data (*GUI button grey, with no colour*)

A request with a zero input number (XX = 0) would return a list of all Presets

e.g. PV 00***, for a config with 6x Inputs, might return

PV 01 01 (*valid data - can be selected*)

PV 02 01

PV 03 01

PV 04 00 (*cannot be selected – no data*)

PV 05 00

PV 06 00

RN xx yy⚡ *Used to request the Router Number where multiple router instances are running on the same PC (e.g. 1, 2, 3, etc) <v1.4>*

Router Number request, xx & yy = don't care (for the request).

Response in the form RN XX YY⚡, XX = Router number, YY = 1 (for a Router).

Note: *Some of these Requests/Responses are only supported in v1.4.x onwards (tagged <v1.4>)*