

# **TCP/IP Router Driver**

***IP\_ROUTER.exe***

***Version 3.4.8***

Written by Paul Wilkins

# Contents

<b>Contents .....</b>	<b>2</b>
<b>1 Overview .....</b>	<b>3</b>
1.1 Description .....	3
1.2 BNCS configuration .....	3
1.3 CSI version .....	3
1.4 External control .....	3
<b>2 Driver setup.....</b>	<b>4</b>
2.1 Introduction .....	4
2.2 [IP_Router] Section .....	4
2.3 [Router_nn] Section .....	6
2.4 BNCS_Interface parameter .....	7
2.5 Optional Configuration sections .....	8
2.6 Source and Destination Offset assignments .....	10
2.7 Driver Menu bar Options .....	11
2.8 Routes.dat files .....	11
2.9 Driver start up and close down .....	11
<b>3 Resilience and redundancy .....</b>	<b>12</b>
<b>4 Driver GUI design .....</b>	<b>13</b>
<b>5 Version history.....</b>	<b>14</b>
5.1 Driver version .....	14
5.2 Document version .....	15

# 1 Overview

## 1.1 Description

This BNCS driver, IP\_Router.exe, will control routers and their system controllers that support any of the specified multilevel protocols on hardware that permits a TCP/IP connection for control communication. This driver will work with these router protocols:

- SWP-02 ( Probel General Switcher )

- SWP-08 ( Probel General Remote Control )

- N-Vision / Miranda – ( the new generation of IP based controllers )

- Quartz protocol for IP controlled Evertz routers and controllers, including the ADMX Audio routers

- Harris Logical Router Controller for Imagine Routing Switchers

- Nevion routers

- Black Magic routers

- Q.E.C Lband router protocol (limited command set)

The driver incorporates the extended protocol functionality within the latest additions for SWP-02 and SWP-08 protocols to work with the current range of Snell Series 800/850 routers along with Miranda hybrid routers which allow for router sizes up to 18000+ destinations.

Version 3.0 of the driver includes virtual routing functionality.

## 1.2 BNCS configuration

The IP\_Router.exe uses TCP/IP sessions to connect and control routers and obtains all its settings from a BNCS dev ini file, this number being passed into the application from the command line –and as a general rule this number will be the first “grd” for BNCS commands / reverts. Section 2.2 *Configuration* for the required settings within the device ini file.

There is also the option of the “-sim” entry as a second command line parameter. When used this forces the driver to run in simulation mode – and thus saves having to edit the simulation flag within the dev ini file.

## 1.3 CSI version

This version of the driver works with V3 CSI, V4 CSI and CSI32.exe.

## 1.4 External control

Version 3.4 onwards of the driver can be hooked into and controlled externally from/by Applcore clients using WM\_CopyData messaging. External control will only apply to / work with the router\_device number defined in the router configuration entry [**Router\_01**].

## 2 Driver setup

### 2.1 Introduction

When IP\_ROUTER.EXE starts up the driver requires a numeric parameter on start-up, which equates to a DEV\_XXX.INI file. When started for the very first time the application will create default entries for all required configuration parameters in the appropriate INI file. If no routers are defined within the configuration, the driver displays a message to that effect and will then halt. The driver determines the BNCS environment it is running within – either a V4/4.5 configuration with a dedicated directory structure or a classic BNCS configuration with files stored in the Windows directory or specified directories as defined within the C:\Bncs\_Config.ini file.

The ini file has 2 required sections :

- [IP\_Router]** --- Driver configuration defining communication settings
- [Router\_nn]** --- Individual router device assignments, sizes and offsets

### 2.2 [IP\_Router] Section

This section covers all the parameters, locking and database update options available for this driver.

Item	Value	Comment
DebugMode	Default: 1 0=off 1=On	If set to 1 debug messages be seen via the driver GUI and DBWIN32.exe. Default is off. Can be changed by an option on the menu bar.
TCP-IP_Address_A	0.0.0.0	IP address for the first controller within the Snell hardware.
TCP-IP_Address_B	0.0.0.0	IP address for the second controller within the hardware. Leave as default value if there is only a single controller card installed in the router.
TCP-IP_Port  TCP-IP_Port_B	Default : 2000	This is the default port for Snell/Probel routers – though value can vary according to how an engineer has set up a Snell router. For NVision routers please use port <b>5194</b> For NEVION routers please use port <b>4381</b> For Black Magic router use port <b>9990</b> For Harris LRC Imagine routers use port <b>52116</b> Can now set a separate IP Port when connecting via IP Address B, else defaults to that used for A
ALT-IP_Port ALT-IP_Port_B	Default 0	If a value is specified the driver on failing / losing connection on the main TCP-IP port, try and connect using this port. Option of a alternative port for B Address

Item	Value	Comment
Protocol_Type	SWP-02 (General Switcher)	protocol which the driver uses to control router. Other valid entries are SWP-08 or AURORA – for Aurora/General Remote NEVION – if controlling a Nevion router. N-VISION or NVISION – for Nvision/Miranda routers QUARTZ or EVERTZ – for Evertz routers/controllers EMR-ADMX – for Evertz ADMX audio router MAGIC – for Black Magic routers HARRIS – for Harris LRC Imagine routers <b>CALREC</b> – control of Calrec audio routers
Enable_Source_Zero	Default: 0 0=no 1=yes	Option to permit the use of zero as a source for protocols where 0 is not normally expected. (example – Harris LRC for use in true IP routing, hardware requires 0 to “disconnect” an IP route) [note – this option is ignored for protocols where 0 is valid already used such as EMR-AMDX and QEC ]
Comms_Error_Limit	Default: 32	This defines the number of sequential commands that the driver has not had a valid response to. If reached the driver will deem that comms to the hardware have been lost.
Simulation	Default: 0 0=Off 1=On	To run the driver in a simulation mode. Note: Any value other than 0 denotes the driver running in simulation mode. Setting this value to 1111 will initialise all destinations with source 1. Setting this value to 1234 will initialise routes src 1 to dest 1, src 2 to dest 2 etc.  Note: adding “-sim” after the device number on the command line parameter saves on having to edit this setting if wishing to quickly run a driver in simulation.
Update_Router_Names	Default: 0 0=no 1=yes	If enabled, BNCS Router modify commands for db 0 or db 1 will update names on the router hardware ( currently Aurora SWP-08 and Black Magic only )
RC_Retry_Routes	Default: 0	An option where the driver will resend Router Crosspoint commands if it does not get the expected response from the hardware to the initial RC – the crosspoint command will be resent, up to the number of retries defined. The option can be enabled / disabled from the driver GUI Menu Options during runtime.
Enable_GUI_Buttons	Default: 0 0=no 1=yes	An option to enable (1) the TAKE, LOCK and UNDO buttons on start-up of the driver.
Broadcast_Routes_on_TXRX	Default: 1 0=no 1=yes	An option to inhibit tally dump of current routes both on start up or when driver becomes TXRX. Default will tally dump routes when appropriate.

In normal running mode the driver will communicate continuously with the hardware, requesting tally status for all the destinations of any defined routers in turn. All updates are reported to the BNCS network.

NOTE regarding the "Protocol\_Type" setting when controlling CALREC Audio Routers :

Although Calrec routers use SWP-08 / Aurora protocol, it is recommended that this setting is explicitly assigned as CALREC. This is primarily because Calrec routers, when using dual controllers (i.e. 2 IP connections from driver), behave differently to how Probel's dual controllers work. All Calrec connections are considered to always be an active connection by the driver. This is important in order to maintain full control when the driver is controlling two separate Calrec cores.

## 2.3 [Router\_nn] Section

The ini file contains details to map the hardware routers onto BNCS driver numbers. Up to 8 "devices / GRDs" can be defined if there are a number of router(s) attached via a controller or router "sub-divided" into a number of smaller units. Hybrid routers consisting of a video and (often large) audio router can be configured using two router sections.

Item	Value	Comment
router_device	Valid entry between 1 and 999	BNCS router device number.
router_level	Default : 0 Quartz Default : V	SWP-08 or N-Vision protocol – enter level number. For Quartz/Evertz – enter the LETTER for the level, V,A,B Leave as 0 for SWP-02 protocol.
router_matrix	Default : 0	Only required in conjunction with routers using the SWP-08 protocol. Leave as 0 for other protocols.
max_source	Needs to be > 0	Maximum number of sources controlled by BNCS. ( includes all physical and any virtual sources )
max_destination	Needs to be > 0	Maximum number of destinations controlled by BNCS. . ( includes all physical and any virtual destinations )
offset_source	Default : 0	Leave as 0 unless the BNCS starting source is not the actual first source on a router. <b>See Section 2.6</b> on offset options
offset_destination	Default : 0	Leave as 0 unless the BNCS starting destination is not the actual first destination on a router. <b>See Section 2.6</b> on offset options
lock_infodriver	Default : 0	Only required if control of destination locks are needed to be seen by the router driver – i.e. to send actual lock commands to the hardware ( SWP-08 only ).
park_source	Default : 0	Defines a source to be used when a '0' source is sent via a RC/IW command.

Item	Value	Comment
BNCS_Interface	GRD-DRIVER or INFO-DRIVER	The default is such that this device will use a GRD mode to which standard BNCS router commands (RC, RP, RL) will be responded to.  The second option means the driver will hook externally into infodriver(s). This option is needed for routers larger than 4096. The driver will respond to IW commands. <b>See Section 2.4</b>
<i>excluded_sources</i> <i>excluded_destinations</i>	<i>Optional parameter</i> – See Section 2.5	<i>List of sources that are omitted from BNCS control</i> <i>List of dests that are omitted from BNCS control</i>
<i>inhibited_routes_dest</i>	<i>Optional parameter</i>	<i>Define per destination list of sources NOT permitted to be routed</i>
<i>virtual_routes</i>	<i>Optional parameter</i>	<i>Define the number of virtual routes to be used by router.</i>
<i>multi_routing</i>	<i>Optional parameter</i> – See Section 2.5	<i>Definition parameters for grouping sources and destinations to enable multiple / salvo routes</i>

The “offset\_source” and “offset\_destination” entries are only required when a large router is “divided” up, into for example, video and audio parts and are required to be seen as separate routers from BNCS, or when the BNCS sources and dests on a router physically begin at a value other than 1. See section 2.6 on the two methods permitted to define offsets.

## 2.4 BNCS\_Interface parameter

The specification of devices within the “Router” Section has a number of options :

- (1) up to 8 routers may be defined to work in the “GRD” BNCS interface mode – provided each of the routers are less than 4096 sources / destinations.
- (2) Or 2 routers where the first (usually the video router) uses the GRD interface and the second (often a large audio router) requiring a number of infodrivers.

Example for Option (1):

```
[Router_01]
router_device=30
router_level=1
router_matrix=0
max_source=72
max_destination=72
offset_source=0
offset_destination=0
lock_infodriver=0
park_source=0
BNCS_Interface=GRD-DRIVER
```

```
[Router_02]
router_device=42
router_level=2
```

Example for Option (2):

```
[Router_01]
router_device=020
router_level=1
router_matrix=0
max_source=684
max_destination=1152
offset_source=0
offset_destination=0
lock_infodriver=0
park_source=0
BNCS_Interface=GRD-DRIVER
```

```
[Router_02]
router_device=21
router_level=2
```

```

router_matrix=0
max_source=72
max_destination=72
offset_source=0
offset_destination=0
lock_infodriver=0
park_source=0
BNCs_Interface=GRD-DRIVER

```

```

router_matrix=0
max_source=10928
max_destination=13824
offset_source=0
offset_destination=0
lock_infodriver=0
park_source=0
BNCs_Interface=INFO-DRIVER

```

## 2.5 Optional Configuration sections

This details the optional entries that may be defined within the each of the [Router\_nn] sections of an ini file.

### EXCLUDED SOURCES AND / OR DESTINATIONS :

These are "excluded\_sources" and "excluded\_destinations". Excluded destinations will neither be polled nor cross-point commands sent to the hardware. This is best explained by way of example:

Consider a router 768 square, comprised of 3 cards of 256 where the first two cards are only half populated:

```

[Router_01]
router_device=11
router_level=0
router_matrix=0
max_source=778
max_destination=768
park_source=129
excluded_sources=129-256,384-512
excluded_destinations=129-256,385-512

```

Excluded entries are comma-delimited lists, wherein ranges can be specified by way of a '-' character. In the example above cards 1 and 2 have 128 valid sources/destinations each, so sources and destinations from 129 to 256 and 385 to 512 are all excluded.

### INHIBITED ROUTES :

Another feature is the ability to define on a per destination basis a list of sources that must never be routed to the specified destination. By way of an example for destinations 3 and 7 :

```

inhibited_routes_dest_3=9,10,11,12,13,14 - sources 9 to 14 will not be routed to destination 3 by the driver
inhibited_routes_dest_7=1,2,3,4,5

```

### VIRTUAL ROUTING :

Version 3 of the driver has added virtual routing capability. The optional parameter "virtual\_routes" can be included in the router\_nn section to define the number of virtual routes to be used by router.



By way of an example, say there are 64 virtual routes for a 512 square router, then the following will be required in the router section:

**virtual\_routes=64**

So if the router has 512 physical / real sources and destinations then the two configuration fields for max\_sources and max\_destinations needs to include all BNCS controlled sources and destinations so their values in this example will become:

**max\_sources=576**

**max\_destinations=576** -- i.e. 512 physical and 64 virtual

Virtual sources and destinations are always assigned to sources and destinations beyond the physical ones and on the driver GUI in the source/destination listboxes will be marked with a (v) indication eg 0513 (v) source/dest\_name.

Note: Since version 3.3.0, the driver requires a real source to have been already routed to a virtual destination before the equivalent virtual source can be routed to any real destinations. If the driver determines there isn't an actual real / traced source to physically routed to the specified destination it will not make any changes to existing routing.

Note: Virtual destinations, that on start-up do not have a defined routing, will be assigned the given Park source from the router initialisation data.

Note: Since version 3.4.8, when using virtuals, only the TXRX driver will send route change commands to the hardware. This is to avoid a race condition between TX and RXonly drivers which could result in the TX version losing the virtual route link for a destination.

#### **MULTI ROUTING :**

Destinations or sources may be grouped together so that multiple routes can be triggered from one single Router Crosspoint command. The data detailing any groupings uses BNCS database files – to enable RM commands to allow runtime changes to be reflected / picked up by an executing driver. Adding this entry into the [Router\_nn] section requires four parameters: a device index, source database, destination database and offset value – an example is

**multi\_routing=123,2,3,0**

The device index will, in most cases, be that of the router number, along with two specified database files that will hold the grouping data. Generally the offset value will be 0. In an aid to cover the situation where the database files for a given router are already in use for other purposes then this definition can specify a different device id or an offset value could be used e.g.: if a router is 1024 square, and all the database files are used then it may be possible to specify an offset of, say in this example, of 2000 – so that db entries from 2001 to 3024 could be used for the two specified dbs.

Database file entries for defined groups consist of a comma delimited list of indices.

e.g. 0001=1,2,3,4 will mean source or destination 1 is the master of a group containing index 1 and 2 and 3 and 4. To clear a grouping the entry can be changed to 0001=0001.

Database entry 0001=2,3,4 means the same as 0001=1,2,3,4 by the way – because the index in the db file is treated as the "master" src or dest for the defined group.

There is a maximum of 16 entries for any defined group. Any physical router index can be assigned to be a member of a group. At this time if virtual routing is also enabled then virtual indices must not be included in defined groups.

The following rules are agreed with respect to multiple routing:

Working with the example source group: 0011=11,13,15,17 dest group: 0001=1,2,3,4

1. If an RC command specifies the "master" destination as its destination then all the destinations defined in the group will be routed to and reverts generated.
  - (a) If the specified source is also the master of a group then each member of the source group will be routed to the equivalent destination member. So using our example : source 11 will be routed to dest 1, src 13 to dest 2, 15 to 3 and 17 to 4.
  - (b) If the specified source is a not a master of a group then this same source will be routed to all members of the destination group.

e.g. RC 123 12 1 will result in source 12 being routed to dests 1,2,3 and 4

2. If there are fewer sources defined in a group than destinations then the "last" source in its group will be routed to all remaining destinations:  
e.g. if source group 0011=11,13 then 11 routed to dest 1, and source 13 routed to all other destinations in its group ie 2,3 and 4, using our example.
3. Routing to a destination that is not the master one, but is defined within a group is treated as a standard route and other group members are not changed.
4. Changing a grouping definition via a RM command will not impact any routes already made. The grouping is only active at the point of an appropriate RC command.

If multi routing is enabled sources / destinations in the driver GUI list boxes will be marked with (m) for the "master" index.

## 2.6 Source and Destination Offset assignments

There are 2 methods by which source and destination offsets can be configured. Offsets are used where a router may be partitioned between video and audio for example and each partition has a separate BNCS device number.

Method (1) offsets specified in the [Router\_nn] section:

For example:     offset\_source=2000  
                  offset\_destination=2000

means that BNCS source / destination 1 equates to source / destination 2001 on the actual router. This method can be used when the same offset is to be applied to all sources or destinations in one block.

Method (2) offsets for each source and destination applied separately:

This method uses two other configuration files: dev\_xxx.sourceMapping and dev\_xxx.destinationMapping.

For example : in dev\_221.sourceMapping  
                  [Mapping]

```
0001=09217,Slot_13-AES_001-LEG_A
0002=09218,Slot_13-AES_001-LEG_B
0003=09219,Slot_13-AES_002-LEG_A
0004=09220,Slot_13-AES_002-LEG_B
0005=09221,Slot_13-AES_003-LEG_A
```

For every BNCS source and destination there will be a uniquely defined mapped source / destination, together with a description relating to the source /destination.

In this example the BNCS source 1 equates to source 9217 within the router.

This method is useful for non-contiguous mapping. Method 2 can only be used if the "offset" values within the [Router\_*nn*] section are both set to 0.

## 2.7 Driver Menu bar Options

There are options to show / inhibit debug messages from the driver giu, force the TXRX status of a driver, suspend / resume cyclic polling of destinations on a router, request or poll for data from router.

Evertz/Quartz and Harris protocols have the capability to receive multiple crosspoint commands grouped into a single command. This driver, as default, uses an inbuilt queuing mechanism to bundle multiple incoming BNCS "RC" commands and package them appropriately for these particular protocols. There is an option on the Menu Options list to enable or disable this functionality.

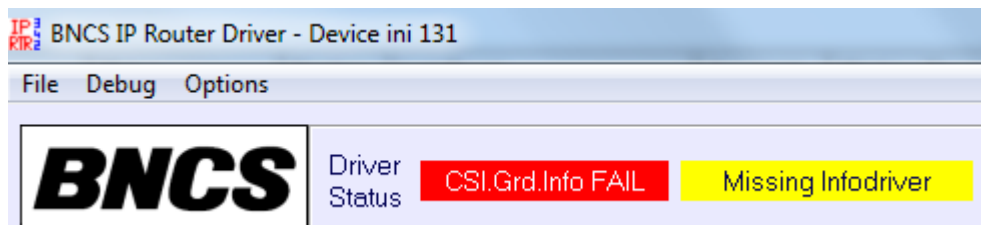
## 2.8 Routes.dat files

The driver will periodically, and on shutdown, save all known routes for each of the defined router(s) into a text based file within the "data" directory of the BNCS system. Filenames use the format of dev\_*nnn*\_routes.dat where *nnn* is the BNCS device index of the router.

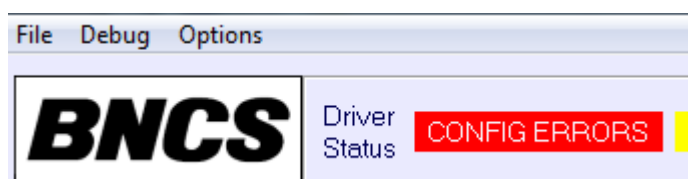
The driver on start-up will look for the existence of such files for the relevant BNCS device numbers and read in any stored routes. These will be read in prior to any initial poll of the hardware for current routing.

## 2.9 Driver start up and close down

If the driver, when configured to use any external infodrivers (via "BNCS\_Interface" setting), fails to hook into any of the infodrivers (e.g. because one of them is not running) – the driver will set itself into RXONLY and remain in a suspended state and not fully running:



Similarly the driver will be set into a suspended state if the required configuration is missing:



The driver, on closing down, will now send "close" messages to any external infodrivers that it is connected to.

## 3 Resilience and redundancy

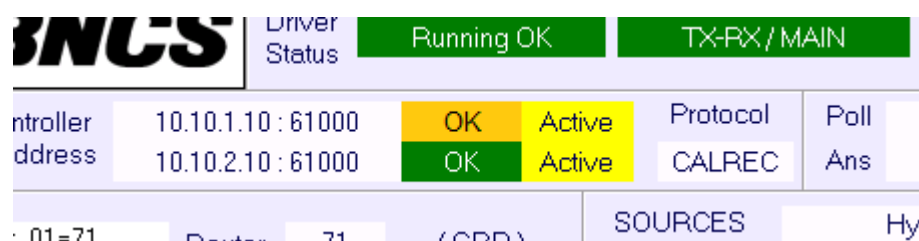
This driver will run in dual driver redundancy mode – where the first driver to run up will become the main driver in TX/RX mode and any other instances will run in RX-only mode.

Should the main driver shutdown cleanly, the reserve one will then become TX RX and run as the "main" driver.

Also the TXRX driver will revert to RX-Only if all the active TCP/IP connection(s) are broken to the hardware. The driver will attempt to return to TX-RX when the connection is later restored.

There is an option in the driver menu to force a driver into a required mode. To force a pair of drivers to fail over – force the current TXRX driver into RX-ONLY – and this will cause the other driver instance to immediately take control and become the TXRX driver.

NOTE on Comms OK field on driver GUI :

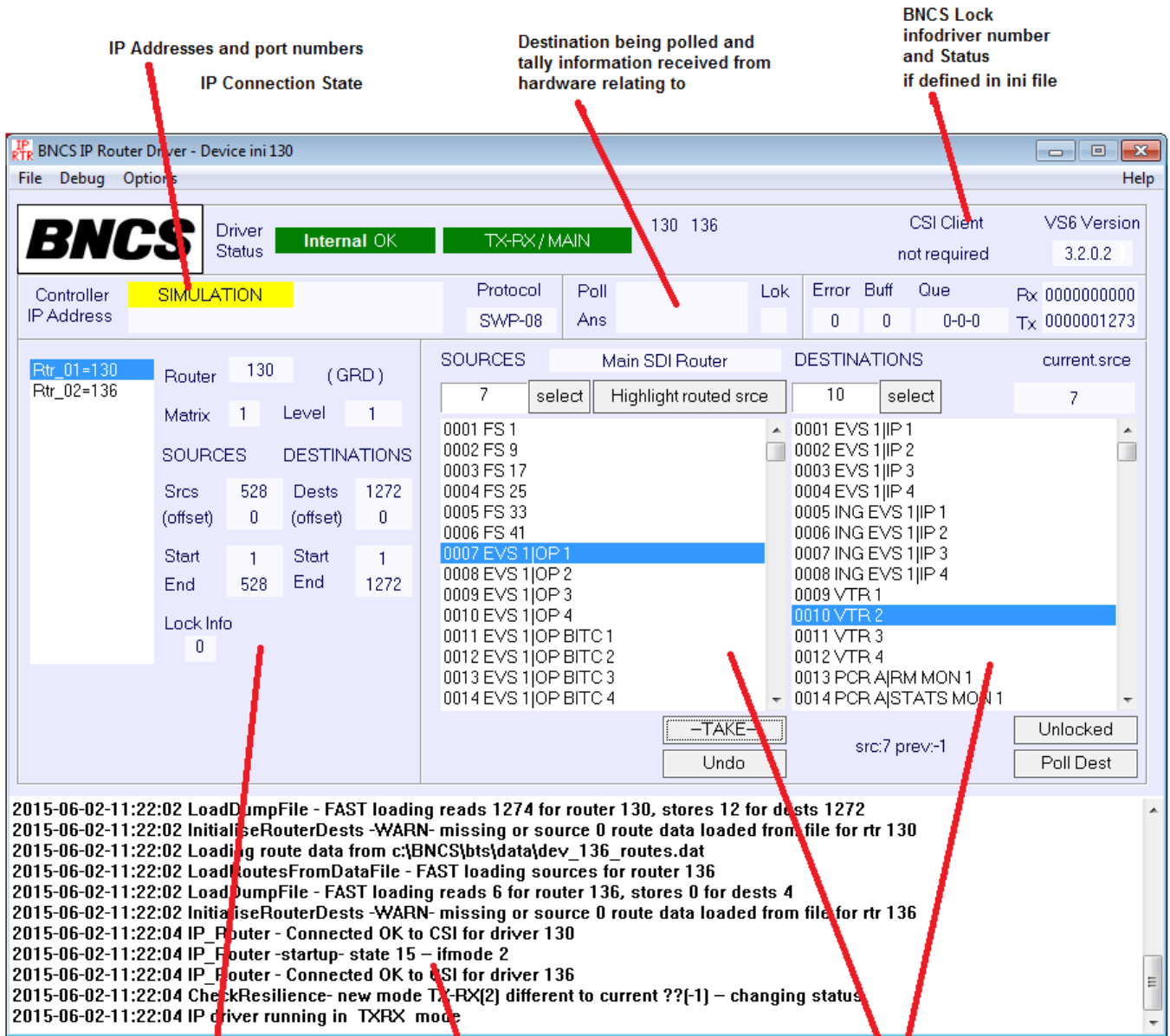


Green means OK – valid connection and appropriate responses are received – all is good.

An Orange coloured field means a valid TCP/IP connection exists but there are missing responses to commands sent – it is an indication of a comms / protocol / router hardware issue. Usually the error counters on the driver GUI will be incrementing too. An orange "Open" with a "?" in the adjacent (yellow) field indicates no response have been received at all from the hardware after a connection has been first made.

Red / FAIL – means no TCP/IP connection and no comms are possible.

## 4 Driver GUI design



There is also the option to enable/disable debug messages from the menu bar – any messages can then be seen by using DBWIN32 (or any Windows Debug message viewer).

Menu options provide to force a driver failover, to poll controller or destination status.

## 5 Version history

### 5.1 Driver version

Version	Date	Notes	Modified By
1.0.0	29/11/12	Initial Release for testing and deployment	Paul Wilkins
1.5.0	25/06/13	Revised SWP-08 message handling and driver will work with N-Vision / Miranda IP Controlled hardware	P.W.
1.5.2	13/09/13	Added definition and use of a defined park source	P.W.
1.6.6	13/02/14	Refinements for 422 Router processing and TX resilience	P.W.
1.7.0	06/03/14	Refinements for GRD and INFO-DRIVER defined routers for Hybrid routing hardware.	P.W.
2.0.0	04/04/14	Added Quartz/Evertz and Black Magic protocols Auto save of routes to database file, load routes from file at start-up, prior to initial router poll.	P.W.
2.1.2	26/08/14	Updates for Black Magic protocol including ability to update source and destination names on hardware Driver GUI enhancements	P.W.
3.0.0	13/10/14	Virtual Routing capability added to driver	P.W.
3.1.0	04/11/14	Fix for parsing SWP-08 dual controllers where old style status message is not in line with protocol documentation	P.W.
3.1.2	28/11/14	Reworked parsing of SWP-02 and SWP-08 Probel Controller responses to handle contradictory implementations by Snell between older and latest controller cards /firmware	P.W.
3.1.3	1/12/14	Multiple routing option added – principally for the Quartz Protocol to make use of their Multi route command. Can be used to send a list of route commands to other the other types of hardware supported	P.W.
3.1.6	1/7/2015	Added limited command set for QEC L-band router. Internal re-design and memory usage improvement. Additional gui enhancements relating to groups. Optional “-sim” command line parameter	P.W.
3.2.0	11/12/2015	Added Harris Protocol for Imagine Switchers. Added mechanism to bundle multiple incoming BNCS RC commands into a single command for Evertz/Quartz, Harris and Probel Aurora protocols.	P.W.
3.3.0	21/03/2016	Revised internal working for determining active / inactive controllers for all protocols. Revised internal working when virtuals are in use. Virtual sources will only route to real destinations if a real/traced source can be determined via the relevant virtual destination(s).	P.W.
3.3.1	06/04/2016	Fix for virtual sources not always clearing previously routed destinations.	P.W.
3.3.2	16/05/2016	Router Crosspoint retry command option	P.W.
3.3.3	28/06/2016	Improved Probel “active” Controller determination Fix for Aurora unlocking destinations	P.W.
3.4.0	01/07/2016	EXTERNAL control ability added to driver ( uses WM_CopyData messaging )	P.W.

3.4.2	24/07/2016	Refinements in comms for CALREC Audio routers	P.W.
3.4.3	17/07/2016	External connections fix to close when driver does	P.W.
3.4.4	21/09/2016	Additions for TXRX / Rxonly failover when driver uses infodrivers. Closes external infodrivers on closedown. Fix for excessive revertives when using vitruals. Option to enable Take, Lock, Undo buttons on start-up. Length of fast-router-poll influenced by router size	P.W.
3.4.5	13/10/16	Fix for the broadcast of zero / invalid routes after start-up if there has been comms problems/failure. Option to inhibit tally dump of current routes on start or when going TXRX	P.W.
3.4.5.556	01/12/2016	Fix for swapped matrix and level parameters in some in SWP-08 commands	P.W.
3.4.6	14/12/2016	Increased internal buffer for larger Black Magic routers (288 square)	P.W.
3.4.7	16/02/2017	Option to use zero as source for use with protocols when working with true IP hardware	P.W.
3.4.8	08/03/2017	Fix for virtual routes being lost when running a resilient pair of router drivers. Fix for polling defined excluded destinations.	P.W.

## 5.2 Document version

Version	Date	Details	Name
1.00	29/11/12	First draft of driver docs	Paul Wilkins
1.1	07/12/12	Additional sections for excluded sources/destinations	P.W.
1.8	13/10/14	Updated docs for virtual routing addition	P.W.
2.0	01/12/14	Updated documentation for multi routing option	P.W.
2.1	01/07/15	Updated docs for version 3.1.6 of driver	P.W.
2.2	11/12/15	Updated docs for version 3.2.0 of driver	P.W.
2.3	21/03/16	Updated driver history for version 3.3.0 of driver	P.W.
2.4	06/04/16	Driver version up to 3.3.1 – fix for virtual routing	P.W.
2.5	16/05/16	Driver version up to 3.3.2 – RC retry option	P.W.
2.6	28/06/16	Driver version up to 3.3.3 – option to resend RCs Fix for Aurora unlocking destinations	P.W.
2.7	01/07/16	Driver may be controlled externally from Applcore clients.	P.W.
2.8	24/07/16	Refinements in comms for CALREC Audio routers Additional documentation to comms state and use of colours on driver GUI	P.W.
2.9	21/09/16	Driver version 3.4.4 – fix for txrx failover when using infodrivers, revertives for virtuals, option for buttons.	P.W.
3.1	08/03/17	Driver version 3.4.8 docs updated for version changes	P.W.