BNCS Device Drivers

Atos IT Solutions and Services

TCP/IP Router Driver

IP_ROUTER.exe

Version 3.7.2

Written by Paul Wilkins

BNCS

Contents

BNCS

1 Overview

1.1 Description

This BNCS driver, IP_Router.exe, will control routers and their system controllers that support the 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 NP0016 – generic TCP/IP based Miranda controllers
GV NP0017 / NV9000 for GV Convergent IP (device+level) routing
NP0017 port based SDI routing using a variant of the GV protocol
Quartz protocol for IP controlled Evertz routers and controllers,
including the ADMX Audio routers
Harris Logical Router Controller for Imagine Routing Switchers and LRC+ for levels
Nevion routers
Black Magic routers
Q.E.C Lband router protocol (limited command set)
DracoTera – KVM Switch
Lightware DVI Routers

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 32000+ destinations.

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. The device ini file index can either be a number or an instance name passed into the application from the command line. If an instance is passed in then the device index is determined from finding the instance name in instances.xml. As a general rule this device index number will be the first/main router for BNCS commands / revertives. 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 and this sets the driver to run in simulation mode (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, CSI32 and BNCS_CSI.exe.

1.4 External control

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**].

BNCS

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 2=On and Show all msgs	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. A value of 2 will also set show all debug messages from start. As this will slow driver considerably please only use this option for start-up debugging.
LogToFile	0 = off 1 = on	When enabled the driver will write its debug messages to a dated file in the BNCS logs directory.
TCP-IP_Address_A	0.0.0.0	IP address for the first controller within 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.

Item	Value	Comment
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 GV / NVision routers use port **9194** or 9193 For NEVION routers please use port **4381** For Black Magic router use port **9990** For Harris LRC Imagine routers use port **52116** For Lightware DVI routers use port **10001** 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
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 NP0016 routers NV9000 or NV-EECP - GV NP0017 Convergent devices NV17 or NVPORTS - GV NP0017 port routing QUARTZ or EVERTZ – for Evertz routers/controllers EMR-ADMX - for Evertz ADMX audio router MAGIC – for Black Magic routers HARRIS – for standard Harris LRC Imagine routers See **"Use_Harris_Levels=1**" for LRC+ levels routing CALREC - control of Calrec audio routers, uses a reduced SWP-08 command set DRACO - Draco Tera KVM Switch LIGHTWARE or LIGHT – Lightware DVI routers
Use_Harris_Names	0=no 0=no 1=yes	Optional entry for when HARRIS is the defined protocol to specify that Harris hardware router names are used rather than integer indices when sending cross-points commands from the driver. Set entry to 1 to enable.
Use_Harris_Levels	**0=no 1=yes**	**Optional entry for HARRIS protocol – when enabled the defined router "level" will be used in routing commands to hardware- useful for multilevel Harris routers, and levels routing for LRC+ / 2110 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 or Probel /SAM SDNO for use in true IP routing, hardware which requires 0 to "disconnect" an IP route) [note – this option defaults to 1, for protocols where 0 is a valid index, such as EMR-AMDX and QEC]

Item	Value	Comment
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 or instance name on the command line parameter saves on having to edit this setting if wishing to quickly run a driver in simulation.
Update_Router_Names Update_Names_Length	0=no 1=yes 12 or 10	If enabled, BNCS Router modify commands for db 0 or db 1 will update names on the router hardware 12 character name lengths for Probel Aurora SWP-08. 10 character names for Quartz/Evertz routers. Entire name can be updated for Black Magic hardware.
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 retires defined. The option can be enabled / disabled from the driver GUI Menu Options during runtime.
RX_Driver_Send_Crosspoints	Default: 1 0=no 1=yes	An option when a driver is running in RXONLY mode to enable/inhibit the sending of cross-point commands to the hardware. Can be altered during run-time from an entry GUI/Menu/Options.
Enable_GUI_Buttons	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.
Infodriver_Divider	Default 4096	An option for large routers using multiple infodrivers to define the number of slots used for destinations. Example – router of 8000 destinations, if this set to 4000, then 2 infodrivers using slots 1-4000 in each rather than 4096 in first and 3904 slots of the second

Item	Value	Comment
Enable_Background_Polling	1 enabled 0 disabled	Option to enable or inhibit the background cyclic polling of destinations. Unusual to inhibit though.
Enable_Router_Locks_Polling	1 enabled 0 disabled	Option to enable or inhibit the background polling of destination locks. Only applicable for SWP-08 (Aurora), Quartz and Magic protocol settings.
Extras_Infodriver		An option to define and have the driver write to an infodriver for source PTI, for comms status and controller status. If defined, then these "Extras" settings can also be assigned. (Really only applicable only for routers under 4000 in size as only a single extras infodriver can be defined) For DRACO please read notes below
Extras_CommsFlag_Slot	Default 4091	Driver uses the defined slot for the status of TCP-IP connection 1 and slot+1 for the status of TCP-IP connection 2. State values are now: 0 = not connected 1 = connection made, but no response received 2 = comms ok, connected and responses as expected For DRACO or NV9000 please read notes below
Extras_ControllerState_Slot	Default 4095	Only TXRX driver writes controller states. If dual controllers are present, then slot+1 is used for the second controller status. Status values: 0=Uncertain, 1=Inactive, 2=Active (Not applicable for NV9000)
Extras_PTI_Slot_Offset	Default 0	Set value to -1 to disable PTI reporting if not required when Extras Infodriver is assigned. Set at zero to commence PTI from slot 1, Set at 1000 to commence PTI from slot 1001 – get the idea.

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. 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.

NOTE regarding the **LIGHTWARE** Protocol type:

Lightware DVI routers do not have destination locks but do have the functionality to MUTE / UNMUTE destination audio. The "lock" button and functions within the driver are used to enable MUTE/UNMUTE from the driver GUI. The lock button is relabelled to state MUTED or Unmuted according to the reported state from the hardware.

NOTE regarding the Protocol_Type setting **DRACO**:

Draco Tera KVM Switch protocol requires by default an external infodriver to hook into. This infodriver will be the same device number as passed into as the start-up parameter (and be the "router_device" defined in the [Router_01] section). The setting "BNCS_Interface" will be set to INFO-DRIVER. Also in this router section the offset source and offset destination values used by the driver will be 1000 and 3000 respectively.

Source 0 in this protocol is defined to mean the "Local Connection".

This protocol type also uses the main external infodriver as its Extras_Infodriver by default, with the PTI offset assigned to slot 1000 onwards. Comms status will be reported from slot 4091. These slot values are fixed and does *not use* the "Extras slot" assignments available for other protocols. Comms states are

0 = not connected
1 = connection made, but no response received
2 = comms ok, connected and responses as expected

NOTE regarding **GV NP0017 (aka NV9000)** and **HARRIS LRC Levels** protocol type:

See section 2.9 for the discussion on how the driver handles routing commands and BNCS revertives for these protocols as they work with a device **and** level notation. These protocols require the use of external infodriver(s) for the driver to hook into. This main infodriver will be the same device number as passed into as the start-up parameter (and be the router_device defined in the [Router_01] section). The setting "BNCS_Interface" will be set to INFO-DRIVER.

These protocols can use the main / first external infodriver by default, to report the TCP-IP connection status reported from slot 4091 (and slot 4092 if a 2nd connection is defined). Comms states are

0 = not connected
1 = connection made, but no response received
2 = comms ok, connected and responses as expected

"Extras_ErrorMsg_Slot" is 4093**,** by default, and will report any Routing Error responses returned from GV.

If Source port PTI is required, then an "Extras Infodriver" value needs to be assigned to be different than the main router infodriver and that infodriver will be used to report source pti data *and comms states*.

There is a variant of NP0017 for Ports based commands and responses. GV SDI routers can use this simpler port based variant and config defined router levels. Assign the Protocol setting to NVPORTS in the [IP_Router] section. This ports-based variant can use the BNCS GRD interface style like most other protocols. This ports-based variant does not require all the extra Nvision config discussed in section 2.9 which is ONLY for the NV9000 Convergent device and level form of the protocol.

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 > 0. 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 with routers using the SWP-08 protocol where the value must be greater than 0. **Leave as 0 for all other protocols**.
Max_source	Needs to be > 0	Maximum number of sources controlled by BNCS. (to include all physical **and** any virtual sources)
max_destination	Needs to be > 0	Maximum number of destinations controlled by BNCS. (to includes all physical and any virtual dests)
offset_source	Default : 0	Leave as 0 unless the BNCS starting source is not the actual first source on a router. **See Section 2.6** on offset options
offset_destination	Default : 0	Leave as 0 unless the BNCS starting destination is not the actual first destination on a router. **See Section 2.6** 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. If assigned, this source will be routed in place of 0, even for routers where 0 is a valid source. Leave unassigned if 0 is really required.
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.

Item	Value	Comment
excluded_sources *excluded_destinations*	*Optional param* *See Section 2.5*	*List of sources that are omitted from BNCS control* *List of dests that are omitted from BNCS control*
inhibited_routes_dest	*Optional parameter*	*Define per destination list of sources NOT permitted to be routed*
virtual_routes	*Optional param*	*Define the number of virtual routes for router.*
multi_routing	*Optional param* *See Section 2.5*	*Definition parameters for grouping sources and destinations to enable multiple / salvo routes*
Metadata_Source_DB *Metadata_Destination_DB*	*Databases indices (greater than 1)*	*Database indices to provide additional data relating to sources and destinations on driver GUI. Expected that database values are greater than 1. Ignored if left at 0*
AVFormat_Source_DB *AVFormat_Destination_DB*	*Databases indices (greater than 3)*	*Database indices to provide VIDEO FORMAT lists for sources and destinations. Applicable only for GVC NP17 and Harris LRC levels protocols only.* *Data from these only used if entry in [IP_ROUTER]* *NP17: NVision_AVFormat_Checks=1* *LRC: Harris_AVFormat_Checks=1* *Expected that database values are greater than 3.*
special_mode	*Optional param* *STEREO* *MAGNUM_VA*	*Can define if router behaves as STEREO router.* *On acting as such, ODD and EVEN sources and destinations are grouped into pairs, with the odd index deemed the master and the subsequent even index its slave. As they are grouped, routing can only occur of the master, that is ODD routes. The driver inhibits routing if either source or destination is an EVEN index.* *Magnum entry is a special case when connecting to a Magnum Controller that has two identical routers defined, one on the V Level (main) and the other on the A level (reserve). Any BNCS Route commands for either defined router will go to both levels on the hardware – the intention is that both routers are always identical and in sync with each other.* *When this mode is used, 2 identical routers for sources and destinations and using router levels V [Router_01] and as level A [Router_02] need setting up in the driver configuration. (requires 2 BNCS device numbers)* *A single BNCS route command (for either driver id) will send a two level route command to the hardware. Two revertives should be returned and issued out from the driver, one for each level.*

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
Metadata_Source_DB=4
Metadata_Destination_DB=5
AVFormat_Source_DB=0
AVFormat_Destination_DB=0
```

```
[Router_02]
router_device=42
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
Metadata_Source_DB=0
Metadata_Destination_DB=0
AVFormat_Source_DB=0
AVFormat_Destination_DB=0
```

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=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
```

VITRUAL 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.

When virtuals are in use and a resilient pair of drivers are running, the RXOnly driver will poll for and respond to BNCS router revertives from it's TXRX counterpart. This is especially important on starting up when virtuals are in use, as significant routing changes may have occurred since the driver was last run, and loading historical saved routes from file is not enough to build the correct and current routing.

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 32 entries for any defined group. Any physical router index can be assigned to be a member of a group. If virtual routing is also enabled then virtual indices can also now be included in defined groups – driver version 3.6.3 onwards.

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 revertives 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.

Note: If the router is optionally defined as a "STEREO" router, sources and destinations are at start-up assigned into paired groups with ODD sources and destinations set as the "master" and the following even index a slave wthin the group. When a router has been set as STEREO then it is not required to set up multi-group records in database files as discussed, - in fact if they are defined they will be skipped.
When defined as a STEREO router, only routing the "master" aka ODD index will be permitted as dual routes. Route commands containing EVEN source or destination indices are ignored.

Note: Writing '-5010' into the mask for an RC command when routing to a multigroup master destination will mean the source index will be solely routed to that destination index alone. (This provides an option to override the multigroup definition for the master destination should it be necessary to route only to that destination for some reason rather than the whole group).

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 and Snapshot.dat files

The driver will periodically, and on shutdown, save all known routes for each of the defined router(s) into a text 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.

Harris Hardware Router Names: When working with Harris Routers – the hardware router source and destination names are used for commands and revertives. The names retrieved from the hardware are stored in text files named dev_*nnn*_HarrisNames.dat. If present these are read in at start up too. Names are requested from the hardware on driver connection and in the periodic cyclic poll too during normal running.

2.8.1 Snapshot.dat files

Two options on the Driver GUI menu provide for a Snapshot save of current routes to a file in the "data" directory for the BNCS system called format of dev_*nnn*_snapshot.dat where *nnn* is the BNCS device index of the router. The routes stored in this file can also then be recalled and routes reasserted to the hardware by the driver. This option is provided to be used as a specific manual task by a user as a means of restoring / reasserting previously saved routes for whatever reason. Snapshots are distinct and different to the routes.dat files above.

All these files of routes / snapshots are text based for readability and easy manipulation if required.

2.9 GV Convergent and Imagine LRC Levels use for 2110/IP Routing

These protocols for router levels routing use a two-part notation to define <device>.<level> for source and destination indices, so routes can be made not just for a source to a destination device but also at a finer granularity of individual levels for a specified destination device. Commands and revertives are pipe delimited strings and by necessity use the BNCS Infodriver mechanism for these two.

There are additional configuration data sections required in the BNCS device ini or database files to specify the GV/Harris defined levels associated for each source and destination device.

Required Levels Configuration

It is suggested that device levels use, at most, a 3-digit numbering in order to fit BNCS commands and revertives for up to 18 levels into the 256-character limitation of infodriver slot data. There are two configuration parts for levels definition required by the driver.

(a) Section listing for the known levels defined within the GV Convergent / Imagine systems:

For example, GV levels could be drawn from these indices:
2 or 200 for HD Video,
3 or 210 for single raster UHD video
20 or 400 for Ancillary metadata
4–19 or 301-316 for 16 channel audio

For example, Harris LRC+ levels could be
1 for 2022-6 Signal
2 for Combined 2110 (HD)
3 for Combined 2110 (3G)
4 for Combined 2110 (UHD)
5 for 2110 Uncompressed Video
6 for 2110 Compressed Video

[NV9000_Levels] – each entry lists the GV level index and a short description

```
0001=200,Video HD
0002=301,Audio A Ch1
0003=302,Audio A Ch2
0004=400,Ancillary Data
```

[HarrisLRC_Levels] – each entry lists the Harris level index and a short description

```
0001=1,2022-6
0002=2,2110 HD
0003=3,2110 3G
0004=4,2110 UHD
```

(b) Section heading or specified BNCS database (eg 2 and 3) which contain a comma delimited list of the levels expected for *each* source and destination device. The levels given will align with the indices of the entries from the NV9000 or Harris_LRC Levels section described in (a). It is recommended to use BNCS database files for this data as it can then be modified during run-time via RM commands.

[*NV9000*_Source_Levels] [*NV9000*_Destination_Levels]

–list of levels per source device - for example:
0001=200,400, 301-316
0002=200,4000, 301,302,303,304,305,306,307,308-316

[*Harris_LRC*_Source_Levels] [*Harris_LRC*_ Destination _Levels]

–list of levels per source device - for example:
0001=1,2,3,4
0002=1,2,21,22

Note: comma delimited list, but a dash '-' may be used to define a contiguous range of levels, thus shortening the text definition when listing many levels.

Note: the first level defined in these lists becomes the "primary" level and is the one shown on the main driver GUI when selecting sources / destinations in the main GUI source and destination list boxes. As part of the driver GUI there is a NV9000 Menu option available when using this protocol to show an extra GUI panel detailing all the level routing for any chosen source and destination devices on the main GUI.

Note: in the [IP_Router] section, database files can be assigned to hold these source and destination levels data rather than the default sections just described. If assigned the designated databases will be read initially at start, if not assigned then the driver will look for level definitions in the two sections listed above. Defining them in device databases provides the added advantage that levels expected / used can be altered using Router Modify commands during run-time.

[IP_Router]

For GV/NV9000 :
NVision_Source_Levels=2
NVision_Destination_Levels=3

For Harris_LRC **:**
HarrisLRC_Source_Levels=2
HarrisLRC_Destination_Levels=3

If a source or destination has no levels defined it will be treated by the driver as being out of commission, at that time, and will not be routable. The driver GUI will flag such indices as "Excluded". Destinations without levels will report '0' as its source.

Optional AV Format Databases and Route Checking

Optional Databases can be assigned to hold valid Video format level for destinations and sources:

[Router_01]
AVFormat_Source_DB=6
AVFormat_Destination_DB=7

The data entries can be a single level or comma delimited list of acceptable video LEVELS per source and destination, or a '' to indicate all video format levels. The databases can be ammended at runtime using RM commands.*

Associated to these entries is the [IP_ROUTER] section entry

***NVision_AVFormat_Checks**=1 or **Harris_ AVFormat_Checks**=1.*

This ini file entry will activate (1) or deactive (0) AV Format checking before all routing commands are processed. This can also be enabled / disabled from the GUI from the NV9000/Harris menu.

Failed routes are reported to the Extras_ErrorMsg_Slot of the Extras_Infodriver.

Device Levels BNCS Routing Commands and Revertives

Using the infodriver hooked into by the application, string commands initiate routes to be made at both device and level granularity and detailed responses list the current level routing returned from the hardware. Each infodriver slot represents a destination device.

(a) Routing Commands:

(1) Single integer value – means route the given source device to the destination device index, in its entirety. NVision hardware will route all possible source levels to their equivalent destination levels.

(2) Complex string entry in form

<destination level>**:**<source device>**.**<source level>

Pipe delimited list for multiple entries of level routing in one command.

The entry is parsed to route the given source device and level to the specified destination level for the destination device slot index. As infodriver strings are limited to 255 characters, entering details for many levels could break this limit, the following variations of the defined entry form can be used to reduce the character count:

<destination level>**:**<source device>

omitting a specific source level means the source level is taken to be the same as that of the destination level.

<source device>**.**<source level>

Omitting the <destination level>: will mean this specified source device and level is routed to ALL the levels for that destination device – granted this is an unusual command request but possible if say a device is just all audio of one format.

When using the complex form, the only routes made will be those listed. Thus, individual shuffling or breakaways for specific audio levels can be achieved by just writing those required entries into the infodriver slot.

Examples of valid routing commands:

"4" – single source device index – driver sends a "Take All" command to hardware, such that the levels for device 4 are routed to same levels for destination device.

"4|301:6|302:6 -- routes source 4 to all levels for destination device and *then* will route source 6 level 301 to destination level 301 and then source 6 level 302 to destination level 302 (example of a breakaway)

"200:6|301:6.311|302:6.312|303:6.313 – shuffling example

Note: Automatics/panels need to include the Ancillary level if it is to follow a routing for the given video level.

The driver will only route levels that it is told to route. This can be useful if an application just wants to send a subset of levels for a specific breakaway.

Driver 3.7.2 – for BT Sport anf GV additional route command:

NVision GV Convergent / Orbit may allow for routing to use the Nvision "302A" routing command which the driver can use to batch up routing commands to the hardware for efficiencies in GV processing.

Set the entry **NVision_302A_Routing=1** to enable this option, but ONLY do so if your GV installation supports this 302A command.

(b) Routing Revertives:

Revertives are always given as a pipe delimited list using the complex form of

<destination level>**:**<source device>**.**<source level> for all the destination levels.

The contraction of the form when the source level matches that of the destination level is extensively used to keep the character count down.

Examples: infodriver slot 4 contains

200:9|400:9|300:9|301:9|302:9|303:9|304:9|305:9|306:9|307:9|

308:9|309:9|310:9|311:9|312:9|313:9|314:9|315:9|316:9

Means that source device 9 is routed to each of the destination levels – where the source level *equates* to that of the destination level.

200:2|400:2|300:2|301:**2.302**|302:**2.301**|303:2|304:2|305:2|306:2|

307:2|308:2|309:2|310:2|311:2|312:2|313:2|314:2|315:2|316:2

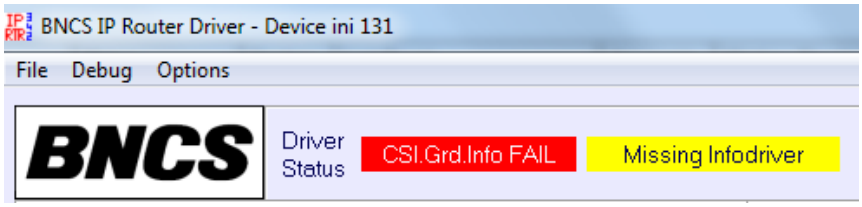
Means source 2 is routed to all levels, however levels 301 and 302 are shuffled using a different source level to that of the destination level

200:2|400:2|300:2|301:**33.301**|302:**66.302**|303:2|304:2|305:2|306:2|
307:2|308:2|309:2|310:2|311:2|312:2|313:2|314:2|315:2|316:2

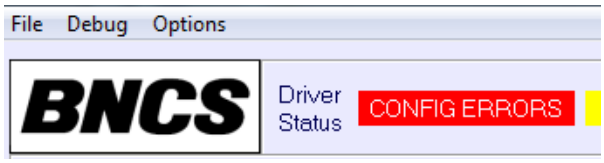
Means levels 301 and 302 have breakaways with audio contributions from different devices and levels.

2.10 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

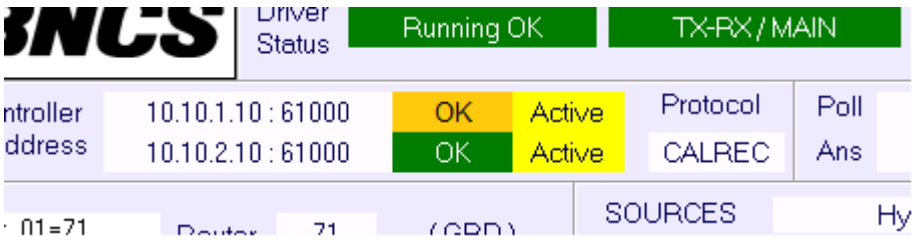
This driver will run in dual driver redundancy mode – where the first driver to run up will become the main driver in TX/RX mode and any other instances will run in RX-only mode.

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 :

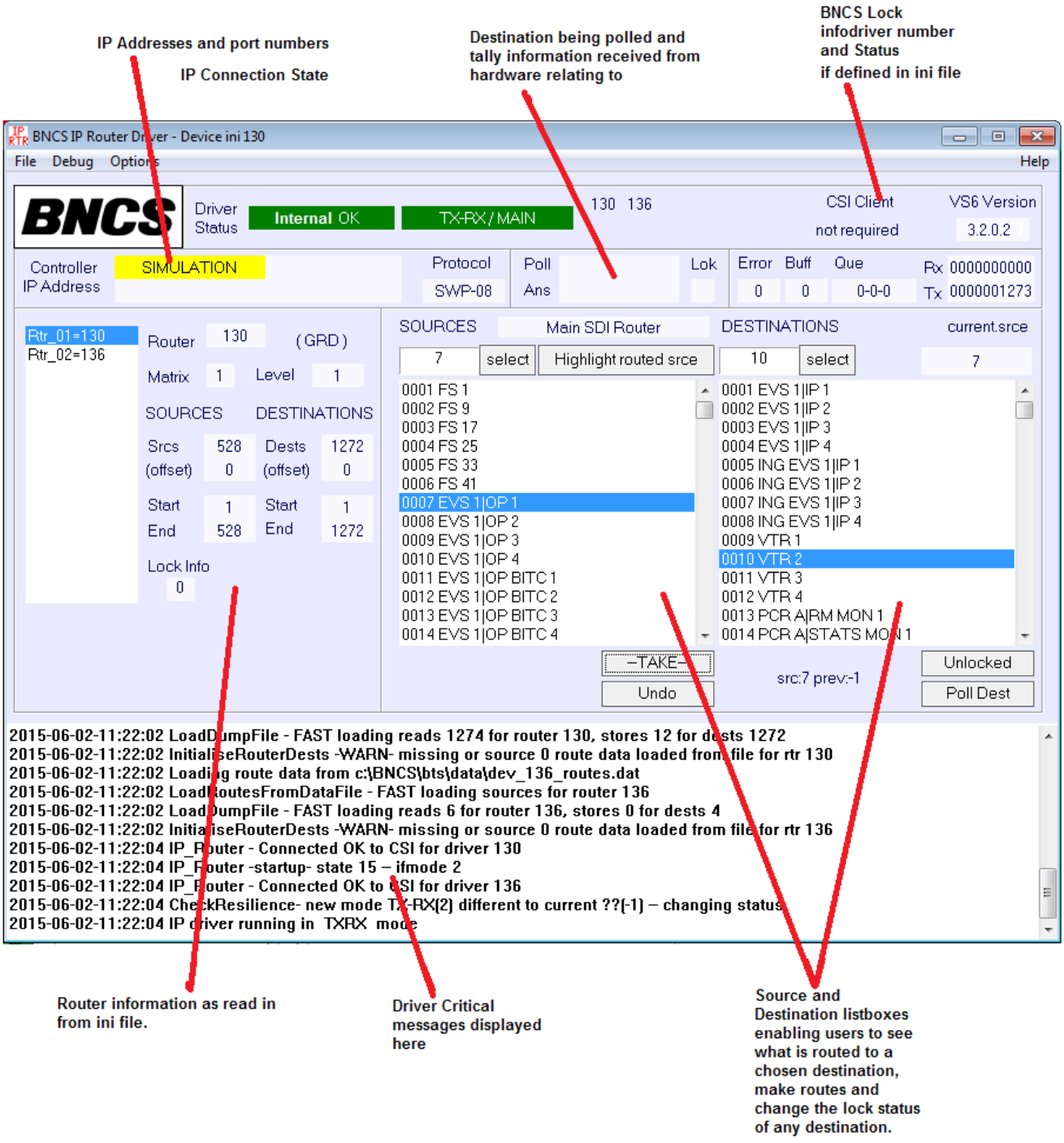


Green means OK – valid connection and appropriate responses are received – all is good.

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 are options to enable/disable debug messages, log debug messages to files in the defined BNCS Log's directory or debug messages can then be sent to DbgView/ DBWIN32 (Windows Debug message viewers).

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

Menu options for the Draco and NV9000 protocols, enabled as and when required.

BNCS

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.
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 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 revs when using virtuals. 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 sq)	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.

3.4.9	22/03/2017 27/04/2017 16/06/2017	Extension of the use of Enable Source_Zero for Probel protocols where SAM IP routing hardware returns 65535 as source when "disconnected" for true IP route. Driver converts to 0 and passes revertive as such. Optional configuration entry when using HARRIS protocol to use router hardware names rather than indices when sending cross point commands. Addition that when virtuals have been defined, the RXOnly driver will poll for at start-up and respond to revertives from the TXRX driver – to maintain accuracy of virtual routes.	P.W.
3.5.0	26/06/2017	Addition of Draco Tera KVM switch protocol. Addition to define an infodriver for PTI, comms status and controller status data for router hardware. Fix for parsing SWP-08 lock responses for destinations below 1024.	P.W.
3.5.1	16/11/2017	Addition of Lightware DVI router protocol.	P.W.
3.5.2	16/12/2017	Option for Harris routers to not poll by default	P.W.
3.5.3	8/01/2018	Option for RX driver to enable or inhibit the sending of cross-point commands to hardware	P.W.
3.5.4	8/02/2018	Modifications for Harris to send Route commands faster	P.W.
3.6.0	13/03/18	Addition of NVision NP0017 / NV9000 protocol for GV Convergent control. Internal rework to handle a router with separate device and level states as used by this protocol.	P.W.
3.6.1	10/04/18	Fix for RM messages when running with BNCS_CSI	P.W.
3.6.2	17/04/18	version for BNCS GRD ranges greater than 4096.	P.W.
3.6.3 3.6.3.2014 3.6.3.2015	20/08/18 04/10/2018 10/01/2019 22/01/2019 17/02/2019 20/02/2019 27/02/2019	Added multi-group functionality to work with virtual source and destinations too. NV9000 levels and mappings can be defined in device database files, enabling run time alteration / definition of source and destination levels and index mapping. Addition of Infodriver_Divider setting to set how many slots per infodriver used for large routers > 4096 dests Instance name can be passed in as start-up parameter or device index. If the former is used then instances.xml is parsed to determine the router number for driver. Fix for Jira issue 505 – Driver will report Network Broken state upon comms lost or no response from hardware. Fix for Jira issue 507 – Fix for client connection. IP_Router driver is now compiled using VS2013 with relevant VS2013 libraries. Fix for Jira issue 515 – resolving virtual routes on driver start-up for non-square routers.	P.W.
3.6.4.1	03/04/2019	Added NP0017 PORTS based protocol. A simpler variant of the GV Convergent NV9000 protocol than that added in version 3.6.0	P.W.
3.4.6.11	23/04/2019	NVPORTS polls for 8 destinations each time to increase speed of working through a whole router. STEREO optional setting for a router definition, which ties paired sources and destinations to be routed as a group.	P.W.
3.4.6.101	13/05/2019	Fix in GV NVPORTS for registration command and response for change of data notifications	P.W.
3.6.4.401	12/07/2019	Option to use '-5010' in RC mask to allow a sole route to the master destination of a defined multigroup rather than routing to the whole group	P.W.
3.6.4.443	13/09/2019	Fixes for NV9000 protocol on driver start up. Fix for Jira 597 – correctly switches to use the alt tcp-ip port on comms loss for relevant connection. Fix for using virtuals with DRACO protocol	P.W.
3.6.4.557	07/10/2019	Fix for reading configuration to reduce number of file reads. NV9000 GV routing – Option to inhibit sending the same route if that route already sent and in process on GV hardware which can take 5 seconds to respond. NV9000 GV routing – fewer levels checks when setting up route cmd	P.W.
3.6.4.702	07/11/2019	Improvements for DRACO KVM routing	P.W.

3.6.5.24	22/11/2019	Addition of new special mode of operation: MAGNUM_VA – for use with the QUARTZ protocol setting – routing via an Evertz Magnum Controller that has main and second router on separate V and A quartz levels. Initially for CCM. Menu options to save and recall / reassert a snapshot of routes and for defined routers	P.W
3.6.5.101	02/02/2020	Harris protocol additions to allow for use of router defined levels in routing commands	P.W.
3.6.5.210 3.6.5.314	01/06/2020 30/07/2020	Fix for Draco protocol when hardware "echoes" commands sent in back out to driver Fix for error in parsing SWP-08 tally dump response for small (<190 dests) routers. Tally dump was only requested at driver connection. Fix to allow source 0 when mapping hardware to bncs sources	P.W.
3.6.6	02/10/2020	Restored debug logging to BNCS log file	P.W.
3.6.7	23/11/2020 18/01/2021 08/03/2001 29/03/2021 19/04/2021 22/04/2021	Fixes for Names updating to hardware for Probel SWP-08 Addition of Names commands for Evertz/Quartz protocol Fix for Source Mapping when the Enable_Source_Zero option is 1. Probel Aurora SWP-08 or SWP-02 now inhibited for setting the option to Enable_Source_Zero as these protocols do not permit a source of 0. 3.6.7.41 fix for responses from hardware when the router is physically larger than that defined for the bncs device (caused exception in GetRouterForGivenMatrixLevel method) 3.6.7.48 additional GV NV9000 Mnemonic poll commands for names per index 3.6.7.103 Metadata database values for additional data to show on driver GUI 3.6.7.204 Poll locks on reconnection for SWP-08 Evertz error responses written to slot 4096 of "extras infodriver", if the infodriver is defined. 3.6.7.502 SWP-08 / AURORA/CALREC hardware responses *strictly* use matrix and level and destination to determine response processing 3.6.7.801 Ini file options to inhibit / enable Background polling for destination routes and hardware lock status. Fix for PTI when sources greater than infodriver_divisor setting 3.6.7.901 Modifications for NV9000 to re-register for change messages on partial comms loss (e.g. 1 of the 2 tcp-ip connections)	P.W.
3.6.8 3.6.8.510	18/06/2021 28/06/2021 24/08/2021 26/10/2021	Driver will always provide a route revertive for destinations, even of 0 for unknown routes. To fulfil router poll requests for CSI etc If a Park source is assigned a value, it will always be routed when RC with source 0 is requested, even of 0 is a valid source. Fix and modifications for comms handling with NV9000 GV hardware Modifications for comms state reporting. When a comms slot and extras infodriver are assigned, the slot is updated every 15 seconds. States: 0=no comms 1=connected but no response 2=comms ok	P.W.
3.7.0	27/10/2021 27/04/2022	Addition of Harris LRC+ Levels routing. Internal rework of driver to extend the same methods and concepts originally developed for the GV NV9000 protocol style of routing to this Harris LRC+ implementation. Optional GVC and Harris LRC AV Format databases and AV format checking via ini file setting or GUI menu that a video route is permissible. Error message reported to Extras Infodriver if route FAILS.	P.W.
3.7.2.115	21/06/2022 27/06/2022	Nvision GV NP16 command 302A option added for batching up of route commands to the hardware for better GVC processing. Enable this option in driver only if hardware has 302A firmware upgrade. Batches all GVC route commands if received in same BNCS packet	P.W.
3.7.2.121	28/06/2022	Fix for External GRD mode to process RP and RM commands internally, if external only processes RCs - Jira 1345	P.W.

BNCS

5.2 Document version

Version	Date	Details	Name
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	16/06/17	Driver version 3.4.9 docs updated for version changes	P.W.
3.2	26/09/17	Driver version 3.5.0 additional options	P.W.
3.3	16/11/2017	Driver version 3.5.1 Lightware routers protocol	P.W.
3.4	8/01/2018	Updated docs for the option for an RX driver to enable or inhibit the sending of cross-point commands to hardware	P.W.
3.5	13/03/18	Addition of the NVision NV9000 protocol	P.W.
3.6	04/10/18	Adding multi groups functions for virtuals and NV9000 levels and mappings can use device database files if required	P.W.
3.7	18/07/19	Added docs for the ports based protocol variant of NV9000/NP17 Revised documentation on NV9000/NP17 protocol Added notes for STEREO special mode option in order to route pairs of sources and destinations as groups by default. Option in RC command for setting in the mask of '-5010' to route solely to the master destination of a multi-group rather than the whole group.	P.W.
3.8	22/11/19	Notes added for the MAGNUM_VA special mode of operation with 2 routers	P.W.
3.9	01/01/20	Snapshot save / reassert of routes as an GUI menu option	P.W.
3.10	13/04/21	Metadata source and destination database assignment in [Router_nn] section for additional data to be shown on GUI	P.W.
3.11	27/10/21	Updated documentation for addition of Harris LRC levels routing	P.W.
3.12	21/06/22	Update docs for Nvision 302A routing command option	P.W