bSerialProxy Driver

bSerialProxy is a driver that, when paired with bSerialSplit, can provide serial connectivity to a remote controller's serial input. Data passing through the driver can be inspected and displayed as ASCII, Hex and/or Binary formats in the standard Lua Output window. bSerialProxy can also inject data into the serial connection.

Setup

1) Within your Control4 project, install an instance of the bSerialProxy driver for every serial connection you would like to inspect or connect to remotely
2) In Composer Connections, connect bSerialProxy's RS-232 Output to a RS-232 Input device
3) Set the Show Ascii/Hex/Binary properties for the type of data you are expecting. Note that not all Hex and Binary data have an Ascii equivalent
4) Select the desired bSerialSplit Server. bSerialProxy will attempt to connect to the auto-populated bSerialSplit Url
5) Depending on the verbosity of the endpoints, you should immediately start seeing the serial data in the Lua Output. Use Debug Level "3 - Info" for the cleanest data view

Properties

Driver Setup

Upgrade Mode	BNet Solutions drivers can automatically update themselves. Options are "Automatic", and "Upgrade Now". See "Upgrade Mode" section below
Admin Server Enabled	Enables / Disables the driver's Admin Server. See "Admin Server" section below
Admin Port	Listening port assigned to Admin Server
Admin Token	Token required to access the driver's Admin Server

License Setup

License Key	Key used to permanently enable driver functionality
License Status	Display current state of driver's license
Driver Version	Installed driver version
MAC Address	Unique network interface identifier for the Controller

bSerialProxy Setup

Show Ascii	Show/Hide the Ascii representation of serial data
Show Hex	Show/Hide the Hex representation of serial data
Show Binary	Show/Hide the Binary representation of serial data
bSerialSplit Server	List of discovered bSerialSplit Servers
bSerialSplit Url	Connection URL of selected bSerialSplit Server

Serial Debug

Input Type	Input data type in Send To Input property
Send To Input	Data to inject/send to the Input endpoint of the serial

	connection. See “Injection” section below for more detail
Line Termination	Character sequence to append to the injected data -CR (Carriage Return) -LF (Line Feed) -CRLF (Carriage Return AND Line Feed)

Debug Settings

Debug Mode	Sets where the driver outputs debug information. See “Troubleshooting” section below for more information
Debug Level	Sets how much debug detail the driver outputs

Actions

Discover bSerialSplit Servers	Sends discovery request for bSerialSplit Servers on local subnet and repopulates bSerialSplit Server property
Connect WebSocket	Connects to the server located at the bSerialSplit Url property
Disconnect WebSocket	Disconnects from the remote bSerialSplit Server
Send To Input	Sends the data entered in Send To Input property. Used to repeatedly send data after the property is set

Injection

bSerialProxy can inject data into the serial connection using the Send To Input property and Action. Set the Input Type property (Ascii, Hex or Binary) to the appropriate data type and enter the data to send in the Send To Input property. The data will be sent when the property is “Set”. For repeat Sends (after the property is set), use the Send To Input Action.

Set the Line Termination property to the appropriate line termination sequence. Options are LF (Line Feed), CR (Carriage Return), CRLF (Carriage Return and Line Feed) and None. The below examples use CR Line Termination (Hex code 0d and binary 00001101).

When sending to input, “Device Name” is replaced by “Debug” in the Lua Output and the Input Type is marked with a “>” to show what was actually sent. For example:

```
Debug --> WebSocket   Ascii : >Test
Debug --> WebSocket   Hex   :  54  65  73  74  0d
Debug --> WebSocket   Binary:  01010100 01100101 01110011 01110100 00001101
```

Remote Serial Connection

My development controller is separate hardware from my production controller and the two are not in close proximity. By connecting a bSerialProxy instance to bSerialSplit’s* Websocket Server, I am able to develop on my development server but use a physical input serial connection from my production controller. In the case of my serial-controller pool equipment, the setup looks like this:

```
[-        Production Env      -]    [-         Development Env         -]
Pool <-RS232 In-> bSerialSplit <-WS-> bSerialProxy <-RS232 Out-> bPentair
```

*bSerialSplit is a separate driver from bSerialProxy and can be found on the BNet Solutions website.

bSerialProxy Driver

Development Driver

bSerialSplit and bSerialProxy are intended for use as development tools. While I have run these drivers on my production controller since early 2020, the volume of data over some serial connections, data transformations (read: lots of strings, bit operations and “math” that Lua is not known to excel at) and the extra serial and socket I/O all equals more “work”.

Admin Server

BNet Solutions drivers' have a built-in webserver that looks and behaves like the Properties, Actions and Lua Output tabs for the driver in Composer. The Admin Server’s default port for the bSerialSplit driver is 41900 and is configurable in the driver's properties. Using a web browser, navigate to http://[controller ip]:41900 where “controller ip" is the IP Address of your Control4 Director (EA5, EA3, EA1, etc). For example, http://192.168.1.100:41900. The Admin Server is protected by a challenge page that requires a token to continue. By default, the token is "bSerialSplitAdmin". Once authenticated, the token is stored in a cookie (technically hashed, then stored) so you won't need to log in every time. The token is configurable via the driver's property page. The Admin Server is enabled by default but can be disabled entirely via the driver's property page.

Upgrade Mode

BNet Solutions drivers can automatically update themselves. New driver functionality or capability is typically packaged as an incremental “Major” version (v3, v4 etc). “Minor” versions (v2.3, v2.4) are typically maintenance releases that update underlying libraries, address a specific issue or usability concern.

Automatic	When “Automatic” is selected, the driver will upgrade/update itself when a new version is available. This is currently the only option.
Update Now	Checks for and upgrades to any newer Major or Minor version.

Known Issues and Limitations

- Requires C4 OS v2.10.X or greater
- C4 defines serial connections on a per-driver basis, not per-input/output connection. This driver is expecting 9600 baud, no parity bit and 1 bit stop (9600/8-N-1). Other serial settings will likely not work properly

Troubleshooting

All BNet Solutions products have an additional ‘Submit’ Debug Mode. With this mode selected, the driver creates a unique log file to capture the Lua output based on the selected Debug Level (usually set to "5 - Debug"). Once ‘Submit’ Debug Mode is deselected, either manually or when the Debug Timer expires, the Submit Debug Log is uploaded to the BNet Solutions Server for analysis.

The server notifies me when Submit files are uploaded but if you have not purchased a license, I will have no way to reach back out to you for troubleshooting so please email me your contact information.

Legal

By using this driver, you are indicating that you have read and agree with the Policies and Terms that govern its usage as published here.

My Contact Information

You can reach me at blucas@bnet4solutions.com for comments or questions.

Change Log

v1 - 08/21	Initial Release