Harkwood Services Ltd

Reference Manual

cmdBridge®

Intelligent command protocol translation

FCC Class A Notice

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the FCC rules. The Class A limits provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause interference. This interference must be corrected at the expense of the user.

About cmdBridge

cmdBridge is a device to assist with the maintenance and management of audio / video (AV) systems. When deploying cmdBridge a target device, under the control of an AV control system, can be easily replaced without the need to re-program the control system. It also enables such equipment exchanges where the source code of the control system is unavailable. This enables maintenance and support companies to provide customer support to a wide range of customer equipment, whilst retaining a more limited spares / loan stock.

cmdBridge can also be deployed to extend the life of systems where the original control system source code is unavailable, or the system cannot be easily updated. For example, a complex room join requires new displays but the original source code is unavailable, cmdBridge can be deployed without the need to re-program the entire system.

cmdBridge may also be used as a diagnostic tool, monitoring the communication between a control system and target device, to help with troubleshooting.

Firmware version information

This manual covers firmware version 1.0.x, if upgrading to a newer firmware version please ensure you obtain the current manual from the support section of the website.

Document Conventions

The following notifications are used within this manual

| CAUTION | Identification of actions required to ensure correct operation of equipment and prevent damage |

| NOTE | Provides information specific to a particular item |

| TIP | Provides helpful information relating to operation of configuration |

Commands entered via the console end with a Carriage Return which is ASCII code 13 or 0x0D, as in the example below;

```
show config<cr>
```

Translation configuration file data is shown in the format below;

```
command {
        source_cmd = "6F 6E 65";
        target_cmd = "6F 6E 65 20 6F 75 74 70 75 74";
        reply {
                desc = "1st reply to command 1";
                target_reply = "72 65 70 6C 79";
                source_reply = "72 65 70 6C 79";
                flag = default;
            }
        reply {
                desc = "2nd reply to command 1";
                source_reply = "69 74 20 77 6f 72 6b 65 64";
            }
    }
end
```

Introduction

About this manual

This manual describes the installation, configuration, functions, and operation of cmdBridge.

Features

RS-232 Ports - Bi-directional RS-232 ports with optional hardware flow control. Presented on 3.5mm pitch phoenix connections.

Ethernet Switch - Integral 2 port 10/100 Ethernet switch with Auto MDI-X automatic crossover support.

> `TIP` Either port may be used to connect cmdBridge to a network.

USB Type A - USB type A port for the connection of USB memory sticks, to import / export configuration data sets.

> `NOTE` Supported file systems are FAT and FAT32

USB Mini-B - USB Mini-B console port presenting as serial over USB for system configuration and monitoring. This port can also be used to supply power so configuration does not require an external power supply.

Power IN - 12v DC power supply, this port takes precedence over Mini-B USB power.

> `CAUTION` Ensure to check the polarity of the connection before applying power.

Status Lights - Indicator lights provide a quick view of port status, physical connection type, and data moving through the system. The lights have an adjustable brightness level.

Pass-through Mode - enables physical interface translation between RS-232 to IP or IP to RS-232, without altering the data.

Translation Mode - enables translation between the source and target devices

Capture Mode - Listens and creates a template configuration file based on communication observed.

Internal Storage - Space to keep multiple translation sets internally. No requirement to keep a USB memory stick attached for deployment.

Installation

Front Panel connections and indicators
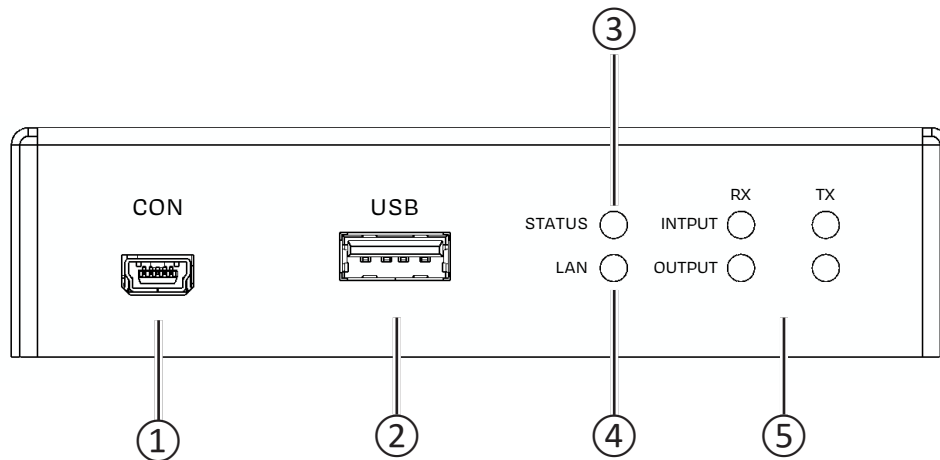
① **Console port** - used to configure cmdBridge.

> NOTE Console port can also be used to supply power, no external PSU is required during configuration or deployment if the USB power source can supply 1A.

> NOTE Console port is a Serial over USB port, running at 230400 baud, 8 bits, no parity.

> NOTE Do not use a USB cable of 2m or longer.

> TIP Recommend serial console software for Windows is 'PuTTY' and for Apple Mac computers 'Serial' by Decisive Tactics.

② **USB Port** - USB memory stick connection to enable configurations to be uploaded / downloaded, captured data to be stored, and for firmware upgrades.

③ **Status LED** - Indicates the system mode or fault condition as below;

Green	Pass-through mode
Blue	Translation mode
Red	System fault, please see troubleshooting section for details

④ **LAN LED** - Indicates the LAN status, the LED is lit when one or more port is connected. The LED colour indicates the MAC address being used as below;

Green	System default
Blue	Custom

> NOTE A solid colour indicates a IP address is set, either statically assigned or provided by DHCP. A blinking colour indicates DHCP is in progress.

⑤ **Data in/out** - Indicates that data is being received (RX) or transmitted (TX) on the respective port, the LED colour indicates the physical interface type as below;

Green	RS232 data
Blue	IP data
Red	TX only, the remote end is not accepting the IP connection, no data could be sent.

Rear Panel connections and indicators
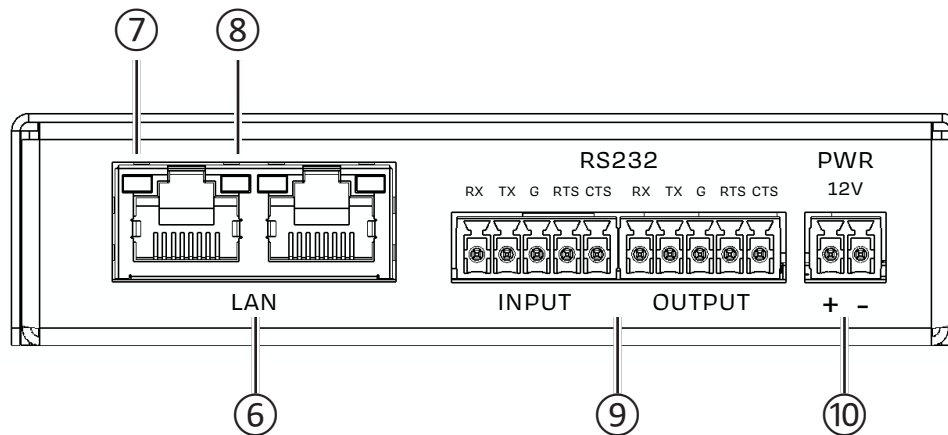
⑥ **Ethernet Switch** - 2 port 10/100 Ethernet switch to connect cmdBridge to an Ethernet network. Both switch ports support MDI-X auto crossover.

NOTE CAT5e shielded or CAT6 and above cables are recommended.

TIP The switch enables cmdBridge to installed in-line with equipment without the need to have an additional network port to be made available.

⑦ **Ethernet Link** - A solid yellow LED indicates a good LAN connection.

⑧ **Ethernet Activity** - A blinking green LED indicates LAN activity.

⑨ **RS-232 Ports** - Serial input and output ports. Connect using a 3.5mm 3 pin connector, or 5 pole connector if hardware handshaking is required.

⑩ **Power connection** - Connect a 12v DC power supply with an output of 500mA or more.

CAUTION To prevent damage verify the voltage and polarity of the power supply before turning on.

CAUTION To prevent damage do not connect the power cable to the RS-232 ports.

System Configuration

cmdBridge is designed to be installed in the signal path between a Source and Target device. The physical Input and Output ports do not need to be the same type or speed.

Configuration of system parameters, physical port types, the loading and saving of translation configuration files is by Command Line Interface (CLI) via the console port.

Getting Started

Connect the Console Port ① to the computer used for configuration, this will also power cmdBridge if no external Power Supply Unit (PSU) is available.

NOTE Do not use a USB cable of 2m or longer.

Using serial console software connect to the Serial port created by the host OS. Configure the port to be 230,400 baud, 8N1, no flow control.

The normal command prompt is;

```
>
```

If there are unsaved changes to the configuration, the prompt is;

```
*>
```

The command line interface commands are grouped into areas depending on function.

SYS

These commands related to system wide functions, such as setting the IP address, system mode, capture and monitor functions.

TRANS

Relate to commands handling translations. Translation files are loaded from a USB memory stick in port ②, they are verified during load. Should errors be present in the file, a list of the problems and their location in the file is provided.

Once loaded the system may be put into translation mode, however the translation will be lost on a reboot. To have a translation available for use on system start-up, it must be saved into one of a number of internal translation slots.

INPUT

Commands that relate to the physical configuration of the input port.

OUTPUT

Commands that relate to the physical configuration of the output port.

NOTE Some commands require a reboot to become active, which may be automatic. Any unsaved changes will be lost.

System Commands.

```
help
```
        Displays a list of commands and their function

```
show config
```
        Shows the running configuration

```
save config
```
        Saves the running configuration

```
sys light <1..100>
```
        Adjusts the brightness of the status lights on the front panel.

```
sys ip add <addr> <mask>
```
        Sets the IP address when DHCP is off

```
sys ip gateway <addr>
```
        Sets the system IP gateway when DHCP is off

```
sys ip dhcp <on | off>
```
        Turns on or off obtaining IP details via DHCP

```
sys mac addr <xx:xx:xx:xx:xx:xx | default>
```
        Set the Ethernet mac address to a specific address, or back to default.

        `NOTE` This command will automatically trigger a reboot.

        `NOTE` When a custom mac address is set, the LAN Led ④ will be blue to indicate a custom address is set when the network is active.

        `TIP` This enables cmdBridge to take the mac address of a device being replaced, negating the need to update a DHCP server that is statically assigning address by mac address.

```
sys firmware load <filename>
```
        Uploads new firmware, once the file is verified the system will automatically reboot.

        `NOTE` The Status LED ③ will flash whilst the firmware file is being read and verified.

```
sys mode <pass | translate>
```
        Changes the mode of operation from pass-through mode to translation mode.

        `NOTE` The Status LED ③ is blue for pass-though mode and green for translation mode.

```
sys monitor <on | off>
```
        Turns the system monitor on or off

```
sys capture <on | off | save>
```
        Turns capture mode on or off, when `save` is used the file is saved to the USB memory stick with the file name capture.txt

```
input enable <ip | serial>
```
        Sets the input port to be IP or serial

```
input serial speed <baud>
```
        Sets the input port serial speed

```
input serial flow <on | off>
```
        Sets the input port hardware flow control

```
input serial line <linectl>
```
      Sets the serial line control, for example 8N1

```
input ip port <port>
```
      Sets the IP port the input port will listen on for a connection from the Source equipment.

```
output enable <ip | serial>
```
      Sets the output port to be IP or serial

```
output serial speed <baud>
```
      Sets the input port serial speed

```
output serial flow <on|off>
```
      Sets the input port hardware flow control

```
output serial line <linectl>
```
      Sets the serial line control, for example 8N1

```
output ip add <x.x.x.x> <port>
```
      Sets the output port Target device IP address and port to connect to

```
list usb <filespec>
```
      List the files on the USB memory stick attached to USB port ②. An option filespec may be used to limit the list, e.g. *.txt

```
trans loadfile <filename>
```
      Reads a translation file from the USB memory stick attached to USB port ②. Any errors in the file will be displayed as the file is verified.

```
trans clear
```
      Clears the current translation.

```
trans loadslot <1..32>
```
      Loads a translation from one of the internal storage slots.

```
trans saveslot <1..32>
```
      Saves the current loaded translation into the indicated slot, any existing translation in the specified slot will be overwritten.

```
trans clearslot <1..32>
```
      Clears the translation stored in the specified slot.

```
trans list
```
      List the translation stored internally

```
trans copyslot <1..32> <filename>
```
      Copy the translation stored in the specified slot to a memory stick attached to USB port ②.

```
reboot
```
      Reboots cmdBridge

      `NOTE` any unsaved changes will be lost.

```
factory reset <config | all>
```
      Resets cmdBridge back to factory default.
          Config   resets just the configuration leaving stored translations intact.
          All   resets both the configuration and all stored translations.

      `NOTE` This command will automatically trigger a reboot.

Physical input / output ports

Source devices connect to input ports and Target devices connect to output ports. The physical port configuration is independent to the translation.

The types of port used may be independent or the same, buffering is used to process commends where the speeds of the interface vary.

Input	Output
Serial	Serial
Serial	IP
IP	Serial
IP	IP

Serial Ports

Supported serial speeds are 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200

Supported line control options are;

8N1   8 bits, no parity, 1 stop bit
8N2   8 bits, no parity, 2 stop bits
8E1   8 bits, even parity, 1 stop bit
8E2   8 bits, even parity, 2 stop bits
8O1   8 bits, odd parity, 1 stop bit
8O2   8 bits, odd parity, 2 stop bits
9N1   9 bits, no parity, 1 stop bit
9N2   9 bits, no parity, 2 stop bits

Serial interfaces may have hardware flow control enabled.

NOTE When serial data is received or transmitted, the associated input/output LED will blink green.

IP Ports (Ethernet)

Source devices connect to the cmdBridge IP address and a specified port number, using the command.

```
input ip port <port>
```

cmdBridge will connect to a Target device IP address and port, using the command;

```
output ip add <x.x.x.x> <port>
```

NOTE When IP data is received or transmitted, the associated input/output LED will blink blue.

NOTE When IP data is transmitted, but a connection to the target device is unavailable, the associated output LED will blink red.

Translation files

Translation files enable a Source device to send commands to a Target device, with the data being translated en-route. Likewise Target device replies are translated into Source device replies. Multiple replies per command are supported, combined with flags to send replies when certain circumstances are met.

The full syntax can be found in Appendix A. Files are plain text and can be edited in a simple text editor, no client specific software is required.

Command files consist of a header element which is followed by one, or more, commands, then an end. Commands contain the replies linked to that specific command.

The end of the file is identified simply with end

```
end
```

Header element

Description

The start of a file may contain a description, as below.

```
desc = "EPSON EB-535W to NEX NP-M302WA";
```

The description is used to name the translation, and is shown in the configuration page when a translation is loaded. It also names a translation slot if the file is stored internally.

TIP Descriptions can also be added to commands and replies, these can be used to help document a translation file.

Variables

Variables may be defined, for example to track the system status. They are numeric, and must be declared before they are used, as below.

NOTE Up to 32 variables may be defined.

```
var power = 0;
```

This declares a variable called power and sets it to an initial value of 0.

Variables may set within commands and replies, and may be formed of compound statements. For example;

```
set power = 5;
set counter = counter + 1;
set power = last_power;
```

Translation data element

Commands

A Command consists of the data received from the source device (the `source_cmd`), if there is a match it is translated into the target command data (the `target_cmd`) and sent to the target device.

If no match is found the source command is discarded.
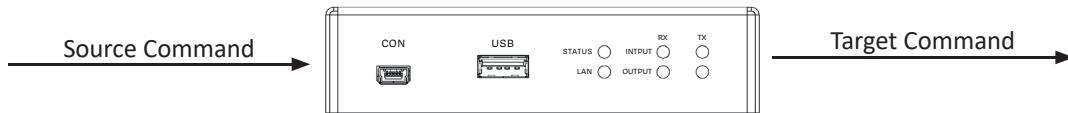

```
command {
      desc = "Mute OFF";
      source_cmd = "4D 55 54 45 20 4F 46 46 0D";
      target_cmd = "02 11 00 00 00 13";
   }
```

NOTE All command / reply data is input in hexadecimal pairs.

NOTE Commands / replies may be up to 768 bytes in size.

In the example above the `source_cmd` in ASCII would be MUTE OFF<cr>

Replies

Replies are context sensitive to the last translated command. A target reply (the `target_reply`) is received from the target device and translated into a source reply (the `source_reply`) and sent the source device.
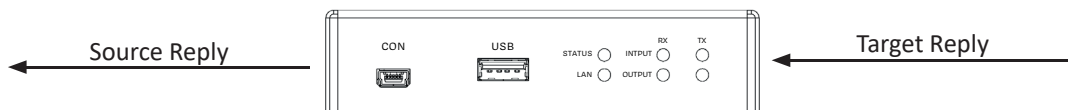

```
command {
      desc = "Mute OFF";
      source_cmd = "4D 55 54 45 20 4F 46 46 0D";
      target_cmd = "02 11 00 00 00 13";
      reply {
             desc =  "Mute OFF OK";
             target_reply = "22 00 01 20 00 43";
             source_reply = "3A"
         }
    }
```

A command may have multiple replies, they are processed in the same order as they are defined in the translation file.

If no match is found then the reply is discarded. However this behaviour can be overridden with the use of reply flags set for each specific reply.

Reply flags

Flags alter the way replies are processed, creating flexibility in replies that can be sent to the source device, even if no actual target reply was received. The available flags are default, conditional, and always.

Using a combination of flags and variables a local state machine can be developed within a translation file, to keep track of a system state of a device.

For example, an original item of equipment may support multiple commands to retrieve the current status of various items, such as power status, active port etc. A replacement device may support only a single global status request. Using variables to track the state of commands as they are received by cmdBridge, and conditional replies, such a situation can be easily managed.

default

If a reply is received from the target device, and there is no other match found, then send this reply.

```
command {
        desc = "Set Mute ON";
        source_cmd = "4D 55 54 45 20 4F 4E 0D";
        target_cmd = "02 10 00 00 00 12
        reply {
                desc = "Reply Mute ON OK";
                target_reply = "22 10 01 20 00 53";
                source_reply = "3A";
                flag = default;
            }
        }
```

always

Always send this reply, even if there is no actual reply from the target device.

```
command {
        desc = "Set Mute ON";
        source_cmd = "4D 55 54 45 20 4F 4E 0D";
        target_cmd = "02 10 00 00 00 12
        reply {
                desc = "Reply Mute ON OK";
                source_reply = "3A";
                flag = always;
            }
        }
```

```
conditional
```

Send this reply if the specified condition evaluates as true, even if no actual reply from the target device is received.

```
command {
        desc = "Get Mute status";
        source_cmd = "4D 55 54 45 3F 0D";
        reply {
                desc = "Variable based reply, off";
                source_reply = "4D 55 54 45 3D 4F 46 46 3A";
                flag = conditional;
                if (mute == 0);
            }
    }
```

Within the `if` statement the following tests are available;

==	is equal to
!=	is not equal to
&&	logical AND
||	logical OR

Statements may also be compounded, for example;

```
if ((mute == 0) && (power == 1));
```

The complete example below shows how variables and conditions can be used to perform status tracking of events.

```
desc = "EPSON EB-535W to NEX NP-M302WA (mute example)";
var mute = 0;

command {
        desc = "Set Mute ON";
        source_cmd = "4D 55 54 45 20 4F 4E 0D";
        target_cmd = "02 10 00 00 00 12";
        reply {
                desc = "Reply Mute ON OK";
                target_reply = "22 10 01 20 00 53";
                source_reply = "3a";
                set mute = 1;
            }
        }

command {
        desc = "Set Mute OFF";
        source_cmd = "4D 55 54 45 20 4F 46 46 0D";
        target_cmd = "02 11 00 00 00 13";
        reply {
                desc = "Reply Mute OFF OK";
                target_reply = "22 11 01 20 00 54";
                source_reply = "3a";
                set mute = 0;
            }
        }

command {
        desc = "Get Mute status";
        source_cmd = "4d 55 54 45 3f 0D";
        reply {
                desc = "Variable based reply, off";
                source_reply = "4d 55 54 45 3d 4f 46 46 3a";
                flag = conditional;
                if (mute == 0);
            }

        reply {
                desc = "Variable based reply, on";
                source_reply = "4d 55 54 45 3d 4f 4e 3a";
                flag = conditional;
                if (mute == 1);
            }
        }
end
```

Managing Translations

Loading translations

Translations can be loaded one of two ways;

1. During configuration from a USB memory stick inserted into port ②. If the file parsed without error, then the file is available for use. Files loaded directly from the memory stick are volatile and will not be loaded on a reboot. To load a file use;

   ```
   trans loadfile <filename>
   ```

   Once a file has been successfully parsed, the loaded file may then be stored internally into one of a number of translation slots. Only translations held in a non-volatile internal slot may be loaded automatically on start-up. To save the currently loaded translation file to a slot use;

   ```
   trans saveslot <1..32>
   ```

2. From an internal non-volatile translation slot, using the command;

   ```
   trans loadslot <1..32>
   ```

   Only translations loaded from an internal slot may be loaded automatically on start-up

Storing translations

Translations can be stored in non-volatile internal translation slots. To be able to store a translation it must first be loaded from a USB memory stick inserted into port ②. Only the file is successfully parsed without error can the file be stored in a non-volatile slot. To save the currently loaded translation file to a slot use;

```
trans saveslot <1..32>
```

NOTE Up to 32 internal slots are available.

Clear Translation

To clear the currently loaded translation, regardless of source, use;

```
trans clear
```

To clear a translation stored in a non-volatile slot, use;

```
trans clearslot <1..32>
```

Copy Translation

A translation stored in a non-volatile internal slot may be copied to a USB memory stick inserted into port ②. To copy the slot use;

```
trans copyslot <1..32> <filename>
```

This will copy the translation file from the specified slot to the specified filename.

Monitor Mode

cmdBridge may also be used as a diagnostic tool, enabling communication between a source and target device to be monitored.

Data received is displayed, in real time, to the console port in ASCII and HEX, from the physically configured input and output ports. The data is shown to be on the logical interface received.

The physical input and output ports can be serial or IP, depending on requirement.

NOTE Monitor may be used even if translation mode is active, so translations may also be viewed in real time.

The output is displayed on the system console, as hex data and as ASCII if printable. Non printable characters are replaced with a full stop.

Example of captured data;

```
Source Command IN  :
50 57 52 20 4F 4E 0D                                    PWR ON.

Target Reply IN    :
4F 4B 0D                                               OK.
```

Capture Mode

Capture mode can be used to help create translation file templates. When inserted in the data path of a working system, and capture mode activated, cmdBridge will record all the commands and associated replies between the source and target device.

The captured data may then be saved to a USB memory stick in inserted into port ②, in the translation file format. This file, in essence, gives one half of the translation file in a ready to use format to create a working translation matrix.

This can help, for example, when taking on a new maintenance contract. During an initial site visit, the existing interaction between a control system and end devices can be captured. Translation files can then be created to match this against the company loan stock, enabling rapid deployment of cmdBridge as and when required.

NOTE Capture mode is only available when in pass-through mode.

During capture the various commands and replies have their description field filled with the command number, and associated replies are also numbered. All data is with reference to the source device, as it would be when forming part of a translation file.

To use Capture mode use the following command;

```
sys capture <on | off | save>
```

```
on    Turns capture mode on
off   Turns capture mode off
save  Saves the data to a memory stick in inserted into port ②, with the file name capture.txt
```

NOTE When turning capture mode on, any previous captured data in memory is deleted.

An example of captured data;

```
desc = "Automatically Captured Data";
command {
        desc = "Command 1";
        source_cmd = "00 85 00 00 01 01 87";
        reply {
               desc =  "Reply 1";
               source_reply = "20 85 01 20 10 FF FF 00 00 00 00 00 00
00 00 00 00 00 00 00 00 D4";
               }
        reply {
               desc =  "Reply 2";
               source_reply = "20 85 01 20 10 00 00 00 00 00 10 FF FF
00 00 FF FF FF FF FF FF DE";
               }
       }
command {
        desc = "Command 2";
        source_cmd = "02 00 00 00 00 02";
        reply {
               desc =  "Reply 1";
               source_reply = "22 00 01 20 00 43";
               }
       }
end
```

Firmware Update

cmdBridge firmware is updated via a USB memory stick inserted into port ②. Firmware is uploaded with the command;

```
sys firmware load <filename>
```

Once the file has been verified, cmdBridge will reboot and the firmware will be upgraded. This process cannot be interrupted.

CAUTION Do not power off cmdBridge whilst the firmware is uploading.

NOTE The Status LED ③ will flash whilst the firmware file is being read and verified.

NOTE The Status LED ③ will go green and the Input / Output RX and TX lights will cycle in a green circular pattern to indicate the firmware writing is in progress.

Should a firmware upgrade fail, the system will revert to the last previously verified firmware automatically.

Troubleshooting

Translation file errors

During the loading of a translation file it is parsed to ensure it is correctly formatted. Should an error be identified within the file this is reported back to the console, along with the approximate line number of where the error is located. Multiple errors may be reported during verification.

The errors relate to either the file syntax, or values being out of bounds.   The errors are self explanatory but may have been caused by a missing semi-colon or bracket further up the file than the indicated line number.

System fault indication

System faults are indicated when the Status LED ③ is red. The specific fault is indicated by another LED being lit and it's colour.

LED	Colour	Issue
Lan	Red	The Ethernet ports could not be initialised. If the problem persists following a power cycle the unit will require service.
Input RX	Red	The internal storage card has failed to read or write. If the problem persists following a power cycle, please contact support for assistance. CAUTION Do not attempt to swap the internal storage card for a generic card, or the unit will not operate correctly.

Firmware update firmware recovery

CAUTION   This method will erase ALL non-volatile and configuration data. It cannot be reversed.

CAUTION   This procedure should ONLY be performed if a normal firmware upgrade procedure is not possible.

To recover cmdBridge from a failed or faulty firmware upgrade, this procedure may be performed to effect a total system reset. This will;

> Revert cmdBridge to the original factory shipped firmware image
> Clear any and all stored translations
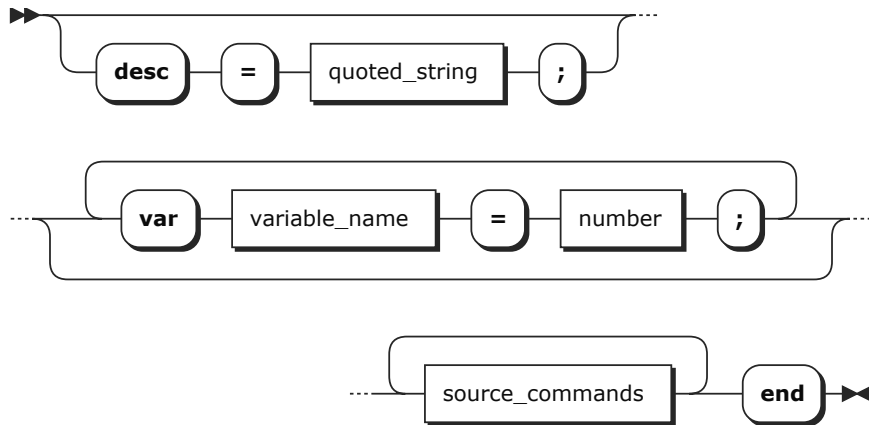> Clear the system configuration

Connect the Input Serial port RX line to the Output Serial port RTS line with a link cable, then turn the system on. Once the firmware writing procedure starts (the Input / Output TX and RX LEDs start to cycle), remove the link cable.
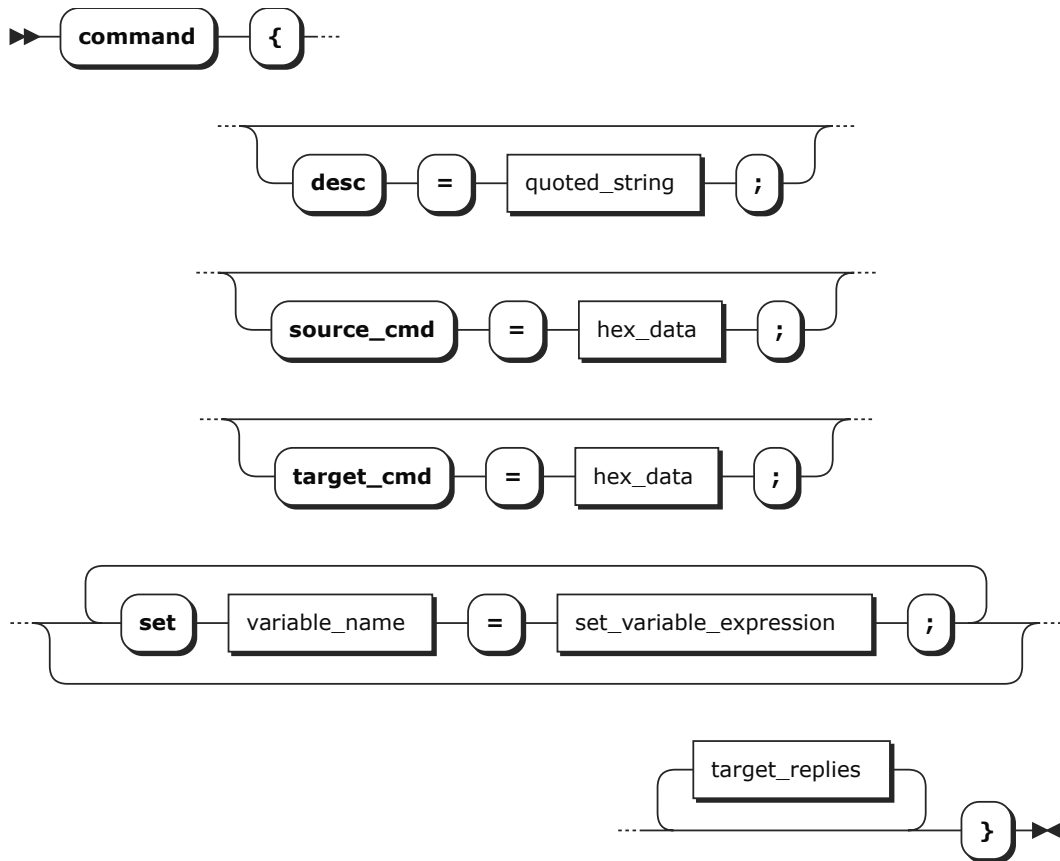
Appendix A

Translation file syntax

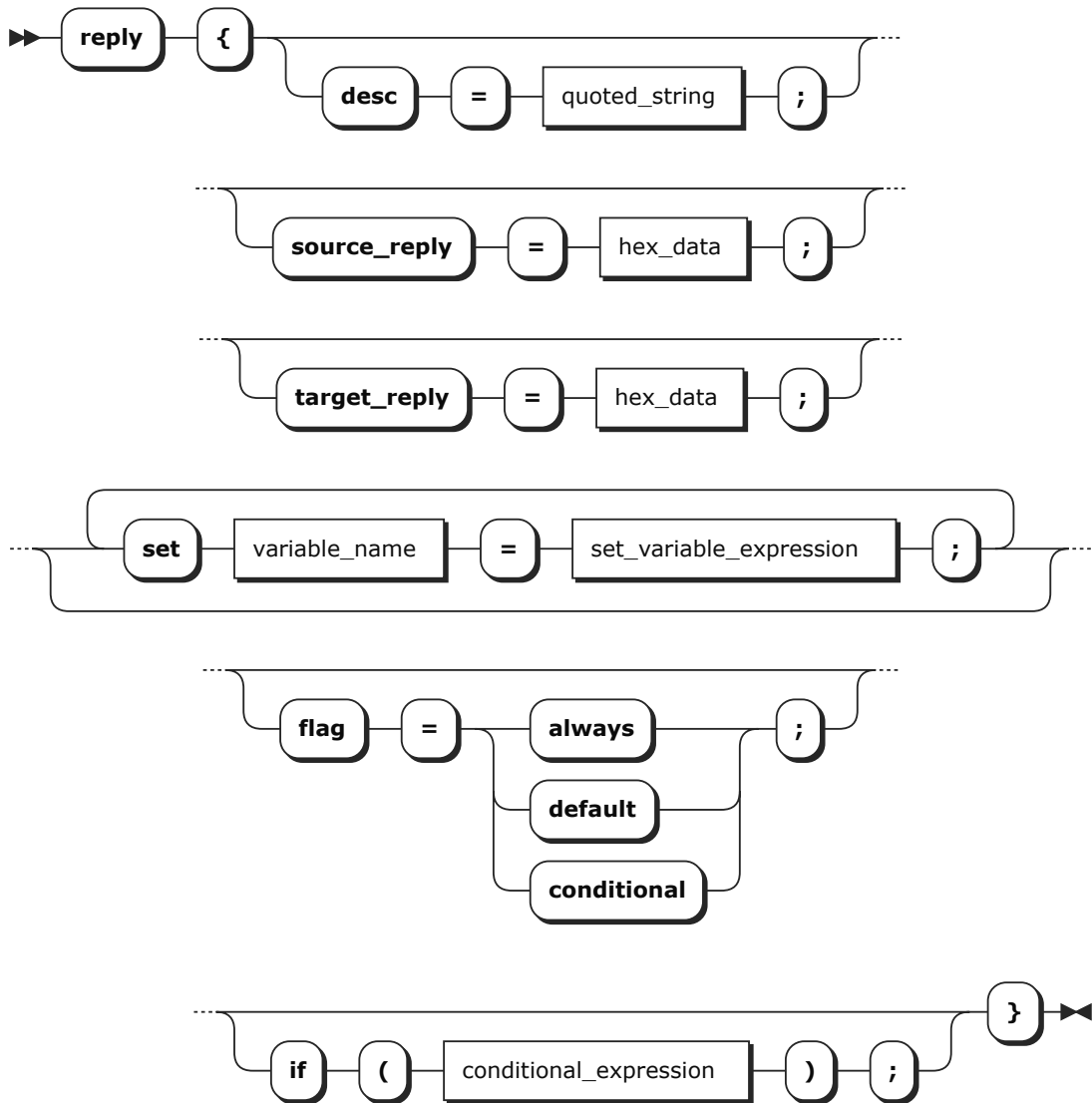The syntax for translation files is provided below as rail road diagrams.

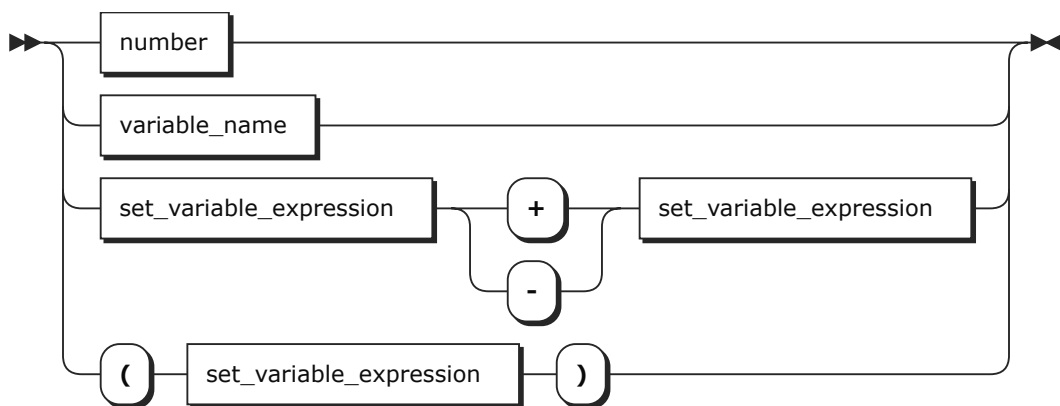### Translation

source_commands

target_replies

```
reply { 
    desc = quoted_string ;
    source_reply = hex_data ;
    target_reply = hex_data ;
    set variable_name = set_variable_expression ;
    flag = always ;
           default
           conditional
    if ( conditional_expression ) ;
}
```

set_variable_expression

```
number
variable_name
set_variable_expression + set_variable_expression
                        -
( set_variable_expression )
```

conditional_expression

number

hex_data

variable_name

quoted_string

➤➤ ─(**"**)─┬─────────────┬─(**"**)─ ►◄
           └─[**<printable character>**]─┘