QUATECH

CONNECT WITH RELIABILITY

Reference Manual

Airborne Command Line Interface (CLI) Enterprise Addendum WLNG-SE/SP/AN/ET-DP500 Series

ABDG-SE/ET-DP500 Series

ABDG-SE/ET-IN5000 Series

Revision 1.3

February 2011

File name: **airborne enterprise command line reference guide.doc**

Document Number: **100-8081-130**

<Page Intentionally Left Blank>

Quatech Confidential

Disclaimer

The information in the document is believed to be correct at the time of print. The reader remains responsible for the system design and for ensuring that the overall system satisfies its design objectives taking due account of the information presented herein, the specifications of other associated equipment, and the test environment.

QUATECH ® Inc. has made commercially reasonable efforts to ensure that the information contained in this document is accurate and reliable. However, the information is subject to change without notice. No responsibility is assumed by QUATECH for the use of the information or for infringements of patents or other rights of third parties. This document is the property of QUATECH ® Inc. and does not imply license under patents, copyrights, or trade secrets.

Quatech, Inc. Headquarters

QUATECH ® Inc.
5675 Hudson Industrial Parkway
Hudson, OH 44236
USA

Telephone: 330.655.9000
Toll Free (USA): 800.553.1170
Fax: 330.655.9010
Technical Support: 800.553.1170 / support@quatech.com

Web Site: www.quatech.com

<Page Intentionally Left Blank>

Contents

Figures

Tables

<Page Intentionally Left Blank>

1.0 Overview

Airborne is a line of highly integrated 802.11 radios and device servers, designed to address the demands of the complex M2M market. Utilizing the latest 802.11, CPU and network technologies, the Airborne™ family of products provide a broad, encompassing solution for wireless applications requiring performance, reliability and advanced technology.

The Airborne Wireless Device Server family includes everything necessary to connect a Serial or Ethernet device to a high-performance 802.11 network. The WLNG-XX-DP500 series includes a full featured 802.11b/g radio and a high performance 32bit ARM9 processor running an embedded OS and Quatech's exclusive Airborne Device Server firmware, allowing the wireless network enabling of almost any device or system.

WPA2-Enterprise (AES-CCMP + EAP) is the security standard for leading-edge enterprise networks. The Airborne Enterprise Device Server supports the latest security standards and more. Fully compliant to the WPA2-Enterprise specification, the device includes a wide range of EAP methods (with certificates), including support for legacy functionality (WPA, WEP and LEAP).

The best security and advanced networking is no good if you cannot connect your device to the Airborne Enterprise Device Server. Airborne offers the widest range of Serial and Ethernet based interfaces in the industry. With flexibility and performance the WLNG-XX-DP500 series lets you decide how you want to use it.

Designed by the Quatech engineers specifically to meet the demands of the industrial, automotive and medical markets, the Airborne Enterprise Device Server has the widest operating temperature range and highest level of reliability available, all backed by a lifetime warranty. Quatech also provides FCC Modular certification, potentially removing the need for further regulatory work.

The two previous generations of Airborne Wireless Device Servers have been integrated and deployed into a wide range of applications and markets, including Medical, Telematics and Logistics.

Quatech's 3rd Generation Wireless Device Server extends the reputation of the family further by drawing on the lessons learned and adding the latest technologies. The Airborne Enterprise Device Server family is the industry-leading solution and represents a breakthrough in 802.11 connectivity, for all M2M markets.

The following manual covers a detailed description of the Airborne Command Line Interface (CLI) used for management, configuration and integration of the Airborne and AirborneDirect Enterprise Device Server products into embedded systems.

2.0 Conventions

The following section outlines the conventions used within the document, where convention is deviated from, the deviation takes precedence and should be followed. If you have any question related to the conventions used or clarification of indicated deviation please contact Quatech Sales or Wireless Support.

2.1 Terminology

Airborne Enterprise Device Server and AirborneDirect Enterprise Device Server is used in the opening section to describe the devices detailed in this document, after this section the term ***module*** will be used to describe the devices.

2.2 Notes

A note contains information that requires special attention. The following convention will be used. The area to the right of the indicator will identify the specific information and make any references necessary.

> The area next to the indicator will identify the specific information and make any references necessary.

2.3 Caution

A caution contains information that, if not followed, may cause damage to the product or injury to the user. The area to the right of the indicator will identify the specific information and make any references necessary.

> The area next to the indicator will identify the specific information and make any references necessary.

2.4 File Format

These documents are provided as Portable Document Format (PDF) files. To read them, you need Adobe Acrobat Reader 4.0.5 or higher. For your convenience, Adobe Acrobat Reader is provided on the Radio Evaluation Kit CD. Should you not have the CD, for the latest version of Adobe Acrobat Reader, go to the Adobe Web site (www.adobe.com).

2.5 Courier Typeface

Commands and other input that a user is to provide are indicated with `Courier` typeface. For example, typing the following command and pressing the Enter key displays the result of the command:

```
wl-info <cr>
Module Firmware Version:        1.00
Radio Firmware Version:         5.0.21-210.p17
Link Status:                    Connected
SSID:                           Quatech_Connected
MAC Address:                    000B6B77619E
BSSID:                          0016B637880D
Transmit Rate (Mb/s):           54
Signal Level (dBm):             -40
Noise Level (dBm):              -92
IP Address:                     192.168.1.100
Subnet Mask:                    255.255.255.0
Default Gateway:                192.168.1.1
Primary DNS:                    68.107.28.42
Secondary DNS:                  68.107.29.42
Up Time (Sec):                  48313
```

3.0 Scope

The CLI Reference Manual documents the Command Line Interface (CLI) for the module. This document is an addendum to the Airborne CLI reference manual and describes the commands introduced or updated with the Enterprise Class product family. The Enterprise Addendum should be used in conjunction with the Airborne CLI Reference Manual for a full description of the available Command Line Interface.

The CLI is one of a number of management interfaces for the product family and is comprised of a set of ASCII text commands and parameters used to provision the module, provide module status and environmental feedback, as well as support firmware and file delivery to the module.

The reference manual includes the following sections.

3.1 Overview

In this section we will review the different device configurations and basic operation and functionality of the module. Support for a specific function is dependent upon the device configuration chosen. It will be noted within each section to which configuration it applies.

3.2 Understanding the CLI

This section will cover the use of the CLI and describe the action and reaction to the specific functional calls and commands.

Methods of connection and delivery of the CLI will also be reviewed. CLI conventions, data types and command responses will also be addressed in this section.

3.3 Typical Development System

An outline and description of a basic development and evaluation system will be covered in this section. It is not necessary to use this exact configuration; however descriptions of connectivity and use, utilized on other sections of the manual, will be based upon the system structure described in this section.

3.4 Serial Device Server Use

In this section the base functionality of the module will be described and examples of use and configuration will be provided to highlight the use of the both it and the CLI. Refer to this section to understand the differences between a command port, data tunnel, TCP/IP vs. UDP use and server vs. device operation.

3.5 Ethernet Bridge Use

A full description of the operation of the Airborne Ethernet Bridge, its place in the network infrastructure and the required parameters will be covered in this section.

3.6 WLAN Security

This section will cover the use of the advanced security features available in the module. Configuration of the module, requirements for successful deployment, examples of configuration for the use of the advanced authentication and wireless security options will be provided.

Descriptions of how to use WEP, WPA and WPA2 will be included. Outlines of the authentication methods supported (EAP), certificate delivery and deployment will be reviewed.

3.7 WLAN Roaming

This section will outline the commands that impact the roaming performance of the module. Discussion of configuration options based upon application requirements is also included.

3.8 FTP Configuration

The Airborne Enterprise Device Server family supports delivery of certificates, private keys, configuration files and module firmware via FTP. This section describes how to configure and use the FTP capabilities.

3.9 Power Management

A review of the CLI commands impacting device power usage will include a description of the power save modes and how to utilize them. A discussion on the impact of power, data latency and module status will be included.

3.10 Command Line Descriptions

This section will describe in detail the syntax, arguments and use of the available commands.

4.0 Supported Devices

This manual supports the Enterprise set of CLI commands across all platforms. Not all commands are supported on all platforms; the command descriptions in Section 19.0 provide guidance on which devices support it.

At the time of writing, the CLI command list represents the v1.40 release of the WLNG-XX-DP500 series of Airborne Device Server firmware. The part numbers supporting the commands described in this document include the following:

Part No.	Description
WLNG-SE-DP5XX	802.11b/g to RS232/422/485 and UART Serial Device Server Module, Enterprise Class
WLNG-AN-DP5XX	802.11b/g to UART Serial Device Server Module, Enterprise Class
WLNG-SP-DP5XX	802.11b/g to SPI Serial Device Server Module, Enterprise Class
WLNG-ET-DP5XX	802.11b/g to 10/100 Ethernet Bridge (NAT Level3) Module, Enterprise Class
WLNG-EK-DP5XX	Enterprise Class Airborne Development and Evaluation Kit
ABDG-SE-DP5XX	802.11b/g to RS232/422/485 Device Server, Enterprise Class
ABDG-ET-DP5XX	802.11b/g to 10/100 Ethernet Bridge (NAT Level3), Enterprise Class
ABDG-SE-HD5XX	802.11b/g to RS232/422/485 Heavy Duty Device Server, Enterprise Class
ABDG-ET-HD5XX	802.11b/g to 10/100 Heavy Duty Ethernet Bridge (NAT Level3), Enterprise Class
ABDG-ET-IN5XXX	802.11b/g to 10/100 Industrial Ethernet Bridge (NAT Level3), 5-36VDC , Enterprise Class
ABDG-SE-IN5XXX	802.11b/g to RS232/422/485 Device Server (Single and Dual Port), Ethernet, 5-36VDC , Enterprise Class

5.0 Overview

The module includes a Command Line Interface (CLI) Server. The CLI Server is the primary user interface for configuring, controlling, and monitoring the module. Users and OEM applications can establish CLI Sessions to the CLI Server via the serial interface or a TCP connection on the wireless and Ethernet interfaces.

This document describes the Command Line Interface extensions introduced or updated with the introduction of the Enerprise module (WLNG-XX-DP500 family). Since different Airborne™ modules differ in functionality, there may be differences in the use of the CLI for each particular device. These differences are clearly identified as part of this document.

There are four primary configurations supported by the module family: these are UART, Serial, SPI and Ethernet. Each device type will be described below. In some cases multiple interface options are available within a specific configuration; the functionality of these interfaces does not vary between device configurations unless specifically noted within the device description.

5.1 UART

The UART (Universal Asynchronous Receiver/Transmitter) interface is a digital interface that supports full-duplex transfer of data serially between the module and a connected host. It supports the following settings:

- BAUD: 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600, 115200, 230400, 460800, 921600
- Flow Control: None, Hardware (CTS/RTS), Software (XON/XOFF)
- **Default settings**: 9600, N, 8, 1, No Flow Control.

5.2 Serial

The Serial device includes both a UART interface control and I/O lines to manage external logic for RS232/422/485 line drivers. It supports the following settings:

- BAUD: 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600, 115200, 230400, 460800, 921600
- Flow Control: None, Hardware (CTS/RTS), Software (XON/XOFF)
- Mode (RS232/485), Tx Enable, Rx Enable.

Default settings: 9600, N, 8, 1, No Flow Control.

5.3 SPI

The SPI interface is a five (5) pin interface that supports full duplex operation. The module acts as a SPI slave and requires the master to supply the SPI clock. The default configuration for the interface is:

- Master SPI Clock: upto 8MHz

- Airborne SPI protocol (see WLNG DP500 Family Data Book, section 7.0 for details)

5.4 Ethernet

The module supports a fully-compliant 10/100 Ethernet interface capable of supporting all full- and half-duplex rates. The rates are configurable through the CLI interface.

The module includes a Broadcom BCM5241A Ethernet PHY, please refer to the manufacturers datasheet for interface details and appropriate design guidelines.

The interface supports the following settings:

- Auto Negotiate, 10Mbps Half Duplex, 10Mbps Full Duplex, 100Mbps Half Duplex, 100Mbps Full Duplex

Default settings: Auto Negotiate.

6.0 Understanding the CLI

CLI Sessions established to the CLI Server may operate in one of three modes: CLI, PASS, or LISTEN. Not all modes are supported on all interfaces of the device. A CLI Session established on the serial interface may operate in any of the three modes. CLI Sessions established on the wireless or Ethernet interfaces are restricted to CLI or PASS Modes.

6.1 Connecting to the CLI Server

Users may connect to the CLI Server on the serial interface using a terminal emulation program such as HyperTerminal or TeraTerm. The module default settings for the serial interface are:

- Bits per second: 9600
- Data bits: 8
- Stop bits: 1
- Parity: none
- Flow control: none

Users may also connect to the CLI Server on the wireless or Ethernet interface using a TCP client such as Windows Telnet. The Module's CLI Server supports a Telnet connection with the following restrictions:

- Telnet commands such as `DO`, `WONT`, and `DON,` must not be issued.
- Network Virtual Terminal codes are not supported.

The CLI Server's network interface is characterized as follows:

- The CLI Server listens on the TCP port specified by the `wl-telnet-port` parameter. The default is 23.
- The CLI Server inactivity timer is configured via the `wl-telnet-timeout` command.
- The CLI Server uses the `wl-telnet-timeout` value to timeout and close TCP connections that are inactive.
- The CLI Server supports multiple, simultaneous TCP sessions.

6.2 CLI Security

The CLI Server supports five (5) levels of security for each CLI Session. The security levels provide a safeguard for the set of CLI commands that may be executed by users. CLI Sessions that are authenticated at a particular security level may execute all CLI commands specified for that security level and below.

The Module's five (5) levels of security are:

- Level 0 (L0) = connectionless
- Level 1 (L1) = connection, not logged in (default)
- Level 2 (L2) = data

- Level 3 (L3) = config
- Level 4 (L4) = OEM
- Level 5 (L5) = Manufacturing (manuf)

Level 0 is the connectionless access level. Access over UDP will use this access level. The L0 level provides access to the name query services. It is not an authenticated level.

Level 1 is the default security level for CLI Sessions over TCP or the serial interface.

CLI Sessions must execute the CLI command `auth` in order to authenticate the CLI Sessions to another security level. The CLI command `logout` returns the CLI Session back to security Level 1.

6.3 CLI Session Modes

The mode of the CLI Session governs the set of actions allowed in the CLI session. The following are descriptions of each mode:

6.3.1 CLI Mode

CLI Mode is the command processing mode of the CLI Session. CLI Mode allows users and OEM applications to simply execute module commands as described in the section, “CLI Commands.”

A CLI Session may transition into CLI Mode automatically at startup of the CLI Session (if so configured). See section “CLI Session Startup Modes” for details on startup modes.

CLI Sessions may transition manually to CLI Mode from the other modes via the use of the CLI escape processing feature in the CLI Server. See section “CLI Server Escape Processing” for details.

6.3.2 PASS Mode

PASS Mode is an active data bridging mode of the CLI Server. PASS Mode allows the user or OEM application to transfer data between a CLI Session on the network interface and the CLI Session on the serial interface.

A CLI Session may transition to PASS Mode automatically at startup of the CLI session (if so configured) or manually from the CLI Mode using the CLI `pass` command. See section “CLI Session Startup Modes” for details on startup modes.

The transition from CLI Mode into PASS Mode differs depending on the attributes of the CLI session. The following sections describe the two PASS Modes.

6.3.3 PASS Mode for the Serial Interface

When the CLI Session on the serial interface attempts a transition to PASS Mode, the CLI Server establishes an outbound connection from the module to a user-specified TCP server and/or UDP server on the network interface. Once a connection is established, data bridging becomes possible between the CLI Session on the serial interface and the TCP Server and/or UDP server. If the connection to the primary TCP server failed, the CLI Server will attempt to connect to a secondary TCP server, if configured. If the transition to PASS Mode was triggered by the automatic startup configuration, the CLI Server will use the `wl-retry-time` configuration parameter to continuously retry connection to the servers.

The IP addresses of the primary TCP and UDP servers are configured using `wl-tcp-ip` and `wl-udp-ip` CLI commands. The secondary TCP server is configured using the `wl-tcp-ip2` command. The TCP server port is configured using `wl-tcp-port` and `wl-udp-port` CLI commands. The retry timer is configured using the `wl-retry-time` CLI command. See section “CLI Commands” for more details on these commands.

6.3.4 PASS Mode for a TCP CLI Session

When the CLI Session on the network interface (TCP CLI session) attempts to transition to PASS Mode, the CLI Server establishes a data bridge to the CLI Session on the serial interface if the following conditions are both true:

- The CLI Session on one or more of the serial interfaces is in LISTEN Mode.
- The number of CLI Session on the network interface, in PASS Mode, is less than the CLI sessions on the serial interfaces in LISTEN mode.
- If more than one of the Serial interfaces is in LISTEN mode, it is possible to direct the TCP CLI Session PASS mode connection to either of the available sessions.

6.3.5 LISTEN Mode (Serial/UART/SPI Interface Only)

LISTEN Mode is a passive data bridging mode of the CLI Session. The LISTEN Mode is only applicable on the serial, UART and SPI interfaces. When the CLI Session on the serial interface enters LISTEN Mode, the module passively waits for a data bridge to be established from a TCP CLI session. The data bridge may be initiated using a CLI Session via the PASS Mode or using the tunneling feature. The CLI Session may transition to CLI Mode using CLI Server escape processing. See section “CLI Server Escape Processing” for details.

When the serial interface CLI Session is in LISTEN Mode, the following are possible:

- TCP connections on the network interface can use the CLI commands `pass`, `putget` or `putexpect` to establish a data bridge.
- TCP connection can establish a data bridge if tunneling is enabled.

6.3.6 CLI Session Startup Modes

The startup behavior of the CLI Session on each interface is determined as follows:

- The CLI Session on the serial interface startup behavior is determined by the value of the `serial-default` parameter.
- CLI Sessions on the network interface using the TCP port specified by `wl-telnet-port` always start in CLI Mode.
- CLI Sessions on the network interface using the TCP port specified by the `wl-tunnel-port` or the UDP port specified by `wl-udp-rxport`, always start in PASS Mode. However, if the CLI Session on the serial interface is not in LISTEN Mode, the TCP connection on the `wl-tunnel-port` will be rejected by the Module.
- Each of the serial ports can have a different CLI Session startup behavior.
- Each serial port can have different configuration settings for the tunnel port.

6.4 CLI Server Escape Processing

The CLI Server includes an escape processing feature which allows CLI Sessions to transition from PASS or LISTEN (data bridging) Mode back to CLI Mode. Escape processing is configurable to:

- disable escape processing
- process the receipt of a user-defined escape string as an escape signal
- process the receipt of the BREAK signal as an escape signal

When escape processing is disabled, the CLI Server will not parse the data stream for any escape sequence. When escape processing is configured to use an escape string, the CLI Server will perform pattern matching for the user-defined escape string in the data stream. The escape sequence must be the last characters delivered to the module for escape parse to be successful. The escape string is a five (5)-character string configurable via the `escape` or `esc-str` CLI commands. When escape processing is configured to use the BREAK signal, the CLI Server will parse the data stream for the BREAK signal.

The `esc-str` CLI command supersedes the `escape` command. It is recommended that the `esc-str` be used.

6.5 Detecting and Executing the Escape Sequence

Upon detection of the escape sequence, the CLI Server applies the follow rules for transitions of the CLI Session on that interface:

- If the CLI Session is in LISTEN Mode and there is no data bridge established, the CLI Session will transition to CLI Mode and send an `OK` response to the CLI Session.
- If the CLI Session is in LISTEN Mode and there is an active data bridge established, the CLI Server will terminate the active data bridge and the CLI

Session will remain in LISTEN Mode. Note that, two escapes are required to transition from active data bridge to CLI mode.

- If the CLI Session is in PASS Mode, the CLI Server will send an `OK` response to the CLI Session and transition to CLI Mode.

The following effects of escape processing require the attention of system implementations:

- If the escape sequence is an escape string, the escape string received on one CLI Session is transmitted to the CLI Session on the other end of the data bridge prior to performing the CLI Session transition. This allows the other end to parse the received data and determine when the data bridge is shutdown.
- If the escape sequence is the BREAK signal, the BREAK received on the serial interface is not transmitted to the wireless interface, but the transition takes place internally.
- The CLI Session that detects the escape sequence will post an `OK` response on its interface if the escape sequence caused the CLI Session to transition to the CLI Mode.
- Escape detection does not close the TCP connection. It only terminates the data bridge. Subsequent use of the `pass` CLI command will re-establish the bridge for that interface.

The CLI Server allows independent configuration of escaping processing for each serial port and for TCP CLI session. The serial interface escape processing is configurable using the CLI parameter `esc-mode-serial`. The TCP CLI Session escape processing is configurable using the CLI parameter `esc-mode-lan`. See section “CLI Commands” for details on these parameters.

6.6 CLI Conventions

The CLI uses the following conventions:

- All commands consist of a string of printable characters, including the command and optional arguments delimited by one or more spaces or tabs. Multiple consecutive spaces or tabs are generally considered as one delimiter.
- Commands and arguments are case sensitive, except hexadecimal values and port IDs, which can be uppercase or lowercase.
- Arguments enclosed within […] are optional.
- All arguments are literal ASCII text, except where indicated.
- Most commands that set the value of a parameter can also obtain the value of the parameter by omitting the argument. Numeric values are returned in aschex format.
- A choice between arguments is indicated with the **|** character. Only one of the choices can be selected.
- All CLI commands are terminated with a <CR>.
- The maximum length of a CLI command line is 256 characters, including spaces and terminating characters.

- Argument types include:
  - <*ASCII Text*> – literal ASCII character string without delimiters (no spaces or tabs).
  - <*integer*> – value represented as a decimal integer or as “aschex” value in the form 0xhhh…hhh.
  - <aschex> – one or more pairs of hexadecimal digits with no prefix in the form hhh…hhh.
  - <*portid*> – an I/O port bit number, from 0 to 7.
  - <*IPadrs*> - Internet Protocol address string in the format: *nnn*.*nnn*.*nnn*.*nnn*; for example: 192.168.10.3 .

6.7 ASCHEX vs. Binary Values

Data can be sent to the module as either binary data or a hexadecimal representation of the actual data being transmitted.

When a LAN device or serial port Host issues a `pass` command, the data is transmitted as binary data. By comparison, when the command `putget` or `putexpect` is issued, the `senddata` content must be encoded as ASCII hexadecimal digit pairs. The data is translated across the Module and received as an ASCII representation of the actual data. This is true whether the transmission initiates from the LAN device or from the Host.

For example, the digits 31 correspond to the ASCII character 1. If you issue a `putget` or `putexpect` command with the `senddata` value of 314151, the destination receives the ASCII characters **1**, **A**, and **Q**.

6.8 Command Responses

The Module responds to CLI commands with a response indicating whether the CLI command was executed successfully. All responses are terminated by `<CR><LF>`.

Multiline responses have each line terminated with `<LF><CR>` with the response terminated by `<CR><LF>`..

After the Module executes a CLI command successfully, it returns the response:

```
OK<CR><LF>
```

Otherwise, it returns an error response. Error responses are returned in the following general format:

```
Error 0xhhhh: error text<CR><LF>
```

In the response the aschex value is the error code. A summary of error code can be found in section 20.0.

> The TCP CLI interface by default echoes back CLI session input. It is possible to turn this feature off by issuing the `telnet-echo disable` command.

7.0 A Typical Development System

A typical evaluation system includes:

- A Serial Host: A computer connected to of the serial ports of the Airborne™ Enterprise Development Board.
- A LAN Host: A computer that communicates wirelessly with the Module through an Access Point (AP).
- An Access Point.
- An Airborne™ Enterprise Development kit.

8.0 Serial Device Server Use

In this section the base functionality of the module will be described, examples of use and configuration will be provided. Refer to this section to understand the differences between a command port, data tunnel, TCP/IP vs. UDP use and server vs. device operation.

The UART, Serial and SPI versions of the module provide the ability to connect a raw serial data stream to a TCP/IP based network, using 802.11 or Ethernet as the primary network connection media. To facilitate this functionality the module supports a number of management and data bridging interfaces on both the serial (Serial/UART/SPI) and network (802.11/Ethernet) interfaces. As described in section 3.2, there are multiple states for the CLI interface; this section will describe the data bridging options and the required CLI configuration for each.

8.1 Data Bridging

The module provides data bridging via the PASS and LISTEN Modes of the CLI Session. During data bridging, the raw payload of an incoming TCP or UDP packet is transmitted to the serial interface while the raw data stream from the serial interface is transmitted as the payload of an outgoing TCP or UDP packet.

There are multiple ways to setup a data bridge using the module. A bridge may be initiated from the Serial Host, from a TCP connection on the `wl-telnet-port`, from a TCP connection on the `wl-tunnel-port`, from a UDP message on the `wl-udp-rxport` or from a Secure Shell (SSH) connection on the `wl-ssh-port`.

Only one CLI session on the network (802.11/Ethernet) interface may be bridged with any single CLI session on the serial interface at a time.

8.1.1 Bridging from the Serial Interface

The CLI Session on a serial interface may initiate a data bridge via the use of the `serial-default` parameter set to “pass” or by manually issuing the `pass` CLI command. Prior to establishing the data bridge, the module must be properly configured to connect to a server on the network that will accept the communications; table X identifies the parameters that need to be set.

Table 1 - CLI Session Default PASS mode parameters

Command	Description
`pass`	Creates a data bridge between the network and serial interface. When issued from the serial CLI session the CLI server initiates a TCP connection using the IP, port and timeout parameters defined for the serial interface issuing the command. This command supports the serial port suffix –`p1` or `-p2`, however they will only apply if issued on the serial port referenced in the suffix. If the suffix is not included, the command applies to the port the serial CLI session is open on.
`serial-default-pX pass`	Configures the default setting for a serial port to behave as if a pass command had been issued by the serial interface CLI session. Creates a data bridge between the network and serial interface. When issued from the serial CLI session the CLI server initiates a TCP connection using the IP, port and timeout parameters defined for the serial interface issuing the command. This command supports the serial port suffix, by replacing the `pX` with `p1` or `p2` the command parameter can be applied to a specific serial port. If the suffix is not included, the command applies to the port the serial CLI session is open on.
`wl-tcp-ip-pX [IP Address]`	The primary target IP address of the TCP server on the network to be used when the CLI session on a serial port issues the PASS command or if the serial-default setting is PASS. If the IP address is empty or the connection attempt is unsuccessful the CLI server will attempt to connect to the IP address defined by `wl-tcp-ip2` (Secondary target IP) This command supports the serial port suffix, by replacing the `pX` with `p1` or `p2` the command parameter can be applied to a specific serial port. If the suffix is not included, the command applies to the port the serial CLI session is open on.

Command	Description
`wl-tcp-ip2-pX [IP Address]`	The secondary target IP address of the TCP server on the network to be used when the CLI session on a serial port issues the PASS command or if the serial-default setting is PASS. This command supports the serial port suffix, by replacing the `pX` with `p1` or `p2` the command parameter can be applied to a specific serial port. If the suffix is not included, the command applies to the port the serial CLI session is open on.
`wl-tcp-port-pX [Port Number]`	The port number used by the CLI server when a serial interface initiates a TCP connection. This value must match the port on which the target TCP server is listening. The port range is 0 – 65535 (default 2571). This command supports the serial port suffix, by replacing the `pX` with `p1` or `p2` the command parameter can be applied to a specific serial port. If the suffix is not included, the command applies to the port the serial CLI session is open on.
`Wl-tcp-timeout-pX [Time seconds]`	Establishes the inactivity timeout for a TCP connection initiated by the CLI session on a serial interface using the `pass` or `serial-default pass` command. A value of `0` disables the timeout. This command supports the serial port suffix, by replacing the `pX` with `p1` or `p2` the command parameter can be applied to a specific serial port. If the suffix is not included, the command applies to the port the serial CLI session is open on.

The following examples illustrate how to configure the Module to initiate a connection to a TCP server:

Figure 1 - Bridging from a serial Interface Manually Using the pass Command

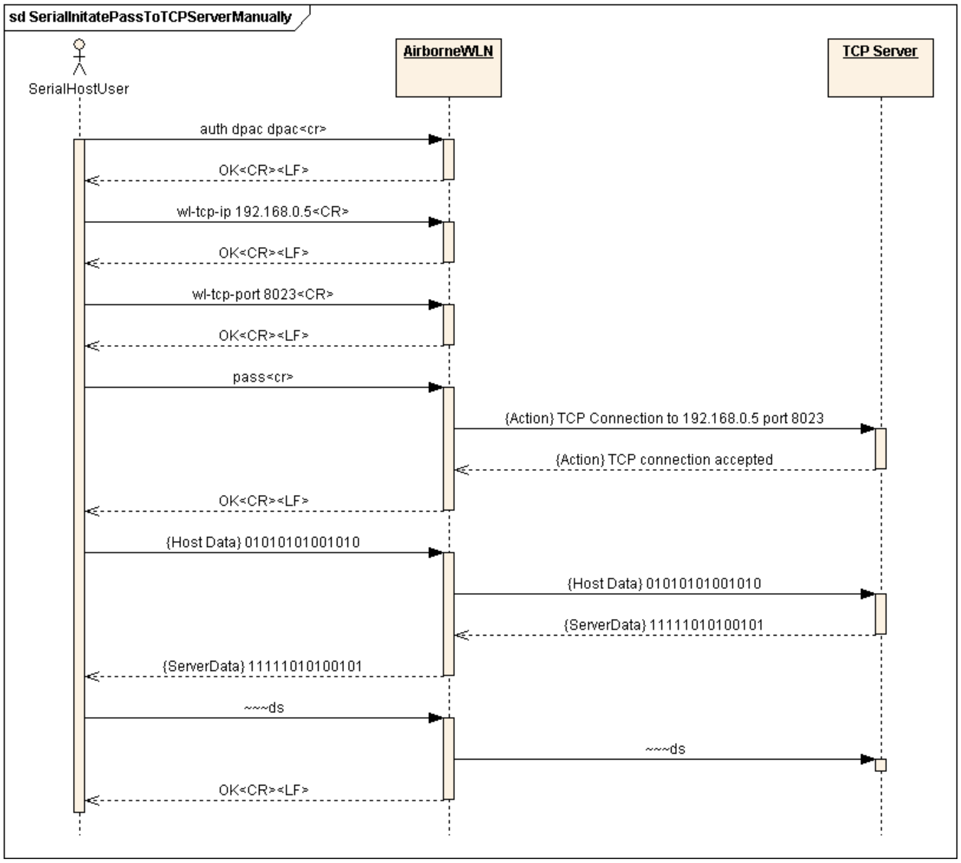
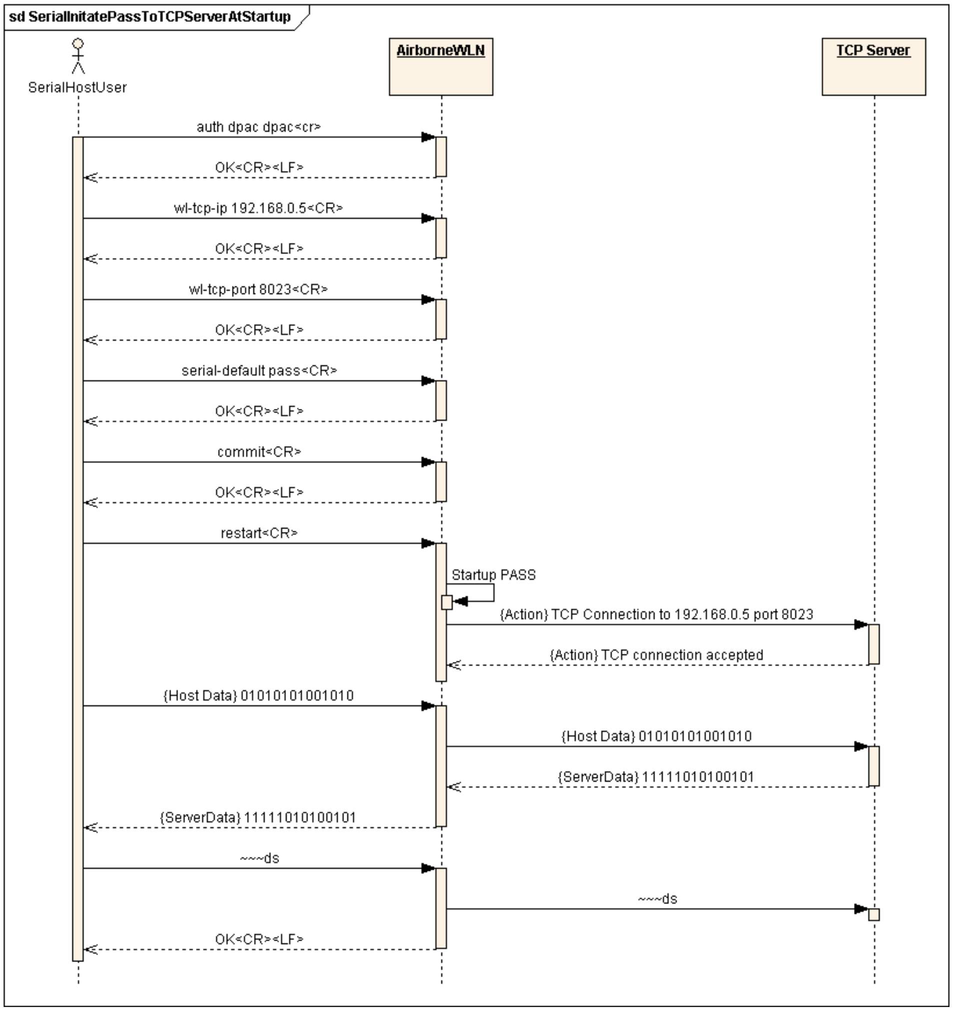

sd SerialInitatePassToTCPServerManually
SerialHostUser
AirborneWLN
TCP Server
auth dpac dpac<cr>
OK<CR><LF>
wl-tcp-ip 192.168.0.5<CR>
OK<CR><LF>
wl-tcp-port 8023<CR>
OK<CR><LF>
pass<cr>
{Action} TCP Connection to 192.168.0.5 port 8023
{Action} TCP connection accepted
OK<CR><LF>
{Host Data} 01010101001010
{Host Data} 01010101001010
{ServerData} 11111010100101
{ServerData} 11111010100101
~~~ds
~~~ds
OK<CR><LF>

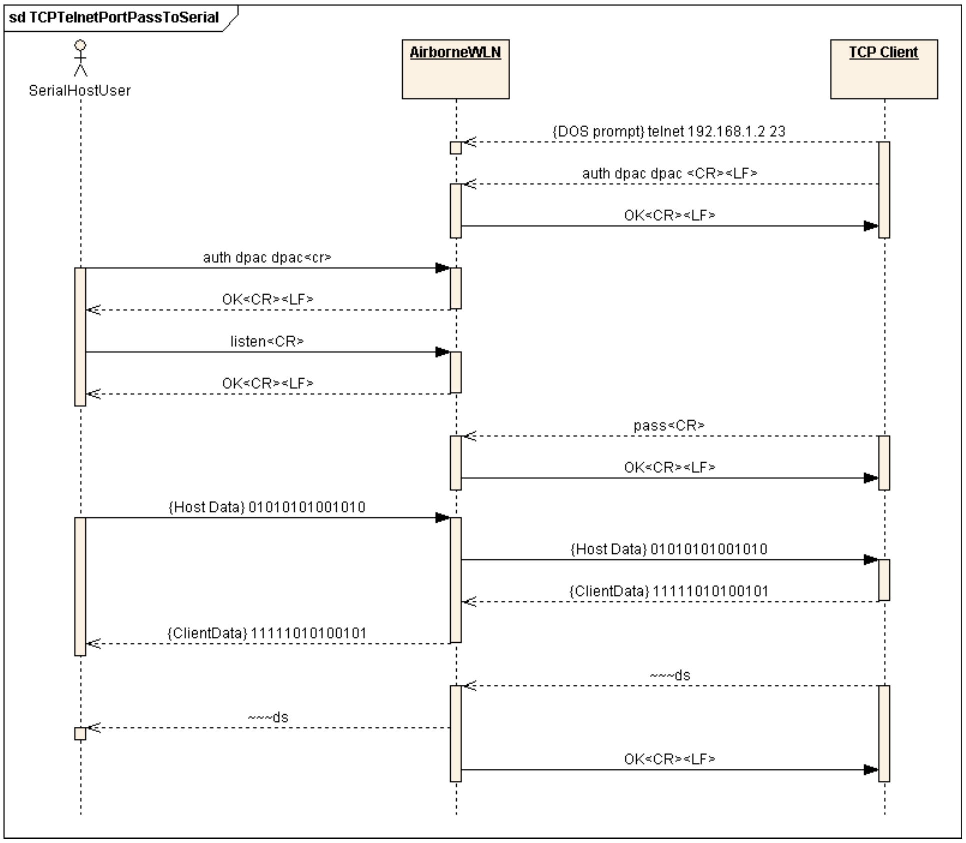
Figure 2 - Bridging from a serial Interface Automatically at Startup Using the Serial-Default Command


sd SerialInitatePassToTCPServerAtStartup
SerialHostUser
AirborneWLN
TCP Server
auth dpac dpac<cr>
OK<CR><LF>
wl-tcp-ip 192.168.0.5<CR>
OK<CR><LF>
wl-tcp-port 8023<CR>
OK<CR><LF>
serial-default pass<CR>
OK<CR><LF>
commit<CR>
OK<CR><LF>
restart<CR>
Startup PASS
{Action} TCP Connection to 192.168.0.5 port 8023
{Action} TCP connection accepted
{Host Data} 01010101001010
{Host Data} 01010101001010
{ServerData} 11111010100101
{ServerData} 11111010100101
~~~ds
~~~ds
OK<CR><LF>

8.1.2 Bridging from a TCP connection on the wl-telnet-port

A user or OEM application connected over TCP to the `wl-telnet-port` of the module may create a data bridge to a serial interface by issuing the `pass` command. The `pass` command will succeed if there is no other data bridge active and the CLI Session on a serial interface is in LISTEN Mode. The following figure illustrates a sequence of commands that create a data bridge from the TCP connection:

Figure 3 - Bridging from a TCP Connection on the wl-telnet-port


sd TCPTelnetPortPassToSerial
SerialHostUser
AirborneWLN
TCP Client
{DOS prompt} telnet 192.168.1.2 23
auth dpac dpac <CR><LF>
OK<CR><LF>
auth dpac dpac<cr>
OK<CR><LF>
listen<CR>
OK<CR><LF>
pass<CR>
OK<CR><LF>
{Host Data} 01010101001010
{Host Data} 01010101001010
{ClientData} 11111010100101
{ClientData} 11111010100101
~~~ds
~~~ds
OK<CR><LF>

8.1.3 *Bridging from a TCP connection on the wl-tunnel-port*

The module supports a tunneling feature that allows bridging between a specific TCP address/port and the module's serial port without requiring authentication with the module. TCP port tunneling is supported by the `wl-tunnel`, `wl-tunnel-mode`, and `wl-tunnel-port` commands. The rules for TCP connections to the `wl-tunnel-port` are as follows:

- `wl-tunnel` must be enabled (set to `1`).
- `wl-tunnel-mode` must be set to `tcp`.
- `wl-tunnel-port` must be set to a non-zero value which is not the same as any previously defined port on the module.
- The CLI Session on a serial interface must be in LISTEN Mode.
- There must be an available serial interface in LISTEN mode, which is not already bridged.

If all of the conditions are met, this TCP connection will become the active bridge. All data payload will be bridged between the CLI Session on a serial interface and the CLI Session on this TCP port.

The data bridge may terminate for any one of the following reasons:

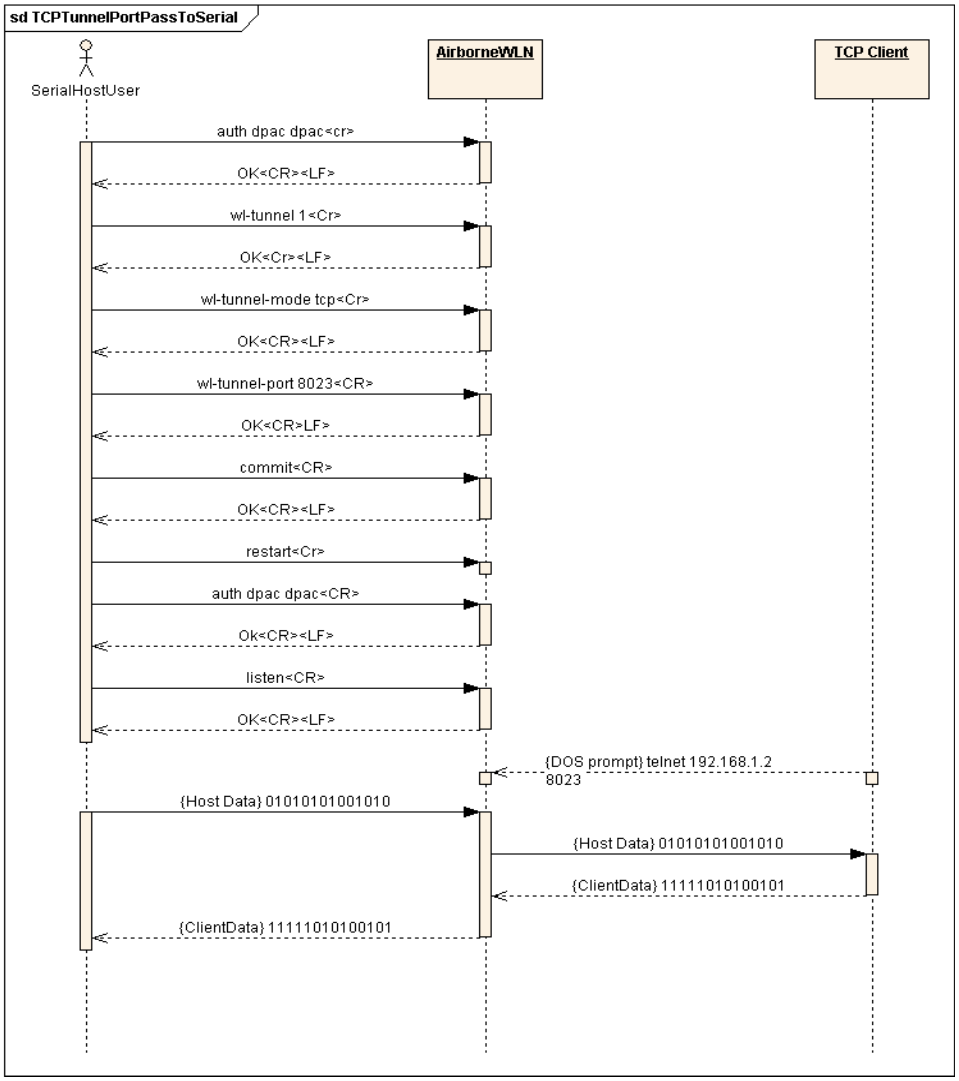
- The `close` CLI command is issued from a secondary network CLI session.
- The `radio-off` CLI command is issued from a secondary network CLI session.
- The network server or host terminates the TCP/IP or UDP session.
- The TCP/IP connection inactivity timer (`wl-tcp-timeout`) expires.
- The escape sequence is detected.

After the data bridge is terminated, the CLI Session on a serial interface remains in LISTEN Mode and escape detection, if configured, is enabled.

Since a tunnel connection does not require authentication to the module it is less secure than other connection type, like SSH or telnet. The tunnel port can only be used for a data connection; it does not support access to the CLI server.

Using the following sequence, a user can configure the module to operate in TCP tunneling mode:

Figure 4 - Bridging From a TCP Connection on the wl-tunnel-port



8.1.4 Bridging Using UDP

The module supports UDP tunneling. This allows the module to forward data from a serial interface to a specific server listening on a specified UDP port or to broadcast a UDP datagram on a specific UDP port. This also allows the module to forward data received on its specified UDP receive port to a serial interface.

The UDP port tunneling feature is configurable via the `wl-tunnel`, `wl-tunnel-mode`, `wl-udp-xmit`, `wl-xmit-type`, `wl-udp-rxport`, `wl-udp-port`, and `wl-udp-ip` CLI commands.

Whenever the CLI Server transitions to PASS Mode either via the startup `serial-default pass` parameter or the `pass-p?` command, the module will use the UDP tunneling configurations to operate the UDP data bridge as follows:

- `wl-xmit-type` is used to enable UDP transmission of data from a serial interface.
- `wl-udp-xmit` is used to enable unicast, or broadcast UDP datagram transmission, or both.
- `wl-udp-ip/wl-udp-port` is used to set the UDP transmission destination IP address/port.
- `wl-udp-rxport` sets the UDP port that the module will receive data on for the bridge.

> If `wl-xmit-type` is set for `both`, then the TCP bridge must remain active for the UDP bridge to remain active. If the TCP server becomes inactive, the UDP bridge will be terminated.

> Only the data payload of the UDP packet is forwarded to a serial interface. All serial data received is sent as the UDP packet payload.

8.1.5 Data Bridging with XMODEM Guidelines

Once a data bridge is established, the endpoints may transfer raw binary data. Some systems may choose to apply a protocol such as ZMODEM or XMODEM, etc.

For systems using XMODEM protocol, the following guildelines must be adhered to:

- XMODEM works with 8-bit connections only. If you communicate with the Module via a serial port connection, configure your communication settings as follows:
  - Data bits: 8
  - Parity: None
  - Stop bits: 1
- Run XMODEM with either no flow control or hardware (RTS/CTS) flow control because the protocol provides no encoding or transparency of control characters. If you run XMODEM with software (XON/XOFF) flow control, your connection will hang. For this reason, configure the flow control parameter in your communication settings to NONE or RTS/CTS, not to XON/XOFF or BOTH.
- During transmission, XMODEM pads files to the nearest 128 bytes. As a result, original file sizes are not retained.

> These guideline apply to the use of Xmodem during firmware, certificate, Private key and configuration file upload to the device server.

8.1.6 Bridging from a SSH connection on the wl-ssh-port

The module supports secure CLI operation and data bridging through use of a Secure Shell (SSH) CLI Session. This feature behaves very similarly to a TELNET CLI Session (see Section 8.1.2). To access the SSH port the connection must use the `wl-ssh-port` value (default `22`), in addition the SSH server must be enabled and correctly configured.

In order to enable use of SSH CLI Sessions it is necessary to perform the following steps to prepare the module for accepting SSH connections:

Table 2 - SSH Initial Configuration

Command	Description
Decide SSH Keysize `ssh-keysize`	The module's administrator must decide the strength of the SSH encryption to use. This is generally a customer site-specific policy (ask your IT department) and is reflected in the value of `ssh-keysize`. The default value of 1024 makes use of 1024-bit RSA public/private key pairs, and is a good compromise of performance vs. strength. The maximum value of 2048 takes significant time both to generate the public/private key pair and to establish connections with the SSH server.
Generate SSH key on module `ssh-keygen`	The RSA public/private key pair used by SSH must be generated by the `ssh-keygen` command. This command can take several minutes to complete, but need only be performed once per module.
Save the generated key `commit`	After the RSA public/private key pair is generated, they must be used to the module's FLASH to be persistent across restarts. If they are not saved they will need to recalculated before the SSH port can be used.
Restart or power cycle the module `restart`	The module must be restarted or power cycled to launch the SSH server. Aftre the module has been restarted the SSH server will then listen to incoming SSH client requests on `wl-ssh-port`. The configuration of `ssh-port` is `off` until keys are generated and commited.

For an SSH client program, Quatech has verified proper operation of TeraTerm, PuTTY and OpenSSH.

The modules own internal SSH client has also been verified.

The first time a given SSH client on a given workstation attempts to connect with the module's SSH server, the SSH client will identify that the SSH

Client/Workstation has not connected to the module before and will ask the user to accept the connection. If the connection is accepted the credentials (RSA public key which was generated in Table 2) will be saved for use with subsequent connections.

> If the module is configured for DHCP on the network interface being used the SSH client will consider it a "new" module any time it's assigned IP address changes and require that the username and password be reentered, even if that client has successfully connected to that module before.

Authentication via the SSH client is functionally identical to authentication over the module's Debug Port. The module's SSH server will prompt the SSH Client for a user name, and the SSH client will accordingly request the user to login and provide a username (actual input request is determined by the SSH Client being used) a similar prompt. After the desired username is entered, the modules SSH server will prompt for the corresponding password. The username and password are the same as used for the CLI `auth` command. Once the password challenge is successful, the user will be in a standard CLI Session, just as if initiated over TELNET. There is no need to re-enter the `auth` command in the CLI Session; the SSH login procedure already securely identified the user to the module.

All CLI commands available to a TELNET CLI Session are available to a SSH CLI Session; establishing a data bridge to a serial interface is identical to the steps described in Section 8.1.2.

8.1.7 Bridging using SSH

The module supports module-initiated secure data bridging through use of a Secure Shell (SSH) tunnel. This feature behaves very similarly to TCP `pass` communication (see Section 8.1.1).

In order for the module to communicate with an SSH server, the same key-generation preparation is necessary as for use of SSH CLI Sessions. This is described in Table 2.

> For an SSH server program, Quatech has verified proper operation of OpenSSH with the module's built-in SSH client.
>
> The modules own SSH server has also been verified.

The first time the module attempts to communicate with a given SSH server, it will, by default, not *trust* that server and will refuse to connect.

This is proper security protocol to avoid SSH server-identity theft. To tell the module that it is acceptable to connect to a previously-unknown SSH server, you must issue the CLI command `ssh-trust 1`. This instructs the module to automatically *trust* new SSH servers until either the CLI command `ssh-trust 0` is issued, or the module is restarted (for security purposes, `ssh-trust 0` is always set after a restart).

A `commit` command must be used to save the SSH server credentials to the module, this will make them persistent across restarts or power cycles.

If the credentials are not saved the module/server will need to be re-trusted the next time the module restarts.

Use of SSH for `pass` data bridging is configured by setting `wl-xmit-type ssh` (for the primary serial/UART interface) or `wl-xmit-type-p2 ssh` (for the secondary serial/UART interface).

If the user is communicating with the module over a CLI Session on a serial interface, when authenticating with the SSH server, the username and password utilized by the modules SSH client is the same as that with which the user entered when the `auth` command was issued at the start of the CLI Session. If the module is automatically establishing the data bridge via `serial-default pass` or `serial-default-p2 pass`, the username and password configured through `ssh-default-user` and `ssh-default-password` are utilized.

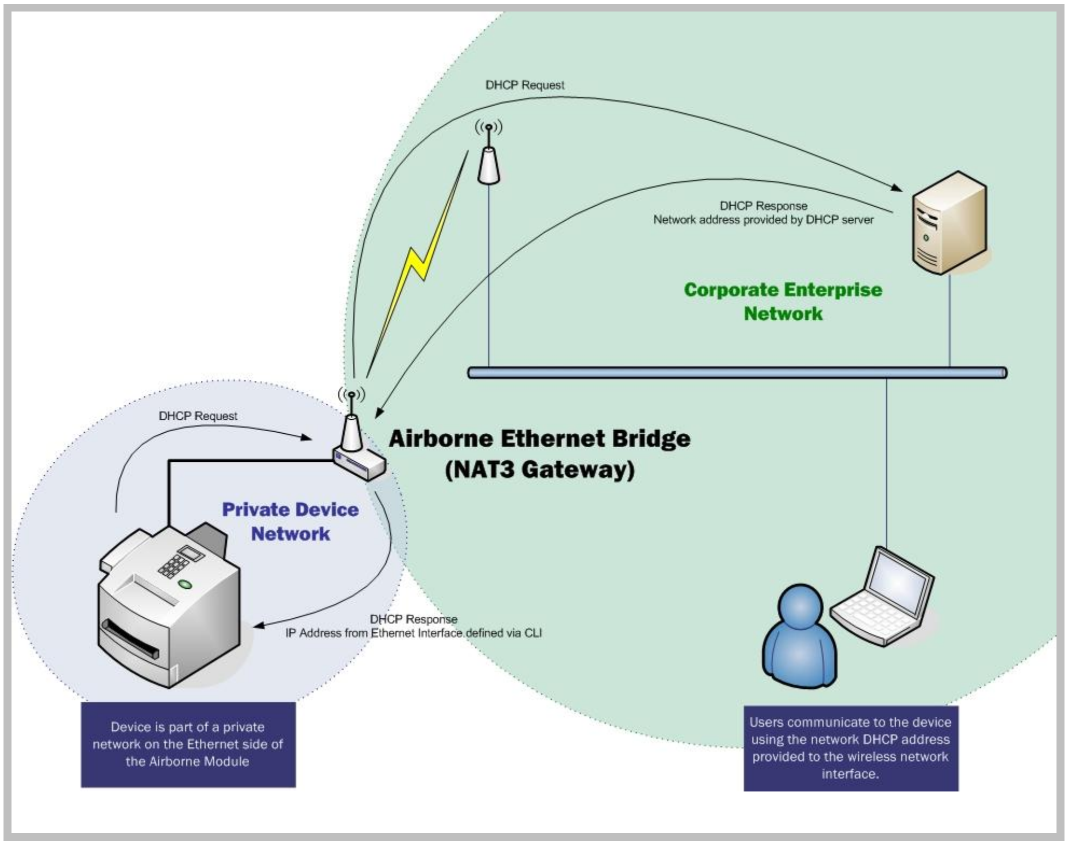
9.0 Ethernet Adapter Use

The Airborne Ethernet Adapter is a fully functional NAT Level 3 router, supporting a public IP address for the wireless interface and a private network for the attached devices on the wired interface.

Network Address Translation (NAT) is the process of modifying network address information in Internet Protocol (IP) packet headers while in transit across a traffic routing device for the purpose of remapping a given address space into another. In the case of a NAT Level 3 device, the modification of the packet headers provides for a translation between a single public IP address (that of the wireless interface) and the IP addresses of the devices on the private network (wired Ethernet interface).

The modules wireless interface is considered the public address and will be the point of contact on the target network (see Figure 5). This interface supports all the wireless and network authentication requirements, including support for WPA2-Enterprise. It can acquire an IP address either through DHCP or user configured static IP. Once configured, association and authentication are handled entirely by the module and require no interaction from a Ethernet client on the private network.

Figure 5 - Ethernet Bridge Functionality


DHCP Request
DHCP Response
Network address provided by DHCP server
Corporate Enterprise Network
DHCP Request
Airborne Ethernet Bridge (NAT3 Gateway)
Private Device Network
DHCP Response
IP Address from Ethernet Interface defined via CLI
Device is part of a private network on the Ethernet side of the Airborne Module
Users communicate to the device using the network DHCP address provided to the wireless network interface.

The Private network is the wired interface provided by the bridge. This interface includes a DHCP server and supports dynamic or static IP address assignment. This means any Ethernet client supporting DHCP can be connected to the wired interface without configuration changes. The private network host can communicate with the module using the bridges Ethernet IP address on the private network.

The modules Ethernet personality supports NAT Level 3 and as such provides the following advantages over the more traditional bridge functionality:

- A single network IP address on the public network. This simplifies management of the devices on the network and avoids issues with some network infrastructure that does not permit a single device to have multiple IP addresses.
- A single point of authentication. The module handles authentication for the public network; this means a single point of contact for all security interaction, simplifying deployment for the network.
- Zero security footprint on the private network host.
- Support for DHCP and static IP on the private network. This capability allows the host to be shipped without any configuration changes.
- Port forwarding. Allows you to decide if web page, TELNET or FTP access should be forwarded to the private network or handled by the module.
- Plug-n-Play. In most cases all that is required for full functionality is configuration of the wireless interface for the target network. This can be done before deployment to minimize deployment time and complexity.

9.1 Public Network Interface

The public network interface is the module's wireless port. This interface must be configured to associate and authenticate with the target network. To successfully configure this interface the following must be configured correctly:

Table 3 - Public Network Configuration

Command	Description
`wl-ssid`	This identifies the target network for the Ethernet bridge.

<table>
<tr><th>Command</th><th>Description</th></tr>
<tr><td>wl-dhcp</td><td>This defines whether or not the device will use DHCP or a static IP address. This address will become the target address for any devices on the network wanting to communicate with the bridge or the device attached to the wired interface.

If DHCP is not being used it is necessary to configure the following parameters:

<table>
<tr><td>wl-ip</td><td>Module Static IP address</td></tr>
<tr><td>wl-subnet</td><td>Subnet mask</td></tr>
<tr><td>wl-gateway</td><td>Network gateway IP address</td></tr>
<tr><td>wl-dns1</td><td>Primary DNS server IP address</td></tr>
<tr><td>wl-dns2</td><td>Secondary DNS server IP address</td></tr>
</table></td></tr>
<tr><td>Security (various commands)</td><td>It is necessary to configure this interface for the appropriate security profile required for authentication to the target network. Please see section 10.0 for details on configuring the security profile.</td></tr>
<tr><td>http-port</td><td>This parameter allows directed traffic on the defined http port to be directed to either the Airborne device server or the device connected on the wired port.

If enabled all traffic on the http port will be handled by the Airborne device.

If the application requires that a web server on the host, attached to the wired port, respond to web page accesses this parameter must be disabled or turned off, alternately the wl-http-port must be changed from the default port to another which does not conflict with the devices http port on the Ethernet interface.</td></tr>
<tr><td>telnet-port</td><td>This parameter allows directed traffic on the configured telnet port to be directed to either the Airborne device server or the device connected on the wired port.

If enabled, all traffic on the telnet port will be handled by the Airborne device.

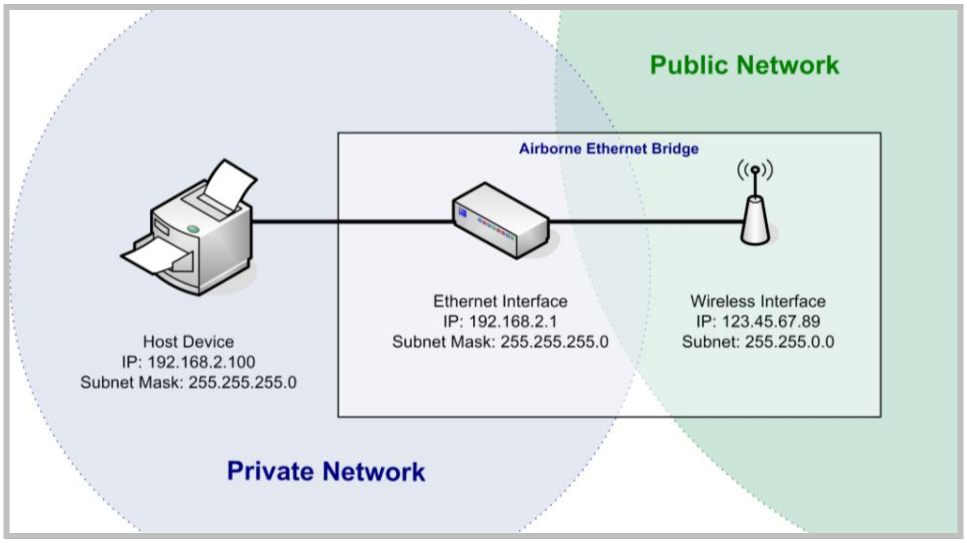
If the application requires that a telnet server on the host, attached to the wired port, respond to remote accesses this parameter must be disabled.</td></tr>
<tr><td>ssh-port</td><td>This parameter controls the availability of the modules SSH server. The SSH port (wl-ssh-port) availability will depend upon the setting for this parameter.

If enabled, all traffic on the SSH port will be handled by the Airborne device.</td></tr>
</table>

The public address becomes the target address for all accesses to the Ethernet clients connected to the private network. In the example shown in Figure 6, any device on the public network wanting to communicate with the Ethernet client (1st Host Device IP: 192.168.2.100), would use the IP address 123.45.67.89, the module will forward all traffic to the private address 192.168.2.100.

The network infrastructure will show the MAC and IP address of the modules wireless interface as the network presence, as a consequence of this all traffic will be identified as being from or to this address.

Figure 6 - Airborne Ethernet Bridge IP Configuration



The public network interface supports the Airborne™ discovery protocol and will respond to discovery requests issued on the public network. Discovery protocol requests are not forwarded to the private network.

9.2 Private Network Interface

The private network interface is on the Ethernet port of the module. The interface supports multiple Ethernet clients with either a static or DHCP sourced IP address. This interface needs minimal configuration and requires the parameters in Table 4 to be configured.

Table 4 - Private Network Interface Configuration

Command	Description
`eth-ip`	This is the base IP address of the private network DHCP server address pool, and is the first IP address the DHCP server will lease to a client on the private network when the client is using DHCP. It is also the default private network IP address used for forwarding traffic from the public network. This address must match the private network client IP address when a single client is attached and is using a static IP address. If this does not match the address, traffic from the public network will NOT be routed correctly. Traffic originating from Ethernet clients will be routed correctly.
`eth-subnet`	This is the subnet mask the DHCP server will provide to the client when the client is using DHCP.
`eth-gateway`	This is the IP address of the Ethernet Interface on the Airborne Ethernet Bridge and is the target address for communications between the Ethernet client and the Airborne Bridge.

<table>
<tr><th>Command</th><th>Description</th></tr>
<tr><td>eth-mode</td><td>The Ethernet interface supports the following configurations; this parameters determines the default mode of the interface.
<table>
<tr><td>auto</td><td>Auto negotiate</td></tr>
<tr><td>10half</td><td>10Mbps, half duplex</td></tr>
<tr><td>10full</td><td>10Mbps, full duplex</td></tr>
<tr><td>100half</td><td>100Mbps, half duplex</td></tr>
<tr><td>100full</td><td>100Mbps, full duplex</td></tr>
</table>
It is recommended that auto be used as this will provided the greatest level of compatibility on the Ethernet interface.</td></tr>
</table>

The private network supports the Airborne™ discovery protocol and will respond to discovery requests on the private network. Discovery protocol requests are not forwarded to the public network.

The subnet for the private network IP addresses (Ethernet Client and Gateway) and public IP address (802.11), obtained by the module via the wireless interface, **MUST NOT** be the same.

Failure to observe this requirement will result in unpredictable behavior of the bridge.

When attempting to make an out-bound connection to a device on the public network, the public network IP address of the device should be used e.g. In Figure 6 the client with address 192.168.2.100 wants to connect to an FTP server, with the address of 123.45.67.99, on the public network to perform a firmware download. The FTP address that would be used in the `ftp-server-address` parameter would be `123.45.67.99.` Note that this is not within the subnet of the Ethernet client, however the NAT router will do the necessary address translations and packet header manipulations to ensure the out-bound and in-bound connections are maintained.

Any traffic between the Airborne Ethernet Bridge Ethernet interface and Ethernet client, on the private network, will not be broadcast on to the public network unless it is directed at the public network.

For most users there will be no modification of the private network settings needed and if the target Ethernet client uses DHCP to obtain an IP address, no change in configuration will be required either.

9.3 Ethernet Firewall Configuration

The module has an in-built rule based firewall, designed to provide a simple solution for limiting access on the network the wireless interface is associated with to just the resources required for the target application. When configured this prevents any system using the Ethernet interface for accessing unauthorized data or resources, protecting the connected network from illegal use by an rogue Ethernet Client.

To utilize the firewall the module must be configured to allow traffic from the Ethernet interface to the wireless interface based upon IP traffic rules, these rules include the ability to block or allow access based upon target IP address, protocol and port. The module supports the use of multiple rules and applies them based upon the priority in the rule list. Priority of the list is based upon the order in which the rules were entered, first being highest last being lowest.

Configuring the firewall requires a use of the commands listed in Table 5.

Table 5 - Ethernet Firewall Commands

<table>
<tr><th>Command</th><th>Description</th></tr>
<tr><td><code>eth-route-default <access&gt;</code></td><td>This sets the default firewall settings.
<table>
<tr><td><code>accept</code></td><td>All packets are relayed to the wireless interface.</td></tr>
<tr><td><code>drop</code></td><td>All packets are dropped and are not relayed to the wireless interface.</td></tr>
</table>
If <code>&lt;access&gt;</code> configured for <code>accept</code> all outgoing requests will be forwarded, except broadcast messages, essentially turning off the firewall. Relaying of broadcast messages must be explicitly enabled with the firewall rules for each port used by the broadcast messages.
If <code>&lt;access&gt;</code> configured for <code>drop</code> no traffic will be forwarded to the wireless interface. In this case adding rules will allow specific traffic to be forwarded to the wireless interface.
The default is <code>accept</code>.</td></tr>
</table>

<table>
<tr><th>Command</th><th>Description</th></tr>
<tr><td>eth-route
<forwarding rule&gt;</td><td>
<p>Specifies a rule against which traffic will be compared and the specified action taken. The rule can apply to the protocol, the IP address and port and will cause the packets to be dropped, forwarded or relayed to the wireless interface.</p>
<p>The format of the rule is:</p>
<p>[protocol] [ip XXX.XXX.XXX.XXX] [port XXXX] [action]</p>
<p>The details of the protocol options include:</p>
<table>
<tr><td>tcp</td><td>Apply rule to traffic identified as TCP.</td></tr>
<tr><td>udp</td><td>Apply rule to traffic identified as UDP.</td></tr>
<tr><td>icmp</td><td>Apply rule to traffic identified as ICMP.</td></tr>
<tr><td>bcast</td><td>Apply rule to broadcast traffic.</td></tr>
<tr><td>all</td><td>Apply rule to all traffic.</td></tr>
</table>
<p>The details of the action options include:</p>
<table>
<tr><td>accept</td><td>If the packet meets the conditions of the rule relay it to the wireless interface.</td></tr>
<tr><td>drop</td><td>If the packet meets the conditions of the rule do not relay it to the wireless interface and drop it.</td></tr>
<tr><td>relay</td><td>If the protocol option is bcast assigning the action to relay will cause UDP traffic with destination address 255.255.255.255 received on the specified port to be relayed to the wireless interface.</td></tr>
</table>
<p>It is not necessary to include both an IP address and Port number if one is omitted the rule will apply to all variants of the missing parameter.</p>
<p>The ip and port prefixes, shown in the rule format, must be included with the address and port number for the rule to be accepted. The port number cannot be specified if the protocol is set for icmp or all.</p>
<p>If the eth-route command is entered without a forwarding rule, the current installed rules will be displayed in the order by which they are applied.</p>
</td></tr>
<tr><td>del-eth-route
<forwarding rule&gt;</td><td>
<p>Deletes the defined eth-route rule defined by the <forwarding rule> parameter. There must be a matching forwarding rule in the rule list for any action to be taken. The full forwarding rule description must be used; the command does not recognize partial rule description.</p>
<p>The format of the rule is:</p>
<p>[protocol] [ip XXX.XXX.XXX.XXX] [port XXXX]</p>
</td></tr>
</table>

It can be seen in Table 5 the eth-route forwarding rules can have a number of formats and are able to support a wide range of options, the following examples provide descriptions of some of the different uses of the rule:

```
eth-route tcp ip 192.168.1.100 port 80 accept
```

Allows TCP/IP traffic for IP address 192.168.1.100 on port 80 to be forwarded to the wireless network.

```
eth-route all ip 192.168.1.100 drop
```

Blocks all traffic for IP address 192.168.1.100.

```
eth-route udp port 55899 accept
```

Allows all UDP traffic on port 55899 to be forwarded to the wireless network.

```
eth-route bcast ip 255.255.255.255 port 55899 relay
```

Allows UDP broadcast traffic on port 55899 to be forwarded to the wireless network.

```
eth-route icmp ip 192.168.1.100 accept
```

Allows all ICMP traffic for IP address 192.168.1.100 to be relayed to the wireless network.

When using the Ethernet firewall it is recommended that the `eth-route-default` be set to `drop` and rules entered to address the exceptions. For instance where an Ethernet client on the modules wired interface needs to access a data server at 192.168.1.100 on port 2929 and a FTP server at 192.168.1.200, while allowing the Ethernet client to ping the data server, the firewall configuration should look like the following:

```
eth-route-default drop
eth-route tcp ip 192.168.1.100 port 2929 allow
eth-route tcp ip 192.168.1.200 port 21 allow
eth-route icmp ip 192.168.1.100 allow
```

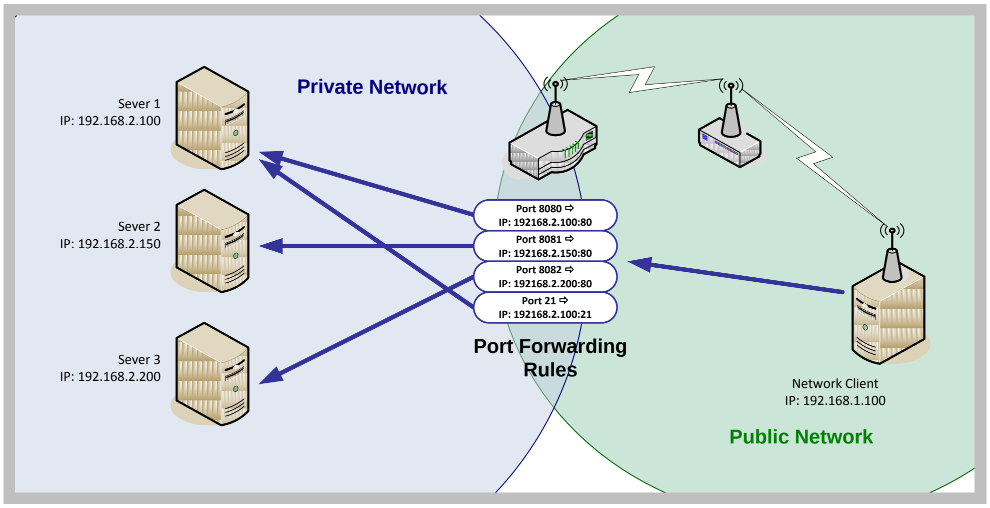
9.4 Router Port Forwarding Configuration

The modules Ethernet interface supports multiple Ethernet clients at one time. The built-in DHCP server will provide IP addresses for multiple devices when the appropriate DHCP requests are seen. When those client wish to access resources on the wireless interface (public network) they can initiate the connection (TCP, UDP, ICMP) and the router will handle all packet forwarding to and from the Ethernet interface. When a resource on the public network wants to access one of the clients on the Ethernet interface this can only be done, in case where there is more than one client, if power forwarding is enabled and an appropriate rule is configured.

To access a specific device on the Ethernet interface, from the public network, it is necessary to create a rule which maps a port on the public interface to an individual IP and port configuration on the Ethernet interface. Since this is a static mapping (is part of a predefined rule) it is recommended that static IP addresses be used on the Ethernet interface when port forwarding is being used.

When configured the public network IP interface will have a number of ports defined and mapped to a group of IP/Port combinations. A single IP address can have multiple rules, there is no restriction on the number of public ports linked to any specific IP/Port combination on the Ethernet interface, Figure 7 demonstrates the use of this.

Figure 7 - Port Forwarding Example



Configuring the firewall requires a use of the commands listed in Table 5.

Table 6 - Port Forwarding Configuration

<table>
<tr><th>Command</th><th>Description</th></tr>
<tr><td>wl-route-default <access&gt;</td><td>This sets the default port forwarding setting.
<table>
<tr><td>forward</td><td>All incoming packets on the wireless interface are forwarded to the address defined by eth-ip.</td></tr>
<tr><td>drop</td><td>All incoming packets on the wireless interface are dropped.</td></tr>
</table>
If <access> configured for forward all incoming requests, except broadcast messages, will be forwarded to the IP address defined by the eth-ip setting. Relaying of broadcast messages must be explicitly enabled with the firewall rules for each port used by the broadcast messages.
If <access> configured for drop no traffic will be forwarded to the Ethernet interface, essentially creating a firewall to the Ethernet interface and clients on the interface. In this case adding rules will allow specific traffic to be forwarded to the Ethernet interface.
The default is forward.</td></tr>
</table>

<table>
<tr><th>Command</th><th>Description</th></tr>
<tr><td>wl-route <forwarding rule&gt;</td><td>Specifies a rule against which traffic will be compared and the specified action taken. The rule can apply to the protocol and the target port and will cause the packets to be dropped, forwarded or relayed to the Ethernet interface.
The format of the rule is:
[protocol] [port XXXX] [action] [IP Address:Port#]
The details of the protocol options include:
<table>
<tr><td>tcp</td><td>Apply rule to traffic identified as TCP.</td></tr>
<tr><td>udp</td><td>Apply rule to traffic identified as UDP.</td></tr>
<tr><td>icmp</td><td>Apply rule to traffic identified as ICMP.</td></tr>
<tr><td>bcast</td><td>Apply rule to broadcast traffic.</td></tr>
<tr><td>all</td><td>Apply rule to all traffic.</td></tr>
</table>
The port number cannot be set if the protocol selection is all or icmp.
The details of the action options include:
<table>
<tr><td>forward</td><td>If the packet meets the conditions of the rule relay it to the specified IP address and port number on the Ethernet interface.</td></tr>
<tr><td>drop</td><td>If the packet meets the conditions of the rule drop it and do not relay it to the Ethernet interface.</td></tr>
<tr><td>relay</td><td>If the protocol option is bcast assigning the action to relay will cause UDP traffic with destination address 255.255.255.255 received on the specified port to be relayed to the Ethernet interface.
If selected the IP address [IP Address:Port#] should not be included in the rule.</td></tr>
</table>
It is not necessary to include a Port number as part of the target IP address for the forwarding rule, if one is omitted the rule will apply the incoming port number to the redirected packet.
The port prefix, shown in the rule format, must be included with the port number for the rule to be accepted.
If the wl-route command is entered without a <forwarding rule>, the current installed rules will be displayed in the order by which they are applied.</td></tr>
<tr><td>del-wl-route <forwarding rule&gt;</td><td>Deletes the defined wl-route rule defined by the <forwarding rule> parameter. There must be a matching forwarding rule in the rule list for any action to be taken. The full forwarding rule description must be used; the command does not recognize partial rule description.
The format of the rule is:
[protocol] [ip XXX.XXX.XXX.XXX] [port XXXX]</td></tr>
</table>

It can be seen in Table 6 the `wl-route` port forwarding rules can have a number of formats and are able to support a wide range of options; the following examples provide descriptions of some of the different uses of the rule:

```
wl-route tcp port 80 forward 192.168.2.101:80
```

Forwards incoming TCP/IP traffic on port 80 to IP address 192.168.2.101 on port 80.

```
wl-route all forward 192.168.2.105
```

Forwards all traffic to IP address 192.168.2.105.

```
wl-route udp port 55899 drop
```

Drops all UDP traffic on port 55899.

```
wl-route bcast port 55899 relay
```

Allows UDP broadcast traffic on port 55899 to be relayed to the Ethernet interface.

```
wl-route icmp drop
```

Drops all ICMP traffic.

When using port forwarding you have the choice of opening the interface and allowing everything to be relayed (`wl-route-default forward`) or to stop all traffic except that which is specific to the Ethernet clients (`wl-route-default drop`) in both cases including rules will allow the specific services to be handled appropriately by allowing to be relayed across the device correctly.

When `wl-route-default drop` is applied -it is necessary to have at least one rule for any traffic to be relayed.

As an example let’s look at the port forwarding configuration for the system shown in Figure 7. Within the configuration of the networks it is necessary to get access to the individual devices web interfaces for configuration and also to access the FTP server on 192.168.2.100, the port forwarding configuration should look like the following:

```
wl-route-default drop
wl-route tcp port 8080 forward 192.168.2.100:80
wl-route tcp port 8081 forward 192.168.2.150:80
wl-route tcp port 8082 forward 192.168.2.200:80
wl-route tcp port 21 forward 192.168.2.100
```

In this case addressing 192.168.1.217:8080 will access the web server on server 1, 192.168.1.217:8081 will access the web server on server 2, 192.168.1.217:8082 will access the web server on server 3 and any FTP access on port 21 will access the FTP server on server 1.

9.5 Ethernet Port mode: Router vs. Client

The Ethernet of the module supports two distinct functional modes, router and client. It is important to understand the differences between the two, when they should be used and the appropriate settings for each.

The router setting must be used when the device is to be an Ethernet Client adapter, where data bridging between the Ethernet and 802.11 interfaces will used. In this mode the module is configured as a NAT3 router, the Ethernet interface is capable of serving IP addresses from its DHCP server. The Ethernet interface of the module will act as the gateway to the 802.11 network for devices attached to the network on the Ethernet interface.

The client setting must be used when the module is to be used as a serial device server and no Ethernet to 802.11 bridging will be required. In this configuration the Ethernet or 802.11 interfaces will be network clients to which the serial ports will tunnel and establish data connections. In this mode only one of the network interfaces (Ethernet or 802.11) is allowed to support DHCP, the other must use a static IP address.

The following tables (Table 7, Table 8) address the specific requirements for each mode and identify the relayed parameters for correct configuration.

Table 7 - Configuring the Ethernet Module as a Router

<table>
<tr><th>Command</th><th>Description</th></tr>
<tr><td>eth-role router</td><td>This configures the Ethernet interface as the gateway for the Ethernet connected network and as a NAT3 router.</td></tr>
<tr><td>eth-ip</td><td>This is the base IP address of the private network DHCP server address pool, and is the first IP address the DHCP server will lease to a client on the private network when the client is using DHCP. It is also the default private network IP address used for forwarding traffic from the public network.
This address must match the private network client IP address when a single client is attached and is using a static IP address. If this does not match the address, traffic from the public network will NOT be routed correctly.
When using static IP addresses it is necessary for the Ethernet host to be capable of responding to the ICMP ARP protocol or for the host to issue a Gratuitous ARP. This is required to make sure wireless traffic is routed correctly.
Traffic originating from Ethernet clients will be routed correctly.</td></tr>
<tr><td>eth-subnet</td><td>This is the subnet mask the DHCP server will provide to the client when the client is using DHCP.</td></tr>
<tr><td>eth-gateway</td><td>This is the IP address of the Ethernet Interface on the Airborne Ethernet Bridge and is the target address for communications between the Ethernet client and the Airborne Bridge.</td></tr>
<tr><td>eth-dhcp-server [state]</td><td>Enables or disables the DHCP server on the private network. If the Ethernet host is using DHCP to acquire an IP address this must be enabled.
The [state] can be one of the following:
<table>
<tr><td>enabled</td><td>Enables the DHCP server. The address configured by eth-ip is the first address issued; subsequent requests will issue address incrementally.</td></tr>
<tr><td>disabled</td><td>Disables the DHCP server. Requires the Ethernet hosts to be configured with static IP addresses, subnet masks and gateway addresses.</td></tr>
</table></td></tr>
</table>

Command	Description
`wl-mac-clone`	Enables or disables MAC address cloning for the module. When this mode is enabled the modules wireless interface will use the MAC address of the first Ethernet host as its own.

Table 8 - Configuring the Ethernet Module as an Ethernet Client

Command	Description
`eth-role client`	This configures the Ethernet interface as the gateway for the Ethernet connected network and as a NAT3 router.
`eth-dhcp-acqlimit`	Determines the number of seconds the module should wait to acquire its IP configuration using DHCP before applying the DHCP fallback algorithm (if enabled). The value should always exceed the DHCP acquire time for the target network. It is recommended that the typical acquire time should be exceeded by a minimum of 15 seconds. A value of zero (0) will disable IP fallback.
`eth-dhcp-client`	Configures the DHCP Client Host Name. This can be used to uniquely identify the client in the DHCP server IP address tables. The default configuration is `AirborneXXXXXX`, where `XXXXXX` are the last six (6) hexadecimal digits of the modules MAC address.
`eth-dhcp-fb`	Enables or disables the fall back algorithm for the Ethernet port. When enabled the `eth-dhcp-fbip`, `eth-dhcp-subnet` and `eth-dhcp-gateway` will be applied after the `eth-dhcp-acqlimit` has been exceeded. When disabled `0.0.0.0` is applied as the IP address of the Ethernet interface.
`eth-dhcp-fbauto`	Enabling the fallback auto mode will cause the module to use the last successful DHCP IP configuration to set `eth-dhcp-fbip`, `eth-dhcp-fbsubnet`, `eth-dhcp-gateway`, `dns-server1` and `dns-server2`. This command requires that `eth-dhcp-fb` is enabled and the `eth-dhcp-acqlimit` is none zero. The changes are not persistent across power cycles or restarts. To make the setting changes persistent please see `eth-dhcp-fbper`.
`eth-dhcp-fbip`	Configures the IP address used by the DHCP fallback algorithm when DHCP fails.
`eth-dhcp-fbsubnet`	Configures the IP subnet used by the DHCP fallback algorithm when DHCP fails.
`eth-dhcp-fbgateway`	Configures the Gateway IP address used by the DHCP fallback algorithm when DHCP fails.
`eth-dhcp-fbper`	Enabling the fallback auto mode will cause the last successful DHCP IP configuration to be persistent across power cycles and restarts. When enabled the last successful configuration will be stored to `eth-dhcp-fbip`, `eth-dhcp-fbsubnet`, `eth-dhcp-gateway`, `dns-server1` and `dns-server2`. This command requires that `eth-dhcp-fb` and eth-dhcp-fbauto are enabled and the `eth-dhcp-acqlimit` is none zero.

Command	Description
`eth-dhcp-vendorid`	Configures the DHCP Vendor Class ID string to use in the DHCP requests for the Ethernet interface.

10.0 WLAN Security

The Airborne Enterprise Wireless Device Server family supports all the latest WiFi security interoperability requirements for 802.11 products; this includes WEP, WPA and WPA2. The Airborne product family supports both Personal and Enterprise versions of WPA2, allowing delivery and storage of certificates and private keys to the module.

The configuration of the module for each of these security configurations is similar, utilizing common security commands with parameter variations to identify the method required. Each method does have supporting information and parameters to be defined, the following sections identify the typical requirements for these different security type.

It is assumed in all of the following descriptions that a valid Service Set Identifier (SSID) has been entered into the device server.

10.1 Disabled (No Security)

Under this mode there is no security applied. The only condition of association is compatibility of the radio with the infrastructure.

A wireless network using this protocol is not secure and is open to attack and intrusion. Devices and data on such a network should be considered at risk. This configuration is not recommended for anything other than initial set-up of the device.

If this security setting is to be used it is recommended all data traffic be performed over SSH (Section 8.1.6 and 8.1.7).

10.2 WEP Security

Wired Equivalent Privacy (WEP) was the original security protocol adopted by 802.11. WEP uses the stream cipher RC4 for confidentiality and CRC-32 checksum for message integrity. The standard was compromised in 2004 and has been deprecated as a security method. Although organizations still utilize WEP, it is not a recommended security protocol.

Standard 64-bit WEP uses a 40 bit key and a 24 bit initialization vector (IV), to form the RC4 traffic key, this is also known as WEP-40. The 128-bit version of WEP utilizes the same 24 bit IV but includes a 104 bit key (WEP-104).

The 64 bit and 128 bit keys are entered manually into the device server. These must match the keys in the target AP.

To configure the module for WEP the following commands must be completed, note that the full description of the commands and available parameters can be found in section 19.0:

Table 9 - WEP Configuration Parameters

Command	Description
`wl-security wep128`	Defines WEP with a 128 bit key.
`wl-auth auto`	Allows the client and AP to decide the most appropriate authentication type.
`wl-def-key 1`	Configures the default WEP key to be used.
`wl-key-1 12345678901234567890123456`	Defines the 128 bit key as 26 hex digits. This key must match the key on the AP.
`clear-wep`	Removes all WEP keys from the device. This command requires a `commit` for the keys to be removed permanently. Once removed the device will no longer be able to establish a connection to any WLAN that requires them.

In addition to the standard WEP configuration the module also supports a security protocol that utilizes LEAP with WEP encryption, the required configuration for this security settings is shown in Table 10.

Table 10 - WEP-LEAP Configuration Settings

Command	Description
`wl-security wep-leap`	Defines WPA with EAP-LEAP authentication. This requires the use of a RADIUS server on the target network; the server must support the LEAP authentication process.
`user-leap MyUserName`	Defines the username to be used for authentication with the RADIUS server. There must be a valid user account with the defined name.
`pw-leap MyUserPassword`	Defines the password for the user name defined by `user-leap`. This must match the password on the RADIUS authentication server.
`wl-def-key 1`	Configures the default WEP key to be used. The key must be Key 1.
`wl-key-1 12345678901234567890123456`	Defines the 128 bit key as 26 hex digits. This key must match the key 1 on the AP.

10.2.1 WPA Migration Mode

WPA migration mode is a Cisco specific mode, where both WPA and non-WPA client can associate to an Access Point using the same Service Set Identifier (SSID).

Quatech has developed and provides a number of options for support of the WPA migration mode, if it is being used by the target infrastructure. These optional parameters are fully described in section 19.0. They allow the use of WPA or WEP as the authentication process.

10.3 WPA Security

WiFi Protected Access (WPA) is a compatibility certification program created by the WiFi Alliance to indicate compliance to a minimum set of security and functional capabilities for 802.11 devices. The WPA certification program was created to mitigate the issues created by the devaluation of the WEP security standard.

WPA utilizes part of the 802.11i security standard but relies upon the same RC4 cipher as WEP. WPA introduced Temporal Key Interchange Protocol (TKIP) to 802.11 security and this significantly mitigated the flaws that existed in WEP. It not only hid the key more securely but provided packet sequencing and Message Integrity Checking (Michael MIC).

The module supports both WPA Personal and WPA-LEAP, the following table identify the settings required for configuration of these security methods.

Table 11 - WPA-Personal (PSK) Configuration

Command	Description
`wl-security wpa-psk`	Defines WPA with a Preshared Key (PSK).
`pw-wpa-psk password`	Defines the preshared key used by the module and must match the same PSK passphrase used by the AP. Must be 8-63 ASCII characters long and cannot include spaces.

Table 12 - WPA-LEAP Configuration

Command	Description
`wl-security wpa-leap`	Defines WPA with EAP-LEAP authentication. This requires the use of a RADIUS server on the target network, the server must support the LEAP authentication process.
`user-leap MyUserName`	Defines the username to be used for authentication with the RADIUS server. There must be a valid user account with the defined name.
`pw-leap MyUserPassword`	Defines the password for the user name defined by `user-leap`. This must match the password on the RADIUS authentication server.

10.4 WPA2 Security

WiFi Protected Access 2 (WPA2) is a compatibility certification program created by the WiFi Alliance to indicate compliance to a minimum set of security and functional capabilities for 802.11 devices. The WPA2 certification program was created to enhance the security provided by WPA and utilize more fully the IEEE 802.11i standard and the available advanced hardware.

WPA2 implements the mandatory elements of the IEEE 802.11i standard and replaces TKIP with AES-CCMP encryption and is considered fully secure at this time. WPA2 has two configurations: Personal and Enterprise. WPA2-Personal utilizes the same Pre-Shared Key (PSK) as supported by WPA, but uses AES-CCMP instead of TKIP.

The implementation of WPA2-Personal follows very closely the WPA example, in fact to the user the configuration is identical, and the underlying security improvements are hidden by the device. The device supports both ASCII string and pre-calculated hex keys as valid input, a description of the configuration requirements can be seen in Table 13 and Table 14.

Table 13 - WPA2-Personal (PSK) ASCII PSK Configuration

Command	Description
`wl-security wpa2-psk`	Defines WPA2 with a Preshared Key (PSK).
`pw-wpa-psk password`	Defines the preshared key used by the module and must match the same PSK passphrase used by the AP. Must be 8-63 ASCII characters long and cannot include spaces.

Table 14 - WPA2-Personal (PSK) Precalculated Key Configuration

Command	Description
`wl-security wpa2-psk`	Defines WPA2 with a Preshared Key (PSK).
`pre-calc-psk password`	Defines the precalculated hex key used by the AP. Must be 64 ASCII Hex digits long.

10.5 Enterprise Security

Enterprise supports a set of EAP (802.1x) protocols to provide the highest level of security available for 802.11 implementations. As defined by the WiFi Alliance, any product claiming WPA-Enterprise or WPA2-Enterprise capability should support the following group of EAP processes:

- EAP-TLS (Mandatory)
- PEAPv0/EAP-MSCHAPv2
- PEAPv1/EAP-GTC
- EAP-TTLS/MSCHAPv2
- EAP-SIM

Since all but the EAP-TLS are optional, many companies claim WPA2-Enterprise compliance with minimal support (EAP-TLS only). Since there is no requirement from the WiFi Alliance to make the implementation of the security standards user-friendly, it is not always the case that configuring an embeddable WiFi device for these advanced security methods is easy, let alone possible. The Quatech module supports all EAP processes except PEAPv1 and EAP-SIM.

The modules support WPA (TKIP) and WPA2 (AES-CCMP) encryption without requiring separate configuration of the EAP process type.

The implementation of WPA2-Enterprise is more complex and requires not only configuration of the device but, in most cases, delivery of certificates and private keys as well. These are small (2K-6K files) that the client uses to authenticate with an infrastructures' RADIUS server. For the different EAP processes to work it is required to define which process and underlying encryption methods to use, along with identification of the appropriate certificates and private keys. Each EAP process has a different requirement. Although they utilize the same common elements, each treats the authentication process differently and accordingly requires the credentials to be presented in a particular way.

The certificates are typically owned and generated by the Information Technology (IT) department of the organization that owns the infrastructure. The certificates have standard formats. It is critical to make sure that all certificates are in the appropriate format for the client to utilize.

Since there are different configuration requirements for each EAP process the following tables (Table 15, Table 16 and Table 17) identify the typical requirements for implementing each type when using a certificate type other than .P12 and .PFX.

Table 15 - EAP-TLS/MSCHAPv2 Configuration

Command	Description
`wl-security tls`	Sets the EAP authentication process to be used.
`eap-ident [client username from RADIUS server]`	Sets the username/EAP Identity for the client. There must be a valid username on the RADIUS server that matches this name. Replace the `[client username from RADIUS server]` with the user name (no parenthesis).
`priv-key-password [client private key password]`	Sets the password for the client private key file. This must be the password on the RADIUS server that matches the key used to build the private key file. Replace the `[client private key password]` with the password for the private key file (no parenthesis).
`ca-cert-filename [CA root cert name].pem`	Identifies the CA root certificate name to be used. Replace `[CA root cert name].pem` with the required filename (no parenthesis). The certificate must be saved to the module with the name identified by this command.
`client-cert-filename [client cert name].pem`	Identifies the client certificate name to be used. Replace `[client cert name].pem` with the required filename (no parenthesis). The certificate must be saved to the module with the name identified by this command.
`priv-key-filename [client private key name].pem`	Identifies he client private key file to be used. Replace `[client private key name].pem` with the required filename (no parenthesis). The private key file must be saved to the module with the name identified by this command.

Table 16 - PEAPv0/EAP-MSCHAPv2 Configuration

Command	Description
`wl-security peap`	Sets the EAP authentication process to be used.
`eap-ident [client username from RADIUS server]`	Sets the username/EAP Identity for the client. There must be a valid username on the RADIUS server that matches this name. Replace the `[client username from RADIUS server]` with the user name (no parenthesis).
`eap-password [Password for client username]`	Sets the password for the client. This must be the password on the RADIUS server that matches the username. Replace the `[Password for client username]` with the password for the account (no parenthesis).

Command	Description
`ca-cert-filename [CA root cert name].pem`	Identifies the CA root certificate name to be used. Replace `[CA root cert name].pem` with the required filename (no parenthesis). The certificate must be saved to the module with the name identified by this command.
`eap-phase1 peaplabel=0`	Identifies the outer authentication type to be used. In this case PEAPv0.
`eap-phase2 auth=MSCHAPV2`	Identifies the inner authentication type to be used. In this case MSCHAPv2

> The module does support PEAPv0 without certificates. Set up for this configuration requires the `ca-cert-filename` to be blank.
>
> This security configuration compromises the strength of the PEAPv0 authentication and is not recommended for implementation.

Table 17 - EAP-TTLS/MSCHAPV2 Configuration

Command	Description
`wl-security ttls`	Sets the EAP authentication process to be used.
`eap-ident [client username from RADIUS server]`	Sets the username/EAP Identity for the client. There must be a valid username on the RADIUS server that matches this name. Replace the `[client username from RADIUS server]` with the user name (no parenthesis).
`eap-password [Password for client username]`	Sets the password for the client. This must be the password on the RADIUS server that matches the username. Replace the `[Password for client username]` with the password for the account (no parenthesis).
`ca-cert-filename [CA root cert name].pem`	Identifies the CA root certificate name to be used. Replace `[CA root cert name].pem` with the required filename (no parenthesis). The certificate must be saved to the module with the name identified by this command.
`eap-anon-ident username@example.com`	The unencrypted anonymous identity string used by EAP-TTLS.
`eap-phase2 auth=MSCHAPV2`	Identifies the inner authentication type to be used. In this case MSCHAPv2

If you are using the Personal Information Exchange format for your certificates please follow the configurations in Table 18.

The .PFX and .P12 private key formats commonly store multiple objects, including the private keys and user certificates required for authentication to a network. Using this format removes the need to identify all the individual certificates for authentication using TLS.

Table 18 – EAP-TLS/MSCHAPv2 Configuration Using .PFX or .P12 Private Key

Command	Description
`wl-security tls`	Sets the EAP authentication process to be used.
`eap-ident [client username from RADIUS server]`	Sets the username/EAP Identity for the client. There must be a valid username on the RADIUS server that matches this name. Replace the `[client username from RADIUS server]` with the user name (no parenthesis).
`ca-cert-filename [CA root cert name].pem`	Identifies the CA root certificate name to be used. Replace `[CA root cert name].pem` with the required filename (no parenthesis). The certificate must be saved to the module with the name identified by this command.
`priv-key-password [client private key password]`	Sets the password for the client private key file or Personal Information Exchange certificate. This must be the password on the RADIUS server that matches the key used to build the private key file. Replace the `[client private key password]` with the password for the private key file (no parenthesis).
`priv-key-filename [client private key name].[pem/pfx/p12]`	Identifies he client private key file or Personal Information Exchange certificate to be used. Replace `[client private key name].[pem/pfx/p12]` with the required filename (no parenthesis). The private key file must be saved to the module with the name identified by this command.

When using .PFX/.P12 certificates with the module it is possible to authenticate to the network without defining the CA Certificate. This is a none preferred configuration and is not recommended.

It is important to know that there are many variations and additional configurations that the module supports. Please contact Quatech Technical Support if your configuration is not covered by the documentation. There are additional parameters available, these are listed in section 19.0.

10.6 Configuring EAP-FAST

EAP-FAST (Flexible Authentication via Secure Tunneling) is a protocol proposal by Cisco Systems as a replacement for LEAP. The protocol was designed to address the weaknesses of LEAP while preserving a lightweight implementation. Use of server certificates is optional in EAP-FAST. EAP-FAST uses a Protected Access Credential (PAC) to establish a TLS tunnel in which client credentials are verified.

The EAP-FAST protocol has three phases:

- Phase 0 is an optional phase in which the PAC can be provisioned manually or dynamically, but is outside the scope of EAP-FAST as defined in RFC4851. PAC provisioning is still officially Work-in-progress, even though there are many implementations. PAC provisioning typically only needs to be done once for a RADIUS server, client pair.
- Phase 1, the client and the AAA server uses the PAC to establish a TLS tunnel.
- Phase 2, the client credentials are exchanged inside the encrypted tunnel.

It is worth noting that the PAC file is issued on a per-user basis. If a new user logs on the network from a device, he needs a new PAC file provisioned first. This is one reason why it is difficult not to run EAP-FAST in the unsecure anonymous provisioning mode. The alternative is to use device passwords instead, but then it is not the user that is validated on the network.

Due to the use of PAC files for provisioning and credential validation the configuration and use of EAP-FAST on the module is slightly different than the earlier enterprise security modes. The module supports the use of EAP fast with either WPA (TKIP) or WPA2 (AES-CCMP), Table 19 highlights the commands required and their use when implementing EAP-FAST on the module.

Table 19 - EAP-FAST Configuration

Command	Description
`wl-security wpa-fast`	Sets the EAP-FAST authentication process using TKIP encryption.
`wl-security wpa2-fast`	Sets the EAP-FAST authentication process using AES-CCMP encryption.

Command	Description

`eap-fast-provisioning <option>`	Determines the method by which the EAP-FAST credentials (PAC) are provisioned between the module and the AAA server.
	The <option> defines the method of interaction and the level of security to be used in the automatic provisioning of the modules credentials by the AAA server. The options are:
	`authenticated`
	The AA server's identity is validated by the module before the credentials are provisioned.
	`unauthenticated`
	The AA server's identity is not validated by the module before the credentials are provisioned.
	`either`
	The module will attempt to use the `authenticated` method first, if this is not possible then the module will use the `unauthenticated`.
	If using `authenticated` or `either` the `ca-cert-filename` must be set for the AAA server to be authenticated during the provisioning process. If no `ca-cert-filename` is set the provisioning process will not fail.
	To use the `ca-cert-filename` the certificate must be stored on the module.

`eap-fast-max-pac-list <#ofServers>`	Configures the number of AAA server credentials that can be held by the module. Changing the default value can impact memory resources, although the memory will only be used if the credentials are installed.
`ca-cert-filename [CA root cert name].pem`	Identifies the CA root certificate name to be used for authentication. Replace `[CA root cert name].pem` with the required filename (no parenthesis). The certificate must be saved to the module with the name identified by this command. If no CE root certificate is being used the file name must be blank.

10.7 Managing Certificates and Private Keys

Since certificates are used by most of the supported EAP protocols it is necessary to upload these files to the module before attempting to configure the device for WPA2-Enterprise security.

The module supports both pushing and pulling of certificates and private key files to the device, utilizing FTP and Xmodem transfer protocols. The different methods can be seen in Figure 8.

The CLI commands that manage the delivery process are described in Table 20.

Table 20 - Certificate Delivery Commands

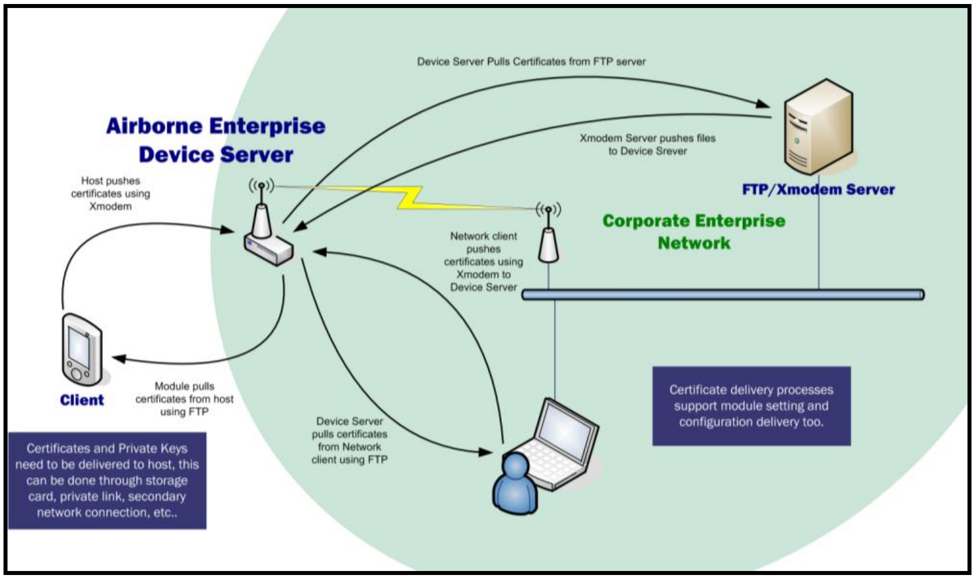
Command	Description
`put-cert [file name]`	Will cause the device server that you are going to push the certificate to, to wait for the attached host to initiate the Xmodem transfer to the module. This method supports Xmodem transfer over a serial interface or in a telnet session. The filename included as the argument will be the name the file is saved with on the device server. This name is the one to be referenced when a certificate is called. No file path should be included. An extension must be included. Once the command is issued the device server waits for the attached host to initiate an Xmodem transfer. Once the transfer of the file is complete the command returns an `OK`. Once the download is complete it is necessary for the `save` command to be issued, this will cause the certificate to be stored to the device server.

`get-cert [file name]`	Will cause the device server to retrieve a certificate from the FTP server identified by the parameters associated with the following commands:
	`ftp-server-path`
	`ftp-server-address`
	`ftp-user`
	`ftp-password`
	`ftp-filename`
	Once the download is complete it is necessary for the `save` command to be issued, this will cause the certificate to be stored to the device server.
	No file path should be included.
	It is required that the device server is associated and authenticated with a network and has a valid IP address before issuing this command.

`ftp-server-address`	This defines the IP address of the target FTP server. The address must be in the standard format XXX.XXX.XXX.XXX. Where XXX can have a value between 0 and 255. The resultant IP address must not be 0.0.0.0.
`ftp-server-path`	This defines the directory path for the subdirectory that contains the target certificate to be downloaded. This does not need to be set if the file is in the default directory for the specified ftp-user.
`ftp-user`	Defines the username for the FTP account, associated to the FTP server defined by `ftp-server-address`.
`ftp-password`	Defines the password for the FTP account, associated to the FTP server defined by `ftp-server-address`.
`ftp-filename`	Defines the name of the certificate or private key file to be uploaded or downloaded. The file extension must be included. The filename does not support wildcards.

The use of these commands depends upon the transfer protocol being used.

Figure 8 - Certificate and Private Key Delivery Methods


Device Server Pulls Certificates from FTP server
Airborne Enterprise Device Server
Xmodem Server pushes files to Device Srever
FTP/Xmodem Server
Host pushes certificates using Xmodem
Corporate Enterprise Network
Network client pushes certificates using Xmodem to Device Server
Client
Module pulls certificates from host using FTP
Device Server pulls certificates from Network client using FTP
Certificate delivery processes support module setting and configuration delivery too.
Certificates and Private Keys need to be delivered to host, this can be done through storage card, private link, secondary network connection, etc..

Control of the certificate and private key files is handled by a separate group of commands these are described in Table 21.

Table 21 - Certificate Management Commands

Command	Description
`list-cert`	This provides a list of certificates resident on the module, including files that have been transferred but not yet saved to the module.
`del-cert [cert name]`	The command deletes certificates that are stored on the module; the command requires a filename argument to be supplied. The filename argument does support wild cards e.g. `del-cert *.*` : Will delete all certificates. `del-cert user*.*` : Will delete all certificates beginning with user It is required to issue the `save` command after this command to make the changes permanent.

Command	Description
clear-cred	This command allows the credentials stored in the module to be cleared prior to any new ones being applied. The use of this command is recommended to guarantee that no artifacts of a previous security configuration impact the success of any new applied configuration. The command clears the following: pre-calc-psk ca-cert-filename ca-cert2-filename client-cert-filename client-cert2-filename priv-key-filename priv-key2-filename dh-parm-filename dh-parm2-filename priv-key-password priv-key2-password eapfast-pac-filename eap-password eap-ident eap-anon-ident eap-phase1 eap-phase2 subject-match subject-match2 alt-subject-match alt-subject-match2 user-wpa-supp-filename cfg-encrypt Resets command to default: pw-wpa-psk passphrase Clears the following files:: EAP-FAST PAC

Command	Description

`clear [parameter]`	This command allows a single parameter to be cleared.
	The following commands can be cleared:
	`ca-cert-filename`
	`ca-cert2-filename`
	`client-cert-filename`
	`client-cert2-filename`
	`priv-key-filename`
	`priv-key2-filename`
	`dh-parm-filename`
	`dh-parm2-filename`
	`priv-key-password`
	`priv-key2-password`
	`eapfast-pac-filename`
	`eap-password`
	`eap-ident`
	`eap-anon-ident`
	`eap-phase1`
	`eap-phase2`
	`subject-match`
	`subject-match2`
	`alt-subject-match`
	`alt-subject-match2`
	`user-wpa-supp-filename`
	`ftp-server-address`
	`ftp-server-path`
	`ftp-user`
	`ftp-password`
	`ftp-filename`
	`ssh-key`

`save`	This command moves any uploaded certificates or private keys to permanent storage, making them persistent across restarts or power cycles. Issuing `save` after `del-cert` makes any certificate deletions permanent.

The module is capable of storing multiple certificates. The number of certificates is limited only by available resources; typically up to twenty (20) certificates can be held by the module at any one time.

This allows multiple individual WPA2-Enterprise configurations to be applied to the device server without needing additional certificates or private keys to be delivered to the module.

11.0 Using Configuration Files

The module allows configuration files, describing a predefined device configuration, to be delivered and stored on the module. There are several advantages to using the configuration files instead of command line or web interface input when configuring the module, the process is not only quicker but is less error prone and can better support configuration control, en mass and in-field updates.

There are two types of configuration file that can be delivered to the module, these are:

User *This configuration file contains configuration information from a particular installation. These parameters are ones which may change from location to location within multiple or single deployments of devices. The file which contains these parameters is called* `user_config.txt`.

OEM *This configuration file contains parameters that would be specific to the required factory defaults of the module integrator. These would represent the out-of-the-box configuration for the OEM product or a pre-defined configuration known by installers or technicians. The file which contains these parameters is called* `oem_config.txt`.

The two types of configuration file provide an option for the user to establish a set of their own factory defaults should a module need to be redeployed or recovered, or an installer incorrectly configures the device. When the device is to be recovered or redeployed the user may use the factory RESET command or hardware input to return the configuration to its original *factory* state. When the factory RESET is performed the `user_config.txt` file is deleted but the `oem_config.txt` is retained.

The **user** type configuration file supports encryption of sensitive parameters, like passwords, passphrases and keys. To use this option it is necessary to turn on the encryption, section 12.0 describes how to use this feature.

The module supports delivery of the configuration files using either Xmodem or the built-in FTP client; Table 22 outlines the processes for creating, delivering and managing the configuration file options.

Table 22 - Using Configuration Files

Command	Description
`Obtain or create configuration file`	A file is required before the transfer to the module is performed. This file must be a plain text file containing the parameters which are to be configured, section XX outlines the file and command format. The file may be created in any text editor and does not need to be called `user_config.txt` or `oem_config.txt`, it is recommended that the file be named in a way that indicates the installation and revision of the configuration. This name will be used by the ftp-filename command to identify the file to be uploaded when suing FTP for the transfer. Alternately a configuration file can be pulled from an existing module. Using the web interface it is possible to list the configuration files available on the module and copy the contents to a local host. This way supports the use of a pre-tested golden unit for configuration in large deployments or in a configuration control system. This method is recommended by Quatech.

`get-cfg [config_filename]`	This command uses the configuration settings for the FTP client and will upload the file identified by `ftp-filename` to the module with the name `[config_filename]`.
	It is necessary that a valid configuration exist for the FTP client before this command is used. See section 14.0 for details.
	`[config_filename]`can be set as:
	`user_config.txt`
	`oem_config.txt`
	`user_enc_config.uue`
	It is necessary to issue a `save` after this command for the configuration file to be persistent across power cycles or restarts.

`put-cfg [config_filename]`	This will use Xmodem to upload the user configuration file to the module with the name `[config_filename]`.
	Once the command has been issued the host connected through the CLI session will need to start the Xmodem transfer using Xmodem or Xmodem-1K.to the module.
	`[config_filename]`can be set as:
	`user_config.txt`
	`oem_config.txt`
	`user_enc_config.uue`
	It is necessary to issue a `save` after this command for the configuration file to be persistent across power cycles or restarts.

`list-cfg`	This will list the installed configuration files on the module. The command will list files that have been uploaded but have not yet been saved to the module as well as those saved to the module.
`del-cfg [config_filename]`	The command deletes configuration files that are stored on the module; the command requires a filename argument `[config_filename]` to be supplied. The `save` command must be issued after this command to make the changes persistent across power cycles and restarts.

Command	Description
save	This command moves any uploaded configuration files to permanent storage, making them persistent across restarts or power cycles. Issuing save after del-cfg makes any certificate deletions permanent.

11.1 Configuration File Format

The unencrypted configuration files are plain text files. The files contain the configuration information for the module. The format of the file contents follows the standard CLI command+parameter format; each line containing a separate command and parameter.

The following is an example of a `user_config.txt` file:

```
#!/bin/qtsh
# /var/etc/config/user_config.txt
#
wl-ssid RADIUS_TEST
wl-security wpa2-psk
esc-mode-serial-p2 off
bit-rate-p2 921600
parity-p2 e
flow-p2 h
eth-dhcp-server enable
eth-role router
wl-route-default forward
eth-route-default accept
```

The first three lines are part of the system generated file and are not necessary for manually generated configuration files.

12.0 Protecting Configuration Settings

Included in the module is the ability to protect sensitive configuration settings from prying eyes. This is achieved through enabling the encryption of those parts of the configuration that are considered sensitive. When enabled the sensitive settings like passwords, passphrase and keys are removed from the displayed configurations and stored in a separate encrypted file.

The default configuration for the module is to include all settings when the `user_config.txt` file is viewed. In this case passwords, passphrases and WEP keys are stored in plain text, in the configuration file. Although access to this file still requires authentication to the module, once authenticated anyone can view the settings.

The encryption setting for the device removes the sensitive parameters for the `user_config.txt` and places them in an encrypted file that cannot be directly viewed even when fully authenticated to the module. The following table describes the settings used to enable and disable the encryption of the sensitive settings; it also describes the impacted parameters.

Table 23 - Encryption of Configuration Files

Command	Description
`cfg-encrypt [enable|disable|locked|protected|permanent]`	The command controls the securing of parameters in the `user_config.txt` file by removing them from the `user_config.txt` and creating an encrypted file `user_enc_config.uue` that contain the parameters. When `enable` is selected the module will split the contents of the unencrypted user_config.txt (if it exists) into two files by removing the sensitive parameters that are present in the files into encrypted versions of the file. These encrypted files will be visible when the configuration files are listed by the `list-cfg` command but cannot be viewed in a plain text editor. A full description of the parameters is shown in section 19.0. The new file created is named `user_enc_config.uue`. If `disable` is selected subsequent to `enable` being selected the contents of the encrypted file are merged with the `user_config.txt` file and the parameters in the encrypted file become visible in plain text. This is useful for testing out the process and confirming the parameter encryption is working. When deploying in the field it is recommended that `locked`, `protected` or `permanent` be used.
`list-cfg`	This command lists the configuration files available on the module. If `cfg-encrypt` is enabled the encrypted file (`user_enc_config.uue`) will be listed in the response.

Command	Description
clear cfg-encrypt	Clears the state of the cfg-encrypt setting when one of the encrypted option has been enabled. The resultant state of the module depends upon the option applied. If the state is locked, issuing the command will change the state of cfg-encrypt to enable. This is a Level 5 (manufacturer) command. If the state is protected, issuing the command will change the state of cfg-encrypt to disable and will delete the user_enc_config.uue file. This will remove all protected settings. This is a Level 5 (manufacturer) command. Caution should be taken when using this option as it may impact the user's ability to connect to the module.
reset	Returns the module to OEM defaults. If the state is permanent, issuing the command will return the module to OEM defaults and delete the user_enc_config.uue file. This is a Level 5 (manufacturer) command.
auth-level	This command allows the required authentication level required for a given command to be changed. When using cfg-encrypt permanent it is recommend that the reset commands authentication level be raised to the same level as the cfg-encrypt command (level 5 - manufacturer). Use the command as follows: auth-level reset 5

12.1 Transferring Encrypted Configurations

It is possible to transfer encrypted configurations in the same way unencrypted configurations can be moved. When transferring the encrypted configuration it is necessary to deliver both the `user_config.txt` and the `user_enc_config.uue` files to the module. The target module must have `cfg-encrypt enable` set, this must be part of the delivered `user-config.txt` file.

The transfer an encrypted configuration the steps in Table 24 must be taken.

Table 24 - Encrypted Configuration Delivery

Command	Description
Copy source configuration files from example module	The user_config.txt and user_enc_config.uue files must be copied from a configured module and saved on the configuration station. The user_config.txt must contain the line: cfg-encrypt enable
get-cfg user_config.txt	This will use the FTP settings (See section 14.0) to upload the user configuration file.
get-cfg user_enc_config.txt	This will use the FTP settings (See section 14.0) to upload the encrypted user configuration file.

Command	Description
`put-cfg user_config.txt`	This will use Xmodem to upload the user configuration file. Once the command has been issued the host connected through the CLI session will need to start the Xmodem transfer using Xmodem or Xmodem-1K.
`put-cfg user_enc_config.txt`	This will use Xmodem to upload the encrypted user configuration file. Once the command has been issued the host connected through the CLI session will need to start the Xmodem transfer using Xmodem or Xmodem-1K.
`save`	This command moves any uploaded configuration files to permanent storage, making them persistent across restarts or power cycles.

Only FTP or Xmodem need to be used for the transfer of the configuration files to the module.

IMPORTANT: Both the `user_config.txt` and `user_enc_config.uue` files must be delivered to the module when using the encrypted option. Failure to deliver both files may cause incorrect operation of the module and cause it to become inaccessible.

If both files are not delivered and the module is inaccessible it is necessary to apply a factory default reset to the module.

13.0 WLAN Roaming

When configured for Infrastructure mode using the `wl-type` command, the Module supports roaming in accordance with the IEEE 802.11 specification. The following set of commands affect the Module’s roaming capabilities:

Table 25 - Commands that Affect Roaming

<table>
<tr><th>Command</th><th>Description</th></tr>
<tr><td>wl-type</td><td>This determines the network type being used by the device server, roaming applies to Infrastructure type only.</td></tr>
<tr><td>wl-ssid</td><td>This defines the Service Set Identifier or network name the device is to associate to.</td></tr>
<tr><td>wl-rate</td><td>This defines the maximum connection rate that the device will connect with in Mbps. It will limit the upper level connection rate but will not prevent auto-fall back rates should network coverage cause a lower rate to be selected.
Using a lower rate may provide a better connection and longer range.</td></tr>
<tr><td>wl-fixed-rate</td><td>This parameter locks the wl-rate and prevents auto fallback.
Use of this feature can cause the device server to not function in most 802.11 networks, unless a basic rate (1Mbps or 2Mbps) is selected by the wl-rate command.
Use of this command is not recommended.</td></tr>
<tr><td>wl-specific-scan</td><td>Determines how the device server scans for AP.
<table>
<tr><td>0</td><td>Use Broadcast Probes to attempt to find an Access Point.</td></tr>
<tr><td>1</td><td>Use Directed Probes to attempt to find an Access Point. In this mode only AP’s with matching SSID’s to the module will be probed.</td></tr>
</table>
When using Broadcast probes all AP advertising their SSID’s will respond to the scan, this will cause a result for wl-scan command that will provide a list of all responding AP’s within range of the device server.
Directed probes will limit responses to only those AP’s with matching SSID’s to the device servers. This will also restrict the wl-scan response to only those AP’s with identical SSID’d within range.</td></tr>
<tr><td>wl-assoc-backoff</td><td>The amount of time in milliseconds to back-off after the configured number of failed association attempts (defined by wl-assoc-retries). During the back-off period the device will not attempt to associate with the AP.
The back-off time has a range of 0-20,000 milliseconds (0 to 20 seconds).
This parameter will impact the aggressiveness of the association process for a device server in fringe coverage or noisy environments.</td></tr>
<tr><td>wl-assoc-retries</td><td>The number of time the device server will attempt to retry an association attempt, after a failure, before backing off.
The number of attempts can range from 0-32, the default is three (3).
This parameter will impact the aggressiveness of the association process for a device server in fringe coverage or noisy environments.</td></tr>
</table>

Command	Description
`wl-beacons-missed`	Configures the number of missed beacons, from an associated AP, that are missed before a roam is attempted. The number of beacons can range from 0-256, the default is six (6). It is not recommended to set this parameter to zero (0). This parameter will impact the roaming aggressiveness of the device server, the smaller the number the faster the device will attempt to roam.

If `wl-ssid` is set to the value `any`, the Device Server will perform a scan of APs and attempt to associate with the first AP that matches the security settings of the module, this is typically the AP with the strongest signal strength. The use of the `any` SSID allows the Device Server to associate with any AP that matches the modules security settings and is in range. Therefore, as the Device Server becomes mobile, it may associate with an AP that is not in your expected network. Due to the functionality of the `any` SSID you have little to no control over the roaming behavior of the device server. The factory default setting require the AP to be open (security disabled).

If `wl-ssid` is set to a value that is not the `any` string, the Device Server will scan for APs that match the SSID and 802.11 capability information header. If a matching AP is found, the Device Server will authenticate and attempt to associate. As the Device Server becomes mobile, it will only roam to APs that match the SSID and 802.11 capability information header.

The decision to roam is made entirely by the device server based upon the conditions of the environment, which includes signal strength, noise, etc. The device server will attempt to maintain as good a connection as possible and, based upon parameter settings in the device server, will decide to move from one AP to another AP when it cannot attain the quality of connection required.

14.0 FTP Configuration

The module includes an FTP client capable of uploading files to the device. The embedded FTP client is capable of authenticating with a network based FTP server and transferring a file to the device using the FTP protocol.

Table 26 - FTP Configuration Commands

Command	Description
`ftp-server-address`	This defines the IP address of the target FTP server. The address must be in the standard format XXX.XXX.XXX.XXX. Where XXX must have an integer value between 0 and 255. The resultant Ip address must not be 0.0.0.0.
`ftp-server-path`	This defines the directory path for the subdirectory that contains the target certificate to be downloaded, from the default directory of the `ftp-user`. This does not need to be set if the file is in the default directory for the specified `ftp-user`.
`ftp-user`	Defines the username for the FTP account, associated to the FTP server defined by `ftp-server-address`.
`ftp-password`	Defines the password for the FTP account, associated to the FTP server defined by `ftp-server-address`.
`ftp-filename`	Defines the name of the certificate or private key file to be uploaded or downloaded. The file extension must be included. The filename does not support wildcards.

To use this function it is necessary to configure the internal FTP Client with the necessary information for the file upload, the related commands can be seen in Table 26. Once the FTP configuration is applied all that is needed is the filename, as listed on the FTP server target directory, to be updated.

The FTP client supports upload of Certificates, Private Keys, Configuration files and Firmware. Separate commands determine the file type to be uploaded; Table 27 shows the different commands. All of these commands require the correct configuration of the FTP server parameters before being used; these parameters are described in Table 26.

Table 27 - FTP Upload Commands

Command	Description
`get-cert`	Uploads Certificates and Private keys from the designated FTP server. Requires the Certificate or Private Key file name as a parameter.
`get-cfg`	Uploads user or OEM configuration files from the designated FTP server. Requires the Certificate or Private Key file name as a parameter.
`update ftp`	Uploads Airborne Device Server firmware image from the designated FTP server.

15.0 Firmware Update

The Airborne Enterprise Device Server supports in-field updating of the devices firmware, to allow devices already deployed access to the latest feature updates and enhancements. The process of firmware update is supported through both the serial and the network ports. A single command is required to initiate and complete the update process.

Only firmware authorized by Quatech should be used. Any attempt to use an alternative image will void the modules warranty.

Delivery of the firmware image can be performed by either a FTP transfer (section 14.0) or through Xmodem transfer (section 15.2). When the FTP process is used the device server will locate the FTP server and pull the identified image file, once the download is complete the firmware update will start automatically.

CRITICAL: When updating firmware, power must be maintained during the entire update process. Removal or interruption of the power supply may cause a corruption of the firmware update and cause the module to stop functioning. If this occurs please contact Quatech Technical Support.

If Xmodem is used it is necessary for the module to be told that the updated image is going to be sent before the attached host initiates an Xmodem transfer of the file to the module. Once the download is completed the firmware update will start automatically.

The update process can take a significant amount of time depending upon the transfer process used to deliver the firmware files. The Firmware image files can be 3MB or larger, use of a slow serial interface (e.g. UART 9600 BAUD) will make file delivery a long process, however when FTP is used the file delivery can take only a few seconds. Regardless of the delivery process the actual firmware update process, once the file is delivered, takes approximately 90 seconds. During the update process it is critical that power is maintained to the device server.

Table 28 - `update` command description

Command	Description
`update`	This single command is used for both the FTP and Xmodem firmware updates. An `ftp` argument is required to initiate an FTP download of the firmware image. A valid FTP configuration must exist for the update to be successful. If Xmodem is used the module will wait for the host to initiate the file transfer after the update command is issued.

The modules configuration (user_config.txt, oem_config.txt and user_enc_config.uue), saved certificates and private key files are preserved across any firmware update.

15.1 Using FTP to Update Firmware

To use the embedded FTP capabilities of the module for firmware update, it is necessary to make sure the following settings are configured and the `update` command is used as defined in Table 29. It is also required that the module is associated to a wireless network or the Ethernet port is connected to a network containing the FTP server defined in the configuration.

It is important to note that the FTP based update provides the quickest update process due to the speed of the image download.

Table 29 - FTP Firmware Update

Command	Description
`ftp-server-address`	This defines the IP address of the FTP server on which the firmware image is being stored. The address must be in the standard format XXX.XXX.XXX.XXX. Where XXX must have an integer value between 0 and 255. The resultant IP address cannot be 0.0.0.0.
`ftp-server-path`	This defines the directory path for the subdirectory that contains the target firmware image to be downloaded, from the default directory of the `ftp-user`. This does not need to be set if the file is in the default directory for the specified `ftp-user`.
`ftp-user`	Defines the username for the FTP account, associated to the FTP server defined by `ftp-server-address`.
`ftp-password`	Defines the password for the FTP account, associated to the FTP server defined by `ftp-server-address`.
`ftp-filename`	Defines the name of the image file to be uploaded. The file extension must be included.
`update ftp`	This initiates the firmware update process. The update process is fully automatic once the command has been sent. The module will automatically download the image file, install the firmware update and restart the module. Note that any user configuration settings will not be lost during the process.

15.2 Using Xmodem to Update Firmware

When using Xmodem to do the firmware update there are no configuration changes required on the module. The process does require that a host device on either the serial or network ports can initiate an Xmodem file transfer, once the device server is ready to receive the firmware image file.

To complete the update process the command in Table 30, must be executed in a CLI session before any file transfer is initiated. Once executed the device server is ready to receive the firmware image, the network host must then initiate the file transfer using Xmodem. This can be done over the serial or network interfaces.

Table 30 - Xmodem Firmware Update

Command	Description
`update`	This initiates the firmware update process. The update process starts when the host system initiates the firmware image file transfer. The module will automatically download the image file, install the firmware update and restart the module. Note that any user configuration settings will not be lost during the process.

16.0 U-Boot Update

The update of the device servers U-Boot code is an infrequent event, however when required the following procedure must be followed. Delivery of the U-Boot image can be made using either the FTP or Xmodem update process. This procedure may be used for U-Boot versions v1.0.0 and higher, if your unit has a U-Boot earlier than this please contact Quatech Technical support.

To successfully achieve the U-Boot update the sequence identified in Table 31 must be followed.

The update cannot be done from the web interface, it is required that a CLI session on the network or serial interface be used to initiate the U-Boot update process.

The FTP update process requires that the unit is successfully associated to a wireless network.

Table 31- U-Boot Update Process

Step/Command	Description
`ver-uboot`	Issuing the `ver-uboot` command will allow identification of the current U-Boot version installed on the Airborne device. The last three numbers of the response indicate the version installed. `U-Boot 1.3.2 (Jul 16 2009 - 15:41:48) Quatech WLNx-9260 1.1.1` The above version of U-Boot is v1.1.1
`Obtain U-Boot .img file`	The U-boot file can be downloaded from the Quatech Support website or requested from Quatech Technical support.
`Configure FTP Server`	If using the FTP client to download the U-Boot firmware image, an FTP server is required to deliver the u-boot file to the module. This server must be on the same network and subnet as the module being updated. An account for the unit must be set-up, the username and password will be needed for configuration of the module. The U-Boot file should be placed in the home directory of the FTP account. If this is not possible the actual directory path from the home directory will need to be known for the configuration of the module.
`Configure Airborne device`	If using the FTP client to download the U-Boot firmware image the module will need the FTP settings configured. See section 14.0 for details of how to do this. It is not necessary to `commit` the FTP settings to the Airborne unit before using. However of not saved they will be lost on any restart or power cycle.
`update-uboot [option]`	Issue the `update-uboot` command with either the FTP or Xmodem update option. If `[option]` is `ftp` the device will download the U-Boot image from the FTP server and automatically start the update process. This requires that the FTP settings are already configured and correct. If `[option]` is `xmodem` the device will wait for an Xmodem transfer to be initiated by the connected host. The U-Boot image file will then be uploaded and the module will automatically start the update process.

Step/Command	Description
`restart`	The Airborne device should be restarted after the update process. This can be achieved by issuing the `restart` command or power cycling the unit.

Only firmware authorized by Quatech may be used. Any attempt to use an alternative image will void the modules warranty and potentially cause the module to stop functioning.

CRITICAL: When updating any firmware, power must be maintained during the entire update process. Removal or interruption of the power supply, during the update, may produce a corruption of the firmware update and cause the module to stop functioning. If this occurs please contact Quatech Technical Support.

17.0 Power Management

Control of the operating and standby power of the module can be critical in many applications; the Airborne Enterprise Device Server family offers various levels of control through the CLI interface, the following power save options are currently supported.

Table 32 – Power-Save Modes

Command	Description
`radio-on`	Enables the 802.11b/g radio. The radio will utilize the power profile defined by `pm-mode`. After this command is issued the radio will initiate and attempt to locate a valid wireless network to associate with. If one is found it will attempt to associate/authenticate.
`radio-off`	Disables the 802.11b/g radio. After the command is issued the device server will close all TCP/IP and UDP connections and power down the radio. When in this state the device server will no longer be associated with a wireless network and any network based communication will not be possible.
`pm-mode`	Sets the device server power management mode. Currently supports the modes described in Table 33.
`wl-sleep-timer`	Sets the inactivity timer for the UART and network interfaces before the module moves into sleep mode.
`radio-startup`	Determines the power state the radio after a device power up or restart. This command allows the radio to be placed into one of three states after the device server has completed its boot cycle. The three states include on (normal operation), sleep (puts the radio into the sleep mode defined in the pm-mode) and off (this is commonly called airplane mode).

The commands in Table 32 provide the most flexible power management options available for any device server. The most important command is `pm-mode`, as this provides automatic power management based upon the device operations and state, the following section covers the various options available for this command.

To correctly utilize the `pm-mode` command it is necessary to understand the available power modes and their impact upon the operation of the device and how this affects use of the device.

The `pm-mode` command allows control of the operation of the radio and its power mode. The available modes can be seen in Table 33, the following sections will detail the impact of each of these modes on the radios state and operation of the device server.

Table 33 - `pm-mode` Parameters

Mode	CPU	OSC/PLL	Radio	Wakeup
`active`	ON	ON	ON	None.
`doze`	STOP	ON	PSPoll	UART/Serial Traffic or directed/broadcast radio packet. Radio wakes on DTIM Period.
`sleep`	STOP	ON	Deep Sleep	UART/Serial Traffic. Device disassociated from network.
`wakeup`	N/A	N/A	N/A	This parameter causes the radio to transition from the sleep mode to either active or doze mode, depending upon the power mode the radio was in prior to entering sleep mode.

17.1 Mode: Active

This is the highest power mode; while `active` the radio is always on. This mode represents 802.11 operation under which the radio will fully interact with the medium and provides no power save functionality for the radio.

While in this mode the CPU utilizes its internal power management processes and attempt to minimize power usage, however the radio will function continually with this state enabled. While in the mode the radio will transmit and receive packets to and from the 802.11 media.

The radio will continue to be associated with any network it has successfully authenticated with.

17.2 Mode: Doze

While in this mode the device server’s radio utilizes the 802.11 power save standard PSPoll. When in the power save mode the radio remains is a low power state and wakes to the active state to receive management frames called beacons.

The period between waking to the active state is determined by the Access Point (AP) and is determined by the DTIM (Delivery Traffic Indication Message) value established by the AP. The greater the number the lower the power; however this impacts the latency of the data.

While in this mode the CPU utilizes its internal power management processes and attempt to minimize power usage, the radio will function in the power save PSPoll mode with this state enabled. While in the mode the radio will transmit and receive packets to and from the 802.11 media based upon the DTIM setting.

The radio will continue to be associated with any network it has successfully authenticated with.

17.3 Mode: Sleep

While in this mode the radio is in its lowest power state.

The radio will lose association with any network it was attached to prior to entering sleep mode. It will not re-associate while in the sleep mode.

17.4 Mode: Wakeup

This mode causes a radio in sleep mode to transition to active or doze mode. The mode the radio transitions to is the same as the mode it was in prior to entering sleep mode.

When the command is issued the radio will transition to the previous power state and will attempt to re-associate with its configured network, if it is available.

The wakeup parameter is not a persistent condition and is not committed to flash if it was the last `pm-mode` parameter issued when a `commit` command is issued.

17.5 Using Sleep Mode

Sleep mode provides the lowest power draw of any operational mode and as such provides significant advantage when used with battery or power sensitive applications. However the use and operation of the sleep mode changes depending upon the state and use of the UART interface, the following will outline the differences between these conditions.

Table 34 - UART Mode Affect on Sleep Mode

UART Mode	CLI	Actions
CLI	`pm-mode sleep`	Puts the radio into sleep mode.
	`pm-mode wakeup`	Transitions radio from sleep mode to either active or doze mode.
Listen	`wl-sleep-timer <integer>`	Defines the sleep activity timeout for the UART.
Pass	`wl-sleep-timer <integer>`	Defines the sleep activity timeout for the UART.

When the UART is in CLI mode the only way for the radio to enter sleep mode is to issue the `pm-mode sleep` command. Similarly to leave sleep mode the `pm-mode wakeup` command must be issued. In CLI mode it is assumed the host system is managing the Device Server and control of the power state would be completely under the hosts' control.

When the UART is in listen mode and the pm-mode has been set to sleep, either by issuing the `pm-mode sleep` command or by setting the `radio-startup sleep` parameter, the Device Server will wake from sleep mode based upon UART traffic. When in sleep mode a UART in listen mode, will not be able to accept incoming connection requests. When UART traffic is detected the radio will wake from sleep and listen for incoming connection requests, if no requests are received before the `wl-sleep-timer` expires, the radio will return to sleep mode. In this mode the host can manage availability of the device by simply sending a single character to the radio, lowering the management overhead and minimizing state changes of the Device Server.

When the UART is in pass mode and a data tunnel has been established the device server will enter sleep mode only if the wl-sleep-timer is set to a value greater the zero (0). When is pass mode the data tunnel will remain active until the inactivity timer wl-sleep-timer expires, when this happens the radio will enter sleep mode. When in sleep mode the device server is not accessible from the network interface and will not respond to any network initiated communications. When UART traffic is detected the radio will wake from sleep and re-establish the data tunnel, if no traffic is received or sent before the wl-sleep-timer expires, the radio will return to sleep mode. Any serial transmitted data sent before the data tunnel has been re-established will be buffered and transmitted when the connection is available. In this use of the sleep mode, the host is relieved of any power management monitoring or control of the device server, while optimizing power usage.

When the UART pass mode is used with power save it is important to note that the TCP/IP timeout is still running and will close the TCP/IP connection if it expires before the device server re-establishes the TCP/IP connection from sleep mode.

If the wl-sleep-timer is being used to manage the power state of the radio, consideration must be made for the finite time the radio takes to re-establish its connection with the network. This is true for both listen and pass mode operation. If the wl-sleep-timer is set to a value that is less than the time it takes for the radio to re-establish the connection it will place the radio back into sleep mode. When in listen mode the time to be considered is the time it takes the radio to associate to the target network (this must include any authentication delays that may be introduced for the Enterprise authentication processes). When in pass mode you must account for the additional network set-up time and packet delivery. We do not recommend setting wl-sleep-timer to a value less than 6 seconds.

18.0 Digital GPIO

The module supports two Digital GPIO ports. The two ports can be configured and written or read via the CLI interface, the following describes the functionality of the GPIO interface.

18.1 Available GPIO Interfaces

There are two GPIO ports available through the CLI interface. These ports are multipurpose and must be configured correctly for use as Digital GPIO. The ports different functions are mutually exclusive, with the exception of the LED indicator interface.

Table 35 - Port Type Summary

Port	Primary Use	Actions
f	UART1 and LED Indicators	`serial-port enable, conn-led enable, post-led enable, rf-link-led enable, wln-cfg-led enable` will restrict the number of available GPIO on the this port. `serial-port disable, conn-led disable, post-led disable, rf-link-led disable, wln-cfg-led disable` will allow all pins to be used as GPIO on the this port.
g	UART2	`serial-port-p2 enable` will restrict the number of available GPIO on the this port. `serial-port-p2 disable` will allow all pins to be used as GPIO on the this port.

Table 36 - Port f Configuration

Port		serial-port enable	serial-port disable
f	0	LED_POST	GPIO
	1	TXD1	GPIO
	2	LED_RF_LINK	GPIO
	3	LED_WLN_CFG	GPIO
	4	RTS1	GPIO
	5	CTS1	GPIO
	6	LED_CON	GPIO
	7	RXD1	GPIO

Table 37 - Port g Configuration

Port		serial-port-p2 enable	serial-port-p2 disable
g	0	GPIO	GPIO
	1	CTS2	GPIO
	2	RTS2	GPIO
	3	N/A	N/A
	4	N/A	N/A
	5	N/A	N/A
	6	RXD2	GPIO
	7	TXD2	GPIO

18.2 Default Configuration of GPIO

By default the GPIO interface is not enabled. It is necessary to reconfigure the GPIO pins as identified in section 18.1 through use of the commands and actions described in section 18.3.

18.3 Configuring GPIO ports

The available GPIO can be configured as inputs or outputs using a set of CLI commands. The commands listed in Table 38 provide control of the GPIO and should be configured to match the application.

Table 38 - GPIO Default Settings Command List

Command	Description
io-dir <portID> <state>	Sets the direction of the indicated port. This command sets the direction without requiring a restart or power cycle. This command is temporary and is not persistent across a restart or power cycle. To set the default direction of the ports the io-dir-f or io-dir-g commands must be used. The command has the same bit restrictions the io-dir-f and io-dir-g command have. The <portID> is a combination of the port name (g or f) and the bit to apply the state to (0 through 7), for instance g0 would affect the first pin on port g. The <state> can be set as either in or out depending upon the desired direction for the GPIO.

Command	Description
`io-pullup <portID> <state>`	Enables or disables the internal pull-up resistors for the specified GPIO pin. This command is temporary and is not persistent across a restart or power cycle. To set the default direction of the ports the `io-pullup-f` or `io-pullup-g` commands must be used. The command has the same bit restrictions the `io-pullup-f` and `io-pullup-g` command have. The internal pull-up resistor is enabled by default. The `<portID>` is a combination of the port name (`g` or `f`) and the bit to apply the state to (0 through 7), for instance `g0` would affect the first pin on port `g`. The `<state>` can be set as either `enable` or `disable`.
`io-dir-f <state>`	Sets the direction of the GPIO pins in port `f`. It is required to issue a `commit` after the command for the parameters to be persistent across restarts or power cycles. This command requires a `restart` or power cycle to be applied. For a pin to be an input it must be set to `1`, for output it must be set to `0`. The `<state>` for the pins is the decimal value of the 8 bit binary value that represents the desired state of the 8 GPIO in the port, e.g. `11111111 = 255` (all pins input), `11110000 = 240` (7,6,5,4 = Input, 3,2,1,0 = output). Requires that the primary UART and LED signals have been disabled.
`io-dir-g <state>`	Sets the direction of the GPIO pins in port `g`. It is required to issue a `commit` after the command for the parameters to be persistent across restarts or power cycles. This command requires a `restart` or power cycle to be applied. For a pin to be an input it must be set to `1`, for output it must be set to `0`. Note that pin 3,4 and 5 are ignored The `<state>` for the pins is the decimal value of the 8 bit binary value that represents the desired state of the 8 GPIO in the port, e.g. `11111111 = 255` (all pins input), `11110000 = 240` (7, 6= Input; 2, 1, 0 = output; 5, 4, 3 = ignored). Requires that the secondary UART has been disabled.

Command	Description
`io-pullup-f <state>`	Enables or disable the internal pull-up resistors of the GPIO pins in port `f`. It is required to issue a `commit` after the command for the parameters to be persistent across restarts or power cycles. This command requires a `restart` or power cycle to be applied. For a pin to be an input it must be set to `1`, for output it must be set to `0`. The `<state>` for the pins is the decimal value of the 8 bit binary value that represents the desired state of the 8 GPIO in the port, e.g. `11111111` = `255` (all pins input), `11110000` = `240` (7,6,5,4 = Input, 3,2,1,0 = output). Requires that the primary UART and LED signals have been disabled.
`io-pullup-g <state>`	Enables or disable the internal pull-up resistors of the GPIO pins in port `g`. It is required to issue a `commit` after the command for the parameters to be persistent across restarts or power cycles. This command requires a `restart` or power cycle to be applied. For a pin to be an input it must be set to `1`, for output it must be set to `0`. Note that pin 3,4 and 5 are ignored The `<state>` for the pins is the decimal value of the 8 bit binary value that represents the desired state of the 8 GPIO in the port, e.g. `11111111` = `255` (all pins input), `11110000` = `240` (7, 6= Input; 2, 1, 0 = output; 5, 4, 3 = ignored). Requires that the secondary UART has been disabled.
`conn-led <state>`	Enables or disables the CONN LED to allow the pin to be used as a GPIO. The `<state>` can be set as either `enable` or `disable`.
`post-led <state>`	Enables or disables the POST LED to allow the pin to be used as a GPIO. The `<state>` can be set as either `enable` or `disable`.
`rf-link-led <state>`	Enables or disables the RF_LINK LED to allow the pin to be used as a GPIO. The `<state>` can be set as either `enable` or `disable`.
`wln-cfg-led <state>`	Enables or disables the WLN_CFG LED to allow the pin to be used as a GPIO. The `<state>` can be set as either `enable` or `disable`.
`serial-port-pX <state>`	Enables or disables the primary serial port (UART1). The `<state>` can be set as either `enable` or `disable`.

If your system uses pull-up resistors on the circuit assembly then it is not necessary to enable the internal pull-up resistors available on the device server, to do this issue `io-pullup-f disable` or `io-pullup-g disable` and `commit` the parameter.

18.4 Using GPIO ports

Once enabled the GPIO ports can written to or read using the CLI interface. Table 39 shows the commands and their use.

Table 39 - GPIO Read/Write CLI Commands

Command	Description
`io-read <portID>`	Reads the state of the GPIO pin identified by the `<portID>`. The `<portID>` is a combination of the port name (`g` or `f`) and the bit to read (0 through 7), for instance `g0` would read the first pin on port `g`. The command requires the `<portID>` be set to `input`.
`io-write <portID> <state>`	Writes the value of `<state>` to the GPIO pin identified by the `<portID>`. The `<portID>` is a combination of the port name (`g` or `f`) and the bit to read (0 through 7), for instance `g0` would read the first pin on port `g`. The `<state>` can equal `1` or `0`. The command requires the `<portID>` be set to `output`.

19.0 Command Descriptions

The following section will describe the commands relating specifically to the Airborne Enterprise Device Server and Ethernet Bridge family.

The CLI interface provides the following on-line help support:

1. Trailing a command with a `?` will return a description of the command function and valid argument list e.g.

   ```
   pm-mode ?
   ```

   returns…

   ```
   Usage: pm-mode [active | doze]
   Sets the Module's power-management mode. Parameters are
   active and doze.
   Default is active.
   ```

2. Entering `?` (after authenticating with the module) will provide a full list of the available CLI commands.
3. Entering `?` after a partial command will return all commands that begin with the characters that precedes the `?`.

   For example:

   ```
   io?
   ```

   returns…

   ```
   io-dir
   io-dir-f
   io-dir-g
   io-pullup
   io-pullup-f
   io-pullup-g
   io-read
   io-write
   OK
   ```

? [Question Mark]

Command	? [Question Mark]
Arguments	none
Device Type	All
Default	none

Description This command provide text help and supports two use cases:

When used by itself at the command prompt it will cause the device server to display all available commands. The list is not device functionality sensitive. This response is identical to the help command.

When used as an argument with a command, the device server will display the arguments for the command and describe the function of the command as an ASCII text response. Note that there must be no other arguments with the command for the help to be displayed.

```
get-cfg ?

Usage: get-cfg [String]

Uses FTP to get a configuration file from an FTP server. It
uses the ftp-server-address, ftp-server-path, ftp-user, and
ftp-password to get the specified configuration file. The
filename should not include any path information. A save
command must be issued for the configuration file to be saved
in flash.
```

Note that there must be no other arguments with the command for the help to be displayed.

alt-subject-match

Command	alt-subject-match
Arguments	[string]
Device Type	All
Default	[blank]
Description	A string of entries, separated by semicolons that are matched against the alternative subject name of the authentication server certificate defined by the `ca-cert-filename` command333333. If this string is set, the server certificate is only accepted if it contains one of the entries in the alternative subject extension. The required string must be entered in the following format: TYPE:VALUE Where the supported types include EMAIL, DNS, URL The value format must match the set TYPE e.g.; EMAIL:guest@example.com DNS:server.example.com;DNS:server2.example.com

alt-subject-match2

Command	alt-subject-match2
Arguments	[string]
Device Type	All
Default	[blank]
Description	A string of entries, separated by semicolons that are matched against the alternative subject name of the authentication server certificate defined by the `ca-cert2-filename` command. If this string is set, the server certificate is only accepted if it contains one of the entries in the alternative subject extension. The required string must be entered in the following format: TYPE:VALUE Where the supported types include EMAIL, DNS, URL The value format must match the set TYPE e.g.; EMAIL:guest@example.com DNS:server.example.com;DNS:server2.example.com The string is used during the inner authentication phase.

apply-cfg

Command	apply-cfg
Arguments	serial | radio | ethernet | firewall | ports
Device Type	All
Default	0
Description	Applies the selected settings immediately, without requiring a restart.

`serial-p#`	Applies following serial port settings. Where `p#` can be `p1` or `p2`. The settings will apply to the port number indicated. The parameter maybe issued without a suffix, in this case the module will apply the configuration to the serial port the command was entered on. If the command was entered from a telnet session without the suffix, it will apply to serial port 1 (UART1). This parameter only applies to a serial and UART devices.	

	`bit-rate-p1`
	`parity-p1`
	`flow-p1`
	`data-bits-p1`
	`stop-bit-p1`
	`input-size-p1`
	`intf-type-p1`
	`serial-assert-p1`
		`bit-rate-p2`
		`parity-p2`
		`flow-p2`
		`data-bits-p2`
		`stop-bit-p2`
		`input-size-p2`
		`intf-type-p2`
		`serial-assert-p2`

`radio`	Applies following radio configurations:	

	`wl-ssid`
	`wl-type`
	`wl-chan`
	`wl-ip`
	`wl-subnet`
	`wl-gateway`
	`wl-udap`
	`wl-dhcp`
	`wl-dhcp-client`
	`wl-dns1`
	`wl-dns2`
	`wl-dhcp-mode`
	`wl-dhcp-interval`
	`wl-dhcp-fb`
	`wl-dhcp-acqlimit`
	`wl-dhcp-fbip`
	`wl-dhcp-fbsubnet`
	`wl-dhcp-fbauto`
	`wl-dhcp-fbper`
	`wl-con-led`
	`wl-security`
	`pw-wpa-psk`
	`pw-leap`
	`user-leap`
	`wl-auth`
	`wl-def-key`
	`wl-wpa-format`
		`wl-key1`
		`wl-key2`
		`wl-key3`
		`wl-key4`
		`wl-rate`
		`wl-region`
		`ca-cert-filename`
		`ca-cert2-filename`
		`client-cert-filename`
		`client-cert2-filename`
		`priv-key-filename`
		`priv-key2-filename`
		`dh-parm-filename`
		`dh-parm2-filename`
		`priv-key-password`
		`priv-key2-password`
		`eapfast-pac-filename`
		`eap-password`
		`eap-ident`
		`eap-anon-ident`
		`eap-phase1`
		`eap-phase2`
		`subject-match`
		`subject-match2`
		`alt-subject-match`
		`alt-subject-match2`
		`user-wpa-supp-filename`

`ethernet`	Applies following Ethernet port settings:
	`eth-ip`
	`eth-gateway`
	`eth-subnet`
	`telnet-port`
	`http-port`
	This parameter only applies to the Ethernet device.

`firewall`	Applies following Ethernet port settings:
	`wl-route-default`
	`eth-route-default`
	`wl-route`
	`eth-route`
	This parameter only applies to the Ethernet device.

`ports`	Applies the following port settings:
	`telnet-port`
	`http-port`

Any settings applied with this command are temporary and will not be persistent across a restart or power cycle. Any settings applied by this command can be made persistent across restarts and power cycles by issuing the `commit` command.

arp-reachable-time

Command	arp-reachable-time
Arguments	[integer]
Device Type	All
Default	120
Description	The average amount of time before sending an ARP to each device in the ARP table. The actual rate is a random amount of time between 0.5 and 1.5 times this value. Value has the range of 1-254 seconds. The default time is 120 seconds. The device server requires a restart or power cycle for this parameter change to take effect.

arp-staleout-time

Command	arp-staleout-time
Arguments	[integer]
Device Type	All
Default	120
Description	The amount of time since the last observation of the IP address before scheduling that entry for removal from the device severs internal ARP table. Value has the range of 1-254 seconds. The default time is 120 seconds. The device server requires a restart or power cycle for this parameter change to take effect.

auth-level

Command	auth-level
Arguments	[ASCII Text: command] [Integer: 1 - 5]
Device Type	All
Default	[blank]
Description	Changes the required authentication (user) level for a given command. The command requires two arguments:

`command`	This identifies the Command Line Interface (CLI) command whose authentication level will be changed by the command. Supported commands: `reset`

`level`	Identifies the authentication level required to execute the command.
	`0 = connectionless (L0)`
	`1 = connection, not logged in (L1)`
	`2 = data (L2)`
	`3 = config (L3)`
	`4 = OEM (L4)`
	`5 = Manufacturing (manuf) (L5)`
	The value cannot be lower than the default value for the command.

Changing the commands authentication level will restrict use of the command by users who do not have the required authentication levels for the command.

blink-post-led

Command	blink-post-led
Arguments	on | off
Device Type	All
Default	[blank]
Description	Changes the state of the POST LED. This function allows the identification of the unit being talked to by the network system, through physical indicator.

on	Causes the POST LED output to blink
off	Causes the POST LED to return to normal operation

bit-rate

Command	bit-rate | bit-rate-p1
Arguments	300 | 600 | 1200 | 2400 | 4800| 9600 | 14400 | 19200 | 28800 | 57600 | 115200 | 230400 | 460800 | 921600
Device Type	UART | Serial | Ethernet
Default	9600
Description	Sets the bit-rate of serial port 1 (UART1) in bits per second. Use of the `–p1` suffix on the command is optional.

bit-rate-p2

Command	bit-rate-p2
Arguments	300 | 600 | 1200 | 2400 | 4800| 9600 | 14400 | 19200 | 28800 | 57600 | 115200 | 230400 | 460800 | 921600
Device Type	UART | Serial | Ethernet
Default	9600
Description	Sets the bit-rate of serial port 2 (UART2) in bits per second.

ca-cert-filename

Command	ca-cert-filename
Arguments	[ASCII Text: CA filename.extension]
Device Type	All
Default	none
Description	This command defines the Certificate Authority (CA) filename to be used with the chosen authentication method. The certificate can contain one or more trusted CA certificates. A trusted CA certificate should always be configured when using EAP-TLS, EAP-TTLS or PEAP. The file must be in PEM or DER format for the device server to recognize it as a valid certificate.

ca-cert2-filename

Command	ca-cert2-filename
Arguments	[ASCII Text: CA filename.extension]
Device Type	All
Default	none
Description	This command defines a second Certificate Authority (CA) filename to be used with the chosen authentication method. The certificate can contain one or more trusted CA certificates and is used during the inner authentication. A trusted CA certificate should always be configured when using EAP-TLS, EAP-TTLS or PEAP. The file must be in PEM or DER format for the device server to recognize it as a valid certificate.

cfg-dump

Command	cfg-dump
Arguments	active | factory | oem | user | wpa | enc
Device Type	All
Default	<none>

Description

Lists current configuration of the module.

The command lists all parameter settings including those not yet committed.

`[no parameter]`	Lists current configuration (all parameters).
`active`	Lists the current active configuration (all parameters).
`factory`	Lists the factory default configuration (all parameters).
`oem`	Lists the OEM configuration (all parameters). If `oem_config.txt` does not exist no parameters will be returned.
`user`	Lists the saved user configuration (all parameters). If `user_config.txt` does not exist no parameters will be returned.
`wpa`	Lists the contents of the WPA supplicant configuration file. This is the contents of `wpa-supplicant.conf` or the file defined by `user-wpa-supp-filename` cli command.
`enc`	Lists the contents of the encrypted user configuration file to the screen. If `user_enc_config.uue` does not exist no parameters will be returned.

The configuration dump will not include passwords or other private security related fields.

cfg-encrypt

Command	cfg-encrypt
Arguments	enable | disable | protected | protected | permanent
Device Type	All
Default	disable
Description	Enables or disables encrypting wireless keys in the module's configuration files.

`enable`	Wireless keys are stored in a separate, encrypted configuration file (`user_enc_config.uue`).
	The following parameters are affected:
	`pw`
	`pw-cfg`
	`pw-leap`
	`pw-manuf`
	`pw-oem`
	`pw-root`
	`pw-wpa-psk`
	`wl-key-1`
	`wl-key-2`
	`wl-key-3`
	`wl-key-4`
	`ftp-password`
	`ssh-default-password`
	`eap-password`
	`is-psk-calc`
	`pre-calc-psk`
	`priv-key-password`
	`priv-key2-password`
	The files will be split after a `commit` and `restart` or power cycle has been completed.

`disable`	Wireless keys are visible as plaintext in the configuration file (`user_config.txt, oem_config.txt`). If `cfg-encrypt disable` is later reconfigured to `cfg-encrypt enable`, the two configuration files will be remerged into a single plaintext `user_config.txt` file upon the next `commit`. Level 4 (OEM) users can issue this command.
`locked`	Wireless keys are stored in a separate, encrypted configuration file (`user_enc_config.uue`). The list of protected parameters is shown in the `enable` option. Only L5 (Manufacturer) users can clear this setting. To clear the setting, the `clear cfg-encrypt` command must be used. When the command is used the `cfg-encrypt` is returned to `enable`.

`protected`	Wireless keys are stored in a separate, encrypted configuration file (`user_enc_config.uue`). The list of protected parameters is shown in the `enable` option. Only L5 (Manufacturer) users can clear this setting. To clear the setting, the `clear cfg-encrypt` command must be used. When the command is used the `cfg-encrypt` is returned to `disable`. The `user_enc_config.uue` is deleted and all settings are lost from the configuration. Caution should be taken when using this option as it may impact the user's ability to connect to the module.
`permanent`	Wireless keys are stored in a separate, encrypted configuration file (`user_enc_config.uue`). The list of protected parameters is shown in the `enable` option. Only L5 (Manufacturer) users can clear this setting. To clear the setting, the `reset` command must be used. When the command is used the module is returned to OEM defaults.

clear

Command	clear
Arguments	ca-cert-filename | ca-cert2-filename | client-cert-filename | client-cert2-filename | priv-key-filename | priv-key2-filename | dh-parm-filename | dh-parm2-filename | priv-key-password | priv-key2-password | eapfast-pac-filename | eap-password | eap-ident | eap-anon-ident | eap-phase1 | eap-phase2 | subject-match | subject-match2 | alt-subject-match | alt-subject-match2 | user-wpa-supp-filename | ssh-password-default | cfg-encrypt
Device Type	All
Default	[blank]
Description	Removes specified parameter value from the user configuration. You must `commit` the changes in order for the user credentials to be permanently cleared from the module. Clearing any single security credential from the module may impact your ability to regain a wireless network connection.

clear-buf

Command	clear-buf | clear-buf-p1
Arguments	none
Device Type	Serial, UART, SPI
Default	[blank]
Description	Clears all data from the Serial 1 (UART1) buffers. When issued after a serial-assert xoff, any data in the serial buffer will be cleared. The `clear-buf` command will not clear all the output data pending for the SPI port. Any data queued for the next output transaction, prior to the command being issued, will be sent. Use of the `–p1` suffix is optional.

clear-buf-p2

Command	clear-buf-p2
Arguments	none
Device Type	Serial, UART, SPI
Default	[blank]
Description	Clears all data from the Serial 2 (UART2) buffers. When issued after a serial-assert xoff, any data in the serial buffer will be cleared.

clear-cred

Command	clear-cred
Arguments	none
Device Type	All
Default	[blank]

Description	Removes all user credentials. You must save the changes in order for the user credentials to be permanently removed from the module.
	The affected parameters are:
	`ca-cert-filename`
	`ca-cert2-filename`
	`client-cert-filename`
	`client-cert2-filename`
	`priv-key-filename`
	`priv-key2-filename`
	`dh-parm-filename`
	`dh-parm2-filename`
	`priv-key-password`
	`priv-key2-password`
	`eapfast-pac-filename`
	`eap-password`
	`eap-ident`
	`eap-anon-ident`
	`eap-phase1`
	`eap-phase2`
	`subject-match`
	`subject-match2`
	`alt-subject-match`
	`alt-subject-match2`
	`user-wpa-supp-filename`
	Resets command to default:
	`pw-wpa-psk passphrase`
	Clears the following files::
	`EAP-FAST PAC`
	Clearing all security credentials from the device server may impact your ability to regain a wireless network connection..

clear-wep

Command	clear-wep
Arguments	none
Device Type	All
Default	[blank]
Description	Removes all WEP keys from the module. You must commit the changes in order for the WEP keys to be permanently removed from the module. ⚡ If you remove all the WEP keys from the module, you may be unable to regain a wireless network connection if the access points require them.

client-cert-filename

Command	client-cert-filename
Arguments	[ASCII Text: filename.extension]
Device Type	All
Default	none
Description	This command defines the Client certificate filename to be used with the chosen authentication method. A client certificate should always be configured when using EAP-TLS. The file must be in PEM or DER format for the device server to recognize it as a valid certificate.

client-cert2-filename

Command	client-cert2-filename
Arguments	[ASCII Text: filename.extension]
Device Type	All
Default	none
Description	This command defines a second Client certificate filename to be used with the chosen authentication method. The certificate is used during the inner authentication phase. A client certificate should always be configured when using EAP-TLS. The file must be in PEM or DER format for the device server to recognize it as a valid certificate.

conn-led

Command	conn-led
Arguments	enable | disable
Device Type	All
Default	enable
Description	Controls the function of the GPIO pin (F6) used for the LED_CON, pin 23.

The CONN LED indicates if a TCP connection or a data tunnel has been established. The specific functionality is described by the `wl-con-led` command.

enable	Defines the output of GPIO pin F6 as the CONN.
disable	Defines the GPIO pin F6 for use as a general purpose digital I/O pin.

The LED_CON must be disabled `for io-dir-f`, `io-pullup-f` and `io-write` to affect GPIO F6.

data-bits

Command	data-bits | data-bits-p1
Arguments	7 | 8
Device Type	UART | Serial | Ethernet
Default	8
Description	Sets the data bit length for serial port 1 (UART1) in bits. Use of the `–p1` suffix is optional.

data-bits-p2

Command	data-bits-p2
Arguments	7 | 8
Device Type	UART | Serial | Ethernet
Default	8
Description	Sets the data bit length for serial port 2 (UART2) in bits.

debug-port

Command	debug-port
Arguments	enable | disable
Device Type	All
Default	Determinded by the device type
Description	Enables or disables the Debug Serial Port.

enable	Enable the Serial Debug Port
disable	Disable the Serial Debug Port

Disabling the Serial Debug port can save power and is recommended during normal operation of the device.

del-cert

Command	del-cert
Arguments	[ASCII Text string]
Device Type	All
Default	[blank]
Description	Removes user certificates and private keys. The argument can be a filename or a wildcard for a group of one or more certificates to be deleted. You must save the changes in order for the user credentials to be permanently removed from the module. `del-cert *.*` : Will delete all certificates. `del-cert user*.*` : Will delete all certificates beginning with user It is required to issue the `save` command after this command to permanently delete the files from the device server.

del-cfg

Command	del-cfg
Arguments	[ASCII Text – filename]
Device Type	All
Default	<none>
Description	Deletes the specified configuration file form the module. Once the download is complete it is necessary for the `save` command to be issued, this will cause the configuration file to be deleted permanently from the device server. The following files can be deleted using this command:

`user_config.txt`	User configuration file. This file contains the user configuration commands and parameters.
`oem_config.txt`	OEM default configuration file. This contains the OEM default settings for the device server. These settings are installed upon the issuing of a factory reset command or hardware factory reset input.
`user_enc_config.uue`	Encrypted user configuration file. This file contains the encrypted user configuration commands and parameters.

del-eth-route

Command	del-eth-route
Arguments	[tcp | udp | icmp | bcast | all] [ip XXX.XXX.XXX.XXX] [port <integer>]
Device Type	Ethernet
Default	<none>
Description	Deletes the rule matching the defined parameters from the current firewall rules. All parameters must match for the rule to be deleted.

`tcp|udp|icmp|bcast|all`	Selects the protocol for the rule.
`ip xxx.xxx.xxx.xxx`	Defines the public network address the rule applies to. The `xxx.xxx.xxx.xxx` must represent a valid IP address, where `xxx` is an integer between 0 and 255, and that the resultant IP address is not 0.0.0.0.
`port <integer>`	Defines the port number for the rule. The port number must be an integer.

The following provides details for each of the parameters:

`icmp`	The rule impacts only ICMP traffic
`tcp`	The rule impacts only TCP/IP traffic.
`udp`	The rule impacts only UDP traffic.

del-wl-route

Command	del-wl-route
Arguments	[tcp | udp | icmp | bcast | all] [port <integer>]
Device Type	Ethernet
Default	<none>

Description Deletes the rule matching the defined parameters from the current port forwarding rules. All parameters must match for the rule to be deleted.

`tcp|udp|icmp|bcast|all`	Selects the protocol for the rule.
`port <integer>`	Defines the port number for the rule. The port number must be an integer.

The following provides details for each of the parameters:

`all`	The rule impacts all network traffic
`tcp`	The rule impacts only TCP/IP traffic.
`udp`	The rule impacts only UDP traffic.

dev-type

Command	dev-type
Arguments	none
Device Type	All
Default	<empty>
Description	Identifies the Airborne device type. The device type specifies the hardware configuration and the functionality of the module, the following list identifies the possible responses:

0	802.11b Airborne UART Module, WPA Security	WLNB-AN-DP2XX
1	802.11b Airborne UART Module, LEAP Security	WLNB-AN-DP5XX
2	802.11b AirborneDirect Serial Module, WPA Security	WLNB-SE-DP2XX ABDB-SE-DP2XX
3	802.11b AirborneDirect Serial Module, LEAP Security	WLNB-SE-DP5XX ABDB-SE-DP5XX
4	802.11b AirborneDirect Ethernet Module, WPA Security	WLNB-ET-DP2XX ABDB-ET-DP2XX
5	802.11b AirborneDirect Ethernet Module, LEAP Security	WLNB-ET-DP5XX ABDB-ET-DP5XX
6	802.11b Airborne SPI Module, WPA Security	WLNB-AN-DP202
7	802.11b Airborne UART Module, LEAP Security	WLNB-AN-DP502
8	802.11b/g Airborne UART Module, LEAP Security	WLNG-AN-DP2XX
9	802.11b/g AirborneDirect Ethernet Module, LEAP Security	WLNG-ET-DP2XX ABDG-ET-DP2XX
10	802.11b/g AirborneDirect Serial Module, LEAP Security	WLNG-SE-DP2XX ABDG-SE-DP2XX
11	802.11b/g Airborne SPI Module, LEAP Security	WLNG-AN-DP202
12	802.11b/g Airborne UART Module, Enterprise Security	WLNG-AN-DP5XX
13	802.11b/g AirborneDirect Ethernet Module, Enterprise Security	WLNG-ET-DP500
14	802.11b/g AirborneDirect Serial Module, Enterprise Security	WLNG-SE-DP5XX
15	802.11b/g Airborne SPI Module, Enterprise Security	WLNG-SP-DP5XX

device-type

Command	device-type
Arguments	uart | ethernet | serial | spi
Device Type	All
Default	Determined by the model number of the device
Description	Configures the personality of the Airborne module and configures ports to preset configurations.

`uart`	UART device server personality (WLNG-AN-DP5XX)
`ethernet`	Ethernet adapter personality (WLNG-ET-DP5XX)
`serial`	Serial (RS232/422/485) device server personality (WLNG-SE-DP5XX)
`spi`	SPI device server personality (WLNG-SP-DP5XX)
`Industrial Serial`	Industrial Serial adapter personality (ABDG-SE-IN5XXX)
`Industrial Ethernet`	Industrial Ethernet adapter personality (ABDG-ET-IN5XXX)

The ports configuration for each personality is preconfigured (enabled/disabled) and is shown in the following table:

Port	UART1	UART2	Ethernet	Debug
`uart`	enabled	enabled	disabled	enabled
`ethernet`	disabled	disabled	enabled	enabled
`serial`	enabled	enabled	disabled	enabled
`spi`	disabled	enabled	disabled	enabled
`Ind Serial`	enabled	enabled	disabled	enabled
`Ind Ethernet`	disabled	disabled	enabled	enabled

The port settings can be modified after the personality has been applied see `ethernet-port`, `serial-port`, `serial-port2` and `debug-port` commands for details.

> The SPI personality removes the availability of the UART1 and Ethernet ports since pins required for the SPI interface are used by these ports.
>
> Not all ports are available to boxed products.

dh-parm-filename

Command	dh-parm-filename
Arguments	[Private Key filename] with PEM extension.
Device Type	All
Default	[blank]
Description	DH/DSA parameters file name (in PEM format). This is an optional configuration file for setting parameters for an ephemeral DH key exchange. In most cases, the default RSA authentication does not use this configuration. However, it is possible to setup RSA to use ephemeral DH key exchange. In addition, ciphers with DSA keys always use ephemeral DH keys. This can be used to achieve forward secrecy. If the file is in DSA parameters format, it will be automatically converted into DH parameters.

dh-parm2-filename

Command	dh-parm2-filename
Arguments	[Private Key filename] with PEM extension.
Device Type	All
Default	[blank]
Description	DH/DSA parameters file name (in PEM format). The file is used during the inner authentication phase. This is an optional configuration file for setting parameters for an ephemeral DH key exchange. In most cases, the default RSA authentication does not use this configuration. However, it is possible to setup RSA to use ephemeral DH key exchange. In addition, ciphers with DSA keys always use ephemeral DH keys. This can be used to achieve forward secrecy. If the file is in DSA parameters format, it will be automatically converted into DH parameters.

discover

Command	discover
Arguments	none
Device Type	All
Default	<none>

Description

Initiates discovery of and lists all Airborne device servers. The device servers must be on the same physical network as the device that initiated the process.

A typical response will be:

```
Device Name                      IP Address       MAC Address   Device Type  FW Ver

-----------------------------------------------------------------------------------------

Veyron_1                         192.168.1.108    000B6B7784C5  AIRBORNE     1.02M
```

This process may take several seconds to respond.

The discovery process uses UDP broadcasts (255.255.255.255) for the discovery protocol, if your network infrastructure does not allow UDP broadcasts the discovery process will not work. In this case no devices will be discovered.

disk-free

Command	disk-free
Arguments	none
Device Type	All
Default	<none>
Description	Displays the approximate free space available on the internal flash disk in bytes.

dns-lookup

Command	dns-lookup
Arguments	[text string]
Device Type	All
Default	<none>
Description	Performs a DNS lookup using `dns-server1` and `dns-server2` as the primary and secondary DNS servers. The input string may be the fully qualified URL or the IP address of the network node: This command returns the IP address that was resolved by the DNS server or an error if not resolved. Responds with the IP address of the URL in a text string format: `xxx.xxx.xxx.xxx`

dns-server1

Command	dns-server1
Arguments	[ASCII Text – IP Address XXX.XXX.XXX.XXX]
Device Type	All
Default	<0.0.0.0>
Description	Configures the Primary DNS Server Address required for DNS lookups with the `dns-lookup` command. If the DHCP Client is enabled, the `dns-server1` value will be updated (if the DHCP Server provides one). Default is `0.0.0.0`.

dns-server2

Command	dns-server2
Arguments	[ASCII Text – IP Address XXX.XXX.XXX.XXX]
Device Type	All
Default	<0.0.0.0>
Description	Configures the Primary DNS Server Address required for DNS lookups with the `dns-lookup` command. If the DHCP Client is enabled, the `dns-server1` value will be updated (if the DHCP Server provides one). Default is `0.0.0.0`.

eap-anon-ident

Command	eap-anon-ident
Arguments	[text string]
Device Type	All
Default	[blank]
Description	Anonymous identity string for EAP. Max length of 64 ASCII characters. Used as the unencrypted identity with EAP types that support different tunneled identity, e.g., EAP-TTLS. Typical format anonident@example.com.

eap-fast-max-pac-list

Command	eap-fast-max-pac-list
Arguments	[Integer]
Device Type	All
Default	10
Description	Defines the maximum number of RADIUS servers for which EAP-FAST PAC provisioning is maintained. This is an integer with a range of 1-255 entries. Default is 10.

eap-fast-provisioning

Command	eap-fast-provisioning
Arguments	unauthenticated | authenticated | either
Device Type	All
Default	authenticated
Description	Defines the method by which EAP-FAST credentials (PAC) can be provisioned between the module and a RADIUS server.

`unauthenticated`	The server's identity is not validated before the credentials are provisioned.
`authenticated`	The server's identity is validated before the credentials are provisioned. Requires `ca-cert-filename` to be configured and certificate loaded to module. If not done the setting will behave the same as `unauthenticated`.
`either`	Instructs the module to use `authenticated` if possible, otherwise use `unauthenticated`.

The setting `unauthenticated` is less secure than `authenticated`.

The default setting is `authenticated`.

eap-ident

Command	eap-ident
Arguments	[text string]
Device Type	All
Default	[blank]
Description	Identity string for EAP. Typically the RADIUS server user login name. Max length of 64 ASCII characters.

eap-password

Command	eap-password
Arguments	[ASCII Text String] or [32hex Digits]
Device Type	All
Default	[blank]
Description	Password string for EAP. Max length of 64 ASCII characters. This field can include either the plaintext password (using ASCII or hex string) or a NtPasswordHash (16-byte MD4 hash of password) in hash:<32 hex digits> format. NtPasswordHash can only be used when the password is for MSCHAPv2 or MSCHAP (EAP-MSCHAPv2, EAP-TTLS/MSCHAPv2, EAP-TTLS/MSCHAP, LEAP). EAP-PSK (128-bit PSK), EAP-PAX (128-bit PSK), and EAP-SAKE (256-bit PSK) is also configured using this field. For EAP-GPSK, this is a variable length PSK.

eap-phase1

Command	eap-phase1
Arguments	peaplabel=0 | peaplabel=1| peapver=0 | peapver=1 | peap_outer_success=0 | include_tls_length=1| result_ind=1 | crypto_binding=0 | crypto_binding=1 | crypto_binding=2 |
Device Type	All
Default	[blank]
Description	Phase1 (outer authentication, i.e., TLS tunnel) parameters.

`peaplable=0`	Forces a new label to be used during key derivation when PEAPv1 or newer is being utilized. Most server PEAPv1 implementations use this value.
`peaplabel=1`	Forces a new label to be used during key derivation when PEAPv1 or newer is being utilized. Some servers may require this setting for use with PEAPv1.
`peapver=0`	Forces use of PEAPv0.
`peapver=1`	Forces use of PEAPv1.
`peap_outer_success=0`	Terminates PEAP authentication on tunneled EAP-Success. This is required with some RADIUS servers that implement draft-josefsson-pppext-eap-tls-eap-05.txt (e.g., Lucent NavisRadius v4.4.0 with PEAP in "IETF Draft 5" mode)
`include_tls_length=1`	Used to force supplicant to include TLS message length field in all TLS messages even if they are not fragmented,
`result_ind=1`	Used to enable EAP-SIM and EAP-AKA to use protected result indication.
`crypto_binding=0`	Do not use Crypto Binding for PEAPv0.
`crypto_binding=1`	Use Crypto Binding for PEAPv0, if the server supports it (default).
`crypto_binding=2`	Require Crypto Binding for PEAPv0.

eap-phase2

Command	eap-phase2
Arguments	auth=MSCHAPV2 | autheap=MSCHAPV2 | autheap=MD5
Device Type	All
Default	[blank]
Description	Phase2 (inner authentication used with TLS tunnel) parameters.

`auth=MSCHAPV2`	Sets the inner encryption to MSCHAPv2. Required for EAP-PEAPv0 or EAP-PEAPv1.
`autheap=MSCHAPV2`	Sets the inner encryption to MSCHAPv2. Required for EAP-TTLS/MSCHAPv2
`autheap=MD5`	Sets the inner encryption to MD5. Required for EAP-TTLS/MD5.

This is a string with field-value pairs, e.g., "auth=MSCHAPV2" for EAP-PEAP or autheap=MSCHAPV2 autheap=MD5" for EAP-TTLS).

The following certificate/private key fields are used in inner Phase2 authentication when using EAP-TTLS or EAP-PEAP:

```
ca-cert2-filename
client-cert2-filename
priv-key2-filename
priv-key2-password
dh-parm2-filename
subject_match2
altsubject_match2
```

escape

Command	escape
Arguments	[ASCII Hex]
Device Type	All
Default	7E7E7E6473 (equivalent to `~~~ds`) {add ABDG serial sequence}
Description	Sets the escape string sequence. The sequence must be 5 ASCII characters long, equivalent to 10 ASCHEX digits. Can be set to a desired sequence or can be disabled with the `off` argument.

> This command has been deprecated, it is recommended that the following commands be used to set the escape string and enable or disable its use.
>
> `esc-str` or `esc-str-p2`
> `esc-str-p2`
> `esc-mode-serial` or `esc-mode-serial-p2`
> `esc-mode-serial-p2`
> `esc-mode-lan` or `esc-mode-lan-p2`
> `esc-mode-lan-p2`

esc-mode-lan

Command	esc-mode-lan | esc-mode-lan-p1
Arguments	off | on
Device Type	UART, Serial
Default	on
Description	Configures the escape processing mode for the wireless interface when a tunnel has been established with Serial 1 (UART1) port.

on	Enables escape sequence checking on the wireless interface.
off	Disables escape sequence checking on the wireless interface.

If escape sequence checking is disabled it will not be possible to break from a data tunnel using the wireless interface connection.

Use of the `-p1` suffix on the command is optional.

esc-mode-lan-p2

Command	esc-mode-lan-p2
Arguments	off | on
Device Type	UART, Serial
Default	on
Description	Configures the escape processing mode for the wireless interface when a tunnel has been established with Serial 2 (UART2) port.

on	Enables escape sequence checking on the wireless interface.
off	Disables escape sequence checking on the wireless interface.

If escape sequence checking is disabled it will not be possible to break from a data tunnel using the wireless interface connection.

esc-mode-serial

Command	esc-mode-serial | esc-mode-serial-p1
Arguments	off | on | brk
Device Type	UART, Serial
Default	on
Description	Configures the escape processing mode for the Serial 1 (UART1) interface when a tunnel has been established.

on	Enables escape sequence checking on the Serial 1 (UART1) interface.
off	Disables escape sequence checking on the Serial 1 (UART1) interface.
brk	Enables escape on UART Break checking on the Serial 1 (UART1) interface.

If escape sequence checking is disabled it will not be possible to break from a data tunnel using the Serial 1 (UART1) interface port.

Use of the `–p1` suffix on the command is optional.

esc-mode-serial-p2

Command	esc-mode-serial-p2
Arguments	off | on
Device Type	UART, Serial
Default	on
Description	Configures the escape processing mode for the Serial 2 (UART2) interface when a tunnel has been established.

on	Enables escape sequence checking on the Serial 2 (UART2) interface.
off	Disables escape sequence checking on the Serial 2 (UART2) interface.
brk	Enables escape on UART Break checking on the Serial 2 (UART2) interface.

If escape sequence checking is disabled it will not be possible to break from a data tunnel using the Serial 2 (UART2) interface port.

esc-str

Command	esc-str | esc-str-p1
Arguments	[ASCII Hex]
Device Type	All
Default	7E7E7E6473 (UART) | FF7E414244 (Serial)
Description	Sets the escape string sequence for data tunnels using the Serial 1 (UART1) port, this string will apply to both the serial and wireless interfaces. The sequence must be 5 ASCII characters long, equivalent to 10 ASCHEX digits.

7E7E7E6473	~~~ds
FF7E414244	~ABD.

Use of the `-p1` suffix on the command is optional.

esc-str-p2

Command	esc-str-p2
Arguments	[ASCII Hex]
Device Type	All
Default	7E7E7E6473 (UART) | FF7E414244 (Serial)
Description	Sets the escape string sequence for data tunnels using the Serial 2 (UART2) port, this string will apply to both the serial and wireless interfaces. The sequence must be 5 ASCII characters long, equivalent to 10 ASCHEX digits.

7E7E7E6473	~~~ds
FF7E414244	~ABD.

eth-dhcp

Command	eth-dhcp
Arguments	enable | disable
Device Type	Ethernet
Default	disable

Description

Configures the DHCP client on the Ethernet interface to be enabled or disabled. If the DHCP client is enabled the Ethernet interface will use DHCP to obtain an IP configuration.

If DHCP fails the Ethernet interface configuration will be determined by the setting for the `eth-dhcp-fb` command.

`enable`	Enables DHCP (Client) on the Ethernet interface.
`disable`	Disbale DHCP (Client) on the Ethernet interface.

🛠 If `eth-dhcp` is enabled, `wl-dhcp` must be disabled and vise versa. The Ethernet DHCP client and the Wireless DHCP Client cannot both be enabled at the same time.

The default setting is `disable`.

eth-dhcp-acqlimit

Command	eth-dhcp-acqlimit
Arguments	[Integer]
Device Type	All
Default	150
Description	Configures the number of seconds that the Module should wait to acquire its IP configuration using DHCP before applying the DHCP fallback algorithm for the Ethernet interface. Requires `eth-dhcp-fb` to be enabled for IP fallback to be utilized. This is an integer with a range of 1-255 seconds. Setting `eth-dhcp-acqlimit 0` will turn IP Fallback off for the Ethernet interface. The default setting is `150`.

eth-dhcp-client

Command	eth-dhcp-client
Arguments	[ASCII Text]
Device Type	All
Default	AirborneXXXXXX
Description	Configures the DHCP Client Host Name String to use in the DHCP requests for the Ethernet interface. Up to 31 ASCII characters. Default is `AirborneXXXXXX` where `XXXXXX` are the last six hexadecimal digits of the Module's Ethernet MAC address.

eth-dhcp-clients

Command	eth-dhcp-clients
Arguments	<none>
Device Type	Ethernet
Default	<none>

Description

Displays a list of the leased IP addresses on the Ethernet interface. The client to which the address has been leased is identified by its MAC address.

The following is an example of the output from this command:

```
Client Address           DHCP Address
00:21:70:76:96:4F        192.168.2.100
00:21:70:76:EF:10        192.168.2.101
00:0B:6B:77:84:C5        192.168.2.102
```

It is important to note that all device listed by the command may not be available. The list provides leased addresses only and does confirm availability of the device prior to the list being displayed.

eth-dhcp-fb

Command	eth-dhcp-fb
Arguments	enable | disable
Device Type	Ethernet
Default	Disable (UART and Serial), Enable (SPI and Ethernet)

Description

Configures DHCP client fallback on the Ethernet interface. If the DHCP client fails to successfully complete DHCP before the `eth-dhcp-acqlimit` time is exceeded, the Ethernet interface will use the fallback settings for the modules IP configuration.

`enable`	Enables DHCP (Client) fallback on the Ethernet interface. Will use the settings from `eth-dhcp-fbip`, `eth-dhcp-fbgateway` and `eth-dhcp-subnet` for the Ethernet IP configuration.
`disable`	Disables DHCP (Client) fallback on the Ethernet interface.

If `eth-dhcp-fb` is disabled and DHCP fails, the Ethernet interface configuration will use `0.0.0.0` for the `eth-ip` and `eth-subnet` values. The `eth-gateway` will use the `wl-gateway` setting.

The default setting is `disable` for the UART and Serial devices and `enable` for the SPI and Ethernet devices.

eth-dhcp-fbauto

Command	eth-dhcp-fbauto
Arguments	0 | 1
Device Type	Ethernet
Default	0
Description	Enabling this will cause the module to set the `eth-dhcp-fbip`, `eth-dhcp-fbgateway`, `eth-dhcp-fbsubnet`, `dns-server1` and `dns-server2` to their current values each time an IP configuration is successfully received during a DHCP process.

`0`	Disables DHCP (Client) auto fallback configuration assignment on the Ethernet interface.
`1`	Enables DHCP (Client) auto fallback configuration assignment on the Ethernet interface. Will store the settings for `eth-dhcp-fbip`, `eth-dhcp-fbgateway` and `eth-dhcp-fbsubnet`, `dns-server1` and `dns-server2` for the DHCP Ethernet IP configuration.

This command requires that `eth-dhcp-fb` is enabled and the `eth-dhcp-acqlimit` is not `0` (zero)

> If `eth-dhcp-fbper` is disabled the assigned configuration from `eth-dhcp-fbauto` will not be persistent across restarts or power cycles.

The default setting is `0`.

eth-dhcp-fbgateway

Command	eth-dhcp-fbgateway
Arguments	[ASCII Text]
Device Type	Ethernet
Default	0.0.0.0
Description	Defines the gateway address used when DHCP fallback configuration is used by the Ethernet port. This setting requires that `eth-dhcp-fb` is enabled and the `eth-dhcp-acqlimit` is not `0` (zero). The default setting is `0.0.0.0`.

eth-dhcp-fbip

Command	eth-dhcp-fbip
Arguments	[ASCII Text]
Device Type	Ethernet
Default	0.0.0.0 – SPI and Ethernet, 192.168.10.1 – UART, Serial
Description	Defines the IP address used when DHCP fallback configuration is used by the Ethernet port. This setting requires that `eth-dhcp-fb` is enabled and the `eth-dhcp-acqlimit` is not `0` (zero). The default setting is `0.0.0.0` for SPI and Ethernet and `192.168.10.1` for UART and Serial devices.

eth-dhcp-fbper

Command	eth-dhcp-fbper
Arguments	0 | 1
Device Type	Ethernet
Default	0
Description	Enabling this will cause `eth-dhcp-fbip`, `eth-dhcp-fbgateway`, `eth-dhcp-fbsubnet`, `dns-server1` and `dns-server2` to be saved to memory each time they change, making them persistent across restarts and power cycles. This command requires that `eth-dhcp-fb` and `eth-dhcp-fbauto` are enabled and that `eth-dhcp-acqlimit` is not `0` (zero).

`0`	Disables fallback persistence.
`1`	Enables fallback persistence.

The default setting is `0`.

eth-dhcp-fbsubnet

Command	eth-dhcp-fbsubnet
Arguments	[ASCII Text – Subnet Mask]
Device Type	Ethernet
Default	255.255.255.0
Description	Defines the subnet mask applied when DHCP fallback configuration is used by the Ethernet port. This setting requires that `eth-dhcp-fb` is enabled and the `eth-dhcp-acqlimit` is not `0` (zero). The default setting is `255.255.255.0`.

eth-dhcp-rel

Command	eth-dhcp-rel
Arguments	[none]
Device Type	Ethernet
Default	[none]
Description	Releases the current DHCP leased IP address on the Ethernet port. This command must be issued before the `eth-dhcp-renew` command can be issued to obtain a new IP address.

eth-dhcp-renew

Command	eth-dhcp-renew
Arguments	[none]
Device Type	Ethernet
Default	[none]
Description	Performs a DHCP renew request for the Ethernet port, either to obtain a new IP configuration or update the DHCP lease with the DHC server. To obtain a new IP configuration the `eth-dhcp-rel` command must be issued before issuing this command.

eth-dhcp-server

Command	eth-dhcp-server
Arguments	disable | enable
Device Type	Ethernet
Default	disable – UART, Serial, SPI; enable - Ethernet

Description

Enables or Disables the DHCP server when the Ethernet interface mode is configures as a router. With the DHCP server enabled the Ethernet interface to provide IP configurations for any DHCP requests from clients on the Ethernet interface.

The issued DHCP configurations are determined as follows:

`disable`	Disables DHCP server on Ethernet interface.
`enable`	Enables DHCP server on Ethernet interface.

This command requires that `eth-role router` be configured.

The default setting is `0`.

eth-dhcp-vendorid

Command	eth-dhcp-vendorid
Arguments	[ASCII Text: Vendor ID]
Device Type	Ethernet
Default	[None]
Description	Configures the DHCP Vendor Class ID String to use in the DHCP requests for the Ethernet interface. Up to 31 characters. Default is an empty string.

eth-gateway

Command	eth-gateway
Arguments	[ASCII Text: Valid IP address]
Device Type	Ethernet
Default	192.168.2.1
Description	Configures the IP address of the Ethernet gateway. This is the IP address used by the client to communicate with the gateway (module). The IP address of the client and the Ethernet gateway must be in the same subnet for IP routing to work correctly. Must be ASCII text string with `XXX.XXX.XXX.XXX` format, where `XXX` can be `0-255`. The resultant IP address must not be 0.0.0.0. ⚡ The subnet for the wired IP and gateway IP addresses (Ethernet) and public IP address (802.11), obtained by the module via the wireless interface, and must not be the same.

eth-info

Command	eth-info
Arguments	[none]
Device Type	Ethernet
Default	[blank]

Description This command provides comprehensive status information on the Ethernet interface of the Airborne Device Server.

Example:

```
Module Firmware Version:        1.10
Link Status:                    Connected
Ethernet MAC Address:           000B280040D2
Link Speed:                     10Mb/s
Duplex:                         Full
IP Address:                     192.168.2.1
Subnet Mask:                    255.255.255.0
Default Gateway:                192.168.1.3
Primary DNS:                    192.168.1.3
Secondary DNS:                  192.168.1.4
Up Time (Sec):                  21854
```

eth-ip

Command	eth-ip
Arguments	[ASCII Text: Valid IP address]
Device Type	Ethernet
Default	192.168.2.100
Description	Configures the IP address of the wired interface client. If the wired interface client is using DHCP, the module will lease this address to the client in response to the DHCP request. If the client is not using DHCP, this address must match the static IP address on the client so that IP routing will work correctly. Any clients, using static IP addresses, must respond to ARP requests. The IP address of the client and the Ethernet gateway must be in the same subnet for IP routing to work correctly. Must be ASCII text string with `xxx.xxx.xxx.xxx` format, where `xxx` can be `0-255`. ⚡ The subnet for the wired IP and gateway IP addresses (Ethernet) and public IP address (802.11), obtained by the module via the wireless interface, and must not be the same.

eth-mac

Command	eth-mac
Arguments	[ASCHEX: 6 Bytes]
Device Type	Ethernet
Default	<varies>
Description	Configures the MAC address of the Ethernet interface. The input is 6 bytes ASCHEX with no colons e.g. `000B280040D2`. The value specified by the argument temporarily overwrites the factory value. For the change to be made the value must be committed and the device server restarted. When a reset is issued or a hardware factory reset is applied the Ethernet interface factory MAC value is recovered. Changing the MAC value must be done with caution. Only a known unique MAC value should be used.

eth-mode

Command	eth-mode
Arguments	auto | 10half |10full | 100half | 100full
Device Type	Ethernet
Default	auto
Description	Configures the connection rate for the wired Ethernet interface.

`auto`	Auto negotiate
`10half`	10Mbps, half duplex
`10full`	10Mbps, full duplex
`100half`	100Mbps, half duplex
`100full`	100Mbps, full duplex

eth-role

Command	eth-role
Arguments	[client | router]
Device Type	Ethernet
Default	client – Serial, router - Ethernet
Description	Configures the Ethernet interface role and determines the packet handling by the interface.

`client`	Disables packet forwarding between the wired and wireless interfaces.
`router`	Enables packet forwarding between the wired and wireless interfaces. Configuring the module and a NAT3 router.

The `router` mode is required when the device is configured as an Ethernet Adapter (WLNG-ET-DP5XX, ABDG-ET-DP5XX, ABDG-ET-IN5XXX).

The `client` mode is preferred when the module is being used as a serial device server (WLNG-AN-DP5XX, WLNG-SE-DP5XX, ABDG-SE-DP5XX, ABDG-SE-IN5XXX).

eth-route

Command	eth-route
Arguments	[all | bcast | icmp | tcp | udp] [ip xxx.xxx.xxx.xxx] [port <integer>] [accept | drop]
Device Type	Ethernet
Default	[blank]

Description Sets a specific rule for incoming Ethernet traffic. Allowing control of which services, IP addresses and ports can be accessed on the public (WAN) network by Ethernet clients on the private network. Through the rules established by this command and the `eth-route-default` setting a device firewall can be constructed to limit unauthorized use of the wireless interface on the network it is enabled for.

`all|icmp|tcp|udp`	Selects the protocol for the rule.
`ip xxx.xxx.xxx.xxx`	Defines the public network address the rule applies to. The `xxx.xxx.xxx.xxx` must represent a valid IP address where `xxx` is an integer between 0 and 255. The resultant IP address must not be 0.0.0.0.
`port <integer>`	Defines the port number for the rule. The port number must be an integer.
`accept|drop`	Defines if the rule allows or blocks traffic.

The following provides details for each of the parameters:

`all`	Allows all traffic to be affected by the rule.
`bcast`	The rule impacts only broadcast traffic.
`icmp`	The rule impacts only ICMP traffic
`tcp`	The rule impacts only TCP/IP traffic.
`udp`	The rule impacts only UDP traffic.

`accept`	This option will allow traffic matching the rules conditions to be forwarded to the wireless interface.
`drop`	This option will stop traffic matching the rules conditions from being forwarded to the wireless interface.
`relay`	May only be used if the selected protocol is `bcast`, assigning the action to `relay` will cause UDP traffic with destination address `255.255.255.255` received on the specified port to be relayed to the wireless interface. If selected, the IP address `[IP Address:Port#]` should not be included in the rule.

Multiple rules can be established to support firewall requirements. The rules set by the `eth-route` command take precedence over the `eth-route-default` setting.

It is not required to include both the IP address and the port number when constructing a rule, if one is omitted the rule assumes it applies to all instances of the missing parameter. In the case of an IP address missing, all port accesses matching the listed value will be affected, regardless of the IP address. In the case of a missing port, all traffic matching the identified IP address will be impacted.

By default all broadcast traffic on the Ethernet interface is dropped. It is necessary to establish a broadcast forwarding rule for broadcast messages with the required port number to be relayed to the wireless interface.

Here are some examples of rules:

`eth-route tcp port 80 drop`	This will cause all TCP/IP traffic using port 80 to be dropped.
`eth-route all ip 192.168.2.10 drop`	This will cause all traffic to IP address 192.168.2.10 to be dropped.
`eth-route tcp ip 192.168.2.10 port 23 accept`	This will cause all TCP/IP traffic meant for IP address 192.168.2.10 on port 23 to be forwarded to the wireless interface.
`eth-route icmp ip 192.168.2.10 accept`	The will allow all ICMP traffic meant for ip address 192.168.2.10 to be forwarded to the wireless interface.

Entering the command with no parameters will display a list of the current Ethernet routing rules in the order they will be applied to incoming traffic.

eth-route-default

Command	eth-route-default
Arguments	[accept | drop]
Device Type	Ethernet
Default	[accept]

Description

Sets the default rule for incoming Ethernet traffic. Allowing or denying access to the public (wireless) network from the private (wired) network. Through the rules established by this command and the `eth-route`, setting a device firewall can be constructed to limit unauthorized use of the wireless interface on the network it is enabled for.

`accept`	Allows all Ethernet traffic meant for the public (wireless) network to be forwarded.
`drop`	Blocks all Ethernet traffic meant for the public (wireless) network.

If the `eth-route-default` is set to drop and no additional rules (using `eth-route`) are added no traffic will be forwarded from the wired to wireless networks.

eth-subnet

Command	eth-subnet
Arguments	[ASCII Text: Subnet Mask]
Device Type	Ethernet
Default	255.255.255.0
Description	Configures the subnet mask for the Ethernet gateway and wired interface client. Must be ASCII text string with `XXX.XXX.XXX.XXX` format, where `XXX` can be `0-255`.

eth-udap

Command	eth-udap
Arguments	[0 | 1]
Device Type	Ethernet
Default	[1]

Description

Configures the UDAP discovery feature to be enabled or disabled on the Ethernet interface.

The UDAP discovery feature is required for the device to be located when used with the Airborne Management Center.

0	Disables the discovery protocol on the Ethernet interface.
1	Enables the discovery protocol on the Ethernet interface.

ethernet-port

Command	ethernet-port
Arguments	enable | disable
Device Type	All
Default	Determined by the device type configuration
Description	Enables or disables the Ethernet Port.

`enable`	Enable the Ethernet Port
`disable`	Disable the Ethernet Port

Disabling the Ethernet port can save power and is recommended during normal operation of the device, if the port is not in use.

flow

Command	flow | flow-p1
Arguments	n | h | s
Device Type	All
Default	n
Description	Defines the flow control for serial port 1 (UART1).

`n`	No Flow Control
`h`	Hardware flow control (RTS, CTS).
`s`	Software flow control (DC1 - XON, DC3 - XOFF).

Use of the `–p1` suffix on the command is optional.

flow-p2

Command	flow-p2
Arguments	n | h | s
Device Type	All
Default	n
Description	Defines the flow control for serial port 2 (UART2).

n	No Flow Control
h	Hardware flow control (RTS, CTS).
s	Software flow control (DC1 - XON, DC3 - XOFF).

ftp-filename

Command	ftp-filename
Arguments	[filename].[extension]
Device Type	All
Default	<blank>
Description	Defines the name of the firmware, certificate or configuration file to be uploaded or downloaded. If not specified, update ftp will uploaded the newest file in the target directory. Must be specified in order for the following command to function correctly: `update ftp`

ftp-password

Command	ftp-password
Arguments	[ASCII text: password]
Device Type	All
Default	<blank>

Description	Defines the password for the FTP account, associated to the FTP server defined by `ftp-server-address`.
	Must be specified in order for the following commands to function correctly:
	`update ftp`
	`get-cert`
	`get-cfg`

ftp-server

Command	ftp-server
Arguments	enable | disable | off
Device Type	All
Default	enable
Description	Enables or disables the access to the internal FTP server.

`enable`	Internal FTP server is enabled on all ports.
`disable`	Internal FTP server is disabled on all ports.
`off`	Internal FTP server in not loaded.

The FTP server is used for delivery of certificates, configuration files and device firmware. When disabled or not loaded these items cannot be delivered to the device server.

ftp-server-address

Command	ftp-server-address
Arguments	[Valid IP address] | [ACSII Text: FTP URL]
Device Type	All
Default	<blank>

Description	This value defines the IP address or URL of the target FTP server used for firmware, certificate or configuration file download.
	The IP address format follows the standard ASCII format XXX.XXX.XXX.XXX, where XXX = 1-254.
	The URL must be a valid and entered using ASCII text. The maximum length of the URL is 127 characters.
	Must be specified in order for the following commands to function correctly:
	`update ftp`
	`get-cert`
	`get-cfg`

ftp-server-listen-port

Command	ftp-server-listen-port
Arguments	[integer]
Device Type	All
Default	21
Description	Configures the port number the internal FTP server listens for connections on.

ftp-server-path

Command	ftp-server-path
Arguments	[ASCII text: directory path]
Device Type	All
Default	<blank>
Description	The path on the target FTP server that contains the firmware, certificate or configuration files to be downloaded. This does not need to be set if the file is in the default directory for the specified ftp-user. Example: `ftp-server-path /firmware/latest` This defines that the file to be uploaded resides in the /firmware/latest subdirectory of the FTP users root directory.

ftp-user

Command	ftp-user
Arguments	[ASCII text: username]
Device Type	All
Default	<blank>

Description	Defines the username for the FTP account, associated to the FTP server defined by ftp-server-address.
	Must be specified in order for the following commands to function correctly:
	`update ftp`
	`get-cert`
	`get-cfg`
	Please note that anonymous user credentials are not supported.

get-cert

Command	get-cert
Arguments	[ASCII Text – filename]
Device Type	All
Default	[blank]

Description	Will cause the device server to retrieve a certificate for the FTP server identified in the parameters defined by the following commands:
	`ftp-server-path`
	`ftp-server-address`
	`ftp-user`
	`ftp-password`
	`ftp-filename`
	Once the download is complete it is necessary for the save command to be issued, this will cause the certificate to be stored to the device server.
	For the Serial/UART/SPI device servers it is required that the device is associated and authenticated with a network and has a valid IP address before issuing this command.
	The Ethernet Bridge server supports the use of this command over the wired interface.

get-cfg

Command	get-cfg
Arguments	[ASCII Text – filename]
Device Type	All
Default	[blank]

Description

Will cause the device server to retrieve a configuration file from the FTP server identified in the parameters defined by the following commands:

```
ftp-server-path
ftp-server-address
ftp-user
ftp-password
```

Once the download is complete it is necessary for the save command to be issued, this will cause the configuration file to be stored to the device server.

There are two valid configuration files that may be down loaded:

`user_config.txt`	User configuration file. This file contains the user configuration parameter names and values.
`oem_config.txt`	OEM default configuration file. This contains the OEM default settings for the device server. These settings are installed upon the issuing of a factory reset command or hardware factory reset input.
`user_enc_config.uue`	Encrypted user configuration file. This file contains sensitive user configuration parameter names and values. See `cfg-encrypt` for dertails.

For the Serial/UART/SPI device servers it is required that the device is associated and authenticated with a network and has a valid IP address before issuing this command.

The Ethernet Bridge server supports the use of this command over the wired interface.

help

Command	help
Arguments	none
Device Type	All
Default	none
Description	This command provides text help. When used by itself at the command prompt it will cause the device server to display all available commands. The list is not device functionality sensitive. This response is identical to the `?` command, when used without a command.

http-port

Command	http-port
Arguments	disable | enable
Device Type	All
Default	enable

Description

Enables or disables access to the modules web browser via the wireless interface.

When enabled the module will transfer all HTTP traffic on the defined listening port (`wl-http-port`) to its internal HTTP server, when disabled all HTTP traffic will be forwarded to the wired interface.

enable	Enable HTTP access via the wireless and Ethernet ports.
disable	Disable HTTP access via the wireless port. Access via Ethernet interface is enabled.
off	Disable HTTP access via all network ports.

Configuring `http-port off` is preferred to `http-port disable` for controlling the access to the wireless port.

> Disabling the http-port will prevent any web interface connections from being accepted by the module on the wireless interface, limiting connections for web interface sessions to the wired interface only. This will restrict the management options available.
>
> This can be overcome by establishing a port forwarding rule that redirects incoming wireless traffic directed to a defined port on the wireless interface to the gateway address of the module using the HTTP port defined by `wl-http-port`.

input-size

Command	input-size | input-size-p1
Arguments	[Integer]
Device Type	UART | Serial | Ethernet
Default	1460
Description	Defines the serial input buffer size in bytes for serial port 1 (UART1). The input buffer size is the threshold at which the buffer will be flushed through the TCP connection. The size range is 1 – 1460 bytes. If software flow control is enabled the size range is 5 – 1460 bytes. Use of the `-p1` suffix on the command is optional.

input-size-p2

Command	input-size-p2
Arguments	[Integer]
Device Type	UART | Serial | Ethernet
Default	1460
Description	Defines the serial input buffer size in bytes for serial port 2 (UART2). The input buffer size is the threshold at which the buffer will be flushed through the TCP connection. The size range is 1 – 1460 bytes. If software flow control is enabled the size range is 5 – 1460 bytes.

intf-type

Command	intf-type
Arguments	rs232 | rs422 | rs485
Device Type	Serial
Default	rs232
Description	Sets the serial interface for RS-232, RS-422, or RS-485 communications. Enables interface pins 17, 19 and 22. (See 802.11b/g High Performance Device Server Product Specification for detailed description of pin function).

io-dir

Command	io-dir
Arguments	f<port number> | g<port number> [in | out]
Device Type	All
Default	f in | g in

Description

Sets the direction of the indicated GPIO port dynamically without restarting the module.

The command requires two parameters the first identifies the GPIO port and bit to be configured the second determines the default direction of the port. The command acts upon all GPIO in the identified port. For example:

`io-dir f1 out`: Sets the f port first bit to be an output

`io-dir g7 in`: Sets the g port seventh bit to be input

The effects of this command are temporary and will not be persistent across a restart. If the port and bit direction are required to persist across a power cycle or restart use the `io-dir-f` and `io-dir-g` commands.

The Port can be read or written to using the `io-read` and `io-write` commands.

Port assignment and exceptions:

f0	Read or Write (POST output)
f1	Read or Write
f2	Read or Write (RF_LINK output)
f3	Read or Write (WLN_CFG output)
f4	Read or Write
f5	Read or Write
f6	Read or Write (LED_CON output)
f7	Read or Write

g0	Read or Write
g1	Read or Write
g2	Read or Write
g3	N/A
g4	N/A
g5	N/A
g6	Read or Write
g7	Read or Write

When the LED signal has not been disabled those bits indicated as LED Outputs can only be read in order to determine the state of the LED output (See WLNG DP500 Family Databook for details). To disable the LED's use one of the following commands `post-led`, `rf-link-led`, `wln-cfg-led` and `conn-led`.

Any attempt to set an unavailable port or configure a port to an illegal state will be ignored.

io-dir-f

Command	io-dir-f
Arguments	[hex]
Device Type	All
Default	1

Description

Sets the direction of the f GPIO port to input or output.

The command requires a single hexidecimal value that represents the bit mask to be applied to the port.

`0`	Bit is set as an output
`1`	Bit is set as an input

Port	f0	f1	f2	f3	f4	f5	f6	f7
Value if Input	`1`	`2`	`4`	`8`	`16`	`32`	`64`	`128`

Any attempt to set an unavailable port or configure a port to an illegal state will be ignored.

This command requires a `commit` command to be made persistent.

Requires a restart to take effect.

The following exceptions apply:

`f0`	Read (POST output)
`f1`	Read or Write
`f2`	Read (RF_LINK output)
`f3`	Read (WLN_CFG output)
`f4`	Read or Write
`f5`	Read or Write
`f6`	Read (LED_CON output)
`f7`	Read or Write

When the LED signal has not been disabled those bits indicated as LED Outputs can only be read in order to determine the state of the LED output (See WLNG DP500 Family Databook for details). To disable the LED's use one of the following commands `post-led`, `rf-link-led`, `wln-cfg-led` and `conn-led`.

Any attempt to set an unavailable port or configure a port to an illegal state will be ignored.

io-dir-g

Command	io-dir-g
Arguments	0 | 1
Device Type	All
Default	0

Description

Sets the direction of the g GPIO port to input or output.

The command requires a single integer value that represents the state to be configured.

`0`	Bit is set as an output
`1`	Bit is set as an input

Any attempt to set an unavailable port or configure a port to an illegal state will be ignored.

This command requires a `commit` command to be made persistent.

Requires a restart to take effect.

The following exceptions apply:

`g0`	Read or Write
`g1`	Read or Write
`g2`	Read or Write
`g3`	N/A
`g4`	N/A
`g5`	N/A
`g6`	Read or Write
`g7`	Read or Write

io-pullup

Command	io-pullup
Arguments	f<port number> | g<port number> [enable | disable]
Device Type	All
Default	N/A

Description

Enables or disables the internal pull-up resistor on indicated GPIO port dynamically without restarting the module.

The command requires two parameters the first identifies the GPIO port and bit to be configured the second determines the state of the internal pull-up resistor, for example:

`io-pullup f1 enable`: Enables the internal pull up resistor for the f port first bit.

`io-pullup g7 disable`: Disables the internal pull-up resistor for the g port seventh bit.

The effects of this command are temporary and will not be persistent across a restart. If the state of the pull-up resistor is required to persist across a power cycle or a restart use the `io-pullup-f` and `io-pullup-g` commands.

Port assignment and exceptions:

f0	Read (POST output)	g0	Read or Write
f1	Read or Write	g1	Read or Write
f2	Read (RF_LINK output)	g2	Read or Write
f3	Read (WLN_CFG output)	g3	N/A
f4	Read or Write	g4	N/A
f5	Read or Write	g5	N/A
f6	Read (LED_CON output)	g6	Read or Write
f7	Read or Write	g7	Read or Write

When the LED signal has not been disabled those bits indicated as LED Outputs can only be read in order to determine the state of the LED output (See WLNG DP500 Family Databook for details). To disable the LED's use one of the following commands `post-led`, `rf-link-led`, `wln-cfg-led` and `conn-led`.

Any attempt to set an unavailable port or configure a port to an illegal state will be ignored.

io-pullup-f

Command	io-pullup-f
Arguments	1 | 0
Device Type	All
Default	1

Description

Enables or disables the internal pull-up resistor for the f GPIO port.

The command requires a single parameter that represents the state to be configured.

`1`	Enables the internal pull-up resistor
`0`	Disables the internal pull-up resistor

Any attempt to set a read only port will be ignored.

This command requires a `commit` command to be made persistent.

Requires a restart to take effect.

The following exceptions apply:

`f0`	Read (POST output)
`f1`	Read or Write
`f2`	Read (RF_LINK output)
`f3`	Read (WLN_CFG output)
`f4`	Read or Write
`f5`	Read or Write
`f6`	Read (LED_CON output)
`f7`	Read or Write

When the LED signal has not been disabled those bits indicated as LED Outputs can only be read in order to determine the state of the LED output (See WLNG DP500 Family Databook for details). To disable the LED's use one of the following commands `post-led`, `rf-link-led`, `wln-cfg-led` and `conn-led`.

Any attempt to set an unavailable port or configure a port to an illegal state will be ignored.

io-pullup-g

Command	io-pullup-g
Arguments	1 | 0
Device Type	All
Default	1

Description

Enables or disables the internal pull-up resistor for the g GPIO port.

The command requires a single parameter that represents the state to be configured.

`1`	Enables the internal pull-up resistor
`0`	Disables the internal pull-up resistor

Any attempt to set a read only port will be ignored.

This command requires a `commit` command to be made persistent.

Requires a restart to take effect.

The following exceptions apply:

`g0`	Read or Write
`g1`	Read or Write
`g2`	Read or Write
`g3`	N/A
`g4`	N/A
`g5`	N/A
`g6`	Read or Write
`g7`	Read or Write

io-read

Command	io-read
Arguments	f<port number> | g<port number>
Device Type	All
Default	N/A

Description

Reads the value of the indicated GPIO port and bit. This command is applied dynamically and does not require a restart.

The command requires a single parameter that identifies the GPIO port and bit to be read, for example:

`io-read f1`:	Reads the value of the f port first bit.
`io-read g7`:	Reads the value of the g port seventh bit.

The returned value will be a zero (0) or one (1) based upon the voltage applied to the input. Please refer to the Airborne DP500 Family data book for the GPIO electrical specification.

Port assignment and exceptions:

`f0`	Read (POST output)	`g0`	Read or Write
`f1`	Read or Write	`g1`	Read or Write
`f2`	Read (RF_LINK output)	`g2`	Read or Write
`f3`	Read (WLN_CFG output)	`g3`	N/A
`f4`	Read or Write	`g4`	N/A
`f5`	Read or Write	`g5`	N/A
`f6`	Read (LED_CON output)	`g6`	Read or Write
`f7`	Read or Write	`g7`	Read or Write

When the LED signal has not been disabled those bits indicated as LED Outputs can only be read. Issuing an `io-read` for any of these ports will return the current status of the LED output (See WLNG DP500 Family Databook for details). To disable the LED's use one of the following commands `post-led`, `rf-link-led`, `wln-cfg-led` and `conn-led`.

io-write

Command	io-write
Arguments	f<port number> | g<port number> [0 | 1]
Device Type	All
Default	N/A

Description

Writes the included value to the indicated GPIO port and bit. This command is applied dynamically and does not require a restart.

The command requires a parameter that identifies the GPIO port and bit to be write to and the state to which it is to be set, for example:

`io-write f1 1`: Writes the value 1 to the f port first bit.

`io-write g7 0`: Writes the value 0 to the g port seventh bit.

The written value, zero (0) or one (1), will be converted to a output voltage on the indicated port and pin. Please refer to the Airborne DP500 Family data book for the GPIO electrical specification.

Port assignment and exceptions:

f0	Read (POST output)	g0	Read or Write
f1	Read or Write	g1	Read or Write
f2	Read (RF_LINK output)	g2	Read or Write
f3	Read (WLN_CFG output)	g3	N/A
f4	Read or Write	g4	N/A
f5	Read or Write	g5	N/A
f6	Read (LED_CON output)	g6	Read or Write
f7	Read or Write	g7	Read or Write

When the LED signal has not been disabled those bits indicated as LED Outputs can only be read. To disable the LED's use one of the following commands `post-led`, `rf-link-led`, `wln-cfg-led` and `conn-led`.

The ports indicated as N/A cannot be written to. Any attempt to set an unavailable port will be ignored.

led-mode

Command	led-mode
Arguments	status | rssi
Device Type	All
Default	status
Description	Controls the function of the CONN, RF_LINK, and POST LEDs, defining their output as either indictors of the modules status or as a RF signal strength meter.

`status`	The three LED's provide feedback on the modules status. See the individual LED definitions for details.
`rssi`	The three LEDs function as a rudimentary RSSI (Signal Strength) meter. The signals have the following meaning in RSSI mode: COMM LED green: Signal Strength <= -80 dBm COMM and LINK LEDs green: -80dBm < Signal Strength < -60dBm All three LEDs green: Signal Strength >= -60 dBm

When using one of the AirborneDirect™ products the following LED names are used:

CONN	=	COMM
RF_LINK	=	LINK
POST	=	POWER

The three LED pins cannot be defined as GPIO for the `led-mode` command to function correctly. The LED pin function is configured using the `conn-led`, `rf-link-led` and `post-led` commands.

list-cert

Command	list-cert
Arguments	[None]
Device Type	All
Default	[None]
Description	Displays a list of all certificate files resident on the device server, including files that have been loaded but not saved.

list-cfg

Command	list-cfg
Arguments	[None]
Device Type	All
Default	[None]
Description	Displays a list of the configuration files resident on the device server, including files that have been loaded but not saved.

parity

Command	parity | parity-p1
Arguments	n | e | o
Device Type	All
Default	n
Description	Defines the parity bit for serial port 1 (UART1).

n	No Parity.
e	Even parity.
o	Odd parity.

Use of the –p1 suffix is optional.

parity-p2

Command	parity-p2
Arguments	n | e | o
Device Type	All
Default	n
Description	Defines the parity bit for serial port 2 (UART2).

n	No Parity.
e	Even parity.
o	Odd parity.

pass

Command pass | pass-p1

Arguments [none]

Device Type Serial, UART

Default [blank]

Description Creates a data bridge between the wireless and Serial 1 (UART1) interface. The behavior of the command depends upon the interface it is issued from and the mode the Serial 1 (UART1) interface is in.

Issuing Interface	UART1 State	Wireless State	Results
Wireless Interface	CLI	CLI	No data bridge formed.
	Listen	CLI	Data bridge formed.
	Pass	CLI	No data bridge formed.
UART1	CLI	N/A	Data bridge formed[1].

1. Network server must be available and that network server parameters have been configured correctly and that transport has been correctly defined. Please refer to section 6.3.3 for the configuration requirements.

Use of the `-p1` suffix is optional.

pass-any

Command	pass-any
Arguments	[none]
Device Type	Serial, UART
Default	[blank]

Description

Creates a data bridge between the wireless and one of the serial interfaces. The command can only be issued form a telnet connection and will create a data tunnel with the first serial interface (UART) found that is in the listen mode.

If both serial interfaces are in listen mode the Serial 1 (UART1) interface will be used before the Serial 2 (UART2) interface.

Issuing Interface	**UART1 State**	**UART2 State**	**Results**
Wireless or Ethernet Interface	CLI	CLI	No data bridge formed.
	Listen	CLI	Data bridge formed on UART1.
	Pass	CLI	No data bridge formed.
	CLI	Listen	Data bridge formed on UART2.
	Listen	Listen	Data bridge formed on UART1.
	Pass	Listen	Data bridge formed on UART2.
	CLI	Pass	No data bridge formed.
	Listen	Pass	Data bridge formed on UART1.
	Pass	Pass	No data bridge formed.

pass-p2

Command	pass-p2
Arguments	[none]
Device Type	Serial, UART
Default	[blank]
Description	Creates a data bridge between the wireless and Serial 2 (UART2) interface. The behavior of the command depends upon the interface it is issued from and the mode the Serial 2 (UART2) interface is in.

Issuing Interface	UART1 State	Wireless State	Results
Wireless Interface	CLI	CLI	No data bridge formed.
	Listen	CLI	Data bridge formed.
	Pass	CLI	No data bridge formed.
UART2	CLI	N/A	Data bridge formed[1].

1. Network server must be available and that network server parameters have been configured correctly and that transport has been correctly defined. Please refer to section 6.3.3 for the configuration requirements.

ping

Command	ping
Arguments	[IPAddress] | [ASCII Text: URL]
Device Type	All
Default	[blank]

Description

This command sends an ICMP ECHO_REQUEST to the specified destination address, and displays various statistics for the result.

The destination address can be an IP address or a website name (URL), such as www.quatech.com.

Example:

```
ping www.quatech.com

PING www.quatech.com (69.36.15.130): 56 data bytes
64 bytes from 69.36.15.130: seq=0 ttl=50 time=98.835 ms
64 bytes from 69.36.15.130: seq=1 ttl=50 time=100.134 ms
64 bytes from 69.36.15.130: seq=2 ttl=50 time=100.166 ms
64 bytes from 69.36.15.130: seq=3 ttl=50 time=97.474 ms

--- www.quatech.com ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 97.474/99.152/100.166 ms
OK
```

or

```
ping 192.168.1.105

PING 192.168.1.105 (192.168.1.105): 56 data bytes
64 bytes from 192.168.1.105: seq=0 ttl=64 time=1.210 ms
64 bytes from 192.168.1.105: seq=1 ttl=64 time=0.588 ms
64 bytes from 192.168.1.105: seq=2 ttl=64 time=0.587 ms
64 bytes from 192.168.1.105: seq=3 ttl=64 time=0.582 ms

--- 192.168.1.105 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 0.582/0.741/1.210 ms
OK
```

pm-mode

Command	pm-mode
Arguments	active | doze | sleep | wakeup
Device Type	All
Default	active
Description	Enables one of the available power-save modes.

Power save features are included in all aspects of the device server, however these specific modes change the sate of both the CPU and radio, it is important to note that use of these modes may impact data latency. The device server will automatically move into the power save mode when inactivity allows.

Parameter	Radio	Action/Condition
`active`	ON	The radio is constantly on.
`doze`	PS-Poll	The radio is operating in the IEE802.11 PS-Poll mode. While in this mode it will transition between active and deep sleep mode using a duty cycle determined by the beacon period and DTIM value provided by the AP. The device maintains association in the state.
`sleep`	Deep Sleep	The radio is in a deep sleep (Lowest power) mode. The device does not maintain association when in this state.
`wakeup`	ON/PS-Poll	Transitions the radio from deep sleep to the persistent setting for pm-mode (`active` or `doze`). Upon transitioning to the pm-mode the radio will attempt to reassociate with the wireless network.

State	CPU	Clock	Radio	Wake Requirements
`active`	ON	ON	ON	None
`doze`	OFF	OFF	PS-Poll	UART traffic, directed or broadcast radio traffic
`sleep`	OFF	OFF	Deep Sleep	UART traffic

The WLNB-AN-DP200 product family offered two additional modes `snooze` and `off`. Due to advancements in the CPU and radio technology there is no longer a need to differentiate between these modes and the ones available in the latest command description.

To support backward compatibility the device server will accept both the `snooze` and `off` parameters, however they will map as follows:

```
snooze  =  doze
off     =  sleep
```

The `pm-mode sleep` settings is dynamic and is applied without a power cycle or restart, however it is not persistent across power cycles or restarts. If a power cycle or restart is performed while the device is in sleep mode the persistent pm-mode the device was in prior to the `pm-mode sleep` being issued will be used (`pm-mode active` or `pm-mode doze`). The exception to this is the setting for the `radio-startup` command; please review this command for a full description of its use.

When `pm-mode sleep` is issued the device will immediately go in to deep sleep and loose association with the network. To bring the device out of sleep mode the `pm-mode wakeup` command must be issued. Once the `wakeup` command has been issued the radio will re-associate with the network, if it is still within coverage of the network.

> During sleep mode the radio looses association with the wireless network. Upon waking the radio re-authenticates and associates with the network. Some networks monitor the number of re-associations a client makes with the network and may block the client if it exceeds the networks limit.
>
> If the client is disassociated, after an amount of time, and can no longer connect to the network please contact the network's administrator to confirm this restriction should not be applied to the client.

The device server will automatically enter the sleep mode if the `wl-sleep-timer` is set to a value greater than zero (0), please refer to the `wl-sleep-timer` command for details on configuring this parameter.

To enter sleep automatically the UART/serial port must be in `listen` or `pass` mode. When in these modes and with the `wl-sleep-timer` set to an inactivity timeout value greater than zero (0). The radio will transition into sleep mode from its initial state once the inactivity (`wl-sleep-timer`) has expired. The radio will remain in the sleep mode until the UART/serial port receives a single character. Once received the radio and device server will return to their original states, prior to the inactivity timeout being triggered.

In the case of the UART/Serial port being in `pass` mode, upon waking from sleep mode the device server will continue to communicate on the established network connection or resume UDP transmission/reception. This assumes that the network socket has not been closed while the device server was in sleep mode. Since the sleep mode causes the device server to lose association, any TCP/IP keep alives from the network will not have been received by the module and are not necessary to maintain the TCP/IP timeout from expiring on the module. The radio will wake upon a single character being transmitted across the serial/UART port. Any data transferred through the UART while the radio is re-establishing the connection with the network will be buffered and transmitted upon successful completion of the connection.

In the case of the UART/Serial port being in `listen` mode, upon waking from sleep mode the device server will continue to listen for any attempted connections. It is important to note that any attempts to connect with the device server while it is in sleep mode will fail. To minimize any network traffic it is important for the network based application to be aware that the device server is in sleep mode and has been disconnected from the network.

post-led

Command	post-led
Arguments	enable | disable
Device Type	All
Default	enable
Description	Controls the function of the GPIO pin (F0) used for the LED_POST, pin 25.

enable	Defines the output of GPIO pin F0 as the POST. The POST LED will turn on when the Airborne adapter has successfully completed its power on self test.
disable	Defines the GPIO pin F0 for use as a general purpose digital I/O pin.

The LED_CON must be disabled `for io-dir-f`, `io-pullup-f` and `io-write` to affect GPIO F0.

priv-key-filename

Command	priv-key-filename
Arguments	[ASCII Text: filename.extension]
Device Type	All
Default	none
Description	This command defines the Client Private Key filename to be used with the chosen authentication method. When PKCS#12/PFX files are used the ca-cert-filename should not be used. The file must be in PEM or DER format for the device server to recognize it as a valid private key.

priv-key-password

Command	priv-key-password
Arguments	[ASCII Text: password]
Device Type	All
Default	[blank]
Description	This command defines the Client Private Key password to be used with the Private Key file identified by the priv-key-filename command. The private key is an ASCII text string provided by the generator of the Private Key file.

priv-key2-filename

Command	priv-key2-filename
Arguments	[ASCII Text: filename.extension]
Device Type	All
Default	none
Description	This command defines a second Client Private Key filename to be used with the chosen authentication method. The Private Key file is used during the inner authentication phase. When PKCS#12/PFX (.P22/.PFX)files are used for the private key the ca-cert-filename and user-cert-filename should not be used. The file must be in PEM, DER, PFX or P22 format for the device server to recognize it as a valid private key.

priv-key2-password

Command	priv-key2-password
Arguments	[ASCII Text: password]
Device Type	All
Default	[blank]
Description	This command defines the Client Private Key password to be used with the Private Key file identified by the priv-key2-filename command. The password is used during the inner authentication phase. The private key is an ASCII text string provided by the generator of the Private Key file.

put-cert

Command	put-cert
Arguments	[ASCII text: filename.extension]
Device Type	All
Default	none
Description	Will cause the device server to wait for an X-modem file transfer of certificate from the host device connected to the serial interface. Once the download is complete it is necessary for the save command to be issued, this will cause the certificate to be stored to the device server. It is required that the host use Xmodem 1K or Xmodem 1K-CRC. This command is supported via the serial interface or a telnet session.

put-cfg

Command	put-cfg
Arguments	user_config.txt | oem_config.txt
Device Type	All
Default	none
Description	Will cause the device server to wait for an Xmodem file transfer of the configuration file from the host device connected to the serial interface. Once the download is complete it is necessary for the `save` command to be issued, this will cause the configuration file to be stored to the device server. There are two valid configuration files that may be down loaded:

`user_config.txt`	User configuration file. This file contains the user configuration commands and parameters.
`oem_config.txt`	OEM default configuration file. This contains the OEM default settings for the device server. These settings are installed upon the issuing of a factory reset command or hardware factory reset input.
`user_enc_config.uue`	Encrypted user configuration file. This file contains sensitive user configuration parameter names and values. See `cfg-encrypt` for details.

It is required that the host use Xmodem 1K or Xmodem 1K-CRC.

This command is supported via the serial interface or a telnet session.

pw-root

Command	pw-root
Arguments	[ACSI Text]
Device Type	All
Default	rootpassword
Description	Configures the Administrator password (“root”). Password must be no longer than 31 ASCII characters and must not include spaces. It is recommended that the Administrator password be changed for all applications, failure to do so potentially leaves the module venerable to attack.

radio-off

Command	radio-off
Arguments	none
Device Type	All
Default	none
Description	Disables the 802.11b/g radio. After the command is issued the device server will close all TCP/IP and UDP connections and power down the radio. When in this state the device server will no longer be associated with a wireless network and any network based communication will not be possible. The device server will lose connection to the wireless network when this command is issued.

radio-on

Command	radio-on
Arguments	none
Device Type	All
Default	none
Description	Enables the 802.11b/g radio. The radio will attempt to regain a wireless network connection.

radio-startup

Command	radio-startup
Arguments	on | off | sleep
Device Type	All
Default	on

Description

This command defines the start-up state of the radio after a power cycle or restart. The command is persistent across power cycles and has significant impact on the operation of the device once the boot cycle has completed.

The options for this command are:

`on`	In this mode the radio is placed in the predefined pm-mode (active or doze) and will immediately attempt to associate with its configured SSID. This constitutes the normal operation of the device server and is the default value.
`off`	This mode is intended for those environments which prohibit radio transmission except under tightly controlled conditions. It is analogous to the *airplane mode* supported by mobile phones. In this mode, the radio driver is loads but the radio is immediately put into a deep sleep. The radio can only be awoken via the `radio-on` or `apply-cfg radio` commands.
`sleep`	In this mode the radio driver loads but the radio is immediately put into a deep sleep. The radio can be awoken by either a single character transmitted on the UART/serial interface or by the `pm-mode wakeup` command. This mode is intended for those applications with low frequency data transmissions.

rf-link-led

Command	rf-link-led
Arguments	enable | disable
Device Type	All
Default	enable
Description	Controls the function of the GPIO pin (F2) used for the LED_RF_LINK, pin 27.

enable	Defines the output of GPIO pin F3 as the RF_LINK. The RF_LINK LED turns on when the Airborne adapter has successfully authenticated with a WLAN.
disable	Defines the GPIO pin F2 for use as a general purpose digital I/O pin.

The LED_CON must be disabled `for io-dir-f`, `io-pullup-f` and `io-write` to affect GPIO F2.

save

Command	save
Arguments	none
Device Type	All
Default	<blank>
Description	Saves all user uploaded certificates, private keys and configuration files to flash. If `save` is not issued after uploading files, all files uploaded after the last `save` command, will be discarded and require uploading after next restart or power cycle.

serial-assert

Command	serial-assert | serial-assert-p1
Arguments	xon | xoff
Device Type	All
Default	xon
Description	Allows the serial port 1 (UART1) software flow control to be asserted or not. This command can be issued to a TCP based CLI session and cause the flow control to be applied immediately on serial port 1 (UART1). This commands argument can be made persistent across restarts or power cycles through issuing a commit after applying the command. The saved value will be applied at start-up. This command requires software flow control to be enabled, see `flow` for more details. Use of the –p1 suffix is optional.

serial-assert-p2

Command	serial-assert-p2
Arguments	xon | xoff
Device Type	All
Default	xon
Description	Allows the serial port 2 (UART2) software flow control to be asserted or not. This command can be issued to a TCP based CLI session and cause the flow control to be applied immediately on serial port 2 (UART2). This commands argument can be made persistent across restarts or power cycles through issuing a commit after applying the command. The saved value will be applied at start-up. This command requires software flow control to be enabled, see `flow-p2` for more details.

serial-default

<table>
<tr><td><b>Command</b></td><td>serial-default | serial-default-p1</td></tr>
<tr><td><b>Arguments</b></td><td>cli | listen | pass</td></tr>
<tr><td><b>Device Type</b></td><td>Serial, UART</td></tr>
<tr><td><b>Default</b></td><td>cli</td></tr>
<tr><td><b>Description</b></td><td>Configures the default mode for the Serial 1 (UART1) interface. The CLI server will use the defined mode at start-up of the device server.
<table>
<tr><td>cli</td><td>The interface will start in CLI mode as defined in section 6.3.1.
In this mode the UART will accept and process CLI commands. Authentication is required for access to the CLI server.</td></tr>
<tr><td>listen</td><td>The interface will start in listen mode as defined in section 6.3.5.
In this mode the Serial 1 (UART1) interface will accept requests to establish a data tunnel.</td></tr>
<tr><td>pass</td><td>The interface will start in pass mode as defined in sections 6.3.2 and 6.3.3.
In this mode the device server will attempt to make a connection with the defined network server. It is necessary that a wireless or Ethernet connection be established for this setting to be successful.
If the network server is not available the device server will continue to attempt to connect until the server becomes available or the Serial 1 (UART1) interface is interrupted by sending the escape sequence to the interface.</td></tr>
</table>
Use of the –p1 suffix is optional.</td></tr>
</table>

serial-default-p2

Command	serial-default-p2
Arguments	cli | listen | pass
Device Type	Serial, UART
Default	cli
Description	Configures the default mode for the Serial 2 (UART2) interface. The CLI server will use the defined mode at start-up of the device server.

cli	The interface will start in CLI mode as defined in section 6.3.1. In this mode the UART will accept and process CLI commands. Authentication is required for access to the CLI server.
listen	The interface will start in listen mode as defined in section 6.3.5. In this mode the Serial 2 (UART2) interface will accept requests to establish a data tunnel.
pass	The interface will start in pass mode as defined in sections 6.3.2 and 6.3.3. In this mode the device server will attempt to make a connection with the defined network server. It is necessary that a wireless or Ethernet connection be established for this setting to be successful. If the network server is not available the device server will continue to attempt to connect until the server becomes available or the Serial 2 (UART2) interface is interrupted by sending the escape sequence to the interface.

serial-port

Command	serial-port | serial-port-p1
Arguments	enable | disable
Device Type	All
Default	Determined by the device type configuration
Description	Enables or disables the Serial Port 1 (UART1).

enable	Enable the Serial Port 1 (UART1).
disable	Disable the Serial Port 1 (UART1)

Disabling the serial port can save power and is recommended during normal operation of the device, if the port is not in use.

Use of the –p1 suffix is optional.

serial-port-p2

<table>
<tr><td><b>Command</b></td><td>serial-port-p2</td></tr>
<tr><td><b>Arguments</b></td><td>enable | disable</td></tr>
<tr><td><b>Device Type</b></td><td>All</td></tr>
<tr><td><b>Default</b></td><td>Determined by the device type configuration</td></tr>
<tr><td><b>Description</b></td><td>Enables or disables the Serial Port 2 (UART2).
<table>
<tr><td>enable</td><td>Enable the Serial Port 2 (UART2).</td></tr>
<tr><td>disable</td><td>Disable the Serial Port 2 (UART2)</td></tr>
</table>
Disabling the serial port can save power and is recommended during normal operation of the device, if the port is not in use.</td></tr>
</table>

ssh-default-password

Command	ssh-default-password
Arguments	[ASCII Text]
Device Type	All
Default	<none>
Description	Configures the default password used to establish an SSH connection when the `pass` or `serial default pass` is used. Use CLI command `clear ssh-default-password` to remove password if not needed. Maximum of password is 32 ASCII characters. Must not use spaces.

ssh-default-user

Command	ssh-default-user
Arguments	[ASCII Text]
Device Type	All
Default	<none>
Description	Configures the default username used to establish an SSH connection when the `pass` or `serial-default pass` is used. Use CLI command `clear ssh-default-user` to remove password if not needed. Maximum length of user name 32 ASCII characters. Must not contain spaces.

ssh-keygen

Command	ssh-keygen
Arguments	none
Device Type	All
Default	<none>
Description	Generates the SSH keys, using the key length specified by `ssh-keysize`. You must issue a `commit` or `save` to store the generated keys. Key generation may take several seconds, the `OK` response will be returned by the device server when the keys have been generated.

ssh-keysize

Command	ssh-keysize
Arguments	[integer]
Device Type	All
Default	1024
Description	Defines the size of the SSH RSA key. The key length must be from 1024-2048 and MUST be divisable by 8. The default is 1024. 🛠 If you change the `ssh-keysize` and SSH keys already exist, you will be prompted to remove the existing keys using `clear ssh-key` and to reissue `ssh-keygen` to generate new SSH keys This command is used by `ssh-keygen`.

ssh-port

<table>
<tr><td><b>Command</b></td><td>ssh-port</td></tr>
<tr><td><b>Arguments</b></td><td>enable | disable | off</td></tr>
<tr><td><b>Device Type</b></td><td>All</td></tr>
<tr><td><b>Default</b></td><td>enable</td></tr>
<tr><td><b>Description</b></td><td>Enables or disables access to the SSH port (Port 22) via the wireless interface.
<table>
<tr><td>enable</td><td>Enable SSH access via the wireless and Ethernet ports.</td></tr>
<tr><td>disable</td><td>Disable SSH access via the wireless port. Access via Ethernet interface is enabled.</td></tr>
<tr><td>off</td><td>Disable SSH access via all network ports. SSH server is not loaded at restart.</td></tr>
</table>
Configuring <code>ssh-port off</code> is preferred to <code>ssh-port disable</code> for controlling the access to the SSH port.</td></tr>
</table>

ssh-trust

Command	ssh-trust
Arguments	0 | 1
Device Type	All
Default	0

Description

Configures the SSH Client on the module to automatically trust the MD5 finger print of any server to which a PASS connection is made. When enabled all MD5 fingerprints are accepted and stored in the SSH Trusted Host File.

0	Disabled and will not automatically trust MD5 finger prints from connected servers.
1	Enables automatic trusting of MD5 finger prints from connected servers.

This option should only be enabled for the initial connection between devices in a network.

The parameter defaults to `0` (disabled) and is not persistent across restarts or power cycles. This parameter is not saved with a `commit`.

Any trusted MD5 fingerprints must be saved by using a `commit`. Once committed they will be recognized during any subsequent connection to the trusted server.

startup-msg

Command	startup-msg
Arguments	0 | 1
Device Type	All
Default	0 (disable)
Description	Displays a start-up message, defined by startup-text, once the device server has completed a restart or power cycle.

`0`	Disables the start-up text. No message will be displayed after a restart or power cycle.
`1`	Enables the start-up text. The `startup-msg` text message will be displayed after a restart or power cycle.

Once the message is displayed the device server is available for interaction on the CLI interface.

startup-text

Command	startup-text
Arguments	[ASCII Text]
Device Type	All
Default	“Ready”
Description	ACSII Text message that is displayed when the device server has completed a restart or power cycle. Once displayed the device is available for interaction using CLI. The ASCII text message can be a maximum of 31 characters terminated by <CR>/<LF>. For the message to be displayed `startup-msg` must be enabled.

stats

Command	stats
Arguments	radio | ethernet
Device Type	All
Default	radio
Description	Displays statistics for the specified interface.

`radio`	Displays radio statistics.
`ethernet`	Displays wired Ethernet statistic. Only applies to Ethernet device.

Example:

```
stats radio

Rx Packets:                     7839
Rx Bytes:                       910915
Rx Errors:                      0
Rx Dropped:                     0
Rx Overruns:                    0
Tx Packets:                     202
Tx Bytes:                       16159
Tx Errors:                      0
Tx Dropped:                     0
Tx Overruns:                    0


stats ethernet

Rx Packets:                     16819
Rx Bytes:                       70915
Rx Errors:                      0
Rx Dropped:                     234
Rx Overruns:                    0
Tx Packets:                     17602
Tx Bytes:                       16159
Tx Errors:                      4
Tx Dropped:                     0
Tx Overruns:                    4
```

stop-bit

Command	stop-bit | stop-bit-p1
Arguments	1 | 2
Device Type	UART | Serial
Default	1
Description	Configures the number of stop bits to use on Serial port 1 (UART1). Use of the `–p1` suffix is optional.

stop-bit-p2

Command	stop-bit-p2
Arguments	1 | 2
Device Type	UART | Serial
Default	1
Description	Configures the number of stop bits to use on Serial port 2 (UART2).

subject-match

Command	subject-match
Arguments	[ASCII Text String]
Device Type	All
Default	[blank]
Description	Substring to be matched against the subject of the authentication server certificate. If this string is set, the server certificate is only accepted if it contains this string in the subject. The subject string is in following format: /C=US/ST=CA/L=San Francisco/CN=Test AS/emailAddress=as@example.com Example: EMAIL:server@example.com Example: DNS:server.example.com;DNS:server2.example.com Following types are supported: EMAIL, DNS, URI

subject-match2

Command	subject-match2
Arguments	[ASCII Text String]
Device Type	All
Default	[blank]
Description	Substring to be matched against the subject of the authentication server certificate. If this string is set, the server certificate is only accepted if it contains this string in the subject. The subject string is in following format: /C=US/ST=CA/L=San Francisco/CN=Test AS/emailAddress=as@example.com Example: EMAIL:server@example.com Example: DNS:server.example.com;DNS:server2.example.com Following types are supported: EMAIL, DNS, URI The string is used during the inner authentication phase.

sys-info

Command	sys-info
Arguments	[none]
Device Type	All
Default	[blank]

Description This command provides comprehensive version, disk and memory information for the module.

Example:

```
Firmware Version:                       1.30
Radio Firmware Version:                 5.0.21.p2-210.
Uboot Version:                          1.1.2
Kernel Version:                         2.6.31.12
Total RAMDisk Space:                    224256
RAMDisk Space Used:                     114688
Percent RAMDisk Space Used:             51%
RAMDisk Space Free:                     109568
FW Partition Total Disk Space:          0
FW Partition Disk Space Used:           0
FW Partition Percent Disk Space Used:   0%
FW Partition Disk Space Free:           0
Total Memory:                           14303232
Memory Used:                            12886016
Percent Memory Used:                    90%
Memory Free:                            1417216
Up Time (Sec):                          339235
```

telnet-echo

Command	telnet-echo
Arguments	disable | enable
Device Type	All
Default	enable
Description	Enables or disables whether characters are echoed back to their source during a telnet connection.

`disable`	Characters will not be echoed.
`enable`	Characters will be echoed.

telnet-port

Command	telnet-port
Arguments	disable | enable
Device Type	Ethernet
Default	enable

Description

Enables or disables access to the modules telnet port via the wireless interface.

This is similar to port filtering, when enabled the module will transfer all traffic on the port number defined by `wl-telent-port` to its internal IP stack, when disabled all traffic will on this port will be forwarded to the wired interface.

`disable`	The module will transfer all traffic on the port defined by `wl-telnet-port` to the wired Ethernet interface.
`enable`	The module will transfer all traffic on the port defined by `wl-telnet-port` to its internal IP stack.

Disabling the telnet-port will prevent any connections on the `wl-telnet-port` from being accepted by the module, limiting TCP/IP connection for CLI session to the wired interface only. This will restrict the management options available.

This can be overcome by establishing a port forwarding rule that redirects incoming wireless traffic directed to a defined port on the wireless interface to the gateway address of the module using the port defined by `wl-telnet-port`.

update

Command	update
Arguments	[blank] | ftp
Device Type	All
Default	[blank]

Description Used to update of the Airborne Device Server firmware. Supports firmware delivery by both FTP and Xmodem transfer.

Only firmware authorized by Quatech should be used with this command. Any attempt to use an alternative image will void the modules warranty.

FTP delivery requires a valid FTP server configuration to have been configured prior to the attempt to update the firmware.

`[blank]`	The module expects a Xmodem transfer to be initiated by a host on one of the available ports.
`ftp`	The module will use the configured FTP settings and attempt to download the firmware update image. The ftp-filename must match the firmware image being down loaded, e.g. `ftp-filename Veyron101.img`

CRITICAL: When updating any firmware, power must be maintained during the entire update process. Removal or interruption of the power supply may cause a corruption of the firmware update and cause the module to stop functioning. If this occurs please contact Quatech Technical Support.

update-uboot

Command	update-uboot
Arguments	xmodem | ftp
Device Type	All
Default	xmodem

Description

Updates the devices U-Boot firmware.

If `update-uboot` is issued without an argument the module will operate as if the `xmodem` argument had been used for the update.

> Only firmware authorized by Quatech should be used with this command. Any attempt to use an alternative image will void the modules warranty.

Requires configuration of the FTP client settings prior to being issued.

`xmodem`	The module expects an Xmodem or Xmodem-1K transfer to be initiated by a host on the connected ports. The file transfer must be the U-Boot update file from Quatech.
`ftp`	The module will use the configured FTP settings and attempt to download the U-Boot update image. The ftp-filename must match the firmware image being down loaded, e.g. `ftp-filename u-boot.ver01_01_02.img`

The device must be restarted or power cycled once the update process has completed.

> CRITICAL: When updating any firmware, power must be maintained during the entire update process. Removal or interruption of the power supply may cause a corruption of the firmware update and cause the module to stop functioning. If this occurs please contact Quatech Technical Support.

ver-fw

Command	ver-fw
Arguments	none
Device Type	All
Default	<none>
Description	Returns the current version of firmware loaded on the module.

ver-radio

Command	ver-radio
Arguments	none
Device Type	All
Default	<none>
Description	Returns the current version of radio firmware being run on the device servers' radio.

ver-uboot

Command	ver-uboot
Arguments	none
Device Type	All
Default	<none>
Description	Returns the version of uboot loader code resident on the device server.

wins-server1

Command	wins-server1
Arguments	[ASCII Text: IP Address]
Device Type	All
Default	0.0.0.0
Description	Configures the Primary WINS Server Address. This value is used for WINS lookups, if the lookup fails using the value from `dns-server1` or `dns-server2`. If the DHCP Client is enabled, the `wins-server1` value will be updated (if the DHCP Server provides one) during the DHCP cycle. Default is 0.0.0.0.

wins-server2

Command	wins-server2
Arguments	[ASCII Text: IP Address]
Device Type	All
Default	0.0.0.0
Description	Configures the Secondary WINS Server Address. This value is used for WINS lookups, if the lookup fails using the value from `dns-server1` or `dns-server2`. If the DHCP Client is enabled, the `wins-server2` value will be updated (if the DHCP Server provides one) during the DHCP cycle. Default is 0.0.0.0.

wln-ant

Command	wln-ant	
Arguments	1 | 2 | d	
Device Type	All	
Default	2	
Description	Determine the antenna settings for transmit and receive.	
	1	Selects CN3 for transmit and receive.
	2	Selects CN2 for transmit and receive.
	d	Allows the device to determine the most appropriate connector (CN2 or CN3) for transmit and receive.

wl-assoc-backoff

Command	wl-assoc-backoff
Arguments	[Integer] Range: 0 -20000
Device Type	All
Default	10000
Description	The amount of time in milliseconds to backoff, after the number of failed association attempts defined by the `wl-assoc-retries` command have been reached. Range 0 - 20000 milliseconds (0 to 20 seconds)

wl-assoc-retries

Command	wl-assoc-retries
Arguments	[Integer] Range: 0 - 32
Device Type	All
Default	3
Description	The number of times to try an association attempt before backing off. Range 0 - 32 (default 3)

wln-cfg-led

Command	wln-cfg-led
Arguments	enable | disable
Device Type	All
Default	enable
Description	Controls the function of the GPIO pin (F3) used for the LED_WLN_CFG, pin 26.

enable	Defines the output of GPIO pin F3 as the LED_WLN_CFG.
disable	Defines the GPIO pin F3 for use as a general purpose digital I/O pin.

The LED_CON must be disabled `for io-dir-f`, `io-pullup-f` and `io-write` to affect GPIO F3.

wl-dhcp-vendorid

Command	wl-dhcp-vendorid
Arguments	[ASCII Text]
Device Type	All
Default	Empty String
Description	Configures the DHCP Vendor Class ID String to use in the DHCP requests. Parameter can by up to 31 ASCII characters long.

wl-http-def

Command	wl-http-def
Arguments	[ASCII Text]
Device Type	All
Default	Index.html
Description	Configures the default home page URL for the internal web server.

wl-http-port

Command	wl-http-port
Arguments	[Integer] Range:
Device Type	All
Default	80
Description	Configures the TCP port number used by the HTTP (Web) server. Range: 0 – XXXXX (Default 80)

wl-mac

Command	wl-mac
Arguments	[ASCHEX: 6 Bytes]
Device Type	All
Default	<varies>
Description	Configures the MAC address of the wireless interface. The input is 6 bytes ASCHEX with no colons e.g. `000B280040AA`. The value specified by the argument temporarily overwrites the factory value. For the change to be made the value must be committed and the device server restarted. When a reset is issued or a hardware factory reset is applied the Ethernet interface factory MAC value is recovered. ⚡ Changing the MAC value must be done with caution. Only a known unique MAC value should be used.

wl-noise

Command	wl-noise
Arguments	[None]
Device Type	All
Default	<none>
Description	Displays the current Noise value (in dBm). If the module is not associated, it will display -99.

wl-retry-time

Command	wl-retry-time | wl-retry-time-p1
Arguments	[integer]
Device Type	Serial, UART
Default	60 <seconds>
Description	Configures the interval, in seconds, between attempts to establish a TCP connection with a Network Server. Used by Serial 1 (UART1) interface when the serial default mode is `pass`. The range for the parameters is 0 – 4,294,967,295 seconds (32 bit binary unsigned). Use of the `–p1` suffix is optional.

wl-retry-time-p2

Command	wl-retry-time-p2
Arguments	[integer]
Device Type	Serial, UART
Default	60 <seconds>
Description	Configures the interval, in seconds, between attempts to establish a TCP connection with a Network Server. Used by Serial 2 (UART2) interface when the serial default mode is `pass`. The range for the parameters is 0 – 4,294,967,295 seconds (32 bit binary unsigned).

wl-route

Command	wl-route
Arguments	[all | tcp | udp | icmp | bcast] [port <integer>] [forward | drop]] [xxx.xxx.xxx.xxx<port>]
Device Type	Ethernet
Default	[blank]

Description

Sets a specific rule for incoming Wireless traffic. This command allows port forwarding rules to be established for incoming wireless traffic. With the command an incoming port can be tied to a wired Ethernet client IP address, allowing network based devices the ability to access clients on the pirvate network.

`all|tcp|udp|icmp|bcast`	Selects the protocol for the rule.
`port <integer>`	Defines the port number for the rule. The port number must be an integer.
`forward|drop`	Defines if the rule forwards or blocks traffic.
`xxx.xxx.xxx.xxx:<port>`	Defines the private network address the port is mapped to. The `xxx.xxx.xxx.xxx` must represent a valid IP address where `xxx` is an integer between 0 and 255. The resultant IP address must not be 0.0.0.0. The `<port>` must be an integer.

The following provides details for the protocol and action parameters:

`all`	Allows all traffic to be affected by the rule.
`tcp`	The rule impacts only TCP/IP traffic.
`udp`	The rule impacts only UDP traffic.
`icmp`	The rule impacts only ICMP traffic.
`bcast`	The rule impacts only UDP traffic sent to the broadcast address (255.255.255.255). You cannot specify an IP address for the bcast protocol, and you must specify the relay action.

`forward`	This action will allow wireless traffic matching the identified port number to be forwarded to the IP address on the wired network.
`drop`	This action will stop traffic matching the identified port from being forwarded to the wired interface.
`relay`	This action will cause UDP broadcast traffic matching the rules conditions to be relayed to the wired interface. This this action is only applicable to the bcast protocol.

Multiple rules can be established to support the communication requirements. The rules set by the `wl-route` command take precedence over the `wl-route-default` setting.

It is required to establish multiple forwarding rules for the different services available to any device on the wired network, if both telnet (port 23) and http (port 80) are required, separate rule are required for forwarding to the different services.

By default all broadcast traffic on the wireless interface is dropped, regardless of the `wl-route-default` setting. To forward broadcast messages from the wireless to the Ethernet interface it is necessary to establish a broadcast forwarding rule with the required port number.

Here are some examples of rules:

`wl-route tcp port 1423 forward 192.168.2.100:80`	This will cause traffic sent to the device server on port 1423 to be forwarded to IP address 192.168.2.100 on port 80.
`wl-route tcp port 1424 forward 192.168.2.100:23`	This will cause traffic sent to the device server on port 1423 to be forwarded to IP address 192.168.2.100 on port 23.

The two rules above will forward http and telnet connections to the device holding the 192.168.2.100 IP address on the private (wired) network. Any device wanting to communicate to the service on the device would access them by using the public (wireless) IP address of the device server along with either port 1423 or 1424.

It is recommended that if port forwarding is to be used, all Ethernet devices on the private (wired) network use static IP addresses.

Entering the command with no parameters will display a list of the current port forwarding rules in the order they will be applied to incoming traffic.

wl-route-default

Command	wl-route-default
Arguments	[forward | drop]
Device Type	Ethernet
Default	[forward]

Description

Sets the default rule for incoming Wireless traffic. Allowing or denying access to the private (wired) network from the public (wireless) network. Through the rules established by this and the `wl-route` command, allowing access to the private network resources can be closely managed.

`forward`	All wireless traffic meant for the private (wired) network to be forwarded to the IP address defined by the `eth-ip` setting.
`drop`	Blocks all wireless traffic meant for the private (wired) network.

If the `wl-route-default` is set to drop and no additional rules (using `wl-route`) are added no traffic will be forwarded from the wireless to wired networks.

If the `wl-route-default` is set to forward and no additional rules are added, using the `wl-route` command, all wireless traffic will be forwarded to the IP address defined by the `eth-ip` setting. This will restrict access to a single IP address on the wired network.

wl-rssi

Command	wl-rssi
Arguments	[None]
Device Type	All
Default	<none>
Description	Displays the current Signal Strength value (in dBm). If the module is not associated, it will display -99.

wl-security

Command	wl-security
Arguments	disable | wep64 |wep228 | wpa-psk | wpa-leap | wpa-leap64 | wpa-leap228| wpa-psk64 | wpa-psk128 | wpa-psk128-tkip |wpa2-psk | wpa2-psk-tkip | tls | ttls | peap | wpa-fast | wpa2-fast | leap-wep |
Device Type	All
Default	disable
Description	Selects the Wireless Security method for Authentication and Encryption.

`disable`	Security is disabled. (default)
`wep64`	WEP, 64-bit key length (sometimes referred to as 40-bit WEP or WEP-40)
`wep228`	WEP, 128-bit key length (sometimes referred to as 104-bit WEP or WEP-104)
`wpa-psk`	WPA Pre-Shared Key
`wpa-leap`	WPA CISCO LEAP
`wpa-leap64`	Migration mode w/ Cipher suite TKIP+40-bit WEP using EAP (LEAP). Requires LEAP username and password.
`wpa-leap228`	Migration mode w/ Cipher suite TKIP+128-bit WEP using EAP (LEAP). Requires LEAP username and password.
`wpa-psk64`	Migration mode w/ Cipher suite TKIP+40-bit WEP using WPA PSK. Requires WPA Passphrase.
`wpa-psk128`	Migration mode w/ Cipher suite TKIP+128-bit WEP using WPA PSK. Requires WPA Passphrase.
`wpa-psk128-tkip`	Migration mode w/ Cipher suite TKIP and/or 128-bit WEP using WPA PSK. Requires WPA Passphrase.
`wpa2-psk`	WPA2 Pre-shared Key, also known as WPA2 Personal. Requires WPA Passphrase.
`Wpa2-psk-tkip`	WPA2 Pre-shared Key with Group Cipher suite TKIP, also known as WPA2 Personal. Requires WPA Passphrase.
`tls`	WPA/WPA2 with EAP-TLS authentication, also known as WPA-Enterprise (TKIP/AES) and WPA2-Enterprise TLS
`ttls`	WPA/WPA2 with EAP-TTLS authentication, also known as WPA-Enterprise (TKIP/AES) and WPA2-Enterprise TTLS
`peap`	WPA/WPA2 with PEAP authentication, also known as WPA-Enterprise (TKIP/AES) and WPA2-Enterprise PEAP v0
`wpa-fast`	EAP-FAST with Cipher suite TKIP.
`wpa2-fast`	EAP-FAST with Cipher suite EAS-CCMP.
`wep-leap`	LEAP with WEP Encryption.

wl-sleep-timer

Command	wl-sleep-timer | wl-sleep-timer-p1
Arguments	[integer]
Device Type	UART | Serial | SPI
Default	0
Description	Configures he inactivity time (in seconds) on Serial 1 (UART1) interface before the radio will transition to sleep mode. Data transfer to and from the UART will reset the timer. The timer has a range of 0 – 300 seconds. A value of zero (0) disables the `wl-sleep-timer`. Use of the –p1 suffix is optional.

wl-sleep-timer-p2

Command	wl-sleep-timer-p2
Arguments	[integer]
Device Type	UART | Serial | SPI
Default	0
Description	Configures he inactivity time (in seconds) on Serial 2 (UART2) interface before the radio will transition to sleep mode. Data transfer to and from the UART will reset the timer. The timer has a range of 0 – 300 seconds. A value of zero (0) disables the `wl-sleep-timer-p2`.

wl-specific-scan

Command	wl-specific-scan
Arguments	0 | 1
Device Type	All
Default	0
Description	Controls how the module scans for Access Points.

`0`	Use Broadcast Probes to attempt to find an Access Point.
`1`	Use Directed Probes to attempt to find an Access Point. In this mode only AP's with matching SSID's to the module will be probed.

Some network administrators disable responses to Broadcast Probes on the Access Point. To support scanning on these networks set `wl-specific-scan 1`.

wl-ssh-port

Command	wl-ssh-port
Arguments	<integer>
Device Type	All
Default	22
Description	Configures the TCP port number used by the SSH (Secure Shell) server.

wl-tcp-ip

Command	wl-tcp-ip | wl-tcp-ip-p1
Arguments	<IP Address: XXX.XXX.XXX.XXX>
Device Type	Serial | UART | SPI
Default	0.0.0.0
Description	Configures the primary network servers IP address for the Serial 1 (UART1) interface to use when the CLI session on the Serial 1 (UART1) interface initiates a TCP connection. The address is used when the `pass` or `serial-default pass` commands are used. If the IP address is empty or the connection is unsuccessful the CLI server will attempt a connection to the server IP address defined by `wl-tcp-ip2`. Use of the –p1 suffix is optional.

wl-tcp-ip2

Command	wl-tcp-ip2 | wl-tcp-ip2-p1
Arguments	<IP Address: XXX.XXX.XXX.XXX>
Device Type	Serial | UART | SPI
Default	0.0.0.0
Description	Configures the secondary network servers IP address for the Serial 1 (UART1) interface to use when the CLI session on the Serial 1 (UART1) interface initiates a TCP connection. This address is used when the `pass` or `serial-default pass` commands are used and either the primary IP address (`wl-tcp-ip`) is empty or the connection attempt to the primary IP address failed. Use of the –p1 suffix is optional.

wl-tcp-ip-p2

Command	wl-tcp-ip-p2
Arguments	<IP Address: XXX.XXX.XXX.XXX>
Device Type	Serial | UART | SPI
Default	0.0.0.0
Description	Configures the primary network servers IP address for the Serial 2 (UART2) interface to use when the CLI session on the Serial 1 (UART1) interface initiates a TCP connection. The address is used when the `pass` or `serial-default-p2 pass` commands are used. If the IP address is empty or the connection is unsuccessful the CLI server will attempt a connection to the server IP address defined by `wl-tcp-ip2-p2`.

wl-tcp-ip2-p2

Command	wl-tcp-ip2-p2
Arguments	<IP Address: XXX.XXX.XXX.XXX>
Device Type	Serial | UART | SPI
Default	0.0.0.0
Description	Configures the secondary network servers IP address for the Serial 2 (UART2) interface to use when the CLI session on the Serial 2 (UART2) interface initiates a TCP connection. This address is used when the `pass` or `serial-default-p2 pass` commands are used and either the primary IP address (`wl-tcp-ip-p2`) is empty or the connection attempt to the primary IP address failed.

wl-tcp-port

Command	wl-tcp-port | wl-tcp-port-p1
Arguments	<Integer >
Device Type	Serial | UART | SPI
Default	2571
Description	Configures the TCP port number for the Serial 1 (UART1) interface to use when the CLI session on the Serial 1 (UART1) interface initiates a TCP connection. The port is used with the network server IP address (`wl-tcp-ip`, `wl-tcp-ip2`) when the `pass` or `serial-default pass` commands are used. The port number must match the port the target network server is listening on for TCP/IP connections. The port number is used for both the primary and secondary target network server IP addresses, defined by `wl-tcp-ip` and `wl-tcp-ip2`. Use of the –p1 suffix is optional.

wl-tcp-port-p2

Command	wl-tcp-port-p2
Arguments	<Integer >
Device Type	Serial | UART | SPI
Default	2571
Description	Configures the TCP port number for the Serial 2 (UART2) interface to use when the CLI session on the Serial 2 (UART2) interface initiates a TCP connection. The port is used with the network server IP address (`wl-tcp-ip-p2`, `wl-tcp-ip2-p2`) when the `pass` or `serial-default-p2 pass` commands are used. The port number must match the port the target network server is listening on for TCP/IP connections. The port number is used for both the primary and secondary target network server IP addresses, defined by `wl-tcp-ip-p2` and `wl-tcp-ip2-p2`. The port range is `0 – 65535`.

wl-tcp-timeout

Command	wl-tcp-timeout | wl-tcp-timeout-p1
Arguments	<Integer >
Device Type	Serial | UART | SPI
Default	0 (disabled)
Description	Configures the inactivity timeout for the Serial 1 (UART1) interface to use when the CLI session on the Serial 1 (UART1) interface initiates a TCP connection. The timeout is applied when the `pass` or `serial-default pass` commands are used. Data to or from the UART interface will cause the timeout to reset. If the `pass` command was issued from the Serial 1 (UART1) interface and the timeout expires, the TCP connection is terminated and the data tunnel broken. The Serial 1 (UART1) interface is returned to the CLI command mode. A value of zero (0) disables the timeout, creating an infinite timeout. The range for the parameters is 0 – 4,294,967,295 seconds (32 bit binary unsigned). Use of the –p1 suffix is optional.

wl-tcp-timeout-p2

Command	wl-tcp-timeout-p2
Arguments	<Integer >
Device Type	Serial | UART | SPI
Default	0 (disabled)
Description	Configures the inactivity timeout for the Serial 2 (UART2) interface to use when the CLI session on the Serial 2 (UART2) interface initiates a TCP connection. The timeout is applied when the `pass` or `serial-default-p2 pass` commands are used. Data to or from the UART interface will cause the timeout to reset. If the `pass` command was issued from the Serial 2 (UART2) interface and the timeout expires, the TCP connection is terminated and the data tunnel broken. The Serial 2 (UART2) interface is returned to the CLI command mode. A value of zero (0) disables the timeout, creating an infinite timeout. The range for the parameters is 0 – 4,294,967,295 seconds (32 bit binary unsigned).

wl-telnet-port

Command	wl-telnet-port
Arguments	[Integer] Range:
Device Type	All
Default	23
Description	Configures the TCP port number used by the CLI server. Range: 0 – XXXXX (Default 80)

wl-tunnel

Command wl-tunnel | wl-tunnel-p1

Arguments 0 | 1

Device Type Serial | UART | SPI

Default 0

Description Enables or disables the tunnel port (`wl-tunnel-port`) assigned to the Serial 1 (UART1) interface, for communications.

The tunnel port does not require authentication using the CLI command (`auth <username> <password>`) and will automatically establish a data tunnel with the Serial 1 (UART1) interface only if it is in `listen` mode.

`0`	Disables the tunnel port.
`1`	Enables the tunnel port.

The tunnel can be enabled/disabled without needing a restart.

The Use of the –p1 suffix is optional.

> ⚡ Opening the tunnel port presents a potential security risk. Since no authentication is needed to establish a data connection, leaving the port enabled may allow unauthorized access to the host system.

wl-tunnel-p2

Command	wl-tunnel-p2
Arguments	0 | 1
Device Type	Serial | UART | SPI
Default	0

Description

Enables or disables the tunnel port (`wl-tunnel-port-p2`) assigned to the Serial 2 (UART2) interface, for communications.

The tunnel port does not require authentication using the CLI command (`auth <username> <password>`) and will automatically establish a data tunnel with the Serial 2 (UART2) interface only if it is in `listen` mode.

`0`	Disables the tunnel port.
`1`	Enables the tunnel port.

The tunnel can be enabled/disabled without needing a restart.

The Use of the –p1 suffix is optional.

> Opening the tunnel port presents a potential security risk. Since no authentication is needed to establish a data connection, leaving the port enabled may allow unauthorized access to the host system.

wl-tunnel-mode

Command	wl-tunnel-mode | wl-tunnel-mode-p1
Arguments	tcp | udp
Device Type	Serial | UART | SPI
Default	tcp
Description	Configures the communication protocol that will be used by the tunnel port (`wl-tunnel-port`) assigned to the Serial 1 (UART1) interface, for incoming communications.

`tcp`	Sets TCP/IP as the protocol on the tunnel port.
`udp`	Sets UDP as the protocol on the tunnel port.

The data tunnel must be enabled (`wl-tunnel 1`) for communications to be successful.

Non-matching protocols attempting to connect to the tunnel port will be ignored.

The use of the `-p1` suffix is optional.

wl-tunnel-mode-p2

Command	wl-tunnel-mode-p2
Arguments	tcp | udp
Device Type	Serial | UART | SPI
Default	tcp
Description	Configures the communication protocol that will be used by the tunnel port (`wl-tunnel-port-p2`) assigned to the Serial 2 (UART2) interface, for incoming communications.

`tcp`	Sets TCP/IP as the protocol on the tunnel port.
`udp`	Sets UDP as the protocol on the tunnel port.

The data tunnel must be enabled (`wl-tunnel-p2 1`)for communications to be successful.

Non-matching protocols attempting to connect to the tunnel port will be ignored.

wl-tunnel-port

Command	wl-tunnel-port | wl-tunnel-port-p1
Arguments	<Integer >
Device Type	Serial | UART | SPI
Default	8023
Description	Configures the tunnel port number for the Serial 1 (UART1) interface. The CLI server will process TCP/IP connection requests on this port as a request to open a CLI session in `pass` mode. The tunnel port does not require authentication using the CLI command (`auth <username> <password>`) and will automatically establish a data tunnel with the Serial 1 (UART1) interface only if it is in `listen` mode. The port range is `0 – 65535`. The use of the `–p1` suffix is optional.

wl-tunnel-port-p2

Command	wl-tunnel-port-p2
Arguments	<Integer >
Device Type	Serial | UART | SPI
Default	8024
Description	Configures the tunnel port number for the Serial 2 (UART2) interface. The CLI server will process TCP/IP connection requests on this port as a request to open a CLI session in `pass` mode. The tunnel port does not require authentication using the CLI command (`auth <username> <password>`) and will automatically establish a data tunnel with the Serial 2 (UART2) interface only if it is in `listen` mode. The port range is `0 – 65535`.

wl-udp-ip

Command	wl-udp-ip | wl-udp-ip-p1
Arguments	<IP Address: XXX.XXX.XXX.XXX>
Device Type	Serial | UART | SPI
Default	0.0.0.0
Description	Configures the network server IP address for the Serial 1 (UART1) interface to use when the CLI session on the Serial 1 (UART1) interface initiates UDP communications. The address is applied when the `pass` or `serial-default pass` commands are used. This address will be used when `wl-xmit-type udp` has been configured. This parameter does not require a commit and restart; it will be applied the next time `pass` is issued, after the address has been changed. Use of the –p1 suffix is optional.

wl-udp-ip-p2

Command	wl-udp-ip-p2
Arguments	<IP Address: XXX.XXX.XXX.XXX>
Device Type	Serial | UART | SPI
Default	0.0.0.0
Description	Configures the network server IP address for the Serial 2 (UART2) interface to use when the CLI session on the Serial 2 (UART2) interface initiates UDP communications. The address is applied when the `pass` or `serial-default-p2 pass` commands are used. This address will be used when `wl-xmit-type-p2 udp` has been configured. This parameter does not require a commit and restart; it will be applied the next time `pass` is issued, after the address has been changed.

wl-udp-ping

Command	wl-udp-ping
Arguments	0 | 1
Device Type	All
Default	0
Description	Periodically ping the configured UDP server. This causes the ARP cache to be periodically refreshed to prevent unnecessary ARPs from being transmitted. Since ARPs are broadcast and pings are unicast packets, total network overhead is reduced if pings are used instead of ARPs.

0	Disabled
1	Enabled

wl-udp-port

Command	wl-udp-port | wl-udp-port-p1
Arguments	<Integer >
Device Type	Serial | UART | SPI
Default	8023
Description	Configures the UDP port number for the Serial 1 (UART1) interface to use when the CLI session on the Serial 1 (UART1) interface initiates UDP transmissions. The port is used with the network server IP address (`wl-udp-ip`) when the `pass` or `serial-default pass` commands are used. For this setting to be used `wl-xmit-type udp` or `wl-xmit-type both` must be set. The port number must match the port the target network UDP server is listening on. The port range is `0 – 65535`. Use of the –p1 suffix is optional.

wl-udp-port-p2

Command	wl-udp-port-p2
Arguments	<Integer >
Device Type	Serial | UART | SPI
Default	8024
Description	Configures the UDP port number for the Serial 2 (UART2) interface to use when the CLI session on the Serial 2 (UART2) interface initiates UDP transmissions. The port is used with the network server IP address (`wl-udp-ip-p2`) when the `pass` or `serial-default pass` commands are used. For this setting to be used `wl-xmit-type-p2 udp` or `wl-xmit-type-p2 both` must be set. The port number must match the port the target network UDP server is listening on. The port range is `0 – 65535`.

wl-udp-rxport

Command	wl-udp-rxport | wl-udp-rxport-p1
Arguments	<Integer >
Device Type	Serial | UART | SPI
Default	8023
Description	Configures the UDP port number for the Serial 1 (UART1) tunnel will listen for UDP communications. The port will accept both unicast and broadcast packets and transfer their data payloads to the Serial 1 (UART1) interface. Data will only be transferred when a data tunnel has been established with Serial 1 (UART1) interface. The `pass` or `serial-default pass` commands, issued from the Serial 1 (UART1) interface are used to establish the data tunnel prior to receiving UDP transmissions. The port number must match the port the network UDP server is transmitting packets to. The port range is `0 – 65535`. Use of the –p1 suffix is optional.

wl-udp-rxport-p2

Command	wl-udp-rxport-p2
Arguments	<Integer >
Device Type	Serial | UART | SPI
Default	8024
Description	Configures the UDP port number for the Serial 2 (UART2) tunnel will listen for UDP communications. The port will accept both unicast and broadcast packets and transfer their data payloads to the Serial 2 (UART2) interface. Data will only be transferred when a data tunnel has been established with Serial 2 (UART2) interface. The `pass` or `serial-default-p2 pass` commands, issued from the Serial 2 (UART2) interface are used to establish the data tunnel prior to receiving UDP transmissions. The port number must match the port the network UDP server is transmitting packets to. The port range is `0 – 65535`.

wl-udp-xmit

Command	wl-udp-xmit | wl-udp-xmit-p1
Arguments	disable | ucast | bcast | both
Device Type	Serial | UART | SPI
Default	disable

Description

Configures the outbound UDP retransmission mode for a TCP/IP data tunnel connected to Serial 1 (UART1) interface. When enabled the device server will retransmit the data payload of a TCP/IP packet using a UDP packet, this parameter determines the UDP packet type to be retransmitted.

`disable`	Disables outbound packet retransmission. No additional UDPO transmissions are made.
`ucast`	Enables UDP unicast retransmission. A UDP Unicast packet is sent using the target address of the TCP/IP packet.
`bcast`	Enables UDP broadcast retransmission. A UDP broadcast packet is sent using the payload of the initial TCP/IP packet.
`both`	Enables both Unicast and Broadcast UDP retransmission.

If `wl-udp-xmit both` is set, three packets will be sent TCP/IP, UDP Unicast and UDP Broadcast.

Use of the `–p1` suffix is optional.

wl-udp-xmit-p2

Command	wl-udp-xmit-p2
Arguments	disable | ucast | bcast | both
Device Type	Serial | UART | SPI
Default	disable
Description	Configures the outbound UDP retransmission mode for a TCP/IP data tunnel connected to Serial 2 (UART2) interface. When enabled the device server will retransmit the data payload of a TCP/IP packet using a UDP packet, this parameter determines the UDP packet type to be retransmitted.

`disable`	Disables outbound packet retransmission. No additional UDPO transmissions are made.
`ucast`	Enables UDP unicast retransmission. A UDP Unicast packet is sent using the target address of the TCP/IP packet.
`bcast`	Enables UDP broadcast retransmission. A UDP broadcast packet is sent using the payload of the initial TCP/IP packet.
`both`	Enables both Unicast and Broadcast UDP retransmission.

If `wl-udp-xmit-p2 both` is set, three packets will be sent TCP/IP, UDP Unicast and UDP Broadcast.

wl-wins1

Command	wl-wins1
Arguments	[IP Address]
Device Type	All
Default	0.0.0.0
Description	This command has been deprecated see `wins-server1`. Configures the Primary WINS Server Address. This value is used for WINS lookups, if the lookup fails using the value from `wl-dns1` or `wl-dns2`. If the DHCP Client is enabled, the `wl-wins1` value will be updated (if the DHCP Server provides one) during the DHCP cycle. Default is 0.0.0.0.

wl-wins2

Command	wl-wins1
Arguments	[IP Address]
Device Type	All
Default	0.0.0.0
Description	This command has been deprecated see `wins-server2`. Configures the Secondary WINS Server Address. This value is used for WINS lookups, if the lookup fails using the value from `wl-dns1` or `wl-dns2`. If the DHCP Client is enabled, the `wl-wins1` value will be updated (if the DHCP Server provides one) during the DHCP cycle. Default is 0.0.0.0.

wl-wpa-proto

Command	wl-wpa-proto
Arguments	auto | wpa | rsn
Device Type	All
Default	auto
Description	Selects the preferred WPA protocol to be used during authentication. Selecting a specific protocol (WPA or RSN) aids in speeding roaming.

`auto`	Device negotiates the protocol to be used for WPA.
`wpa`	Uses WPA (TKIP) for the protocol.
`rsn`	Uses RSN (WPA2) for the protocol.

wl-xmit-type

Command	wl-xmit-type | wl-xmit-type-p1
Arguments	tcp | udp | ssh | both
Device Type	Serial | UART | SPI
Default	tcp

Description

Configures the outbound traffic transmission protocol for the Serial 1 (UART1) interface when a data tunnel has been established.

`tcp`	Only TCP/IP protocol is used for data transmission.
`udp`	Only UDP protocol is used for data transmission.
`ssh`	Only TCP/IP protocol traffic, encrypted within a Secure Shell (SSH) is allowed.
`both`	Both TCP and UDP protocols are used for data transmission. Two packets are sent

It is required that the data tunnel, TCP and UDP server configurations have been completed for any given transmission protocol to be used.

A data tunnel must exist on the Serial 1 (UART1) interface for transmissions to occur.

Use of the `-p1` suffix is optional.

wl-xmit-type-p2

Command	wl-xmit-type-p2
Arguments	tcp | udp | ssh | both
Device Type	Serial | UART | SPI
Default	tcp

Description

Configures the outbound traffic transmission protocol for the Serial 2 (UART2) interface when a data tunnel has been established.

`tcp`	Only TCP/IP protocol is used for data transmission.
`udp`	Only UDP protocol is used for data transmission.
`ssh`	Only TCP/IP protocol traffic, encrypted within a Secure Shell (SSH) is allowed.
`both`	Both TCP and UDP protocols are used for data transmission. Two packets are sent

It is required that the data tunnel, TCP and UDP server configurations have been completed for any given transmission protocol to be used.

A data tunnel must exist on the Serial 2 (UART2) interface for transmissions to occur.

20.0 Error Codes

When the Airborne Device Server firmware encounters an error during operation the connected interfaces will display one of the following error codes in Table 40. The identified code will aid in isolation of the cause of the error.

Table 40 - Error Codes

Error Code	Description
0xF800	An unknown error has occurred.
0xF801	Invalid parameter.
0xF802	Command not recognized.
0xF803	Operation timed out.
0xF804	Invalid character.
0xF805	Insufficient memory.
0xF806	Not authorized.
0xF807	Parameter length invalid.
0xF808	Command not implemented.
0xF809	File not found.
0xF80A	Invalid port.
0xF80B	Port busy.
0xF80C	Invalid user or password.
0xF80D	Timeout waiting for update file.
0xF80E	Update file error.
0xF80F	Update cancelled.
0xF810	Invalid XMODEM Packet Sequence.
0xF811	Processing another inquiry.
0xF812	Unable to connect to server.
0xF813	Command not allowed in script.
0xF814	Join failed
0xF815	Join in progress
0xF816	Port assigned to another service
0xF818	Socket Busy.
0xF819	Insufficient socket memory.
0xF81A	No IP route.
0xF81B	Socket not connected.
0xF81C	No TCP data.
0xF81D	DNS: Transaction Failed.
0xF81E	DNS: Hostname not found.
0xF81F	DNS: internal error.
0xF820	DNS: invalid hostname.
0xF821	DNS: Server not configured.
0xF823	Header Failure
0xF82D	Mixed use of Legacy Escape command and Newer Escape commands.
0xF82E	TCP outbound configuration invalid.
0xF832	SPI: read failed.
0xF833	SPI: write failed.
0xF834	SPI: dir failed.
0xF835	SPI: GPIO pin reserved for SPI.
0xF837	Invalid flow control type.
0xF838	File write error.
0xF839	Error applying configuration.
0xF83A	Error parsing command line options.
0xF83B	Missing ftp-server-address.

Error Code	Description
0xF83C	Missing ftp-user.
0xF83D	Missing ftp-password.
0xF841	Error opening serial device.
0xF842	Error allocating host memory.
0xF843	Unable to set up TCP server socket.
0xF844	Unable to set up UDP server socket.
0xF845	Unable to accept TCP connection.
0xF846	Error reading host data.
0xF847	Error writing host data.
0xF848	Error reading TCP data.
0xF849	Error writing TCP data.
0xF84A	Error reading UDP data.
0xF84B	Error writing UDP data.
0xF84C	Error updating firmware
0xF84D	Error generating SSH key.
0xF84E	SSH key already exists.
0xF84F	Error writing GPIO pin
0xF850	Error reading GPIO pin
0xF851	Error setting GPIO pin direction
0xF852	Host not trusted.
0xF853	Disconnected from server.
0xF854	Could not create temp file - disk may be full
0xF855	Missing ftp-filename.
0xF856	Error during FTP transfer.
0xF857	ftp-user or ftp-password incorrect.
0xF858	Cannot connect to FTP server.
0xF859	File not found on FTP server.
0xF85A	Ethernet port not enabled.
0xF85B	Ethernet DHCP Server and Client both enabled. Reverting to factory default.
0xF85C	DHCP and Wireless DHCP both enabled. Reverting to factory default.
0xF85D	wl-dhcp disabled and wl-ip not set. Reverting to factory default.
0xF85E	Cannot set led-mode to rssi without a radio. Reverting to factory default.
0xF85F	wl-dhcp disabled and wl-subnet not set. Reverting to factory default.
0xF860	eth-role router and eth-gateway or eth-subnet not set. Reverting to factory default.
0xF861	Personality change not supported for boxed products.
0xF862	Port not enabled in hardware capabilities.
0xF863	Disable of Debug Port not supported by current version of Uboot.
0xF864	eth-dhcp disabled and eth-ip not set. Reverting to factory default.
0xF865	eth-dhcp disabled and eth-subnet not set. Reverting to factory default.
0xF866	Must use "clear cfg-encrypt" to change this setting

Comments/Notes:

21.0 Change Log

The following table indicates all changes made to this document:

Version	Date	Section	Change Description	Author
1.0	04/16/2009	-	Preliminary Release.	ACR
1.1	06/30/2009	-	Multiple typographical corrections.	ACR
		5.3	Added section for SPI interface.	
		10.4	Added text and tables to support configuration of module using .pfx or .p22 private key formats.	
		15.0	**get-cfg** command: Corrected configuration file names.	
			blink-post-led command: Added description	
			Reordered commands to be alphabetical	
			Changed all instances of OEM_config.txt to oem_config.txt	
			alt-subject-match command: Added description	
			alt-subject-match2 command: Added description	
			eth-info command: Added description	
			list-cert command: Added description	
			update command: Added description	
		16.0	Added Error Codes	
1.2	09/22/2010	All	Major document overhaul all sections impacted. Supports v1.4 of Airborne Enterprise firmware.	ACR
1.3	02/03/2011	All	Typo corrections in multiple sections.	ACR
		19.0	**wl-ant** command: Added description	
			wl-dhcp-rel command: Added command	
			wl-dhcp-renew command: Added command	
			eth-udap command: Added command	
			wl-wpa-proto command: Added command	

QUATECH ® Inc.
5675 Hudson Industrial Parkway
Hudson, OH 44236
USA

Telephone: 330.655.9000
Toll Free (USA): 800.553.1170
Fax: 330.655.9010
E-mail Support : support@quatech.com
Web Site: www.quatech.com