Reference Manual

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

Revision 1.0

April 09

<Page Intentionally Left Blank>

Quatech Confidential

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: 714-899-7543 / wirelesssupport@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 including, WPA, WEP and LEAP.

The best security and advanced networking is no good if you cannot connect your device to the Airborne 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 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 device servers have been integrated and deployed into a wide range of applications and markets, including Medical, Telematics and Logistics.

Quatech's 3rd Generation Device Server extends the reputation of the family further by drawing on the lessons learned and adding the latest technologies. The Quatech Enterprise 802.11 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 modules 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 next to 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 shaded area next to 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 a 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 Airborne Device Server family of products. This document is an addendum to the Airborne CLI reference manual and describes the commands introduced 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 compromises 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 will include the following sections. Please refer to the appropriate section the required information.

3.1   Overview

In this section we will review the different device configurations and basic operation and functionality of the Airborne Device Servers and Bridges. 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 device server will be described and examples of use and configuration will be provided to highlight the best use of the module and 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 Airborne Enterprise 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 the use of WEP, WPA and WPA2 will be included. Outlines of the authentication methods supported (EAP) and the certificates 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 14.0 provide guidance on which devices support it.

At the time of writing, the CLI command list represents the v1.03 release of the WLRG-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

5.0   Overview

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

This document describes the CLI in full. Since different Airborne™ devices differ in functionality, there may be differences in the use of the CLI for particular devices. These differences are clearly identified as part of this document.

There are four primary module configurations supported by the Airborne Enterprise Device Server family, these are UART, Serial, SPI and Ethernet. Each device types will be described below. In some cases multiple interface option 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   Ethernet

The Ethernet interface 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

- Rx+, Rx-, Tx+ and Tx-

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 interface 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. The DPAC 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 interface using a TCP client such as Windows Telnet. The Module's CLI Server supports a Telnet connection with the following restrictions:

- Telnet option negotiation should be turned off.

- Telnet commands such as `DO`, `WONT`, and `DON,` must not be issued.

- Network Virtual Terminal codes are not supported.

- NUT 7-bit encoding does not allow 8–bit data transfers.

- The CLI Server's wireless 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 up to three (3) 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) = MFG

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 definition tables in the following chapter include a column labeled **Ln** that indicates the access level required to execute each command. 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 Airborne™ WLN 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 wireless 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 Airborne™ WLN Module to a user-specified TCP server and/or UDP server on the wireless 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 the Wireless Interface

When the CLI Session on the wireless interface 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 the serial interface is in LISTEN Mode.
- No other CLI Session on the wireless interface is in PASS Mode.

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 interface. When the CLI Session on the serial interface enters LISTEN Mode, the Airborne™ WLN Module passively waits for a data bridge to be established over the wireless interface. 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 wireless 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 wireless interface using the TCP port specified by `wl-telnet-port` always start in CLI Mode.

- CLI Sessions on the wireless 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.

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 string is a five (5)-character string configurable via the `escape` CLI command. When escape processing is configured to use the BREAK signal, the CLI Server will parse the data stream for the BREAK signal.

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. Basically, 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. Subsequence use of the `pass` CLI command will re-establish the bridge for that interface.

The CLI Server allows independent configuration of escaping processing for the serial and wireless interfaces. The serial interface escape processing is configurable using the CLI parameter `esc-mode-serial`. The wireless interface 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 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 TBD.

7.0   A Typical Development System

A typical evaluation system includes:

- A Serial Host: A computer connected to the serial port of the Airborne™ WLN Module.

- A LAN Host: A computer that communicates wirelessly with the Module through an Access Point (AP).

- An Access Point.

- An Airborne™ WLN Module.

8.0 Serial Device Server Use

In this section the base functionality of the Serial/UART device server will be described and examples of use and configuration will be provided to highlight the best use of the module and 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.

The Airborne Enterprise Serial Device server provides the ability to connect a raw serial data stream to a TCP/IP based network using 802.11 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) 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 Airborne™ WLN Module provides data bridging via the PASS and LISTEN Modes of the CLI Session. During data bridging, the raw payload of the 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 the outgoing TCP or UDP packet.

There are multiple ways to setup a data bridge using the Airborne™ WLN 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`, or from a UDP message on the `wl-udp-rxport`.

> Only one CLI session on the network (802.11) interface may be bridged with a CLI session on the serial interface.

8.1.1 Bridging from the Serial Interface

The CLI Session on the 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 Airborne™ WLN Module must be properly configured to connect to a server on the network that will accept the communications. The following examples illustrate how to configure the Module to initiate a connection to a TCP server:

Figure 1 - Bridging from the Serial Interface Manually Using the pass Command
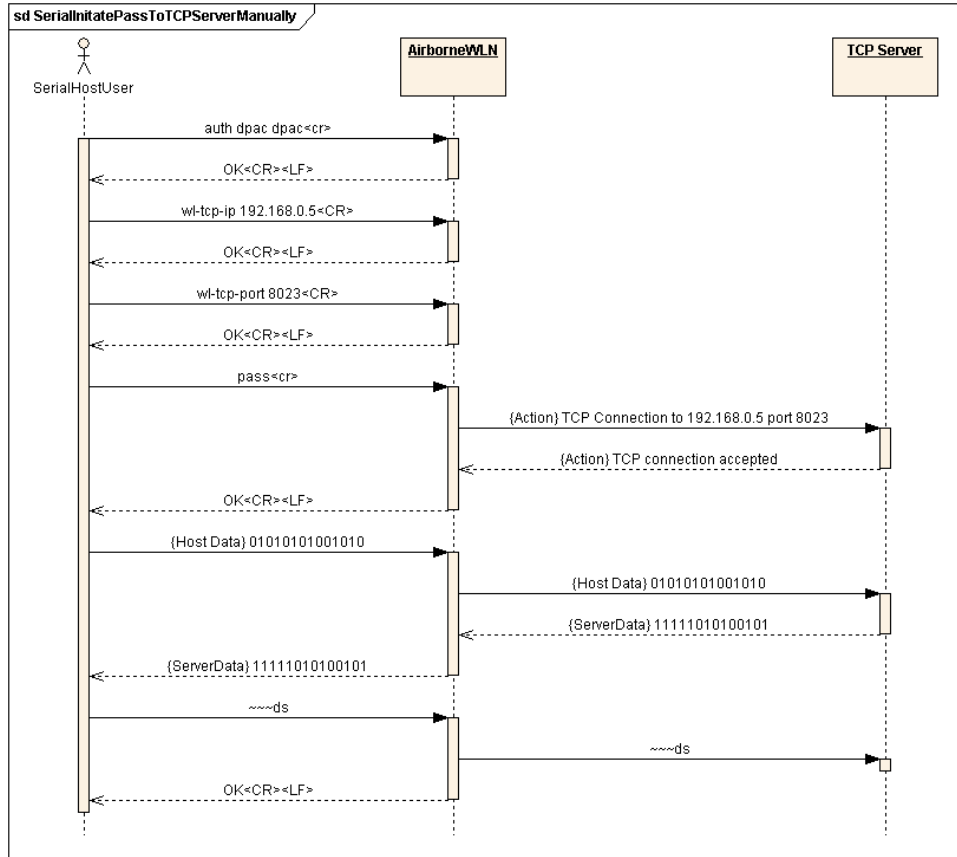
Figure 2 - Bridging from the Serial Interface Automatically at Startup Using the Serial-Default Command
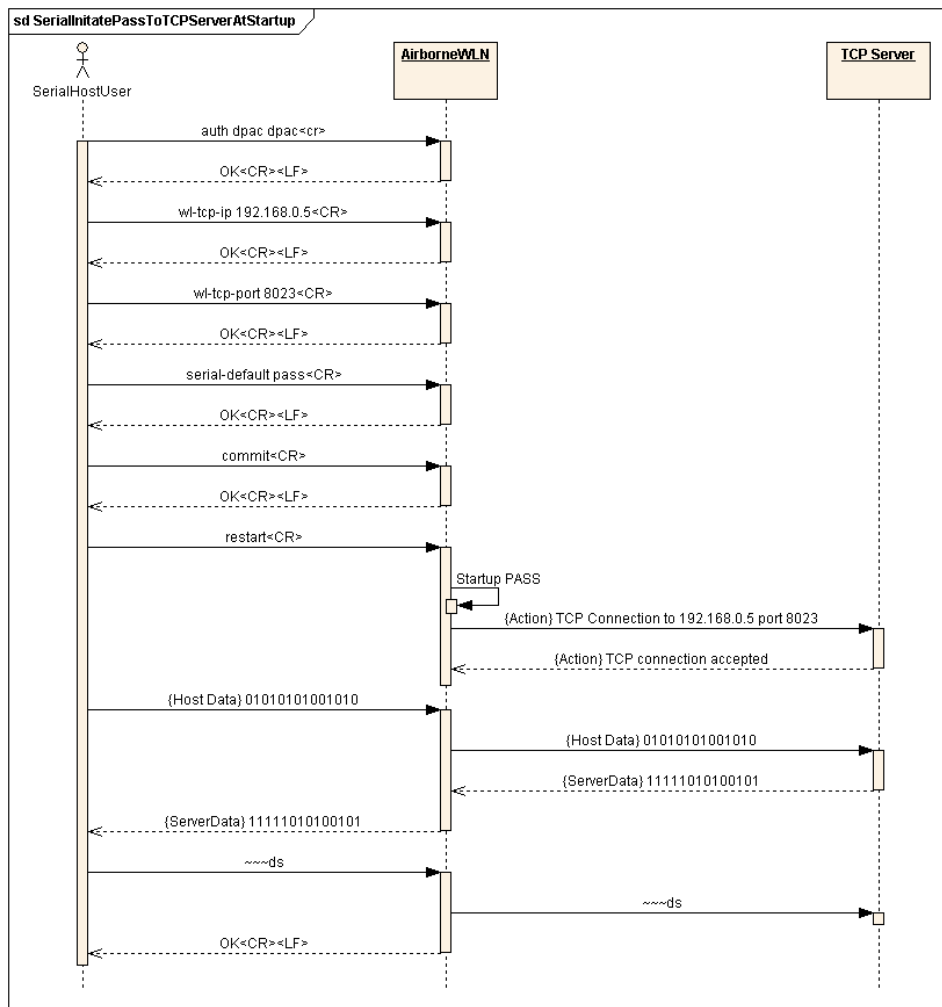
8.1.2   Bridging from a TCP connection on the wl-telnet-port

A user or OEM application connected over TCP to the `wl-telnet-port` of the Module may create a data bridge to the 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 the 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
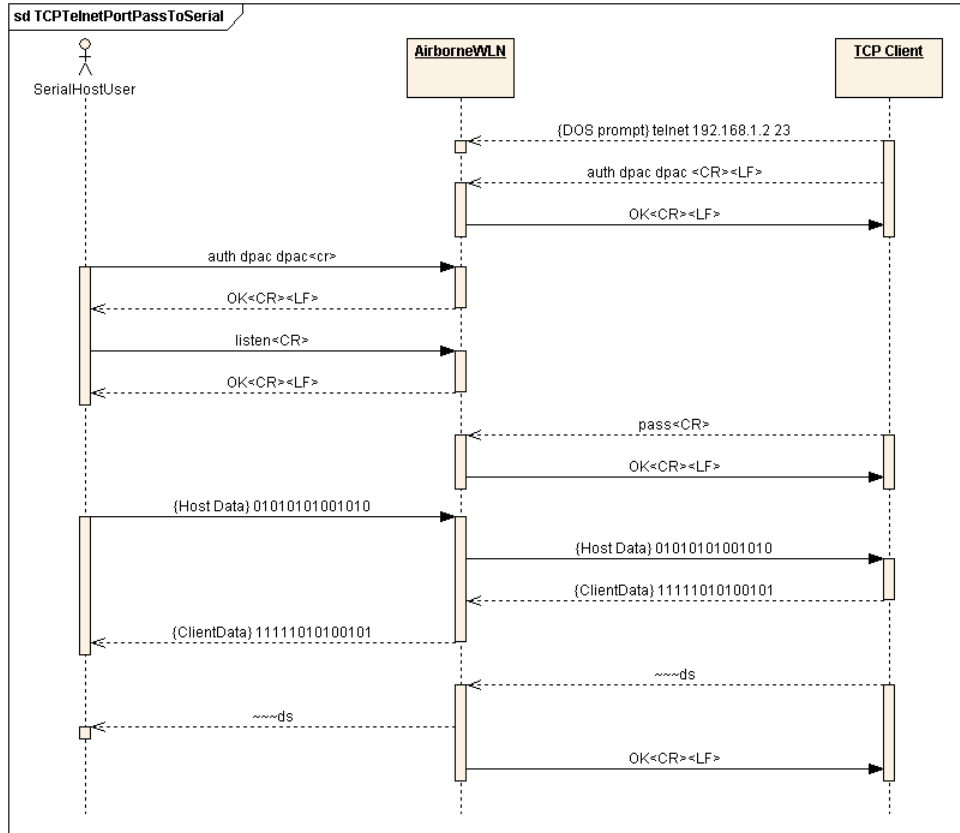
8.1.3 Bridging from a TCP connection on the wl-tunnel-port

The Module supports a tunneling feature that allows bridging between a specific TCP address/port and the Module's serial port. TCP port tunneling is supported by the `wl-tunnel`, `wl-tcp-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` or `udp`.

- `wl-tunnel-port` must be set to a non-zero value which is not the same as the Web Server port or the telnet port.

- The CLI Session on the serial interface must be in LISTEN Mode.

- There are no other CLI Sessions currently bridged.

If all of the previous conditions are met, this TCP connection will become the active bridge. All data payload will be bridged between the CLI Session on the 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 the serial interface remains in LISTEN Mode and escape detection is enabled if configured.

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
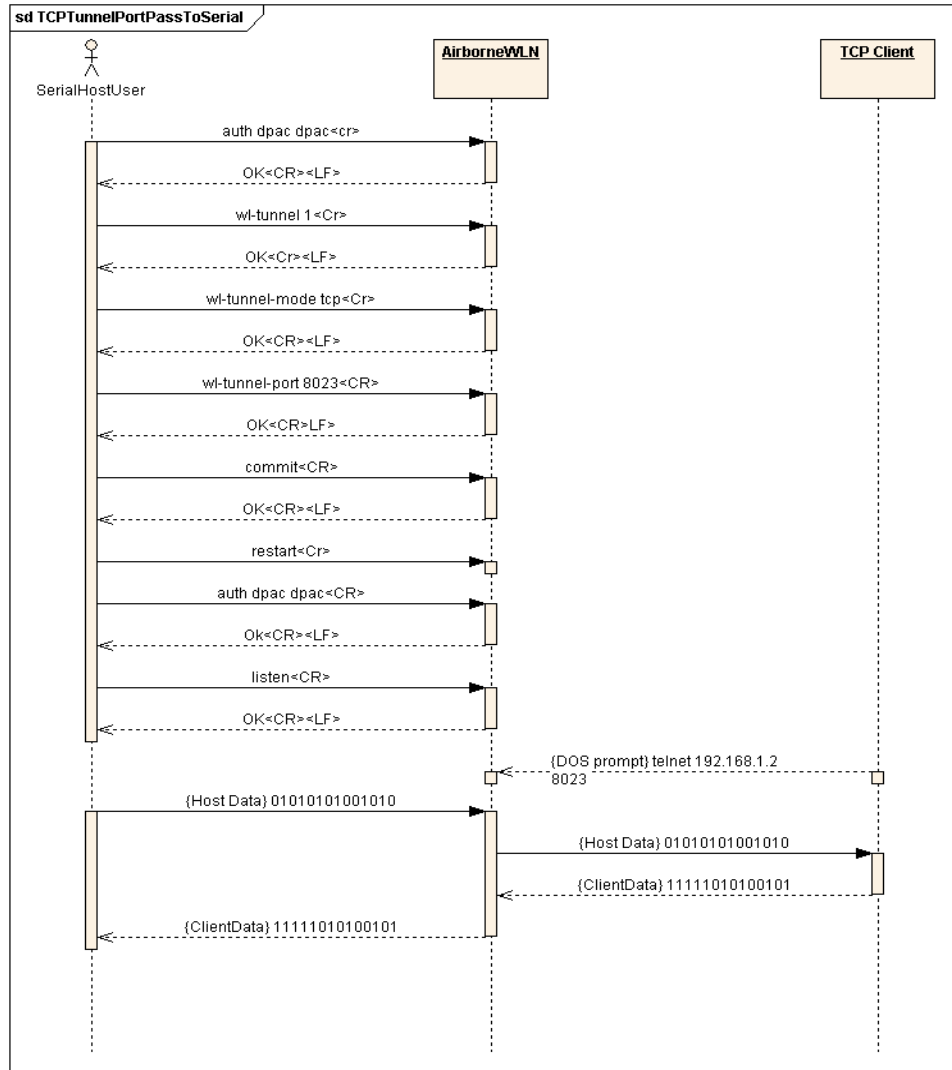
8.1.4 Bridging Using UDP

The Module supports UDP tunneling. This allows the Module to forward data from the 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 the 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` parameter or the `pass` 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 the 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 if forwarded to the 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.

9.0 Ethernet Bridge Use

The Airborne Ethernet Bridge 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 datagram 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 address of the devices on the private network (wired Ethernet interface).

The Airborne Bridge 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-Enterpise. It can acquire an IP address through both DHCP or user configured static IP. Configuration, association and authentication is handled entirely by the Airborne Bridge and requires no interaction from the wired host on the private network.

Figure 5 - Ethernet Bridge Functionality
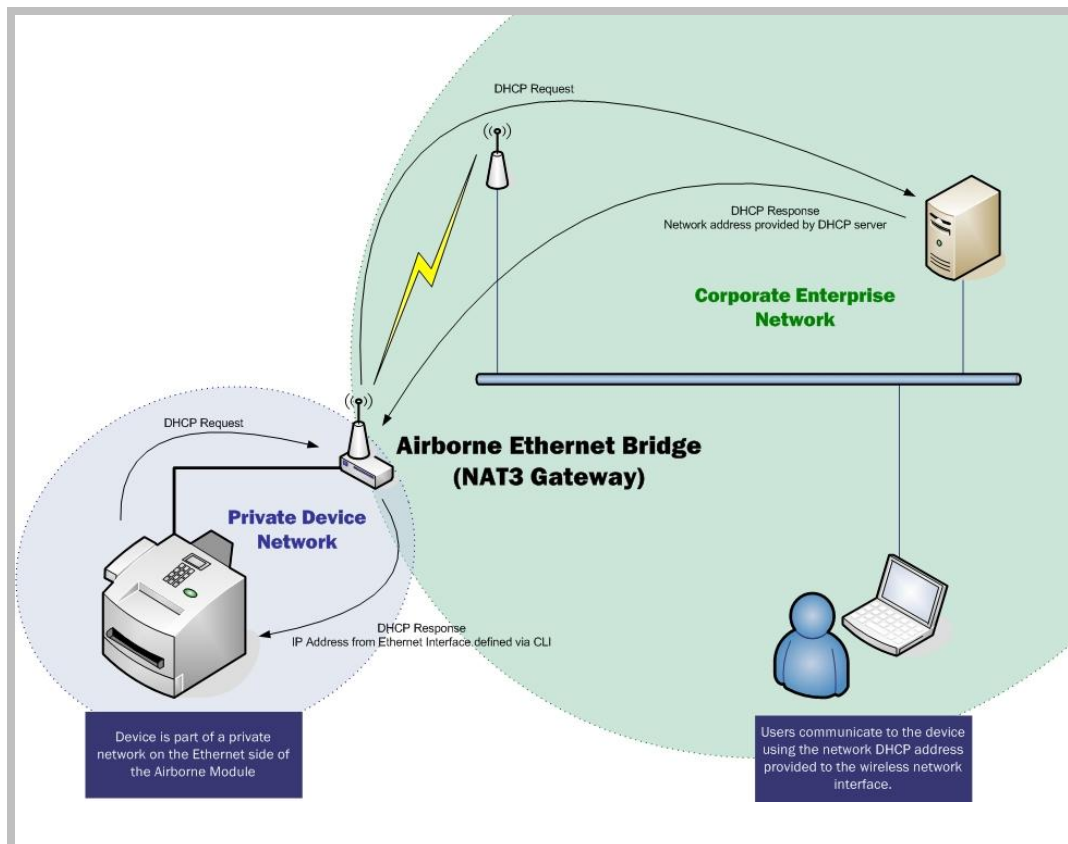
The Private network is the wired interface provided by the bridge. This interface includes a DHCP server and supports dynamic and static IP address assignment. This means any Ethernet client supporting DHCP can be connected to the wired interface without any configuration changes. The private network host can communicate with the Airborne Bridge using the bridges Ethernet IP address on the private network.

The Airborne Ethernet Bridge 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.

- A single point of authentication. The Airborne device 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 Airborne Bridge.

- 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 Airborne Bridge'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 1 - Public Network Configuration

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

Command	Description
`Security (various commands)`	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.
`http-port`	This parameter allows directed traffic on the http port 80 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.
`telnet-port`	This parameter allows directed traffic on the telnet port 23 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.

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

The network infrastructure will show the MAC and IP address of the Airborne Bridges 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
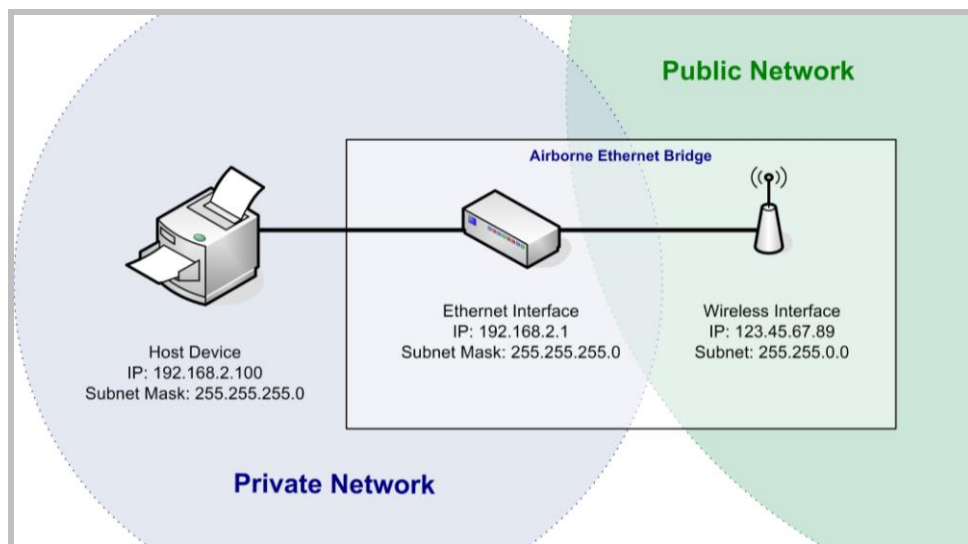
The public network interface supports the Airborne discovery protocol and will respond to discovery requests issued on the public network.

9.2   Private Network Interface

The private network interface is on the Ethernet port of the Airborne Bridge. The interface supports a single Ethernet client with either a static or DHCP sourced IP address. This interface needs minimal configuration and requires the parameters in Table 2 to be configured.

Table 2 - Private Network Interface Configuration

Command	Description
`eth-ip`	This is the IP address the DHCP server will lease to the client when the client is using DHCP.
`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.
`eth-mode`	The Ethernet interface supports the following configurations, this parameters determines the default mode of the interface. | `auto`	Auto negotiate
`10half`	10Mbps, half duplex
`10full`	10Mbps, full duplex
`100half`	100Mbps, half duplex
`100full`	100Mbps, full duplex | It is recommended that auto be used as this will provided the greatest level of compatibility on the Ethernet interface.

The private network supports the Airborne discovery protocol and will respond to discovery requests.

> 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.

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.
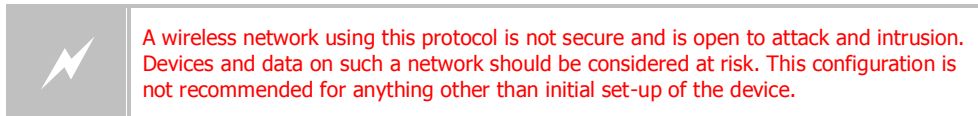
10.2 WEP Security

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 depreciated as a security method. Although organizations still utilize WEP, it is not a recommended as a 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 14.0:

Table 3 - 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.

Command	Description
`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.

10.2.1 WPA Migration Mode

Cisco infrastructure supports a migration mode that allows both legacy WEP and WPA client can coexist on the same network.

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 14.0. 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).

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

Table 4 - 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 AP. Must be 8-63 ASCII characters long.

Table 5 - 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.

Command	Description
`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, the Personal version utilizes the PSK as supported by WPA, the Enterprise supports a set of EAP (802.1x) protocols to provide the highest level of security available for 802.11 implementations.

WPA2-Enterprise, as defined by the WiFi Alliance, requires any product to support the following 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 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 precalculated hex keys as valid input, a description of the configuration requirements can be seen in Table 6 and Table 7.

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 8, Table 9 and Table 10) identify the typical requirements for implementing each type.

Table 6 - 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 AP. Must be 8-63 ASCII characters long.

Table 7 - 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.

Table 8 - 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 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-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

Table 9 - 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 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

Table 10 - 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 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.

Command	Description
`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.

It is important to know that there are many variations and additional configurations that the Airborne Device server 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 14.0.

10.5   Managing Certificates and Private Keys

Since certificates are required for most of the supported EAP protocols it will be necessary to upload these files to the Airborne Device Server before attempting to configure the device for WPA2-Enterprise security.

The Airborne Device Server 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 7.

The CLI commands that manage the delivery process are described in Table 11.

Table 11 - 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 the 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.

Command	Description

get-cert	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 1 and 254.
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 7 - Certificate and Private Key Delivery Methods
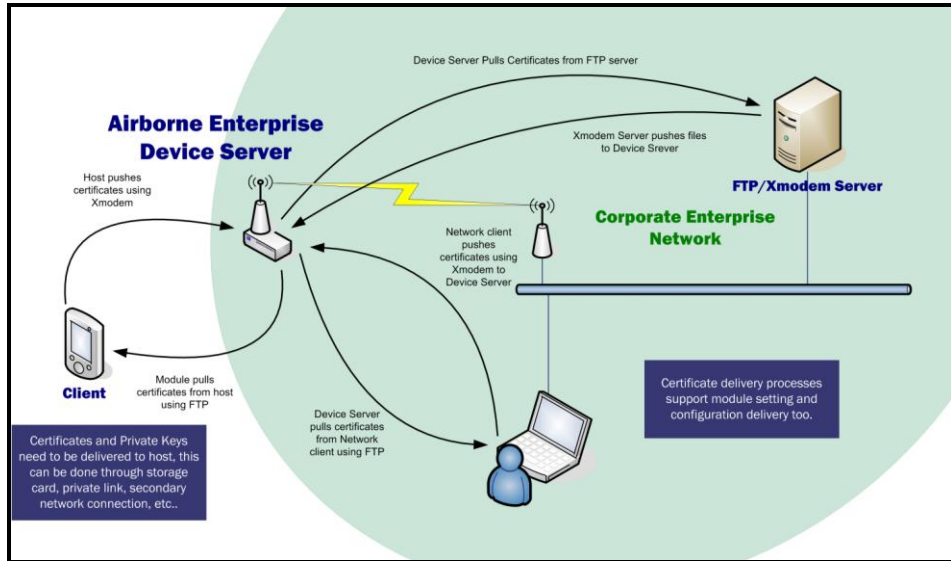
Control of the certificate and private key files is handled by separate group of commands these are described in Table 12.

Table 12 - 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. The command will list files that have been delivered but not saved.
`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:
	 `wl-security`
	 `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`

`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`

Command	Description
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 Airborne Enterprise Device Server 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 device server 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 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 13 - Commands that Affect Roaming

Command	Description
`wl-type`	This determines the network type being used by the device server, roaming applies to Infrastructure type only.
`wl-ssid`	This defines the Service Set Identifier or network name the device is to associate to.
`wl-rate`	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.
`wl-fixed-rate`	[needs confirmation] 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.
`wl-specific-scan`	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.
`wl-assoc-backoff`	The amount of time in milliseconds to back-off after three (3) failed association attempts. 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.
`wl-assoc-retries`	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.

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 open AP that responds quickest to a request to associate, this is typically the AP with the strongest signal strength. The use of the `any` SSID allows the Device Server to associate with any open AP that 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.

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.

12.0 FTP Configuration

The Airborne Enterprise Device Server family includes an FTP capability for delivery of 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 14 - 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 a value between 1 and 254.
`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 facilitate this function it is necessary to configure the internal FTP Client with the necessary information for the file upload, the commands in Table 14. 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 15 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 14.

Table 15 - 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.

13.0 Power Save

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 16 – Power-Save Modes

Command	Description
radio-on	Enables power to 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 power to 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	Set's the device server power management mode. Currently supports the modes described in Table 17.

Table 17 - 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.

14.0 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 a ? after authentication will provide a full list of the available CLI commands.

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-DP1XX
1	802.11b Airborne UART Module, LEAP Security	WLNB-AN-DP5XX
2	802.11b AirborneDirect Serial Module, WPA Security	WLNB-SE-DP1XX ABDB-SE-DP1XX
3	802.11b AirborneDirect Serial Module, LEAP Security	WLNB-SE-DP5XX ABDB-SE-DP5XX
4	802.11b AirborneDirect Ethernet Module, WPA Security	WLNB-ET-DP1XX ABDB-ET-DP1XX
5	802.11b AirborneDirect Ethernet Module, LEAP Security	WLNB-ET-DP5XX ABDB-ET-DP5XX
6	802.11b Airborne SPI Module, WPA Security	WLNB-AN-DP102
7	802.11b Airborne UART Module, LEAP Security	WLNB-AN-DP502
8	802.11b/g Airborne UART Module, LEAP Security	WLNG-AN-DP1XX
9	802.11b/g AirborneDirect Ethernet Module, LEAP Security	WLNG-ET-DP1XX ABDG-ET-DP1XX
10	802.11b/g AirborneDirect Serial Module, LEAP Security	WLNG-SE-DP1XX ABDG-SE-DP1XX
11	802.11b/g Airborne SPI Module, LEAP Security	WLNG-AN-DP102
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

wl-security

Command	wl-security
Arguments	disable	wep64 |wep128	wpa-psk	wpa-leap	wpa-leap64	wpa-leap128| wpa-psk64	wpa-psk128	wpa2-psk	tls	ttls	peap
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)
`wep128`	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-leap128`	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.
`wpa2-psk`	WPA2 Pre-shared Key, also known as WPA2 Personal.
`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

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_cfg.txt`	User configuration file. This file contains the user configuration commands and parameters.
`OEM_cfg.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.

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.

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.

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.

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.
	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.

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.
	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.

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-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.
	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-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 private key is an ASCII text string provided by the generator of the Private Key file.

? [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.

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.

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.

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-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-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-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
```

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

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:
```
wl-security
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
```

Clearing all security credentials from the device server may impact your ability to regain a wireless network connection..

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
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 device server may impact your ability to regain a wireless network connection..

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.

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.

default-cfg

Command	default-cfg
Arguments	none
Device Type	All
Default	[blank]
Description	This will set the module configuration to the factory default settings.

The settings in memory will be set to default values. You must commit the changes if you desire them to remain in the default state after a module restart.

> All user settings will be lost if you issue this command and commit the changes. This will potentially make the device server unable to connect to valid wireless network or communicate over the serial interface.
>
> Make sure that the factory default settings are known before issuing this command.

radio-on

Command	radio-on
Arguments	none
Device Type	All
Default	none
Description	Turns on power to the radio.
	The radio will attempt to regain a wireless network connection.

radio-off

Command	radio-off
Arguments	none
Device Type	All
Default	none
Description	Disables power to 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.

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:
	
	<pre>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</pre>
	or
	<pre>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</pre>

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
```

ftp-server-address

Command	ftp-server-address
Arguments	[Valid IP address]
Device Type	All
Default	<blank>
Description	This value defines the IP address 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

Must be specified in order for the following commands to function correctly:

```
update ftp
get-cert
get-cfg
```

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.

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-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
```

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.

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 telnet (port 23) traffic to its internal IP stack, when disabled all telnet traffic will be forwarded to the wired interface.

disable	The module will transfer all telnet (port 23) traffic to the wired Ethernet interface.
enable	The module will transfer all telnet (port 23) traffic to its internal IP stack.

Disabling the telnet-port will prevent any telnet (port 23) connections from being accepted by the module, limiting telnet connection for CLI session to the wired interface only. This will restrict the management options available.

http-port

Command	http-port
Arguments	disable | enable
Device Type	Ethernet
Default	enable
Description	Enables or disables access to the modules web browser (Port 80) via the wireless interface.

This is similar to port filtering, when enabled the module will transfer all HTTP traffic (port 80) traffic to its internal HTTP server, when disabled all HTTP traffic will be forwarded to the wired interface.

`disable`	The module will transfer all HTTP (port 80) traffic to the wired Ethernet interface.
`enable`	The module will transfer all HTTP (port 80) traffic to its internal IP stack.

> Disabling the http-port will prevent any web (port 80) connections from being accepted by the module, limiting web connections for web interface sessions to the wired interface only. This will restrict the management options available.

eth-ip

Command	eth-ip
Arguments	[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.
	The IP address of the client and the Ethernet gateway must be in the same subnet for IP routing to work correctly.

> 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-gateway

Command	eth-gateway
Arguments	[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.

> 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-subnet

Command	eth-subnet
Arguments	[Valid subnet mask]
Device Type	Ethernet
Default	255.255.255.0
Description	Configures the subnet mask for the Ethernet gateway and wired interface client.

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

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`.

apply-cfg

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

`serial`	Applies following serial port settings:
	
	`bit-rate`
	`parity`
	`flow`
	`data-bits`
	`stop-bit`
	`input-size`
	`intf-type`
	`serial-assert`
	
	This parameter only applies to the Serial and UART devices.

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

`ethernet`	Applies following Ethernet port settings:
	
	`eth-ip`
	`eth-gateway`
	`eth-subnet`
	`telnet-port`
	`http-port`
	
	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.

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 three (3) failed association attempts.
	Range 0 - 20000 milliseconds (0 to 20 seconds)

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.

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.

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 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.

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).

list-cfg

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

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, once generated.

> 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`.

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.

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.

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.

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-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-wins1

Command	wl-wins1
Arguments	[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 `wl-dns1` or `wl-dns2`. If the DHCP Client is enabled, the `wl-wins1` value will be updated (if the DHCP Server provides one).
	Default is 0.0.0.0.

wl-wins2

Command	wl-wins1
Arguments	[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 `wl-dns1` or `wl-dns2`. If the DHCP Client is enabled, the `wl-wins1` value will be updated (if the DHCP Server provides one). Default is 0.0.0.0.

cfg-dump

Command	cfg-dump
Arguments	[ASCII Text]
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).
`user`	Lists the saved user configuration (all parameters).
`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`.

Comments/Notes:

15.0 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