



Meru Wireless Networking Product

1.0 *Beta* Command-Line Interface Reference

July 2003

PRELIMINARY



Revision History

Revision Date	Revision	Description
July 2003	0.1	1.0 <i>Beta</i> release.

Information in this document is provided in connection with Meru® products. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document. Except as provided in Meru's Terms and Conditions of Sale for such products, Meru assumes no liability whatsoever, and Meru disclaims any express or implied warranty, relating to sale and/or use of Meru products including liability or warranties relating to fitness for a particular purpose, merchantability, or infringement of any patent, copyright or other intellectual property right. Meru products are not intended for use in medical, life saving, or life sustaining applications.

Meru may make changes to specifications and product descriptions at any time, without notice.

Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Meru reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

The product may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Meru Networks, Inc. might have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give any license to these patents.

This document as well as the described in it are furnished under license and may only be used or copied in accordance with the terms of the license. The information in this document is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Meru Networks, Inc.. Meru Networks, Inc. assumes no responsibility or liability for any errors or inaccuracies that may appear in this document or any software that may be provided in association with this document. Except as permitted by such license, no part of this document may be reproduced, stored in a retrieval system, or transmitted in any form or by any means without the express written consent of Meru Networks, Inc..

Contact your local Meru sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an ordering number and are referenced in this document, or other Meru literature may be obtained by calling or by visiting Meru's website at <http://www.merunetworks.com>

Copyright © Meru Networks, Inc., 2003. All rights reserved.

*Other names and brands may be claimed as the property of others.

This product includes software developed by parties other than Meru. **See the back page of this document for a list of copyrights and license agreements.**



Contents

	About This Document	vii
	Audience	vii
	In This Guide.	vii
	Other Sources of Information	viii
	Typographic Conventions	viii
	Syntax Notation	ix
	Contacting Meru	ix
	Web and Internet Sites	ix
	Customer Support Technicians	x
Chapter 1	Key Concepts	1
	Network Layout	1
	Node Identification	2
	Serial Numbers and Node Numbers	2
	Using the Controller Console and the CLI	2
Chapter 2	CLI Command Reference	1
	Overview	2
	Alarms Commands	3
	alarms	3
	asc: Subcontroller Commands	5
	asc all	5
	asc get	5
	asc ids	6
	asc set	7

ats: Access Point Commands8
ats all8
ats get8
ats ids9
ats images.	10
ats mappings	10
ats scripts	10
ats set	10
ats upgrade	11
auth: RADIUS Authentication Commands	12
auth all	12
auth del.	12
auth get.	12
auth new	13
auth set.	13
authstats get.	14
Channel BSSID Commands	15
channel all	15
channel get	15
channel ids	16
channel set	16
connect or remote: Remote Connection Commands.	17
{connect remote}	17
connect asc	17
connect ats	17
Console Commands	18
console get	18
console paging.	19
console set	19
db: Configuration Backup Commands	20
db backup	20
db delete	20
db list	20
db restore	21
Help Commands	22
help	22
History Commands	23
history	23
if...stats: Interface Statistics Commands	24
if80211stats all	24
if80211stats get	24
if80211stats ids	25
ifstats.	25
ifstats all	26
ifstats get	26
ifstats ids	27

if: Interface Commands	29
{interface ifc}	29
interface all	29
interface get	29
interface ids	30
interface set	31
qoS or codec: Quality of Service Commands	32
qoscodec all	32
qoscodec del	32
qoscodec get	32
qoscodec ids	33
qoscodec new	34
qosrule or ruleQoS: Rule Commands	36
qosrule all	36
qosrule del	36
qosrule get	36
qosrule ids	38
qosrule new	38
qosstats	39
qosvars	40
qosvars get	40
qosvars set	40
Quit Commands	42
quit	42
Reboot (or restart) Commands	43
{reboot restart}	43
reboot all	43
reboot asc	43
reboot ats	43
reboot wnc	44
security: RADIUS Security Commands	45
{security sec}	45
security get	45
security set	46
SNMP Commands	47
snmp del	47
snmp get	47
snmp new	48
snmp set	48
Station Commands	50
station all	50
station del	50
station get	50
station set	51
{stationstats ststats}	51
stationstats all	52
topo: Network Topology Commands	53

{topoascats ascats}	53
topoascats all	53
topoats	53
topoats all.	54
{topoatsats atsats}	54
topoatsats all	54
{topostaats staats}	55
topostaats all	55
topostation all	55
Watchdog Commands.	57
{watchdog wd}.	57
watchdog get	57
watchdog set	57
watchdog get	58
watchdog set	58
wirelessif or wif: Wireless Interface Commands	59
wirelessif all.	59
wirelessif get	59
wirelessif ids	61
wirelessif set	61
wnc: Controller Commands	63
wnc get	63
wnc set	64
Glossary	67
Alphabetic List of Terms and Abbreviations	67
Index	71



About This Document

This document describes the command-line interface (CLI) commands for the Meru Wireless Networking Product.

It briefly explains some of the concepts that you need to know before using the commands.

Audience

This guide is intended for network administrators who will install, configure, and maintain a Meru wireless network. It assumes that you are familiar with the following:

- Wireless networking
- Basic IP routing concepts

In This Guide

This guide includes the following chapters:

- **Chapter 1, “Key Concepts,”** which provides useful information about Meru wireless networking
- **Chapter 2, “CLI Command Reference,”** which lists alphabetically all of the commands in the Meru command-line interface
- **“Glossary”** which defines some terms used in this document

Other Sources of Information

This guide is part of the Meru wireless network documentation set, which also includes:


- *Meru Wireless Networking Product Command-Line Interface Reference (CLI Reference)*, which describes the Meru wireless network (software development kit), including the <<>>, which consists of the <<>>
- *Meru Wireless Networking Product System Administrator's Guide*, which provides <<>> using the Meru wireless network
- *Meru Wireless Networking Product Release Notes (Release Notes)*, which lists information about the latest software release
- *Meru wireless network for the controller board Installation and Quick Start Guide (Installation Guide)*, which describes how to install the Meru wireless network and set up the networking environment


In addition, the Meru Web site provides valuable information on products, support, and the company. See "Contacting Meru" on page ix.


Typographic Conventions

This document uses the following typographic conventions to help you locate and identify information:

<i>Italic text</i>	Used for new terms, emphasis, and book titles; also identifies arguments in syntax descriptions.
Bold text	Identifies keywords and punctuation in syntax descriptions.
Courier font	Identifies file names, folder names, and text that either appears on the screen or that you are required to type.

 **NOTE:** Provides extra information, tips, and hints regarding the topic.

 **CAUTION:** Identifies important information about actions that could result in damage to or loss of data or could cause the application to behave in unexpected ways.

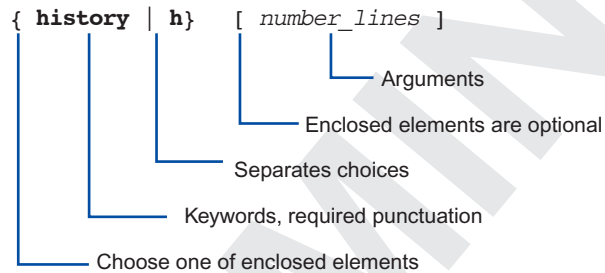
 **WARNING!** Identifies critical information about actions that could result in equipment failure or bodily injury.

Syntax Notation

Courier font is used for code. In syntax descriptions, **bold** indicates required keywords and a punctuation. In examples, **bold** highlights interesting parts. *Italics* indicate values that are to be replaced, such as arguments or file names.

bold	Required keywords and a punctuation.
<i>italic</i>	Arguments.
[]	Optional elements are enclosed by square brackets.
	Choices among elements are separated by vertical bars.
{ }	Required choice: Braces indicates that one of the enclosed elements must be used.
...	One or more of the preceding element is allowed.

The following figure shows a sample of syntax notation.



Contacting Meru

You can reach Meru's automated support services 24 hours a day, every day at no charge. The services contain the most up-to-date information about Meru products. You can access installation instructions, troubleshooting information, and general product information.

Web and Internet Sites

You can use the Internet to download software updates, troubleshooting tips, installation notes, and more.

For specific types of information and services, go to the following Web and Internet sites:

- **Corporate:** <http://www.merunetworks.com>
- **Wireless networking products:** <http://www.merunetworks.com/>
- **FTP host:** download.merunetworks.com
- **FTP directory:** [/support/network/](http://support/network/)

Contacting Meru

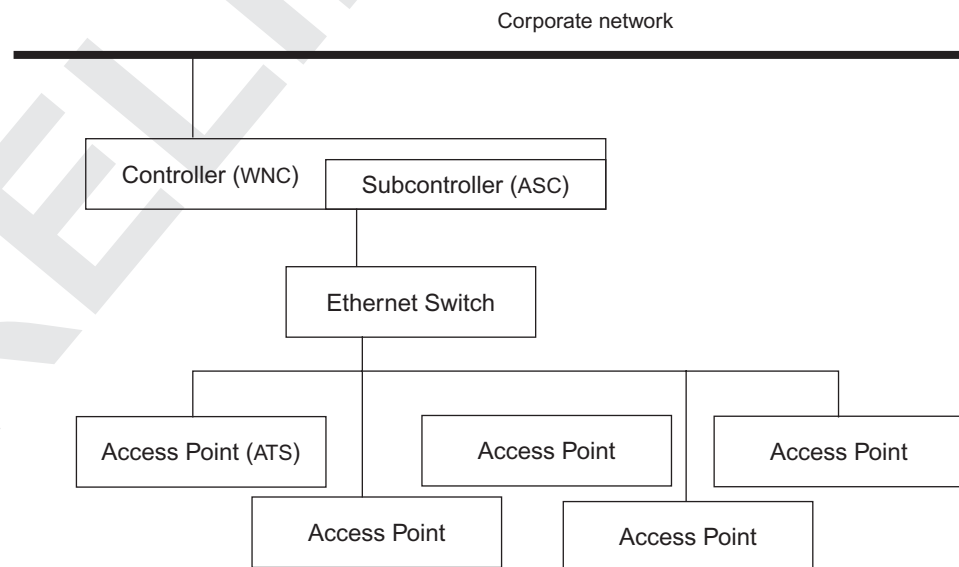
**Customer
Support
Technicians**

- **United States and Canada:** (7:00 - 17:00 M-F Pacific Time)

PRELIMINARY

Key Concepts

Network Layout



Node Identification

A *node* is a piece of equipment in a Meru wireless network. Each node is uniquely identified by two pieces of information:

- The **node type**: One of WNC (a controller), ASC (a subcontroller; contained within the controller), or ATS (an access point).
- The **node number**: An integer; unique within the node type.

Many commands require only a node number because the command is specific to a node type. Other commands require both the node type and the node number.

For example, if a network contains one controller (which always contains a subcontroller) and three access points, they are identified as follows:

Description	Node Type	Node Number
Controller	WNC	1
Subcontroller	ASC	1
First installed access point	ATS	1
Second installed access point	ATS	2
Third installed access point	ATS	3

If the first access point is removed from the network and a new one is installed, the new one becomes ATS 4, not ATS 1.

Serial Numbers and Node Numbers

When you connect an access point to your Meru wireless network, the controller automatically reads its serial number and assigns it a permanent node number. This node number acts as an alias for the serial number, so if an access point is unplugged and placed elsewhere in the network, the controller recognizes the access point as the same node number.

When you configure an access point, the configuration in the controller belongs to the node number, so the configuration can follow the access point.

Using the Controller Console and the CLI

You can connect to the controller using one of three methods:

- SSH
- Telnet
- Serial port

Your connection serves as the console for the controller. Logging in to the controller places you into the Meru command-line interface (CLI), which is similar to a command shell. The prompt for the CLI is

```
wnc>
```

Use the `console` command to adjust the appearance of the console display.

The CLI keeps a history of the commands typed during your current session. Use the `history` command to list these commands. For example:

```
wnc> history
1 help
2 console set columns=80 rows=60
3 history
```

Each history line is displayed with a number. You can redisplay any previous command for editing by typing an exclamation point (!) followed by the line number. For example:

```
wnc> !2
wnc> console set columns=80 rows=60
```

The cursor remains at the end of the redisplayed line so that you can edit the command. The CLI supports the following keystrokes to position the cursor for editing:

Keyboard key	
Home	Position cursor at the beginning of the command line.
End	Position cursor at the end of the command line.
Right arrow (-->)	Move the cursor to the right.
Left arrow (<--)	Move the cursor to the left.
Backspace, Delete, Del	Remove the character to the left of the cursor position.
Up arrow, down arrow	Scroll through the command history and allow editing on whichever line the cursor is positioned. This command becomes your current command.
ESC	Clears the command line.

For example:

```
wnc> history
1 help
2 console set columns=80 rows=60
3 history
4 console set columns=80 rows=20
```

Pressing the up arrow redisplay the preceding commands one at a time, in reverse order, on the command line. For example, if you scroll through the commands to the first command executed and then press return, it executes that command line, so your next command history is:

Using the Controller Console and the CLI

```
1 help
2 console set columns=80 rows=60
3 history
4 console set columns=80 rows=20
5 help
```

PRELIMINARY

CLI Command Reference

This chapter describes the syntax and operation of all CLI commands.

Commands are organized alphabetically within categories. The categories are

- "Alarms Commands" on page 3
- "asc: Subcontroller Commands" on page 5
- "ats: Access Point Commands" on page 8
- "auth: RADIUS Authentication Commands" on page 12
- "Channel BSSID Commands" on page 15
- "connect or remote: Remote Connection Commands" on page 17
- "Console Commands" on page 18
- "db: Configuration Backup Commands" on page 20
- "Help Commands" on page 22
- "History Commands" on page 23
- "if...stats: Interface Statistics Commands" on page 24
- "if: Interface Commands" on page 29
- "qoS or codec: Quality of Service Commands" on page 32
- "qosrule or ruleQoS: Rule Commands" on page 36
- "Quit Commands" on page 42
- "Reboot (or restart) Commands" on page 43
- "security: RADIUS Security Commands" on page 45

- “SNMP Commands” on page 47
- “Station Commands” on page 50
- “topo: Network Topology Commands” on page 53
- “Watchdog Commands” on page 57
- “wirelessif or wif: Wireless Interface Commands” on page 59
- “wnc: Controller Commands” on page 63

Overview

The CLI recognizes three classes of user. The ability to use commands in this chapter depends on the user’s class. Users can be:

User Class	Description
guest	Can use only those commands that display information.
admin	Can use most commands, including all the commands that <i>guest</i> can use.
support	Can use all commands.

Alarms Commands

These commands provide information about alarm or error status in the Meru wireless network.

alarms

Displays the controller's alarm log.

Syntax

alarms

Description

Displays the controller's alarm log file, showing the date and time of each event along with the originating node and the severity of the alarm. The log lists all alarms for all nodes in the Meru wireless network that have occurred since the controller's most recent reboot.

If there have been no alarms, the command displays

No entries.

Otherwise, the command lists the following information for each alarm:

Information	Description
Time	Date and time of the alarm in UTC (MM/DD hh:mm:ss), where: <ul style="list-style-type: none"> • MM = Month number (01 - 12). • DD = Day number. • hh = Hour (00 - 23). • mm = Minute. • ss = Second.
Node	Consists of two parts: <ul style="list-style-type: none"> • Node Type, one of ATS (an access point), ASC (a subcontroller), or WNC (a controller) • Node number within that type: An integer

Information	Description
Type	Alarm type. One of the following: <ul style="list-style-type: none"> • LINK UP • LINK DOWN • ASC DOWN • ATS DOWN • WATCHDOG FAILURE
Severity	The severity of the alarm, either CRITICAL or CLEAR (alarm state has been cleared)

Example

To view the alarm log:

```
wnc> alarms
```

This produces output similar to the following:

Time (UTC)	Node	Type	Severity
07/17 17:25:55	ATS 0001	LINK UP	CLEAR

asc: Subcontroller Commands

These commands manage aspects of subcontrollers.

asc all

Displays configuration information for all subcontrollers recognized by the controller. See `asc get` for details.

Syntax `asc all`

asc get

Displays configuration information for one or more subcontrollers.

Syntax `asc get node_id [...]`

Argument	Description
<code>node_id</code>	One or more node numbers identifying subcontrollers that are associated with this controller. Information is displayed for all the listed subcontrollers. NOTE: To list information for all subcontrollers, use <code>asc all</code> .

Description Displays the following configuration information for each of the specified subcontrollers:

Field	Description
Node ID	The unique numeric ID of the subcontroller.
Serial Number	Serial number of the subcontroller.
Description	A text description of this subcontroller.

Field	Description
Uptime	The uptime of the subcontroller, in hours, minutes, and seconds (hh:mm:ss).
Location	A text description of the location of this subcontroller.
Contact	Person or organization responsible for this subcontroller.
Operational State	The operational state of the subcontroller: <ul style="list-style-type: none"> • ENABLED: The subcontroller is operating correctly. • DISABLED: The subcontrollers is found by the controller but it is not operating correctly.
Availability Status	The availability of the subcontroller: <ul style="list-style-type: none"> • OFFLINE: The controller cannot find the subcontroller. • ONLINE: The controller can find the subcontroller.
Alarm State	The severity of the current alarm on the subcontroller. If more than one alarm is current, the highest severity is displayed. In order of increasing severity, the states are <ul style="list-style-type: none"> • NO ALARM: The subcontroller is not in an alarm state. • MINOR • MAJOR • CRITICAL.
Virtual MAC Address	The virtual MAC address of the wireless network to which this subcontroller belongs. This value is set with the wnc set command.
Software Version	The version of the software running on the subcontroller.

asc ids

Displays the node numbers for all known subcontrollers.

Syntax

```
asc ids
```

Description

Displays a list of node numbers of type ASC that are associated with this controller. Numbers are displayed one on each output line.

Example

asc set

Sets one or more of the configuration values for the specified subcontroller.

Syntax

```
asc set node_id
      [ desc=<val> ]
      [ location=<val> ]
      [ contact=<val> ]
      [ {wncdns=<val> | dns=<val>} ]
```

Argument	Description
node_id	The unique numeric ID of the subcontroller.
desc	A text description for this subcontroller
location	A text description of the location of this subcontroller.
contact	Text name for the person or organization responsible for this subcontroller.

Description

You must specify at least one configuration value.

ats: Access Point Commands

These commands manage aspects of access points.

ats all

Displays configuration information for all access point recognized by the controller. See `ats get` for details.

Syntax `ats all`

ats get

Displays configuration information for one or more access points.

Syntax `ats get node_id [...]`

Argument	Description
<code>node_id</code>	One or more node numbers identifying access points that are associated with this controller. Information is displayed for all the listed access point. NOTE: To list information for all access points, use <code>ats all</code> .

Description Displays the following configuration information for each of the specified access points:

Field	Description
Node ID	The unique numeric ID of the access point.
Serial Number	Serial number of the access point.
Description	A text description for this access point.

Field	Description
Uptime	The uptime of the access point, in hours, minutes, and seconds (hh:mm:ss).
Location	Text description of the location of this access point.
Contact	Person or organization responsible for this access point.
Operational State	The operational state of the access point: <ul style="list-style-type: none"> • ENABLED: The access point is operating correctly. • DISABLED: The access point is known by the controller but it is not operating correctly.
Availability Status	The availability of the access point: <ul style="list-style-type: none"> • OFFLINE: The controller cannot find the access point. • ONLINE: The controller can find the access point.
Alarm State	The severity of the current alarm on the access point. If more than one alarm is current, the highest severity is displayed. In order of increasing severity, the states are NO ALARM, MINOR, MAJOR, or CRITICAL.
Bound to ASC	The subcontroller node ID to which the access point is bound.
Security Mode	The security mode that the access point is in; either OPEN or 802.1x.
Privacy Bit	The privacy state that the access point is in; one of ON, OFF, or AUTO.
Boot Script	The script to run when the access point boots.
Virtual MAC Address	The virtual MAC address of the access point.
Boot Image Version	The version of the ROM boot image on the access point.
Runtime Image Version	The version of the runtime image on the access point.
FPGA Version	The version of the FPGA chip on the access point.

ats ids

Displays the node numbers for all known access points.

Syntax

```
ats ids
```

ats: Access Point Commands

Description Displays a list of node numbers of type ATS that are associated with this controller. Numbers are displayed one on each output line.

ats images

Lists access point upgrade image versions that have been copied to the controller.

ats mappings

Displays access point comm node and nms node ID mappings.

ats scripts

Lists available access point boot scripts.

See `wnc set` for information on how to assign a default script for all access points.
See `ats set` for how to assign a script to a specific access point.

ats set

Sets one or more of the configuration values for the specified access point.

Syntax

```
ats set <node_id>
  [ desc=text_desc ]
  [ location=text_loc ]
  [ contact=text_contact ]
  [ bootscript=script_name ]
```


Argument	Description
desc	Text description for this access point.
location	Text description of the location of this access point.
contact	Person or organization responsible for this access point.
bootscript	The name of the script to run when the access point boots. Use <code>ats scripts</code> to display the names of valid scripts. See <code>wnc set</code> for additional information.

ats upgrade

Installs upgraded software onto the ATS.

Syntax

```
ats upgrade {<node-id> | all}
           <version>
           [noreboot | <delay>]
```

To upgrade the software on an ATS, enter its node id and the version that was previously downloaded onto the WNC. Available images may be listed using the 'ats images' command. After the upgrade the ATS will reboot immediately when the upgrade is complete, unless the 'noreboot' option or a delay (in seconds) is given as the last argument.

auth: RADIUS Authentication Commands

These commands allow a user to manage RADIUS authentication information (authentication secret and RADIUS IP address).

auth all

Displays configuration information for all RADIUS authentications. See `auth get` for details.

Syntax `auth all`

auth del

Deletes the authentications specified by the authentication IDs.

Syntax `auth del <auth_id> [...]`

auth get

Displays configuration information for one or more authentications.

Syntax `auth get <auth_id> [...]`

Argument	Description
auth_id	One or more numbers identifying authentications that are associated with this controller. Information is displayed for all the listed items. NOTE: To list information for all subcontrollers, use <code>auth all</code> .

Description Displays the following fields for each of the authentications requested:

Field	Description
Authentication ID	A unique alphanumeric ID of the RADIUS server information.
RADIUS Server Secret	The string to hold the RADIUS secret key.
RADIUS Server IP Address	IP address of the RADIUS server in nnn.nnn.nnn.nnn format.

auth new

Create a new authentication.

Syntax

```
auth new <auth_id>
    [ {radiussecret=<val> | secret=<val>} ]
    [ {radiusip=<val> | ip=<val>} ]
```

Argument	Description
{radiussecret=<val> secret=<val>}	The string to hold the RADIUS secret key.
{radiusip=<val> ip=<val>}	IP address of the RADIUS server in nnn.nnn.nnn.nnn format.

auth set

Sets one or more of the configuration values for the specified authentication.

Syntax

```
auth set <auth_id>
    [ {radiussecret=<val> | secret=<val>} ]
    [ {radiusip=<val> | ip=<val>} ]
```

Argument	Description
{radiussecret=<val> secret=<val>}	The string to hold the RADIUS secret key.
{radiusip=<val> ip=<val>}	IP address of the RADIUS server in nnn.nnn.nnn.nnn format.

authstats get

Display authentication statistics.

Syntax

```
authstats get
```

Description

Displays information about Web and 802.1x authentication requests.

Displays the number of web and 802.1x authentication requests. The number of successes and failures is less than or equal to the number of requests for each originating method. The station count is the number of stations that are currently authorized by the indicated method.

802.1x Authorization Request Count.

802.1x Authorization Success Count.

802.1x Authorization Failure Count.

802.1x Authorization Station Count.

WWW Authorization Request Count.

WWW Authorization Success Count.

WWW Authorization Failure Count.

WWW Authorization Station Count.

Channel BSSID Commands

These commands allow a user to manage channel BSSID assignments.

channel all

Displays configuration information for all channels. See `channel get` for details.

Syntax `channel all`

channel get

Displays configuration information for one or more channels.

Syntax `channel get <channel_number> [...]`

Argument	Description
channel_number	One or more numbers identifying channels that are associated with this controller. Information is displayed for all the listed items. NOTE: To list information for all channels, use <code>channel all</code> .

Description Displays the following fields for each of the channels requested:

Field	Description
Channel Number:	The unique numeric channel number in the channel table.
BSSID:	The BSS (Basic Service Set) identifier assigned to the channel.
Active:	Indicates whether the channel is active ('on') or inactive ('off').

channel ids

Displays the channel numbers for all known channels.

Syntax `channel ids`

Description Displays a list of channel numbers. Numbers are displayed one on each output line.

channel set

Sets one or more of the configuration values for the specified channel.

Syntax

```
channel set <channel_number>
  [ bssid=<val> ]
  [ active=<val> ]
```

Argument	Description
bssid=<val>	The BSS (Basic Service Set) identifier assigned to the channel.
active=<val>	Indicates whether the channel is active ('on') or inactive ('off').

connect or remote: Remote Connection Commands

{connect | remote}

Connects to a remote ats or asc node. Use 'exit' or 'quit' to disconnect.

Syntax {connect | remote}

connect asc

Connects to remote asc node <node-id>.

Syntax connect asc <node-id>

connect ats

Syntax connect ats <node-id>

Connects to remote ats node <node-id>.

Console Commands

These commands manage display format and wrapping.

For additional information about using the console, see “Using the Controller Console and the CLI” on page 2.

console get

Displays format and scrolling configuration information for the console.

Syntax `{console | cons} get`

Description Displays the following fields:

Field	Description
Columns	The column width of the display. This should be set to match the actual screen width to insure proper word wrap on many of the output from many of the commands. A 0 (zero) sets width to an arbitrarily large value. The default is 80.
Rows	The row height of the display. This should be set to match the actual screen height to insure proper operation of 'more' processing of many lines of text. A 0 (zero) set the height to an arbitrarily large value. The default is 24.
Radix (base 10)	The radix or base in which numeric values should be displayed. The output of some commands, but not all, is affected by this. Also note that input radix is also not affected by this. The default is 10
Output Style	The style (or type of consumer) in which output should be presented. Valid styles are 'c', 'h', or 's' where 'c' is the default 'console' style intended for human readers, 'h' indicates HTML tags will be output, and 's' indicates a style that could be fed back into the command line interface. The default is C.

console paging

Enable (on) or disable (off) screen output paging.

Syntax `{console | cons} paging {on | off}`

console set

Sets one or more of the configuration values for the controller's console.

Syntax `{console | cons} set {columns=<val> | cols=<val>}
 {rows=<val> | lines=<val>}
 {radix=<val> | base=<val>}
 {output=<val> | style=<val>}`

Argument	Description
{columns cols}	The column width of the display. This should be set to match the actual screen width to insure proper word wrap on many of the output from many of the commands. A 0 (zero) sets width to an arbitrarily large value.
{rows lines}	The row height of the display. This should be set to match the actual screen height to insure proper operation of 'more' processing of many lines of text. A 0 (zero) set the height to an arbitrarily large value.
{radix base}	The radix or base in which numeric values should be displayed. The output of some commands, but not all, is affected by this. Also note that input radix is also not affected by this.
{output style}	The style (or type of consumer) in which output should be presented. Valid styles are 'c', 'h', or 's' where 'c' is the default 'console' style intended for human readers, 'h' indicates HTML tags will be output, and 's' indicates a style that could be fed back into the command line interface.

Description

These values remain in effect only during the current logged-in session. They are reset to defaults when you log out (exit the CLI; see the `quit` command).

db: Configuration Backup Commands

These commands allow a user to perform backup and restore of the system's configuration. Backups are made to local flash files which may be listed and deleted.

db backup

Syntax `db {backup | b} [<filename>]`

Backup current configuration database to the default location or to the given filename.

db delete

Syntax `db {delete | d} <filename>`

Delete the given backup file.

db list

Syntax `db {list | l}`

List the available database backup files.

db restore

Syntax `db {restore | r} [<filename>]`

Restore the configuration database from the default location or from the given filename.

PRELIMINARY

Help Commands

help

Syntax `{help | ?} [all | <category>]`

If no arguments are given, then this command prints one line summaries of command categories.

If one or more <category> arguments are given, then it prints out detailed descriptions of those categories.

If 'all' is supplied as the first argument then it prints out a detailed description of all command categories.

History Commands

history

Syntax `{history | h} [<number-lines>]`

If no arguments are given, then this command prints all the command lines that have been entered since the start of the session. If the argument `number-lines` is provided then the last `number-lines` from the history are printed.

if...stats: Interface Statistics Commands

These commands display the 802.11 interface statistics in the system. To get statistics on a single 802.11 interface, use the 'get' command and enter the node ID.

if80211stats all

Displays configuration information for all wireless interfaces recognized by the controller. See `if80211stats get` for details.

Syntax `if80211stats all`

if80211stats get

Displays statistics for the wireless interfaces for one or more access points.

Syntax `if80211stats get <node_id> [...]`

Argument	Description
node_id	One or more numbers identifying access points that are associated with this controller. Information is displayed for all the listed items. NOTE: To list information for all access points, use <code>if80211stats all</code> .

The following information is displayed for each node or interface:

Field	Description
Node ID	
TX Fragment Count	
MCast TX Frame Count	

Field	Description
Failed Count	
IF Retry Count	
Multiple Retry Count	
Frame Duplicate Count	
RTS Success Count	
RTS Failure Count	
ACK Failure Count	
RCV Fragment Count	
MCast RCV Frame Count	
FCS Frame Count	
TX Frame Count	
WEP Undecryptable Count	

if80211stats ids

Displays the interface numbers for the wireless interfaces on all access points known to this controller.

Syntax `if80211stats ids`

Description Displays a list of node numbers of type ATS that are associated with this controller. Numbers are displayed one on each output line.

ifstats

Syntax `ifstats`

The interface commands allow a user to display the statistics on the system's Ethernet interfaces. To display statistics on a single interface, the user must enter node type ('wnc', 'asc', or 'ats'), the node id, and the interface number on that node. The definitions of these fields may be found in IETF 1213.

ifstats all

Displays configuration information for all interfaces recognized by the controller. See `ifstats get` for details.

Syntax `ifstats all`

ifstats get

Displays statistics for one or more interfaces.

Syntax `ifstats get <node_type> <node_id> <index> [...]`

Argument	Description
node_id	One or more numbers identifying interfaces that are associated with this controller. Information is displayed for all the listed items. NOTE: To list information for all interfaces, use <code>ifstats all</code> .
node_type	See "Node Identification" on page 2 for more information on these arguments.
index	

Displays the following fields for each of the nodes requested:

Field	Description
Node Type	Identifies the type of node this interface belongs to ('WNC', 'ASC' or 'ATS').
Node ID	The unique numeric ID of the node.

Field	Description
Index	The unique numeric index of the ethernet interface in the interface statistics table.
Description	The descriptive name of the ethernet interface.
In Octets	Number of bytes received by the interface, including framing bytes.
In Unicast Packets	Number of unicast packets received by the interface and delivered to a higher-layer protocol.
In Non Unicast Packets	Number of non unicast packets received by the interface and delivered to a higher-layer protocol.
In Discard Packets	Number of non-errored packets received and discarded by the interface (e.g. buffer overflow prevents delivery to higher-layer protocol).
In Error Packets	Number of errored packets received by the interface.
In Unknown Protocols	Number of packets received and discarded because they were of unknown or unsupported protocols.
Out Octets	Number of bytes sent by the interface, including framing bytes.
Out Unicast Packets	Number of unicast packets that higher-layer protocols requested to be sent by the interface.
Out Non Unicast Packets	Number of non unicast packets that higher-layer protocols requested to be sent by the interface.
Out Discard Packets	Number of non-errored outbound packets discarded by the interface due to problems such as buffer overflows.
Out Error Packets	Number of outbound packets that could not be sent because of errors.
Out Queue Packet Length	

ifstats ids

Displays the node type, node number, and interface numbers for all Ethernet interfaces known to the controller.

Syntax `ifstats ids`

Description Displays a list of interface information for Ethernet interfaces that this controller knows about. This includes such interfaces on the controller, subcontrollers, and access points. The values for one interface are displayed on each output line.

PRELIMINARY

if: Interface Commands

These commands allow a user to set and display the attributes of ethernet interfaces in the system. Each interface is represented as a row in a table with the attributes as column or field values. Each row is uniquely identified by the node type, node ID, and interface index.

{interface | ifc}

Syntax {interface | ifc}

interface all

Displays configuration information for all interfaces recognized by the controller. See `interface get` for details.

Syntax interface all

interface get

Displays configuration information for one or more interfaces.

Syntax interface get <node_type> <node_id> <index> [...]

Argument	Description
node_type	See <code>ifstats get</code> .
node_id	One or more numbers identifying interfaces that are associated with this controller. Information is displayed for all the listed items. NOTE: To list information for all interfaces, use <code>ifstats all</code> .
index	

Description

Displays the following fields for each of the rows requested:

Field	Description
Node Type	Identifies the type of node this interface belongs to ('WNC', 'ASC' or 'ATS').
Node ID	The unique numeric ID of the node.
Index	The unique numeric index of the ethernet interface in the interface table.
Description	The descriptive name of the ethernet interface.
Type	The ARP type of the interface. Typically Typically 'Ethernet' or '802.11'.
Maximum Transfer Unit (bytes)	The MTU of the ethernet interface. From 0 to 65536 bytes. Node must be rebooted for a new value to take effect.
Interface Speed (Mbits/sec)	The speed of the ethernet interface, in megabits/second.
Physical Address	The MAC address of the ethernet interface. Node must be rebooted for a new value to take effect.
Operational Status	The operational status of the interface: 'UP' or 'DOWN'.
Last Change Time	Time of last operational state change of the interface.
Default Gateway	Default gateway (in nnn.nnn.nnn.nnn format) for the node on which this interface exists. Node must be rebooted for a new value to take effect.

interface ids

Displays the node type, node number, and interface numbers for all Ethernet interfaces known to the controller.

Syntax	<code>interface ids</code>
Description	Displays a list of interface information for Ethernet interfaces that this controller knows about. This includes such interfaces on the controller, subcontrollers, and access points. The values for one interface are displayed on each output line.

interface set

Sets one or more of the configuration values for the specified interface.

Syntax

```
interface set <node_type> <node_id> <index>
  {mtu=<val> | m=<val>}
  {physaddress=<val> | p=<val>}
  {defaultgateway=<val> | g=<val>}
```

Argument	Description
mtu	The MTU of the ethernet interface. From 0 to 65536 bytes. Node must be rebooted for a new value to take effect.
physaddress	The MAC address of the ethernet interface. Node must be rebooted for a new value to take effect.
defaultgateway	Default gateway (in nnn.nnn.nnn.nnn format) for the node on which this interface exists. Node must be rebooted for a new value to take effect.

qoS or codec: Quality of Service Commands

These commands manage codec-to-flow-spec mapping rules. Each rule is uniquely identified by its rule ID. The definition and meaning of most of these fields are clarified in IETF RFC 2210.

qoscodec all

Displays configuration information for all codecs. See `qoscodec get` for details.

Syntax `{qoscodec | codec} all`
Returns all the fields for all the rows in the table.

qoscodec del

Deletes the rows specified by the row IDs.

Syntax `{qoscodec | codec} del <rule_id> [...]`

qoscodec get

Displays configuration information for one or more codec rules.

Syntax `{qoscodec | codec} get <rule_id> [...]`

Argument	Description
rule_id	One or more numbers identifying codec flow-spec mapping rules that are associated with this controller. Information is displayed for all the listed items. NOTE: To list information for all codecs, use <code>qoscodec all</code> .

Description

Displays the following fields for each of the rules requested:

Field	Description
Rule ID	The unique numeric ID of the QoS codec flow spec mapping rule.
Codec	The codec type. Possible values are: default, g711u, 1016, g721, gsm, g723.1, dv14, dv14-2, lpc, g711a, g722, g722.1, mpa, g728, g729, 97red, siren, h261, or h263.
QoS Protocol	QoS protocol - sip, http or h323.
Traffic Spec Token Bucket Rate (bytes/sec)	The traffic spec token bucket rate. From 0 to 1,000,000 bytes/second.
Token Bucket Size (bytes)	Token bucket size in bytes. From 0 to 16,000 bytes.
Traffic Spec Peak Rate (bytes/sec)	Traffic spec peak rate. From 0 to 1,000,000 bytes/second.
Max Datagram Size (bytes)	Max datagram size. From 0 to 1,500 bytes.
Min Policed Unit (bytes)	Minimum policed units. From 0 to 1,500 bytes.
Reservation Spec Rate (bytes/sec)	Reservation spec rate. From 0 to 1,000,000 bytes/second.
Reservation Spec Slack (microsec)	Reservation spec slack. From 0 to 1,000,000 microseconds.
Sample Rate (bytes/sec)	Sample rate. From 0 to 200 bytes/second.

qoscodec ids

Displays the mapping rule numbers for all defined rules.

Syntax {qoscodec | codec} ids

Description Displays a list of rule numbers that are defined for this controller. Numbers are displayed one on each output line.

qoscodec new

Creates a new codec-to-flow-spec mapping .

```
{qoscodec | codec} new <rule_id>
codec=<val>
{qosprotocol=<val> | qp=<val>}
{tokenbucketrate=<val> | tbr=<val>}
{maxdatagramsize=<val> | max_pkt=<val>}
{minpolicedunit=<val> | min_unit=<val>}
{samplerate=<val> | sample=<val>}
[tokenbucketsize=<val> | tbs=<val>]
[peakrate=<val> | peak=<val>]
[rspecrate=<val> | rrate=<val>]
[rspecslack=<val> | rslack=<val>]
```

Description

Argument	Description
codec	The codec type. Possible values are: default, g711u, 1016, g721, gsm, g723.1, dv14, dv14-2, lpc, g711a, g722, g722.1, mpa, g728, g729, 97red, siren, h261, or h263.
{qosprotocol qp}	QoS protocol - sip, http or h323.
{tokenbucketrate tbr}	The traffic spec token bucket rate. From 0 to 1,000,000 bytes/second.
{maxdatagramsize max_pkt}	Max datagram size. From 0 to 1,500 bytes.
{minpolicedunit min_unit}	Minimum policed units. From 0 to 1,500 bytes.
{samplerate sample}	Sample rate. From 0 to 200 bytes/second.
tokenbucketsize tbs	Defaults to "8". Token bucket size in bytes. From 0 to 16,000 bytes.
peakrate peak	Defaults to "0". Traffic spec peak rate. From 0 to 1,000,000 bytes/second.

Argument	Description
rspecrate rrate	Defaults to "0". Reservation spec rate. From 0 to 1,000,000 bytes/second.
rspecslack rslack	Defaults to "0". Reservation spec slack. From 0 to 1,000,000 microseconds.

PRELIMINARY

qosrule or ruleQoS: Rule Commands

These commands manage Quality of Service (QoS) rules. Each rule is uniquely identified by its rule ID.

qosrule all

Displays configuration information for all QoS rules defined on the controller. See `qosrule get` for details.

Syntax `{qosrule | rule} all`

qosrule del

Deletes the rows specified by the row IDs.

Syntax `{qosrule | rule} del <rule_id> [...]`

qosrule get

Displays information about QoS rules.

Syntax `{qosrule | rule} get <rule_id> [...]`

Argument	Description
rule_id	One or more numbers identifying QoS rules that are associated with this controller. Information is displayed for all the listed items. NOTE: To list information for all rules, use <code>qosrule all</code> .

Description

Displays the following fields for each of the rules requested:

Field	Description
Rule ID	The unique numeric ID of the QoS rule.
Destination IP Address	The destination IP address.
Destination Port	The destination IP port.
Destination Netmask	The destination address netmask.
Source IP Address	The source IP address.
Source Port	The source IP port.
Source Netmask	The source address netmask.
Network Protocol	Network protocol: 'udp' or 'tcp'.
QoS Protocol	QoS protocol: 'sip', 'none', or 'h.323'.
VLAN ID	The virtual LAN ID: from 0 to 4095.
Average Packet Rate	Average packet rate: from 0 to 200 packets per second.
Action	Action: - 'forward', 'capture' (default) or 'drop'.
Drop Policy.	Drop Policy - tail, head.
TSpec Token Bucket rate	TSpec Token Bucket rate, from 0 to 1,000,000 bytes per second
Priority value 0-low 8-high)	The number (0-8) that specifies best effort priority queue where 0 is default (no priority) and 8 is highest priority
Traffic Control Policing	Valid values are 0 (zero) which turns it off, and P, which turns it on. By default is not set <

qosrule ids

Displays the rule numbers for all qos mapping rules defined for this controller.

Syntax `{qosrule | rule} ids`

Description Displays a list of rule numbers that are associated with this controller. Numbers are displayed one on each output line.

qosrule new

Creates a new QoS rule.

Syntax `{qosrule | rule} new <rule_id>
 {dstip=<val> | di=<val>}
 {dstport=<val> | dp=<val>}
 {dstmask=<val> | dm=<val>}
 {srcip=<val> | si=<val>}
 {srcport=<val> | sp=<val>}
 {srcmask=<val> | sm=<val>}
 {netprotocol=<val> | nprot=<val>}
 {qosprotocol=<val> | qprot=<val>}
 [vlanid=<val> | vl=<val>]
 [avgpktrate=<val> | avg=<val>]
 [action=<val> | act=<val>]
 [droppolicy=<val> | dr=<val>]
 [tokenbucketrate=<val> | tbr=<val>]
 [priority=<val>]
 [trafficcontrol=<val> | tc=<val>]`

Description See `qosrule get` for details.

Argument	Description
{dstip di}	The destination IP address.
{dstport dp}	The destination IP port.
{dstmask dm}	The destination address netmask.
{srcip si}	The source IP address.

Argument	Description
{srcport sp}	The source IP port.
{srcmask sm}	The source address netmask.
{netprotocol nprot}	Network protocol: 'udp' or 'tcp'.
{qosprotocol qprot}	QoS protocol: 'sip', 'http', or 'h.323'.
vlanid vl	Defaults to "0". The virtual LAN ID: from 0 to 4095.
avgpktrate avg	Defaults to "0". Average packet rate: from 0 to 200.
action act	Defaults to "capture". Action: - 'forward', 'capture' (default) or 'drop'.
droppolicy dr	Defaults to "tail". Drop Policy - tail, head.
tokenbucketrate tbr	Defaults to "0". TSpec Token Bucket rate
priority	Defaults to "0". The number (0-8) that specifies best effort priority queue where 0 is default (no priority) and 8 is highest priority
trafficcontrol tc	Defaults to "0". The only current value is [P]olicing that will turn on the flag By default is not set

qosstats

Displays QoS global statistics.

Syntax `qosstats {get | g}`

Description Displays the following QoS global statistics:

- H.323, SIP, and total session counts.
- H.323, SIP, and total rejected counts.
- H.323, SIP, and total pending counts.
- QoS active flow count.
- Qos pending flow count.

qosvars

These commands manages Quality of Service (QoS) by setting global parameters.

qosvars get

Displays information about QoS variables.

Syntax

```
qosvars get
```

Description

Displays the following fields:

Field	Description
QoS State	QoS state: 'on' or 'off'.
Admission Control	Admission Control: admitall, pending, reject
Drop Policy	Drop Policy - tail, head
Time to live	Default time to live, in seconds.
UDP time to live	Default UDP time to live, in seconds.
TCP time to live	Default TCP time to live, in seconds.
Steal time	Default steal time, in seconds.
UDP steal time	UDP steal time, in seconds.
TCP steal time	TCP steal time, in seconds.
Bandwidth Scaling	Scale factor for Tspec bandwidth in percent. May range from 1% to as high as 100,000%, although 100% is typical.

qosvars set

Sets one or more of the configuration values for the specified interface.

Syntax

```
qosvars set
```

```

[onoff=<val>]
[admission=<val>
[{droppolicy=<val> | drop=<val>}]
[ttl=<val>]
[udpttl=<val>]
[tcpttl=<val>]
[stealtime=<val>
[{udpstealtime=<val> | udpsteal=<val>}]
[{tcpstealtime=<val> | tcpsteal=<val>}]
[{percentbwscaling=<val> | bwscaling=<val>}]

```

Description For details, see `qosvars get`.

Argument	Description
onoff	QoS state ON or OFF
admission	Admission Control. Valid values are <ul style="list-style-type: none"> • admitall • pending • reject
droppolicy (or drop)	Drop Policy. Valid values are tail or head
ttl	Default time to live in seconds
udpttl	Default UDP time to live
tcpttl	Default TCP time to live
stealtime	Default steal time
{udpstealtime udpsteal}	UDP steal time
{tcpstealtime tcpsteal}	TCP steal time
{percentbwscaling bwscaling}	Scale factor for Tspec bandwidth

Quit Commands

quit

Exit the command line interface.

Syntax `{quit | exit}`

PRELIMINARY

Reboot (or restart) Commands

{reboot | restart}

Reboot one or all the nodes in the network.

Syntax `{reboot | restart}`

reboot all

Reboot all the nodes in the WLAN infrastructure.

Syntax `reboot all`

reboot asc

Syntax `reboot asc <node_id>`

Reboot the ASC node indicated by the argument `<node_id>`.

reboot ats

Reboot the ATS node indicated by the argument `<node_id>`.

Syntax `reboot ats <node id>`

reboot wnc

Reboot the WNC node indicated by the optional argument <node_id>, otherwise reboot the current WNC node.

Syntax `reboot wnc [<node_id>]`

PRELIMINARY

security: RADIUS Security Commands

{security | sec}

Syntax {security | sec}

These commands allow a user to set and query RADIUS and related security and authentication information.

security get

Displays configuration information for how a controller interfaces with a RADIUS server.

Syntax security get

Displays the following fields:

Field	Description
Node ID	The unique numeric ID of the node.
Privacy Bit	The privacy state that the ATs should use; either ON, OFF, or AUTO.
Security Mode	The security mode that the ATs should use; either OPEN or 802.1x.
RADIUS Server IP Address	IP address of the RADIUS server in nnn.nnn.nnn.nnn format.
RADIUS Server Port	Port number to use to connect to the RADIUS server.
RADIUS Secret	The RADIUS secret.
Rekey Period (seconds)	The rekey period, in seconds.

security set

Sets one or more of the configuration values for security.

Syntax

```
security set
  [privacy=<val>]
  [{securitymode=<val> | mode=<val>}]
  [{radiusip=<val> | ip=<val>}]
  [{radiusport=<val> | port=<val>}]
  [{radiussecret=<val> | secret=<val>}]
  [rekeyperiod=<val> | rekey=<val>]
```

Description

Sets one or more of the following values:

Argument	Description
privacy	The privacy state that the ATSS should use; valid values are ON, OFF, or AUTO.
securitymode (or mode)	The security mode that the ATSS should use; either OPEN or 802.1x.
radiusip (or ip)	IP address of the RADIUS server in nnn.nnn.nnn.nnn format. The controller must be rebooted for this to take effect.
radiusport (or port)	Port number to use to connect to the RADIUS server. The controller must be rebooted for this to take effect.
radiussecret (or secret)	The RADIUS secret. The controller must be rebooted for this to take effect.
rekeyperiod (or rekey)	The rekey period, in seconds. The controller must be rebooted for this to take effect.

Description

The controller must be rebooted for the changes to take effect.

SNMP Commands

These commands manage SNMP access privileges and trap destinations.

snmp del

Delete an existing SNMP community or trap destination entry.

Syntax

```
snmp del {community <communityName>
          | trap <ipAddress>}
```

Description When deleting a community entry, you must supply the <communityName> of the entry to delete. When deleting a trap destination entry, you must supply the <ipAddress> in dot format.

snmp get

Displays one or all existing SNMP community or trap destination entries.

Syntax

```
snmp get {community [<communityName>]
          | trap [<ipAddress>]}
```

Argument	Description
communityName	One or more numbers identifying SNMP entries that are managing or monitoring this controller. Information is displayed for all the listed items. NOTE: To list information for all SNP entries, use <code>snmp get community</code> .
ipAddress	

To display one community entry, you must supply the <communityName>, otherwise all entries are displayed. To display one trap destination entry, you must supply the <ipAddress> in dot format, otherwise all entries are displayed.

snmp new

Create a new SNMP community or trap destination entry.

Syntax

```
snmp new {community <communityName> <accessPrivilege> <IP Address>
| trap <ipAddress> <communityName>}
```

When adding a new community entry, you must supply a new <communityName>, the <accessPrivilege> (RW or RO) and allowed <IP Address>, which must be dot format or 'any' for any IP addresses. When adding a new trap destination entry, you must supply <ipAddress> in dot format and a <communityName> that is accepted by the receiver.

snmp set

Sets one or more of the configuration values for the specified SNMP management community.

Syntax

```
snmp set
{community <communityName> <accessPrivilege> <IP Address>
| trap <ipAddress> <communityName>}
```

Set either the SNMP community authentication and trap destination values.

When changing community information you must supply an existing community

Argument	Description
<communityName> and the new values for both <accessPrivilege> and <IP Address>	
<accessPrivilege>	is RW or RO, and
<IP Address>	must be in dot format, or 'any' for any IP addresses.
<ipAddress>	When changing trap destinations, you need to supply

Argument	Description
<communityName>	ditto
<ipAddress>	is the trap receiver IP address and is
<communityName>	is accepted by the receiver.

PRELIMINARY

Station Commands

The station commands manage stations. Each station is uniquely identified by a MAC address.

station all

Displays configuration information for all stations. See `station get` for details.

Syntax `station all`

station del

Deletes the rows specified by the row IDs.

Syntax `station del <macaddress> [...]`

station get

Displays configuration information for one or more stations.

Syntax `station get <macaddress> [...]`

Argument	Description
macaddress	One or more MAC addresses that are associated with this controller. Information is displayed for all the listed items.
	NOTE: To list information for all stations, use <code>station all</code> .

Description Displays the following fields for each of the stations requested:

Field	Description
MAC Address	The MAC address of the station.
Description	The descriptive name of the ethernet interface.
Assignment Type	The mechanism by which the IP address was assigned, either STATIC or DYNAMIC.
IP Address	The IP address assigned to the station.
ATS Node	The ATS node associated with the station.
ASC Node	The ASC node associated with the station.
Availability Status	The availability of the node: 'OFFLINE' or 'ONLINE'.

station set

Sets one or more of the configuration values for the station that has the specified MAC address.

Syntax

```
station set <macaddress>
  desc=<val>
  {ipaddress=<val> | ip=<val>}
```

Argument	Description
desc	The descriptive name of the ethernet interface.
ipaddress (or ip)	The IP address assigned to the station.

{stationstats | stastats}

Displays station statistics.

Syntax

```
{stationstats | stastats}
```

Displays the following station statistics:

Field	Description
	Station's MAC address.
	Number of successful DHCP requests.
	Number of successful address changes.
	Number of seamless (voluntary) hand-offs.
	Number of disassociated (involuntary) handoffs.
	SIP video reserved bandwidth, in bytes per second.
	SIP video actual bandwidth, in bytes per second.
	SIP video flows.
	SIP audio reserved bandwidth, in bytes per second.
	SIP audio actual bandwidth, in bytes per second.
	SIP audio flows.
	H.323 video reserved bandwidth, in bytes per second.
	H.323 video actual bandwidth, in bytes per second.
	H.323 video flows.
	H.323 audio reserved bandwidth, in bytes per second.
	H.323 audio actual bandwidth, in bytes per second.
	H.323 audio flows.

stationstats all

Displays statistical information for all stations. See {stationstats | stastats} for details.

Syntax `stationstats all`

topo: Network Topology Commands

{topoascats | ascats}

Displays all subcontroller-access point relationships.

Syntax {topoascats | ascats}

Description Display ASC-ATS relationships. Each row displays the ASC and ATS node IDs and the relationship between each pair. The relationship can be 'None', 'Bound', or 'Visible'.

topoascats all

Displays information about relationship between subcontrollers and access points.

Syntax topoascats all

topoats

Display access points seen by the controller. Each row displays the following:

Syntax topoats

Field	Description
NodeID	The ATS node ID.
RsrcRqst	Fraction of bandwidth, in microseconds per second, for that ATS.
RsrcAllcd	Fraction of bandwidth, in microseconds per second, for the group.

Field	Description
NghbrCnt	Number of visible ATS neighbors.
AttchCnt	Number of stations probed and associated.
AssgdCnt	Number of stations that have been associated.
RsrcAllcFrqcy	Resource allocation frequency, in times per second. Typically between 10 and 50.

topoats all

Displays information about all access points and summary information about their neighboring access points.

Syntax `topoats all`

{topoatsats | atsats}

Syntax `{topoatsats | atsats}`

Display the Received Signal Strength Indicator (RSSI) between all pairs of ATSs that are within range of each other. Each entry displays the RSSI of the source (HeadID) at the receiving end (TailID). The RSSI is a unitless value with typical values between 17 and 42.

topoatsats all

Displays information about relationship among access points.

Syntax `topoatsats all`

{topostaats | staats}

Display station/ats edge records.

Syntax {topostaats | staats}

An entry will be displayed for every station that is within range of an ATS. In addition to the station MAC address and ATS ID, a flag is displayed indicating which ATS the station has been assigned to. The Received Signal Strength Indicator (RSSI) for the station at the ATS is also displayed. RSSI values (a relative, and therefore unitless, measure) typically range from 17 to 42.

topostaats all

Displays information about relationship between stations and access points.

Syntax topostaats all

topostation all

Displays information about relationship among stations that are known by the controller.

Syntax {topostation | toposta} all

Field	Description
MAC	Station's MAC address.
RsrcRqst	Fraction of bandwidth, in microseconds per second.
AssgnATS	The ATS to which the station is assigned.
Handoff Time	Date and time of last handoff.
DataRate	Data rate in megabits per second.
#ATSAtch	The ATS to which the station is attached.
PwerSvMde	Power save mode; 'on' or 'off'.

Field	Description
AssocState	Current association state between the station and attached ATS: 'probe' (probing), 'auth' (authenticating), or 'assoc' (associated).
Chnl	Channel in use (1 to 11).
LastActiveTime	Time of last activity seen from station.
LastPollTime	Last time that station polled the ATS.

PRELIMINARY

Watchdog Commands

{watchdog | wd}

Syntax `{watchdog | wd}`

To disable the watchdog, set the polling period to zero. To enable the watchdog, set the polling rate to any period larger than zero.

watchdog get

Displays the current state of the watchdog.

Syntax `watchdog get`

watchdog set

Set the heartbeat polling period to `<polling-time>` seconds. Setting the polling-time to zero disables the watchdog

Syntax `watchdog set <polling-time>`

watchdog get

Displays the current state of the watchdog.

Syntax `watchdog get`

watchdog set

Set the heartbeat polling period to <polling-time> seconds.

Syntax `watchdog set <polling-time>`

Setting the polling-time to zero disables the watchdog

wirelessif or wif: Wireless Interface Commands

These commands manage 802.11 wireless interfaces (the antennae on access points). Each interface is uniquely identified by a node ID.

wirelessif all

Displays configuration information for all wireless interfaces managed by the controller. See `wirelessif get` for details.

Syntax `wirelessif all`

wirelessif get

Displays configuration information for one or more wireless interfaces.

Syntax `wirelessif get <node_id> [...]`

Argument	Description
node_id	One or more node numbers identifying wireless interfaces that are associated with this controller. Information is displayed for all the listed interfaces. NOTE: To list information for all wireless interfaces, use <code>wirelessif all</code> .

Description Displays the following fields for each of the nodes requested:

Field	Description
Node ID	The unique numeric ID of the node.
Description	A text description for this entry.
Operational Status	The operational status of the interface: 'UP' or 'DOWN'.
Last Change Time	Time of last operational state change of the interface.
Type	The type of the wireless interface.
BSSID	The Basic Service Set ID of the wireless interface.
ESSID	The Extended Service Set ID of the wireless interface.
Channel	The channel number the wireless interface is to use.
Low Transmit Power (dBm):	The low transmission power of the wireless interface, from 12 dBm to 22 dBm.
Medium Transmit Power (dBm):	The medium transmission power of the wireless interface, from 12 dBm to 22 dBm.
High Transmit Power (dBm):	The high transmission power of the wireless interface, from 12 dBm to 22 dBm.
Antenna Set:	The antenna set, either 'internal' or 'external', to use.
Antenna Number:	The antenna to use, either '1', '2', or 'both'.
Base Transmit Rate 1 (Mbits/sec):	First base transmit rate, in megabits per second.
Base Transmit Rate 2 (Mbits/sec):	Second base transmit rate, in megabits per second.
Base Transmit Rate 3 (Mbits/sec):	Third base transmit rate, in megabits per second.
Base Transmit Rate 4 (Mbits/sec):	Fourth base transmit rate, in megabits per second.
Supported Transmit Rate 1 (Mbits/sec):	First supported transmit rate, in megabits per second.
Supported Transmit Rate 2 (Mbits/sec):	Second supported transmit rate, in megabits per second.
Supported Transmit Rate 3 (Mbits/sec):	Third supported transmit rate, in megabits per second.
Supported Transmit Rate 4 (Mbits/sec):	Fourth supported transmit rate, in megabits per second.

wirelessif ids

Displays the node numbers for all known subcontrollers.

Syntax `wirelessif ids`

Description Displays a list of node numbers for wireless interfaces known by this controller. Numbers are displayed one on each output line.

wirelessif set

Sets one or more of the configuration values for the specified interface.

Syntax

```
wirelessif set <node_id>
  desc=<val>
  channel=<val>
  {lowpower=<val> | low=<val>}
  {mediumpower=<val> | medium=<val>}
  {highpower=<val> | high=<val>}
  antennaset=<val>
  antennum=<val>
  {baserate1=<val> | rate1=<val>}
  {baserate2=<val> | rate2=<val>}
  {baserate3=<val> | rate3=<val>}
  {baserate4=<val> | rate4=<val>}
  txrate1=<val>
  txrate2=<val>
  txrate3=<val>
  txrate4=<val>
```

Description For details, see `wirelessif get`.

Argument	Description
desc	A text description for this entry.
channel	The channel number the wireless interface is to use.
{lowpower low}	The low transmission power of the wireless interface, from 12 dBm to 22 dBm.

Argument	Description
{mediumpower medium}	The medium transmission power of the wireless interface, from 12 dBm to 22 dBm.
{highpower high}	The high transmission power of the wireless interface, from 12 dBm to 22 dBm.
antennaset	The antenna set, either 'internal' or 'external', to use.
antennanum	The antenna to use, either '1', '2', or 'both'.
{baserate1 rate1}	First base transmit rate.
{baserate2 rate2}	Second base transmit rate.
{baserate3 rate3}	Third base transmit rate.
{baserate4 rate4}	Fourth base transmit rate.
txrate1	First supported transmit rate.
txrate2	Second supported transmit rate.
txrate3	Third supported transmit rate.
txrate4	Fourth supported transmit rate.

wnc: Controller Commands

These commands manage information about the operation of the controller and certain global system parameters.

wnc get

Displays configuration information for the controller.

Syntax

```
wnc get
```

Displays the following fields:

Field	Description
Node ID	The unique numeric ID of the node.
Description	A text description for this entry.
Uptime	The uptime of the node, in hours, minutes, and seconds (hh:mm:ss).
Location	Description of the location of this controller.
Contact	Person or organization responsible for this controller.
Operational Status	The operational state of the controller: <ul style="list-style-type: none"> • ENABLED: The controller is operating correctly. • DISABLED: The controller is known by the controller but it is not operating correctly.
Availability Status	The availability of the controller: <ul style="list-style-type: none"> • OFFLINE: The controller cannot find the controller. • ONLINE: The controller can find the controller.
Alarm State	The severity of the current alarm on the controller. If more than one alarm is current, the highest severity is displayed. In order of increasing severity, the states are NO ALARM, MINOR, MAJOR, or CRITICAL.
Beacon Interval (msec)	Beacon Interval in milliseconds.

Field	Description
Assignment Algorithm	Assignment Algorithm.
Virtual IP Subnet Address	This value, when masked by the virtual netmask value, specifies the virtual IP subnet for mobile clients. All nodes must be rebooted for a new value to take effect.
Virtual Netmask	This value is used to mask the virtual IP subnet address to specify the virtual IP subnet for mobile clients. All nodes must be rebooted for a new value to take effect.
DHCP Server IP Address	IP address of the DHCP server.
Virtual MAC Address	The one virtual MAC address for all access points (ATs).
ESSID	The Extended Service Set ID for the network.
Default ATS Init Script	Default ATS init script.
Statistics Polling Period	Statistics polling period, in seconds (0 = no polling).

wnc set

Sets one or more of the configuration values for the specified controller.

Syntax

```
wnc set
  desc=<val>
  location=<val>
  contact=<val>
  {beaconinterval=<val> | beacon=<val>}
  {assignmentalgo=<val> | assign=<val>}
  virtualip=<val>
  virtualnetmask=<val>
  dhcp=<val>
  virtualmac=<val>
  essid=<val>
  atsscript=<val>
  polling=<val>
  audit=<val>
```

Argument	Description
desc	A user provided description for this entry.
location	Description of the location of this piece of equipment.
contact	Person or organization responsible for this piece of equipment.
beaconinterval beacon	Beacon Interval in milliseconds.
assignmentalgo assign	Assignment Algorithm. For internal use only.
virtualip	This value, when masked by the virtual netmask, specifies the virtual IP subnet for mobile clients. You must reboot the controller for this to take effect.
virtualnetmask	This value is used to mask the virtual IP subnet address to specify the virtual IP subnet for mobile clients. You must reboot the controller for this to take effect.
dhcp	IP address of the DHCP server. You must reboot the controller for this to take effect.
virtualmac	The one virtual MAC address for all access points (ATs) managed by this controller. You must reboot the controller for this to take effect. The default is For more information, see "Roaming Using Shared Virtual MAC Address" on page 65. Because all controllers use the same default value, if you have more than one Meru wireless network with overlapping ranges and you do <i>not</i> want mobile stations to roam transparently between the networks, you can prevent it by changing this value for one of the controllers.
ssid	The Extended Service Set ID for the network You must reboot the controller for this to take effect.
atsscript	Default access point initialization script. See <code>ats scripts</code> for more information.
polling	Statistics polling period, in seconds (0 = no polling). How often the controller updates information about the number of packets passed or dropped and so on.
audit	Audit polling period, in seconds (0 = no audit). How often the controller updates information about

Roaming Using Shared Virtual MAC Address

To enable the Meru wireless network to easily handle roaming mobile stations, all access points managed by a single controller use the same virtual MAC address. Therefore, all access points look like essentially the same access point, so that a mobile station's transfer from one access point to the next is quick and seamless.

PRELIMINARY

Glossary

This glossary contains a collection of terms and abbreviations used in this document.

Alphabetic List of Terms and Abbreviations

100baseT A fast Ethernet standard that uses two pairs of twisted wire and allows up to 100 megabits per second (Mbps).

A

access point

ANC Obsolete term for *subcontroller*.

ATS Obsolete term for *access point*.

B

BSSID Basic Service Set ID, a means of uniquely identifying an *access point*, usually intended for machine use rather than human use. See also *ESSID*.

C

CLI Command-line interpreter. Similar to a shell for giving instructions to a controller.

controller

D

DHCP Dynamic Host Configuration Protocol. A protocol that allows a network administrator to manage and automatically assign IP numbers to client computers. IP numbers are released to client machines with a lease time. The lease time determines how long the client can retain the IP number without using it.

E

ESSID The extended service set identifier (ID) for one or more access points. This is a string of up to 32 characters that is intended to be viewed by humans. A set of access points can share an ESSID. In this case, a mobile station can roam among the access points. See also *BSSID*.

G

H

I

IP Number Also called an IP address. A 32-bit binary number used to identify senders and receivers of traffic across the Internet. It is usually expressed in the form *nnn.nnn.nnn.nnn* where *nnn* is a number from 0 to 256.

J

K

L

M

Mbps Million bits (megabits) per second.

mobile station

N

NAT Network address translation. A system for converting the IP numbers used on one network to the IP numbers used in another network. Usually one network is the inside network and one network is the outside network. Usually the IP numbers on the inside network form a relatively large set of IP numbers, which must be compressed into a small set of IP numbers on the outside network.

O

P

PSTN Public Switched Telephone Network. The usual way of making telephone calls in the late 20th century, designed around the idea of using wires and switches. Perhaps to be supplanted by *VoIP* in the 21st century.

Q

QoS Quality of Service. A set of technologies for managing and allocating Internet bandwidth, often used to ensure timely delivery of multimedia traffic.

R

RADIUS Remote Authentication Dial-In User Service. A service that authorizes connecting users and allows them access to requested systems or services. The Microsoft ISA server is a RADIUS server.

S

SIP Session Initiation Protocol. SIP is a protocol for finding users, usually human, and setting up multimedia communication among them, typically a *VoIP* phone call.

ssh Secure SHell. A terminal-emulation program that allows users to log onto a remote device and execute commands. It encrypts the traffic between the client and the host.

subcontroller

T

U

UTC Abbreviation for Coordinated Universal Time (as defined by the International Telecommunication Union (ITU)). Also known as Greenwich Mean Time. The time is not adjusted for time zones or for daylight savings time.

V

virtual private network (VPN)

VoIP Voice over IP. A set of protocols for phone calls where the Internet, rather than the PSTN, is used to connect users.

VPN Abbreviation for *virtual private network*.

W

WEP Wired Equivalent Privacy. A means of encrypting data broadcast over a wireless link. WEP can be based on either a 64-bit or a 128-bit key and is part of the 802.11 standard.

WNC Obsolete term for *controller*.

X

Y

Z

Index

A

alarms command 3
alphabetical listing of terms 67
asc all command 5
ASC definition 2
asc get command 5
asc ids command 6
ATS definition 2

C

command prompt 2
connect command 17
cons command 18
console command 18
conventions, typographical viii
cross-reference formats 3

K

key point, explanation of viii

N

note, explanation of viii

P

prompt for CLI 2

R

reference, explanation of viii
remote command 17

S

symbols viii

T

terminology

alpha listing 67
typographical conventions viii

WXYZ

warning, explanation of viii
wirelessif all command 59
wirelessif set command 61
WNC definition 2
wnc get command 63
wnc> prompt 2

PRELIMINARY

Meru License Agreement (Site)

READ BEFORE COPYING, INSTALLING OR USING.

associated materials (collectively, the “Software”) until you have carefully read, understood, and agreed to the terms of this Agreement. You may not copy, modify, or use the Software.

Meru Networks, Inc. (“Meru”) grants to you the following non-exclusive, non-assignable, license to use the Software. The Software may include portions offered on terms in addition to those set out here, including the following:

• Installation, installation or development utilities, and other materials, including but not limited to network access points. Developer tools are provided in binary format. You may use them as licensed hereunder, but you may not redistribute them.

• Libraries. Libraries may be provided in source code and binary formats. You may use them as licensed hereunder, but you may not redistribute source code. You may use them only with your products that use Meru processors (controller, access point). Other software may require an additional license.

• Binaries provided in binary format only. You may copy and distribute Licensed Binaries with

the Binaries provided in source or binary format. You may use Meru Source internally for the purposes of this Agreement, but you may not redistribute source code. You may copy and distribute Meru Binaries with non-Meru products. Distribution with non-Meru products may require an additional license.

• Trial Software. Trial Software may be used solely for internal test and evaluation purposes for sixty (60) days. To continue use after sixty (60) days you may need to purchase a license from Meru. Trial Software, except that you may copy and distribute compiled, binary versions of the Software for network processors.

• Materials intended for end users. You may copy, modify, and distribute them.

• Software on this CD, which may be subject to different licensing requirements and restrictions. Software will be governed by the terms and conditions contained in the licensing.txt file. Other Software. You hereby acknowledge that all files and directories that do not have a license file are subject to the terms of this Agreement.

• Meru reserves the right to discontinue distribution of the portions of the Software that you are licensed to use. Meru may issue an update, upgrade, or new version of the Software and to make reasonable efforts to notify you of such updates to your customers who have received the Software herein.

• Meru may reverse-engineer any software provided solely in binary form.

• The following limitations: you (i) are solely responsible to your customers for any update or modification to the Software, (ii) do not make any statement that your product is endorsed or approved by Meru, (iii) do not use Meru’s name or trademarks to market your product without Meru’s prior written consent, (iv) shall not reverse engineer, decompile, disassemble, or otherwise reverse engineer, and (v) shall indemnify, hold harmless, and defend Meru from and against all lawsuits, including attorney’s fees, that arise or result from your distribution of any software.

WARRANTY. Title to all copies of the Software remains with Meru or its suppliers. The Software is provided “as is” without any express or implied warranties, including warranties of fitness for a particular purpose, in the United States and other countries, and by international treaty provisions. Meru may make changes to the Software, or to items referenced therein, at any time without notice. Meru may discontinue or update the Software. Except as otherwise expressly provided, Meru grants you a limited license to use the Software for your internal business operations. Meru reserves all copyrights, trademarks, or other intellectual property rights. You may transfer the Software to another user only if you agree to these terms and if you retain no copies of the Software.

• If the Software has been delivered by Meru on physical media, Meru warrants the media to be free of defects for ninety days after delivery by Meru. If such a defect is found, return the media to Meru and Meru will replace the media as Meru may select.

DISCLAIMER. Except as provided above, the software is provided “as is” without any express or implied warranties, including warranties of merchantability, noninfringement, or fitness for a particular purpose. Meru makes no representation or warranty as to the accuracy or completeness of any information, text, graphics, links, or other items contained in the Software.

GOVERNMENT RESTRICTED RIGHTS. The Software is provided with "RESTRICTED RIGHTS." Use, duplication, or disclosure by the Government is subject to restrictions as set forth in FAR52.227-14 and DFAR252.227-7013 et seq. or its successor. Use of the Software by the Government constitutes acknowledgment of Meru's proprietary rights therein. Contractor or Manufacturer is Meru Networks, Inc., CA.

PRELIMINARY



Meru Networks, Inc.
www.merunetworks.com

Revision 0.1, July 2003