



NEXWARE COMMAND LINE INTERFACE CLI

NexConnect Business Class II Firmware Version AG1.2.2.0

User Manual

Copyright

©2010 Nexaira, Inc. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form by any means, electronic, mechanical, photocopying, recording or otherwise, without the proper written consent of Nexaira, Inc., 6650 Lusk Blvd, Ste B203 San Diego, CA 92121

Trademarks

Nexaira, the Nexaira logo, Nexaira Wireless, NexWare, i3 GUI, NexConnect and HANA are all registered trademarks of Nexaira, Inc. All other trademarks herein are property of their respective owners.

Disclaimer

Nexaira, Inc. assumes no responsibility for errors or omissions in this document, nor does Nexaira, Inc. make any commitment to update the information contained herein.

Table of Contents

Introduction.....	1
Features	1
Command Line Interface	2
NexWare Technical Support and Services	13
Sales and ordering information.....	13

Introduction

The Command Line Interface (CLI) for Open WRT is provided to manage NexConnect NexWare™ controllers.

Multiple systems containing NexWare controllers may be managed using the CLI via a command line or script. CLI is useful in environments where a graphical user interface (GUI) is not available.

Features

The NexWare CLI provides the basic functionality of the NexWare i3 GUI management utility through a Command Line Interface. You can view array status and version information, and perform maintenance functions. CLI also includes advanced features for reconfiguring the router online.

Command Line Interface

Status information on NexAira Business Class routers is handled through two applications: routerstatus and cardstatus.

Usage is as follows:

```
=====
Datacard information:
=====
```

Get datacard ESN:
cardstatus esn

Get datacard firmware:
cardstatus gmr

Get datacard prl:
cardstatus prl

Get datacard mdn:
cardstatus mdn

Get datacard manufacturer:
cardstatus manufacturer

Get datacard model:
cardstatus model

Get datacard IP Address:
cardstatus ipaddr

Get datacard gateway:
cardstatus gateway

Get datacard DNS servers:
cardstatus dns

Get datacard interface name:
cardstatus ifname

Get datacard signal quality:
cardstatus signal

Get all datacard information:
cardstatus all

OMA DM FUNCTIONS: <PORT> is the command port of the card. To get the active command port: uci get network.datacard.device. In these examples, we'll use /dev/ttyUSB0.

Update PRL:
cardupdate /dev/ttyUSB0 prl

Update Firmware:
cardupdate /dev/ttyUSB0 fumo

Activate Card:
cardupdate /dev/ttyUSB0 omadm

Send AT Command to Modem:
readmodem /dev/ttyUSB0 AT_COMMAND (AT, ATI, AT+CSQ, etc...)

=====
Router information:
=====

MAC / IP Address of all router interfaces:
ifconfig

MAC / IP Address of all connected devices:
routerstatus connecteddevices

Status of local network devices:
routerstatus localdevs

Status of Active WAN:
routerstatus wan

Status of WiFi:
iwconfig
AND
uci show wireless

WiFi Password Get and Set:
GET: uci get wireless.main.key
SET: uci set wireless.main.key=VALUE

Admin Password Set:
passwd

Router Restore to Default:

factorydefault

Routing Table Get:
ip route show

Syslog Get:
logread

Processor Occupancy:
top

Memory Occupancy:
free

QOS Settings get and set:
GET: uci show qos
SET: uci set qos.wan.SETTING=VALUE

All local router information:
routerstatus all

Router configuration backup:
config save <filename>

Router configuration load:
config load <filename>

Firmware upgrade:
sysupgrade <filename>

Get router firmware version:
routerstatus firmware

IP Address Set - sets the IP Address of the interface belonging to specified network. In this example, we're setting the IP of the LAN (eth0.1 on BC2) to 192.168.4.1:

```
uci set network.lan.ipaddr=192.168.4.1
```

To show what settings the network has:
uci show network

Static DHCP Leases:
Show static leases:
uci show luci_ethers

Example output:
luci_ethers.@static_lease[0]=static_lease
luci_ethers.@static_lease[0].macaddr=00:21:9B:EB:28:BE

```
luci_ethers.@static_lease[0].ipaddr=192.168.1.129
```

Add static lease:

```
uci add luci_ethers static_lease
```

```
uci set luci_ethers.@static_lease[-1].macaddr=<MAC ADDRESS OF  
INTERFACE>
```

```
uci set luci_ethers.@static_lease[-1].ipaddr=<DESIRED IP ADDRESS>
```

```
=====
```

IPsec

```
=====
```

Section: tunnel

Description: Name of this tunnel

Datatype: String

Required: True

```
-----
```

```
add tunnel: uci set ipsec.mytunnel=tunnel
```

Section: enabled

Description: Enable this tunnel

Datatype: Boolean

Required: True

```
-----
```

```
uci set ipsec.mytunnel.enabled=1 (enabled)
```

```
uci set ipsec.mytunnel.enabled=0 (disabled)
```

Section: auto

Description: How to initiate this tunnel

Datatype: String

Values:

route : start this tunnel when traffic is detected

start : start this tunnel immediately

```
-----
```

```
uci set ipsec.mytunnel.auto=start
```

Section: waniface

Description: WAN interface used by this tunnel

Datatype: String

Value References: network.interface

Value example: wan

Required: True

```
-----
```

```
uci set ipsec.mytunnel.waniface=wan
```

Section: keyexchange

Description: Which version of IKE to use

Datatype: String

Values:

ikev1 : IKE v1

ikev2 : IKE v2

Required : true

uci set ipsec.mytunnel.keyexchange=ikev1

Section: local_lan

Description: Local IP of LAN

Datatype: IPv4 Address

Required: True

uci set ipsec.mytunnel.local_lan=192.168.1.1

Section: local_lan_network

Description: Subnet of local LAN

Datatype: IPv4 Address

Required: True

uci set ipsec.mytunnel.local_lan_network=255.255.255.0

Section: remote_wan

Description: Public IP Address of remote peer

Datatype: IPv4 Address

Required: True

uci set ipsec.mytunnel.remote_wan=1.1.1.1

Section: remote_lan

Description: IP Address of LAN serviced by remote peer

Datatype: IPv4 Address

Required: True

uci set ipsec.mytunnel.remote_lan=10.0.0.1

Section: remote_lan_network

Description: Subnet of remote LAN

Datatype: IPv4 Address

Required: True

uci set ipsec.mytunnel.remote_lan_network=255.255.0.0

Section: integrity
Description: Hash algorithm to use for tunnel
Datatype: String
Required: True
Values:
sha1 : SHA1
md5 : MD5

uci set ipsec.mytunnel.integrity=md5

Section: encryption
Description: Encryption algorithm to use for tunnel
Datatype: String
Required: True
Values:
aes : AES Encryption
3des : 3DES Encryption

uci set ipsec.mytunnel.encryption=aes

Section: dhgroup
Description: bits for Diffie-Helman key exchange
Datatype: String
Required: True
Values:
modp1024 : DH Group 2 (1024 bit)
modp1536 : DH Group 5 (1536 bit)
modp2048 : DH Group 14 (2048 bit)
modp3072 : DH Group 15 (3072 bit)
modp4096 : DH Group 16 (4096 bit)
modp6144 : DH Group 17 (6144 bit)
modp8192 : DH Group 18 (8192 bit)

uci set ipsec.mytunnel.dhgroup=modp1024

Section: strict
Description: use strict settings when authenticating
Datatype: Boolean
Required: False

uci set ipsec.mytunnel.strict=1 (enabled)
uci set ipsec.mytunnel.strict=0 (disabled)

Section: key
Description: Pre-Shared Key
Datatype: String

Required: True

uci set ipsec.mytunnel.key=mypassword

=====
Email Alerts

=====
Section: enable

Description: Enable or Disable email alerts

Datatype: Boolean

Required: True

uci set logmail.config.enabled=1 (enable)

uci set logmail.config.enabled=0 (disable)

Section: tail

Description: Send tail of syslog (last 50 lines) only

Datatype: Boolean

Required: false

uci set logmail.config.tail=1 (enable)

uci set logmail.config.tail=0 (disable)

Section: boot

Description: Send alert on system boot

Datatype: boolean

Required: false

uci set logmail.config.boot=1 (enable)

uci set logmail.config.boot=0 (disable)

Section: failover

Description: Send alert on interface failover

Datatype: boolean

Required: false

uci set logmail.config.failover=1 (enable)

uci set logmail.config.failover=0 (disable)

Section: ifupdown

Description: Send alert on interface up/down

Datatype: boolean

Required: false

uci set logmail.config.ifupdown=1 (enable)
uci set logmail.config.ifupdown=0 (disable)

Section: name
Description: The name of this device
Datatype: String
Required: false

uci set logmail.config.name=MyRouter

Section: domain
Description: domain (URL) of mail server
Datatype: String
Required: True

uci set logmail.config.domain=myemailserver.com

Section: to
Description: Email address which will be receiving alerts
Datatype: String
Required: True

uci set logmail.config.to=myemail@mydomain.com

Section: mailhub
Description: URL to the SMTP server
Datatype: String
Required: True

uci set logmail.config.mailhub=mail.mydomain.com

Section: port
Description: Port used by SMTP server
Datatype: unsigned integer
Required: True

uci set logmail.config.port=25

Section: username
Description: user account used to send these emails
Datatype: String
Required: True

uci set logmail.config.username=alerts@mydomain.com

Section: password
Description: password to user account configured in logmail.config.user
Datatype: String
Required: True

uci set logmail.config.password=mypassword

=====
GRE Interfaces

=====
GRE Interfaces are defined in /etc/config/network

Add GRE Interface:
uci set network.mygre=interface
uci set network.mygre.proto=gre

Section: ipaddr
Description: IP Address of this GRE interface
Datatype: IPv4 Address
Required: True

uci set network.mygre.ipaddr=10.0.0.1

Section: netmask
Description: Subnet mask for this GRE interface
Datatype: IPv4 Address
Required: True

uci set network.mygre.netmask=255.255.255.0

Section: destwan
Description: Remote WAN IP for GRE interface
Datatype: IPv4 Address
Required: True

uci set network.mygre.destwan=1.1.1.1

Section: destlan
Description: Remote LAN IP for GRE interface
Datatype: IPv4 Address

Required: True

uci set network.mygre.destlan=192.168.2.1

Section: destnet

Description: Remote LAN subnet for GRE interface

Datatype: IPv4 Address

Required: True

uci set network.mygre.destnet=255.255.255.0

Section: source

Description: Source interface of this tunnel

Datatype: String

Value References: network.interface

Example Value: wan

Required: True

uci set network.mygre.source=wan

=====
SSH

SSH configuration is located in /etc/config/dropbear

Section: enabled

Description: Enable/Disable SSH server

Datatype: Boolean

Required: False

uci set dropbear.@dropbear[0].enabled=1

Section: passwordauth

Description: Enable password authentication. If enabled, SSH Keys must be used.

Datatype: Boolean

Required: False

uci set dropbear.@dropbear[0].passwordauth=1

Section: port

Description: Listening port for SSH server

Datatype: Integer

Required: True

uci set dropbear.@dropbear[0].port=22

Section: remote

Description: Allow SSH access from WAN side

Datatype: Boolean

Required: False

uci set dropbear.@dropbear[0].remote=0

Section: acl

Description: SSH access whitelist

Datatype: Boolean

Required: False

uci set dropbear.@dropbear[0].acl=1

Section: ip

Description: IP Addresses with permission to access SSH server from WAN side

Datatype: List

Required: False

Dependency of: dropbear.acl

uci add_list dropbear.@dropbear[0].ip=1.1.1.1

NexWare Technical Support and Services

Product information, Frequently Asked Questions, software upgrades, driver files and other support are available through the Nexaira world wide website at <http://www.nexaira.com>.

For specific answers to questions or to give feedback about the product, visit our Web site at <http://www.nexaira/support> and use our convenient e-mail form.

Sales and ordering information

For sales information to obtain a NexConnect Router or order NexWare firmware please send an email to sales@nexaira.com or visit our <http://nexaira.com>