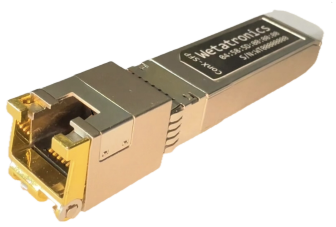
Conx Sfp User Manual

Date: 2026-07-02



Contents

Table of Contents

ConX-SFP User Manual

Wetatronics Secure Communication Platform

Version 1.1.0 | Document Revision 2.1

Table of Contents

1. Introduction

1.1 Product Overview

The ConX-SFP is a secure console server integrated into an SFP (Small Form-factor Pluggable) module. It provides encrypted remote access to serial console ports on network devices such as routers, switches, and firewalls.

Key Features:

Feature	Description
SSH Server	Secure Shell access with modern cryptography
Telnet Server	Legacy plaintext access for compatibility
Dual Serial Ports	Two RS-232 ports via single RJ45 connector
Cloud Platform	Secure connectivity to Wetatronics management platform
Robust Recovery	Multi-stage bootloader with firmware recovery options

1.2 Security Features

The ConX-SFP implements industry-standard security protocols:

- SSH-2.0 Protocol: Secure remote access with strong encryption
- TLS 1.2: Secure cloud connectivity with certificate validation
- Modern Cryptography: Ed25519 signatures, Curve25519 key exchange, AES encryption
- Perfect Forward Secrecy: Ephemeral key exchange protects past sessions
- Secure Password Storage: Passwords are hashed and never stored in plaintext

---

2. Getting Started

2.1 Default Configuration

Setting	Default Value
IP Address	DHCP (fallback: 192.168.0.232/24)
SSH Port	22
Telnet CLI Port	23
Serial 1 TCP Port	3001
Serial 2 TCP Port	3002
Username	`weta`
Password	`wetatronics`
Serial Baud Rate	9600 N81

2.2 Physical Installation

1. Insert the ConX-SFP into an available SFP port on your network device
2. Connect the RJ45 cable to the target device's console port
3. Wait approximately 2 seconds for the SSH server to become ready

The ConX-SFP emulates a 1000BASE-LX fiber SFP and is compatible with standard SFP ports on network switches, routers, and other equipment.

2.3 First Connection

Connect via SSH using the default credentials:

```
ssh weta@192.168.0.232
```

When prompted, enter the password: `wetatronics`

You will see the CLI prompt:

```
Welcome to ConX-SFP
Type 'help' for available commands

conxsfp>
```

2.4 Initial Configuration Checklist

1. [ ] Change the default password: `config user delete weta` then `config user add weta <new_password> admin`
2. [ ] Add additional admin users: `config user add <username> <password> admin`
3. [ ] Configure network settings if not using DHCP
4. [ ] Test serial port connectivity: `serial 1`
5. [ ] Save configuration: `system save`

---

3. Network Connectivity

3.1 DHCP Configuration

DHCP is enabled by default with vendor option support. The ConX-SFP will request an IP address from your network's DHCP server on boot and can receive device configuration via DHCP vendor options for zero-touch provisioning (see Section 3.5).

To check current network configuration:

```
conxsfp> show network
Network Status:
  Current Runtime Configuration:
    IP Address: 192.168.1.100
    Netmask: 255.255.255.0
    Gateway: 192.168.1.1

  Configured (Saved) Settings:
    DHCP: Enabled
    DHCP Vendor Opts: Enabled
    Hostname: Wetatronics_0036
    SSH Port: 22
    Primary DNS: 8.8.8.8
    Secondary DNS: 1.1.1.1
```

To disable DHCP:

```
conxsfp> config network dhcp off
DHCP disabled
```

To enable/disable DHCP vendor options:

```
conxsfp> config network dhcp vendor on
DHCP vendor options enabled

conxsfp> config network dhcp vendor off
DHCP vendor options disabled
```

3.2 Static IP Configuration

Set a static IP address:

```
conxsfp> config network static 192.168.1.100 255.255.255.0 192.168.1.1
Static IP configuration set:
  IP:      192.168.1.100
  Netmask: 255.255.255.0
  Gateway: 192.168.1.1
  DHCP:    disabled
Note: Save configuration with 'system save' and reboot to apply
```

Configure DNS servers:

```
conxsfp> config network dns 8.8.8.8 1.1.1.1
Primary DNS: 8.8.8.8
Secondary DNS: 1.1.1.1
```

Save and apply changes:

```
conxsfp> system save
Saving configuration to flash...
Configuration saved to flash successfully

conxsfp> system restart
```

3.3 Hostname Configuration

Set a custom hostname:

```
conxsfp> config network hostname my-console-server
Hostname set to: my-console-server
```

The hostname is used in DHCP requests and helps identify the device on your network.

3.4 LLDP Support

The ConX-SFP supports Link Layer Discovery Protocol (LLDP), allowing it to advertise its presence to neighboring network devices.

Benefits: - The ConX-SFP will appear as an LLDP neighbor on any network device that supports the protocol - Helps with device discovery when the IP address is unknown - Provides device identification information to network management systems

Finding the device via LLDP:

On Cisco devices:

```
show lldp neighbors detail
```

On Juniper devices:

```
show lldp neighbors
```

This is particularly useful for discovering the device's IP address when DHCP is in use and you don't know the assigned address.

3.5 DHCP Vendor Options

DHCP vendor options enable zero-touch mass deployment of ConX-SFP devices. A DHCP server can automatically configure the portal URL, serial port settings, SSH, telnet, power control, and assign a deployment group ID -- eliminating the need to manually configure each device.

When vendor options are enabled: - The device sends Option 60 with the vendor class identifier `wetatronics` to identify itself - The DHCP server responds with Option 43 (vendor-specific sub-options) and/or Option 160 (portal URL) - Configuration is applied at DHCP lease time without user intervention

Enable/disable vendor options:

```
conxsfp> config network dhcp vendor on
conxsfp> config network dhcp vendor off
```

Vendor options are enabled by default on new devices.

3.5.1 Option 43 Sub-Options Reference

Sub-opt	Name	Type	Description
1	Portal URL	string (max 127)	WebSocket portal URL (scheme stripped automatically)
2	Autoconnect	uint8 (0/1)	Enable portal auto-reconnect
3	Serial 1 Baud	uint32	USART1 baud rate (1200-460800, network byte order)
4	Serial 2 Baud	uint32	USART3 baud rate (1200-460800, network byte order)
5	Device Name	string (max 63)	Override hostname
6	Telnet CLI Port	uint16	CLI port (0=disabled, default 23)
7	Serial1 Port	uint16	Passthrough port (0=disabled, default 3001)
8	Serial2 Port	uint16	Passthrough port (0=disabled, default 3002)
9	SSH Port	uint16	SSH port (0=disable with sub-opt 10 magic)
10	SSH Disable	uint8	Must be 0xAD with port=0 to disable SSH
11	Power	uint8 (0/1)	External power control
12	Group ID	string (max 16)	Deployment group identifier
13	Persist	uint8 (0/1)	Write config to flash (only if changed)
14	Break Detection	uint8 (0/1)	USART1 hardware break detection

3.5.2 Behaviour Notes

- All values update the runtime configuration (visible via CLI). Flash is only written when sub-option 13 (Persist) is set to 1.
- Portal URL: The scheme prefix (`wss://` , `https://`) is stripped automatically. Only `host[:port][/path]` is needed. Missing URL components inherit from the current saved config.
- Baud rates are only applied on initial DHCP lease, not on renewal.
- SSH disable requires BOTH port=0 (sub-opt 9) AND magic byte 0xAD (sub-opt 10) as a safety measure.
- Option 160 (portal URL string) overrides sub-option 1 if both are present.
- `show network` displays which sub-options the DHCP server set and the group ID.
- Sub-options that are not sent by the server keep their current values unchanged.

3.5.3 DHCP Server Configuration

MikroTik RouterOS:

```
/ip dhcp-server option
add name="wetatronics-portal" code=1 value="'portal.example.com:8443/ws/device/'"
add name="wetatronics-autoconnect" code=2 value="0x01"
add name="wetatronics-group" code=12 value="'site-A'"

/ip dhcp-server option sets
add name="wetatronics-set" options=wetatronics-portal,wetatronics-autoconnect,wetatronics-
group

/ip dhcp-server network
set [find] dhcp-option-set=wetatronics-set
```

Note: MikroTik encodes option 43 sub-options using the `code=N` within the option set. The values use MikroTik encoding: strings in single quotes, raw hex with 0x prefix.

ISC DHCP (Linux):

```
option space wetatronics;
option wetatronics.portal-url      code 1  = text;
option wetatronics.autoconnect     code 2  = unsigned integer 8;
option wetatronics.serial1-baud    code 3  = unsigned integer 32;
option wetatronics.serial2-baud    code 4  = unsigned integer 32;
option wetatronics.device-name     code 5  = text;
option wetatronics.telnet-port     code 6  = unsigned integer 16;
option wetatronics.serial1-port    code 7  = unsigned integer 16;
option wetatronics.serial2-port    code 8  = unsigned integer 16;
option wetatronics.ssh-port        code 9  = unsigned integer 16;
option wetatronics.ssh-disable     code 10 = unsigned integer 8;
option wetatronics.power           code 11 = unsigned integer 8;
option wetatronics.group-id        code 12 = text;
option wetatronics.persist         code 13 = unsigned integer 8;
option wetatronics.break-detect    code 14 = unsigned integer 8;

class "wetatronics" {
    match if option vendor-class-identifier = "wetatronics";
}

pool {
    allow members of "wetatronics";
    vendor-option-space wetatronics;
    option wetatronics.portal-url "portal.example.com:8443/ws/device/";
    option wetatronics.autoconnect 1;
    option wetatronics.group-id "building-7";
}
```

Windows DHCP Server:

- Create a Vendor Class named "wetatronics"
- Define vendor options 1–14 under that class with the appropriate types (string, uint8, uint16, uint32)
- Assign to the relevant scope or reservation

General Guidance:

- The device identifies itself with Option 60 = `wetatronics`
- Your DHCP server must match this vendor class to return Option 43
- Sub-options use standard TLV (Type-Length-Value) encoding within Option 43
- Option 160 can be used as a simpler alternative for just the portal URL
- Only send the sub-options you need -- unset options keep their current values

3.5.4 Verifying DHCP Vendor Options

To verify that vendor options were received and applied:

```
conxsfp> show network
  ...
    DHCP: Enabled
    DHCP Vendor Opts: Enabled
    ...
    DHCP opt43: 1,2,12
    Group ID: building-7
```

The `DHCP opt43` line lists which sub-options were received from the server. The `Group ID` shows the deployment group assigned to this device.

---

4. SSH Access

4.1 Standard SSH Connection

Connect to the ConX-SFP CLI using any SSH client:

```
ssh weta@192.168.0.232
```

Supported SSH Features:

Feature	Support
Protocol Version	SSH-2.0
Key Exchange	Curve25519-SHA256
Host Key	Ed25519
Encryption	AES-256-CTR
MAC	HMAC-SHA256
Password Auth	Yes
Public Key Auth	Yes

4.2 Direct Serial Passthrough via SSH

Connect directly to a serial port by appending the port number (and optional baud rate) to the username:

Syntax:

```
ssh username:port[:baudrate]@hostname
```

Examples:

```
# Connect to Serial Port 1 at default baud rate
ssh weta:1@192.168.0.232

# Connect to Serial Port 1 at 115200 baud
ssh weta:1:115200@192.168.0.232

# Connect to Serial Port 2 at 9600 baud
ssh weta:2:9600@192.168.0.232
```

When connected in passthrough mode, all keyboard input is sent directly to the serial device, and all serial output is displayed in your terminal.

4.3 Exiting SSH Serial Passthrough

To exit serial passthrough mode and return to the CLI (or disconnect), use the break sequence:

```
Press: Ctrl+]  Ctrl+]  (press Ctrl+] twice within 1 second)
```

After entering the break sequence, you will see:

```
** Break sequence detected - exiting passthrough **
conxsfp>
```

4.4 SSH Connection Limits

The ConX-SFP supports a maximum of 2 simultaneous SSH connections. Each connection is independent with its own session state.

4.5 SSH ControlMaster Limitation

The ConX-SFP supports one SSH channel per connection. SSH clients using ControlMaster (connection multiplexing) will experience issues with additional sessions.

Symptoms: - First SSH connection works normally - Second session shows: `mux_client_request_session: session request failed`

Solutions:

1. Disable ControlMaster for ConX-SFP in `~/.ssh/config` : `Host 192.168.0.232 ControlMaster no`

2. Override on command line: `bash ssh -o ControlMaster=no weta@192.168.0.232`
3. Close master connection first: `bash ssh -O exit weta@192.168.0.232 ssh weta@192.168.0.232`

4.6 SSH Public Key Authentication

The ConX-SFP supports Ed25519 public key authentication.

To view configured keys:

```
conxsfp> show keys
Host Keys:
  Ed25519: Configured

User Keys:
  weta: 1 key configured
```

Note: Public key upload requires administrative access and is typically performed during initial provisioning.

---

5. Telnet Access

5.1 Telnet CLI Access

Connect to the CLI via Telnet on port 23:

```
telnet 192.168.0.232
```

You will be prompted for username and password, then presented with the CLI:

```
Trying 192.168.0.232...
Connected to 192.168.0.232.

ConX-SFP Console
Login: weta
Password: ********

conxsfp>
```

> Security Warning: Telnet transmits credentials in plaintext. Use SSH whenever possible for secure access.

5.2 Telnet Direct Serial Access

Connect directly to serial ports via dedicated TCP ports:

Serial Port	TCP Port	Command
Serial 1	3001	`telnet 192.168.0.232 3001`
Serial 2	3002	`telnet 192.168.0.232 3002`

Example:

```
telnet 192.168.0.232 3001
```

This provides a direct, unauthenticated connection to the serial port. All data is passed through transparently.

To exit: Use your local telnet client's escape sequence (typically `Ctrl+]` then `q`).

5.3 Telnet Service Configuration

Enable or disable Telnet services via the CLI:

```
# Disable all Telnet services
conxsfp> config telnet disable

# Enable all Telnet services
conxsfp> config telnet enable

# Enable/disable CLI service
conxsfp> config telnet cli enable
conxsfp> config telnet cli disable

# Enable/disable Serial 1 passthrough
conxsfp> config telnet serial1-enable
conxsfp> config telnet serial1-disable

# Enable/disable Serial 2 passthrough
conxsfp> config telnet serial2-enable
conxsfp> config telnet serial2-disable

# Change Telnet CLI port
conxsfp> config telnet port 2323
```

View Telnet configuration:

```
conxsfp> show telnet
Telnet Services:
  CLI (port 23): Enabled
  Serial 1 (port 3001): Enabled
  Serial 2 (port 3002): Enabled
```

6. Serial Port Passthrough

6.1 Serial Port Overview

The ConX-SFP provides two RS-232 serial ports via a single RJ45 connector:

Port	Default Baud	Use Case
Serial 1	9600 N81	Primary console port
Serial 2	9600 N81	Secondary console port

6.2 Accessing Serial Ports

There are four methods to access serial ports:

Method 1: CLI Serial Command (SSH/Telnet)

```
conxsfp> serial 1
Entering serial passthrough mode (Serial 1 at 9600 baud)
Exit with break sequence: Ctrl+] Ctrl+]
```

Method 2: SSH Direct Passthrough

```
ssh weta:1:115200@192.168.0.232
```

Method 3: Telnet Direct Port

```
telnet 192.168.0.232 3001
```

Method 4: Serial Console CLI (Hardware Break)

Connect via serial console cable and send a hardware break signal to access the CLI directly through the serial port.

6.3 Serial Port Configuration

View current configuration:

```
conxsfp> show serial
Serial Port Configuration:
  Port 1:
    Baud Rate: 9600
    Data Bits: 8
    Stop Bits: 1
    Parity: None
    Flow Control: None

  Port 2:
    Baud Rate: 9600
    Data Bits: 8
    Stop Bits: 1
    Parity: None
    Flow Control: None
```

Configure serial port parameters:

```
# Set baud rate for Serial Port 1
conxsfp> config serial 1 baud 115200

# Full configuration example
conxsfp> config serial 1 baud 9600
conxsfp> config serial 1 data-bits 8
conxsfp> config serial 1 stop-bits 1
conxsfp> config serial 1 parity none

# Save changes
conxsfp> system save
```

Supported Baud Rates: 300, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400, 460800, 921600

6.4 Break Sequence Detection

When in serial passthrough mode, the ConX-SFP monitors for a configurable break sequence to exit passthrough and return to the CLI.

Default Break Sequence: `Ctrl+]` pressed twice within 1 second

How it works:

1. Enter passthrough mode (`serial 1` or direct SSH)
2. Interact with the connected device normally
3. Press `Ctrl+]` twice quickly to exit
4. Return to CLI prompt

> Note: The break sequence detection has a 1-second timeout. If you press `Ctrl+]` once and wait more than 1 second, the sequence resets.

6.5 Hardware Break Detection (Serial CLI)

When accessing the ConX-SFP through a physical serial console connection, you can activate the CLI by sending a hardware break signal.

Using Minicom:

```
Press: Ctrl+A  F
```

Using PuTTY:

```
Right-click → Special Command → Break
```

Using screen:

```
Press: Ctrl+A  Ctrl+B
```

After sending the break, you'll see:

```
** Hardware Break - CLI Active **
Type 'help' for commands, 'exit' to quit
serial-cli>
```

This method is useful for initial configuration when network access is not yet available.

---

7. Command Line Interface

7.1 CLI Overview

The ConX-SFP CLI provides a hierarchical command structure for device management. Commands follow a strict 2-tier hierarchy: `<primary> [secondary] [arguments]`.

Command Categories:

Category	Purpose
`show`	Display configuration and status
`config`	Modify settings
`system`	System operations (save, restart)
`serial`	Enter serial passthrough
`help`	Display help information

User Access Levels:

The CLI supports two access levels that control which commands are available:

Level	Access	Description
USER	Restricted	Can use: `help`, `exit`, `serial` (passthrough) only
ADMIN	Full	All CLI commands available

The default `weta` user has ADMIN access. See Section 7.6 for details on user level management.

7.2 Command Syntax

Commands follow this general pattern:

```
<category> <subcategory> [parameters]
```

Examples:

```
show network
config network dhcp off
config serial 1 baud 115200
system save
```

7.3 Command Abbreviation (Prefix Matching)

The CLI has built-in prefix matching. Any unique prefix of a command works automatically:

Abbreviation	Full Command
`sh net`	`show network`
`sh stat`	`show status`
`c n dhcp off`	`config network dhcp off`
`h`	`help`
`q` or `ex`	`exit`

Error Messages:

If you enter an invalid subcommand, the CLI provides helpful feedback:

```
conxsfp> show invalid
'invalid' is not a valid 'show' subcommand
Type 'show ?' for available subcommands.
```

7.4 Getting Help

The CLI provides a tiered help system that organizes commands by category:

Top-level overview:

```
conxsfp> help
ConXSFP CLI - Commands:

Session:
  help                    Use 'help help'
  exit                  - Exit CLI session

Show:
  show                    Use 'help show'

Config:
  config                  Use 'help config'

System:
  system                  Use 'help system'

Network:
  ping                  - ICMP echo test
  dns                   - DNS lookup
  ...

Type 'help <command>' for subcommands.
```

View subcommands for a category:

```
conxsfp> help show
'show' commands:
  show     status      - System status overview
  show     network     - Network configuration
  show     serial      - Serial port configuration
  show     users       - List configured users
  show     cert        - Show TLS certificates
  show     portal      - Portal status
  ...

Type 'show <subcommand> ?' for detailed help.
```

View detailed help for a specific command:

```
conxsfp> help config network
config network - Network configuration
Usage: config network <dhcp|static|hostname|ssh-port|dns> [args]

  dhcp [on|off]             - Configure DHCP
  static <ip> <mask> <gw>   - Set static IP
  hostname <name>           - Set hostname
  ...
```

> Note: USER level accounts will only see commands they have access to (help, exit, serial).

Context-Sensitive Help with `?`:

You can also append `?` to any command for context-sensitive help:

```
conxsfp> show ?
Available 'show' subcommands:
  show status        - System status overview
  show network       - Network configuration
  ...
```

7.5 Session Timeout

CLI sessions have a configurable idle timeout (default: 300 seconds).

Configure timeout:

```
conxsfp> config system timeout 600
Session timeout set to 600 seconds
```

Valid range: 30-3600 seconds

7.6 User Access Levels

The ConX-SFP implements a two-level access control system to manage user privileges.

Access Levels

Level	Value	Permissions
USER	0	Restricted access: `help`, `exit`, `serial` passthrough only
ADMIN	1	Full access to all CLI commands

Default Behavior

- The default `weta` user has ADMIN access
- New users are created with USER level by default
- Serial console CLI always has ADMIN access (physical access = trusted)

Managing User Levels

View user levels:

```
conxsfp> show users
Configured Users:
  weta (admin)
  operator (user)
```

Add a new admin user:

```
conxsfp> config user add newadmin password123 admin
User 'newadmin' added (admin)
```

Add a new restricted user:

```
conxsfp> config user add operator password123
User 'operator' added (user)
```

Change an existing user's level:

```
conxsfp> config user level operator admin
User 'operator' level changed to admin
```

Admin Protection Rules

To prevent accidental lockout, the following protection rules are enforced:

1. Cannot delete the last admin user – At least one admin must always exist
2. Cannot demote the last admin user – Must have at least one admin
3. Cannot change your own access level – Prevents self-demotion

Example:

```
conxsfp> config user level weta user
Error: Cannot change your own access level
```

USER Level Experience

When logged in as a USER level account, only basic commands are available:

```
operator@conxsfp> help
Available commands:
  help        - Show this help
  exit        - Exit CLI session
  serial      - Enter serial passthrough

operator@conxsfp> show status
Error: Permission denied
```

8. Cloud Connectivity

8.1 Wetatronics Cloud Platform

The ConX-SFP includes secure connectivity to the Wetatronics cloud management platform, enabling:

- Remote Device Management: Monitor and configure devices from anywhere
- Centralized Dashboard: View all your console servers in one place
- Firmware Updates: Receive and deploy firmware updates remotely
- Monitoring and Alerts: Get notified of device status changes

8.2 Security Standards

All cloud communications use industry-standard security:

Component	Standard
Protocol	TLS 1.3
Key Exchange	ECDHE with P-384 curves
Cipher Suite	TLS_AES_128_GCM_SHA256
Certificate Validation	X.509 chain verification with SNI
Session Resumption	TLS 1.3 session tickets for faster reconnection

8.3 Certificate Management

The ConX-SFP includes pre-installed certificates for secure cloud connectivity.

View stored certificates:

```
conxsfp> show cert
Certificate Storage:
  Bank 0: Wetatronics Root CA (valid)
  Bank 1: Empty
  Bank 2: Empty
```

View certificate details:

```
conxsfp> show cert detail 0
Certificate Details (Bank 0):
  Subject: CN=Wetatronics Root CA
  Issuer: Self-signed
  Valid From: 2024-01-01
  Valid To: 2034-01-01
  Fingerprint: 96:BC:EC:06:26:49...
```

8.4 Cloud CLI Commands

View connection status:

```
conxsfp> portal status
Portal Status:
  State: Connected
  URL: wss://devices.wetatronics.com:443/ws/device/
  Uptime: 2h 15m
  Messages: TX=45 RX=38
```

Manual connection control:

```
conxsfp> portal connect          # Connect to cloud platform
conxsfp> portal disconnect       # Disconnect from cloud
```

Claim device via web portal:

The `portal claim` command links your device to your account on the Wetatronics cloud portal:

1. Log in to the cloud portal and start the "Add Device" workflow
2. A 6-character claim code is displayed on the portal
3. Enter the code on the device:

```
conxsfp> portal claim ABC123
Sending claim request...
```

To check current claim status:

```
conxsfp> portal claim
Claim status: Claimed (owner: user@example.com)
```

Clear TLS session (troubleshooting):

```
conxsfp> portal clear-session
TLS session ticket cleared
```

This forces a full TLS handshake on the next connection, useful if auto-reconnect fails after a server restart.

DNS resolution:

```
conxsfp> show dns             # Show cached DNS entries
conxsfp> dns example.com      # Perform DNS lookup
conxsfp> clear dns            # Clear DNS cache
```

8.5 Cloud Platform Access

Access the Wetatronics cloud platform at:

https://cloud.wetatronics.com

Contact your Wetatronics representative for account setup and device registration.

---

9. Troubleshooting

9.1 General Connectivity Issues

> First Step: Always start troubleshooting by testing basic network connectivity with ping. From another device on your network, ping the ConX-SFP's IP address. If ping fails, the issue is at the network layer (cabling, VLAN, IP configuration). If ping succeeds but services don't respond, the issue is at the application layer.

If you don't know the IP address: - Check your DHCP server's lease table for the device hostname - Use LLDP on your network switch to discover the device (see Section 3.4) - Access the device via serial console and run `show network` - Try the fallback IP address: 192.168.0.232

9.2 Cannot Connect via SSH

Symptoms: SSH connection refused or timeout

Check: 1. Ping the device first to verify basic network connectivity 2. Verify IP address is correct (`show network` via serial console) 3. Confirm SSH service is running (port 22) 4. Verify firewall rules allow SSH traffic 5. Check that the device has completed booting (wait 2–3 seconds after power-on)

9.3 Cannot Connect via Telnet

Symptoms: Telnet connection refused

Check: 1. Ping the device first to verify basic network connectivity 2. Verify Telnet service is enabled: `show telnet` 3. Enable if disabled: `config telnet enable` 4. Check correct port number (default: 23)

9.4 Serial Passthrough Not Working

Symptoms: No data when connected to serial port

Check: 1. Verify baud rate matches target device: `show serial` 2. Check cable connectivity (see Appendix B for pinout) 3. Confirm target device console is active 4. Try loopback test (connect TX to RX on cable)

9.5 Break Sequence Not Detected

Symptoms: Cannot exit serial passthrough mode

Check: 1. Ensure you're pressing `Ctrl+]` (not `Ctrl+\` or other keys) 2. Press the sequence twice within 1 second 3. Verify terminal is not intercepting the sequence 4. Alternative: Close the SSH/Telnet connection entirely

9.6 Configuration Not Saved

Symptoms: Settings reset after reboot

Check: 1. Ensure you ran `system save` after making changes 2. Verify save confirmation message appeared 3. Check for errors in the save confirmation

9.7 Cloud Connection Issues

Symptoms: Cannot connect to Wetatronics cloud platform

Check: 1. Verify network connectivity and DNS resolution 2. Check certificate validity: `show cert` 3. Ensure firewall allows outbound HTTPS (port 443) 4. Contact Wetatronics support if issues persist

9.8 Firmware Update Fails

Symptoms: HTTP download errors, flash failures, or XMODEM errors

Check for HTTP updates (`update main remote` , `update boot remote`): 1. Verify the firmware server is running and accessible 2. Test connectivity: `connect <server-ip> <port>` or `ping <server-ip>` 3. Verify firmware file is ELF format (.bin / .elf) 4. Check `update status` for error details 5. Try `update main local` for browser-based upload instead

Check for XMODEM recovery (emergency bootloader): 1. Check serial connection is stable at 9600 baud 2. Use XMODEM-1K mode (not XMODEM-CRC or YMODEM) 3. Ensure terminal emulator is not sending extra characters 4. Try verify-only mode first to test transfer

---

10. Frequently Asked Questions

General

Q: What is the default IP address?

A: The ConX-SFP uses DHCP by default. If DHCP fails, it falls back to 192.168.0.232/24. Check your DHCP server's lease table or access via serial console to determine the assigned address.

Q: What are the default credentials?

A: Username: `weta` , Password: `wetatronics` . You should change these after initial setup for security.

Q: How many simultaneous connections are supported?

A: The ConX-SFP supports: - 2 simultaneous SSH connections - 1 Telnet CLI session - 2 Telnet serial passthrough sessions (one per port)

Q: Can I use both serial ports simultaneously?

A: Yes, Serial Port 1 and Serial Port 2 can be used independently and simultaneously by different connections.

SSH

Q: What SSH clients are compatible?

A: Any SSH-2.0 compatible client works, including: - OpenSSH (Linux, macOS, Windows 10+) - PuTTY (Windows) - Termius (iOS, Android) - SSH apps on mobile devices

---

Q: Why do I get "Host key verification failed"?

A: This occurs when connecting to a device whose key has changed (e.g., after firmware reflash). Remove the old key:

```
ssh-keygen -R 192.168.0.232
```

Q: Can I use SSH keys instead of passwords?

A: Yes, Ed25519 public key authentication is supported. Contact your administrator for key provisioning.

Serial Ports

Q: What cable do I need?

A: A standard Ethernet cable (straight-through) works with Cisco-style console ports. For other devices, you may need a rollover (console) cable. See Appendix B for pinout details.

Q: What is the maximum baud rate?

A: The ConX-SFP supports up to 921600 baud. However, reliability at very high speeds depends on cable quality and length.

Q: Why am I seeing garbled characters?

A: This usually indicates a baud rate mismatch. Check the serial port configuration matches your target device:

```
show serial
config serial 1 baud 115200
```

Q: How do I use hardware flow control?

A: Hardware flow control (RTS/CTS) requires using Serial Port 1 only, which repurposes Serial Port 2 pins for flow control. Contact support for configuration details.

Network

Q: Can I use VLANs?

A: The ConX-SFP operates on the access VLAN configured on its switch port. It does not support 802.1Q VLAN tagging internally.

Q: Does it support IPv6?

A: Currently, only IPv4 is supported. IPv6 support is planned for a future release.

Firmware Updates

Q: How do I know what firmware version I'm running?

A: Use the `show status` command:

```
conxsfp> show status
System Status:
  Firmware: 1.1.0.2
  Uptime: 2d 5h 30m
  ...
```

Q: Can I downgrade firmware?

A: Yes, any valid firmware image can be flashed regardless of version. However, configuration compatibility is not guaranteed when downgrading.

Q: What happens if power is lost during update?

A: The multi-stage boot system provides recovery. If the main application is corrupted, the bootloader can recover the device. The emergency bootloader can recover all other components. See Appendix D for details.

Security

Q: Is Telnet secure?

A: No, Telnet transmits all data including passwords in plaintext. Use SSH whenever possible. Telnet is provided for legacy compatibility only.

Q: How are passwords stored?

A: Passwords are cryptographically hashed with a random salt. The actual password is never stored.

Q: Can I disable Telnet entirely?

A: Yes:

```
config telnet disable
system save
```

Q: What encryption is used for SSH?

A: SSH uses: - Key Exchange: Curve25519-SHA256 (ECDH) - Encryption: AES-256-CTR - Integrity: HMAC-SHA256 - Host Key: Ed25519

Q: What are user access levels?

A: The ConX-SFP has two access levels: - USER (0): Restricted access - can only use `help`, `exit`, and `serial` passthrough - ADMIN (1): Full access to all CLI commands

The default `weta` user has ADMIN access. New users are created with USER level by default. See Section 7.6 for details.

Q: How do I create a restricted user account?

A: Create a user without specifying `admin`:

```
conxsfp> config user add operator password123
User 'operator' added (user)
conxsfp> system save
```

The user can only access serial passthrough, help, and exit commands.

Appendix A: CLI Command Reference

> Complete Reference: See ConX-SFP-CLI-Reference.md for the full CLI command documentation.

Quick Reference

Session Commands: - `help` - Show top-level command overview - `help <command>` - Show subcommands (e.g., `help show`, `help config`) - `help <command> <subcommand>` - Detailed help for specific command - `?` - Alias for help; append to any command for context-sensitive help - `exit` - Exit CLI session

Show Commands (commonly used): - `show status` - System status including Serial CLI state and break detection count - `show network` - IP configuration, DHCP status, DNS - `show serial` - Serial port configuration - `show serial-cli` - Serial CLI status, break detection, and active session info - `show users` - Configured users with access levels - `show telnet` - Telnet service status - `show dns` - Cached DNS entries - `show portal` - Portal/cloud connection status - `show cert` - TLS certificates

Config Commands (commonly used): - `config network dhcp [on|off]` - Enable/disable DHCP - `config network static <ip> <mask> <gw>` - Set static IP - `config serial <1|2> baud <rate>` - Set serial baud rate (300-921600) - `config user add <user> <pass> [admin|user]` - Add new user - `config system break-detection <on|off>` - Enable/disable break detection

System Commands: - `system save` - Save configuration to flash - `system restart` - Restart device

Network & Cloud Tools: - `ping <ip>` - ICMP ping test - `dns <hostname>` - DNS lookup - `serial <1|2>` - Enter serial passthrough mode - `portal connect` / `portal disconnect` - Cloud connection control - `portal status` - Cloud connection status - `portal claim <code>` - Claim device using portal code - `clear session <0-1>` - Disconnect an SSH session (admin) - `clear serial-cli` - Force exit active serial CLI session (admin) - `clear dns` - Clear DNS cache

Update Commands: - `update main local` - Reboot to bootloader upload mode - `update main remote <URL>` - Preflight check + bootloader fetch - `update boot local [port]` - Start bootloader upload server - `update boot remote <URL>` - Download + flash bootloader - `update status` - Show update progress - `update abort` - Abort update or stop server

For detailed documentation of all 65+ commands, subcommands, and options, see the complete CLI Reference document.

Appendix B: Hardware Pinout

RJ45 Connector Pinout

The ConX-SFP RJ45 connector supports either: - One serial port with hardware flow control, or - Two serial ports without hardware flow control

Looking at the RJ45 socket on the end of the SFP (label facing up):

Pin	Function (Dual Port)	Function (Single Port + Flow)
1	RX2 – Receive Data Serial 2	CTS – Clear To Send
2	Reserved I/O	Reserved I/O
3	RX1 – Receive Data Serial 1	RX – Receive Data
4	GND – Ground	GND – Ground
5	GND – Ground	GND – Ground
6	TX1 – Transmit Data Serial 1	TX – Transmit Data
7	PWR – External Power Output	PWR – External Power Output
8	TX2 – Transmit Data Serial 2	RTS – Request To Send

External Power Output (Pin 7)

The ConX-SFP provides a configurable external power output on pin 7 of the RJ45 connector. This feature is intended to support future expansion modules.

Specifications: – Maximum current: 90mA – State: Enabled or disabled via CLI – Default: Disabled

Enable/disable via CLI:

```
# Enable external power output
conxsfp> config system power on
External power output enabled

# Disable external power output
conxsfp> config system power off
External power output disabled

# Check current status
conxsfp> show system
...
External Power: Enabled
...
```

> Note: Do not exceed the 90mA current limit on pin 7 as this may damage the device.

Cable Requirements

For Cisco/Juniper/Arista Console Ports: - Use a standard straight-through Ethernet cable - Cisco console ports use the same pinout natively

For Other Devices: - Use a rollover (console) cable, or - Use a Y-splitter cable to break out both serial ports

Y-Splitter Cable Wiring

For accessing both serial ports simultaneously:

```
ConX-SFP                  Serial 1 RJ45        Serial 2 RJ45
RJ45                      (DB9 Adapter)        (DB9 Adapter)
________                  _____________        _____________
Pin 1 (RX2) ──────────────────────────────────── Pin 3 (RX)
Pin 3 (RX1) ──────────────── Pin 3 (RX)
Pin 4 (GND) ──────────────── Pin 5 (GND) ──────── Pin 5 (GND)
Pin 5 (GND) ──────────────── Pin 5 (GND) ──────── Pin 5 (GND)
Pin 6 (TX1) ──────────────── Pin 6 (TX)
Pin 8 (TX2) ──────────────────────────────────── Pin 6 (TX)
```

Appendix C: Technical Specifications

Network Specifications

Specification	Value
Ethernet Speed	1 Gbps
IP Addressing	IPv4, DHCP or Static
DNS	Dual DNS server support
SFP Form Factor	1000BASE-LX compatible

Serial Port Specifications

Specification	Value
Number of Ports	2
Voltage Levels	RS-232
Baud Rate Range	300 – 921,600 bps
Data Bits	5, 6, 7, or 8
Stop Bits	1 or 2
Parity	None, Odd, or Even
Flow Control	None, Hardware (single port mode only)

SSH Specifications

Specification	Value
Protocol Version	SSH-2.0
Key Exchange	Curve25519-SHA256
Host Key	Ed25519
Encryption	AES-256-CTR
MAC	HMAC-SHA256
Max Connections	2

TLS Specifications

Specification	Value
Protocol Version	TLS 1.3
Key Exchange	ECDHE (P-384)
Cipher Suite	TLS_AES_128_GCM_SHA256
Certificate Validation	X.509 chain verification
SNI	Supported
Session Resumption	TLS 1.3 session tickets

Environmental

Specification	Value
Operating Temperature	0°C to 50°C
Storage Temperature	-20°C to 70°C
Power	From SFP host port
Power Consumption	< 1W typical

Appendix D: Firmware Updates

D.1 Overview

The ConX-SFP includes a robust multi-stage boot system that enables firmware updates and recovery from failed updates. Multiple update methods are supported for flexibility.

Boot Components:

Component	Purpose
Boot Selector	Initial boot target selection and validation
Emergency Bootloader	Recovery mode with serial (XMODEM) update
Main Bootloader	Full-featured update with HTTP/ELF support
Main Application	Normal device operation

Supported Firmware Format:

Format	Extension	Description
ELF	.elf / .bin	Native RISC-V executable format, streaming support

Update Methods Summary:

Method	Command	Protocol	Use Case
Web Interface	`update main local`	HTTP Server (bootloader)	Manual upload via browser
HTTP Fetch	`update main remote <URL>`	HTTP Client (bootloader)	Pull from network server
Bootloader Upload	`update boot local [port]`	HTTP Server (main app)	Upload bootloader via browser
Bootloader Fetch	`update boot remote <URL>`	HTTP Client (main app)	Pull bootloader from server
XMODEM	`eloader`	Serial	Recovery when network unavailable

D.2 Accessing Bootloaders from CLI

Reboot to Main Bootloader (for browser upload):

```
conxsfp> update main local
Rebooting into bootloader update mode...
After reboot, open browser to http://<device-ip>/ to upload firmware
```

Reboot to Main Bootloader (fetch from server):

```
conxsfp> update main remote http://192.168.1.100:8080/firmware.bin
Pre-flight check passed:
  Host: 192.168.1.100
  Port: 8080
  File: /firmware.bin
Jumping to bootloader...
```

Jump to Emergency Bootloader:

```
conxsfp> eloader
Jumping to emergency bootloader...
```

> Note: After exiting a bootloader, you must restart the device to return to normal operation.

D.3 Boot Selector Commands

During the 10-second boot window, connect via serial at 9600 baud and use these commands:

Key	Action
`b`	Boot to Main Bootloader
`m`	Boot to Main Application
`e`	Boot to Emergency Bootloader
`s`	Show boot status and versions
`h`	Display help

Example boot status output:

```
ConX-SFP Boot Selector v1.0.0
Validation:
  Boot Selector: v1.0.0 [OK]
  Emergency Loader: v1.0.0 [OK]
  Main Application: v0.8.0 [OK]
  Main Bootloader: v1.0.0 [OK]

Press key within 10s: [b]ootloader [m]ain [e]mergency [s]tatus [h]elp
```

D.4 HTTP Web Interface (Main Bootloader)

The main bootloader provides a web-based firmware update interface.

Steps:

1. Reboot to bootloader (via `update main local` command or boot menu)
2. Open a web browser and navigate to `http://<device_ip>/`
3. Select firmware file (.bin / .elf format)
4. Click "Flash" to program, or "Verify" to validate only
5. Wait for completion (progress bar shows status)
6. Restart the device to boot new firmware

Bootloader CLI Commands:

```
# Show current configuration
> config

# Set static IP (if DHCP fails)
> setip 192.168.1.100

# Test network connectivity
> ping 192.168.1.1

# Boot to main application
> boot

# Reset device
> reset
```

D.5 HTTP Client Fetch (Main Bootloader)

The main bootloader can fetch firmware directly from an HTTP server. The `update main remote` command performs a DNS lookup and TCP connectivity check before rebooting into the bootloader.

From Main Application CLI:

```
conxsfp> update main remote http://192.168.1.100:8080/firmware.bin
Pre-flight check passed:
  Host: 192.168.1.100
  Port: 8080
  File: /firmware.bin

Device will reboot into bootloader to download and flash firmware.
Jumping to bootloader...
```

URL Format:

```
http://<host>[:port]/<path>/<filename>
```

Examples:

```
# Fetch from default HTTP port 80
conxsfp> update main remote http://192.168.1.100/firmware.bin

# Fetch from custom port with path
conxsfp> update main remote http://192.168.1.100:8080/fw/conxsfp.bin

# Using hostname (DNS resolved before reboot)
conxsfp> update main remote http://fw.example.com:8080/firmware.bin
```

Setting up an HTTP Server:

See section D.10 for the recommended fwserver tool, or use any HTTP server:

```
# Using fwserver (recommended - see D.10)
python3 -m fwserver --http-port 8080 --directory ./firmware

# Using Python built-in (quick testing, no logging)
cd /path/to/firmware
python3 -m http.server 8080

# Using nginx, Apache, or any HTTP server
# Place firmware files in web root
```

D.7 Bootloader Update — Remote Fetch (From Main Application)

The `update boot remote` command downloads and flashes bootloader firmware from an HTTP server while the device remains operational.

Important: This command protects the running application code. It can update: - Boot Selector - Emergency Bootloader - Main Bootloader

It cannot update the main application itself (use `update main` commands for that).

Commands:

```
# Download and flash bootloader firmware
conxsfp> update boot remote http://192.168.1.100:8080/bootloader.bin

# Check update status
conxsfp> update status

# Abort in-progress update
conxsfp> update abort
```

URL Format:

```
http://<host>[:port]/<path>/<filename>
```

Examples:

```
# Fetch from server
conxsfp> update boot remote http://192.168.1.100:8080/bootloader.bin

# Using hostname
conxsfp> update boot remote http://fw.example.com/bootloader.bin
```

D.8 Bootloader Update — Browser Upload (From Main Application)

The `update boot local` command starts an HTTP server that accepts firmware uploads via web browser. This provides the same web-based update experience as the main bootloader, but runs while the device is in normal operation.

Important: This command protects the running application code. It can update: - Boot Selector - Emergency Bootloader - Main Bootloader

It cannot update the main application itself (use `update main` commands for that).

Security: The upload server is disabled by default at every boot. It must be explicitly enabled via CLI and is intended for controlled maintenance windows only.

Commands:

```
# Start upload server on default port (8080)
conxsfp> update boot local
HTTP update server enabled on port 8080
Upload firmware via browser: http://<device-ip>:8080/

# Start upload server on custom port
conxsfp> update boot local 9090
HTTP update server enabled on port 9090
Upload firmware via browser: http://<device-ip>:9090/

# Stop the upload server
conxsfp> update abort
```

Using the Web Interface:

1. Start the upload server via `update boot local [port]`
2. Open a web browser and navigate to `http://<device_ip>:<port>/`
3. Select firmware file (.bin / .elf format)
4. Click "Flash" to program, or "Verify" to validate only

5. Wait for completion (progress bar shows upload status)
6. Stop the server when done: `update abort`

Web Interface Features: - Progress bar with percentage display - Flash and Verify buttons - Modal dialog showing success/failure result - Identical interface to main bootloader web UI

Upload Endpoints:

Endpoint	Method	Description
`/`	GET	Web interface with file upload form
`/flash`	POST	Upload and program firmware
`/verify`	POST	Upload and verify firmware (no flash)

Example using curl:

```
# Flash firmware via command line
curl -X POST --data-binary @bootloader.elf http://192.168.1.100:8080/flash

# Verify firmware without flashing
curl -X POST --data-binary @bootloader.elf http://192.168.1.100:8080/verify
```

D.9 XMODEM Firmware Update (Emergency Bootloader)

If the main bootloader is unavailable, use the emergency bootloader with XMODEM-1K protocol.

Steps:

1. Boot to Emergency Loader (press 'e' during boot or `eloader` from CLI)
2. Select option 1 for Flash (or 6 for Verify-only)
3. Initiate XMODEM-1K transfer from your terminal
4. Wait for completion

Emergency Bootloader Menu:

```
ConX-SFP Emergency Bootloader
1. Flash firmware (XMODEM)
2. Erase main bootloader
3. Boot main bootloader
4. System reset
6. Verify firmware (no flash)

Select option:
```

Using Minicom for XMODEM:

```
# Start minicom at 9600 baud
minicom -D /dev/ttyUSB0 -b 9600

# In emergency bootloader, press '1'
# Then: Ctrl+A  S  → select "xmodem" → select firmware.bin
```

Using lrzsz:

```
# Press '1' in emergency bootloader, then:
sx -k firmware.bin < /dev/ttyUSB0 > /dev/ttyUSB0
```

D.10 Testing Connectivity

Before initiating firmware downloads, you can test network connectivity.

Test TCP Connection (from main app):

```
conxsfp> connect 192.168.1.100 8080
Testing connection to 192.168.1.100:8080...
Connection successful
```

> Note: The `update main remote` command automatically performs DNS lookup and TCP connectivity checks before rebooting into the bootloader.

Test Ping (from bootloader):

```
> ping 192.168.1.100
Reply from 192.168.1.100: time=1ms
```

D.11 Firmware Server Tool (fwserver)

Wetatronics provides a simple firmware server tool that serves firmware files over both HTTP and TFTP protocols. No installation is required – just Python 3.9 or later.

Features: – Combined HTTP and TFTP server in one tool – Zero dependencies (uses Python standard library only) – Automatic logging with timestamps – Cross-platform (Windows, macOS, Linux)

Quick Start:

```
# 1. Create a firmware directory and copy your firmware files
mkdir firmware
cp boot/bin/bootloader.elf firmware/
cp bin/conxsfp.elf firmware/

# 2. Run the server (from the tools/fwserver directory)
cd tools/fwserver
python3 -m fwserver --http-port 8080 --directory ../../firmware

# Or run directly using uv (if available)
uv run fwserver --http-port 8080 --directory ../../firmware
```

Server Output:

```
============================================================
Firmware Server v1.0.0
============================================================
Firmware directory: /path/to/firmware
Available files: 2
  - bootloader.elf (59216 bytes)
  - conxsfp.elf (276324 bytes)
------------------------------------------------------------
HTTP | Starting server on all interfaces:8080
HTTP | Serving files from: /path/to/firmware
------------------------------------------------------------
Press 'q' or ESC to stop the server
------------------------------------------------------------
```

Common Usage Examples:

```
# HTTP only on port 8080 (most common)
python3 -m fwserver --http-port 8080 --directory ./firmware

# HTTP on port 80 (requires admin/root privileges)
sudo python3 -m fwserver --http-port 80 --directory ./firmware

# Both HTTP and TFTP servers
python3 -m fwserver --http-port 8080 --tftp --directory ./firmware

# TFTP only on non-privileged port
python3 -m fwserver --no-http --tftp --tftp-port 6969 --directory ./firmware

# Verbose mode with custom log directory
python3 -m fwserver --http-port 8080 --directory ./firmware -v --log-dir ./logs
```

Command-Line Options:

Option	Description	Default
`-d, --directory`	Firmware file directory	`./firmware`
`--http-port`	HTTP server port	`80`
`--http-host`	HTTP bind address	`0.0.0.0`
`--no-http`	Disable HTTP server	(enabled)
`--tftp`	Enable TFTP server	(disabled)
`--tftp-port`	TFTP server port	`69`
`--tftp-host`	TFTP bind address	`0.0.0.0`
`--log-dir`	Directory for log files	(current)
`-v, --verbose`	Enable debug logging	(off)
`-q, --quiet`	Suppress non-error output	(off)

Using with ConX-SFP:

Once the server is running, update firmware from the ConX-SFP:

```
# Update main application (reboots to bootloader for HTTP fetch)
conxsfp> update main remote http://192.168.1.100:8080/conxsfp.bin

# Update bootloader firmware (downloads from main app)
conxsfp> update boot remote http://192.168.1.100:8080/bootloader.bin
```

Server Logs:

The server creates timestamped log files in the log directory:

```
fwserver_20251205.log
```

Log entries show all file requests and transfers:

```
2025-12-05 10:30:15 | INFO    | HTTP | 192.168.1.50 | Serving: bootloader.elf (59216 bytes)
2025-12-05 10:30:18 | INFO    | HTTP | 192.168.1.50 | Completed: bootloader.elf (59216 bytes
sent)
```

Stopping the Server:

Press `q`, `ESC`, or `Ctrl+C` to stop the server gracefully.

D.12 Recovery Procedures

Scenario 1: Main Application Corrupted

1. Power cycle device
2. Wait for boot selector prompt
3. Press `b` to boot main bootloader
4. Use web interface to flash main application
5. Use `boot` command or let it auto-boot

Scenario 2: Main Bootloader Corrupted

1. Boot selector will detect invalid firmware
2. Device falls through to emergency bootloader automatically
3. Flash main bootloader via XMODEM
4. Test by pressing `3` to boot main bootloader

Scenario 3: Remote Bootloader Update (Device Accessible)

1. From main application CLI: `conxsfp> update boot remote http://192.168.1.100/bootloader.bin`
2. Wait for completion
3. Verify with `system restart` followed by boot to bootloader

Scenario 4: Complete Recovery (911 Mode)

> Warning: This is an emergency recovery procedure. Only use if specifically instructed by Wetatronics support.

1. Boot to emergency bootloader (must be intact)
2. Enter command sequence: `9`, then `1`, then `1`
3. Type `YES` when prompted to confirm
4. Flash boot selector via XMODEM
5. Device will auto-reset after successful flash

Document Revision History

Version	Date	Changes
2.1	2026-02-23	CLI Restructuring: All `ws` commands renamed to `portal` (Section 8.4, Appendix A); `bloader` commands replaced by `update main local|remote` and `update boot local|remote` (Appendix D); removed TFTP references; updated firmware format to ELF-only (.bin/.elf); `bridge` / `unbridge` unified to `portal serial <port> on|off`
2.0	2026-02-16	Code Sync & Branding: Standardized product name to ConX-SFP throughout; fixed `config telnet` commands (was incorrectly documented as `config system telnet`); updated `config user` subcommands to match code (`delete` not `del`, added `level`, removed `passwd` / `list`); added `clear session` command; fixed user creation output messages; updated password change workflow
1.9	2026-02-02	CLI Help & Cloud Commands: Updated Section 7.4 with tiered help system (`help`, `help <command>`, `help <cmd> <subcmd>`); added Section 8.4 documenting cloud CLI commands including `portal claim`, `portal clear-session`, `show dns`, `clear dns`; updated Appendix A quick reference
1.8	2026-01-27	TLS 1.3 Update: Updated TLS specifications from TLS 1.2 to TLS 1.3 with P-384 curves and session resumption support
1.7	2025-12-11	CLI Reference Split: Extracted Appendix A to separate ConX-SFP-CLI-Reference.md; added new `clear serial-cli` command for remote serial CLI session management; enhanced `show status` with Serial CLI state and break detection count; enhanced `show serial-cli` with detailed session information
1.6	2025-12-10	CLI v2 & User Levels: Added Section 7.6 documenting USER/ADMIN access levels; updated Section 7.3 with prefix matching and improved error messages; added context-sensitive help (`?` suffix) to Section 7.4; updated config user commands with level management; removed obsolete `show system` and `show build` commands (use `show status`); added user level FAQs
1.5	2025-12-07	Added section D.8 documenting HTTP Update Server for browser-based firmware uploads from main application; renumbered subsequent sections; updated Appendix A with updateserver CLI commands
1.4	2025-12-05	Added section D.10 documenting fwserver tool for HTTP/TFTP firmware distribution; updated D.5 and D.6 with fwserver references
1.3	2025-12-04	Major Appendix D rewrite: added ELF format support, HTTP/TFTP client fetch methods, system update from main application; updated Appendix A with new bootloader and system update CLI commands
1.2	2025-12-03	Added LLDP support section; enhanced troubleshooting with ping guidance; documented external power output (pin 7); updated pinout table
1.1	2025-12-03	Removed internal technical details; streamlined cloud section
1.0	2025-12-02	Initial release for firmware v0.7.0

ConX-SFP is a product of Wetatronics. For support, contact support@wetatronics.com