



Z H O N E™

For software version 1.14.1
Document Part Number: 830-01623-02
December 7, 2007

EtherXtend 3300 Series 1.14.1 Release Notes

This document describes the EtherXtend 1.14.1 release and includes the following sections:

- [Features, page 2](#)
- [Problems Fixed, page 6](#)
- [Known issues, page 6](#)
- [Upgrade the system software, page 7](#)
- [Create an EFM or N2Nbond group, page 9](#)
- [Configure a management interface on a WAN port, page 11](#)
- [Create EtherXtend bridges, page 13](#)
- [Downgrade the system software, page 14](#)
- [Back up and restore the system configuration, page 15](#)
- [Contact Global Service and Support, page 16](#)

This release note provides a summary of the new EtherXtend features using the feature default settings and related procedures.

For more detailed information and procedures, see the *EtherXtend User's Guide* or the *ZHONE CLI Reference Guide* at:

<http://zhone.com/support/manuals/>

For firmware downloads, see the Support download page at:

<http://zhone.com/support/downloads/>

Zhone Technologies
@Zhone Way
7001 Oakport Street
Oakland, CA 94621
USA

Copyright © 2007
Zhone Technologies, Inc.

The Zhone EtherXtend 4-port and 8-port devices are products in a family of next-generation Ethernet-enabled CPE devices. Each EtherXtend model performs the function of a network extender, an emerging class of products that lengthens the reach of packets passing from the source or destination to or from a service provider, a distance known as Ethernet in the First Mile (EFM). Zhone's EtherXtend focuses on providing high-availability links for business use that needs Ethernet over copper technology. By bonding two or more copper pairs, the EtherXtend provides several megabits of throughput for a fraction of the cost of a T1 or E1.

The firmware version for this release is 1.14.1.

This release uses ROM version 1.13.2.203 or later.



Note: For complete information about EtherXtend operations, see the *EtherXtend Users Guide*.

FEATURES

Features not supported in 1.14.1

No routing support

ZMS not supported

New features supported in 1.14.1

Bridging

EtherXtend now supports Transparent Lan Service (TLS).

TLS bridges are only used with other TLS bridges. TLS bridges learn MAC addresses and forward packets to learned destinations. Broadcasts and unknown unicasts are flooded out all interfaces except the ingress interface.

Packets that enter the system on a TLS interface have their source MAC addresses learned and associated with the interface so that frames from the network that come in on other TLS bridges in the VLAN can be sent to the correct interface.

VLAN support

The EtherXtend VLAN features include:

- VLAN insertion/removal
- VLAN Q-in-Q
- Mapping VLAN to LAN Ethernet ports; map VLAN A to port 1 and VLAN B to port 2.

Multiple bond groups

EtherXtend supports multiple bond groups. This allows EtherXtend bond groups to be connected to other EtherXtend bond groups that could be connected to a variety of Ethernet devices.

Ethernet OAM 802.3ah

Release 1.14.1 introduces OAM support on MALC-EFM-SHDSL-24 interfaces connected to CPEs including the EtherXtend.

EFM OAM uses an in-band link layer OAM packet exchange between MALC EFM interfaces and OAM capable CPEs. The EtherXtend functions as a remote peer to provide discovery, OAM pdu statistics, and peer information.

When OAM is configured on a MALC EFM interface in active mode, the discovery process initiates a search for a peer OAM-enabled EtherXtend configured in passive mode and physically connected to the MALC EFM port of that interface. If the discovery process does not find an EtherXtend peer, the discovery process begins again after a five second pause and continues until a peer OAM-enabled EtherXtend is found.

IPSLA for EtherXtend

The IP Service Level Agreement (IPSLA) feature in EtherXtend 1.14.1 allows service providers and network operators to enforce and monitor access network connections and performance. IPSLA uses ICMP ping messages over configured IPSLA paths to track Round Trip Times (RTTs) and ECHO REQs/RSPs between initiator and responder devices to determine network performance and delays. Typically, one initiator device is used to monitor other responder devices in the network. The EtherXtend can be both an initiator device or a responder device. A maximum of 32 IPSLA paths can be configured per MALC and 4 IPSLA paths per EtherXtend.

Initiator devices must be running IPSLA to request data for a responder device. Responder devices must be accessible through the ping command in the IP network, but do not need to run IPSLA. Responder devices not running IPSLA display limited statistical data and functionality.

802.3ah EFM bonding

Both Net to Net (N2N) bonding and IEEE 802.3ah EFM bonding are supported.

By default, all EtherXtend ports are configured with N2N bonding and can be re-configured for EFM bonding.

EtherXtend statistics

The EtherXtend 1.14.1 release provides the ability to view the following statistics:

- DSL port statistics as reported through the **dslstat** *[name/type]* command (as shown below).
 - received packets
 - received octets
 - received errored packets
 - packets discarded on receive
 - transmitted packets
 - transmitted octets
 - packets discarded on transmit
 - port uptime
 - port train rate
- Bond group statistics reported in the **bond stats** *[name/type]* command (as shown below).
 - received unicast packets
 - received multicast packets
 - received broadcast packets
 - received octets
 - received errored packets
 - packets discarded on receive
 - transmitted unicast packets
 - transmitted multicast packets
 - transmitted broadcast packets
 - transmitted octets
 - packets discarded on transmit
 - aggregate bandwidth of bond group
- Bond group up/down alarms/traps sent when first port joins group and last port leaves group.

To view DSL port statistics enter:

```

zSH> dslstat 1-1-1-0/shdsl
General Stats:
-----
AdminStatus.....UP
DslUpLineRate (bitsPerSec).....0
DslDownLineRate (bitsPerSec).....0
DslMaxAttainableUpLineRate (bitsPerSec).....569600
DslMaxAttainableDownLineRate (bitsPerSec)....569600
Out Octets.....0
Out Pkts/Cells.....0
Out Discards.....0
Out Errors.....0
In Octets.....0
In Pkts/Cells.....0
In Discards.....0
In Errors.....0

DSL Physical Stats:
-----
DslLineSnrMgn (tenths dB).....0
DslLineAtn (tenths dB).....0
DslCurrOutputPwr (tenths dB).....0
LOFS.....0
LOLS.....0
LOSS.....0
ESS.....5791
CRC Errors.....0
Inits.....0

```

To view bond statistics enter:

```
zSH> bond stats 1-1-202-0/n2nbond
```

```
***** Bond group statistics *****
```

Group Info

Slot	GrpId	Interface Name	IfIndex
1	202	1-1-202-0/n2nbond	21

AdminStatus	OperStatus	Bandwidth	Last Change
UP	DOWN	0	0.00:50:22

Group Members

Port	Interface Name	IfIndex
2	1-1-2-0/shdsl	11
3	1-1-3-0/shdsl	13
1	1-1-1-0/shdsl	9
4	1-1-4-0/shdsl	15

Statistics (Received)

Octets	0
Ucast	0

Mcast	0
Bcast	0
Discards	0
Errors	0
Statistics (Transmitted)	
Octets	0
Ucast	0
Mcast	0
Bcast	0
Discards	0

EtherXtend QoS

EtherXtend now provides these quality of service (QoS) features:

- 802.1p COS marking
- Prioritization based on 802.1p and TOS
- Four priority queues

Problems Fixed

- TR 60450** The IP conflict between EtherXtends that have a default IP on a wire bridge connection.

Known issues

This section describes known issues in this release. Known Issues with an asterisk (*) are new to this release.

During configuration, system messages may be reported. These messages may be useful for troubleshooting but should not affect operation of the system.

General

- ◇ * A configsync file is not sent by EtherXtend (ZMS is not supported).
- ◇ *After performing an onreboot restore, a dhcpclient error message appears.
- ◇ *When adding a default route, a dhcpclient error message appears.
- ◇ *On rare occasions, EtherXtend shows G.SHDSL lines and bond groups up when the MALC shows the lines are actually down.

- ◇ *On rare occasions, links between the SHDSL ports on the MALC and the Ethernet ports EtherXtend do not come up after a reboot. In this case, for the links to come up, perform disable, enable on those ports.
- ◇ *BPDUs from SHDSL crossing untagged TLS bridges flood all the Ethernet ports with VLAN tags.

Upgrade the system software

This section explains how to upgrade the EtherXtend software.



Note: Before upgrading to the 1.14.1 release, the EtherXtend must be running ROM 1.13.2.203 or later.

Upgrading the system software

Verify that the EtherXtend is running ROM version 1.13.2.203 or later:

```
zSH> slots 1
Type                : G.SHDSL ETHERXTEND
Card Version        : 18710106
EEPROM Version      : 2
Serial #            : 4151731
CLEI Code           : No CLEI
Card-Profile ID     : 1/1/7101
Shelf               : 1
Slot                : 1
ROM Version         : ETHX CAN 1.13.2.103
Software Version:   : ETHX CAN 1.13.2.110
State               : RUNNING
Mode                : NONE
Heartbeat check     : enabled
Longest hbeat       : 0
Fault reset         : enabled
Uptime              : 1 hour, 14 minutes
zSH>
```

To upgrade the system software:

- 1 Verify you are at the root of the flash card:

```
zSH> cd /card1
zSH> pwd
/card1/
```

- 2 If desired, save the configuration file to a host on the network.

To save the configuration file, backup the configuration file using the dump command. The dump command uses the following syntax:

```
dump [console] [file filename] [network host filename]
```

To save the configuration file locally:

```
zSH> dump file ethxshdsl.cfg.date
```

To save the configuration file on a remote host:

```
zSH> dump network 192.168.8.21 ethxshdsl.cfg
```

- 3 If desired, save the existing software images on the flash card by copying them to a different file name. For example:

```
zSH> copy ethxshdsl.bin ethxshdsl.bin.<ver>
```

Copy the rest of the software images in the same way.

- 4 Download the latest system image software for the device to the system flash using the **image** command. The command uses the following syntax:

```
image download 192.168.2.89 sourcefilename  
[destinationfilename]
```



Note: If you do not specify a *destinationfilename*, it will be the same as the *sourcefilename*.

For example:

```
zSH> image download 192.168.2.89 ethxshdsl.bin
```



Note: If a download error occurs check for and delete unnecessary files to regain flash space.

- 5 Use the **systemreboot** command to reboot the system with the new configuration.

The system reboots again with the new configuration. This may take up to 5 minutes to complete.

Performing local ROM upgrades



Note: Only perform a local ROM upgrade if the ROM is running software older than 1.13.2.203.

This procedure describes how to perform a local upgrade of EtherXtend ROM using the console connection.

- 1 Connect to the console port of the EtherXtend and log in to the device.
- 2 Use the **slots 1** command to verify the EtherXtend software and ROM versions to be upgraded

- 3 Configure an IP address on the EtherXtend and establish connectivity to the TFTP Server.
- 4 Download the ROM image from the TFTP server.

```
zSH> image download IPofTftpServer ethxshdslrom.bin
ethxshdslrom.bin
```

- 5 Use the **systemreboot** to reboot the EtherXtend
- 6 During the reboot type *zhone* when the message *Preparing to load full image...* appears. The message *Boot process interrupted* will appear.
- 7 Update the ROM using the **rom update ethxshdslrom.bin** command.

```
zSH> rom update ethxshdslrom.bin
WARNING: This operation will erase the boot flash
and overwrite it with the specified image.
Continue? (yes or no) [no] yes
Erasing boot flash... passed
Programming boot flash... passed
Verifying boot flash... passed
Resetting card...
Boot EthXtend Unit
Starting ethXShdslROM.
```

- 8 Wait 5 minutes for the EtherXtend to reboot.
- 9 Login to the EtherXtend.
- 10 Use the **slots 1** command to verify that the ROM Version has been updated.



Note: Zhone recommends that you back-up your configuration to a file with the **dump** command before upgrading software.

Create an EFM or N2Nbond group

Creating bond groups

Add a single N2N or EFM bond group and verify bond group.



Note: When using the **bond group add** commands, the bond group ID must be in the range of 25-99.

```
zSH> bond add group 1-1-50-0/n2nbond
```

```
zSH> bond add group 1-1-50-0/efmbond
```

To add a new member to an existing bond group:

```
zSH> bond add member 1-1-101-0/n2nbond 1-1-1-0/shdsl
```

To create a bond group with multiple members:

```
zSH> bond add member 1-1-50-0/n2nbond 1-1-3-0/shdsl
1-1-4-0/shdsl
```

Displaying bond groups

After the EtherXtend reboots and the SHDSL lines come back up, the bond groups can be displayed.

Bond groups can be displayed for all existing groups, a specific group, a specific slot, or link.

To display all configured bond groups:

```
zSH> bond show all
```

Bond Groups					
Slot	GrpId	Name	Type	State	
1	201	1-1-201-0	n2nbond	ACT	

To display a specific bond group:

```
zSH> bond show group 1-1-201-0/n2nbond
```

Bond Groups					
Slot	GrpId	Name	Type	State	
1	201	1-1-201-0	n2nbond	ACT	
Group Members					
Slot	Port	Name	Type	State	
1	1	1-1-1-0	shdsl	ACT	
1	2	1-1-2-0	shdsl	ACT	
1	3	1-1-3-0	shdsl	ACT	
1	4	1-1-4-0	shdsl	ACT	

To display bond group on slot 1:

```
zSH> bond show slot 1
```

Bond Groups					
Slot	GrpId	Name	Type	State	
1	201	1-1-201-0	n2nbond	ACT	

To display the bond link:

```
zSH> bond show link 1-1-3-0/shdsl
```

Bond Groups					
Slot	GrpId	Name	Type	State	
1	201	1-1-201-0	n2nbond	ACT	
Group Members					
Slot	Port	Name	Type	State	
1	1	1-1-1-0	shdsl	ACT	
1	2	1-1-2-0	shdsl	ACT	
1	3	1-1-3-0	shdsl	ACT	
1	4	1-1-4-0	shdsl	ACT	

Changing bond group type

Bond group type can be changed for individual bond groups or all bond groups used in a specified slot using **bond modify** command. Bond groups and bond members can be deleted with the **bond delete** command.

```
zSH> bond modify efm group 1-1-201-0/n2nbond
```

```
zSH> bond modify n2n group 1-1-101-0/efmbond
```

```
zSH> bond modify efm slot 1
```

```
zSH> bond delete group 1-1-201-0/n2nbond
```

```
zSH> bond delete member 1-1-201-0/n2nbond 1-1-3-0/shdsl
```

Configure a management interface on a WAN port

This section describes how to use the default management interface to remotely manage the EtherXtend and how to modify the default interface for management from a different subnetwork. By default, the AutoConfig interface (1-1-*bondgroupnumber-0/n2nbond*) is built on the default N2N bond group as a DHCP client. The bond group number may vary depending on which port is first to receive link.

Managing EtherXtend using the default management interface

The AutoConfig interface (1-1-*bondgroupnumber-0/n2nbond*) is built on the default N2N bond group as a DHCP client using VLAN 7.

The MALC that connects to the EtherXtend can be configured either as a local DHCP server or as bridge to an external DHCP server. Refer to the MALC documentation for detailed procedures for DHCP configuration.

To configure Telnet access from the MALC to the EtherXtend, use the **host add** command to configure a host connection between the MALC and the EtherXtend and the bond group associated with the management interface.

```
zSH> host add 1-3-201-0/n2nbond vlan 7 dynamic 1 1
```

You can now Telnet to the MALC and then Telnet to the EtherXtend device.

Use the **interface show** and **route show** commands on the EtherXtend to verify the default management interface:

```
zSH> route show
Destination Routing Table
Dest                Nexthop            Cost    Owner    Fallback
-----
0.0.0.0/0           10.250.1.254      1       STATICLOW
10.250.1.0/24       1/1/201/0/ip      1       LOCAL
```

```
zSH> interface show
1 interface
Interface          Status  Rd/Address          Media/Dest Address  IfName
-----
1/1/201/0/ip      UP      1 10.250.1.54/24    00:01:47:36:2e:03  AutoConfig
```

Managing EtherXtend using a non-default management interface

To manage the EtherXtend through an interface different than the default AutoConfig address, delete the default AutoConfigIP address and then add the desired interface. If the new IP address is not compatible with the address of the management PC, the connection to the device will be lost. To restore the connection, change the address of the management PC to be compatible with the device address.

The following example configures the IP address for the system:

- 1 Delete the AutoConfig address.

```
zSH> delete ip-interface-record AutoConfig/ip
```

- 2 Add the IP interface to the bond group.

```
zSH> interface add 1-1-50-0/n2nbond 172.24.200.133/24
Created ip-interface-record 1-1-(25-99)-0-n2nbond/ip
```



Note: The default interface is reset if a **set2default** is performed without the restore option.

Verifying the interface

Use the **interface show** command to verify that the IP interface was configured correctly:

```
zSH> interface show
1 interface
Interface          Status  Rd/Address          Media/Dest Address  IfName
-----
1/1/4/0/ip         UP      1 172.24.200.133/24  00:01:47:07:ef:ee  AutoConfig
```

Creating a default route

The following example creates a default route using the gateway 192.168.8.1 with a cost of 1 (one):

```
zSH> route add default 192.168.1.1
```

Verifying the route

Use the **route show** command to verify that the routes were added:

```
zSH> route show
Dest                               Nexthop           Cost   Owner
-----
0.0.0.0/0                          192.168.8.1      1      STATICLOW
192.168.8.1/24                     1/1/1/0/ip       1      LOCAL
```

Create EtherXtend bridges

This section describes how to create bridges for one or more Ethernet interfaces on the EtherXtend

Creating a bridge for a bond group

Create a bridge for the bond group.

- 1 Use the **bridge add** command to add a bridge, then create the bridge path with the **bridge-path add** command as follows

```
zSH> bridge add 1-1-201-0/n2nbond downlink vlan 1
Adding bridge on 1-1-201-0/n2nbond
Created bridge-interface-record 1-1-201-0-n2nbond/
bridge
```

```
zSH> bridge-path add 1-1-202-0-n2nbond/bridge vlan 1
default
Bridge-path added successfully
```

- 2 Use the **bridge add** command to create a bridge for one or more Ethernet interfaces on the EtherXtend, then use the **bridge-path add** command to create the bridge path for that interface:

```
zSH> bridge add 1-1-1-0/eth downlink vlan 1
Adding bridge on 1-1-1-0/eth
Created bridge-interface-record 1-1-1-0-eth/bridge
```

```
zSH> bridge-path add 1-1-1-0-eth/bridge vlan 1 default
Bridge-path added successfully
```

- 3 Use the **bridge show** command to view the bridges.

```

zSH> bridge show
Typ VLAN          Bridge
-----
dwn              1 1-1-201-0-n2nbond/bridge
                UP D 00:13:72:dd:51:89
                S VLAN 1 default [U: 3600
sec, M: 150 sec, I: 0 sec]
dwn              1 1-1-1-0-eth/bridge
                UP D 00:13:72:de:92:2e
                D 00:1a:6d:13:19:83

```

Downgrade the system software

Follow these steps to downgrade the system software.

Downgrading a system

For unplanned downgrades, contact Zhone GSS before performing the downgrade procedures.

- 1 Verify that you are at the root of the flash card:

```

zSH> cd /card1
zSH> pwd
/card1

```

- 2 Rename the saved configuration file from the previous release:

```

zSH> cd onreboot
zSH> copy restore.done restore

```

The restore file can be validated using the **restore validate <filename>** command.

- 3 Download the old system image software using the **image** command. The command uses the following syntax:

```

image download TFTPserverAddress sourcefilename
[destinationfilename]

```



Note: If you do not specify a *destinationfilename*, it will be the same as the *sourcefilename*.

For example:

```

zSH> image download TFTPserverAddress ethxshdsl.bin

```



Note: If a download error occurs check for and delete unnecessary files to regain flash space.

- 4 Use the **set2default** command to erase the system configuration:

```

zSH>set2default

```

```
Ok to reset to default (system will reboot) ? [yes] or
[no]: yes
```

During the reboot, the system will look for the file named **restore** in the **onreboot** directory, then use that file to recreate the configuration files in the **datastor** directory. After the restore is finished, it renames the **restore** file to **restore.done**. A system warning will appear if a restore file is not found.

Back up and restore the system configuration

This section explains how to backup and restore a EtherXtend system configuration.

Backing up and restoring the configuration

You can back up your configuration using the **dump** command. You can either save the configuration to the console, a local file, or the network.

The **dump** command uses the following syntax:

```
dump [ console ] [ file filename ] [ network host filename ]
```

To back up and restore the configuration to the network:

- 1 Create the file in the destination location of the TFTP server and make it writable.
- 2 Save the configuration. The following example saves the configuration to a file named *ethxshdsl.cfg* on the host 192.168.8.21:

```
zSH> dump network 192.168.8.21 ethxshdsl.cfg
```

- 3 On the flash card, create a directory named **onreboot**:

```
zSH> cd /card1
zSH> mkdir onreboot
```

- 4 Copy the previously saved dump file to the **onreboot** directory and rename it **restore**:

```
zSH>file download 192.168.2.89 dumpfilename onreboot/
restore
```

where **192.168.2.89** is the hostname the restore file was saved to and **dumpfilename** is the name of the restore file.

The restore file can be validated using the **restore validate <filename>** command.

- 5 Use the **set2default** command to erase the system configuration and perform a reboot running the 1.14.1 software:

```
zSH> set2default
No restore file (/card1/onreboot/restore) found.
Setting to default will result in an empty database.
Continue? (yes or no) [no]: yes
```

```
Ok to reset to default (system will reboot) ? [yes] or [no]: yes  
Do you want to exit from this request? (yes or no) [yes] no  
Are you sure? (yes or no) [no] yes
```



Note: Traffic and in-band management will be down until the EtherXtend bridging and IP interfaces are manually reconfigured.

During the reboot, the system will look for the file named **restore** in the **onreboot** directory, then use that file to recreate the configuration files in the **datastor** directory. After the restore is finished, it renames the **restore** file to **restore.done**. A system warning will appear if a restore file is not found.

Contact Global Service and Support

Contact Global Service and Support (GSS) if you have any questions about this or other Zhone products. Before contacting GSS, make sure you have the following information:

- Zhone product you are using
- System configuration
- Software version running on the system
- Description of the issue

Technical support

If you require assistance with the installation or operation of your product, or if you want to return a product for repair under warranty, contact GSS. The contact information is as follows:

E-mail	support@zhone.com
Telephone (North America)	877-ZHONE20
Telephone (International)	510-777-7133
Internet	www.zhone.com/support

If you purchased the product from an authorized dealer, distributor, Value Added Reseller (VAR), or third party, contact that supplier for technical assistance and warranty support.

Service requirements

If the product malfunctions, all repairs must be performed by the manufacturer or a Zhone-authorized agent. It is the responsibility of users requiring service to report the need for service to GSS.