



Z H O N E™

For software version 1.14.2.119
Document Part Number: 830-01623-05
June 26, 2008

EtherXtend 3300 Series 1.14.2.119 Release Notes

This document describes the EtherXtend 1.14.2.119 release and includes the following sections:

- [Features, page 2](#)
- [Problems fixed, page 3](#)
- [Known issues, page 3](#)
- [Upgrade the system software, page 3](#)
- [Create an EFM or N2Nbond group, page 6](#)
- [Configure a management interface on a WAN port, page 8](#)
- [Create EtherXtend bridges, page 10](#)
- [Downgrade the system software, page 14](#)
- [Back up and restore the system configuration, page 16](#)
- [Contact Global Service and Support, page 17](#)

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 © 2008
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 default bonding method for the EtherXtend 3300 is EFM bonding although you can also configure the EtherXtend with N2N bonding.



Note: The EtherXtend 3300 supports up to 2500 MAC addresses. Care should be taken not to exceed 2500 MAC addresses.

The firmware version for this release is 1.14.2.119.

This release uses ROM version 1.14.1.103 or later.

Zhone recommends that you leave the EtherXtend and the MALC in adaptive rate mode. If it is desired to run in a fixed rate mode, it is better to keep the EtherXtend in adaptive mode and configure the MALC in fixed rate mode.



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

FEATURES

New features supported in 1.14.2.119

EtherXtend auto-provisioning

EtherXtend auto-provisioning configures the EtherXtend as a DHCP client on VLAN 7 and creates an untagged and tagged (VLAN 0) TLS bridge on the SHDSL bond group (*1-1-99-0/efmbond*) and an untagged and tagged (VLAN 0) TLS bridge on the four LAN Ethernet ports. Ethernet frames can now pass between the TLS bridges.

When the EtherXtend is upgraded to an auto-provisioning version of the software from a version of the software not running auto-provisioning, the auto-provisioning feature is automatically activated. This means that all bridges configured when running software with auto-provisioning will be preserved and not overwritten after future upgrades.

Bridges configured before upgrading to software with the auto-provisioning feature will be deleted and overwritten with auto-provisioned bridges.

EtherXtend statistics

The EtherXtend 1.14.2.119 release supports statistics at the bond group level as well as statistics on a per line basis.

Moving PC cables on Ethernet ports

If a PC cable is moved from one Ethernet port to another on the EtherXtend, a **bridge flush** command may be given from the EtherXtend CLI to force re-learning the PC's MAC address on the new Ethernet port. If the **bridge flush** command is not given, the MAC address timeout is 3600 seconds.

Problems fixed

- ◇ BPDUs from SHDSL crossing untagged TLS bridges flood all the Ethernet ports with VLAN tags.

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

- ◇ After performing an onreboot restore, a dhcpclient error message appears.
- ◇ When adding a default route, a dhcpclient error message appears.

Upgrade the system software

This section explains how to upgrade the EtherXtend software.



Note: Before upgrading to the 1.14.2.119 release, the EtherXtend must be running ROM 1.14.1.103 or later.



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

Upgrading the system software

Verify that the EtherXtend is running ROM version 1.14.1.103 or later:

To upgrade the system software:

```
zSH> slots 1
Type           : G.SHDSL ETHERXTEND 3300
Card Version   : 1
EEPROM Version : 2
Serial #       : 12602766
CLEI Code      : No CLEI
Card-Profile ID : 1/1/7108
Shelf          : 1
Slot           : 1
ROM Version    : ETHX CAN 1.14.1.103
Software Version: ETHX 1.14.2.119
State          : RUNNING
Mode           : NONE
Heartbeat check : enabled
Longest hbeat  : 0
Fault reset    : enabled
Uptime         : 4 hours, 9 minutes
```

- 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 ethx3300.cfg.date
```

To save the configuration file on a remote host:

```
zSH> dump network 192.168.8.21 ethx3300.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 ethx3300.bin ethx3300.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 ethx3300.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: It is only necessary to perform a local ROM upgrade if the ROM is running software older than 1.14.1.103.

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 ethx3300rom.bin
ethx3300rom.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 ethx3300rom.bin** command.

```
zSH> rom update ethx3300rom.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 ethX3300ROM.
```

- 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.

Create an EFM or N2Nbond group

Creating bond groups

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



Note: Bond groups created with CLI commands must be greater than 24 and less than 100. 100-series bond group IDs are used by ZMS and 200-series bond groups are auto-provisioned/discovered.

```
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/shds1
```

To create a bond group with multiple members:

```
zSH> bond add member 1-1-50-0/n2nbond 1-1-3-0/shds1
1-1-4-0/shds1
```

Displaying bond groups

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	99	1-1-99-0	efmbond	ACT	

To display a specific bond group:

```
zSH> bond show group 1-1-99-0/efmbond
```

Bond Groups					
Slot	GrpId	Name	Type	State	
1	99	1-1-99-0	efmbond	ACT	

Slot	Port	Group Members		
		Name	Type	State
1	3	1-1-3-0	shdsl	ACT
1	2	1-1-2-0	shdsl	ACT
1	1	1-1-1-0	shdsl	ACT
1	8	1-1-8-0	shdsl	ACT
1	7	1-1-7-0	shdsl	ACT
1	6	1-1-6-0	shdsl	ACT
1	5	1-1-5-0	shdsl	ACT
1	4	1-1-4-0	shdsl	ACT

To display bond group on slot 1:

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

Slot	GrpId	Bond Groups		
		Name	Type	State
1	99	1-1-99-0	efmbond	ACT

To display the bond link:

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

Slot	GrpId	Bond Groups		
		Name	Type	State
1	99	1-1-99-0	efmbond	ACT
Group Members				
Slot	Port	Name	Type	State
1	3	1-1-3-0	shdsl	ACT
1	2	1-1-2-0	shdsl	ACT
1	1	1-1-1-0	shdsl	ACT
1	8	1-1-8-0	shdsl	ACT
1	7	1-1-7-0	shdsl	ACT
1	6	1-1-6-0	shdsl	ACT
1	5	1-1-5-0	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 is 1-1-99-0/efmbond and is built on the EFM 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, 1-1-99-0/efmbond, is built on the default EFM 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 1-1-99-0/efmbond 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       1/1/99/0/ip       1       LOCAL

```

```

zSH> interface show
1 interface
Interface          Status  Rd/Address          Media/Dest Address  IfName
-----
1/1/99/0/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 172.24.200.254 with a cost of 1 (one):

```
zSH> route add default 172.24.200.254 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.

EtherXtend auto-provisioning

Release 1.14.2.119 has an auto-provisioning feature that automatically re-configures the EtherXtend bridges and management IP address when the 1.14.2.119 software runs for the first time.

The auto-provisioning deletes any existing bridges and creates tagged (VLAN 0) and untagged TLS bridges on each of the LAN Ethernet ports and on the SHDSL bond group.

The auto-provisioning also deletes any existing IP addresses on the Ethernet ports or SHDSL bond group and creates a DHCP client on VLAN 7 for the SHDSL bond group. The MALC will need to be configured as a DHCP server and a host will need to be added for each bond group, for example:

```
host add 1-1-99-0/efmbond vlan 7 dynamic 1 1
```

The auto-provisioning runs on the EtherXtend only after upgrading from a version of the EtherXtend software that did not support auto-provisioning to a version that does support auto-provisioning, or following a **set2default**.

```
zSH> imCreated default bridges
Created interface AutoConfig/ip
Created default OAM interface
Auto-provisioning is complete.
```

```
zSH> bridge show
```

Type	VLAN	Bridge	St	Table	Data
tls	7	1-1-99-0-efmbond/bridge	UP		
tls Tagged		1-1-99-0-efmbond-0/bridge	UP		
tls	7	1-1-1-0-eth/bridge	DWN		

```

tls Tagged      1-1-1-0-eth-0/bridge      DWN
tls             7 1-1-4-0-eth/bridge      DWN
tls Tagged      1-1-4-0-eth-0/bridge      DWN
tls             7 1-1-3-0-eth/bridge      DWN
tls Tagged      1-1-3-0-eth-0/bridge      DWN
tls             7 1-1-2-0-eth/bridge      DWN
tls Tagged      1-1-2-0-eth-0/bridge      DWN
zSH>

```

```
zSH> interface show
```

```

1 interface
Interface      Status  Rd/Address          Media/Dest Address  IfName
-----
1/1/99/0/ip    UP      1 0.0.0.0/0         00:01:47:08:a4:32   AutoConfig
-----
zSH>

```

To view the VLAN ID and DHCP client configuration, enter:

```

zSH> get ip-interface-record 1/1/99/0/ip
ip-interface-record 1/1/99/0/ip
vpi: -----> {0}
vci: -----> {0}
rdindex: -----> {1}
dhcp: -----> {client}
addr: -----> {10.250.1.89}
netmask: -----> {255.255.255.0}
bcastaddr: -----> {10.250.1.255}
destaddr: -----> {0.0.0.0}
farendaddr: -----> {0.0.0.0}
mru: -----> {1500}
reasmmaxsize: -----> {0}
ingressfiltername: -----> {}
egressfiltername: -----> {}
pointtopoint: -----> {no}
mcastenabled: -----> {yes}
ipfwdenabled: -----> {yes}
mcastfwdenabled: -----> {yes}
natenabled: -----> {no}
bcastenabled: -----> {yes}
ingressPacketRuleGroupIndex: --> {0}
egressPacketRuleGroupIndex: --> {0}
ipaddrdynamic: -----> {dhcpclient}
dhcpserverenable: -----> {false}
subnetgroup: -----> {0}
unnumberedindex: -----> {0}
mcastcontrollist: -----> {}
vlanid: -----> {7}
maxVideoStreams: -----> {0}
tosOption: -----> {disable}
tosCOS: -----> {0}
vlanCOS: -----> {0}
s-tagTPID: -----> {0x8100}
s-tagId: -----> {0}

```

```
s-tagIdCOS: -----> {0}
```

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
Type 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
```

Creating an s-tagged uplink bridge

Follow these steps to create an s-tagged (double tagged) uplink bridge.

- 1 To configure an s-tagged uplink bridge enter **bridge add interface/type uplink tagged**.

```
zSH> bridge add 1-1-2-0/eth uplink tagged
Adding bridge on 1-1-2-0/eth
```

- 2 To create a bridge path for the uplink bridge, enter **bridge-path add interface/type global**.

```
zSH> bridge-path add ethernet2-0-0/bridge global
Bridge-path added successfully
```

- 3 To create a downlink bridge for both the VLAN and the SLAN, enter **bridge add interface/type VLAN 200 SLAN 400**.

```
zSH> bridge add 1-13-201-0/n2nbond downlink vlan 200
slan 400
Adding bridge on 1-13-201-0/n2nbond
```

```
Created bridge-interface-record 1-13-201-0-n2nbond/
bridge
```

- 4 To verify the bridges just created, enter **bridge show**.

```
zSH> bridge show
Type VLAN      Bridge
-----
dwn  200/400    1-13-201-0-n2nbond/bridge      PND
upl  STagged    ethernet2-0-0/bridge          UP S Global
default [U:
3600 sec, M: 150 sec, I: 0 sec]
zSH>
```

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.

When using the **systemreboot** command:

- a** Copy the infoserv backup files to the release version of the previous release (infoserv.v[version]).
- b** Download the MALC binary (.bin) files for the previous release.
- c** Enter the **systemreboot** command.

You use the *restore* file to restore to an existing configuration once. After the *restore* file is used by the system, it is renamed *restore.done* and is not read again. This file provides a means to save a working configuration when needed.

When using the onreboot restore method:

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

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

- b** 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.

- c** Download the old system image software for each slot card installed in the system 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 ethx3300.bin
```



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

- d** Initialize the flash card's boot partition with the new image:

```
zSH> image flash ethx3300.bin 1 1
```

- e** 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.

Downgrading a system

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

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

```
zSH> file download 192.168.2.89 dumpfilename /card1/
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.

- 2 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 ethx3300.bin
```



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

- 3 Reboot the system.

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 *ethx3300.cfg* on the host 192.168.8.21:

```
zSH> dump network 192.168.8.21 ethx3300.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 1.14.2.119 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.

