



## **Command Reference**

Version: 2082

Copyright 2007-2010 ImageStream Internet Solutions, Inc., All rights Reserved.



# Table of Contents

<b>Command Reference</b> .....	<b>1</b>
A.....	1
abr {pcr} {mcr}.....	1
B.....	1
bandwidth {bits_per_second}.....	1
baud {bits_per_second}.....	2
bond {device_name}.....	2
bridge-group {Bridge ID} [spanning-disabled].....	2
C.....	3
cbr {peak} {cell} {rate}.....	3
crc {0 1 2}.....	3
D.....	3
debug {hardware protocol} {1 2 3 4 5 6 7 8 9}.....	3
default-router IPv4_Address.....	4
description string.....	4
dctype {0 1 2}.....	5
disable-hdlc-processing.....	5
dns-server <IPv4_Address> [IPv4_Address].....	5
domain-name domain.....	6
duplex {autofullhalf}.....	6
E.....	6
encapsulation {frame-relay hdlc none ppp raw l25 atmlaal5 snaplaal5 autopp}.....	6
F.....	7
frame-relay interval LMI_Interval.....	7
frame-relay lmi-type {ansi ccitt cisco}.....	7
frame-relay map ip IPv4_Address DLCI.....	8
frame-relay mode {dtel dce}.....	8
G.....	8
H.....	9
host IPv4_Address hardware-address MAC_Address.....	9
I.....	9
ip address IPv4_Address IPv4_Netmask [secondary].....	9
ip dhcp excluded-address IPv4_Address IPv4_Address.....	9
ip dhcp debug.....	10
ip dhcp pool pool_ID [ shared-network <name> ].....	10
ip pim candidate-bsr.....	11
ip pim candidate-rp.....	12
ip pim debug.....	12
ip pim sparse-mode.....	13
ip route IPv4_Address IPv4_Netmask Destination [metric metric].....	13
J.....	14
K.....	14
L.....	14
lease time {minutes hours}.....	14
loopback.....	14
M.....	14
mtu MTU_Size.....	14
multilink-group group_id.....	15
N.....	15

# Table of Contents

## Command Reference

netbios-name-server IPv4_Address.....	15
network IPv4_Address IPv4_Netmask.....	16
Q.....	16
P.....	16
pointtopoint address IPV4_Address.....	16
pvc [name] vci/<vpi>.....	16
ppp multilink.....	17
ppp multilink fragmentation {enabled disabled}.....	17
Q.....	17
R.....	17
rate-limit bits_per_second {input output} [latency milliseconds] [buffer kilobytes].....	17
S.....	18
service-module ds3 clock source <line   internal>.....	18
service-module ds3 scrambling {on off}.....	18
service-module e1 clock source <line   internal>.....	19
service-module e1 crc4.....	19
service-module e1 data-coding <Data_code>.....	19
service-module e1 egl.....	19
service-module e1 framing <Line_frame>.....	20
service-module e1 linecode <hdb3   ami >.....	20
service-module e1 timeslots <Timeslot_range>.....	20
service-module e1 unstructured.....	21
service-module e3 clock source <Clock_source>.....	21
service-module e3 egl.....	21
service-module e3 lbo.....	21
service-module t1 clock source <line   internal>.....	21
service-module t1 data-coding [normal inverted].....	22
service-module t1 egl.....	22
service-module t1 framing [esf   sf].....	22
service-module t1 lbo {0 7 15 22.5}.....	23
service-module t1 linecode [b8zs   ami].....	23
service-module t1 timeslots {all   Timeslot_range} [speed {64 56}].....	23
service-module t3 egl.....	24
service-module t3 lbo.....	24
shutdown.....	24
speed < auto   100   10 >.....	24
T.....	25
transport < e1   t1   e3   ds3 > <g751>.....	25
tunnel destination <IPv4_Address> <port>.....	25
tunnel key <key>.....	26
tunnel mode openvpn <mode>.....	26
tunnel options <tunnel_options>.....	26
tunnel source <IPv4_Address> <port>.....	27
U.....	27
ubr.....	27
V.....	27
vbr <pcr> <cdvt> <scr> <mbs>.....	27
W.....	28

# Table of Contents

## Command Reference

X.....	28
x21-clockmode.....	28
Y.....	28
Z.....	28



# Command Reference

## A

### **abr {*pcr*} {*mcr*}**

#### **Description**

Available bit rate, or abr, specifies a minimum available bit rate for the interface.

#### **Parameters**

*pcr* - Peak cell rate

*mcr* - Minimum cell rate

*No Parameters* - If no parameters are set the abr command will use the bandwidth command to calculate the pcr and mcr values

#### **Examples**

**abr 500 100** - Sets the abr to a peak cell rate of 500 cells per second and a minimum cell rate of 100 cells per second

**abr** - Sets the abr using the bandwidth command to calculate the pcr and mcr

## B

### **bandwidth {*bits\_per\_second*}**

#### **Description**

Sets the intended bandwidth in bits per second. This command does not set internal clock speeds. See baud to set internal clocking for interfaces.

When used in conjunction with ATM QoS commands this value is used to calculate the correct ATM QoS settings. When this command is used on a Frame-relay interface this is used to setup rate-limiting on the interface.

#### **Parameter**

*bits\_per\_second* - Non-negative integer which represents how much data an interface can push.

#### **Examples**

**bandwidth 2000000** - Sets the interface to 2 Mbps

**bandwidth 100000000** - Sets the interface to 100Mbps

## **baud {*bits\_per\_second*}**

### **Description**

Configure internal clocking for the interface in bits per second. Use the baud command only with cards that do not have integrated CSU/DSUs. Use the service-module command to set clocking parameters for cards with integrated CSU/DSUs.

### **Parameter**

*bits\_per\_second* - Non-negative integer which represents how much data an interface can push.

### **Examples**

**baud 1500000** - Sets the interface to 1.5 Mbps  
**baud 45000000** - Sets the interface to 45 Mbps

## **bond {*device\_name*}**

### **Description**

Used to add serial devices or PVCs to a bond interface.

### **Parameter**

*device\_name* - Device\_name is a previously defined network device. This is only on Inetics interfaces, does not function for Ethernet interfaces.

### **Example**

**bond Serial0** - Attach Serial0 to the Bond interface.

## **bridge-group {*Bridge ID*} [spanning-disabled]**

### **Description**

This option goes into an interface section. It adds the interface to a group identified by <Bridge ID>. This is functionally equivalent to a switch containing all interfaces tagged bridge-group <Bridge ID>.

### **Parameter**

*Bridge ID* - An integer used to group ports in each bridge group. "Interface bvi<Bridge ID>" is used to assign an IP address to the bridge group.  
spanning-disabled - This optional parameter disables spanning tree protocol on the configured interface.  
priority - This optional parameter sets the bridge port priority on the configured interface.  
path-cost - This optional parameter sets the port's path cost (values 1-65535) on the configured interface.



## Examples

**bridge-group 1** - Adds the interface to group 1

**bridge-group 2 spanning-disabled** - Adds the interface to group 2 and disables spanning tree protocol

## C

### **cbr {*peak*} {*cell*} {*rate*}**

#### Description

Constant bit rate, or cbr, specifies a constant available bit rate for the interface

#### Parameters

*peak* - peak

*cell* - cell

*rate* - rate

#### Examples

**cbr** - Set the cbr

### **crc {0/1/2}**

#### Description

Sets the length of the Cyclic Redundancy Check (CRC). This is used to validate data transmitted across the link and must match the other side.

#### Parameter

0 - Sets the CSU / DSU to do no CRC checking

1 - Sets the CSU / DSU to do 16-bit CRC checking

2 - Sets the CSU / DSU to do 32-bit CRC checking

#### Example

**crc 0** - Disable all CRC checking

## D

### **debug {hardware|protocol} {1|2|3|4|5|6|7|8|9}**

## Description

Used to set the debugging level for protocol or hardware layers

## Parameters

hardware - Set debug level for hardware  
 protocol - Set debug level for protocol  
 1 - Module information only  
 2 - Adds hardware details (Default)  
 3 - Adds port state change  
 4 - Adds port error details  
 5 - Adds advanced hardware details  
 6 - Adds tx & rx packet counts  
 7 - Adds lock debugging  
 8 - Adds function entry/exit debugging  
 9 - Turns on everything possible

## Examples

**debug hardware 9** - Display all hardware debugging information

**debug protocol 1** - Display protocol module load/unload

## default-router *IPv4\_Address*

### Description

Set the default route, or gateway, associated with this pool of dhcp addresses. Generally this will be the IP address of the interface providing dhcp.

### Parameters

*IPv4\_Address* - Series of four numbers, 0 to 255, separated by periods. For more information see the Wikipedia

### Examples

**default-router 192.168.42.254** - Sets the DHCP server to hand out 192.168.42.254 as the gateway.

## description *string*

### Description

Used to add a comment (description) for tracking what is attached to a particular interface.

**Parameter**

*string* - Any alpha-numeric characters are allowed.

**Example**

**description** Link to CO - Documents this interface as connecting to CO

**dctype {0|1|2}****Description**

Several WANic series cards allow for software setting of the hardware interface type. Valid only for cards with a multi-interface daughtercard (WANic 524, 604 and 608 only)

**Parameters**

0 - V.35 (Default)  
1 - RS-422  
2 - RS-232

**Example**

**dctype 2** - Set the interface to RS-232

**disable-hdlc-processing****Description**

Valid only for 600 and 800 series cards. Disables HDLC frame processing for that particular interface on the WAN card for non-IP or non-framed data applications. Typically used with the driver suite's character mode operation (encapsulation type "none").

**Parameters**

None

**Example**

**disable-hdlc-processing** - Disables HDLC frame processing for the configured interface.

**dns-server <IPv4\_Address> [IPv4\_Address]****Description**

Set the DNS server(s) for the DHCP server to hand out in a given pool. A maximum of two servers are allowed.

## Parameters

*IPv4\_Address* - Series of four numbers, 0 to 255, separated by periods. For more information see the Wikipedia

## Examples

**dns-server 192.168.42.254 192.168.42.42** - Set the dhcp server to hand out 192.168.42.254 and 192.168.42.42 as dns servers.

## domain-name *domain*

### Description

Set the domain name for the DHCP server to hand out to a given pool.

### Parameters

*domain* - DNS Domain name, usually something similar to COMPANY.COM

### Examples

**domain-name mydomain.com** - Set the DHCP server to hand out mydomain.com

## duplex {auto|full|half}

### Description

Set the ethernet interface to either full or half duplex. If this option is not specified, the interface will auto negotiate.

### Parameters

auto - Auto-negotiate full or half duplex.

full - Sets the interface to full duplex ( Send and receive at the same time ).

half - Sets the interface to half duplex ( Send or receive at a given time ).

### Example

**duplex half** - Sets the interface to half duplex.

## E

### encapsulation

**{frame-relay|hdlc|none|ppp|raw|x25|atm|aal5snap|aal5autopp}**

## Description

Set the protocol for a serial interface.

## Parameters

*frame-relay* - Frame Relay protocol (Uses IETF standard)

*hdlc* - High-Level Data Link Control (HDLC) protocol for serial interface. This encapsulation method provides the synchronous framing and error detection functions of HDLC without windowing or retransmission

*none* - Raw character device read/write for external applications

*ppp* - Point-to-Point Protocol

*raw* - Raw IP encapsulation

*x25* - X.25 protocol

*atm* - ATM VC multiplex encapsulation

*aal5snap* - ATM Aal5snap encapsulation (valid only on an atm sub-interface)

*aal5autopp* - ATM Aal5autopp encapsulation (valid only on an atm sub-interface)

## Examples

**encapsulation frame-relay** - Set the protocol on the interface to frame-relay.

**encapsulation raw** - No protocol, an external program will process the raw data from the line.

# F

## frame-relay interval *LMI\_Interval*

### Description

Set the LMI interval in Mhz

### Parameters

*LMI\_Interval* - LMI Interval in Mhz, for more information on LMI, see the Wikipedia

### Examples

**frame-relay interval 200** - Sets LMI to 200Mhz

## frame-relay lmi-type {ansi|ccitt|cisco}

### Description

Set the lmi type for an interface. Valid only in main interface configurations and not in subinterfaces.

For more information on LMI, see the Wikipedia

## Parameters

*ansi* - Annex D defined by ANSI standard T1.617 (Default)

*ccitt* - ITU-T Q.933 Annex A

*cisco* - LMI type defined jointly by Cisco, DEC, StrataCom and NorTel (Northern Telecom).

## Examples

**frame-relay lmi-type ccitt** - Set the system to use ccitt frame-relay

## **frame-relay map ip *IPv4\_Address* *DLCI***

### Description

Maps an IP address to a frame relay dlcid number. Provided for backwards compatibility only. Configure PVCs using subinterfaces instead of using frame-relay map ip statements.

### Parameters

*IPv4\_Address* - Series of four numbers, 0 to 255, separated by periods. For more information see the Wikipedia

*DLCI* - Positive integer used to identify a PVC. This is normally assigned by telco.

## Examples

**frame-relay map ip 192.168.42.2 3** - Attaches the address 192.168.42.2 to DLCI number 3

## **frame-relay mode {*dte*|*dce*}**

### Description

Sets the frame-relay mode to dte or to dce

### Parameters

*dte* - Data Terminal Emulation, normally used by customer equipment (Default)

*dce* - Data Carrier Emulation, commonly the setting used by Telco equipment.

## Examples

**frame-relay mode dce** - Set the line to dce mode

# G

## H

### host *IPv4\_Address* hardware-address *MAC\_Address*

#### Description

Associates a MAC address with a specific IP address in a DHCP pool. As MAC addresses must be unique on a given LAN, it effectively assigns it to the device.

#### Parameters

*IPv4\_Address* - Series of four numbers, 0 to 255, separated by periods. For more information see the Wikipedia

*MAC\_Address* - Series of six hex numbers 0 - 255, separated by colons

#### Examples

**host 192.168.42.1 hardware-address 00:08:02:0C:C8:EA** - Assign the IP address 192.168.42.1 to the system identified by the MAC address 00:08:02:0C:C8:EA

## I

### ip address *IPv4\_Address IPv4\_Netmask* [secondary]

#### Description

To set IP addresses for an interface, use the ip address command.

#### Parameters

*IPv4\_Address* - Series of four numbers, 0 to 255, separated by periods. For more information see the Wikipedia

*IPv4\_Netmask* - Four integers from zero to 255 separated by periods

secondary - Specifies additional IP addresses (aliases)

#### Examples

**ip address 10.1.1.199 255.0.0.0** - Assigns the interface the IP address 10.1.1.199 with a Class A network mask

**ip address 192.168.1.254 255.255.255.0 secondary** - Assigns the interface the alias IP address 192.168.1.254 with a Class C network mask

### ip dhcp excluded-address *IPv4\_Address IPv4\_Address*

## Description

Sets a range of addresses to be excluded from use in the address pool. You should exclude the IP address of the interface which is providing DHCP service at a minimum.

## Parameters

*IPv4\_Address* - Series of four numbers, 0 to 255, separated by periods. For more information see the Wikipedia

*IPv4\_Address* - Series of four numbers, 0 to 255, separated by periods. For more information see the Wikipedia

## Examples

**ip dhcp excluded-address 192.168.42.250 192.168.42.254** - Set DHCP to not hand out addresses from 192.168.42.250 to 192.168.42.254

Turns on debugging in a dhcp pool

## ip dhcp debug

### Description

Enables debug information for the dhcp service.

### Parameters

None

### Examples

**ip dhcp debug** - Turn on debug.

## ip dhcp pool *pool\_ID* [ shared-network <name> ]

### Description

Required command to create the DHCP pool.

### Parameters

*pool\_ID* - Positive integer used to identify the group of addresses(or pool)

shared-network *name* - Required parameter when configuring multiple DHCP pools on a single interface. Each pool sharing an interface must have the same shared-network name parameter specified.



## Examples

**ip dhcp pool 1** - creates a dhcp pool and assigns it pool ID 1

**ip dhcp pool 2 shared-network WirelessCustomers1** - creates a shared dhcp pool with ID 1 and shared network name *WirelessCustomers1*

**ip dhcp pool 3 shared-network WirelessCustomers1** - creates a second dhcp pool on the shared network *WirelessCustoemrs1*

## ip pim candidate-bsr

### Description

Configures the router as a potential BootStrap Router (BSR).

### Parameters

None

### Examples

Basic PIM setup with router as a candidate RP and BSR:

```
interface Ethernet0
 ip pim sparse-mode
 ip address 192.168.109.1 255.255.255.0
!
interface Ethernet1
 ip pim sparse-mode
 ip address 192.168.110.1 255.255.255.252
!
# We want to be a Candidate RP using our Ethernet0 IP advertising every 15 seconds for the (default)
# full Class D group
# Priority: smaller value means higher priority

ip pim candidate-rp 192.168.109.1 priority 0 interval 15

# We want to be a Candidate BSR using our Ethernet0 IP
# Priority: larger value means higher priority

ip pim candidate-bsr 192.168.109.1 priority 10
```

More advanced example with multicast boundaries:

```
interface Ethernet0
 ip pim sparse-mode
 ip address 192.168.109.1 255.255.255.0
!
interface Ethernet1
 # Don't allow 239.0.0.0 and 224.0.1.39 to be forwarded to this interface
 ip pim sparse-mode deny 239.0.0.0 255.255.255.0 deny 224.0.1.39 255.255.255.0
 ip address 192.168.110.1 255.255.255.252
!

# Be a candidate RP for 239.0.1.0 and 239.0.2.0
```

```
ip pim candidate-rp 192.168.109.1 priority 0 interval 15 group 239.0.1.0 255.255.255.0 group 239.0.2.0 255.255.255.0
ip pim candidate-bsr 192.168.100.141 priority 5

# Switch from shared tree to spt at 50 Kbps
ip pim spt-threshold 50

# Enable PIM debugging to syslog
ip pim debug
```

## ip pim candidate-rp

### Description

Configures the router as a potential Rendezvous Point (RP).

### Parameters

None

### Examples

Basic PIM setup with router as a candidate RP:

```
interface Ethernet0
 ip pim sparse-mode
 ip address 192.168.109.1 255.255.255.0
!
interface Ethernet1
 ip pim sparse-mode
 ip address 192.168.110.1 255.255.255.252
!
# We want to be a Candidate RP using our Ethernet0 IP advertising every 15 seconds for the (default)
# full Class D group
# Priority: smaller value means higher priority

ip pim candidate-rp 192.168.109.1 priority 0 interval 15
```

## ip pim debug

### Description

Enables global syslog debugging output for PIM sparse-mode multicast routing.

### Parameters

None

### Examples

**ip pim debug** - Turn on debug.

## ip pim sparse-mode

### Description

Enables PIM sparse-mode multicast routing on the selected interface.

### Parameters

None

### Examples

Basic PIM setup:

```
interface Ethernet0
 ip pim sparse-mode
 ip address 192.168.109.1 255.255.255.0
!
interface Ethernet1
 ip pim sparse-mode
 ip address 192.168.110.1 255.255.255.252
!
```

The router will learn the multicast Rendezvous Point (RP) automatically.

## ip route *IPv4\_Address IPv4\_Netmask Destination [metric metric]*

### Description

To set IP addresses for an interface, use the ip address command.

### Parameters

*IPv4\_Address* - Series of four numbers, 0 to 255, separated by periods. For more information see the Wikipedia

*IPv4\_Netmask* - Four integers from zero to 255 separated by periods

*Destination* - Can be either an *IPv4\_Address* or a device name.

*metric metric* - The metric keyword is used to denote a preference setting for multiple routes, the metric value actually assigns preference. 1 would be first choice, 2 would be second choice, etc.

### Examples

**ip route 0.0.0.0 0.0.0.0 Serial0 metric 1** - Set the gateway or default route to Serial0 and make it the first choice.

**ip route 0.0.0.0 0.0.0.0 Serial1 metric 2** - Set the gateway or default route to Serial1 and make it the second choice.

**ip route 192.168.1.0 255.255.255.0 192.168.0.1** - Route 192.168.1.0/24 to 192.168.0.1

## J

## K

## L

### **lease *time* {minutes|hours}**

#### **Description**

Set the lease time, or duration for clients to keep their addresses in a specific pool before they request configuration again.

#### **Parameters**

*time* - Positive integer.

minutes - Specifies that *time* is measured in minutes.

hours - Specifies that *time* is measured in hours.

#### **Examples**

**lease 50 minutes** - The DHCP lease expires in 50 minutes

**lease 5 hours** - The DHCP lease expires in 5 hours

## **loopback**

#### **Description**

Sets the card into loopback mode

#### **Parameters**

No parameters

#### **Examples**

**loopback** - Enables loopback mode

## **M**

### **mtu *MTU\_Size***

## Description

Configure the maximum transfer unit(MTU) size for this interface. 800 series cards default to 4470 byte MTU. All other cards default to a 1500 byte MTU. The mtu command is valid in Serial, Bonded and Ethernet devices. 10 Mbps and 100 Mbps Ethernet do not support an MTU above 1500. Gigabit Ethernet supports larger MTU sizes.

## Parameters

*MTU\_Size* - positive integer, default values are set per interface type

## Examples

**mtu 3000** - Sets the MTU to 3000 bytes

## **multilink-group *group\_id***

### Description

Sets the group number of the multilink device, this number names the multilink interface which binds the group together. i.e. group 1 would be bound to Multilink1

### Parameter

*group\_id* - Identifies the multilink group this interface is part of.

### Examples

**multilink-group 3** - Assigns the interface to group 3

## N

## **netbios-name-server *IPv4\_Address***

### Description

Set the netbios nameserver for DHCP to hand out to clients in the configured pool. This is used by Windows to resolve SMB hostnames.

### Parameters

*IPv4\_Address* - Series of four numbers, 0 to 255, separated by periods. For more information see the Wikipedia

## Examples

**netbios-name-server 192.168.42.254** - Assign the IP address 192.168.42.1 to the system identified by the MAC address 00:08:02:0C:C8:EA

## **network *IPv4\_Address IPv4\_Netmask***

### Description

Sets the range of addresses to be used by the DHCP server in a given pool.

### Parameters

*IPv4\_Network* - Series of four numbers, 0 to 255, separated by periods. For more information see the Wikipedia

*IPv4\_Netmask* - Series of four numbers 0 - 255, separated by periods. The netmask indicates which addresses are local, and which are not. for more information see the Wikipedia

## Examples

**network 192.168.42.0 255.255.255.0** - Tells the DHCP server to hand out any address between 192.168.42.1 and 192.168.42.254.

## O

## P

## **pointtopoint address *IPV4\_Address***

### Description

Configure the destination address for this interface. For use with PPP connections when the destination router will not negotiate this setting.

### Parameter

*IPV4\_Address* - Series of four numbers, 0 to 255, separated by periods. For more information see the Wikipedia

### Example

**pointtopointaddress 192.168.2.253** - Sets the far side of a point to point link to 192.168.2.253

## **pvc [*name*] *vci*/*<vpi>***

## Description

Sets up a PVC, or Permanent Virtual Circuit, using a specified VPI and VCI identifier- The name field is present for compatibility reasons. It is not required and is not used.

## Parameters

*name* - Specifies a name for the pvc. Unused by the system.

*vci* - vci

*vpi* - vpi

## Example

**pvc home 3/30** - Use vci 3 and vpi 30 for this PVC

## ppp multilink

### Description

Enables multilink on a device using the PPP protocol.

## ppp multilink fragmentation {enabled|disabled}

### Description

### Parameters

enabled - Turn on the feature(Default)

disabled - Turn off the feature

### Examples

**ppp multilink fragmentation disabled** - Turns off multilink fragmentation ( Common when dealing with Cisco routers )

## Q

## R

## rate-limit *bits\_per\_second* {input|output} [*latency milliseconds*] [*buffer kilobytes*]

### Description

Restricts data rates on the interface to the bits per second value specified

## Parameters

*bits\_per\_second* - Non-negative integer which represents how much data an interface can push  
input - Applies the rate-limiting policy to packets received on this interface only  
output - Applies the rate-limiting policy to packets sent on this interface only  
latency *milliseconds* - Highest allowed latency value on the interface (affects buffer size and packet drop rates)  
buffer *bytes* - Highest allowed buffer value on the interface (affects latency and packet drop rates)

## Examples

**rate-limit 500000 input latency 50 buffer 50000** - Limits device input to 500Kbps with a maximum latency of 50ms and a maximum buffer of 50Kb

**rate-limit 400000 output latency 40 buffer 40000** - Limits device output to 400Kbps with a maximum latency of 40ms and a maximum buffer of 50Kb

## S

### **service-module ds3 clock source <line / internal>**

#### Description

Set the clock clock or line sync for an interface

#### Parameter

line - Set the interface to sync from the line.  
internal - Set the interface to generate sync on the line.

#### Example

*service-module ds3 clock source internal* - Sets the internal CSU/DSU to generate clock signal for the connection

### **service-module ds3 scrambling {on|off}**

#### Description

Turn scrambling mode on the DS3 CSU on or off.

#### Parameter

on - Enable CSU scrambling mode. Larscom compatible.  
off - Disable CSU scrambling mode.



### Example

**service-module ds3 scrambling** - Enable CSU scrambling  
**service-module ds3 scrambling on** - Enable CSU scrambling

## **service-module e1 clock source <line | *internal*>**

### Description

Set the clock clock or line sync for an interface

### Parameter

*line* - Set the interface to sync from the line.  
*internal* - Set the interface to generate sync on the line.

### Example

**service-module e1 clock source internal** - Sets the internal CSU/DSU to generate clock signal for the connection

## **service-module e1 crc4**

### Description

Enables the generation of CRC4 (per ITU recommendation G.703 and G.704) to improve data integrity. NOTE: The 550 and N2 series cards do not support this option.

## **service-module e1 data-coding <Data\_code>**

### Description

Set the internal CSU's coding to normal or inverted

### Parameter

<Data\_code> - Either 'inverted' or 'normal'. Inverted switches the data coding. Normal does not.

### Example

*service module e1 data-coding inverted* - Sets the internal CSU/DSU to use inverted line coding

## **service-module e1 egl**

## Description

Enable the equalizer on the integrated CSU

## service-module e1 framing <Line\_frame>

### Description

Set the internal CSU's framing to CAS or CCS

### Parameters

<Line\_frame> - Can be set to either 'CAS' or 'CCS'

### Example

*service module e1 framing cas* - Sets the internal CSU/DSU to use cas

## service-module e1 linecode < hdb3 / ami >

### Description

Set the inline coding on the CSU

### Parameters

*ami* - AMI is Alternate Mark Inversion.

*hdb3* - HDB3 is High Density Bipolar of order 3 code.

### Examples

*service module e1 linecode ami* - Sets the internal CSU/DSU to use AMI linecoding

## service-module e1 timeslots <Timeslot\_range>

### Description

Set timeslot usage and speed per timeslot of the internal CSU. E1 lines have thirty two 64Kbit channels.

### Parameters

<Timeslot\_range> - Can either be 'all', or a range using the syntax 'range <range> '.

### Example

*service-module e1 timeslots range 1-4,10,12,14,16* - Sets the internal CSU/DSU to use timeslots 1, 2, 3, 4, 10, 12, 14, and 16 for a total of 512Kb bandwidth

## service-module e1 unstructured

### Description

Sets unstructured mode on the e1 CSU/DSU, making all 32 timeslots available for use. NOTE: The 550 and N2 series cards do not support this option.

## service-module e3 clock source <Clock\_source>

### Description

Set the clock clock or line sync for an interface

### Parameter

<Clock\_source> - Either 'internal' or 'line'. Internal causes the card to generate line sync. Line pulls sync from the line.

### Example

*service module e3 clock source internal* - Sets the internal CSU/DSU to generate clock signal for the connection

## service-module e3 egl

### Description

Enable the equalizer on the integrated CSU

## service-module e3 lbo

### Description

Enable line build-out on the internal CSU. Use only if the cable between your card's TX connector to the demarcation point is greater than 225 feet.

## service-module t1 clock source <line / *internal*>

### Description

Set the clock clock or line sync for an interface

### Parameter

line - Set the interface to sync from the line.  
internal - Set the interface to generate sync on the line.

## Example

*service-module t1 clock source internal* - Sets the internal CSU/DSU to generate clock signal for the connection

## service-module t1 data-coding [normal|inverted]

### Description

Set the internal CSU's coding to normal or inverted

### Parameter

*normal* - Leaves t1 data unmodified.  
*inverted* - Inverts t1 data coding.

## Example

*service module t1 data-coding inverted* - Sets the internal CSU/DSU to use inverted line coding

## service-module t1 egl

### Description

Enable the equalizer on the integrated CSU

## service-module t1 framing [esf | sf ]

### Description

Set the internal CSU's framing to SuperFrame(SF) or Extended SuperFrame(ESF)

### Further Reading

[http://en.wikipedia.org/wiki/Extended\\_Super\\_Frame](http://en.wikipedia.org/wiki/Extended_Super_Frame)  
[http://en.wikipedia.org/wiki/Super\\_Frame](http://en.wikipedia.org/wiki/Super_Frame)

### Parameters

*esf* - Extended super frame  
*sf* - Super frame

## Example

*service module t1 framing sf* - Sets the internal CSU/DSU to use SuperFrame

## service-module t1 lbo {0|-7|-15|-22.5}

### Description

Set the internal CSUs line buildout. Use only if the cable between your cards TX connector to the demarcation point is greater than 225 feet

### Parameters

0 - Set line buildout to 0db.  
 -7.5 - Set line buildout to -7.5db.  
 -15 - Set line buildout to -15db.  
 -22.5 - Set line buildout to -22.5db.

### Example

**service-module t1 lbo -7.5** - Set the buildout to -7.5db

## service-module t1 linecode [b8zs / *ami*]

### Description

Set the inline coding on the CSU.

### Further Reading

<http://en.wikipedia.org/wiki/B8ZS>  
[http://en.wikipedia.org/wiki/Alternate\\_Mark\\_Inversion](http://en.wikipedia.org/wiki/Alternate_Mark_Inversion)

### Parameters

b8zs - B8ZS is Bipolar with 8 Zeros Substitution  
*ami* -AMI is Alternate Mark Inversion.

### Examples

*service module t1 linecode ami* - Sets the internal CSU/DSU to use AMI linecoding

## service-module t1 timeslots {all | *Timeslot\_range*} [speed {64|56}]

### Description

Set timeslot usage and speed per timeslot of the internal CSU. T1 lines have twenty four timeslots at either 56Kbit or 64Kbit. NOTE: use of speed 56 requires the use of sf or d4 framing

## Parameters

*all* - Use all twenty four timeslots available on the line.

*Timeslot\_range* - Range of timeslots using the syntax such as *start-stop,slot* .

64 - Set the interface to 64Kbit timeslots

56 - Set the interface to 56Kbit timeslots

## Example

**service-module t1 timeslots range 1-4,10,12,14,16** - Sets the internal CSU/DSU to use timeslots 1, 2, 3, 4, 10, 12, 14, and 16 for a total of 512Kbps bandwidth

**service-module t1 timeslots range 1-4,10,12,14,16 speed 56** - Sets the internal CSU/DSU to use timeslots 1, 2, 3, 4, 10, 12, 14, and 16 for a total of 448Kbps bandwidth

## service-module t3 egl

### Description

Enable the equalizer on the integrated CSU

## service-module t3 lbo

### Description

Enable line build-out on the internal CSU. Use only if the cable between your card's TX connector to the demarcation point is greater than 225 feet.

## shutdown

### Description

Shutdown an interface.

## speed < auto / 100 / 10 >

### Description

Set the speed of an ethernet interface. If this option is not specified, the device will auto negotiate.

## Parameters

100 - Set the interface to 100Mbps.

10 - Set the interface to 10Mbps.

auto - auto-negotiate the speed of the interface.

### Example

*speed 10* - Sets the ethernet interface to 10Mbps

## T

**transport < e1 | t1 | e3 | ds3 > <g751>**

### Description

This command sets the mode of operation for the interface. Default setting and valid options are dependant on the card. 440, 431, and 530-TEJ cards are t1/e1 and default to t1. While 530-DE cards are t3/e3 and default to t3.

### Parameter

*e1* - specify e1 mode.

*t1* - specify t1 mode.

*e3* - specify e3 mode.

*ds3* - specify t3 mode.

*g751* - specifies g751, only valid with "transport e3"

### Example

*transport e1* - Sets the interface to use e1 mode (if allowed).

*transport e3* - Sets the interface to use e3 mode (if allowed).

*transport e3 g751* - Sets the interface to e3 g751 mode.

**tunnel destination <IPv4\_Address> <port>**

### Description

Set the IP Address and Port used by the other side of the tunnel.

### Parameters

<IPv4\_Address> - Series of four numbers, 0 to 255, separated by periods. For more information see the Wikipedia

<port> - The port used by the other side of the tunnel.

### Examples

*tunnel destination 192.168.42.42 1104* - Connect to 192.168.42.42:1104

## tunnel key <key>

### Description

Set the key used for tunnel encryption.

### Parameters

<key> - 32 character hex value used to encrypt the tunnel. Not strictly required for a tunnel, but without it there is no encryption.

### Examples

*tunnel key 8232f92562b8e187af624312cc643147* - use '8232f92562b8e187af624312cc643147' as the tunnel key

## tunnel mode openvpn <mode>

### Description

Set the tunnel to client, server, or peer to peer mode.

### Parameters

<mode> - Choices are 'server', 'client', or blank. Server and client are server and client mode respectively, blank sets peer to peer mode.

### Examples

*tunnel mode openvpn server* - Sets the interface to server mode.  
*tunnel mode openvpn* - Sets the interface to peer to peer mode.

## tunnel options <tunnel\_options>

### Description

Specify command line options to OpenVPN, any command line option can be passed this way. You should be familiar with OpenVPN before using this command, as not all options are valid on all types of tunnels.

### Parameters

<tunnel\_options> - Command-line options to be passed to openvpn. This can be used to circumvent limitations in the wan.conf command structure.



## Examples

*tunnel options --push "route 192.168.42.0 255.255.255.0"* - Tell the other side to send traffic to 192.168.42.0/24 here.

## **tunnel source <IPv4\_Address> <port>**

### Description

Set the IP Address and Port used by this side of the tunnel.

### Parameters

<IPv4\_Address> - Series of four numbers, 0 to 255, separated by periods. For more information see the Wikipedia

<port> - The port used by the other side of the tunnel.

## Examples

*tunnel source 192.168.42.42 1104* - Connect from 192.168.42.42:1104

## U

### **ubr**

### Description

Sets Unspecified Bit Rate parameters.

### Parameters

No parameters

## Examples

*ubr* - Set the system to use unspecified bit rate

## V

### **vbr <pcr> <cdvt> <scr> <mbs>**

### Description

Sets Variable Bit Rate parameters.

## Parameters

<pcr> - peak cell rate

<cdvt> - cell delay variation tolerance

<scr> - sustainable cell rate

<mbs> - maximum burst size

*No Parameters* - If no parameters are set the vbr command will use the bandwidth command to calculate the pcr and cdvt values

## Examples

*vbr* - Sets the vbr using the bandwidth command to calculate the pcr and cdvt

*vbr 8500 50 8500 10* - Sets vbr using the following parameters; pcr to 8500 the cdvt to 50 the scr to 8500 and the mbs to 10

## W

## X

### x21-clockmode

#### Description

Valid only for cards using an RS422 interface. Setting this clocking option on cards using an RS422 interface and the standard X.21 DB26-DB15 cable disable the use of the carrier signal (DCD) and enables the dual clocking required for X.21 interfaces. Note: Remember to set "dctype 1" to set RS422 mode on cards that allow for software setting of the hardware interface type.

## Y

## Z