SpeedTouch™

510(i)/530/570(i)

Multi-User ADSL Gateways

CLI Reference Guide

500 SERIES

THOMSON

speedtouch™ ALCATEL

SpeedTouch™

500Series

CLI Reference Guide

Release R4.0

speedtouch™
ALCATEL

Status	Released
Change Note	PeckelbeenS
Short Title	CLI ST500Series R4.0
Copyright	

speedtouch™
ALCATEL

Content

DHCP Commands 67

About this Document

Introduction

Welcome to the SpeedTouch™500Series Command Line Interface (CLI) Reference Guide!

For the Service Provider this Reference Guide aims to give a concise and practical document for creating a customized configuration profile file, based on the SpeedTouch™500Series CLI command set, to be used by the end-user to configure the SpeedTouch™500Series and PC(s).

For the fastidious user this Reference Guide aims to give a handbook for advanced and detailed configuration and troubleshooting of the SpeedTouch™500Series via its character based Command Line Interface.

This CLI Reference Guide covers the CLI commands of the following DSL SpeedTouch™ products:

- SpeedTouch™510
- SpeedTouch™510i
- SpeedTouch™530
- SpeedTouch™570
- SpeedTouch™570i

Contents

The Reference Guide consists of two main parts:

- CLI Navigation
  The CLI navigation is decribed in this chapter. Here the user is familiarised with the SpeedTouch™500Series CLI. It describes general manipulations to navigate through and to perform some operations on the CLI.
- CLI Command Description
  The other chapters describe all the available CLI commands of the SpeedTouch™500Series per command group and in alphabetical order.
  Each command is described in a systematic manner:
  - The full name of the CLI command (including the group selection)
  - A short description of the CLI command, if needed completed by a description of the possible impact on the user and/or the SpeedTouch™500Series
  - The syntax of the command with a description of each parameter
  - An example to demonstrate the use of the CLI command
  - A list of related CLI commands.

Trademarks

The following trademarks are used in this document:

- SpeedTouch™ is a trademark of THOMSON
- Netscape, and Netscape Navigator, are registered trademarks of Netscape Communications Corporation
- Windows and Internet Explorer are trademarks of Microsoft Corporation
- Apple, and Mac,OS are registered trademarks of Apple Computer Inc.
- UNIX, is a registered trademark of UNIX System Laboratories, Inc.
- Ethernet is a trademark of Xerox Corporation.

Other products may be trademarks or registered trademarks of their respective manufacturers.

Updates

Due to the continuous evolution of DSL technology, existing products are regularly upgraded.

For more information on the latest technological innovations, software upgrades, and documents, please visit the SpeedTouch™ web site at:

http://www.speedtouch.com

Directive

Unless expressly and unambiguously approved by THOMSON, you may not:

- disassemble, de-compile, reverse engineer, trace or otherwise analyze the equipment, its content, operation, or functionality, or otherwise attempt to derive source code (or the underlying ideas, algorithms, structure or organization) from the equipment or from any other information provided by THOMSON, except to the extent that this restriction is expressly prohibited by local law;
- copy, rent, loan, re-sell, sublicense, or otherwise transfer or distribute the equipment to others;
- modify, adapt or create a derivative work of the equipment;
- remove from any copies of the equipment any product identification, copyright or other notices;
- disseminate performance information or analysis (including, without limitation, benchmarks) from any source relating to the equipment.
- Such acts not expressly approved by THOMSON will result in the loss of product warranty and will invalidate the user's authority to operate this equipment.

Accessing the Command Line Interface

Users can access the Command Line Interface via:

- A Telnet session
  This requires that TCP/IP connectivity exists between the host from which the Telnet session is opened and the SpeedTouch™500Series.

As soon a session to the CLI is opened, the SpeedTouch™500Series banner pops up, followed by the CLI prompt.

In case the SpeedTouch™500Series is protected by a System password, authentication will be required before access is granted.

The following figure shows an example of the SpeedTouch™500Series banner after opening a session and authentication.

```
/home/doejohn{1}$ telnet 10.0.0.138
Trying 10.0.0.138...
Connected to 10.0.0.138.
Username :
Password : ######
------------------------------------------------------------------------
*
*                             ______  Speedtouch  510
*                         ___/_____/\
*                        /         /\\  Version R4.0.0.9.0
*                  _____/__       /  \\
*                _/       /\_____/___ \  Copyright (c) 1999-2002,
*               //       /  \       /\ \              THOMSON multimedia
*       ________//_______/   \     / _\/______
*      /      / \       \    /    / /        /\
*   __/      /   \       \  /    / /        / _\__
*  / /      /     \_______\/    / /        / /   /\
* /_/______/___________________/ /________/ /___/  \
* \ \      \    ___________    \ \        \ \   \  /
*  \_\      \  /          /\    \ \        \ \___\/
*     \      \/          /  \    \ \        \  /
*      \_____/          /    \    \ \________\/
*           /__________/      \    \  /
*           \   _____  \      /_____\/
*            \ /    /\  \    /___\/
*             /____/  \  \  /
*             \    \  /___\/
*              \____\/
*
------------------------------------------------------------------------
=>
```

Basic Navigation and Manipulation

Manipulation commands are commands that manipulate operations on the command line, for example changing the command group, go to the beginning of the command line, go to the end of the command line, etc.

Command group Navigation

From top level, you can change to a command group by executing the name of the desired command group.

To obtain a list of all available command groups, execute help from the top level.

The example below shows every possible CLI command group for the SpeedTouch™510.

```
=>help
Following commands are available :
help             : Displays this help information
menu             : Displays menu
?                : Displays this help information
exit             : Exits this shell.
..               : Exits group selection.
saveall          : Saves current configuration.

Following command groups are available :

adsl             atm              bridge           cip              config
dhcp             dns              env              eth              firewall
ip               ipoa             mer              nat              password
phonebook        ppp              pptp             qosbook          software
system           td               upnp

=>
```

However:

- eth is only available on SpeedTouch™510 variants equipped with a single Ethernet port and the SpeedTouch™530
- upnp is only available for the SpeedTouch™510 variants and the SpeedTouch™530
- wireless is only available for the SpeedTouch™570

To return to top level, or to descend one level (in case of nested command groups) execute .. .

EXAMPLE:

```
=>phonebook
[phonebook]=>
[phonebook]=>..
=>
```

The Help Command

Execute help or ? from top level to list all available command groups for the SpeedTouch™500Series.

EXAMPLE for the SpeedTouch™570:

```
=>help
Following commands are available :
help     : Displays this help information
menu     : Displays menu
?        : Displays this help information
exit     : Exits this shell.
..       : Exits group selection.
saveall  : Saves current configuration.

Following command groups are available :

adsl            atmstats        bridge          cip             config
dhcp            dns             env             eth             firewall
ip              mer             nat             password        phonebook
ppp             pptp            qosbook         software        system
td              wireless
=>
```

You can execute the help or ? command from each command group selection. This results in a list of the available commands (and nested command groups, if available) in this particular command group.

EXAMPLE:

```
=>firewall
[firewall]=>
[firewall]=>?
Following commands are available :

tron         : Enables verbose console messaging.
troff        : Disables verbose console messaging.
match        : Defines an ip packet match.
assign       : Assign a chain to an entry point.
list         : Shows a list of all the hooks with the chain attached.
flush        : Clears all hooks. If a hook is provided, that hook is cleared.

Following command groups are available :

chain         rule

[firewall]=>
```

As both help and ? have the exact same functionality in the SpeedTouch™500Series CLI, the help command may always be equally replaced by the ? command.

Executing e.g. help firewall from top level gives the same result as executing help from the firewall command group selection.

EXAMPLE:

```
=>firewall help
Following commands are available :

tron          : Enables verbose console messaging.
troff         : Disables verbose console messaging.
match         : Defines an ip packet match.
assign        : Assign a chain to an entry point.
list          : Shows a list of all the hooks with the chain attached.
flush         : Clears all hooks. If a hook is provided, that hook is cleared.

Following command groups are available :

chain     rule

=>
```

Entering help followed by a specific command, e.g. help firewall assign (starting from top level) or help assign (executed from the the firewall command group selection) results in a description of the syntax for the command.

EXAMPLE:

```
=>help firewall assign
Assign a chain to an entry point.
Syntax: assign hook = <{input|sink|forward|source|output}> chain = <string>

parameters :
     hook = <{input|sink|forward|source|output}>
          Name of hook to assign chain to.
     chain = <string>
          Name of chain to use.

=>
```

Executing help all will generate the complete listing of all available CLI commands with syntax description.

Command Completion

The CLI features command completion, which means that when starting to enter a command it can be completed by pressing the TAB key.

For the completion to be successful, the part to be added must be unique. Completion works for the command groups, for the commands, for the options, but not for values.

For example, pressing a and TAB at the firewall command group selection results in the full assign command being completed. Entering firewall a and pressing the TAB key from top level gives the same result.

EXAMPLE:

```
=>firewall
[firewall]=>"a+TAB"
[firewall]=>assign
```

Going to the beginning or end of the Command Line

To go to …	Press …
the beginning of the Command Line	CTRL+A
the end of the Command Line	CTRL+E

Breaking off Commands

Press CTRL+G to break off a command. This can be useful in a situation where a user is prompted to enter a value which he does not know and wants to quit the command. Instead of being prompted over and over again for the same value, this allows to quit the command and return to the command line prompt.

In the example below CTRL+G is pressed after the third prompt chain =

```
[firewall]=>match
chain =
chain =
chain = “CTRL+G”
[firewall]=>
```

History of Commands

To select previously executed commands, use the UP and DOWN ARROW keys to select a previously executed command. Press ENTER to execute the selected command.

EXAMPLE:

```
=>firewall
[firewall]=>list
assign     hook=input chain=input
assign     hook=sink chain=sink
assign     hook=forward chain=forward
assign     hook=source chain=source
[firewall]=> “UP ARROW”
[firewall]=>:firewall list
```

Command Line Interface Top Level Structure

The following command groups are always available:

- atm
- bridge
- cip
- config
- dhcp
- dns
- env
- firewall
- ip
- ipoa
- mer
- nat
- phonebook
- ppp
- pptp
- qosbook
- software
- system
- td

Following command groups are optional:

- adsl (only applicable for the SpeedTouch™510, SpeedTouch™570 ADSL/POTS, and the SpeedTouch™510i and SpeedTouch™570i ADSL/ISDN variants)
- eth (only applicable for models equipped with a single 10/100Base-T Ethernet port)
- upnp (only applicable for the SpeedTouch™510 and SpeedTouch™530 variants)
- wireless (only applicable for the SpeedTouch™570 variants)

Command Line Interface Commands

All CLI commands are commands that operate on, or configure, the SpeedTouch^{TM}500Series settings.

You can execute these commands from top level, preceded by the name of the command group from which the command should be executed (e. g. firewall list).

You can also execute the commands from the command group itself, using the reduced form of the command (e.g. list at the firewall command group selection).

! in a command means 'NOT', e.g. the [!]syn parameter in the firewall rule create command.

EXAMPLE:

```
=>firewall list
assign     hook=input              chain=input
assign     hook=sink              chain=sink
assign     hook=forward     chain=forward
assign     hook=source     chain=source
=>firewall
[firewall]=>list
assign     hook=input              chain=input
assign     hook=sink              chain=sink
assign     hook=forward     chain=forward
assign     hook=source     chain=source
[firewall]=>
```

Instead of entering a completely built-up command with all its parameters, you can also enter just the command itself, without its parameters. After this you are prompted to complete the command with the required and the optional parameters. For the optional parameters you can simply press enter without giving a value.

The example below is the equivalent of firewall assign hook=input chain=input. To break of such incomplete command press CTRL+G.

EXAMPLE:

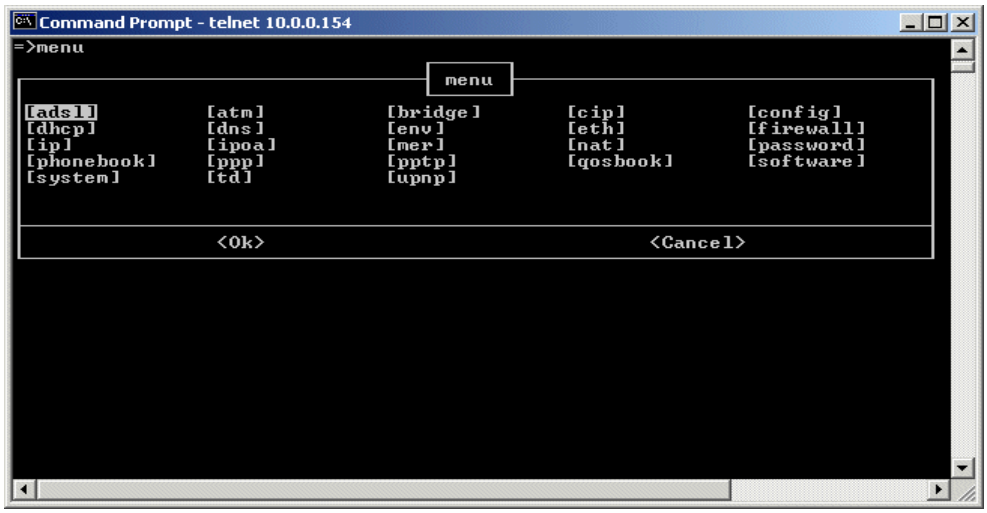
```
=>firewall assign
hook  =  input
chain =  input
=>
```

Menu-driven CLI Navigation

To improve the user-friendliness of the SpeedTouch™500Series CLI, the CLI features a menu-driven interface.

To enter the menu-driven interface, simply execute menu from the CLI prompt:

EXAMPLE (for the SpeedTouch™510):



The semi-graphical CLI offers you an attractive and easy-to-use configuration environment for the CLI.

You can browse through the CLI command groups via the ARROW keys. Press ENTER to execute your selection. From each level you can select '..' to go back one level up.

Use the TAB key to change from the CLI command menu to the control menu and vice versa.

To setup a CLI command, simply press ENTER on its name. You can configure and overview its various parameters at one time. In case the parameter provides preset values, you can go through these via the ARROW keys. If you are satisfied, use the TAB key to go to the OK field and press ENTER.

Note Do not forget to save your changes by executing saveall (from any CLI prompt).

Configuration Profile Files and the CLI

Configuration Profile Files500Series

The configuration profile files, used by the Setup wizard, or directly uploaded via the SpeedTouch™500Series web pages, consist of a set of CLI commands. Only CLI commands, which are part of the SpeedTouch™500Series CLI command set may be used in the .ini and .def files. For readability, configuration profile files are partitioned in paragraphs. These paragraphs reflect a CLI command group selection in the SpeedTouch™500Series CLI. However, due to internal processing, not all paragraph names are the same as their corresponding CLI command group selection.

In the following overview the relationship between the available SpeedTouch™500Series CLI commands and the [<configuration>.ini] paragraphs is given:

[<configuration>.ini] paragraphs	Available SpeedTouch™500Series CLI commands
[adslisdn.ini]	The adsl CLI command group in case of a ADSL/ISDN variant.
[adslpots.ini]	The adsl CLI command group in case of a ADSL/POTS variant.
[brfilter.ini]	The hidden brfilter CLI command group.
[bridge.ini]	The bridge CLI command group.
[cip.ini]	The cip CLI command group.
[dhcc.ini]	The dhcp client CLI command subgroup.
[dhcp.ini]	The dhcp server CLI command subgroup.
[dnsd.ini]	The dns CLI command group.
[env.ini]	The env CLI command group.
[eth.ini]	The eth CLI command group.
[ip.ini]	The ip CLI command group.
[ipoa.ini]	The cip CLI command group.
[mer.ini]	The mer CLI command group.
[nat.ini]	The nat CLI command group.
[oam.ini]	The atm oam CLI command subgroup.
[pfilter.ini]	• The firewall chain CLI command subgroup • The firewall rule CLI command subgroup.
[pfirewall.ini]	The firewall CLI command group (except those part of the firewall chain and firewall rule CLI command subgroups).
[phone.ini]	The phonebook CLI command group.
[ppp.ini]	The ppp CLI command group.
[pptp.ini]	The pptp CLI command group.
[qos.ini]	The qosbook CLI command group.

[<configuration>.ini] paragraphs	Available SpeedTouch™500Series CLI commands
[snmp.ini]	The snmp CLI command group.
[system.ini]	The system CLI command group.
[wireless.ini]	The wireless CLI command group.

CLI Commands in Configuration Profile Files

CLI commands in a paragraph of a configuration profile file should always be constructed in their complete form. Uncompleted CLI commands, i.e. commands in which required parameters are not specified, used in a configuration profile file will be discarded by the CLI commend interpreter. This may result in a wrongly configured SpeedTouch™500Series.

In configuration profile files, the use of customization variables, allow the Setup wizard to invite the end-user to provide some input regarding the settings of the SpeedTouch™500Series. The declaration of such variables must be done in the [env.ini] paragraph. Further use of the resulting variable TAG is allowed through all other paragraphs, even multiple times.

To make sure that a variable always will result in a valid variable TAG, the [env.ini] paragraph also allows to declare a preset value for the variable.

In case the variable TAG is used in a CLI command, the value of the variable should always be in conformity with the syntax of the CLI command.

Customizing Configuration Profile Files

For more information on the customization possibilities of the SpeedTouch™500Series, the Setup wizard and the configuration profile files, please check the SpeedTouch™ support pages at:

http://www.speedtouch.com

Direct FTP Access

The SpeedTouch™500Series File System

The SpeedTouch™500Series permanent storage, further referred to as 'file system', exists of nonvolatile memory responsible for storing, retrieving and maintaining the SpeedTouch™500Series software image(s), configuration profile files and optionally default settings files.

The file system of the SpeedTouch™500Series is accessible via the FTP transport protocol. This allows to transfer the SpeedTouch™500Series software image(s) and/or configuration profile files and default settings files.

Moreover, via FTP's quote site command you can execute CLI commands from the FTP prompt.

Proceed as indicated in the example below to open an FTP session to the SpeedTouch™500Series file system:

```
/home/doejohn{1}$ftp 10.0.0.138
Connected to 10.0.0.138
220 Inactivity timer = 120 seconds. Use 'site idle <secs>' to change.
Name (10.0.0.138:doejohn):
331 SpeedTouch™ (00-90-D0-01-02-03) User 'doejohn' OK. Password required.
Password : ######
330 OK
ftp>
```

SpeedTouch™500Series File System Structure

The files system features a tiny multilevel directory structure with a single root node called 'root' and two leaf nodes called 'active' and 'dl'.

The 'root' contains next to the two subdirectories 'active' and 'dl' all necessary files for the SpeedTouch™500Series to boot correctly.

The 'active' subdirectory always contains the software image in execution.

The 'dl' directory contains dormant software image. In case you have made changes to the SpeedTouch™500Series configuration and saved them, be it via a Telnet session, via the web pages or via the Setup wizard, a user.ini configuration profile file is created in the 'dl' subdirectory.

In other words, after each 'Save all', or config save call, the user.ini configuration profile file present in the 'dl' subdirectory reflects the current configuration of the SpeedTouch™500Series.

SpeedTouch™500Series File System Access Rights

Following access rights apply on the file system:

- 'root' Directory
  Listing of 'root' directory files (dir)
- 'active' Subdirectory
  Listing of 'active' subdirectory files (dir)
  FTP (m)get of (multiple) 'active' subdirectory files
- 'dl' Subdirectory
  Listing of 'dl' subdirectory files (dir)
  FTP (m)get of (multiple) 'dl' subdirectory files
  FTP (m)put of (multiple) 'dl' subdirectory files
  FTP (m)delete of (multiple) 'dl' subdirectory files.

FTP File Transfer

To allow correct file transfers, set the transfer mode to “binary”.

Note Turn on the hashing option to see the progression of the file transfer.

EXAMPLE:

```
/home/doejohn{1}$ftp 10.0.0.138
Connected to 10.0.0.138
220 Inactivity timer = 120 seconds. Use 'site idle <secs>' to change.
Name (10.0.0.138:doejohn):
331 SpeedTouch™ (00-90-D0-01-02-03) User 'doejohn' OK. Password required.
Password : ######
330 OK
ftp>
ftp>bin
200 TYPE is now 8-bit binary
ftp>
ftp>hash
200Hash mark printing on (8192 bytes/hash mark).
ftp>
```

ADSL Commands

Contents

This chapter covers the following commands:

adsl config

Show/set the Asymmetric Digital Subscriber Line (ADSL) configuration.

Although the same command for both SpeedTouch™ ADSL/POTS and SpeedTouch™ ADSL/ISDN variants, the command features specific parameter values per variant:

SYNTAX FOR ADSL/Plain Old Telephone Service (POTS) variants:

`adsl config`	`[opermode = <{ansi|g.dmt_annex_a|g.lite|multimode}>]` `[maxbitspertoneUS = <number{10-14}>]`

[opermode]	The operational mode of the SpeedTouch™ modem. Choose between: • ansi • g.dmt_annex_a • g.lite • multimode By default the SpeedTouch™ will start in multimode.	OPTIONAL
maxbitspertoneUS	A number between 10 and 14 (bits per tone). Represents the maximum number of bits which can be allocated to each ADSL DMT tone in the upstream direction. By default the modem will use up to 13 bits per tone.	OPTIONAL

SYNTAX FOR ADSL/Integrated Services Digital Network (ISDN) variants:

`adsl config`	`[opermode = <{etsi|g.dmt_annex_b|multimode}>]` `[maxbitspertoneUS = <number{10-14}>]`

[opermode]	The operational mode of the SpeedTouch™ modem. Choose between: • ansi • g.dmt_annex_b • multimode By default the SpeedTouch™ will start in multimode.	OPTIONAL
maxbitspertoneUS	A number between 10 and 14 (bits per tone). Represents the maximum number of bits whic hcan be allocated to each ADSL DMT tone in the upstream direction. By default the modem will use up to 13 bits per tone.	OPTIONAL

Note Executing the :adsl config without specifying parameters, shows the current ADSL configuration.

adsl info

Show ADSL statistics and information about the SpeedTouch™ DSL line status.

Although the same command for both SpeedTouch™ ADSL/POTS and SpeedTouch™ ADSL/ISDN variants, the command features specific output parameters and counters pe variant:

SYNTAX:

```
adsl info
```

EXAMPLE (for a SpeedTouch™500Series ADSL/POTS variant):

```
=>adsl info
Modemstate              :  up
Operation Mode          :  G.DMT Annex A  [POTS Overlay Mode]
Channel Mode            :  fast
Number of resets        :  1

Vendor (ITU)               Local           Remote
  Country               :  0f              0f
  Vendor                :  ALCB            ALCB
  VendorSpecific        :  0000            0000
  StandardRevisionNr    :  01              01

                           Downstream       Upstream
Margin       [dB]       :  25              31
Attenuation  [dB]       :  26              15

Available Bandwidth        Cells/s          Kbit/s
 Downstream     :          7924             3360
 Upstream       :          452              192

Transfer statistics
  Total since power On     Cells            Kbit
    Downstream           : 10153            4304
    Upstream             : 3399             1441
  Current Connection
    Downstream           : 10153            4304
    Upstream             : 3399             1441
    Errors
      Received FEC       : 0
      Received CRC       : 0
      Received HEC       : 0
```

.. Continued output on following page ..

.. Continued output ..

```
    Far End Failure
      No Failure
    Near end failure
      No failure
    Far end failures since reset
      Loss of frame:          0 failures
      Loss of signal:         0 failures
      Loss of power:          0 failures
      Loss of link:           0 failures
      Errored seconds:        0 seconds
    Far end failures last 15 minutes
      Loss of frame:          0 seconds
      Loss of signal:         0 seconds
      Loss of power:          0 seconds
      Loss of link:           0 seconds
      Errored seconds:        0 seconds
    Far end failures current day
      Errored seconds:        0 seconds
    Far end failures previous day
      Errored seconds:        0 seconds
    Near end failures since reset
      Loss of frame:          0 failures
      Loss of signal:         0 failures
      Loss of power:          0 failures
      Errored seconds:        0 seconds
    Near end failures last 15 minutes
      Loss of frame:          0 seconds
      Loss of signal:         0 seconds
      Loss of power:          0 seconds
      Errored seconds:        0 seconds
    Near end failures current day
      Errored seconds:        0 seconds
    Near end failures previous day
      Errored seconds:        0 seconds
=>
```

ATM Commands

Contents

This chapter covers the following commands:

Note The command :atm portstats is replaced by the command :atmstats portstats for the SpeedTouch™570. The command :atmstats portstats is the only ATM command for the SpeedTouch™570.

atm portstats

Show port specific Asynchronous Transfer Mode (ATM) statistics.

SYNTAX:

`atm portstats`	`port = <{DSL0|ATM|ATMF25|ATM3|ATM4|ATM5} or number>`

port	The port to show the ATM statistics of. Choose between: • DSL0 • ATM • ATMF25 • ATM3 • ATM5 Or specify a port number. **Note** The parameter DSL0 is replaced by the parameter ADSL0 for the SpeedTouch™570.	REQUIRED

EXAMPLE:

```
=>atm portstats port=DSL0
                  # of received octets  =  547649.
                  # of transmitted octets =  191648.
                  # of received cells  =  10333.
                  # of transmitted cells =  3616.
                  # of unknown cells   =  0.
                  # of errors on the input =  0.
                  # of errors on output  =  0.
=>
```

atm oam config

Configure Operation and Maintenance (OAM) cell settings.

SYNTAX:

`atm oam config`	`clp = <number{0-1}>`

clp	The CLP bit value of the OAM cells. Select either: • 0 • 1	REQUIRED

atm oam status

Show OAM data blocking mode of all ports.

SYNTAX:

```
atm oam status
```

EXAMPLE:

```
=>atm oam status
  OAM config dump
-------------------
        CLP bit value : 1

=>
```

Bridge Commands

Contents

This chapter covers the following commands:

bridge config

Show/set bridge ageing policy for dynamically learned Medium Access Control (MAC) addresses.

SYNTAX:

```
bridge config          [age =  <number {10 - 100000}>]
```

[age]	A number between 10 and 100000 (seconds). Represents the lifetime of a dynamically learned MAC address. By default the ageing timer is 300 seconds.	OPTIONAL

EXAMPLE:

```
=>bridge config
Ageing : 300
=>bridge config age=600
=>bridge config
Ageing : 600
=>
```

bridge flush

Flush bridge interfaces and parameters.

Note The flush command does not impact previously saved configurations.

SYNTAX:

```
bridge flush
```

bridge ifadd

Create a bridged Ethernet interface.

SYNTAX:

```
bridge ifadd          [intf = <string>]
                      [dest = <available ETHoA phonebook entries>]
```

[intf]	The bridged Ethernet interface name. If not specified, the destination parameter must be specified. In this case the name of the destination will double as interface name.	OPTIONAL
[dest]	The destination address for the new interface. Typically a phonebook entry. Browse through the available entries via the ARROW UP and ARROW DOWN keys.	OPTIONAL

EXAMPLE:

```
=>bridge iflist
OBC         :  Internal
               Connection State: connected  Port: OBC  PortState: forwarding
               RX bytes:  75783     frames:  572
               TX bytes:  82768372  frames:  341221  dropframes:  0
eth0        :  Internal
               Connection State: connected  Port: eth0  PortState: forwarding
               RX bytes:  156344216 frames:  5899238
               TX bytes:  75689     frames:  425     dropframes:  5558017
usb_bridge :  dest : usb_port
               Retry : 10  QoS : default  Encaps : llc/snap  Fcs : off
               Connection State: connected
               Port: wan2   PortNr: 4    PortState: forwarding
               RX bytes: 0          frames: 0
               TX bytes: 0          frames: 0            dropframes: 0
=>bridge ifadd intf=TestBridge dest=Br1
=>bridge iflist
OBC         :  Internal
               Connection State: connected  Port: OBC  PortState: forwarding
               RX bytes:  75783     frames:  572
               TX bytes:  82843610  frames:  341554  dropframes:  0
eth0        :  Internal
               Connection State: connected  Port: eth0  PortState: forwarding
               RX bytes:  156472129 frames:  5903256
               TX bytes:  75689     frames:  425     dropframes:  5561702
usb_bridge :  dest : usb_port
               Retry : 10  QoS : default  Encaps : llc/snap  Fcs : off
               Connection State: connected
               Port: wan2   PortNr: 4    PortState: forwarding
               RX bytes: 0          frames: 0
               TX bytes: 0          frames: 0            dropframes: 0
TestBridge :  dest : Br1
               Retry: 10  QoS: default  Encaps: llc/snap  Fcs: off
               Connection State: not-connected Port: (Unassigned)  PortState: forwarding
=>
```

RELATED COMMANDS:

bridge ifattach	Attach a bridge interface.
bridge ifdetach	Detach a bridge interface.
bridge ifdelete	Delete a bridge interface.
bridge ifconfig	Configure a bridge interface.
bridge iflist	Show current bridge configuration.

bridge ifattach

Attach (i.e. connect) a bridged Ethernet interface.

SYNTAX:

```
bridge ifattach      intf = <available Bridged Ethernet interfaces>
```

intf	The name of the interface to attach. Browse through the available entries via the ARROW UP and ARROW DOWN keys.	REQUIRED

EXAMPLE:

```
=>bridge iflist
OBC         :  Internal
               Connection State: connected  Port: OBC  PortState: forwarding
               RX bytes:  75783     frames:  572
               TX bytes:  82843610  frames:  341554  dropframes:  0
eth0        :  Internal
               Connection State: connected  Port: eth0  PortState: forwarding
               RX bytes:  156472129 frames:  5903256
               TX bytes:  75689     frames:  425     dropframes:  5561702
usb_bridge :  dest : usb_port
               Retry : 10  QoS : default  Encaps : llc/snap  Fcs : off
               Connection State: connected
               Port: wan2   PortNr: 4    PortState: forwarding
               RX bytes: 0          frames: 0
               TX bytes: 0          frames: 0            dropframes: 0
TestBridge :  dest : Br1
               Retry:  10    QoS:  default   Encaps:  llc/snap   Fcs:  off
               Connection State: not-
connected Port:  (Unassigned)                PortState:  forwarding
=>bridge ifattach intf=TestBridge
=>bridge iflist
OBC         :  Internal
               Connection State: connected  Port: OBC  PortState: forwarding
               RX bytes:  75783     frames:  572
               TX bytes:  82843610  frames:  341554  dropframes:  0
eth0        :  Internal
               Connection State: connected  Port: eth0  PortState: forwarding
               RX bytes:  156472129 frames:  5903256
               TX bytes:  75689     frames:  425     dropframes:  5561702
usb_bridge :  dest : usb_port
               Retry : 10  QoS : default  Encaps : llc/snap  Fcs : off
               Connection State: connected
               Port: wan2   PortNr: 4    PortState: forwarding
               RX bytes: 0          frames: 0
               TX bytes: 0          frames: 0            dropframes: 0
TestBridge :  dest : Br1
               Retry:  10   QoS:  default   Encaps:  llc/snap    Fcs:  off
               Connection State:  connected   Port:  wan0  PortState:  forwarding
               RX bytes:  75        frames:  12
               TX bytes:  30246     frames:  91     dropframes:  0
=>
```

RELATED COMMANDS:

bridge ifadd	Create a bridge interface.
bridge ifdetach	Detach a bridge interface.
bridge ifdelete	Delete a bridge interface.
bridge ifconfig	Configure a bridge interface.
bridge iflist	Show current bridge configuration.

bridge ifconfig

Configure a bridge interface.

SYNTAX:

```
bridge ifconfig      intf = <intfname>
                     [dest = <available interface name>]
                     [qos = <string>]
                     [encaps = <{llc/snap|vcmux}>]
                     [fcs = <{off|on}>]
                     [portstate = <{disabled|learning|forwarding}>]
                     [retry = <number {0-65535}>]
```

intf	The name of the bridge interface to configure.	REQUIRED
[dest]	The destination for this interface. Typically a phonebook entry. This parameter needs only to be specified in case of an interface created without specified destination.	OPTIONAL
[qos]	The name of the Quality Of Service book entry to apply on this bridge interface.	OPTIONAL
[encaps]	The type of encapsulation to be used for this bridge interface. Choose between: • llc/snap • vcmux	OPTIONAL
[fcs]	Whether or not to include the Ethernet FCS in the packet header on the WAN side. Choose between: • off • on	OPTIONAL
[portstate]	The bridge portstate for this interface. Choose between: • disabled • learning • forwarding	OPTIONAL
[retry]	A number between 0 and 65535. Represents the number of WAN connection setup retries before giving up. By default the retry value is 10.	OPTIONAL

Note In case of a SpeedTouch™530 device, never change, or delete the usb_bridge interface!

EXAMPLE:

```
=>bridge iflist intf=TestBridge
TestBridge : dest : Br1
             Retry: 10    QoS: default   Encaps: llc/snap    Fcs: off
             Connection State:  connected  Port:  wan0  PortState: forwarding
             RX bytes: 75          frames: 12
             TX bytes: 30246       frames: 91      dropframes: 0
=>bridge ifconfig intf=TestBridge encaps=vcmux retry=15
=>bridge iflist intf=TestBridge
TestBridge : dest : Br1
             Retry: 15    QoS: default   Encaps: vcmux       Fcs: off
             Connection State:  connected  Port:  wan0  PortState: forwarding
             RX bytes: 83          frames: 13
             TX bytes: 30740       frames: 102     dropframes: 0
=>
```

RELATED COMMANDS:

bridge ifadd	Create a bridge interface.
bridge ifattach	Attach a bridge interface.
bridge ifdetach	Detach a bridge interface.
bridge ifdelete	Delete a bridge interface.
bridge iflist	Show current bridge configuration.

bridge ifdelete

Delete a bridge interface.

SYNTAX:

`bridge ifdelete`	`intf = <available Bridged Ethernet interfaces>`

intf	The name of the interface name to delete. Browse through the available entries via the ARROW UP and ARROW DOWN keys.	REQUIRED

EXAMPLE:

```
=>bridge ifdelete intf=TestBridge
=>bridge iflist
OBC         :  Internal
               Connection State: connected  Port: OBC  PortState: forwarding
               RX bytes:  75783     frames:  572
               TX bytes:  82843610  frames:  341554  dropframes:  0
eth0        :  Internal
               Connection State: connected  Port: eth0  PortState: forwarding
               RX bytes:  156472129 frames:  5903256
               TX bytes:  75689     frames:  425     dropframes:  5561702
usb_bridge :  dest : usb_port
               Retry : 10  QoS : default  Encaps : llc/snap  Fcs : off
               Connection State: connected
               Port: wan2   PortNr: 4    PortState: forwarding
               RX bytes: 0          frames: 0
               TX bytes: 0          frames: 0            dropframes: 0
=>
```

Note In case of a SpeedTouch™530 device, never change, or delete the usb_bridge interface!

RELATED COMMANDS:

bridge ifadd	Create a bridge interface.
bridge ifattach	Attach a bridge interface.
bridge ifconfig	Configure a bridge interface.
bridge ifdetach	Detach a bridge interface.
bridge iflist	Show current bridge configuration.

bridge ifdetach

Detach (i.e. disconnect) a bridge interface.

SYNTAX:

```
bridge ifdetach      intf = <available Bridged Ethernet interfaces>
```

intf	The name of the bridge interface to detach. Browse through the available entries via the ARROW UP and ARROW DOWN keys.	REQUIRED

EXAMPLE:

```
=>bridge iflist intf=TestBridge
TestBridge :  dest : Br1
              Retry: 10   QoS: default   Encaps: llc/snap   Fcs: off
              Connection State: connected       Port: wan0            PortState: forwarding
              RX bytes: 75        frames: 12
              TX bytes: 30246     frames: 91     dropframes:  0
=>bridge ifattach intf=TestBridge
=>bridge iflist intf=TestBridge
TestBridge :  dest : Br1
              Retry: 10   QoS: default  Encaps: llc/snap   Fcs: off
              Connection State: not-connected   Port:  (unassigned)   PortState: forwarding
=>
```

Note In case of a SpeedTouch™530 device, never change, or delete the usb_bridge interface!

RELATED COMMANDS:

bridge ifadd	Create a bridge interface.
bridge ifattach	Attach a bridge interface.
bridge ifconfig	Configure a bridge interface.
bridge ifdelete	Delete a bridge interface.
bridge iflist	Show current bridge configuration.

bridge iflist

Show the current state of all or the selected bridge interfaces.

SYNTAX:

`bridge iflist`	`[intf = <available Bridged Ethernet interfaces>]`	
[intf]	The name of the bridge interface to show the configuration of. Browse through the available entries via the ARROW UP and ARROW DOWN keys. If not specified all bridge interfaces are shown.	OPTIONAL

EXAMPLE OUTPUT:

```
=>bridge iflist
OBC         :  Internal
               Connection State: connected  Port: OBC  PortState: forwarding
               RX bytes:  75783     frames:  572
               TX bytes:  82768372  frames:  341221  dropframes:  0
eth0        :  Internal
               Connection State: connected  Port: eth0  PortState: forwarding
               RX bytes:  156344216 frames:  5899238
               TX bytes:  75689     frames:  425     dropframes:  5558017
usb_bridge :  dest : usb_port
               Retry : 10  QoS : default  Encaps : llc/snap  Fcs : off
               Connection State: connected
               Port: wan2   PortNr: 4    PortState: forwarding
               RX bytes: 0          frames: 0
               TX bytes: 0          frames: 0            dropframes: 0
TestBridge :  dest : Br1
               Retry:  15    QoS:  default   Encaps:  vcmux   Fcs:  off
               Connection State: connected    Port:  wan0            PortState:  forwarding
               RX bytes:  83        frames:  13
               TX bytes:  30740     frames:  102    dropframes:  0
=>
```

Note In case of a SpeedTouch™530 device, never change, or delete the usb_bridge interface!

DESCRIPTION::

RX bytes	The number of Received bytes
TX bytes	The number of Transmitted bytes
OBC	Short for On Board Controller and indicates the physical bridge port.

RELATED COMMANDS:

bridge ifadd	Create a bridge interface.
bridge ifattach	Attach a created bridge interface.
bridge ifconfig	Configure a bridge interface.
bridge ifdelete	Delete a bridge interface.
bridge ifdetach	Detach a bridge interface.

bridge macadd

Add a static MAC address to the filtering database. Allows to manually add static addresses, which should normally be dynamically discovered by the bridge itself.

SYNTAX:

```
bridge macadd          intf = <available bridge interfaces>
                       hwaddr = <hardware-address>
```

intf	The name of the bridge interface to add the MAC address for.	REQUIRED
hwaddr	The MAC address of the new entry.	REQUIRED

EXAMPLE:

```
=>bridge maclist
00:90:d0:01:02:03   --   static,    OBC
ff:ff:ff:ff:ff:ff   --   static,    OBC
01:80:c2:00:00:00   --   static,    OBC
01:80:c2:00:00:01   --   static,    OBC
...
01:80:c2:00:00:10   --   static,    OBC
00:01:42:5f:7d:81   --   dynamic,   eth0,    597 seconds
00:50:8b:31:cc:aa   --   dynamic,   eth0,    513 seconds
08:00:20:c1:9a:12   --   dynamic,   eth0,    600 seconds
...
=>bridge macadd intf=eth0 hwaddr=00:80:9f:01:23:45
=>bridge maclist
00:90:d0:01:02:03   --   static,    OBC
ff:ff:ff:ff:ff:ff   --   static,    OBC
01:80:c2:00:00:00   --   static,    OBC
01:80:c2:00:00:01   --   static,    OBC
...
01:80:c2:00:00:10   --   static,    OBC
00:80:9f:01:23:45   --   permanent, eth0
00:01:42:5f:7d:81   --   dynamic,   eth0,    598 seconds
00:50:8b:31:cc:aa   --   dynamic,   eth0,    379 seconds
08:00:20:c1:9a:12   --   dynamic,   eth0,    600 seconds
00:08:c7:c3:5f:fc   --   dynamic,   eth0,    215 seconds
...
=>
```

RELATED COMMANDS:

bridge macdelete	Delete a MAC address entry.
bridge maclist	Show current filtering database.

bridge macdelete

Remove a MAC address from the filtering database.

SYNTAX:

```
bridge macdelete    hwaddr = <hardware-address>
```

hwaddr	The MAC address of the entry to delete.	REQUIRED

EXAMPLE:

```
=>bridge maclist
00:90:d0:01:02:03   --   static,    OBC
ff:ff:ff:ff:ff:ff   --   static,    OBC
01:80:c2:00:00:00   --   static,    OBC
01:80:c2:00:00:01   --   static,    OBC
...
01:80:c2:00:00:10   --   static,    OBC
00:80:9f:01:23:45   --   permanent, OBC
00:01:42:5f:7d:81   --   dynamic,   eth0,    597 seconds
00:50:8b:31:cc:aa   --   dynamic,   eth0,    513 seconds
08:00:20:c1:9a:12   --   dynamic,   eth0,    600 seconds
...
=>bridge macdelete hwaddr=00:80:9f:01:23:45
=>bridge maclist
00:90:d0:01:02:03   --   static,    OBC
ff:ff:ff:ff:ff:ff   --   static,    OBC
01:80:c2:00:00:00   --   static,    OBC
01:80:c2:00:00:01   --   static,    OBC
...
01:80:c2:00:00:10   --   static,    OBC
00:01:42:5f:7d:81   --   dynamic,   eth0,    598 seconds
00:50:8b:31:cc:aa   --   dynamic,   eth0,    379 seconds
08:00:20:c1:9a:12   --   dynamic,   eth0,    600 seconds
00:08:c7:c3:5f:fc   --   dynamic,   eth0,    215 seconds
...
=>
```

RELATED COMMANDS:

bridge macadd	Add a static MAC address entry.
bridge maclist	Show current filtering database.

bridge maclist

Show current MAC address filtering database.

SYNTAX:

```
bridge maclist
```

EXAMPLE:

```
=>bridge maclist
=>bridge maclist
00:90:d0:01:02:03   --   static,    OBC
ff:ff:ff:ff:ff:ff   --   static,    OBC
01:80:c2:00:00:00   --   static,    OBC
01:80:c2:00:00:01   --   static,    OBC
...
01:80:c2:00:00:10   --   static,    OBC
00:80:9f:24:ab:cf   --   static,    OBC
00:01:42:5f:7d:81   --   dynamic,   eth0,    598 seconds
00:50:8b:31:cc:aa   --   dynamic,   eth0,    379 seconds
08:00:20:c1:9a:12   --   dynamic,   eth0,    600 seconds
00:08:c7:c3:5f:fc   --   dynamic,   eth0,    215 seconds
08:00:20:a8:f4:34   --   dynamic,   eth0,    600 seconds
08:00:20:83:b7:26   --   dynamic,   eth0,    600 seconds
00:10:83:1b:13:18   --   dynamic,   eth0,    599 seconds
...
=>
```

RELATED COMMANDS:

bridge macadd	Add a static MAC address entry.
bridge macdelete	Delete a MAC address entry.

CIP Commands

Contents

This chapter covers the following commands:

cip flush

Flush complete Classical IP over ATM (IPoA) configuration.

Note The flush command does not impact previously saved configurations.

SYNTAX:

```
cip flush
```

cip ifadd

Create a Classical IPoA interface at the local side of the Logical IP Subnet (LIS).

SYNTAX:

```
cip ifadd          addr = <ip-address>
                   [netmask = <ip-mask (dotted or cidr)>]
                   [uniaddr = <portspec:address[.selector]>]
```

addr	The Classical IPoA interface's local IP address in the LIS.	REQUIRED
netmask	The LIS's subnetmask.	OPTIONAL
uniaddr	The UNI-address/port specification for incoming connections, e.g. 'A0:*.04': ADSL port, any address, selector 3. Only applicable in an Switched Virtual Channel (SVC) environment. In most cases the Classical IPoA LIS is built in a Permanent Virtual Channel (PVC) environment.	OPTIONAL

EXAMPLE:

```
=>cip iflist
cip1         addr = 172.16.0.5  mask = 255.255.255.0
             UNI address = A0:*.04
             inarp_reqs_in  =  0   inarp_repl_in  =  0   inarp_inv_in  =  0
             inarp_reqs_out =  0   inarp_repl_out =  0   inarp_inv_out =  0
=>cip ifadd addr=172.16.1.1 netmask=255.255.255.0
=>cip iflist
cip1         addr = 172.16.0.5  mask = 255.255.255.0
             UNI address = A0:*.04
             inarp_reqs_in  =  0   inarp_repl_in  =  0   inarp_inv_in  =  0
             inarp_reqs_out =  0   inarp_repl_out =  0   inarp_inv_out =  0
cip0         addr = 172.16.1.1  mask = 255.255.255.0
             UNI address = A0:*.03
             inarp_reqs_in  =  0   inarp_repl_in  =  0   inarp_inv_in  =  0
             inarp_reqs_out =  0   inarp_repl_out =  0   inarp_inv_out =  0
=>
```

RELATED COMMANDS:

cip ifdelete	Delete a Classical IPoA interface.
cip ifadd	Show current Classical IPoA configuration.

cip ifdelete

Delete a Classical IPoA interface at the local side of the Logical IP Subnet (LIS).

SYNTAX:

```
cip ifdelete          addr = <ip-address>
```

addr	The Classical IPoA interface's local IP address in the LIS.	REQUIRED

EXAMPLE:

```
=>cip iflist
cip0         addr = 172.16.1.1  mask = 255.255.255.0
             UNI address = A0:*.03
             inarp_reqs_in  =  0   inarp_repl_in  =  0   inarp_inv_in  =  0
             inarp_reqs_out =  0   inarp_repl_out =  0   inarp_inv_out =  0
cip1         addr = 172.16.0.5  mask = 255.255.255.0
             UNI address = A0:*.04
             inarp_reqs_in  =  0   inarp_repl_in  =  0   inarp_inv_in  =  0
             inarp_reqs_out =  0   inarp_repl_out =  0   inarp_inv_out =  0
=>cip ifdelete addr=172.16.1.1
=>cip iflist
cip1         addr = 172.16.0.5  mask = 255.255.255.0
             UNI address = A0:*.04
             inarp_reqs_in  =  0   inarp_repl_in  =  0   inarp_inv_in  =  0
             inarp_reqs_out =  0   inarp_repl_out =  0   inarp_inv_out =  0
=>
```

RELATED COMMANDS:

cip ifadd	Create a Classical IPoA interface.
cip iflist	Show current Classical IPoA configuration.

cip iflist

Show current Classical IPoA configuration.

SYNTAX:

```
cip iflist
```

EXAMPLE OUTPUT:

```
=>cip iflist
cip0         addr = 172.16.1.1  mask = 255.255.255.0
             UNI address = A0:*.03
             inarp_reqs_in  =  0   inarp_repl_in  =  0   inarp_inv_in  =  0
             inarp_reqs_out =  0   inarp_repl_out =  0   inarp_inv_out =  0
cip1         addr = 172.16.0.5  mask = 255.255.255.0
             UNI address = A0:*.04
             inarp_reqs_in  =  0   inarp_repl_in  =  0   inarp_inv_in  =  0
             inarp_reqs_out =  0   inarp_repl_out =  0   inarp_inv_out =  0
=>
```

DESCRIPTION:

inarp_reqs_in/inarp_reqs_out	Incoming/outgoing inverse ARP requests
inarp_repl_in/inarp_repl_out	Incoming/outgoing inverse ARP replies
inarp_inv_in/inarp_inv_out	Incoming/outgoing invalid inverse ARP messages

EXAMPLE INPUT/OUTPUT: EVOLUTION OF ARP REQUESTS IN A NETWORKED ENVIRONMENT:

```
=>cip iflist
cip0         addr = 200.200.200.138  mask = 255.255.255.0
             UNI address = A0:*.03
             inarp_reqs_in  = 18   inarp_repl_in  = 75   inarp_inv_in  =  0
             inarp_reqs_out = 18   inarp_repl_out = 75   inarp_inv_out =  0
=>cip iflist
cip0         addr = 200.200.200.138  mask = 255.255.255.0
             UNI address = A0:*.03
             inarp_reqs_in  = 22   inarp_repl_in  = 75   inarp_inv_in  =  0
             inarp_reqs_out = 22   inarp_repl_out = 75   inarp_inv_out =  0
=>cip iflist
cip0         addr = 200.200.200.138  mask = 255.255.255.0
             UNI address = A0:*.03
             inarp_reqs_in  = 22   inarp_repl_in  = 76   inarp_inv_in  =  0
             inarp_reqs_out = 22   inarp_repl_out = 76   inarp_inv_out =  0
=>
```

RELATED COMMANDS:

cip ifadd	Create a Classical IPoA interface.
cip ifdelete	Delete a Classical IPoA interface.

cip pvcadd

Create a PVC Address Resolution Protocol (ARP) entry for destinations which are not RFC 1577/RFC2225 compliant.

SYNTAX:

```
cip pvcadd          dest = <phonebookname>
                    [destaddr = <ip-address>]
                    [mtu = <number {273-20000}>]
```

dest	The ATM address (hardware address) of the destination host. Typically a phonebook name.	REQUIRED
[destaddr]	The IP address of the destination host.	OPTIONAL
[mtu]	A number between 273 and 20000 (bytes). Represents the maximum ATM Adaption Layer 5 (AAL5) packet size for this connection. By default the mtu is 9180 bytes.	OPTIONAL

EXAMPLE:

```
=>phonebook list
Name         Type    Use   Address
Br1          bridge  1     8.35
Br2          bridge  1     8.36
Br3          bridge  1     8.37
Br4          bridge  0     8.38
RELAY_PPP1   ppp     0     8.48
RELAY_PPP2   ppp     0     8.49
RELAY_PPP3   ppp     0     8.50
RELAY_PPP4   ppp     0     8.51
PPP1         ppp     1     8.64
PPP2         ppp     1     8.65
PPP3         ppp     1     8.66
DHCP_SPOOF   ppp     1     8.67
CIPPVC1      cip     0     8.80
CIPPVC2      cip     0     8.81
CIPPVC3      cip     0     8.82
CIPPVC4      cip     0     8.83
=>cip pvclist
=>cip pvcadd dest CIPPVC1 destaddr 172.16.1.2 mtu 546
=>cip pvclist
CIPPVC1       atmport  = 0     vpi  = 8    vci  = 80   dest_ip  = 172.16.1.2
              encaps   = llc   mtu  = 546
=>
```

RELATED COMMANDS:

cip pvcdelete	Delete a PVC ARP entry.
cip pvclist	Show current PVC ARP entries.

cip pvcdelete

Delete a PVC ARP entry.

SYNTAX:

`cip pvcdelete`	`dest = <phonebookname>`

dest	Typically a phonebook entry name. Represents the ATM address (hardware address) or name of the entry to delete.	REQUIRED

EXAMPLE:

```
=>cip pvclist
CIPPVC1       atmport  = 0     vpi  = 8    vci  = 80   dest_ip  = 172.16.1.2
              encaps   = llc   mtu  = 546
=>cip pvcdelete dest=CIPPVC1
=>cip pvclist
=>
```

RELATED COMMANDS:

cip pvcadd	Create a PVC ARP entry.
cip pvclist	Show current PVC ARP entries.

cip pvclist

Show current PVC ARP entries.

SYNTAX:

```
cip pvclist
```

EXAMPLE OUTPUT:

```
=>cip pvclist
CIPPVC1       atmport = 0     vpi = 8    vci = 80   dest_ip = 172.16.1.2
              encaps  = llc   mtu = 546
=>
```

EXAMPLE INPUT/OUTPUT IN A NETWORKED ENVIRONMENT:

```
=>cip iflist
cip0         addr = 200.200.200.138  mask = 255.255.255.0
             UNI address = A0:*.03
             inarp_reqs_in  =  0   inarp_repl_in  = 75   inarp_inv_in  =  0
             inarp_reqs_out =  0   inarp_repl_out = 75   inarp_inv_out =  0
=>cip pvclist
699           atmport = 0     vpi = 6    vci = 99   dest_ip = 172.16.1.3
              encaps  = llc   mtu = 9180
8.50          atmport = 0     vpi = 8    vci = 50   dest_ip = 200.200.200.14
              encaps  = llc   mtu = 9180
=>
```

RELATED COMMANDS:

cip pvcdelete	Delete a PVC ARP entry.
cip pvcadd	Create a PVC ARP entry.

Config Commands

Contents

This chapter covers the following commands:

config erase

Physically remove all saved configurations, i.e. the user configuration file, saved in the SpeedTouch™ permanent storage, is deleted.

SYNTAX:

```
config erase
```

RELATED COMMANDS:

config flush	Flush complete runtime configuration.
config load	Load complete saved or default configuration.
config save	Save complete runtime configuration.

config flush

Flush complete current configuration without affecting saved configurations.

This combines all flush commands: bridge flush, cip flush, dhcp client flush, dhcp server flush, dns flush, env flush, firewall flush, firewall rule flush, ipoa flush, mer flush, nat flush, phonebook flush, ppp flush, pptp flush, qosbook flush, system flush and optionally ip flush.

SYNTAX:

`config flush`	`[flush_ip = <{no|yes}>]`

[flush_ip]	Keep current IP configuration (yes) or not (no). Not keeping the IP settings could cause lost IP connectivity in the LAN. By default IP settings are preserved.	OPTIONAL

EXAMPLE:

```
=>ip rtlist
   Destination        Source        Gateway        Intf     Mtrc
   10.0.0.0/24        10.0.0.0/24   10.0.0.140     eth0     0
   172.16.0.5/32      0.0.0.0/0     172.16.0.5     cip1     0
   10.0.0.140/32      0.0.0.0/0     10.0.0.140     eth0     0
   127.0.0.1/32       0.0.0.0/0     127.0.0.1      loop     0
   10.0.0.0/24        0.0.0.0/0     10.0.0.140     eth0     0
   172.16.0.0/24      0.0.0.0/0     172.16.0.5     cip1     1
=>config flush flush_ip=no
=>ip rtlist
   Destination        Source        Gateway        Intf     Mtrc
   10.0.0.0/24        10.0.0.0/24   10.0.0.140     eth0     0
   10.0.0.140/32      0.0.0.0/0     10.0.0.140     eth0     0
   127.0.0.1/32       0.0.0.0/0     127.0.0.1      loop     0
   10.0.0.0/24        0.0.0.0/0     10.0.0.140     eth0     0
=>config flush flush_ip=yes

######### ALL TCP/IP CONNECTIVITY IS LOST ##########
```

RELATED COMMANDS:

config erase	Physically remove all saved configurations.
config load	Load complete saved or default configuration.
config save	Save current runtime configuration.

config load

Load complete saved or default configuration.

Note Execute config flush prior to config load.

SYNTAX:

```
config load          [load_ip = <{no|yes}>]
                     [defaults = <{yes|no}>]
```

[load_ip]	Load IP settings (yes) or not (no). Not keeping the IP settings could cause lost IP connectivity in the LAN.	OPTIONAL
[defaults]	Load default configuration (yes) or saved configuration (no). Not specifying this parameter loads the saved configuration	OPTIONAL

EXAMPLE:

```
=>ip rtlist
   Destination        Source        Gateway        Intf     Mtrc
   10.0.0.0/24        10.0.0.0/24   10.0.0.140     eth0     0
   172.16.0.5/32      0.0.0.0/0     172.16.0.5     cip1     0
   10.0.0.140/32      0.0.0.0/0     10.0.0.140     eth0     0
   127.0.0.1/32       0.0.0.0/0     127.0.0.1      loop     0
   10.0.0.0/24        0.0.0.0/0     10.0.0.140     eth0     0
   172.16.0.0/24      0.0.0.0/0     172.16.0.5     cip1     1
=>config flush flush_ip=no
=>ip rtlist
   Destination        Source        Gateway        Intf     Mtrc
   10.0.0.0/24        10.0.0.0/24   10.0.0.140     eth0     0
   10.0.0.140/32      0.0.0.0/0     10.0.0.140     eth0     0
   127.0.0.1/32       0.0.0.0/0     127.0.0.1      loop     0
   10.0.0.0/24        0.0.0.0/0     10.0.0.140     eth0     0
=>config load load_ip=yes
=>ip rtlist
   Destination        Source        Gateway        Intf     Mtrc
   10.0.0.0/24        10.0.0.0/24   10.0.0.140     eth0     0
   10.0.0.140/32      0.0.0.0/0     10.0.0.140     eth0     0
   172.16.0.5/32      0.0.0.0/0     172.16.0.5     cip1     0
   127.0.0.1/32       0.0.0.0/0     127.0.0.1      loop     0
   10.0.0.0/24        0.0.0.0/0     10.0.0.140     eth0     0
   172.16.0.0/24      0.0.0.0/0     172.16.0.5     cip1     1
=>
```

RELATED COMMANDS:

config erase	Physically remove all saved configurations.
config flush	Flush complete runtime configuration.
config save	Save current runtime configuration.

config save

Save all existing configurations and modifications entered by the user.
The result of executing this command is a user.ini file saved in the SpeedTouch™ permanent storage. This file can be downloaded via the SpeedTouch™ web pages or via an FTP session.

SYNTAX:

```
config save
```

EXAMPLE:

```
=>config save
=>
```

RELATED COMMANDS:

config erase	Physically remove all saved configurations.
config flush	Flush complete current configuration.
config load	Load complete saved or default configuration.

DHCP Commands

Contents

dhcp client clear

Clear Dynamic Host Configuration Protocol (DHCP) client statistics.

SYNTAX:

```
dhcp client clear
```

EXAMPLE:

```
=>dhcp client stats
DHCP client statistics:
Corrupted packet recv   :  0
OFFERs     recv         :  0
ACKs     recv           :  0
NAKs     recv           :  0
Pure BOOTP REPLIES      :  0
Other message types     :  0
DISCOVERs sent          :  253
REQUESTs sent           :  9
DECLINEs sent           :  0
RELEASEs  sent          :  0
INFORMs   sent          :  0
Number of dynamic interfaces:  1
Memory usage:
Table size of dyn leases: 19,  in use: 1,  free: 94 %
=>dhcp client clear
=>dhcp client stats
DHCP client statistics:
Corrupted packet recv   :  0
OFFERs     recv         :  0
ACKs     recv           :  0
NAKs     recv           :  0
Pure BOOTP REPLIES      :  0
Other message types     :  0
DISCOVERs sent          :  0
REQUESTs sent           :  0
DECLINEs sent           :  0
RELEASEs  sent          :  0
INFORMs   sent          :   0
Number of dynamic interfaces:  1
Memory usage:
Table size of dyn leases: 19,  in use: 1,  free: 94 %
=>
```

RELATED COMMANDS:

dhcp client stats	Show DHCP client statistics.

dhcp client config

Show/set DHCP client configuration.

SYNTAX:

```
dhcp client config    [trace = <{off|on}>]
```

[trace]	Enable tracing (on) or not (off).	OPTIONAL

EXAMPLE:

```
=>dhcp client config
tracing: off
=>dhcp client config trace=on
=>dhcp client config
tracing: on
=>
```

RELATED COMMANDS:

dhcp client ifconfig	Configure a DHCP lease created for a specific interface.

dhcp client flush

Flush complete DHCP client configuration and dynamic interfaces.

Note The flush command does not impact previously saved configurations.

SYNTAX:

```
dhcp client flush
```

EXAMPLE:

```
=>dhcp client iflist
NewMer     :  [SELECTING]
              flags=  uc
              IP address    :  10.0.0.10
              HW address    :  0:90:d0:01:47:de
              DHCP server   :  255.255.255.255
              hostname      :  NewLease
              req.leasetime=   10800 s
              trying to get a lease for 8 min, 32 sec
              transmission of DISCOVER in 57 sec
              retransmission timeout:  64
              nbr of retransmissions:  14
Number of leases:  1
Total size of table: 19,  in use: 1,  free: 94 %
=>dhcp client flush
=>dhcp client iflist
No dynamic interfaces defined.
=>
```

dhcp client ifadd

Create a DHCP lease for a specific interface.

SYNTAX:

```
dhcp client ifadd     intf = <interface name>
```

intf	The name of an existing interface, e.g. created via :mer ifadd.	REQUIRED

EXAMPLE:

```
=>dhcp client iflist
No dynamic interfaces defined.
=>dhcp client ifadd intf=NewMer
=>dhcp client iflist
NewMer    :  [INIT]
             flags=  uc
             IP address    :  0.0.0.0
             HW address    :  00:90:d0:01:47:de
             DHCP server   :  255.255.255.255
Number of leases:     1
Total size of table: 19,  in use: 1,  free: 94 %
=>
```

RELATED COMMANDS:

dhcp client ifattach	Attach a DHCP lease to an interface.
dhcp client ifconfig	Configure a DHCP lease created for a specific interface.
dhcp client ifdelete	Delete a dynamic interface.
dhcp client iflist	Show all dynamic interfaces.

dhcp client ifattach

Attach a DHCP lease to a dynamic interface.

Note Create the interface first with :dhcp client ifadd.

SYNTAX:

```
dhcp client ifattach    intf = <interface name>
```

intf	The name of the dynamic interface.	REQUIRED

EXAMPLE:

```
=>dhcp client iflist
NewMer    :  [INIT]
             flags=  uc
             IP address      :  0.0.0.0
             HW address      :  00:90:d0:01:47:de
             DHCP server     :  255.255.255.255
Number of leases:  1
Total size of table: 19,   in use: 1,   free: 94 %
=>dhcp client ifattach intf=NewMer
=>dhcp client iflist
NewMer    :  [SELECTING]
             flags=  uc
             IP address      :  10.0.0.10
             HW address      :  0:90:d0:01:47:de
             DHCP server     :  255.255.255.255
             hostname        :  NewLease
             req.leasetime   =  10800 s
             trying to get a lease for 8 min, 32 sec
             transmission of DISCOVER in 57 sec
             retransmission timeout: 64
             nbr of retransmissions: 14
Number of leases:  1
Total size of table: 19,   in use: 1,   free: 94 %
=>
```

RELATED COMMANDS:

dhcp client ifadd	Create a DHCP lease for a specific interface.
dhcp client ifconfig	Configure a DHCP lease created for a specific interface.
dhcp client ifrelease	Release a lease attached to a dynamic interface.
dhcp client iflist	Show all dynamic interfaces.

dhcp client ifconfig

Show/set the configuration of DHCP lease created for a specific interface.

Note Execute the dhcp client ifrelease command before configuring the dhcp client.

SYNTAX:

```
dhcp client ifconfig    intf = <interface name>
                        [clientid = <{client-id|none}>]
                        [hostname = <hostname|"">]
                        [addr = <ip-address>]
                        [leasetime = <number>]
                        [addrtrans = <{none|pat}>]
                        [dns = <{off|on}>]
                        [gateway = <{off|on}>]
                        [metric = <number{0-100}>]
```

intf	The name of the dynamic interface to be configured.	REQUIRED
[clientid]	The client identity to be associated with the lease. Use none in case no clientid should be associated with this lease.	OPTIONAL
[hostname]	The host name of the client to be associated with the lease. Use “” in case no hostname should not be associated with this lease.	OPTIONAL
[addr]	The preferred dynamic IP address.	OPTIONAL
[leasetime]	A number between 0 and 1814400 (seconds). Represents the preferred time the client wants to use an address. By default the leasetime is 7200 seconds (2 hours). Specifying -1 makes the lease permanent.	OPTIONAL
[addrtrans]	Automatically enable address translation for this dynamic interface (pat) or not (none).	OPTIONAL
[dns]	Request (and accept) DNS server IP addresses (on) or not (off).	OPTIONAL
[gateway]	Request (and accept) gateway IP addresses (on) or not (off).	OPTIONAL
[metric]	A number between 10 and 100000. Represents the gateway route metric By default the gateway route metric is 1.	OPTIONAL

EXAMPLE:

```
=>dhcp client iflist
NewMer    :  [INIT]
             flags=  uc
             IP address       :  0.0.0.0
             HW address       :  00:90:d0:01:47:de
             DHCP server      :  255.255.255.255
Number of leases:  1
Total size of table: 19,   in use: 1,   free: 4 %
=>dhcp client ifconfig intf=NewMer hostname=NewLease addr=10.0.0.10 leasetime=10800
=>dhcp client iflist
NewMer    :  [INIT]
             flags=  uc
             IP address       :  10.0.0.10
             HW address       :  00:90:d0:01:47:de
             DHCP server      :  255.255.255.255
             hostname         :  NewLease
             req.leasetime    =  10800 s
Number of leases:  1
Total size of table: 19,   in use: 1,   free: 4 %
=>
```

RELATED COMMANDS:

dhcp client ifadd	Create a DHCP lease for a specific interface.
dhcp client ifdelete	Delete a dynamic interface.
dhcp client iflist	Show all dynamic interfaces.
dhcp client ifrelease	Release a lease attached to a dynamic interface.

dhcp client ifdelete

Delete a dynamic interface.

SYNTAX:

```
dhcp client ifdelete    intf = <interface name>
```

intf	The name of the dynamic interface.	REQUIRED

EXAMPLE:

```
=>dhcp client iflist
NewMer    :  [SELECTING]
             flags=  uc
             IP address       :  10.0.0.10
             HW address       :  00:90:d0:01:47:de
             DHCP server      :  255.255.255.255
             hostname         :  NewLease
             req.leasetime    =  10800 s
             trying to get a lease for 8 min, 32 sec
             transmission of DISCOVER in 57 sec
             retransmission timeout:  64
             nbr of retransmissions:  14
Number of leases:  1
Total size of table: 19,   in use: 1,   free: 94 %
=>dhcp client ifdelete intf NewMer
=>dhcp client iflist
No dynamic interfaces defined.
=>
```

RELATED COMMANDS:

dhcp client ifadd	Create a DHCP lease for a specific interface.
dhcp client ifattach	Attach a DHCP lease to an interface.
dhcp client ifconfig	Configure a DHCP lease created for a specific interface.
dhcp client iflist	Show all dynamic interfaces.
dhcp client ifrelease	Release a lease attached to a dynamic interface.

dhcp client iflist

Show all dynamic interfaces.

SYNTAX:

```
dhcp client iflist
```

EXAMPLE:

```
=>dhcp client iflist
NewMer    :  [INIT]
             flags=  uc
             IP address       :  0.0.0.0
             HW address       :  00:90:d0:01:47:de
             DHCP server      :  255.255.255.255
Number of leases:  1
Total size of table: 19,   in use: 1,   free: 94 %
=>
```

EXAMPLE INPUT/OUTPUT IN A NETWORKED ENVIRONMENT:

The SpeedTouch™ is configured as DHCP client on its Ethernet interface eth0.

```
=>dhcp client iflist
eth0      :  [BOUND]
             flags=  uc
             IP address     :  10.0.0.3
             HW address     :  00:90:d0:01:47:f1
             DHCP server    :  10.10.1.1
             lease renewal in      5  days, 1  h, 26 min, 45 sec
             lease rebinding in    8  days, 20 h, 34 min, 15 sec
             lease expires in      10 days, 2  h, 56 min, 45 sec
Number of leases:  1
Total size of table: 18,   in use: 1,   free: 94 %
=>dhcp client iflist
eth0      :  [BOUND]
             flags=  uc
             IP address     :  10.0.0.3
             HW address     :  00:90:d0:01:47:f1
             DHCP server    :  10.10.1.1
             lease renewal in      5  days, 1  h, 25 min, 27 sec
             lease rebinding in    8  days, 20 h, 32 min, 57 sec
             lease expires in      10 days, 2  h, 55 min, 27 sec
Number of leases:  1
Total size of table: 18,   in use: 1,   free: 94 %
=>
```

RELATED COMMANDS:

dhcp client ifadd	Create a DHCP lease for a specific interface.
dhcp client ifdelete	Delete a dynamic interface.

dhcp client ifrelease

Release a lease attached to a dynamic interface.

SYNTAX:

```
dhcp client ifrelease   intf = <interface name>
```

intf	The name of the dynamic interface.	REQUIRED

EXAMPLE:

```
=>dhcp client iflist
NewMer    :  [SELECTING]
             flags=  uc
             IP address       :  10.0.0.10
             HW address       :  00:90:d0:01:47:de
             DHCP server      :  255.255.255.255
             hostname         :  NewLease
             req.leasetime    =  10800 s
             trying to get a lease for 8 min, 32 sec
             transmission of DISCOVER in 57 sec
             retransmission timeout:  64
             nbr of retransmissions:  14
Number of leases:  1
Total size of table: 19,   in use: 1,   free: 94 %
=>dhcp client ifattach intf=NewMer
=>dhcp client iflist
NewMer    :  [SELECTING]
             flags=  uc
             IP address       :  0.0.0.0
             HW address       :  00:90:d0:01:47:de
             DHCP server      :  255.255.255.255
             hostname         :  NewLease
             req.leasetime    =  10800 s
Number of leases:  1
Total size of table: 19,   in use: 1,   free: 94 %
=>
```

EXAMPLE INPUT/OUTPUT IN A NETWORKED ENVIRONMENT:

The SpeedTouch™ is configured as DHCP client on its Ethernet interface eth0.

```
=>dhcp client iflist
eth0      :  [BOUND]
             flags=  uc
             IP address    :  10.0.0.3
             HW address    :  00:90:d0:01:47:f1
             DHCP server   :  10.10.1.1
             lease renewal in      5  days, 58 min, 45 sec
             lease rebinding in    8  days, 20 h, 6  min, 18 sec
             lease expires in      10 days, 2  h, 28 min, 48 sec
Number of leases:  1
Total size of table: 18,   in use: 1,   free: 94 %
=>dhcp client stats
DHCP client statistics:
Corrupted packet recv   :               0
DECLINEs sent           :               0
RELEASEs  sent          :               0
INFORMs   sent          :               0
Number of dynamic interfaces:  1
Memory usage:
Table size of dyn leases: 19,   in use: 1,   free: 94 %
=>dhcp client ifrelease intf=eth0
=>(CTRL + Q)
=>STATE ACTIVATE !
STATE IDLE !
STATE ACTIVATE !
dhcc: intf 1  releases 10.0.0.3 to server 10.10.1.1.
dhcc: 10.0.0.3 deleted: ok.
STATE IDLE !
STATE ACTIVATE !
.............
dhcc: intf 1 in init state.
n_send() broadcast triggered; To be verified
dhcc: broadcast discover on intf 1.
=>(CTRL + S)
=>dhcp client stats
DHCP client statistics:
Corrupted packet recv   :               0
DECLINEs sent           :               0
RELEASEs  sent          :               1
INFORMs   sent          :               0
Number of dynamic interfaces:  1
Memory usage:
Table size of dyn leases: 19,   in use: 1,   free: 94 %
=>
```

RELATED COMMANDS:

dhcp client ifattach	Attach a DHCP lease to an interface.
dhcp client ifconfig	Configure a DHCP lease created for a specific interface.
dhcp client ifdelete	Delete a dynamic interface.

dhcp client ifrenew

Renew the lease of a dynamic interface.

SYNTAX:

```
dhcp client ifrenew  intf = <interface name>
```

intf	The name of the dynamic interface.	REQUIRED

EXAMPLE:

```
=>dhcp client iflist
NewMer    :  [BOUND]
             flags=  uc
             IP address    :  10.0.0.10
             HW address    :  00:90:d0:01:47:f1
             DHCP server   :  255.255.255.255
             hostname      :  NewLease
             req.leasetime =  10800 s
             lease renewal in      5  days, 58 min, 48 sec
             lease rebinding in    8  days, 20 h, 6  min, 18 sec
             lease expires in      10 days, 2  h, 28 min, 48 sec
Number of leases:  1
Total size of table: 19,   in use: 1,   free: 94 %
=>dhcp client ifrenew intf=NewMer
=>dhcp client iflist
NewMer    :  [SELECTING]
             flags=  uc
             IP address      :  10.0.0.10
             HW address      :  00:90:d0:01:47:de
             DHCP server     :  255.255.255.255
             hostname        :  NewLease
             req.leasetime   =  10800 s
             trying to get a lease for 12 sec
             transmission of DISCOVER in 24 sec
             retransmission timeout:  64
             nbr of retransmissions:  11
Number of leases:  1
Total size of table: 19,   in use: 1,   free: 94 %
=>
```

EXAMPLE INPUT/OUTPUT IN A NETWORKED ENVIRONMENT:

The SpeedTouch™ is configured as DHCP client on its Ethernet interface eth0.

```
=>dhcp client stats
DHCP client statistics:
Corrupted packet recv   :               0
OFFERs     recv         :               0
ACKs     recv           :               0
NAKs     recv           :               0
Pure BOOTP REPLIES      :               0
Other message types     :               0
DISCOVERs sent          :               0
REQUESTs sent           :               0
DECLINEs sent           :               0
RELEASEs  sent          :               1
INFORMs   sent          :               0
Number of dynamic interfaces:  1
Memory usage:
Table size of dyn leases: 18,   in use: 1,   free: 94 %
=>dhcp client ifrenew intf=eth0
=>dhcp client stats
DHCP client statistics:
Corrupted packet recv   :               0
OFFERs     recv         :               1
ACKs     recv           :               1
NAKs     recv           :               0
Pure BOOTP REPLIES      :               0
Other message types     :               0
DISCOVERs sent          :               1
REQUESTs sent           :               1
DECLINEs sent           :               0
RELEASEs  sent          :               1
INFORMs   sent          :               0
Number of dynamic interfaces:  1
Memory usage:
Table size of dyn leases: 18,   in use: 1,   free: 94 %
=>(CTRL + Q)
..................
STATE IDLE !
STATE ACTIVATE !
dhcc: intf 1 renews lease 10.0.0.3.
dhcc: intf 1 requests 10.0.0.3 from 10.10.1.1
dhcc: 10.10.1.1 acks 10.0.0.3 to intf 1.
dhcc: lease 10.0.0.3 bound to intf 1.
STATE IDLE !
STATE ACTIVATE !
...........
=>(CTRL + S)
```

RELATED COMMANDS:

dhcp client ifadd	Create a DHCP lease for a specific interface.
dhcp client ifattach	Attach a DHCP lease to an interface.

dhcp client stats

Show DHCP client statistics.

SYNTAX:

```
dhcp client stats
```

EXAMPLE:

```
=>dhcp client stats
DHCP client statistics:
Corrupted packet recv   :               0
OFFERs     recv         :               1
ACKs     recv           :               1
NAKs     recv           :               0
Pure BOOTP REPLIES      :               0
Other message types     :               0
DISCOVERs sent          :               244
REQUESTs sent           :               9
DECLINEs sent           :               0
RELEASEs  sent          :               1
INFORMs   sent          :               0
Number of dynamic interfaces:  1
Memory usage:
Table size of dyn leases: 19,   in use: 1,   free: 94 %
=>
```

RELATED COMMANDS:

dhcp client clear	Clear DHCP client statistics.

dhcp server clear

Clear SpeedTouch™ DHCP server statistics.

SYNTAX:

```
dhcp server clear
```

EXAMPLE:

```
=>dhcp client stats
Corrupted packet recv   :               0
OFFERs     recv         :               9575
ACKs     recv           :               121
NAKs     recv           :               0
Pure BOOTP REPLIES      :               0
Other message types     :               0
DISCOVERs sent          :               9552
REQUESTs sent           :               142
DECLINEs sent           :               0
RELEASEs  sent          :               0
INFORMs   sent          :               0
Number of dynamic interfaces:  1
Memory usage:
Table size of dyn leases: 19,   in use: 1,   free: 94 %
=>dhcp server clear
=>dhcp client stats
DHCP client statistics:
Corrupted packet recv   :               0
OFFERs     recv         :               0
ACKs     recv           :               0
NAKs     recv           :               0
Pure BOOTP REPLIES      :               0
Other message types     :               0
DISCOVERs sent          :               0
REQUESTs sent           :               0
DECLINEs sent           :               0
RELEASEs  sent          :               0
INFORMs   sent          :               0
Number of dynamic interfaces:  1
Memory usage:
Table size of dyn leases: 19,   in use: 1,   free: 94 %
=>
```

RELATED COMMANDS:

dhcp server stats	Show DHCP server statistics.

dhcp server config

Show/set SpeedTouch™ DHCP server configuration settings.

SYNTAX:

```
dhcp server config    [autodhcp = <{off|on}>]
                      [scantime = <number>]
                      [spoofing = <{off|on}>]
                      [trace = <{off|on}>]
```

[autodhcp]	Allow the SpeedTouch™ to present itself as DHCP client (AutoDHCP mode) at boot time and probe for another DHCP server on the network for some time before starting its own DHCP server (yes) or immediately start the DHCP server (no).	OPTIONAL
[scantime]	A number between 0 and 1814400 (seconds). Represents the time the SpeedTouch™ scans for another DHCP server to be active in the network. By default the scantime is 20 seconds.	OPTIONAL
[spoofing]	Allow a remote DHCP server to hand out IP addresses negotiated by PPP on WAN side (yes) or not (no). DHCP spoofing is used to relay local DHCP requests to an external PPP connection having a specific IP address negotiation mechanism. DHCP replies are in turn generated by the DHCP server based on the IP address information received by the PPP link.	OPTIONAL
[trace]	Disable verbose console logging and generation of debug traces (off) or enable verbose console logging and generation of debug traces (on). By default tracing is disabled.	OPTIONAL

EXAMPLE:

```
=>dhcp server config
autodhcp:     on
scantime:     20s
spoofing:     off
tracing:      off
=>dhcp server config scantime=30 tracing=on
=>dhcp server config
autodhcp:     on
scantime:     30s
spoofing:     off
tracing:      on
=>
```

RELATED COMMANDS:

dhcp server stats	Show current DHCP server state and statistics.

dhcp server flush

Flushes all DHCP server pool and lease entries.

Note The flush command does not impact previously saved configurations.

SYNTAX:

```
dhcp server flush
```

RELATED COMMANDS:

dhcp server stats	Show current DHCP server state and statistics.
dhcp server config	Show/set current DHCP server configuration.

dhcp server policy

Show/set SpeedTouch™ DHCP server policy.

SYNTAX:

```
dhcp server policy    [verifyfirst = <off|on>]
                      [trustclient = <off|on>]
```

[verifyfirst]	Probe the network for conflicting IP addresses before giving a suggested IP address to the requesting DHCP client (on) or not (off).	OPTIONAL
[trustclient]	Take the IP address suggested by a DHCP client into account (on) or not (off).	OPTIONAL

EXAMPLE:

```
=>dhcp server policy
Verify first:   off
Trust client:   on
=>dhcp server policy verifyfirst=on trustclient=off
=>dhcp server policy
Verify first:   on
Trust client:   off
=>
```

RELATED COMMANDS:

dhcp server stats	Show current DHCP server state and statistics.
dhcp server config	Show/set current DHCP server configuration.

dhcp server spoof

Show/set DHCP spoofing parameters. Only applicable in case of a PPP-to-DHCP Spoofing connection. (See dhcp server config command).

SYNTAX:

```
dhcp server spoof    [failtime = <number>]
                     [errorlt = <number>]
                     [dodlt = <number>]
```

[failtime]	A number between 0 and 1814400 (seconds). Represents the time to wait for a PPP link to successfully negotiate an IP address. This parameter determines how long the SpeedTouch™ should try to set up a PPP connection before returning to normal DHCP mode, i.e. in case the PPP connection cannot be established within the time lapse determined by failtime, the SpeedTouch™ DHCP server will allocate an local private IP address to the DHCP client. By default the failtime is 4 seconds.	OPTIONAL
[errorlt]	A number between 0 and 1814400 (seconds). Represents the leasetime of the private address issued when a PPP link fails. In case the PPP link fails after failtime has elapsed, this parameter determines how long the private DHCP lease must be maintained before retrying to set up the PPP link again. By default the error lease time is 60 seconds.	OPTIONAL
[dodlt]	A number between 0 and 1814400 (seconds). Represents the leasetime of the temporary private IP address in case of a dial-on-demand PPP link. In case of a dial-on-demand PPP link, this parameter determines the interval at which the the temporary DHCP lease must be maintained before checking whether a public IP address negotiated by a triggered PPP link is available. By default the dial-on-demand lease time is 10 seconds.	OPTIONAL

EXAMPLE:

```
=>dhcp server spoof failtime=8 errorlt=120 dodlt=20
=>dhcp server spoof
Failure timeout (!DoD):    8 sec
Failure lease time (!DoD): 120 sec
Temp. lease time (DoD):    20 sec
=>
```

RELATED COMMANDS:

dhcp server policy	Set DHCP server policy.
dhcp server stats	Show current DHCP server state and statistics.

dhcp server start

Start SpeedTouch™ DHCP server.

SYNTAX:

```
dhcp server start
```

EXAMPLE:

```
=>dhcp server stats
DHCP Server State:    Stopped
DHCP server statistics:
Corrupted packet recv    :      0
.....
=>dhcp server start
=>dhcp server stats
DHCP server state:    Running
DHCP server statistics:
Corrupted packet recv    :      0
.....
=>
```

RELATED COMMANDS:

dhcp server stats	Show current DHCP server state and statistics.
dhcp server stop	Stop DHCP server.

dhcp server stats

Show SpeedTouch™ DHCP server statistics.

SYNTAX:

```
dhcp server stats
```

EXAMPLE OUTPUT:

```
=>dhcp server stats
DHCP Server State:   Stopped
DHCP server statistics:
Corrupted packet recv   :         0
DISCOVER                :         2451
REQUEST                 :         28
DECLINE                 :         0
RELEASE                 :         22
INFORM                  :         1
Pure BOOTP REQUESTS     :         2
Other message types     :         0
OFFERs sent             :         2451
ACKs sent               :         19
NAKs sent               :         0
Lease table got full    :         no
Ping table got full     :         no
Second dhcp server seen :         no
Total size of lease table: 32, in use: 16, free: 50 %
=>
```

DESCRIPTION:

DHCP server state	Indicates the state of the SpeedTouch™ DHCP server.
Corrupted packet recv	Indicates the number of corrupted packets (not complaint to RFC2131) received from the LAN.
DISCOVER	Indicates the number of DHCP server discovery packets received from the Local Area Network (LAN). These broadcasts are sent by potential DHCP clients to locate available DHCP servers.
REQUEST	Indicates the number of DHCP address lease requests received from the LAN.
DECLINE	Indicates the number of DHCP address lease requests declined.
RELEASE	Indicates the number of DHCP address release requests received from DHCP clients.
INFORM	Indicates the number of information requests received from DHCP clients.
Pure BOOTP requests	Indicates the number of BOOTP requests received from the LAN.
OFFERs sent	Indicates the number of IP address offers sent in reply to DHCP requests.
ACKs sent	Indicates the number of ACKnowledgement replies sent to successfully configured DHCP clients.
NAKs sent	Indicates the number of Not-AcKnowledgement replies sent to wrongly configured DHCP clients.
Lease table got full	Indicates whether the maximum number of DHCP leases is reached or not.
Ping table got full	Indicates whether the history list of IP address pings got full or not. These pings are sent by the SpeedTouch™ DHCP server to verify whether the IP address is already in use on the LAN or not. (dhcp server policy verifyfirst=yes)
Second DHCP server	Indicates whether a concurrent DHCP server was found on the LAN or not.

RELATED COMMANDS:

dhcp server clear	Clear DHCP server statistics.

dhcp server stop

Stop SpeedTouch™ DHCP server.

SYNTAX:

```
dhcp server stop
```

EXAMPLE:

```
=>dhcp server stats
DHCP Server State:   Running
DHCP server statistics:
Corrupted packet recv  :              0
.....
=>dhcp server start
=>dhcp server stats
DHCP server state:   Stopped
DHCP server statistics:
Corrupted packet recv  :              0
.....
=>
```

RELATED COMMANDS:

dhcp server start	Start DHCP server.
dhcp server stats	Show current DHCP server state and statistics.

dhcp server lease add

Assign a DHCP server lease to a DHCP host in the local network.

SYNTAX:

```
dhcp server lease add   clientid = <client-id>
                        pool = <string>
                        [addr = <ip-address>]
                        [offset = <number>]
                        [leasetime = <number>]
                        [hostname = <{hostname|""}>]
```

clientid	The DHCP client identification string of the booting host.	REQUIRED
pool	The name of the DHCP server pool from which the DHCP lease should be taken from. Use :dhcp server pool list for a list of available DHCP server pools.	REQUIRED
[addr]	The favoured IP address for this DHCP host. This IP address, if specified, must be in the range of the DHCP pool specified.	OPTIONAL
[offset]	A number between 0 and the integer number defined by the number of available IP addresses in the DHCP server pool. Represents the IP address offset in the DHCP server pool preserved for this host. Not specifying this parameter does not preserve an IP address for the host.	OPTIONAL
[leasetime]	A number between 0 and 1814400 (seconds). Represents the time the host is allowed to use this address, before renewing. Specifying -1 makes the lease permanent.	OPTIONAL
[hostname]	The hostname to add to the local Domain Name System (DNS) table for this host. Use “” in case no hostname is associated with this lease.	OPTIONAL

EXAMPLE:

```
=>dhcp server lease list
Lease          Pool          TTL          State      Clientid
0  0.0.0.0     dhcp_pool_1   00:26:40     FREE       00:90:D0:12:34:56
=>dhcp server lease add clientid=01:23:55:67:89:ab pool=Local_pool leasetime=3600
=>dhcp server lease list
Lease          Pool          TTL          State      Clientid
0  0.0.0.0     dhcp_pool_1   00:26:40     FREE       00:90:D0:12:34:56
1  10.0.0.1    local_pool    00:59:22     USED       01:23:45:67:89:AB
=>
```

RELATED COMMANDS:

dhcp server lease delete	Delete a DHCP lease.
dhcp server lease flush	Delete all DHCP leases.
dhcp server lease list	Show current DHCP leases.

dhcp server lease delete

Delete a DHCP lease.

SYNTAX:

```
dhcp server lease delete  clientid = <clientid>
                          [index = <number>]
```

clientid	The DHCP client identification string of the DHCP lease.	REQUIRED
[index]	The index number of the entry to be deleted. Use :dhcp server lease list to see a list of the index numbers of all current DHCP leases.	OPTIONAL

EXAMPLE:

```
=>dhcp server lease list
Lease          Pool          TTL          State      Clientid
0  0.0.0.0     dhcp_pool_1   00:26:40     FREE       00:90:D0:12:34:56
1  10.0.0.1    local_pool    00:59:22     USED       01:23:45:67:89:AB
=>dhcp server lease delete index=0
=>dhcp server lease list
Lease          Pool          TTL          State      Clientid
1  10.0.0.1    local_pool    00:59:22     USED       01:23:45:67:89:AB
=>
```

RELATED COMMANDS:

dhcp server lease add	Add a DHCP lease manually.
dhcp server lease flush	Delete all DHCP leases.
dhcp server lease list	Show current DHCP leases.

dhcp server lease flush

Flush complete DHCP server configuration and dynamic leases.

Note The flush command does not impact previously saved configurations.

SYNTAX:

```
dhcp server lease flush
```

EXAMPLE:

```
=>dhcp server lease list
Lease         Pool         TTL          State      Clientid
0  0.0.0.0    dhcp_pool_1  00:26:40     FREE       00:90:D0:12:34:56
1  10.0.0.1   local_pool   00:59:22     USED       01:23:45:67:89:AB
2  10.0.0.101 local_pool   00:21:01     USED       01:23:89:AB:80:CD
3  10.0.0.132 local_pool   00:45:37     USED       09:D0:25:CE:F1:31
5  10.0.0.5   local_pool   00:21:11     USED       AB:33:A1:7C:89:DD
4  10.0.0.6   local_pool   00:59:01     USED       E3:81:9F:11:11:11
8  10.0.0.8   local_pool   00:01:00     USED       08:80:09:90:AB:DC
9  10.0.0.15  local_pool   00:00:23     USED       08:93:DA:AE:01:AF
=>dhcp server lease flush
=>dhcp server lease list
=>
```

RELATED COMMANDS:

dhcp server lease add	Add a DHCP lease manually.
dhcp server lease delete	Delete a DHCP lease.
dhcp server lease list	Show current DHCP leases.

dhcp server lease list

List current DHCP leases, indicated by their index number.

SYNTAX:

```
dhcp server lease  list
```

EXAMPLE OUTPUT:

```
=>dhcp server lease list
=>dhcp server lease list
Lease          Pool          TTL          State      Clientid
0  0.0.0.0     dhcp_pool_1   00:26:40     FREE       00:90:D0:12:34:56
1  10.0.0.1    local_pool    00:59:22     USED       01:23:45:67:89:AB
2  10.0.0.101  local_pool    00:21:01     USED       01:23:89:AB:80:CD
3  10.0.0.132  local_pool    00:45:37     USED       09:D0:25:CE:F1:31
5  10.0.0.5    local_pool    00:21:11     USED       AB:33:A1:7C:89:DD
4  10.0.0.6    local_pool    00:59:01     USED       E3:81:9F:11:11:11
8  10.0.0.8    local_pool    00:01:00     USED       08:80:09:90:AB:DC
9  10.0.0.15   local_pool    00:00:23     USED       08:93:DA:AE:01:AF
=>
```

RELATED COMMANDS:

dhcp server lease add	Add a DHCP lease manually.
dhcp server lease delete	Delete a DHCP lease.
dhcp server lease flush	Delete complete DHCP server configuration and dynamic leases.

dhcp server pool add

Add a DHCP server pool.

SYNTAX:

```
dhcp server pool add   [name = <string>]
                       [index = <number>]
```

[name]	A name for the DHCP server pool. If not specified, the name is "dhcp_pool_x", where x is a subsequent number.	OPTIONAL
[index]	A number between 0 (highest priority) and the highest number (lowest priority) found in the list of existing DHCP server pools. Represents a (higher) priority for the DHCP server pool. If not specified, the DHCP pool is given the lowest subsequent priority.	OPTIONAL

EXAMPLE:

```
=>dhcp server pool list
Pool            Start           End             State           PPP
0 dhcp_pool_1   0.0.0.0         0.0.0.0         FREE
1 My_LAN_Pool   10.0.0.1        10.0.0.254      USED
2 dhcp_pool_2   0.0.0.0         0.0.0.0         FREE
=>dhcp server pool add
=>dhcp server pool list
Pool            Start           End             State           PPP
0 dhcp_pool_1   0.0.0.0         0.0.0.0         FREE
1 My_LAN_Pool   10.0.0.1        10.0.0.254      USED
2 dhcp_pool_2   0.0.0.0         0.0.0.0         FREE
3 dhcp_pool_3   0.0.0.0         0.0.0.0         FREE
=>dhcp server pool add name=POOL_EXTRA1
=>dhcp server pool list
Pool            Start           End             State           PPP
0 dhcp_pool_1   0.0.0.0         0.0.0.0         FREE
1 My_LAN_Pool   10.0.0.1        10.0.0.254      USED
2 dhcp_pool_2   0.0.0.0         0.0.0.0         FREE
3 dhcp_pool_3   0.0.0.0         0.0.0.0         FREE
4 POOL_EXTRA1   0.0.0.0         0.0.0.0         FREE
=>ppp ifconfig name=PPP_Test pool=POOL_EXTRA1
=>dhcp server pool list
Pool            Start           End             State           PPP
0 dhcp_pool_1   0.0.0.0         0.0.0.0         FREE
1 My_LAN_Pool   10.0.0.1        10.0.0.254      USED
2 dhcp_pool_2   0.0.0.0         0.0.0.0         FREE
3 dhcp_pool_3   0.0.0.0         0.0.0.0         FREE
4 POOL_EXTRA1   0.0.0.0         0.0.0.0         FREE            PPP_Test
=>
```

RELATED COMMANDS:

dhcp server pool delete	Delete a DHCP pool.
dhcp server pool flush	Delete all DHCP pools.
dhcp server pool list	Show current DHCP pools.

dhcp server pool config

Configure an existing DHCP pool. Before you are able to configure the DHCP pool, you must create it via :dhcp server pool add.

SYNTAX:

```
dhcp server pool config   name = <string>
                          [index = <number>]
                          [poolstart = <ip-address>]
                          [poolend = <ip-address>]
                          [netmask = <ip-mask{dotted or cidr}>]
                          [gateway = <ip-address|0>]
                          [primdns = <ip-address|0>]
                          [secdns = <ip-address|0>]
                          [leasetime = <number>]
```

name	The name of the DHCP server pool to configure.	REQUIRED
[index]	A number between 0 (highest priority) and the highest number (lowest priority) found in the list of existing DHCP server pools. Represents a (higher) priority for the DHCP server pool.	OPTIONAL
[poolstart]	The lowest IP address in the DHCP address range to use for leasing. Default value of this parameter is 0.0.0.0 (not specified), which means that the lowest IP address of the pool will be defined by the remote server via Internet Protocol Control Protocol (IPCP) as soon as the Point-to-Point Protocol (PPP) IPCP subnetmasking connection is established.	OPTIONAL
[poolend]	The highest IP address in the DHCP address range to use for leasing. Default value of this parameter is 0.0.0.0 (not specified), which means that the highest IP address of the pool will be defined by the remote server via IPCP as soon as the PPP IPCP subnetmasking connection is established.	OPTIONAL
[netmask]	The applicable netmask for the DHCP leases.	OPTIONAL
[gateway]	The IP address of the default gateway for the DHCP clients. Default value of this parameter is 0 (not specified), which means that the gateway IP address will be communicated by the remote server as soon as the PPP IPCP subnetmasking connection is established or that the SpeedTouch™ acts as the LAN default gateway.	OPTIONAL
[primdns]	The IP address of the primary DNS server for the DHCP clients. Default value of this parameter is 0 (not specified), which means that the IP address of the DNS server will be communicated by the remote server as soon as the PPP IPCP subnetmasking connection is established or that the SpeedTouch™ acts as the LAN DNS server.	OPTIONAL

[secdns]	The IP address of the optional secondary DNS server for DHCP clients. Default value of this parameter is 0 (not specified), which means that the gateway IP address will be communicated by the remote server as soon as the PPP IPCP subnetmasking connection is established.	OPTIONAL
[leasetime]	A number between 0 and 1814400 (seconds). Represents the time for which a client can use its dynamically allocated IP address. By default the leasetime is 2 hours (7200 seconds). Specifying -1 makes the lease permanent.	OPTIONAL

EXAMPLE:

```
=>dhcp server pool list
Pool            Start           End             State           PPP
0 LAN_Private   10.0.0.1        10.0.0.254      USED
=>dhcp server pool config name=My_Pool poolstart=192.6.11.101 |
poolend=192.6.11.254 netmask=255.255.255 gateway=192.6.11.100 leasetime=21600
=>dhcp server pool list
Pool            Start           End             State           PPP
0 LAN_Private   10.0.0.1        10.0.0.254      USED
1 My_Pool       192.6.11.101    192.6.11.254    USED
=>
```

RELATED COMMANDS:

dhcp server pool list	Show current DHCP pools.

dhcp server pool delete

Delete a DHCP pool.

SYNTAX:

```
dhcp server pool delete       name = <string>
```

name	The name of the DHCP server pool to delete. Use :dhcp server pool list to see a list of all current DHCP leases.	REQUIRED

EXAMPLE:

```
=>dhcp server pool list
Pool            Start           End             State           PPP
0 LAN_Private   10.0.0.1        10.0.0.254      USED
1 My_Pool       192.6.11.101    192.6.11.254    USED
=>dhcp server pool delete name=My_Pool
=>dhcp server pool list
Pool            Start           End             State           PPP
0 LAN_Private   10.0.0.1        10.0.0.254      USED
=>
```

RELATED COMMANDS:

dhcp server pool add	Add a DHCP pool.
dhcp server pool flush	Delete all DHCP pools.
dhcp server pool list	Show current DHCP pools.

dhcp server pool flush

Flush all DHCP pools

Note The flush command does not impact previously saved configurations.

SYNTAX:

```
dhcp server pool flush
```

EXAMPLE:

```
=>dhcp server pool list
Pool            Start           End             State           PPP
0 dhcp_pool_1   0.0.0.0         0.0.0.0         FREE
1 My_LAN_Pool   10.0.0.1        10.0.0.254      USED
2 POOL_EXTRA2   0.0.0.0         0.0.0.0         FREE
3 dhcp_pool_2   0.0.0.0         0.0.0.0         FREE
4 dhcp_pool_3   0.0.0.0         0.0.0.0         FREE
5 POOL_EXTRA1   0.0.0.0         0.0.0.0         FREE
=>dhcp server pool flush
=>dhcp server pool list
=>
```

RELATED COMMANDS:

dhcp server pool add	Add a DHCP pool.
dhcp server pool delete	Delete a DHCP pool.
dhcp server pool list	Show current DHCP pools.

dhcp server pool list

List current DHCP leases.

SYNTAX:

```
dhcp server pool list
```

EXAMPLE:

```
=>dhcp server pool list
Pool            Start           End             State           PPP
0 dhcp_pool_1   0.0.0.0         0.0.0.0         FREE
1 My_LAN_Pool   10.0.0.1        10.0.0.254      USED
2 POOL_EXTRA2   0.0.0.0         0.0.0.0         FREE
3 dhcp_pool_2   0.0.0.0         0.0.0.0         FREE
4 dhcp_pool_3   0.0.0.0         0.0.0.0         FREE
5 POOL_EXTRA1   0.0.0.0         0.0.0.0         FREE
=>
```

RELATED COMMANDS:

dhcp server pool add	Add a DHCP pool.
dhcp server pool delete	Delete a DHCP pool.
dhcp server pool flush	Delete all DHCP pools.

DNS Commands

Contents

This chapter covers the following commands:

dns add

Add a static DNS entry for IP hosts who do not reveal their hostname in the DHCP request, or do not support DHCP.

SYNTAX:

dns add	hostname = <string> [addr = <ip-address>]

hostname	The name of the IP host (without the (sub)domain name).	REQUIRED
[addr]	The IP address of the host (without mask).	OPTIONAL
	Note If this parameter is not specified, the hostname applies to the SpeedTouch™ itself.	

EXAMPLE:

```
=>dns list
Domain: business.lan
Nr.       Hostname             IP Address
0         SpeedTouch™           *.*.*.*
1         TestHost             10.0.0.140
2         HTTP_Server          10.0.0.8
Total Table Size: 73 entries
Amount used: 3 (4%)
=>dns add hostname=FTP_Server addr=10.0.0.7
=>dns list
Domain: business.lan
Nr.       Hostname             IP Address
0         SpeedTouch™           *.*.*.*
1         TestHost             10.0.0.140
2         HTTP_Server          10.0.0.8
3         FTP_Server           10.0.0.7
Total Table Size: 73 entries
Amount used: 4 (5%)
=>
```

RELATED COMMANDS:

dns list	List current DNS entries.
dns delete	Delete a DNS entry.

dns clear

Delete current DNS entries.

SYNTAX:

```
dns clear
```

EXAMPLE:

```
=>dns list
Domain: business.lan
Nr.             Hostname            IP Address
0               SpeedTouch™          *.*.*.*
1               TestHost            10.0.0.140
2               HTTP_Server         10.0.0.8
3               FTP_Server          10.0.0.7
Total Table Size: 73 entries
Amount used: 4 (5%)
=>dns clear
=>dns list
Domain: business.lan
Nr.             Hostname            IP Address
Total Table Size: 73 entries
Amount used: 0 (0%)
=>
```

RELATED COMMANDS:

dns list	List current DNS entries.

dns clrstats

Clear DNS statistics.

SYNTAX:

```
dns clrstats
```

EXAMPLE:

```
=>dns stats
DNS Statistics:
Corrupted packets recv              :     0
Local questions resolved            :     0
Local neg answers sent              :     4
Total DNS packets fwd               :     0
External answers recv               :     0
Fwd table full, discard             :     0
Spurious answers                    :     0
Unknown query types                 :     0

Total number of packets received    :     4

=>dns clrstats
DNS statistics cleared.
=>dns stats
DNS Statistics:
Corrupted packets recv              :     0
Local questions resolved            :     0
Local neg answers sent              :     0
Total DNS packets fwd               :     0
External answers recv               :     0
Fwd table full, discard             :     0
Spurious answers                    :     0
Unknown query types                 :     0

Total number of packets received    :     0

=>
```

RELATED COMMANDS:

dns stats	Show DNS server/forwarder statistics.

dns delete

Delete a DNS entry.

SYNTAX:

dns delete	index = <number>

index	The index number of the entry to be deleted. Use :dns list to see a list of the index numbers of all current DNS entries.	REQUIRED

EXAMPLE:

```
=>dns list
Domain: business.lan
Nr.             Hostname            IP Address
0               SpeedTouch™          *.*.*.*
1               TestHost            10.0.0.140
2               HTTP_Server         10.0.0.8
3               FTP_Server          10.0.0.7
Total Table Size: 73 entries
Amount used: 4 (5%)
=>dns delete index=2
=>dns list
Domain: business.lan
Nr.             Hostname            IP Address
0               SpeedTouch™          *.*.*.*
1               TestHost            10.0.0.140
3               FTP_Server          10.0.0.7
Total Table Size: 73 entries
Amount used: 3 (4%)
=>
```

RELATED COMMANDS:

dns add	Add a static DNS entry.
dns list	List current DNS entries.

dns domain

Set local DNS (sub)domain name.

SYNTAX:

dns domain	domain = <string>

domain	The local DNS (sub)domain name.	REQUIRED

EXAMPLE:

```
=>dns list
Domain: business.lan
Nr.             Hostname             IP Address
0               SpeedTouch™           *.*.*.*
1               TestHost             10.0.0.140
2               HTTP_Server          10.0.0.8
3               FTP_Server           10.0.0.7
Total Table Size: 73 entries
Amount used: 4 (5%)
=>dns domain domain=office.home.lan
=>dns list
Domain: office.home.lan
Nr.             Hostname             IP Address
0               SpeedTouch™           *.*.*.*
1               TestHost             10.0.0.140
2               HTTP_Server          10.0.0.8
3               FTP_Server           10.0.0.7
Total Table Size: 73 entries
Amount used: 4 (5%)
```

RELATED COMMANDS:

dns list	List current DNS entries.

dns flush

Flush the complete SpeedTouch™ DNS server/forwarder configuration and static entries.

Note The flush command does not impact previously saved configurations.

SYNTAX:

```
dns flush
```

EXAMPLE:

```
=>dns list
Domain: office.home.lan
Nr.             Hostname            IP Address
4*              Z7V1D8              10.0.0.29
0               SpeedTouch™          *.*.*.*
1               TestHost            10.0.0.140
2               Default             10.0.0.8
3               ftpserver           172.16.0.1
Total Table Size: 73 entries
Amount used: 5 (6%)
=>dns flush
=>dns list
Domain: lan
Nr.             Hostname            IP Address
3*              Z7V1D8              10.0.0.29
Total Table Size: 73 entries
Amount used: 1 (1%)
=>
```

dns fwdadd

Add a DNS forwarding entry. The entries in the forwarding list determine which DNS server should be used for which PC. If an identification cannot be established within the local LAN, the request is forwarded to another DNS server, on another network (Internet/LAN to LAN connection). The connection is negotiated within a PPP link.

SYNTAX:

```
dns fwdadd        dns = <ip-address>
                  src = <ip-address>
                  mask = <ip-mask (dotted or cidr)>
                  [direct = <number>]
```

dns	The IP address of the (remote) DNS server.	REQUIRED
src	The source IP address (pool) of the host(s) using this DNS server.	REQUIRED
mask	The appropriate source IP (sub)netmask.	REQUIRED
[direct]	Determines whether DNS replies are sent directly back to the client (1) or relayed by the SpeedTouch™ DHCP server's DNS forwarder (0) in case of PPP-to-DHCP spoofing connections.	OPTIONAL

EXAMPLE:

```
=>dns fwdlist
DNS forwarding servers:
DNS             SRC             MASK            Direct
10.0.0.138      10.0.0.2        255.255.255.0   yes
=>dns fwdadd dns=10.0.0.138 src-10.0.0.3 mask=24 direct=1
Dns forwarding server added.
=>dns fwdlist
DNS forwarding servers:
DNS             SRC             MASK            Direct
10.0.0.138      10.0.0.2        255.255.255.0   yes
10.0.0.138      10.0.0.3        255.255.255.0   yes
=>
```

RELATED COMMANDS:

dns fwddelete	Delete a DNS forwarding entry.
dns fwdlist	Show current DNS forwarding entries.

dns fwddelete

Delete a DNS forwarding entry.

SYNTAX:

```
dns fwddelete     src = <ip-address>
                  mask = <ip-mask (dotted or cidr)>
                  [dns = <ip-address>]
```

src	The source IP address (pool) of the hosts to remove the entry for.	REQUIRED
mask	The source IP (sub)netmask.	REQUIRED
[dns]	The IP address of the (remote) DNS server (in case of multiple DNS server entries).	OPTIONAL

EXAMPLE:

```
=>dns fwdlist
DNS forwarding servers:
DNS             SRC             MASK            Direct
10.0.0.138      10.0.0.0        255.255.255.0   yes
192.6.11.150    192.6.11.0      255.255.255.0   yes
=>dns fwddelete src-192.6.11.0 mask=24 dns=192.6.11.150
Dns forwarding server deleted.
=>dns fwdlist
DNS forwarding servers:
DNS             SRC             MASK            Direct
10.0.0.138      10.0.0.0        255.255.255.0   yes
=>
```

RELATED COMMANDS:

dns fwdadd	Add a DNS forwarding entry.
dns fwdlist	Show current DNS forwarding entries.

dns fwdlist

Show current DNS forwarding entries.

SYNTAX:

```
dns fwdlist
```

EXAMPLE OUTPUT:

```
=>dns fwdlist
DNS forwarding servers:
DNS             SRC             MASK            Direct
10.0.0.138      10.0.0.0        255.255.255.0   yes
192.6.11.150    192.6.11.0      255.255.255.0   yes
=>
```

RELATED COMMANDS:

dns fwdadd	Add a DNS forwarding entry.
dns fwddelete	Delete a DNS forwarding entry.
dns fwdtable	Show DNS forwarding table.

dns fwdtable

Show DNS forwarding table, i.e. list all curently unresolved DNS requests.

SYNTAX:

```
dns fwdtable
```

EXAMPLE OUTPUT:

```
=>dns fwdtable
Forwarding table:
Nr.  Ip Address    (port#):id(hex)  (expiry)     dns server     tries
0    10.10.10.12   (54751):8331     (13 sec)     10.10.10.112   1
Timeout: 15 seconds
Table size: 10
amount of table used: 1 (10%)
=>
```

RELATED COMMANDS:

dns fwdlist	Show current DNS forwarding entries.

dns list

Show current DNS entries.

SYNTAX:

```
dns list
```

EXAMPLE OUTPUT:

```
=>dns list
Domain: office.home.lan
Nr.             Hostname            IP Address
4*              Z7V1D8              10.0.0.29
0               SpeedTouch™          *.*.*.*
1               TestHost            10.0.0.140
2               Default             10.0.0.8
3               ftpserver           172.16.0.1
Total Table Size: 73 entries
Amount used: 5 (6%)
=>
```

EXAMPLE INPUT/OUTPUT IN A NETWORKED ENVIRONMENT:

The SpeedTouch™ is configured as DNS server.

```
=>dns list
Domain: SpeedLAN.local
Nr.             Hostname            IP Address
0               SpeedTouch™          *.*.*.*
1               Server              10.10.1.1
2               Client              10.0.0.3
Total Table Size: 73 entries
Amount used: 3 (4%)
=>
```

RELATED COMMANDS:

dns add	Add a static DNS entry.
dns delete	Delete a DNS entry (via its index number).

dns nslookup

Search the hostname (via a known IP address) or the IP address (via a known hostname) of a DNS host.

SYNTAX:

`dns nslookup`	`lookup = <string>`

lookup	The DNS hostname or IP address to query.	REQUIRED

EXAMPLE:

```
=>dns list
Domain: office.home.lan
Nr.             Hostname             IP Address
4*              Z7V1D8               10.0.0.29
0               SpeedTouch™           *.*.*.*
1               TestHost             10.0.0.140
2               Default              10.0.0.8
3               ftpserver            172.16.0.1
Total Table Size: 73 entries
Amount used: 5 (6%)
=>dns nslookup lookup=TestHost
Name:    TestHost
Address: 10.0.0.140
=>dns nslookup lookup=10.0.0.29
Name:    Z7V1D8
Address: 10.0.0.29
=>
```

RELATED COMMANDS:

dns list	List current DNS entries.

dns start

Start SpeedTouch™ DNS server/forwarder.

SYNTAX:

```
dns start
```

EXAMPLE:

```
=>dns status
DNS server status: Stopped
DNS table size                          :  73,  in use:  4,  free:  94 %
DNS forwarding table size               :  10,  in use:  0,  free:  100 %
DNS forwarding dns servers table size   :  25,  in use:  4,  free:  84 %
No dns cache.
Tracing: off
=>dns start
DNS server started.
=>dns status
DNS server status: Started
DNS table size                          :  73,  in use:  4,  free:  94 %
DNS forwarding table size               :  10,  in use:  0,  free:  100 %
DNS forwarding dns servers table size   :  25,  in use:  4,  free:  84 %
No dns cache.
Tracing: off
=>
```

RELATED COMMANDS:

dns status	Show DNS server/forwarder configuration.
dns stop	Stop DNS server/forwarder.

dns stats

Show SpeedTouch™ DNS server/forwarder statistics.

SYNTAX:

```
dns stats
```

EXAMPLE INPUT/OUTPUT IN A NETWORKED ENVIRONMENT:

The SpeedTouch™ is configured as DNS server.

```
=>dns list
Domain: SpeedLAN.local
Nr.             Hostname             IP Address
0               SpeedTouch™           *.*.*.*
1               Server               10.10.1.1
2               Client               10.0.0.3
Total Table Size: 73 entries
Amount used: 3 (4%)
=>dns stats
DNS Statistics:
Corrupted packets recv             :     0
Local questions resolved           :     1
Local neg answers sent             :     0
Total DNS packets fwd              :     0
External answers recv              :     0
Fwd table full, discard            :     0
Spurious answers                   :     0
Unknown query types                :     0
Total number of packets received   :     1
=>(Ping Client.SpeedLAN.local)
=>(CTRL + Q)
dnsd: Internet class type A request received from 10.10.1.1.
dnsd: Client.SpeedLAN.local found in local database.
dnsd: Client.SpeedLAN.local resolved into 10.0.0.3.
=>(Ping Server.SpeedLAN.local)
dnsd: Internet class type A request received from 10.10.1.1.
dnsd: Server.SpeedLAN.local found in local database.
dnsd: Server.SpeedLAN.local resolved into 10.0.0.3.
=>(CTRL + S)
=>dns stats
DNS Statistics:
Corrupted packets recv             :     0
Local questions resolved           :     3
Local neg answers sent             :     0
Total DNS packets fwd              :     0
External answers recv              :     0
Fwd table full, discard            :     0
Spurious answers                   :     0
Unknown query types                :     0
Total number of packets received   :     3
=>
```

RELATED COMMANDS:

dns clrstats	Clear DNS server/forwarder statistics.

dns status

Show SpeedTouch™ DNS server/forwarder configuration.

SYNTAX:

```
dns status
```

EXAMPLE OUTPUT:

```
=>dns status
DNS server status: Stopped
DNS table size                          :  73,  in use:  4,  free:  94 %
DNS forwarding table size               :  10,  in use:  0,  free:  100 %
DNS forwarding dns servers table size   :  25,  in use:  4,  free:  84 %
No dns cache.
Tracing: off
=>
```

dns stop

Stop SpeedTouch™ DNS server/forwarder.

SYNTAX:

```
dns stop
```

EXAMPLE:

```
=>dns status
DNS server status: Started
DNS table size                          :  73,  in use:  4,  free:  94 %
DNS forwarding table size               :  10,  in use:  0,  free:  100 %
DNS forwarding dns servers table size   :  25,  in use:  4,  free:  84 %
No dns cache.
Tracing: off
=>dns stop
DNS server stopped.
=>dns status
DNS server status: Stopped
DNS table size                          :  73,  in use:  4,  free:  94 %
DNS forwarding table size               :  10,  in use:  0,  free:  100 %
DNS forwarding dns servers table size   :  25,  in use:  4,  free:  84 %
No dns cache.
Tracing: off
=>
```

RELATED COMMANDS:

dns status	Show DNS server/forwarder configuration.
dns start	Start DNS server/forwarder.

dns toutfwd

Set DNS forwarding timeout.

SYNTAX:

dns toutfwd	timeout = <number>

timeout	A number (seconds). Represents the query forwarding timeout. This parameter determines how long the SpeedTouch™ DNS server should try to contact a (remote) DNS server before (temporarily) declaring the DNS requests unresolved. By default the timeout is 15 seconds.	REQUIRED

EXAMPLE:

```
=>dns fwdtable
Forwarding table:
Nr.  Ip Address    (port#):id(hex)  (expiry)    dns server    tries
0    10.10.10.12   (54751):8331     (13 sec)    10.10.10.112  1
Timeout: 15 seconds
Table size: 10
amount of table used: 1 (10%)
=>dns toutfwd timeout=20
Current timeout: 15 seconds
Timeout set to: 20 seconds
=>dns fwdtable
Forwarding table:
Nr.  Ip Address    (port#):id(hex)  (expiry)    dns server    tries
0    10.10.10.12   (54751):8331     (13 sec)    10.10.10.112  1
Timeout: 20 seconds
Table size: 10
amount of table used: 1 (10%)
=>
```

RELATED COMMANDS:

dns fwdtable	Show DNS forwarding table.
dns fwdlist	Show current DNS forwarding entries..
dns fwdadd	Add a DNS forwarding entry.
dns fwddelete	Delete a DNS forwarding entry.

dns troff

Disable verbose console messaging. No debug traces are generated.

SYNTAX:

```
dns troff
```

EXAMPLE:

```
=>dns status
DNS server status: Started
DNS table size                          :  73,  in use:  4,  free:  94 %
DNS forwarding table size               :  10,  in use:  0,  free:  100 %
DNS forwarding dns servers table size   :  25,  in use:  4,  free:  84 %
No dns cache.
Tracing: on
=>dns troff
=>dns status
DNS server status: Started
DNS table size                          :  73,  in use:  4,  free:  94 %
DNS forwarding table size               :  10,  in use:  0,  free:  100 %
DNS forwarding dns servers table size   :  25,  in use:  4,  free:  84 %
No dns cache.
Tracing: off
=>
```

RELATED COMMANDS:

dns fwdtable	Show DNS forwarding table.
dns fwdlist	Show current DNS forwarding entries..
dns status	Show DNS server/forwarder configuration.
dns tron	Enable verbose console messaging.

dns tron

Enable verbose console messaging. Debug traces are generated.

SYNTAX:

```
dns tron
```

EXAMPLE:

```
=>dns status
DNS server status: Started
DNS table size                          :  73,  in use:  4,  free:  94 %
DNS forwarding table size               :  10,  in use:  0,  free:  100 %
DNS forwarding dns servers table size   :  25,  in use:  4,  free:  84 %
No dns cache.
Tracing: off
=>dns tron
Tracing on.
=>dns status
DNS server status: Started
DNS table size                          :  73,  in use:  4,  free:  94 %
DNS forwarding table size               :  10,  in use:  0,  free:  100 %
DNS forwarding dns servers table size   :  25,  in use:  4,  free:  84 %
No dns cache.
Tracing: on
=>(CTRL + Q)
dnsd: Internet class type A request received from 10.0.0.10.
dnsd: aa.aa.be is outside our domain: forward.
dnsd: forwarding request from 10.0.0.10 (1318,0x0001) to 138.203.68.61
      (try=1): 'reply to ant' mode.
dnsd: Internet class type A request received from 10.0.0.10.
dnsd: aa.aa.be is outside our domain: forward.
dnsd: forwarding request from 10.0.0.10 (1318,0x0001) to 138.203.68.11
      (try=2): 'reply to ant' mode.
dnsd: forward answer from 138.203.68.11 to 10.0.0.10 (1318,0001).
dnsd: Internet class type A request received from 10.0.0.10.
dnsd: aa.aa.be.lan unknown: return error.
......
=>(CTRL + S)
```

RELATED COMMANDS:

dns fwdtable	Show DNS forwarding table.
dns fwdlist	Show current DNS forwarding entries..
dns status	Show DNS server/forwarder configuration.
dns troff	Disable verbose console messaging.

Env Commands

Contents

This chapter covers the following commands

env def

Define an environment variable.

Note This command is for internal use by the Setup wizard application only. Do not use it.

SYNTAX:

```
env def
```

env flush

Flush all non-system environment variables.

SYNTAX:

```
env flush
```

EXAMPLE:

```
=>env list
_COMPANY_NAME=THOMSON multimedia
_COMPANY_URL=http://www.thomson-multimedia.com
_PROD_NAME=SpeedTouch
_PROD_URL=http://www.speedtouch.com
_PROD_FRIENDLY_NAME=SpeedTouch 510
_PROD_DESCRIPTION=DSL Internet Gateway
_PROD_NUMBER=510
_BOARD_SERIAL_NBR=0238J1035
_BUILD=4.0.0.9.0
_BUILDNAME=Sascha4.0090
_PRL=3EC36939AAAA
_FIA=ND
_BOARD_NAME=ADNT-Q
_COMPANY_ID=ALCL
_COPYRIGHT=Copyright (c) 1999-2002, THOMSON multimedia
_TPVERSION=1.1.0
_MACADDR=00-90-D0-03-02-01
_UDN=uuid:UPnP-SpeedTouch510-1_00-90-D0-03-02-01
CONF_REGION=World
CONF_PROVIDER=Generic ISP
CONF_DESCRIPTION=Relayed PPPoA Packet Service configuration
CONF_SERVICE=Relayed PPPoA
CONF_VERSION=4.1.1.0
HOST_SETUP=auto
UPGRADE_URL=http://www.speedtouch.com/upgrade500.htm
COLUMNS=80
ROWS=24
SESSIONTIMEOUT=0
DSL_ADDR=8*35
CONF_DATE=January 2003
=>env flush
=>env list
_COMPANY_NAME=THOMSON multimedia
_COMPANY_URL=http://www.thomson-multimedia.com
_PROD_NAME=SpeedTouch
_PROD_URL=http://www.speedtouch.com
_PROD_FRIENDLY_NAME=SpeedTouch 510
_PROD_DESCRIPTION=DSL Internet Gateway
_PROD_NUMBER=510
_BOARD_SERIAL_NBR=0238J1035
_BUILD=4.0.0.9.0
_BUILDNAME=LLT6AA4.0090
_PRL=3EC36939AAAA
_FIA=ND
_BOARD_NAME=ADNT-Q
_COMPANY_ID=ALCL
_COPYRIGHT=Copyright (c) 1999-2002, THOMSON multimedia
_TPVERSION=1.1.0
_MACADDR=00-90-D0-03-02-01
_UDN=uuid:UPnP-SpeedTouch510-1_00-90-D0-03-02-01
=>
```

RELATED COMMANDS:

env list	List all environment variables.

env get

Get the current value of a environment variable.

SYNTAX:

env get	var = <string>

var	The name of the environment variable. Execute env list to see a list of all environment variables.	REQUIRED

EXAMPLE:

```
=>env list
_COMPANY_NAME=THOMSON multimedia
_COMPANY_URL=http://www.speedtouch.com
_PROD_NAME=SpeedTouch
_PROD_FRIENDLY_NAME=SpeedTouch 610 Business DSL Router
_PROD_NUMBER=610
_BOARD_SERIAL_NBR=012345678
_BUILD=R4.1.0.9
_BUILDNAME=Sascha4.109
_PRL=3EC99999XXXX
_BOARD_NAME=ADNT-R
_COMPANY_ID=NWBGL
_COPYRIGHT=Copyright(c) 2002 THOMSON multimedia
_MACADDR=00-90-D0-01-02-03
_UDN=uuid:UPnP-SpeedTouch™ 520-1_00-90-D0-01-02-03
CONF_REGION=Belgium
CONF_PROVIDER=peckelbs
CONF_DESCRIPTION=Default Bridging configuration
CONF_SERVICE=Bridging on 0/35 and 8/35
CONF_DATE=May 2002
COLUMNS=80
ROWS=24
SESSIONTIMEOUT=120
HOST_SETUP=user
VPI_VCI=8*35
=>env get var=VPI_VCI
8*35
=>
```

RELATED COMMANDS:

env set	Create and set a non-system environment variable.
env list	List all current environment variables.

env list

Show all currently available environment variables.

SYNTAX:

```
env list
```

EXAMPLE:

```
=>env list
_COMPANY_NAME=THOMSON multimedia
_COMPANY_URL=http://www.speedtouch.com
_PROD_NAME=SpeedTouch
_PROD_FRIENDLY_NAME=SpeedTouch 610 Business DSL Router
_PROD_NUMBER=610
_BOARD_SERIAL_NBR=012345678
_BUILD=R4.1.0.9
_BUILDNAME=Sascha4.109
_PRL=3EC99999XXXX
_BOARD_NAME=ADNT-R
_COMPANY_ID=NWBGL
_COPYRIGHT=Copyright(c) 2002 THOMSON multimedia
_MACADDR=00-90-D0-01-02-03
_UDN=uuid:UPnP-SpeedTouch™ 520-1_00-90-D0-01-02-03
CONF_REGION=Belgium
CONF_PROVIDER=peckelbs
CONF_DESCRIPTION=Default Bridging configuration
CONF_SERVICE=Bridging on 0/35 and 8/35
CONF_DATE=May 2002
COLUMNS=80
ROWS=24
SESSIONTIMEOUT=120
HOST_SETUP=user
VPI_VCI=8*35
=>
```

env set

Create and set a non-system environment variable or change the value of a non-system environment variable.

SYNTAX:

```
env set                 var = <string>
                        value = <string>
```

var	The name of the environment variable. When creating an environment variable, any name is allowed, however spaces are not allowed and the name may not start with “CONF”, “HOST”, an underscore “_” or the dollar sign “$”.	REQUIRED
value	A quoted string which defines the value of the environment variable. The value of system variables (built-in variables with names starting with an underscore “_”, “CONF” or “HOST”) can not be changed.	REQUIRED

EXAMPLE:

```
=>env list
_COMPANY_NAME=THOMSON multimedia
_COMPANY_URL=http://www.speedtouch.com
_PROD_NAME=SpeedTouch
_PROD_FRIENDLY_NAME=SpeedTouch 610 Business DSL Router
.....
CONF_DESCRIPTION=Default Bridging configuration
CONF_SERVICE=Bridging on 0/35 and 8/35
CONF_DATE=May 2002
HOST_SETUP=user
=>env set var=VPI_VCI value="8*35"
=>env get var=VPI_VCI
8*35
=>env set var=VPI_VCI value="11*35"
=>env get var=VPI_VCI
11*35
=>
```

RELATED COMMANDS:

env get	Show the value of an environment variable.
env list	List all current environment variables.
env unset	Delete a non-system environment variable.

env unset

Delete a non-system environment variable.

SYNTAX:

`env unset`	`var = <string>`

var	The name of the environment variable to delete. System variables (built-in variables with names starting with an underscore "_", "CONF" or "HOST") can not be unset, nor changed or deleted.	REQUIRED

EXAMPLE:

```
=>env list
_COMPANY_NAME=THOMSON multimedia
_COMPANY_URL=http://www.speedtouch.com
_PROD_NAME=SpeedTouch
.....
CONF_DATE=May 2002
HOST_SETUP=user
VPI_VCI=8*35
=>env unset var=VPI_VCI
=>env list
_COMPANY_NAME=THOMSON multimedia
_COMPANY_URL=http://www.speedtouch.com
_PROD_NAME=SpeedTouch
.....
CONF_DATE=May 2002
HOST_SETUP=user
=>
```

RELATED COMMANDS:

env set	Create and set a non-system environment variable.
env list	List all current environment variables.

Eth Commands

Introduction

The eth command group is only applicable to SpeedTouch™500Series variants equipped with a single Ethernet port (i.e. no switch variants).

Contents

This chapter covers the following topics:

eth ifconfig

Configure the Ethernet port.

SYNTAX:

```
eth ifconfig        intf = <number {0-0}>
                    type = <{auto|10BaseTHD|10BaseTFD|100BaseTHD|100BaseTFD}
                    >
```

intf	The Ethernet interface to configure. Currently only one (the single Ethernet port) is available: intf=0.	REQUIRED
type	The Ethernet interface's Ethernet type. Select either: • auto • Auto negotiation of Ethernet communication speed (10Mb/s or 100Mb/s) and Duplex mode (half duplex or full duplex). • 10BaseTHD • 10Mb/s communication speed in half duplex mode. • 10BaseTFD • 10Mb/s communication speed in full duplex mode. • 100BaseTHD • 100Mb/s communication speed in half duplex mode. • 100BaseTFD • 100Mb/s communication speed in full duplex mode. By default the Ethenet type is set to auto and should never be changed, except in case of communication problems.	REQUIRED

RELATED COMMANDS:

eth iflist	Show Ethernet port configuration and current operating mode.

eth iflist

Show the Ethernet port configuration and current operating status.

SYNTAX:

```
eth iflist
```

EXAMPLE:

```
=>eth iflist
Intf   Type            Result Type
0      auto            100BaseTFD
=>
```

DESCRIPTION:

Type	Indicates the configured Ethernet communication speed and duplex mode.
Result type	Indicates the effective operating status in case Type=auto. In other cases, when the Ethernet types do NOT match, Result type=unknown and no Ethernet connectivity will exist.

RELATED COMMANDS:

eth ifconfig	Configure the Ethernet port.

Firewall Commands

Contents

This chapter covers the following topics:

firewall assign

Assign a chain to an entry point. An entry point, also referred to as hook or a Packet Interception Point (PIP) is the location where packets are intercepted to be compared against a chain of rules

SYNTAX:

```
firewall assign      hook = <{input|sink|forward|source|output}>
                     chain = <string>
```

hook	The entry point's name to assign a chain to. Choose between: • input : The point off all incoming traffic. At this point it can be determined whether the packet is allowed to reach the SpeedTouch™ IP router or local host. • sink : The point off all traffic destined to the SpeedTouch™ IP router itself. At this point it can be determined whether the packet is allowed to address the local host. • forward : The point off all traffic to be forwarded by the SpeedTouch™ IP router. At this point it can be determined whether the packet is allowed to be handled, i.e. routed. • source : The point off all traffic sourced by the SpeedTouch™ IP router. At this point it can be determined whether the packet is allowed to leave the local host. • output : The point off all outgoing traffic. At this point it can be determined whether the packet is allowed to leave the SpeedTouch™ IP router or local host.	REQUIRED
chain	The name of the chain to use.	REQUIRED

EXAMPLE:

```
=>firewall list
assign     hook=sink        chain=sink
assign     hook=forward     chain=forward
assign     hook=source      chain=source
=>firewall chain create chain Telnet
=>firewall assign hook=sink chain=Telnet
=>firewall list
assign     hook=sink        chain=Telnet
assign     hook=forward     chain=forward
assign     hook=source      chain=source
=>
```

RELATED COMMANDS:

firewall chain create	Create a chain.
firewall chain list	Show a list of all current chains.

firewall flush

Flush all associations between a hook and its chain(s). The chain itself is not removed.

Note The flush command does not impact previously saved configurations.

SYNTAX:

```
firewall flush        [hook = <{input|sink|forward|source|output}>]
```

[hook]	The name of the hook to clear. Choose between: • input • sink • forward • source • output. **Note** If this parameter is not specified, all hooks are cleared.	OPTIONAL

EXAMPLE:

```
=>firewall load
=>firewall list
assign     hook=sink        chain=sink
assign     hook=forward     chain=forward
assign     hook=source      chain=source
=>firewall flush hook=sink
=>firewall list
assign     hook=forward     chain=forward
assign     hook=source      chain=source
=>firewall flush
=>firewall list
=>
```

RELATED COMMANDS:

firewall assign	Assign a chain to an entry point.

firewall list

Show association(s) between all hooks and their chain(s) or of one specified hook

SYNTAX:

`firewall list`	`[hook = <{input|sink|forward|source|output}>]`

[hook]	The name of the hook to show the associations for. Choose between: • input • sink • forward • source • output. **Note** If this parameter is not specified, the associations for all hooks are shown.	OPTIONAL

EXAMPLE:

```
=>firewall list
assign     hook=sink        chain=sink
assign     hook=forward     chain=forward
assign     hook=source      chain=source
=>firewall list hook=input
=>firewall list hook=forward
assign     hook=forward     chain=forward
=>
```

RELATED COMMANDS:

firewall assign	Assign a chain to an entry point.
firewall flush	Clear associations for all or a selected entry point(s).

firewall troff

Disable verbose console messaging.

SYNTAX:

```
firewall troff
```

EXAMPLE:

```
=>firewall troff
```

RELATED COMMANDS:

firewall tron	Enable verbose console messaging.

firewall tron

Enable verbose console messaging.

SYNTAX:

```
firewall tron
```

EXAMPLE:

```
=>firewall tron
```

RELATED COMMANDS:

firewall troff	Disable verbose console messaging.

firewall unassign

Unassign all chains from a hook.

SYNTAX:

`firewall unassign`	`hook = <{input|sink|forward|source|output}>`

hook	The hook's name to unassign all chain from. Choose between: • input • sink • forward • source • output.	REQUIRED

RELATED COMMANDS:

ipsec policy assign	Assign a chain to a hook.
ipsec policy flush	Clear all hooks.
ipsec policy list	Show a list of all chain assignments.

firewall chain create

Create a new chain.

SYNTAX:

```
firewall chain create   chain = <string>
```

chain	The name of the chain to create.	REQUIRED

EXAMPLE:

```
=>firewall chain list
Tempo, source, forward, sink
=>firewall chain create chain=Telnet
=>firewall chain list
Telnet, Tempo, source, forward, sink
=>
```

RELATED COMMANDS:

firewall assign	Assign a chain to an entry point.
firewall chain delete	Delete a chain.
firewall chain list	Show a list of all current chains.

firewall chain delete

Delete a chain.

SYNTAX:

```
firewall chain delete   chain = <string>
```

chain	The name of the chain to be deleted.	REQUIRED

EXAMPLE:

```
=>firewall chain list
Telnet, Tempo, source, forward, sink
=>firewall chain list
Telnet, Tempo, source, forward, sink
=>firewall chain delete chain=Tempo
=>firewall chain list
Telnet, source, forward, sink
=>
```

RELATED COMMANDS:

firewall assign	Assign a chain to an entry point.
firewall chain create	Create a chain.
firewall chain list	Show a list of all chains.

firewall chain list

Show a list of all current chains.

SYNTAX:

```
firewall chain list
```

EXAMPLE INPUT/OUTPUT:

```
=>firewall chain list
source, forward, sink
=>firewall chain create chain Telnet
=>firewall chain list
Telnet, source, forward, sink
=>firewall chain list
Telnet, source, forward, sink
=>
```

RELATED COMMANDS:

firewall assign	Assign a chain to an entry point.
firewall chain create	Create a chain.
firewall chain delete	Delete a chain.

firewall rule clear

Clear statistics for a given rule.

SYNTAX:

```
firewall rule clear   [chain = <string>]
                      [index = <number>]
```

[chain]	The name of the chain in which the rule is to be found.	OPTIONAL
[index]	The index number (determined by the position) of the rule in the chain	OPTIONAL

EXAMPLE:

```
=>firewall rule stats
Chain Telnet,   index 0,     packets 0,     bytes 0
Chain Telnet,   index 1,     packets 0,     bytes 0
Chain Telnet,   index 2,     packets 0,     bytes 0
Chain source,   index 0,     packets 203,   bytes 15229
Chain source,   index 1,     packets 0,     bytes 0
Chain source,   index 2,     packets 0,     bytes 0
Chain forward,  index 0,     packets 0,     bytes 0
Chain sink,     index 0,     packets 202,   bytes 10159
Chain sink,     index 1,     packets 0,     bytes 0
Chain sink,     index 2,     packets 0,     bytes 0
=>firewall rule clear chain=source index=0
=>firewall rule stats
Chain Telnet,   index 0,     packets 0,     bytes 0
Chain Telnet,   index 1,     packets 0,     bytes 0
Chain Telnet,   index 2,     packets 0,     bytes 0
Chain source,   index 0,     packets 11,    bytes 559
Chain source,   index 1,     packets 0,     bytes 0
Chain source,   index 2,     packets 0,     bytes 0
Chain forward,  index 0,     packets 0,     bytes 0
Chain sink,     index 0,     packets 409,   bytes 21535
Chain sink,     index 1,     packets 0,     bytes 0
Chain sink,     index 2,     packets 0,     bytes 0
=>
```

RELATED COMMANDS:

firewall rule create	Create a rule.
firewall rule delete	Delete a specified rule in a chain.
firewall rule flush	Delete all rules in a chain.
firewall rule list	Show a list of all (or a specified) chains' rules.
firewall rule stats	Show statistics for all (or a specified) chains' rules.

firewall rule create

Create a rule.

SYNTAX:

```
firewall rule create  chain = <string>
                      [index = <number>]
                      [srcintf [!]= <string>]
                      [srcintfgrp [!]= <{wan|local|lan}>]
                      [srcbridgeport [!]= <number>]
                      [src [!]= <ip-address>]
                      [srcmsk = <ip-mask(dotted or cidr)>]
                      [dstintf [!]= <string>]
                      [dstintfgrp [!]= <{wan|local|lan}>]
                      [dst [!]= <ip-address>]
                      [dstmsk = <ip-mask(dotted or cidr)>]
                      [tos [!]= <number{1-255}>]
                      [prot [!]= <{<supported IP protocol name>|<number>}>]
                      [syn = <yes|no>]
                      [urg = <yes|no>]
                      [ack = <yes|no>]
                      [srcport [!]= <{<supported TCP/
                      UDP port name>|<number>}>]
                      [srcportend = <{<supported TCP/
                      UDP port name>|<number>}>]
                      [dstport [!]= <{<supported TCP/
                      UDP port name>|<number>}>]
                      [dstportend = <{<supported TCP/
                      UDP port name>|<number>}>]
                      [icmptype [!]= <{<supported ICMP type name>|<number>}>]
                      [icmpcode [!]= <number{0-15}>]
                      [icmpcodeend = <number{0-15}>]
                      [clink = <string>]
                      action = <{accept|deny|drop|count}>
```

chain	The name of the chain to insert the rule in.	REQUIRED
[index]	The number of the rule before which the new rule must be added.	OPTIONAL
[srcintf]	The name of the interface the packet should [or should NOT] arrive on to make this rule apply. (NOT applicable if used in a chain assigned to the output hook)	OPTIONAL

[srcintfgrp]	The interface group the packet should [or should NOT] arrive on. Choose between: • wan • local • lan (NOT applicable if used in a chain assigned to the output hook)	OPTIONAL
[srcbridgeport]	A number between 0 and 6. Represents the bridge port the virtual packet should [or should NOT] arrive on. Use :bridge iflist for a list of available bridge ports.	OPTIONAL
[src]	The source IP address (range) the packet should [or should NOT] come from. (Supports cidr notation).	OPTIONAL
[srcmsk]	The source IP address mask defining the range (see src).	OPTIONAL
[dstintf]	The name of the interface the packet should [or should NOT] be going to. (NOT applicable if used in a chain assigned to the input hook)	OPTIONAL
[dstintfgrp]	The interface group the packet should [or should NOT] be going to. Choose between: • wan • local • lan (NOT applicable if used in a chain assigned to the input hook)	OPTIONAL
[dst]	The destination IP address (range) the packet should [or should NOT] be going to. (supports cidr notation).	OPTIONAL
[dstmsk]	The destination IP address mask defining the range (see dst).	OPTIONAL
[tos]	A number between 0 and 255. Represents the Type Of Service specification which should be expected [or NOT expected] in the IP packet. The Type of Service numbering specification is in accordance to the latest version of RFC1700: Assigned numbers.	OPTIONAL
[prot]	The protocol (name or number) expected [or NOT expected] in the IP packet. Select one of the supported protocol names (See " Supported Internet Protocol (IP) Protocol Names" on page 289 for a listing of protocol names supported by the SpeedTouch™). Alternatively, specify the protocol number.	OPTIONAL
[syn]	Expect TCP SYN flag set (yes) or not (no). In combination with TCP ACK this allows selection of incoming versus outgoing TCP connections.	OPTIONAL
[urg]	Expect TCP URG flag set (yes) or not (no).	OPTIONAL
[ack]	Expect TCP ACK flag set (yes) or not (no).	OPTIONAL

[srcport]	The TCP/UDP port (or beginning of range) the packet should [or should NOT] be from. Select one of the supported TCP/UDP port names (See " Supported TCP/UDP Port Names" on page 290 for a listing of TCP/UDP port names supported by the SpeedTouch™). Alternatively, specify the protocol number.	OPTIONAL
[srcportend]	The source TCP/UDP port range end (inclusive). (Only applicable for ranges) Select one of the supported TCP/UDP port names (See " Supported TCP/UDP Port Names" on page 290 for a listing of TCP/UDP port names supported by the SpeedTouch™). Alternatively, specify the protocol number.	OPTIONAL
[dstport]	The TCP/UDP port (or beginning of range) the packet should [or should NOT] be going to. Select one of the supported TCP/UDP port names (See " Supported TCP/UDP Port Names" on page 290 for a listing of TCP/UDP port names supported by the SpeedTouch™). Alternatively, specify the protocol number.	OPTIONAL
[dstportend]	The destination TCP/UDP port range end (inclusive). (Only applicable for ranges) Select one of the supported TCP/UDP port names (See " Supported TCP/UDP Port Names" on page 290 for a listing of TCP/UDP port names supported by the SpeedTouch™). Alternatively, specify the protocol number.	OPTIONAL
[icmptype]	The expected [or NOT expected] ICMP type (name or number) of the packet. Select one of the supported ICMP type names (See " Supported ICMP Type Names" on page 293 for a listing of ICMP type names supported by the SpeedTouch™). Alternatively, specify the protocol number.	OPTIONAL
[icmpcode]	A number between 0 and 15. Represents the expected [or NOT expected] ICMP code (or beginning of range) of the packet as specified in the latest version of RFC1700: Assigned number.	OPTIONAL
[icmpcodeend]	A number between 0 and 15. Represents the ICMP code range end. Only applicable for ranges.	OPTIONAL
[clink]	The name of the chain to be parsed when this rule applies. (action is ignored).	OPTIONAL
action	Action to be taken when this rule applies. Choose between: • accept : the packet may pass. • deny : ICMP error destination unreachable. An error message is sent back to the sender. • drop : packet disappears. It is silently dropped, that is, without sending an error message to the sender. • count : update of statistics. Has no influence on the packet.	REQUIRED

RELATED COMMANDS:

firewall rule clear	Clear statistics of a given rule.
firewall rule delete	Delete a specified rule in a chain.
firewall rule flush	Delete all rules in a chain.
firewall rule list	Show a list of all (or a specified) chains' rules.
firewall rule stats	Show statistics for all (or a specified) chains' rules.

firewall rule delete

Delete a rule.

SYNTAX:

```
firewall rule delete   chain = <string>
                       index = <number>
```

chain	The name of the chain in which to delete the rule.	REQUIRED
index	The index number of the rule in the chain. Use :firewall rule list first to determine the index number of the applicable rule.	REQUIRED

EXAMPLE:

```
=>firewall rule list   chain=Telnet
:firewall  rule create chain=Telnet index=0 srcintfgrp=lan src=10.0.0.0/8
dst=200.200.200.1/32 prot=tcp srcport=1024 srcportend=65535 dstport=telnet
action=accept
:firewall  rule create chain=Telnet index=1 srcintfgrp=wan
src=200.200.200.1/32 dst=10.0.0.0/8 prot=tcp srcport=telnet dstport=1024
dstportend=65535 action=accept
:firewall  rule create chain=Telnet index=2 action=drop
=>firewall rule delete chain=Telnet index=1
=>firewall rule list   chain=Telnet
:firewall  rule create chain=Telnet index=0 srcintfgrp=lan src=10.0.0.0/8
dst=200.200.200.1/32 prot=tcp srcport=1024 srcportend=65535 dstport=telnet
action=accept
:firewall  rule create chain=Telnet index=1 action=drop
=>
```

RELATED COMMANDS:

firewall rule clear	Clear statistics of a given rule.
firewall rule create	Create a rule.
firewall rule flush	Delete all rules in a chain.
firewall rule list	Show a list of all (or a specified) chains' rules.
firewall rule stats	Show statistics for all (or a specified) chains' rules.

firewall rule flush

Flush all rules created for a chain(s). The chain itself is not removed.

Note The flush command does not impact previously saved configurations.

SYNTAX:

```
firewall rule flush  [chain = <string>]
```

[chain]	The name of the chain to empty. **Note** If this parameter is not specified, all rules for all chains are deleted.	OPTIONAL

EXAMPLE:

```
=>firewall rule list   chain=Telnet
:firewall  rule create chain=Telnet index=0 srcintfgrp=lan src=10.0.0.0/8 |
dst=200.200.200.1/32 prot=tcp srcport=1024 srcportend=65535 dstport=telnet |
action=accept
:firewall  rule create chain=Telnet index=1 srcintfgrp=wan |
src=200.200.200.1/32 dst=10.0.0.0/8 prot=tcp srcport=telnet dstport=1024 |
dstportend=65535 action=accept
:firewall  rule create chain=Telnet index=2 action=drop
=>firewall rule flush  chain=Telnet
=>firewall rule list   chain=Telnet
=>
```

RELATED COMMANDS:

firewall rule clear	Clear statistics of a given rule.
firewall rule create	Create a rule.
firewall rule delete	Delete a specified rule.
firewall rule list	Show a list of all (or a specified) chains' rules.
firewall rule stats	Show statistics for all (or a specified) chains' rules.

firewall rule list

Show a list of rules.

SYNTAX:

```
firewall rule list    [chain = <string>]
```

[chain]	The name of the chain to list the rules of. **Note** If this parameter is not specified, all rules for all chains are shown.	OPTIONAL

EXAMPLE INPUT AND OUTPUT:

```
=>firewall rule list   chain=Telnet
:firewall  rule create chain=Telnet index=0 srcintfgrp=lan src=10.0.0.0/8 |
dst=200.200.200.1/32 prot=tcp srcport=1024 srcportend=65535 dstport=telnet |
action=accept
:firewall  rule create chain=Telnet index=1 srcintfgrp=wan |
src=200.200.200.1/32 dst=10.0.0.0/8 prot=tcp srcport=telnet dstport=1024 |
dstportend=65535 action=accept
:firewall  rule create chain=Telnet index=2 action=drop
=>firewall rule list
:firewall  rule create chain=source index=0 dstintfgrp=!wan action=accept
:firewall  rule create chain=source index=1 prot=udp dstport=dns |
action=accept
:firewall  rule create chain=source index=2 prot=udp dstport=67 action=accept
:firewall  rule create chain=source index=3 action=drop
:firewall  rule create chain=forward index=0 srcintfgrp=wan dstintfgrp=wan |
action=drop
:firewall  rule create chain=sink index=0 srcintf=eth0 srcbridgeport=1 |
action=accept
:firewall  rule create chain=sink index=1 srcintfgrp=!wan action=accept
:firewall  rule create chain=sink index=2 prot=udp dstport=dns action=accept
:firewall  rule create chain=sink index=3 prot=udp dstport=68 action=accept
:firewall  rule create chain=sink index=4 action=drop
:firewall  rule create chain=Telnet index=0 srcintfgrp=lan src=10.0.0.0/8 |
dst=200.200.200.1/32 prot=tcp srcport=1024 srcportend=65535 dstport=telnet |
action=accept
:firewall  rule create chain=Telnet index=1 srcintfgrp=wan |
src=200.200.200.1/32 dst=10.0.0.0/8 prot=tcp srcport=telnet dstport=1024 |
dstportend=65535 action=accept
:firewall  rule create chain=Telnet index=2 action=drop
=>
```

RELATED COMMANDS:

firewall rule clear	Clear statistics of a given rule.
firewall rule create	Create a rule.
firewall rule delete	Delete a specified rule.
firewall rule flush	Delete all rules in a chain.
firewall rule stats	Show statistics for all (or a specified) chains' rules.

firewall rule stats

Show statistics, i. e. the number of packets and bytes which have passed the hooks.

SYNTAX:

```
firewall rule stats   [chain = <string>]
                      [index = <number>]
```

[chain]	The name of the chain of which the statistics must be listed. In case this parameter is not specified the statistics for the rules applicable to all chains are shown.	OPTIONAL
[index]	The index number of the chain's rule of which the statistics must be listed. Execute firewall rule list first to determine the index number of the applicable rule. In case this parameter is not specified the statistics for all rules applicable to the specified chain are shown.	OPTIONAL

EXAMPLE OUTPUT:

```
=>firewall rule list   chain=Test
:firewall  rule create chain=Test index=0 srcintfgrp=lan src=200.200.0.1/32
dst=200.200.0.2/32 prot=udp srcport=0 srcportend=65535 dstport=telnet
action=deny
=>firewall  rule clear
=>firewall  rule stats
Chain sink,    index 0,  packets 43,   bytes 1743
Chain sink,    index 1,  packets 0,    bytes 0
Chain sink,    index 2,  packets 0,    bytes 0
Chain sink,    index 3,  packets 0,    bytes 0
Chain forward, index 0,  packets 0,    bytes 0
Chain source,  index 0,  packets 43,   bytes 1977
Chain source,  index 1,  packets 0,    bytes 0
Chain source,  index 2,  packets 0,    bytes 0
Chain Test,    index 0,  packets 0,    bytes 0
=>firewall rule stats
Chain sink,    index 0,  packets 104,  bytes 6143
Chain sink,    index 1,  packets 0,    bytes 0
Chain sink,    index 2,  packets 0,    bytes 0
Chain sink,    index 3,  packets 0,    bytes 0
Chain forward, index 0,  packets 0,    bytes 0
Chain source,  index 0,  packets 43,   bytes 1977
Chain source,  index 1,  packets 0,    bytes 0
Chain source,  index 2,  packets 0,    bytes 0
Chain Test,    index 0,  packets 44,   bytes 21032
=>
```

DESCRIPTION:

The statistics for the ‘Test‘ chain are the result of sending udp packets to the SpeedTouch™. The chain ‘Test‘ is assigned to the hook ‘input‘ and prohibits the sending of udp packets from one host to another.

EXAMPLE INPUT/OUTPUT IN A NETWORKED ENVIRONMENT:

The SpeedTouch™ is configured as DHCP client on its Ethernet interface eth0.

```
=>firewall rule list   chain=Sending
:firewall  rule create chain=Sending index=0 srcintfgrp=lan src=10.0.0.3/32
dst=10.10.1.1/32 prot=icmp action=count
:firewall  rule create chain=Sending index=1 srcintfgrp=lan src=10.10.1.1/32
dst=10.0.0.3/32 prot=icmp action=count
=>firewall rule stats
Chain source,   index 0,  packets 0,    bytes 0
Chain source,   index 1,  packets 0,    bytes 0
Chain source,   index 2,  packets 0,    bytes 0
Chain source,   index 3,  packets 0,    bytes 0
Chain forward,  index 0,  packets 0,    bytes 0
Chain sink,     index 0,  packets 0,    bytes 0
Chain sink,     index 1,  packets 144,  bytes 5844
Chain sink,     index 2,  packets 0,    bytes 0
Chain sink,     index 3,  packets 0,    bytes 0
Chain sink,     index 4,  packets 0,    bytes 0
Chain sink,     index 5,  packets 0,    bytes 0
Chain sending,  index 0,  packets 0,    bytes 0
Chain sending,  index 1,  packets 0,    bytes 0
=>firewall rule clear
=>(Ping from server 10.10.1.1 to client 10.0.0.3)
=>firewall rule stats
Chain source,   index 0,  packets 0,    bytes 0
Chain source,   index 1,  packets 0,    bytes 0
Chain source,   index 2,  packets 0,    bytes 0
Chain source,   index 3,  packets 0,    bytes 0
Chain forward,  index 0,  packets 0,    bytes 0
Chain sink,     index 0,  packets 0,    bytes 0
Chain sink,     index 1,  packets 42,   bytes 1782
Chain sink,     index 2,  packets 0,    bytes 0
Chain sink,     index 3,  packets 0,    bytes 0
Chain sink,     index 4,  packets 0,    bytes 0
Chain sink,     index 5,  packets 0,    bytes 0
Chain sending,  index 0,  packets 4,    bytes 240
Chain sending,  index 1,  packets 4,    bytes 240
=>
```

RELATED COMMANDS:

firewall rule clear	Clear statistics of a given rule.
firewall rule create	Create a rule.
firewall rule delete	Delete a specified rule.
firewall rule flush	Delete all rules in a chain.
firewall rule list	Show a list of all (or a specified) chains' rules.

IP Commands

Contents

This chapter covers the following commands:

ip apadd

Assign an Internet Protocol (IP) address to an interface.

SYNTAX:

```
ip apadd                addr = <ip-address>
                        [netmask = <ip-mask (dotted or cidr)>]
                        intf = <interface name>
                        [pointopoint = <ip-address>]
                        [addrtrans = <{none|pat}>]
                        [addroute = <{no|yes}>]
                        [type = <number>]
```

addr	The new IP address to add.	REQUIRED
[netmask]	The subnetmask associated with this address.	OPTIONAL
intf	The interface name.	REQUIRED
[pointopoint]]	The remote IP address in case of a dedicated point-to-point link.	OPTIONAL
[addrtrans]	Indicates whether network address translation mode is allowed (pat) for this IP address or not (none).	OPTIONAL
[addroute]	Add typical net/subnet routes automatically according to the default (or specified) subnet mask (yes) or not (no).	OPTIONAL
[type]	The type of address classification. For internal use only.	OPTIONAL

EXAMPLE:

```
=>ip aplist
1 eth0  Type:Ethernet  HWaddr  00:80:9f:24:ab:cf  BRHWaddr  ff:ff:ff:ff:ff:ff
  inet  addr:10.10.10.147    Bcast:10.10.10.255     Mask:255.0.0.0
  UP RUNNING  pat  MTU:1500  ReasmMAX:65535  Group:2
  IPRX  bytes:19791886  unicastpkts:11341  brcastpkts:290555
  IPTX  bytes:839550    unicastpkts:11477  brcastpkts:0  droppkts:0
  HWRX  bytes:0         unicastpkts:0      brcastpkts:0
  HWTX  bytes:0         unicastpkts:0      brcastpkts:0  droppkts:0
0 loop  Type:0
  inet  addr:127.0.0.1       Bcast:127.255.255.255  Mask:255.0.0.0
  UP RUNNING       MTU:1500  ReasmMAX:65535  Group:1
  IPRX  bytes:116       unicastpkts:0      brcastpkts:2
  IPTX  bytes:0         unicastpkts:0      brcastpkts:0  droppkts:0
  HWRX  bytes:0         unicastpkts:0      brcastpkts:0
  HWTX  bytes:0         unicastpkts:0      brcastpkts:0  droppkts:0
=>ip apadd addr=10.0.0.2 netmask=255.255.255.0 intf=eth0 addrtrans=pat
addroute=yes
=>ip aplist
2 eth0  Type:Ethernet  HWaddr  00:80:9f:24:ab:cf  BRHWaddr  ff:ff:ff:ff:ff:ff
  inet  addr:  10.0.0.2      Bcast:10.0.0.255       Mask:255.255.255.0
  UP RUNNING  pat  MTU:1500  ReasmMAX:65535  Group:2
  IPRX  bytes:0         unicastpkts:0      brcastpkts:0
  IPTX  bytes:0         unicastpkts:0      brcastpkts:0  droppkts:0
  HWRX  bytes:0         unicastpkts:0      brcastpkts:0
  HWTX  bytes:0         unicastpkts:0      brcastpkts:0  droppkts:0
1 eth0  Type:Ethernet  HWaddr  00:80:9f:24:ab:cf  BRHWaddr  ff:ff:ff:ff:ff:ff
  inet  addr:  10.10.10.147  Bcast:  10.10.10.255     Mask:  255.0.0.0
  UP RUNNING  pat  MTU:1500  ReasmMAX:65535  Group:2
  IPRX  bytes:19810763  unicastpkts:11515  brcastpkts:290669
  IPTX  bytes:853114    unicastpkts:11662  brcastpkts:0  droppkts:0
  HWRX  bytes:0         unicastpkts:0      brcastpkts:0
  HWTX  bytes:0         unicastpkts:0      brcastpkts:0  droppkts:0
0 loop  Type:0
  inet  addr:127.0.0.1       Bcast:127.255.255.255    Mask:255.0.0.0
  UP RUNNING       MTU:1500  ReasmMAX:65535  Group:1
  IPRX  bytes:116       unicastpkts:0      brcastpkts:2
  IPTX  bytes:0         unicastpkts:0      brcastpkts:0  droppkts:0
  HWRX  bytes:0         unicastpkts:0      brcastpkts:0
  HWTX  bytes:0         unicastpkts:0      brcastpkts:0  droppkts:0
=>
```

RELATED COMMANDS:

ip apdelete	Remove an IP address from an interface.
ip aplist	Show current IP addresses.

ip apdelete

Remove an IP address from an interface.

SYNTAX:

```
ip apdelete          addr = <ip-address>
```

addr	The IP address to delete.	REQUIRED

EXAMPLE:

```
=>ip aplist
2 eth0  Type:Ethernet  HWaddr  00:80:9f:24:ab:cf  BRHWaddr  ff:ff:ff:ff:ff:ff
  inet  addr:10.0.0.2        Bcast:10.0.0.255       Mask:255.255.255.0
  UP RUNNING  pat       MTU:1500  ReasmMAX:65535  Group:2
  IPRX   bytes:0         unicastpkts:0      brcastpkts:0
  IPTX   bytes:0         unicastpkts:0      brcastpkts:0  droppkts:0
  HWRX   bytes:0         unicastpkts:0      brcastpkts:0
  HWTX   bytes:0         unicastpkts:0      brcastpkts:0  droppkts:0
1 eth0  Type:Ethernet  HWaddr  00:80:9f:24:ab:cf  BRHWaddr  ff:ff:ff:ff:ff:ff
  inet   addr:10.10.10.147   Bcast:   10.10.10.255  Mask:   255.0.0.0
  UP RUNNING  pat  MTU:1500  ReasmMAX:65535  Group:2
  IPRX   bytes:19791886  unicastpkts:11341  brcastpkts:290555
  IPTX   bytes:839550    unicastpkts:11477  brcastpkts:0  droppkts:0
  HWRX   bytes:0         unicastpkts:0      brcastpkts:0
  HWTX   bytes:0         unicastpkts:0      brcastpkts:0  droppkts:0
0 loop  Type:0
  inet   addr:127.0.0.1      Bcast:127.255.255.255  Mask:255.0.0.0
  UP RUNNING            MTU:1500  ReasmMAX:65535  Group:1
  IPRX   bytes:116       unicastpkts:0      brcastpkts:2
  IPTX   bytes:0         unicastpkts:0      brcastpkts:0  droppkts:0
  HWRX   bytes:0         unicastpkts:0      brcastpkts:0
  HWTX   bytes:0         unicastpkts:0      brcastpkts:0  droppkts:0
=>ip apdelete addr=10.0.0.2
=>ip aplist
1 eth0  Type:Ethernet  HWaddr  00:80:9f:24:ab:cf  BRHWaddr  ff:ff:ff:ff:ff:ff
  inet  addr:10.10.10.147    Bcast:   10.10.10.255  Mask:   255.0.0.0
  UP   RUNNING    pat   MTU:1500    ReasmMAX:65535   Group:2
  IPRX   bytes:19791886  unicastpkts:11341  brcastpkts:290555
  IPTX   bytes:839550    unicastpkts:11477  brcastpkts:0  droppkts:0
  HWRX   bytes:0         unicastpkts:0      brcastpkts:0
  HWTX   bytes:0         unicastpkts:0      brcastpkts:0  droppkts:0
0 loop         Type:0
  inet   addr:127.0.0.1      Bcast:127.255.255.255  Mask:255.0.0.0
  UP   RUNNING          MTU:1500      ReasmMAX:65535   Group:1
  IPRX   bytes:116       unicastpkts:0      brcastpkts:2
  IPTX   bytes:0         unicastpkts:0      brcastpkts:0  droppkts:0
  HWRX   bytes:0         unicastpkts:0      brcastpkts:0
  HWTX   bytes:0         unicastpkts:0      brcastpkts:0  droppkts:0
=>
```

RELATED COMMANDS:

ip apadd	Add an IP address to an interface.
ip aplist	Show current IP addresses.

ip aplist

Show a list of all configured IP addresses.

SYNTAX:

```
ip aplist
```

EXAMPLE:

```
=>ip aplist
2 eth0  Type:Ethernet  HWaddr  00:80:9f:24:ab:cf  BRHWaddr  ff:ff:ff:ff:ff:ff
  inet addr:   10.0.0.2      Bcast:   10.0.0.255     Mask:   255.255.255.0
  UP RUNNING    pat   MTU:1500    ReasmMAX:65535   Group:2
  IPRX   bytes:0         unicastpkts:0      brcastpkts:0
  IPTX   bytes:0         unicastpkts:0      brcastpkts:0     droppkts:0
  HWRX   bytes:0         unicastpkts:0      brcastpkts:0
  HWTX   bytes:0         unicastpkts:0      brcastpkts:0     droppkts:0
1 eth0  Type:Ethernet  HWaddr  00:80:9f:24:ab:cf  BRHWaddr  ff:ff:ff:ff:ff:ff
  inet addr:   10.10.10.147    Bcast:   10.10.10.255    Mask:   255.0.0.0
  UP RUNNING    pat   MTU:1500    ReasmMAX:65535   Group:2
  IPRX   bytes:19791886  unicastpkts:11341  brcastpkts:290555
  IPTX   bytes:839550    unicastpkts:11477  brcastpkts:0     droppkts:0
  HWRX   bytes:0         unicastpkts:0      brcastpkts:0
  HWTX   bytes:0         unicastpkts:0      brcastpkts:0     droppkts:0
0 loop  Type:0
  inet   addr:127.0.0.1     Bcast:127.255.255.255    Mask:255.0.0.0
  UP RUNNING          MTU:1500    ReasmMAX:65535   Group:1
  IPRX   bytes:116       unicastpkts:0      brcastpkts:2
  IPTX   bytes:0         unicastpkts:0      brcastpkts:0     droppkts:0
  HWRX   bytes:0         unicastpkts:0      brcastpkts:0
  HWTX   bytes:0         unicastpkts:0      brcastpkts:0     droppkts:0
=>
```

RELATED COMMANDS:

ip apadd	Add an IP address to an interface.
ip apdelete	Remove an IP address from an interface.

ip arpadd

Add a static entry to the SpeedTouch™ ARP cache.

SYNTAX:

```
ip arpadd              intf = <interface name>
                       ip = <ip-address>
                       [hwaddr = <hardware-address>]
```

intf	The interface name.	REQUIRED
ip	The IP address.	REQUIRED
[hwaddr]	The hardware address (e.g. the Ethernet MAC address).	OPTIONAL

EXAMPLE:

```
=>ip arplist
Intf            IP-address        HW-address              Type
eth0            10.0.0.1          00:01:42:5f:7d:81       DYNAMIC
eth0            10.0.0.8          00:a0:24:ae:66:e1       DYNAMIC
eth0            10.0.1.99         52:41:53:20:20:4d       STATIC
eth0            10.0.1.100        52:41:53:20:f0:90       STATIC
=>ip arpadd intf=eth0 ip=10.0.0.2 hwaddr=00:10:a4:d0:9a:db
=>ip arplist
Intf            IP-address        HW-address              Type
eth0            10.0.0.1          00:01:42:5f:7d:81       DYNAMIC
eth0            10.0.0.8          00:a0:24:ae:66:e1       DYNAMIC
eth0            10.0.1.99         52:41:53:20:20:4d       STATIC
eth0            10.0.1.100        52:41:53:20:f0:90       STATIC
eth0            10.0.0.2          00:10:a4:d0:9a:db       STATIC
=>
```

RELATED COMMANDS:

ip arpdelete	Delete an ARP entry.
ip arplist	Show current ARP cache.

ip arpdelete

Remove an entry from the SpeedTouch™ ARP cache.

SYNTAX:

```
ip arpdelete          intf = <interface name>
                      ip = <ip-address>
                      [hwaddr = <hardware-address>]
```

intf	The interface name.	REQUIRED
ip	The IP address.	REQUIRED
[hwaddr]	The hardware address.	OPTIONAL

EXAMPLE:

```
=>ip arplist
Intf            IP-address        HW-address               Type
eth0            10.0.0.1          00:01:42:5f:7d:81        DYNAMIC
eth0            10.0.0.8          00:a0:24:ae:66:e1        DYNAMIC
eth0            10.0.1.99         52:41:53:20:20:4d        STATIC
eth0            10.0.1.100        52:41:53:20:f0:90        STATIC
eth0            10.0.0.2          00:10:a4:d0:9a:db        STATIC
=>ip arpdelete intf=eth0 ip=10.0.0.2 hwaddr=00:10:a4:d0:9a:db
=>ip arplist
Intf            IP-address        HW-address               Type
eth0            10.0.0.1          00:01:42:5f:7d:81        DYNAMIC
eth0            10.0.0.8          00:a0:24:ae:66:e1        DYNAMIC
eth0            10.0.1.99         52:41:53:20:20:4d        STATIC
eth0            10.0.1.100        52:41:53:20:f0:90        STATIC
eth0            10.0.0.2          00:10:a4:d0:9a:db        STATIC
=>
```

RELATED COMMANDS:

ip arpadd	Add a static ARP entry.
ip arplist	Show current ARP cache.

ip arplist

Show the SpeedTouch™ ARP cache.

SYNTAX:

```
ip arplist
```

EXAMPLE OUTPUT:

```
=>ip arplist
Intf            IP-address        HW-address              Type
eth0            10.0.0.1          00:01:42:5f:7d:81       DYNAMIC
eth0            10.0.0.8          00:a0:24:ae:66:e1       DYNAMIC
eth0            10.0.1.99         52:41:53:20:20:4d       STATIC
eth0            10.0.1.100        52:41:53:20:f0:90       STATIC
eth0            10.0.0.2          00:10:a4:d0:9a:db       STATIC
=>
```

RELATED COMMANDS:

ip arpadd	Add a static entry to the ARP cache.
ip arpdelete	Delete an entry from the ARP cache.

ip config

Show/set global IP stack configuration options.

SYNTAX:

```
ip config               [forwarding = <{off|on}>]
                        [firewalling = <{off|on}>]
                        [redirects = <{off|on}>]
                        [sourcerouting = <{off|on}>]
                        [netbroadcasts = <{off|on}>]
                        [ttl = <number{0-255}>]
                        [fraglimit = <number{1-1024}>]
                        [defragmode = <{normal|always|nat}>]
                        [addrcheck = <{off|own|static|dynamic}>]
                        [mssclamping = <{off|on}>]
```

[forwarding]	Disable (off) or enable (on) the IP routing functionality.	OPTIONAL
[firewalling]	Enable (on) or disable (off) IP firewalling (master switch). If applicable the CLI firewall level allows configuration of the SpeedTouch™ firewall. For security reasons this parameter is enabled per default. It is strongly recommended never to disable the SpeedTouch™ firewall.	OPTIONAL
[redirects]	Disable (off) or enable (on) the sending of ICMP redirect messages. A router can send a redirect message in case a shorter path than the path followed is discovered. For security reasons this parameter is disabled per default.	OPTIONAL
[sourcerouting]	Disallow (off) or allow (on) IP source routed packets. IP source routed packets are packets with the route to follow specified in the header. For security reasons this parameter is disabled per default.	OPTIONAL
[netbroadcasts]	Disallow (off) or allow (on) net directed broadcasts. This parameter is per default disabled. In case netbroadcasts are allowed no traces of netbroadcasts are generated.	OPTIONAL
[ttl]	A number between 0 and 255. Represents the default time-to-live (ttl) for locally generated IP packets. This parameter determines the number of hop-counts the IP packet may pass before it is dropped. Generally the time-to-live is 64 hop-counts. By limiting the time-to-live continuous circulation of IP packets on the network without ever reaching a destination is avoided.	OPTIONAL

[fraglimit]	A number between 1 and 1024. Represents the maximum number of IP packet fragments waiting for completion. Generally the fragmentation limit is 64. By limiting the fragmentation limit the depletion of the buffer is avoided.	OPTIONAL
[defragmode]	Define which packets are reassembled under which circumstances. Choose between: • normal Packets to be forwarded will not be reassembled. Packets with local destination, i.e. destined for the SpeedTouch™ , are reassembled. • always Packets are always reassembled. • nat Same behaviour as normal except for packets to be forwarded through the Network Address Translation (NAT) engine. Packets on which address translation is performed are reassembled as the NAT engine requires the entire packet.	OPTIONAL
[addrcheck]	Set the level of IP address checks. Choose between: • off No address checking is performed. For advanced users only; in normal circumstances there should always be some kind of address checking. • own Minimum level of checking. Only the address configuration on the SpeedTouch™ is checked. • static Checking of the address configuration of the SpeedTouch™ and also of traffic: addresses of incoming packets; this checking is related to constants (e. g. an address may not be entirely composed of one's or zero's). • dynamic Besides the address configuration of the SpeedTouch™ itself, and besides the checking of traffic on a constants level, additional checking is performed on the IP addresses that are determined by the configuration, more specifically by the network.	OPTIONAL
[mssclamping]	Disable (off) or enable (on) mss clamping for low mtu interfaces. Mss clamping assures that the size of a TCP packet never exceeds the available mtu of the outgoing interface. It is recommended not to disable this parameter.	OPTIONAL

EXAMPLE:

```
=>ip config
Forwarding on
Firewalling off
Sendredirects off
Sourcerouting on
NetBroadcasts off
Default TTL 128
Fraglimit 32 fragments
Fragcount currently  0 fragments
Defragment mode : always
Address checks : static
Mss clamping : on
=>ip config firewalling=on ttl=64 fraglimit=64 defragmode=nat
=>ip config
Forwarding on
Firewalling on
Sendredirects off
Sourcerouting on
NetBroadcasts off
Default TTL 64
Fraglimit 64 fragments
Fragcount currently  0 fragments
Defragment mode : nat
Address checks : static
Mss clamping : on
=>
```

RELATED COMMANDS:

ip ifconfig	Configure interface parameters.

ip flush

Flush complete IP configuration. Dynamic configurations (e.g. from PPP or CIP links) remain.

Note As an ip flush causes all local IP connectivity to be deleted, do not execute this command during an IP based local connection, e.g. a Telnet CLI session, or web based CLI access.

Note The flush command does not impact previously saved configurations.

SYNTAX:

```
ip flush
```

EXAMPLE:

```
=>ip aplist
3  cip1     Type:ATM
   inet   addr:172.16.0.5     Bcast:172.16.0.255     Mask:255.255.255.0
   UP  RUNNING    pat  MTU:9180    ReasmMAX:65535   Group:0
   IPRX   bytes:0      unicastpkts:0     brcastpkts:0
   IPTX   bytes:0      unicastpkts:0     brcastpkts:0     droppkts:0
   HWRX   bytes:0      unicastpkts:0     brcastpkts:0
   HWTX   bytes:0      unicastpkts:0     brcastpkts:0     droppkts:0
2  eth0     Type:EthernetHWaddr 00:80:9f:24:ab:cf  BRHWaddr ff:ff:ff:ff:ff:ff
   inet addr: 10.0.0.2   Bcast: 10.0.0.255         Mask: 255.255.255.0
   UP  RUNNING    pat  MTU:1500    ReasmMAX:65535   Group:2
   IPRX   bytes:0      unicastpkts:0     brcastpkts:0
   IPTX   bytes:0      unicastpkts:0     brcastpkts:0     droppkts:0
   HWRX   bytes:0      unicastpkts:0     brcastpkts:0
   HWTX   bytes:0      unicastpkts:0     brcastpkts:0     droppkts:0
0  loop    Type:0
   inet   addr:127.0.0.1     Bcast:127.255.255.255    Mask:255.0.0.0
   UP  RUNNING         MTU:1500    ReasmMAX:65535   Group:1
   IPRX   bytes:116    unicastpkts:0     brcastpkts:2
   IPTX   bytes:0      unicastpkts:0     brcastpkts:0     droppkts:0
   HWRX   bytes:0      unicastpkts:0     brcastpkts:0
   HWTX   bytes:0      unicastpkts:0     brcastpkts:0     droppkts:0
=>ip flush
=>ip aplist
3  cip1     Type:ATM
   inet   addr:172.16.0.5     Bcast:172.16.0.255     Mask:255.255.255.0
   UP  RUNNING    pat  MTU:9180    ReasmMAX:65535   Group:0
   IPRX   bytes:0      unicastpkts:0     brcastpkts:0
   IPTX   bytes:0      unicastpkts:0     brcastpkts:0     droppkts:0
   HWRX   bytes:0      unicastpkts:0     brcastpkts:0
   HWTX   bytes:0      unicastpkts:0     brcastpkts:0     droppkts:0
0  loop     Type:0
   inet addr: 127.0.0.1 Bcast:127.255.255.255      Mask:255.0.0.0
   UP  RUNNING         MTU:1500    ReasmMAX:65535   Group:1
   IPRX   bytes:116    unicastpkts:0     brcastpkts:2
   IPTX   bytes:0      unicastpkts:0     brcastpkts:0     droppkts:0
   HWRX   bytes:0      unicastpkts:0     brcastpkts:0
   HWTX   bytes:0      unicastpkts:0     brcastpkts:0     droppkts:0
=>
```

ip ifconfig

Configure interface parameters.

SYNTAX:

```
ip ifconfig          intf = <interface name>
                     [mtu = <number{293-20000}>]
                     [status = <{down|up}>]
                     [hwaddr = <hardware-address>]
                     [group = <{wan|local|lan}>]
```

intf	The IP interface name.	REQUIRED
[mtu]	A number between 293 and 20000. Represents the maximum transmission unit, i.e. the maximum packet size (including IP header) to use on this interface. The default value depends on the connection and packet service for which the interface was created.	OPTIONAL
[status]	The administrative status of the interface. Choose between: • down • up	OPTIONAL
[hwaddr]	The hardware address (e.g. the Ethernet MAC address) of this interface.	OPTIONAL
[group]	The group this interface belongs to (e.g. for oriented firewalling).	OPTIONAL

EXAMPLE:

```
=>ip iflist
Interface   GRP  MTU   RX        TX        TX-DROP  STATUS  HWADDR
0   loop    1    1500  116       0         0        UP
1   eth0    2    3000  21045795  1019664   0        UP      00:80:9f:24:ab:cf
2   NewMer  0    1500  0         0         0        UP      00:80:9f:24:ab:cf
5   cip0    0    9180  0         0         0        UP
=>ip ifconfig intf=eth0 mtu=1500
=>ip iflist
Interface  GRP   MTU   RX        TX        TX-DROP  STATUS  HWADDR
0   loop    1    1500  116       0         0        UP
1   eth0    2    1500  21054963  1025417   0        UP      00:80:9f:24:ab:cf
2   NewMer  0    1500  0         0         0        UP      00:80:9f:24:ab:cf
5   cip0    0    9180  0         0         0        UP
=>
```

RELATED COMMANDS:

ip config	Show/set global IP stack configuration options.

ip iflist

Show all current interfaces.

SYNTAX:

```
ip iflist
```

EXAMPLE OUTPUT:

```
=>ip iflist
0   loop    1    1500  116       0        0        UP
1   eth0    2    3000  21045795  1019664  0        UP      00:80:9f:24:ab:cf
2   NewMer  0    1500  0         0        0        UP      00:80:9f:24:ab:cf
5   cip0    0    9180  0         0        0        UP
=>
```

RELATED COMMANDS:

ip ifconfig	Configure interface parameters.

ip ping

Send ICMP ECHO_REQUEST packets.

SYNTAX:

```
ip ping                 addr = <ip-address>
                        [count = <number{1-1000000}>]
                        [size = <number{1-20000}>]
                        [interval = <number{100-1000000}>]
                        [listen = <{off|on}>]
```

addr	The destination IP address.	REQUIRED
[count]	A number between 1 and 1000000. Represents the number of pings to send.	OPTIONAL
[size]	A number between 1 and 20000 (bytes). Represents the size of the ping packet(s).	OPTIONAL
[interval]	A number between 100 and 10000000 (milliseconds). Represents the intermediate interval between two sent ICMP packets.	OPTIONAL
[listen]	Listen for incoming ICMP packets (on) or only send ICMP packets (off).	OPTIONAL

EXAMPLE:

```
=>ip ping addr=10.0.0.148 listen=off
=>ip ping addr=10.0.0.148 listen=on
9 bytes from 10.0.0.148: Echo Request
=>ip ping addr=10.0.0.148 count=15 listen=on
9 bytes from 10.0.0.148: Echo Request
9 bytes from 10.0.0.148: Echo Request
9 bytes from 10.0.0.148: Echo Request
9 bytes from 10.0.0.148: Echo Request
9 bytes from 10.0.0.148: Echo Request
9 bytes from 10.0.0.148: Echo Request
9 bytes from 10.0.0.148: Echo Request
9 bytes from 10.0.0.148: Echo Request
9 bytes from 10.0.0.148: Echo Request
9 bytes from 10.0.0.148: Echo Request
9 bytes from 10.0.0.148: Echo Request
9 bytes from 10.0.0.148: Echo Request
9 bytes from 10.0.0.148: Echo Request
9 bytes from 10.0.0.148: Echo Request
9 bytes from 10.0.0.148: Echo Request
=>
```

RELATED COMMANDS:

ip sendto	Send UDP packets.

ip rtadd

Add a route to the SpeedTouch™ routing table.

SYNTAX:

```
ip rtadd                dst = <ip-address>
                        [dstmsk = <ip-mask(dotted or cidr)>]
                        [src = <ip-address>]
                        [srcmsk = <ip-mask(dotted or cidr)>]
                        [gateway = <ip-address>]
                        [intf = <interface name>]
                        [metric = <number{0-100}>]
                        [type = <number>]
```

dst	The destination IP address(es) for this route. Supports cidr notation.	REQUIRED
[dstmsk]	The destination IP address mask.	OPTIONAL
[src]	The source IP address(es) allowed to use this route. Supports cidr notation.	OPTIONAL
[srcmsk]	The source IP address mask.	OPTIONAL
[gateway]	The IP address of the next hop. Must be directly connected. The parameters 'gateway' and 'intf' are mutually exclusive.	OPTIONAL
[intf]	Only for special interface routes : the outgoing IP interface name. The parameters 'gateway' and 'intf' are mutually exclusive.	OPTIONAL
[metric]	The metric for this route (currently not used).	OPTIONAL
[type]	Route classification. For internal use only.	OPTIONAL

EXAMPLE:

```
=>ip rtlist
   Destination       Source          Gateway       Intf    Mtrc
   10.0.0.0/24       10.0.0.0/24     10.0.0.140    eth0    0
   10.0.0.140/32     0.0.0.0/0       10.0.0.140    eth0    0
   127.0.0.1/32      0.0.0.0/0       127.0.0.1     loop    0
=>ip rtadd dst=10.10.0.0/24 src=10.0.0.0/24 gateway=10.0.0.140
=>ip rtlist
   Destination       Source          Gateway       Intf    Mtrc
   10.0.0.0/24       10.0.0.0/24     10.0.0.140    eth0    0
   10.10.0.0/24      10.0.0.0/24     10.0.0.140    eth0    0
   10.0.0.140/32     0.0.0.0/0       10.0.0.140    eth0    0
   127.0.0.1/32      0.0.0.0/0       127.0.0.1     loop    0
=>
```

RELATED COMMANDS:

ip rtdelete	Remove a route from the routing table.
ip rtlist	Show current routing table.

ip rtdelete

Delete a route from the SpeedTouch™ routing table.

SYNTAX:

```
ip rtdelete          dst = <ip-address>
                     [dstmsk = <ip-mask(dotted or cidr)>]
                     [src = <ip-address>]
                     [srcmsk = <ip-mask(dotted or cidr)>]
                     [gateway = <ip-address>]
                     [intf = <interface name>]
```

dst	The destination IP address(es) of the route. Supports cidr notation.	REQUIRED
[dstmsk]	The destination IP address mask.	OPTIONAL
[src]	The source IP address(es) of the route. Supports cidr notation.	OPTIONAL
[srcmsk]	The source IP address mask.	OPTIONAL
[gateway]	The IP address of the next hop. Must be directly connected. The parameters 'gateway' and 'intf' are mutually exclusive.	OPTIONAL
[intf]	Only for special interface routes : the outgoing IP interface name. The parameters 'gateway' and 'intf' are mutually exclusive.	OPTIONAL

EXAMPLE:

```
=>ip rtlist
   Destination       Source          Gateway       Intf    Mtrc
   10.0.0.0/24       10.0.0.0/24     10.0.0.140    eth0    0
   10.10.0.0/24      10.0.0.0/24     10.0.0.140    eth0    0
   10.0.0.140/32     0.0.0.0/0       10.0.0.140    eth0    0
   127.0.0.1/32      0.0.0.0/0       127.0.0.1     loop    0
=>ip rtdelete dst=10.10.0.0/24 src=10.0.0.0/24 gateway=10.0.0.140
=>ip rtlist
   Destination       Source          Gateway       Intf    Mtrc
   10.0.0.0/24       10.0.0.0/24     10.0.0.140    eth0    0
   10.0.0.140/32     0.0.0.0/0       10.0.0.140    eth0    0
   127.0.0.1/32      0.0.0.0/0       127.0.0.1     loop    0
=>
```

RELATED COMMANDS:

ip rtadd	Add a route to the routing table.
ip rtlist	Show current routing table.

ip rtlist

Show current SpeedTouch™ routing table.

SYNTAX:

```
ip rtlist
```

EXAMPLE OUTPUT:

```
=>ip rtlist
   Destination       Source          Gateway         Intf    Mtrc
   10.0.0.0/24       10.0.0.0/24     10.0.0.140      eth0    0
   172.16.0.5/32     0.0.0.0/0       172.16.0.5      cip1    0
   0.0.0.140/32      0.0.0.0/0       10.0.0.140      eth0    0
   127.0.0.1/32      0.0.0.0/0       127.0.0.1       loop    0
   10.0.0.0/24       0.0.0.0/0       10.0.0.140      eth0    0
   172.16.0.0/24     0.0.0.0/0       172.16.0.5      cip1    1
=>
```

RELATED COMMANDS:

ip rtadd	Add a route to the routing table.
ip rtdelete	Remove a route from the routing table.

ip sendto

Send UDP packets.

SYNTAX:

```
ip sendto              addr = <ip-address>
                       [count = <number{1-1000000}>]
                       [size = <number{1-20000}>]
                       [interval = <number{100-1000000}>]
                       [listen = <{off|on}>]
                       [srcport = <number{1-65535}>]
                       dstport = <number{1-65535}>
```

addr	The destination IP address.	REQUIRED
[count]	A number between 1 and 1000000. Represents the number of UDP packets to send.	OPTIONAL
[size]	A number between 1 and 20000 (bytes). Represents the size of the ping packet(s).	OPTIONAL
[interval]	A number between 100 and 10000000 (milliseconds). Represents the intermediate interval between two sent UDP packets.	OPTIONAL
[listen]	Listen for incoming UDP packets (on) or only send UDP packets (off).	OPTIONAL
[srcport]	The UDP source port number to use.	OPTIONAL
dstport	The UDP destination port number to send to.	REQUIRED

EXAMPLE:

```
=>ip sendto addr=10.0.0.148 listen=on srcport=19 dstport=1025
=>ip sendto addr=10.0.0.148 listen=on srcport=19 dstport=1025
1 bytes from 10.0.0.148:1025
41                                              A
=>ip sendto addr=10.0.0.148 count=3 listen=on srcport=19 dstport=1025
1 bytes from 10.0.0.148:1025
41                                              A
1 bytes from 10.0.0.148:1025
41                                              A
1 bytes from 10.0.0.148:1025
41                                              A
=>
```

RELATED COMMANDS:

ip ping	Send ICMP ECHO_REQUEST packets.

IPoA Commands

Contents

This chapter covers the following commands:

ipoa flush

Flush complete Routed IPoA configuration.

Note The flush command does not impact previously saved configurations.

SYNTAX:

```
ipoa flush
```

RELATED COMMANDS:

ipoa ifadd	Create a new Routed IPoA interface.
ipoa ifconfig	Configure a Routed IPoA interface.
ipoa ifdelete	Delete a Routed IPoA interface.
ipoa ifdetach	Detach a Routed IPoA interface.
ipoa iflist	Show current Routed IPoA interfaces.

ipoa ifadd

Create a new Routed IPoA interface.

SYNTAX:

```
ipoa ifadd        [intf = <string>]
                  [dest = <phonebook entry>]
```

[intf]	The name for the new Routed IPoA interface. If not specified, the destination parameter must be specified. In this case the name of the destination will double as interface name.	OPTIONAL
[dest]	The destination for the new Routed IPoA interface. Typically, a phonebook entry.	OPTIONAL

EXAMPLE:

```
=>ipoa iflist
IPoA_1     :   dest : IPoA_1
               Retry : 10    QoS : default    Encaps : llc/snap    Fcs : off
               Connection State : connected

=>phonebook list
Name      Type    Use  Address
IPoA_1    ipoa    1    8.35
IPoA_2    ipoa    0    8.36
=>ipoa ifadd dest=IPoA_2
=>ipoa iflist
IPoA_1     :   dest : IPoA_1
               Retry : 10    QoS : default    Encaps : llc/snap    Fcs : off
               Connection State : connected

IPoA_2     :   dest : IPoA_2
               Retry : 10    QoS : default    Encaps : llc/snap    Fcs : off
               Connection State : not-connected

=>
```

RELATED COMMANDS:

ipoa ifattach	Attach an Routed IPoA interface.
ipoa ifconfig	Configure a Routed IPoA interface.
ipoa ifdelete	Delete a Routed IPoA interface.
ipoa ifdetach	Detach a Routed IPoA interface.
ipoa iflist	Show current Routed IPoA interfaces.

ipoa ifattach

Attach (i.e. connect) an Routed IPoA interface.

SYNTAX:

```
ipoa ifattach     intf = <intfname>
```

intf	The name of the Routed IPoA interface to attach.	REQUIRED

EXAMPLE:

```
=>ipoa iflist
IPoA_PVC1   :   dest : Br4
                Retry : 10    QoS : default    Encaps : llc/snap    Fcs : off
                Connection State : connected

Br3      :   dest : Br3
                Retry : 10    QoS : default    Encaps : llc/snap    Fcs : off
                Connection State : not-connected

=>ipoa ifattach intf=Br3
IPoA_PVC1   :   dest : Br4
                Retry : 10    QoS : default    Encaps : llc/snap    Fcs : off
                Connection State : connected

Br3      :   dest : Br3
                Retry : 10    QoS : default    Encaps : llc/snap    Fcs : off
                Connection State : connected

=>
```

RELATED COMMANDS:

ipoa ifadd	Create a new Routed IPoA interface.
ipoa ifconfig	Configure a Routed IPoA interface.
ipoa ifdelete	Delete a Routed IPoA interface.
ipoa ifdetach	Detach a Routed IPoA interface.
ipoa iflist	Show current Routed IPoA interfaces.

ipoa ifconfig

Configure an IPoA interface.

SYNTAX:

```
ipoa ifconfig     intf = <IPoA_PVC>
                  [dest = <intfname>]
                  [qos = <string>]
                  [encaps = <{llc/snap|vcmux}>]
                  [retry = <number {0-65535}>]
```

intf	The name of the Routed IPoA interface to configure.	REQUIRED
[dest]	The destination for this interface. Typically a phonebook entry. This parameter needs only to be specified in case of an interface created without specified destination.	OPTIONAL
[qos]	The name of a configured Quality Of Service book entry. If not specified, the default Quality Of Service book entry will be used.	OPTIONAL
[encaps]	The type of encapsulation to be used for this bridge interface. Choose between: • llc/snap • vcmux	OPTIONAL
[retry]	A number between 0 and 65535. Represents the number of Wide Area Network (WAN) connection setup retries before giving up. By default the retry value is 10.	OPTIONAL

EXAMPLE:

```
=>ipoa iflist
IPoA_PVC1   :   dest : Br4
                Retry : 10    QoS : default    Encaps : llc/snap    Fcs : off
                Connection State : connected

=>ipoa ifconfig intf=IPoA_PVC1 encaps=llc/snap retry=15
=>ipoa iflist
Moreipoa    :   dest : Br4
                Retry : 15    QoS : default    Encaps : llc/snap    Fcs : off
                Connection State : connected
                RX bytes: 0      frames: 0
                TX bytes: 0      frames: 0      dropframes: 0
=>
```

RELATED COMMANDS:

ipoa ifadd	Create a new Routed IPoA interface.
ipoa ifattach	Attach a Routed IPoA interface.
ipoa ifdelete	Delete a Routed IPoA interface.
ipoa ifdetach	Detach a Routed IPoA interface.
ipoa iflist	Show current Routed IPoA interfaces.

ipoa ifdelete

Delete a Routed IPoA interface.

SYNTAX:

```
ipoa ifdelete     intf = <intfname>
```

intf	The name of the Routed IPoA interface.	REQUIRED

EXAMPLE:

```
=>ipoa iflist
Newipoa     :   dest : Br3
                Retry : 10    QoS : default    Encaps : llc/snap    Fcs : off
                Connection State : connected
                RX bytes: 0      frames: 0
                TX bytes: 0      frames: 0     dropframes: 0
Moreipoa        dest : Br4
                Retry : 10    QoS : default    Encaps : llc/snap    Fcs : off
                Connection State : not-connected
=>ipoa ifdelete intf=Moreipoa
=>ipoa iflist
Newipoa     :   dest : Br3
                Retry : 10    QoS : default    Encaps : llc/snap    Fcs : off
                Connection State : connected
                RX bytes: 0      frames: 0
                TX bytes: 0      frames: 0     dropframes: 0
=>
```

RELATED COMMANDS:

ipoa ifadd	Create a new Routed IPoA interface.
ipoa ifattach	Attach a Routed IPoA interface.
ipoa ifconfig	Configure a Routed IPoA interface.
ipoa ifdetach	Detach a Routed IPoA interface.
ipoa iflist	Show current Routed IPoA interfaces.

ipoa ifdetach

Detach a Routed IPoA interface.

SYNTAX:

```
ipoa ifdetach   intf = <intfname>
```

intf	The name of the Routed IPoA interface.	REQUIRED

EXAMPLE:

```
=>ipoa iflist
Newipoa     :   dest : Br3
                Retry : 10    QoS : default    Encaps : llc/snap    Fcs : off
                Connection State : connected
                RX bytes: 0      frames: 0
                TX bytes: 0      frames: 0      dropframes: 0
Moreipoa        dest : Br4
                Retry : 10    QoS : default    Encaps : llc/snap    Fcs : off
                Connection State : connected
                RX bytes: 0      frames: 0
                TX bytes: 0      frames: 0      dropframes: 0
=>ipoa ifdetach intf=Moreipoa
=>ipoa iflist
Newipoa     :   dest : Br3
                Retry : 10    QoS : default    Encaps : llc/snap    Fcs : off
                Connection State : connected
                RX bytes: 0      frames: 0
                TX bytes: 0      frames: 0      dropframes: 0
Moreipoa        dest : Br4
                Retry : 10    QoS : default    Encaps : llc/snap    Fcs : off
                Connection State : not-connected
=>
```

RELATED COMMANDS:

ipoa ifadd	Create a new Routed IPoA interface.
ipoa ifattach	Attach a Routed IPoA interface.
ipoa ifconfig	Configure a Routed IPoA interface.
ipoa ifdelete	Delete a Routed IPoA interface.
ipoa iflist	Show current Routed IPoA interfaces.

ipoa iflist

Show all or a specified Routed IPoA interface(s).

SYNTAX:

```
ipoa iflist       [intf = <intfname>]
```

[intf]	The name of the Routed IPoA interface. If not specified all Routed IPoA interfaces are listed.	OPTIONAL

EXAMPLE OUTPUT:

```
=>ipoa iflist
Newipoa     :   dest : Br3
                Retry : 10    QoS : default    Encaps : llc/snap    Fcs : off
                Connection State : connected
                RX bytes: 0      frames: 0
                TX bytes: 0      frames: 0      dropframes: 0
Moreipoa        dest : Br4
                Retry : 10    QoS : default    Encaps : llc/snap    Fcs : off
                Connection State : connected
                RX bytes: 0      frames: 0
                TX bytes: 0      frames: 0      dropframes: 0
=>
```

RELATED COMMANDS:

ipoa ifadd	Create a new Routed IPoA interface.
ipoa ifattach	Attach a Routed IPoA interface.
ipoa ifconfig	Configure a Routed IPoA interface.
ipoa ifdelete	Delete a Routed IPoA interface.
ipoa detach	Detach a Routed IPoA interface.

MER Commands

Contents

This chapter covers the following topics:

mer flush

Flush complete Routed Ethernet, often referred to as MAC Encapsulated Routing (MER), configuration.

Note The flush command does not impact previously saved configurations.

SYNTAX:

```
mer flush
```

EXAMPLE:

```
=>mer iflist
NewMer     : dest : Br3
             Retry : 10     QoS : default     Encaps : llc/snap     Fcs : off
             Connection State : connected
             RX bytes: 0    frames: 0
             TX bytes: 0    frames: 0         dropframes: 0
=>mer flush
=>mer iflist
=>
```

mer ifadd

Create a new Routed Ethernet interface.

SYNTAX:

```
mer ifadd          [intf = <string>]
                   [dest = <phonebook entry>]
```

[intf]	The name for the new Routed Ethernet interface. **Note** If not specified, the destination parameter must be specified. In this case the name of the destination will double as interface name.	OPTIONAL
[dest]	The destination for the new Routed Ethernet interface. Typically, an phonebook entry.	OPTIONAL

EXAMPLE:

```
=>mer iflist
NewMer     : dest : Br3
             Retry : 10     QoS : default     Encaps : llc/snap     Fcs : off
             Connection State : connected
             RX bytes: 0    frames: 0
             TX bytes: 0    frames: 0         dropframes: 0
=>phonebook list
Name         Type       Use     Address
Br1          bridge     1       8.35
Br2          bridge     1       8.36
CIPPVC3      cip        1       8.82
CIPPVC4      cip        1       8.83
=>mer ifadd intf=MoreMer dest=Br4
=>mer iflist
NewMer     : dest : Br3
             Retry : 10     QoS : default     Encaps : llc/snap     Fcs : off
             Connection State : connected
             RX bytes: 0    frames: 0
             TX bytes: 0    frames: 0         dropframes: 0
MoreMer    : dest : Br4
             Retry : 10     QoS : default     Encaps : llc/snap     Fcs : off
             Connection State : not-connected
=>
```

RELATED COMMANDS:

mer ifattach	Attach a Routed Ethernet interface.
mer ifconfig	Configure a Routed Ethernet interface.
mer ifdelete	Delete a Routed Ethernet interface.
mer ifdetach	Detach a Routed Ethernet interface.
mer iflist	Show current Routed Ethernet interfaces.

mer ifattach

Attach (i.e. connect) a Routed Ethernet interface.

SYNTAX:

```
mer ifattach      intf = <intfname>
```

intf	The name of the Routed Ethernet interface to attach.	REQUIRED

EXAMPLE:

```
=>mer iflist
NewMer     : dest : Br3
             Retry : 10     QoS : default     Encaps : llc/snap     Fcs : off
             Connection State : connected
             RX bytes: 0    frames: 0
             TX bytes: 0    frames: 0         dropframes: 0
MoreMer    : dest : Br4
             Retry : 10     QoS : default     Encaps : llc/snap     Fcs : off
             Connection State : not-connected
=>mer ifattach intf=MoreMer
=>mer iflist
NewMer     : dest : Br3
             Retry : 10     QoS : default     Encaps : llc/snap     Fcs : off
             Connection State : connected
             RX bytes: 0     frames: 0
             TX bytes: 0     frames: 0        dropframes: 0
MoreMer    : dest : Br4
             Retry : 10      QoS : default    Encaps : llc/snap     Fcs : off
             Connection State : connected
             RX bytes: 0     frames: 0
             TX bytes: 0     frames: 0        dropframes: 0
=>
```

RELATED COMMANDS:

mer ifadd	Create a new Routed Ethernet interface.
mer ifconfig	Configure a Routed Ethernet interface.
mer ifdelete	Delete a Routed Ethernet interface.
mer ifdetach	Detach a Routed Ethernet interface.
mer iflist	Show current Routed Ethernet interfaces.

mer ifconfig

Configure a Routed Ethernet interface.

SYNTAX:

```
mer ifconfig      intf = <intfname>
                  [dest = <intfname>]
                  [qos = <string>]
                  [encaps = <{llc/snap|vcmux}>]
                  [retry = <number {0-65535}>]
```

intf	The name of the Routed Ethernet interface to configure.	REQUIRED
[dest]	The destination for this interface. Typically a phonebook entry. This parameter needs only to be specified in case of an interface created without specified destination.	OPTIONAL
[qos]	The name of a configured Quality Of Service book entry. This parameter never needs to be specified.	OPTIONAL
[encaps]	The type of encapsulation to be used for this bridge interface. Choose between: • llc/snap • vcmux	OPTIONAL
[retry]	A number between 0 and 65535. Represents the number of WAN connection setup retries before giving up. By default the retry value is 10.	OPTIONAL

EXAMPLE:

```
=>mer iflist
MoreMer    : dest : Br4
             Retry : 10     QoS : default     Encaps : vcmux     Fcs : off
             Connection State : connected
             RX bytes: 0    frames: 0
             TX bytes: 0    frames: 0         dropframes: 0
=>mer ifconfig intf=MoreMer encaps=llc/snap retry=15
=>mer iflist
MoreMer    : dest : Br4
             Retry : 15     QoS : default     Encaps : llc/snap  Fcs : off
             Connection State : connected
             RX bytes: 0    frames: 0
             TX bytes: 0    frames: 0         dropframes: 0
=>
```

RELATED COMMANDS:

mer ifadd	Create a new Routed Ethernet interface.
mer ifattach	Attach a Routed Ethernet interface.
mer ifdelete	Delete a Routed Ethernet interface.
mer ifdetach	Detach a Routed Ethernet interface.
mer iflist	Show current Routed Ethernet interfaces.

mer ifdelete

Delete a Routed Ethernet interface.

SYNTAX:

```
mer ifdelete      intf = <intfname>
```

intf	The name of the Routed Ethernet interface.	REQUIRED

EXAMPLE:

```
=>mer iflist
NewMer     : dest : Br3
             Retry : 10     QoS : default     Encaps : llc/snap     Fcs : off
             Connection State : connected
             RX bytes: 0    frames: 0
             TX bytes: 0    frames: 0         dropframes: 0
MoreMer    : dest : Br4
             Retry : 10     QoS : default     Encaps : llc/snap     Fcs : off
             Connection State : not-connected
=>mer ifdelete intf=MoreMer
=>mer iflist
NewMer     : dest : Br3
             Retry : 10     QoS : default     Encaps : llc/snap     Fcs : off
             Connection State : connected
             RX bytes: 0    frames: 0
             TX bytes: 0    frames: 0         dropframes: 0
=>
```

RELATED COMMANDS:

mer ifadd	Create a new Routed Ethernet interface.
mer ifattach	Attach a Routed Ethernet interface.
mer ifconfig	Configure a Routed Ethernet interface.
mer ifdetach	Detach a Routed Ethernet interface.
mer iflist	Show current Routed Ethernet interfaces.

mer ifdetach

Detach a Routed Ethernet interface.

SYNTAX:

```
mer ifdetach    intf = <intfname>
```

intf	The name of the Routed Ethernet interface.	REQUIRED

EXAMPLE:

```
=>mer iflist
NewMer     : dest : Br3
             Retry : 10     QoS : default    Encaps : llc/snap     Fcs : off
             Connection State : connected
             RX bytes: 0    frames: 0
             TX bytes: 0    frames: 0         dropframes: 0
MoreMer    : dest : Br4
             Retry : 10      QoS : default    Encaps : llc/snap     Fcs : off
             Connection State : connected
             RX bytes: 0     frames: 0
             TX bytes: 0     frames: 0         dropframes: 0
=>mer ifdetach intf=MoreMer
=>mer iflist
NewMer     : dest : Br3
             Retry : 10     QoS : default     Encaps : llc/snap     Fcs : off
             Connection State : connected
             RX bytes: 0    frames: 0
             TX bytes: 0    frames: 0         dropframes: 0
MoreMer    : dest : Br4
             Retry : 10     QoS : default     Encaps : llc/snap     Fcs : off
             Connection State : not-connected
=>
```

RELATED COMMANDS:

mer ifadd	Create a new Routed Ethernet interface.
mer ifattach	Attach a Routed Ethernet interface.
mer ifconfig	Configure a Routed Ethernet interface.
mer ifdelete	Delete a Routed Ethernet interface.
mer iflist	Show current Routed Ethernet interfaces.

mer iflist

Show all or a specified Routed Ethernet interface(s).

SYNTAX:

```
mer iflist        [intf = <intfname>]
```

[intf]	The name of the Routed Ethernet interface.	OPTIONAL
	Note If not specified, all Routed Ethernet interfaces are listed.	

EXAMPLE OUTPUT:

```
=>mer iflist
NewMer     : dest : Br3
             Retry : 10     QoS : default     Encaps : llc/snap    Fcs : off
             Connection State : connected
             RX bytes: 0    frames: 0
             TX bytes: 0    frames: 0         dropframes: 0
MoreMer    : dest : Br4
             Retry : 10     QoS : default     Encaps : llc/snap     Fcs : off
             Connection State : connected
             RX bytes: 0    frames: 0
             TX bytes: 0    frames: 0         dropframes: 0
=>
```

RELATED COMMANDS:

mer ifadd	Create a new Routed Ethernet interface.
mer ifattach	Attach a Routed Ethernet interface.
mer ifconfig	Configure a Routed Ethernet interface.
mer ifdelete	Delete a Routed Ethernet interface.
mer detach	Detach a Routed Ethernet interface.

NAT Commands

Contents

This chapter covers the following commands:

nat applist

List available Network Address and Port Translation (NAPT) protocol helpers.

Certain protocols are 'sensitive' to NAPT in that they do not function properly when dealing with it. This list shows which 'NAPT-sensitive' applications are supported on the SpeedTouch™, i.e. the inherent knowledge of the SpeedTouch™ on this matter.

SYNTAX:

```
nat applist
```

EXAMPLE OUTPUT:

```
=>nat applist
Application  Proto   DefaultPort
GRE          gre        1   INCOMING
PPTP         tcp     1723   OUTGOING  INCOMING
ESP          50         1   OUTGOING INCOMING
IKE          udp      500   OUTGOING INCOMING
SIP          udp     5060   OUTGOING INCOMING
ILS          tcp        0   OUTGOING
H245         tcp        0   OUTGOING INCOMING
H323         tcp     1720   OUTGOING INCOMING
RAUDIO(PNA)  tcp     7070   OUTGOING
RTSP         tcp      554   OUTGOING
IRC          tcp     6667   OUTGOING
FTP          tcp       21   OUTGOING
=>
```

RELATED COMMANDS:

nat bind	Create a new helper/port binding.
nat bindlist	List current NAPT helper/port bindings.
nat unbind	Delete an existing helper/port binding.

nat bind

Create a new helper/port binding.

SYNTAX:

`nat bind`	`application = <string>` `port = <{<supported TCP/UDP port name>|<number>}>`

application	The name of a NAPT application helper. The name must be spelled exactly as listed in the application list (:nat applist).	REQUIRED
port	The TCP/UDP port this application handler should work on. Select one of the supported TCP/UDP port names (See " Supported TCP/UDP Port Names" on page 290 for a listing of TCP/UDP port names supported by the SpeedTouch™). Alternatively, specify the protocol number.	REQUIRED

EXAMPLE:

```
=>nat bindlist
Application  Proto Port
SIP          udp   5060
GRE          gre   1
PPTP         tcp   1723
ILS          tcp   1002
ILS          tcp   389
H323         tcp   1720
FTP          tcp   21
IRC          tcp   6667
RAUDIO(PNA)  tcp   7070
=>nat bind application=RAUDIO(PNA) port=7071
=>nat bindlist
Application  Proto Port
SIP          udp   5060
GRE          gre   1
PPTP         tcp   1723
ILS          tcp   1002
ILS          tcp   389
H323         tcp   1720
FTP          tcp   21
IRC          tcp   6667
RAUDIO(PNA)  tcp   7070
RAUDIO(PNA)  tcp   7071
=>
```

RELATED COMMANDS:

nat applist	List available NAPT protocol helpers.
nat bindlist	List current NAPT helper/port bindings.
nat unbind	Delete an existing helper/port binding.

nat bindlist

List current NAPT helper/port bindings.

SYNTAX:

```
nat bindlist
```

EXAMPLE OUTPUT:

```
=>nat bindlist
Application  Proto Port
SIP          udp   5060
GRE          gre   1
PPTP         tcp   1723
ILS          tcp   1002
ILS          tcp   389
H323         tcp   1720
FTP          tcp   21
RTSP         tcp   554
IRC          tcp   6667
RAUDIO(PNA)  tcp   7070
=>
```

RELATED COMMANDS:

nat applist	List available NAPT protocol helpers.
nat bind	Create a new NAPT helper/port binding.
nat unbind	Delete an existing helper/port binding.

nat create

Create a static NAPT entry. Typically used to install specific servers behind the SpeedTouch™ NAPT device.

SYNTAX:

```
nat create          protocol = <{<supported IP protocol name>|<number>}>
                    inside_addr = <ip-address>
                    [inside_port = <{<supported TCP/
                    UDP port name>|<number>}>]
                    outside_addr = <ip-address>
                    [outside_port = <{<supported TCP/
                    UDP port name>|<number>}>]
                    [foreign_addr = <ip-address>]
                    [foreign_port = <{<supported TCP/
                    UDP port name>|<number>}>]
```

protocol	The IP protocol name (or number) of the incoming stream. Select one of the supported protocol names (See " Supported Internet Protocol (IP) Protocol Names" on page 289 for a listing of protocol names supported by the SpeedTouch™). Alternatively, specify the protocol number.	REQUIRED
inside_addr	The IP address of the local host (intended to receive the incoming traffic) behind the SpeedTouch™ 's NAPT device. Typically, a private IP address.	REQUIRED
[inside_port]	The port of the application on the local host. Select one of the supported TCP/UDP port names (See " Supported TCP/UDP Port Names" on page 290 for a listing of TCP/UDP port names supported by the SpeedTouch™). Alternatively, specify the protocol number. Applicable for TCP and UDP protocols only. Other protocols do not need a port to be specified.	OPTIONAL
outside_addr	The apparent host IP address this application is running on, i.e. the NAPT enabled WAN IP address of the SpeedTouch™ . Use '0' to create a template. Such template will then be valid for any of SpeedTouch™ 's NAPT enabled IP addresses, e.g. also dynamically assigned/negotiated IP addresses.	REQUIRED
[outside_port]	The apparent port number this application is running on. Select one of the supported TCP/UDP port names (See " Supported TCP/UDP Port Names" on page 290 for a listing of TCP/UDP port names supported by the SpeedTouch™). Alternatively, specify the protocol number. Applicable for TCP and UDP protocols only. Other protocols do not need a port to be specified.	OPTIONAL
[foreign_addr]	The IP address of the in-front-of-NAPT routable address. Use '0' to match all foreign addresses.	REQUIRED

[foreign_port]	The port of the routable host. Select one of the supported TCP/UDP port names (See " Supported TCP/UDP Port Names" on page 290 for a listing of TCP/UDP port names supported by the SpeedTouch™). Alternatively, specify the protocol number. Do not use '0' in case a foreign IP address is specified. Applicable for TCP and UDP protocols only. Other protocols do not need a port to be specified.	OPTIONAL

nat defserver

Define the default server behind the SpeedTouch™ NAPT device that receives all (unknown) incoming packets.

In typical LAN configurations one local 'default' server will be responsible for all WAN-LAN mail, http, ftp, ... connectivity. This command allows to specify this server. For specific services, use :nat create.

SYNTAX:

`nat defserver`	`[addr = <ip-address>]`

[addr]	The IP address of the server (on the 'inside') that will receive all (unknown) incoming packets. If not specified the current default server is shown.	OPTIONAL

EXAMPLE INPUT/OUTPUT:

```
=>nat defserver
Default server is undefined
=>nat defserver addr=10.0.0.1
=>nat defserver
Default server is 10.0.0.1
=>
```

nat delete

Delete a static NAPT entry.

SYNTAX:

`nat delete`	`protocol = <{<supported IP protocol name>|<number>}>`
	`inside_addr = <ip-address>`
	`[inside_port = <{<supported TCP/UDP port name>|<number>}>]`
	`outside_addr = <ip-address>`
	`[outside_port = <{<supported TCP/UDP port name>|<number>}>]`
	`[foreign_addr = <ip-address>]`
	`[foreign_port = <{<supported TCP/UDP port name>|<number>}>]`

protocol	The IP protocol name (or number) of the incoming stream. Select one of the supported protocol names (See " Supported Internet Protocol (IP) Protocol Names" on page 289 for a listing of protocol names supported by the SpeedTouch™). Alternatively, specify the protocol number.	REQUIRED
inside_addr	The IP address of the local host (intended to receive the incoming traffic) behind the SpeedTouch™ 's NAPT device. Typically, a private IP address.	REQUIRED
[inside_port]	The port of the application on the local host. Select one of the supported TCP/UDP port names (See " Supported TCP/UDP Port Names" on page 290 for a listing of TCP/UDP port names supported by the SpeedTouch™). Alternatively, specify the protocol number. Applicable for TCP and UDP protocols only. Other protocols do not need a port to be specified.	OPTIONAL
outside_addr	The apparent host IP address this application is running on, i.e. the NAPT enabled WAN IP address of the SpeedTouch™. Use '0' to delete an entry valid for any of SpeedTouch™ 's NAPT enabled IP addresses, e.g. also dynamically assigned/negotiated IP addresses.	REQUIRED
[outside_port]	The apparent port number this application is running on. Select one of the supported TCP/UDP port names (See " Supported TCP/UDP Port Names" on page 290 for a listing of TCP/UDP port names supported by the SpeedTouch™). Alternatively, specify the protocol number. Applicable for TCP and UDP protocols only. Other protocols do not need a port to be specified.	OPTIONAL
[foreign_addr]	The IP address of the in-front-of-NAPT routable address.	REQUIRED

[foreign_port]	The port of the routable host. Select one of the supported TCP/UDP port names (See " Supported TCP/UDP Port Names" on page 290 for a listing of TCP/UDP port names supported by the SpeedTouch™). Alternatively, specify the protocol number. Applicable for TCP and UDP protocols only. Other protocols do not need a port to be specified.	OPTIONAL

nat disable

Disable NAPT on a SpeedTouch™ IP address.

SYNTAX:

```
nat disable       addr = <ip-address>
```

addr	The SpeedTouch™ IP address for which NAPT must be disabled.	REQUIRED

EXAMPLE:

```
=>nat list
Indx Prot Inside-addr:Port Outside-addr:Port Foreign-addr:Port Flgs Expir State  Control
1    6    10.0.0.138:80    172.16.0.5:1080   0.0.0.0:0         19   8     9
2    17   10.0.0.138:138   10.0.0.140:138    10.0.0.20:138     11   20    10
3    17   10.0.0.138:137   10.0.0.140:137    10.0.0.254:137    11   20    10
4    17   10.0.0.138:7938  10.0.0.140:7938   10.0.0.96:4756    11   20    10
5    17   10.0.0.138:513   10.0.0.140:513    10.0.0.109:513    11   20    10
6    17   10.0.0.138:111   10.0.0.140:111    10.0.0.96:4756    11   20    10
=>nat disable addr 172.16.0.5
=>nat list
Indx Prot Inside-addr:Port Outside-addr:Port Foreign-addr:Port Flgs Expir State Control
1    17   10.0.0.138:138   10.0.0.140:138    10.0.0.20:138     11   20    10
2    17   10.0.0.138:137   10.0.0.140:137    10.0.0.254:137    11   20    10
3    17   10.0.0.138:7938  10.0.0.140:7938   10.0.0.96:4756    11   20    10
4    17   10.0.0.138:513   10.0.0.140:513    10.0.0.109:513    11   20    10
5    17   10.0.0.138:111   10.0.0.140:111    10.0.0.96:4756    11   20    10
=>
```

RELATED COMMANDS:

nat create	Create a static NAPT entry.
nat delete	Delete a static NAPT entry.
nat enable	Enable NAPT on one of the SpeedTouch™ IP addresses.
nat list	List NAPT connection database.

nat enable

Enable NAPT on a SpeedTouch™ IP address.

SYNTAX:

```
nat enable        addr = <ip-address>
                  [type = <{none|pat}>]
```

addr	The SpeedTouch™ IP address for which NAPT must be enabled.	REQUIRED
[type]	Enable port translation (pat) or not (none).	OPTIONAL

EXAMPLE:

```
=>ip aplist
1 eth0  Type:Ethernet  HWaddr  00:80:9f:24:ab:cf  BRHWaddr  ff:ff:ff:ff:ff:ff
  inet   addr:10.10.10.147    Bcast:10.10.10.255     Mask:255.0.0.0
  UP RUNNING           MTU:1500    ReasmMAX:65535    Group:2
  IPRX   bytes:19791886  unicastpkts:11341  brcastpkts:290555
  IPTX   bytes:839550    unicastpkts:11477  brcastpkts:0     droppkts:0
  HWRX   bytes:0         unicastpkts:0      brcastpkts:0
  HWTX   bytes:0         unicastpkts:0      brcastpkts:0     droppkts:0
0 loop  Type:0
  inet   addr:127.0.0.1       Bcast:127.255.255.255  Mask:255.0.0.0
  UP RUNNING           MTU:1500    ReasmMAX:65535    Group:1
  IPRX   bytes:116       unicastpkts:0      brcastpkts:2
  IPTX   bytes:0         unicastpkts:0      brcastpkts:0     droppkts:0
  HWRX   bytes:0         unicastpkts:0      brcastpkts:0
  HWTX   bytes:0         unicastpkts:0      brcastpkts:0     droppkts:0
=>nat enable addr=10.10.10.147 type=pat
=>ip aplist
1 eth0  Type:Ethernet  HWaddr  00:80:9f:24:ab:cf  BRHWaddr  ff:ff:ff:ff:ff:ff
  inet   addr:10.10.10.147    Bcast:10.10.10.255     Mask:255.0.0.0
  UP RUNNING    pat    MTU:1500    ReasmMAX:65535    Group:2
  IPRX   bytes:19791886  unicastpkts:11341  brcastpkts:290555
  IPTX   bytes:839550    unicastpkts:11477  brcastpkts:0     droppkts:0
  HWRX   bytes:0         unicastpkts:0      brcastpkts:0
  HWTX   bytes:0         unicastpkts:0      brcastpkts:0     droppkts:0
0 loop  Type:0
  inet   addr:127.0.0.1       Bcast:127.255.255.255  Mask:255.0.0.0
  UP RUNNING           MTU:1500    ReasmMAX:65535    Group:1
  IPRX   bytes:116       unicastpkts:0      brcastpkts:2
  IPTX   bytes:0         unicastpkts:0      brcastpkts:0     droppkts:0
  HWRX   bytes:0         unicastpkts:0      brcastpkts:0
  HWTX   bytes:0         unicastpkts:0      brcastpkts:0     droppkts:0
=>
```

RELATED COMMANDS:

nat create	Create a static NAPT entry.
nat delete	Delete a static NAPT entry.
nat disable	Disable NAPT on one of the SpeedTouch™ IP addresses.
nat list	List NAPT connection database.

nat flush

Flush complete NAPT configuration.

Note The flush command does not impact previously saved configurations.

SYNTAX:

```
nat flush
```

RELATED COMMANDS:

nat create	Create a static NAPT entry.
nat delete	Delete a static NAPT entry.
nat disable	Disable NAPT on one of the SpeedTouch™ IP addresses.
nat enable	Enable NAPT on one of the SpeedTouch™ IP addresses.

nat list

Show NAPT connection database.

SYNTAX:

`nat list`	`[addr = <ip-address>]`

[addr]	The SpeedTouch™ IP address for which the NAPT connection database must be shown. **Note** If the parameter is not specified, the NAPT connection database for all IP addresses is shown.	OPTIONAL

EXAMPLE INPUT/OUTPUT:

```
=>nat list
Indx Prot Inside-addr:Port Outside-addr:Port Foreign-addr:Port Flgs Expir State Control
1    6    10.0.0.138:80    172.16.0.5:1080   0.0.0.0:0         19   8     9
2    17   10.0.0.138:135   10.0.0.140:135    10.0.0.155:1034   11   20    10
3    17   10.0.0.138:138   10.0.0.140:138    10.0.0.20:138     11   20    10
4    17   10.0.0.138:137   10.0.0.140:137    10.0.0.254:137    11   20    10
5    17   10.0.0.138:7938  10.0.0.140:7938   10.0.0.96:4756    11   20    10
6    17   10.0.0.138:513   10.0.0.140:513    10.0.0.109:513    11   20    10
7    17   10.0.0.138:111   10.0.0.140:111    10.0.0.96:4756    11   20    10
=>
```

RELATED COMMANDS:

nat create	Create a static NAPT entry.
nat delete	Delete a static NAPT entry.
nat disable	Disable NAPT on one of the SpeedTouch™ IP addresses.
nat enable	Enable NAPT on one of the SpeedTouch™ IP addresses.

nat unbind

Delete an existing helper/port binding.

SYNTAX:

`nat unbind`	`application = <string>` `port = <{<supported TCP/UDP port name>|<number>}>`

application	The name of a NAPT application helper. The name must be spelled exactly as listed in the application list (:nat applist).	REQUIRED
port	The TCP/UDP port this application handler is working on. Select one of the supported TCP/UDP port names (See " Supported TCP/UDP Port Names" on page 290 for a listing of TCP/UDP port names supported by the SpeedTouch™). Alternatively, specify the protocol number.	REQUIRED

EXAMPLE:

```
=>nat applist
Application  Proto   DefaultPort
ils          tcp     0     OUTGOING
H254         tcp     0     OUTGOING    INCOMING
H323         tcp     1720  OUTGOING    INCOMING
RAUDIO(PNA)  tcp     7070  OUTGOING
RTSP         tcp     554   OUTGOING
IRC          tcp     6667  OUTGOING
FTP          tcp     21    OUTGOING    INCOMING
=>nat bindlist
Application  Proto   Port
RAUDIO(PNA)  tcp     7071
H323         tcp     1720
FTP          tcp     21
RTSP         tcp     554
IRC          tcp     6667
RAUDIO(PNA)  tcp     7070
=>
=>nat unbind application=RAUDIO(PNA) port=7071
=>nat bindlist
Application  Proto   Port
H323         tcp     1720
FTP          tcp     21
RTSP         tcp     554
IRC          tcp     6667
RAUDIO(PNA)  tcp     7070
=>
```

RELATED COMMANDS:

nat applist	List available NAPT protocol helpers.
nat bindlist	List current NAPT helper/port bindings.
nat bind	Create a new helper/port binding.

Phonebook Commands

Contents

This chapter covers the following comands:

phonebook add

Add a phonebook entry.

SYNTAX:

```
phonebook add    name = <string>
                 addr = <[port.]vpi.vci>
                 type = <{any|ethoa|pppoa|ipoa}>
```

name	A free to choose phonebook name for the destination. Two limitations apply: • The name of a phonebook entry intended for the Relayed PPPoA (PPPoA-to-PPTP Relaying) packet service may not start with capital P or capital T • The name of a phonebook entry intended for the PPP-to-DHCP spoofing packet service must start with DHCP, e.g. 'DHCP_Spoof01'.	REQUIRED
addr	The ATM address for this destination. It is composed of a Virtual Path Identifier (VPI) and a Virtual Channel Identifier (VCI) identifying ATM virtual channels. In most cases the values are provided by the Service Provider. Accepted VPI: a number between 0 and 15 Accepted VCI: a number between 0 and 511.	REQUIRED
type	The Connection Service supported by the destination. Choose between: • any (All Packet Services) • ethoa (Bridged Ethernet, Routed Ethernet, Bridged PPPoE, Routed PPPoE) • pppoa (Routed PPPoA and Relayed PPPoA) • ipoa (Classical IPoA and Routed IPoA).	REQUIRED

EXAMPLE:

```
=>phonebook list
Name       Type     Use     Address
usb_port   any      1       usb.0.35
PVC1       any      1       8.35
PVC2       ethoa    0       8.36
Br4        ethoa    0       8.38
CIPPVC3    ipoa     1       8.82
=>phonebook add name=PVC_Test addr=8.68 type=pppoa
=>phonebook list
Name       Type     Use     Address
usb_port   any      1       usb.0.35
PVC1       any      1       8.35
PVC2       ethoa    0       8.36
Br4        ethoa    0       8.38
CIPPVC3    ipoa     1       8.82
PVC_Test   pppoa    0       8.68
=>
```

Note The usb_port phonebook entry is only applicable in case of a SpeedTouch™530 variant. Do not change this phonebook entry in any way, or delete it from the phonebook!

RELATED COMMANDS:

phonebook delete	Remove a phonebook entry.
phonebook list	Show current phonebook.

phonebook autolist

Show auto PVCs, if supported by the Central Office DSLAM.

SYNTAX:

```
phonebook autolist
```

EXAMPLE INPUT/OUTPUT:

```
=>phonebook autolist
8.35
=>
```

RELATED COMMANDS:

phonebook list	Show current phonebook.

phonebook delete

Remove an unused phonebook entry.

SYNTAX:

```
phonebook delete    name = <string>
```

name	The name of the phonebook entry to delete. Only applicable for phonebook entries that are not used, i.e. not configured for any packet service. Execute phonebook list to check whether the entry is used (Use=1) or not (Use=0).	REQUIRED

EXAMPLE:

```
=>phonebook list
Name       Type     Use     Address
usb_port   any      1       usb.0.35
PVC1       any      1       8.35
PVC2       ethoa    0       8.36
Br4        ethoa    0       8.38
CIPPVC3    ipoa     1       8.82
PVC_Test   pppoa    0       8.68
=>phonebook delete name=PVC_Test
=>phonebook list
Name       Type     Use     Address
usb_port   any      1       usb.0.35
PVC1       any      1       8.35
PVC2       ethoa    0       8.36
Br4        ethoa    0       8.38
CIPPVC3    ipoa     1       8.82
=>
```

Note In case of a SpeedTouch™530 variant, never delete the usb_port phonebook entry!

RELATED COMMANDS:

phonebook add	Add a phonebook entry.
phonebook list	Show current phonebook.

phonebook flush

Flush complete phonebook.

Note The flush command does not impact previously saved configurations.

SYNTAX:

```
phonebook flush
```

EXAMPLE:

```
=>phonebook list
Name       Type     Use     Address
usb_port   any      1       usb.0.35
PVC1       any      1       8.35
PVC2       ethoa    0       8.36
Br4        ethoa    0       8.38
CIPPVC3    ipoa     1       8.82
PVC_Test   pppoa    0       8.68
=>phonebook flush
=>phonebook list
Name       Type     Use     Address
=>
```

Note In case of a SpeedTouch™530 variant, flushing the Phonebook may cause the device te become unaccesible from its USB interface.

phonebook list

Show current phonebook.

SYNTAX:

```
phonebook list    [opt  = <{long}>]
```

[opt]	Select output format. For internal use only.	OPTIONAL

EXAMPLE INPUT/OUTPUT:

```
=>phonebook list
Name       Type    Use    Address
usb_port   any     1      usb.0.35
PVC1       any     1      8.35
PVC2       ethoa   0      8.36
Br4        ethoa   0      8.38
CIPPVC3    ipoa    1      8.82
PVC_Test   pppoa   0      8.68
=>
```

Note The usb_port phonebook entry is only applicable in case of a SpeedTouch™530 variant. Do not change this phonebook entry in any way, or delete it from the phonebook!

RELATED COMMANDS:

phonebook add	Add a phonebook entry.
phonebook autolist	Show auto PVCs.
phonebook delete	Remove a phonebook entry.

PPP Commands

Contents

This chapter covers the following commands:

ppp flush

Flush complete Routed PPP over ATM (PPPoA) and Routed PPP over Ethernet (PPPoE) configuration.

Note The flush command does not impact previously saved configurations.

SYNTAX:

```
ppp flush
```

EXAMPLE:

```
=>ppp iflist
PPP1: dest : PPP1
  Retry:  10  QoS  default encaps  VC-MUX
  mode = IP Routing
  flags = echo  magic  accomp  mru  addr  route  savepwd  PPPOA
  trans addr = pat    mru = 1500
  route = 0.0.0.0/0 - 0.0.0.0/0 (metric 0)
  user name = guest    password = ********
  admin state = down    oper state = down    link state = not-connected
  LCP:    state = initial    retransm = 10    term. reason =
  IPCP:   state = initial    retransm = 0     term. reason =
=>ppp flush
=>ppp iflist
=>
```

ppp ifadd

Create a new Routed PPP(oA/oE) interface.

SYNTAX:

```
ppp ifadd       [intf = <string>]
                [dest = <phonebook entry>]
                [encaps = <{vcmux|llc}>]
                [speed = <number{4800-10000000}>]
```

[intf]	The name for the new Routed PPP(oA/oE) interface. If not specified, the destination parameter must be specified. In this case the name of the destination will double as interface name.	OPTIONAL
[dest]	The destination for the new Routed PPP(oA/oE) interface. Typically, an phonebook entry.	OPTIONAL
[encaps]	The type of encapsulation to be used for this Routed PPP(oA/oE) interface. Choose between: • vcmux - Virtual Channel MUltipleXing (VCMUX) • llc/snap - Logical Link Control (LLC)/Sub Network Access Protocol (SNAP)	OPTIONAL
[speed]	A number between 4800 and 10000000 (bits per second). Represents the speed of the peer-to-peer connection. Use for backward compatibility. Use Quality Of Service instead.	OPTIONAL

EXAMPLE:

```
=>ppp iflist
PPP1: dest : PPP1
  Retry: 10    QoS    default    encaps    VC-MUX
  mode = IP Routing
  flags = echo  magic  accomp  mru  addr  route  savepwd  PPPOA
  trans addr = pat    mru = 1500
  route = 0.0.0.0/0 - 0.0.0.0/0 (metric 0)
  user name = guest    password = ********
  admin state = down    oper state = down    link state = not-connected
  LCP: state = initial    retransm = 10    term. reason =
  IPCP: state = initial    retransm = 0    term. reason =
=>ppp ifadd intf=PPP2 dest=PVC2
=>ppp iflist
PPP1: dest : PPP1
  Retry : 10    QoS    default    encaps    VC-MUX
  mode = IP Routing
  flags = echo  magic  accomp  mru  addr  route  savepwd  PPPOA
  trans addr = pat    mru = 1500
  route = 0.0.0.0/0 - 0.0.0.0/0 (metric 0)
  user name = guest    password = ********
  admin state = down    oper    state = down    link state = not-connected
  LCP: state = initial    retransm = 10    term. reason =
  IPCP: state = initial   retransm = 0     term. reason =

PPP2: dest : PVC2
  Retry: 10    QoS    default    encaps    VC-MUX
  mode = IP Routing
  flags = echo  magic  accomp  mru  addr  route  savepwd  PPPOA
  mru = 1500
  user name =    password =
  admin state = down    oper state = down    link state = not-connected
  LCP: state = initial    retransm = 10   term. reason =
  IPCP: state = initial   retransm = 0    term. reason =
=>
```

RELATED COMMANDS:

ppp ifattach	Attach a Routed PPP(oA/oE) interface.
ppp ifconfig	Configure a Routed PPP(oA/oE) interface.
ppp ifdelete	Delete a Routed PPP(oA/oE) interface.
ppp ifdetach	Detach a Routed PPP(oA/oE) interface.
ppp iflist	Show current Routed PPP(oA/oE) configuration.

ppp ifattach

Attach (i.e. connect) a Routed PPP(oA/oE) interface.

SYNTAX:

ppp ifattach	intf = <intfname>

intf	The name of the Routed PPP(oA/oE) interface to attach.	REQUIRED

EXAMPLE:

```
=>ppp iflist
PPP1: dest : PPP1
  Retry: 10    QoS    default    encaps    LLC
  mode = IP Routing
  flags = echo  magic  accomp  restart  mru  addr  route  savepwd  PPPOA
  trans addr = pat    mru = 1492
  route = 0.0.0.0/0 - 0.0.0.0/0 (metric 0)
  user name = guest    password = ********
  admin state = down    oper state = down    link state = not-connected
  LCP: state = initial    retransm = 10    term. reason =
  IPCP: state = initial    retransm = 0    term. reason =
=>ppp ifattach =intf=PPP1
=>ppp iflist
PPP1: dest : PPP1
  Retry: 10    QoS    default    encaps    LLC
  mode = IP Routing
  flags = echo  magic  accomp  restart  mru  addr  route  savepwd  PPPOA
  trans addr = pat    mru = 1492
  route = 0.0.0.0/0 - 0.0.0.0/0 (metric 0)
  user name = guest    password = ********
  admin state = up      oper state = down    link state = connected
  LCP: state = initial    retransm = 10    term. reason =
  IPCP: state = initial    retransm = 0    term. reason =
=>ppp iflist
PPP1: dest : PPP1
  Retry: 10    QoS    default    encaps    LLC
  mode = IP Routing
  flags = echo  magic  accomp  restart  mru  addr  route  savepwd  PPPOA
  trans addr = pat    mru = 1492
  route = 0.0.0.0/0 - 0.0.0.0/0 (metric 0)
  user name = guest    password = ********
  admin state = up      oper state = up      link state = connected
  LCP: state = initial    retransm = 10    term. reason =
  IPCP: state = initial    retransm = 0    term. reason =
=>
```

RELATED COMMANDS:

ppp ifadd	Create a Routed PPP(oA/oE) interface.
ppp ifconfig	Configure a Routed PPP(oA/oE) interface.
ppp ifdelete	Delete a Routed PPP(oA/oE) interface.
ppp ifdetach	Detach a Routed PPP(oA/oE) interface.
ppp iflist	Show current Routed PPP(oA/oE) configuration.

ppp ifconfig

Configure a Routed PPP(oA/oE) interface.

Note The interface to be configured may not be connected at the time of configuration. Execute ppp ifdetach prior to executing the ppp ifconfig command.

SYNTAX:

```
ppp ifconfig      intf = <intfname>
                  [dest = <phonebook entry>]
                  [user = <string>]
                  [password = <string>]
                  [qos = <string>]
                  [proto = <{pppoa|pppoe}>]
                  [acname = <string>]
                  [servicename = <string>]
                  [encaps = <{vcmux|llc}>]
                  [pcomp = <{off|on}>]
                  [accomp = <{on|off|negotiate}>]
                  [trace = <{off|on}>]
                  [pap = <{off|on}>]
                  [restart = <{off|on}>]
                  [retryinterval = <number{0-65535}>]
                  [passive = <{off|on}>]
                  [silent = <{off|on}>]
                  [echo = <{off|on}>]
                  [mru = <number{293-8192}>]
                  [laddr = <ip-address>]
                  [raddr = <ip-address>]
                  [netmask = <ip-mask(dotted or cidr)>]
                  [format = <{dotted|cidr}>]
                  [pool = <{poolname|none}>]
                  [savepwd = <{off|on}>]
                  [demanddial = <{off|on}>]
                  [primdns = <ip-address>]
                  [secdns = <ip-address>]
                  [idle = <number{0-1000000}>]
                  [addrtrans = <{none|pat}>]
                  [unnumbered = <{off|on}>]
                  [poolstart = <ip-address>]
                  [poolend = <ip-address>]
                  [status = <{down|up}>]
```

intf	The name of the Routed PPP(oA/oE) interface to configure.	REQUIRED

[dest]	The destination for this Routed PPP(oA/oE) interface. Typically, a phonebook entry. Use: • PPPoA (ppp) phonebook entries For the Routed PPPoA packet service. • ETHoA (bridge) phonebook entries For the Routed PPPoE packet service.	OPTIONAL
[user]	The user name for remote PAP/CHAP authentication.	OPTIONAL
[password]	The password for remote PAP/CHAP authentication.	OPTIONAL
[qos]	The name of a configured Quality Of Service book entry. This parameter never needs to be specified.	OPTIONAL
[proto]	The encapsulation method for the PPP frames, i.e. the applicable packet service for the connection. Select: • pppoa For a Routed PPPoA connection. • pppoe For a Routed PPPoE connection. Per default the PPPoA protocol applies.	OPTIONAL
[acname]	The Access Concentrator name for a Routed PPPoE connection. This parameter is applicable only for Routed PPPoE PPP interfaces (proto=pppoe). Execute the ppp ifscan command to see the names of available access concentrators, if any.	OPTIONAL
[servicename]	The Service Name for a Routed PPPoE connection. This parameter is applicable only for Routed PPPoE PPP interfaces (proto=pppoe). Execute the ppp ifscan command to see the available service names, if any.	OPTIONAL
[encaps]	The type of encapsulation to be used for this Routed PPP(oA/oE) interface. Choose between: • vcmux (default) Standard encapsulation method for PPPoA (ppp) frames. • llc Standard encapsulation method for ETHoA (bridge) frames.	OPTIONAL
[pcomp]	Try (on) or do not try (off) to negotiate PPP protocol compression (LCP PCOMP). Per default the negotiation is disabled (off).	OPTIONAL

[accomp]	Try (on), do never try (off) or negotiate (negotiate) to negotiate PPP address & control field compression (LCP ACCOMP). In the very most cases LCP ACCOMP should not be disabled nor negotiated, i.e. the address field FF-03 should not be sent over ATM. Therefore by default this parameter is enabled (on). In case the accomp parameter is set 'negotiate' the local side of the PPP connection demands to do ACCOMP and adapts itself to the result of this negotiation.	OPTIONAL
[trace]	Enable (on) or disable (off) verbose console logging. By default tracing is disabled (off).	OPTIONAL
[pap]	Select the authentication protocol. Choose between: • on: Password Authentication Protocol (PAP) authentication will be forced, CHAP authentication is not allowed • off: Challenge Handshake Authentication Protocol (CHAP) authentication will be used if available, PAP will be used in cases where CHAP is not available. For security reasons PAP negotiation is disabled (off) per default.	OPTIONAL
[restart]	Automatically restart the connection when Link Control Protocol (LCP) link goes down (on) or do not restart automatically (off). By default restart is disabled (off).	OPTIONAL
[retryinterval]	A number between 0 and 65535 (seconds). Represents the intermediate interval between two retries to establish the connection on ATM level..	OPTIONAL
[passive]	Put the link in listening state in case LCP times out (on) or not (off). This parameter allows to determine whether the link should be left open to wait for incoming messages from the remote side after 10 unsuccessful tries to establish the connection or not. Per default the listening state is disabled.	OPTIONAL
[silent]	Do not send anything at startup and just listen for incoming LCP messages (on) or retry up to 10 times to establish the connection (off). Per default the silent state is disabled.	OPTIONAL
[echo]	Send LCP echo requests at regular intervals (on) or not (off). Per default the sending of LCP echo requests is enabled.	OPTIONAL
[mru]	A number between 293 and 8192. Represents the maximum packet size the SpeedTouch™ should negotiate to be able to receive.	OPTIONAL
[laddr]	The local IP address of the peer-to-peer connection. Specifying a local IP address forces the remote side of the PPP link (if it allows to) to accept this IP address as the SpeedTouch™ PPP session IP address. If not specified, the SpeedTouch™ will accept any IP address. Typically the local IP address parameter is not specified.	OPTIONAL

[raddr]	The remote IP address of the peer-to-peer connection. Specifying a remote IP address forces the remote side of the PPP link (if it allows to) to accept this IP address as its PPP session IP address. If not specified, the SpeedTouch™ will accept any IP address. Typically the remote IP address parameter is not specified.	OPTIONAL
[netmask]	The subnetmask associated with this address. Specifying a subnetmask forces the remote side (if it allows to) to accept this subnetmask as the PPP session subnetmask. If not specified, the SpeedTouch™ will accept any subnetmask. The SpeedTouch™ will only request/accept a subnetmask if a DHCP server pool is associated, i.e. if the [pool] parameter is specified.	OPTIONAL
[format]	The negotiated subnetmask specified in the netmask parameter is specified in the dotted format (dotted) or in Classles Inter Domain Routing (CIDR) format (cidr). Per default the format is CIDR.	OPTIONAL
[pool]	The name of the free DHCP server pool to which the acquired IP subnet must be assigned to.	OPTIONAL
[savepwd]	Save password (on), if supplied, or do not save the password (off). Per default the saving of the password is disabled.	OPTIONAL
[demanddial]	Enable (on) or disable (off) the dial-on-demand feature.	OPTIONAL
[primdns]	The IP address of the primary DNS server. In case a primary DNS server is specified the SpeedTouch™ will negotiate this IP address with the remote side. If not specified, the SpeedTouch™ will accept any IP address.	OPTIONAL
[secdns]	The IP address of the (optional) secondary DNS server. In case a secondary DNS server is specified the SpeedTouch™ will negotiate this IP address with the remote side. If not specified, the SpeedTouch™ will accept any IP address.	OPTIONAL
[idle]	A number between 1 and 1000000 (seconds). Represents after how many seconds an idle link goes down.	OPTIONAL
[addrtrans]	Automatically enable address translation for the IP address of this link (pat) or do not use address translation (none).	OPTIONAL
[unnumbered]	Takes the local IP address from 'laddr' field and remote IP address from the IP address pool assigned to the incoming PPP link. In case the unnumbered parameter is disabled the same IP address is used for each connection on the server side, thus reducing the number of used IP addresses.	OPTIONAL
[poolstart]	The lower bound of the IP address pool assigned to the incoming PPP link.	OPTIONAL
[poolend]	The upper bound of the IP address pool assigned to the incoming PPP link.	OPTIONAL
[status]	Force automatically to attach the PPP interface (up) or use the regular ppp ifattach command (down). Per default the startup status is down (recommended).	OPTIONAL

EXAMPLE:

```
=>ppp iflist
PPP1: dest : PPP1
  Retry: 10    QoS    default    encaps    LLC
  mode = IP Routing
  flags = echo  magic  accomp  restart  mru  addr  route  savepwd  PPPOA
  trans addr = pat    mru = 1492
  route = 0.0.0.0/0 - 0.0.0.0/0 (metric 0)
  user name = My_Connection@MY_ISP    password = ********
  admin state = down    oper state = down    link state = not-connected
  LCP: state = initial    retransm = 10    term. reason =
  IPCP: state = initial    retransm = 0    term. reason =
=>ppp ifconfig intf=PPP1 prot=pppoa encaps=vcmux
=>ppp iflist
PPP1: dest : PPP1
  Retry: 10    QoS    default    encaps    VC-MUX
  mode = IP Routing
  flags = echo  magic  accomp  restart  mru  addr  route  savepwd  PPPOA
  trans addr = pat    mru = 1492
  route = 0.0.0.0/0 - 0.0.0.0/0 (metric 0)
  user name = My_Connection@MY_ISP    password = ********
  admin state = down    oper state = down    link state = not-connected
  LCP: state = initial    retransm = 10    term. reason =
  IPCP: state = initial    retransm = 0    term. reason =
=>
```

RELATED COMMANDS:

ppp ifadd	Create a Routed PPP(oA/oE) interface.
ppp ifattach	Attach a Routed PPP(oA/oE) interface.
ppp ifdelete	Delete a Routed PPP(oA/oE) interface.
ppp ifdetach	Detach a Routed PPP(oA/oE) interface.
ppp iflist	Show current Routed PPP(oA/oE) configuration.

ppp ifdelete

Delete a Routed PPP(oA/oE) interface.

SYNTAX:

```
ppp ifdelete      intf = <intfname>
```

intf	The name of the Routed PPP(oA/oE) interface to delete.

EXAMPLE:

```
=>ppp iflist
PPP1: dest : PPP1
  Retry: 10    QoS    default    encaps    VC-MUX
  mode = IP Routing
  flags = echo  magic  accomp  restart  mru  addr  route  savepwd  PPPOA
  trans addr = pat    mru = 1500
  route = 0.0.0.0/0 - 0.0.0.0/0 (metric 0)
  user name = guest    password = ********
  admin state = down    oper state = down    link state = not-connected
  LCP: state = initial    retransm = 10    term. reason =
  IPCP: state = initial    retransm = 0    term. reason =

PPP2: dest : PVC2
  Retry: 10    QoS    default    encaps    VC-MUX
  mode = IP Routing
  flags = echo  magic  accomp  restart  mru  addr  savepwd  PPPOA
  mru = 1500
  user name =    password =
  admin state = down    oper state = down    link state = not-connected
  LCP: state = initial    retransm = 10    term. reason =
  IPCP: state = initial    retransm = 0    term. reason =
=>ppp ifdelete intf=PPP2
=>ppp iflist
PPP1: dest : PPP1
  Retry: 10    QoS    default    encaps    VC-MUX
  mode = IP Routing
  flags = echo  magic  accomp  restart  mru  addr  route  savepwd  PPPOA
  trans addr = pat    mru = 1500
  route = 0.0.0.0/0 - 0.0.0.0/0 (metric 0)
  user name = guest    password = ********
  admin state = down    oper state = down    link state = not-connected
  LCP: state = initial    retransm = 10    term. reason =
  IPCP: state = initial    retransm = 0    term. reason =
=>
```

RELATED COMMANDS:

ppp ifadd	Create a Routed PPP(oA/oE) interface.
ppp ifattach	Attach a Routed PPP(oA/oE) interface.
ppp ifconfig	Configure a Routed PPP(oA/oE) interface.
ppp ifdetach	Detach a Routed PPP(oA/oE) interface.
ppp iflist	Show current Routed PPP(oA/oE) configuration.

ppp ifdetach

Detach a Routed PPP(oA/oE) interface.

SYNTAX:

```
ppp ifdetach      intf = <intfname>
```

intf	The name of the Routed PPP(oA/oE) interface.	REQUIRED

EXAMPLE:

```
=>ppp iflist
PPP1: dest : PPP1
  Retry: 10    QoS    default    encaps    VC-MUX
  mode = IP Routing
  flags = echo  magic  accomp  restart  mru  addr  route  savepwd  PPPOA
  trans addr = pat    mru = 1492
  route = 0.0.0.0/0 - 0.0.0.0/0 (metric 0)
  user name = guest    password = ********
  admin state = up      oper state = up      link state = connected
  LCP: state = initial    retransm = 0      term. reason =
  IPCP: state = initial    retransm = 10    term. reason =
=>ppp ifdetach =intf=PPP1
=>ppp iflist
PPP1: dest : PPP1
  Retry: 10    QoS    default    encaps    VC-MUX
  mode = IP Routing
  flags = echo  magic  accomp  restart  mru  addr  route  savepwd  PPPOA
  trans addr = pat    mru = 1492
  route = 0.0.0.0/0 - 0.0.0.0/0 (metric 0)
  user name = guest    password = ********
  admin state = down    oper state = down    link state = not-connected
  LCP: state = initial    retransm = 10    term. reason =
  IPCP: state = initial    retransm = 0    term. reason =
=>
```

RELATED COMMANDS:

ppp ifadd	Create a Routed PPP(oA/oE) interface.
ppp ifattach	Attach a Routed PPP(oA/oE) interface.
ppp ifconfig	Configure a Routed PPP(oA/oE) interface.
ppp ifdelete	Delete a Routed PPP(oA/oE) interface.
ppp iflist	Show current Routed PPP(oA/oE) configuration.

ppp iflist

Show current configuration of all or a specified Routed PPP(oA/oE) interface(s).

SYNTAX:

```
ppp iflist        [intf = <intfname>]
```

intf	The name of the Routed PPP(oA/oE) interface. If this parameter is not specified, all Routed PPP(oA/oE) interfaces are shown.	OPTIONAL

EXAMPLE INPUT/OUTPUT :

```
=>ppp iflist
PPP1: dest : PPP1
  Retry: 10    QoS    default    encaps    VC-MUX
  mode = IP Routing
  flags = echo  magic  accomp  restart  mru  addr  route  savepwd  PPPOA
  trans addr = pat    mru = 1500
  route = 0.0.0.0/0 - 0.0.0.0/0 (metric 0)
  user name = guest    password = ********
  admin state = down    oper state = down    link state = not-connected
  LCP: state = initial    retransm = 10    term. reason =
  IPCP: state = initial    retransm = 0    term. reason =

PPP2: dest : PVC2
  Retry: 10    QoS    default    encaps    VC-MUX
  mode = IP Routing
  flags = echo  magic  accomp  restart  mru  addr  savepwd  PPPOA
  mru = 1500
  user name =    password =
  admin state = down    oper state = down    link state = not-connected
  LCP: state = initial    retransm = 10    term. reason =
  IPCP: state = initial    retransm = 0    term. reason =
=>
```

RELATED COMMANDS:

ppp ifadd	Create a Routed PPP(oA/oE) interface.
ppp ifattach	Attach a Routed PPP(oA/oE) interface.
ppp ifconfig	Configure a Routed PPP(oA/oE) interface.
ppp ifdelete	Delete a Routed PPP(oA/oE) interface.
ppp ifdetach	Detach a Routed PPP(oA/oE) interface.

ppp ifscan

Scan a Routed PPPoE interface (proto=pppoe) for available Access Concentrator names and Service Names.

Note Execute the ppp ifdetach command for this interface before performing a scan on it.

SYNTAX:

```
ppp ifscan        intf = <intfname>
                  [time = <number{0-36000}>]
                  [kit = <number{0-8}>]
```

intf	The name of the RoutedPPPoE interface to scan.	REQUIRED
[time]	A number between 0 and 36000 (seconds). Represents the time to scan for services.	OPTIONAL
[kit]	A number between 0 and 8. Represents the way the scan progress is visually indicated. Per default no progress indicator is applied (kit=0). kit=1 up to kit=8 are diverse progress indicators. Try it !	OPTIONAL

EXAMPLE:

```
=>ppp iflist
PPP1: dest : PPP1
  Retry: 10    QoS    default    encaps    VC-MUX
  mode = IP Routing
  flags = echo  magic  accomp  restart  mru  addr  route  savepwd  PPPOA
  trans addr = pat    mru = 1492
  route = 0.0.0.0/0 - 0.0.0.0/0 (metric 0)
  user name = guest    password = ********
  admin state = down    oper state = down    link state = not-connected
  LCP: state = initial    retransm = 10    term. reason =
  IPCP: state = initial    retransm = 0    term. reason =
=>ppp ifscan intf=PPP1 time=45
        Service Name                          Access Concentrator

Done !
=>
```

RELATED COMMANDS:

ppp ifconfig	Configure a Routed PPP(oA/oE) interface.

ppp rtadd

Automatically add a route configuration to the routing table in case the specified Routed PPP(oA/oE) interface link comes up.
This route configuration will determine which local hosts are allowed to use this link and/or which remote destinations should be or should not be reachable.

Note Execute the ppp ifdetach command for this interface prior to configuring routes.

SYNTAX:

```
ppp rtadd          intf = <intfname>
                   dst = <ip-address>
                   [dstmsk = <ip-mask(dotted or cidr)>]
                   [src = <ip-address>]
                   [srcmsk = <ip-mask(dotted or cidr)>]
                   [metric = <number{0-100}>]
```

intf	The name of the Routed PPP(oA/oE) interface.	REQUIRED
dst	The destination IP address for the route to be added when the link comes up.	REQUIRED
[dstmsk]	The destination IP mask. Depending on the destination netmask: • Any remote destination is reachable, i.e. the Routed PPP(oA/oE) connection acts as default route (dstmsk=0) • Only the remote (sub)net is reachable (dstmsk=1) • The actual destination mask will be the default netmask applicable for destination IP address • Only the single remote host is reachable (dstmsk=32) • Any valid (contiguous) netmask in case of Variable Length Subnet Masking (VLSM).	OPTIONAL
[src]	The source IP address specification for the route to be added when the link comes up.	OPTIONAL

[srcmsk]	The source IP mask. Depending on the source netmask: • Everybody is allowed to use this Routed PPP(oA/oE) connection (dstmsk=0) • Only members of the same subnet as the host which opened the Routed PPP(oA/oE) connection are allowed to use the Routed PPP(oA/oE) connection (dstmsk=1) • The actual destination mask will be the netmask applicable for the IP address of the host which opened the Routed PPP(oA/oE) connection. • Only the host which opened the Routed PPP(oA/oE) connection is allowed to use the Routed PPP(oA/oE) connection. (dstmsk=32) • Any valid (contiguous) netmask in case of VLSM.	OPTIONAL
[metric]	The route metric, i. e. the cost factor of the route. Practically, the cost is determined by the hop count. **Note** It is recommended not to use this parameter.	OPTIONAL

EXAMPLE:

```
=>ppp iflist
PPP1: dest : PVC3
  Retry: 10    QoS    default    encaps    LLC
  mode = IP Routing
  flags = echo  magic  accomp  restart  mru  addr  route  savepwd  PPPOE
  trans addr = pat    mru = 1492
  user name = guest    password = ********
  admin state = down    oper state = down    link state = not-connected
  LCP : state = initial    retransm = 10   term. reason =
  IPCP: state = initial    retransm = 0    term. reason =
=>ppp rtadd intf=PPP1 dst=172.16.0.5 dstmsk=24 src=10.0.0.2 srcmask=24
=>ppp iflist
PPP1: dest : PVC3
  Retry: 10    QoS    default    encaps    LLC
  mode = IP Routing
  flags = echo  magic  accomp  restart  mru  addr  route  savepwd  PPPOE
  trans addr = pat    mru = 1492
  route = 10.0.0.2/24 - 172.16.0.5/24 (metric 1)
  user name = guest    password = ********
  admin state = down    oper state = down    link state = not-connected
  LCP : state = initial    retransm = 10   term. reason =
  IPCP: state = initial    retransm = 0    term. reason =
=>
```

RELATED COMMANDS:

ppp rtdelete	Delete the route specification for an upcoming Routed PPP(oA/oE) link.

ppp rtdelete

Delete the route specification for a Routed PPP(oA/oE) link.

Note Execute the ppp ifdetach command for this interface prior to deleting route configurations.

SYNTAX:

```
ppp rtdelete      intf = <intfname>
```

intf	The Routed PPP(oA/oE) interface name for which to delete the route settings.	REQUIRED

EXAMPLE:

```
=>ppp iflist
PPP1: dest : PVC3
  Retry: 10    QoS    default    encaps    LLC
  mode = IP Routing
  flags = echo  magic  accomp  restart  mru  addr  route  savepwd  PPPOE
  trans addr = pat    mru = 1492
  route = 10.0.0.2/24 - 172.16.0.5/24 (metric 1)
  user name = guest    password = ********
  admin state = down    oper state = down    link state = not-connected
  LCP : state = initial    retransm = 10   term. reason =
  IPCP: state = initial    retransm = 0    term. reason =
=>ppp rtdelete intf=PPP1
=>ppp iflist
PPP1: dest : PVC3
  Retry: 10    QoS    default    encaps    LLC
  mode = IP Routing
  flags = echo  magic  accomp  restart  mru  addr  route  savepwd  PPPOE
  trans addr = pat    mru = 1492
  user name = guest    password = ********
  admin state = down    oper state = down    link state = not-connected
  LCP : state = initial    retransm = 10   term. reason =
  IPCP: state = initial    retransm = 0    term. reason =
=>
```

RELATED COMMANDS:

ppp rtadd	Configure a route specification for an upcoming Routed PPP(oA/oE) link.

PPTP Commands

Contents

This chapter covers the following commands:

pptp flush

Flush complete Relayed PPPoA, often referred to as PPPoA/Point-to-Point Tunneling Protocol (PPTP) configuration.

Note The flush command does not impact previously saved configurations.

SYNTAX:

```
pptp flush
```

EXAMPLE:

```
=>pptp profadd name=Relay_PPP1 encaps=nlpid ac=always
=>pptp proflist
Profile      QoS      Encaps      AC
Relay_PPP1   default  nlpid       always
=>pptp flush
=>pptp proflist
=>
```

pptp list

Show current Relayed PPPoA configuration.

SYNTAX:

```
pptp list
```

EXAMPLE INPUT/OUTPUT:

```
=>pptp list
Dialstr    Destination   QoS       Encaps   AC      State       User
           DIALUP_PPP3   default   vcmux    never   CONNECTED   (10.0.0.2)
=>
```

pptp profadd

Define a new Relayed PPPoA profile.

SYNTAX:

```
pptp profadd      name = <string>
                  [qos = <string>]
                  [encaps = <{vcmux|nlpid}>]
                  [ac = <{never|always|keep}>]
```

name	The name for the Relayed PPPoA profile.	REQUIRED
[qos]	The name of the Quality Of Service book entry. This parameter never needs to be specified.	OPTIONAL
[encaps]	The type of encapsulation applicable to Relayed PPPoA interfaces using this Relayed PPPoA profile. Choose between: • vcmux • nlpid - Network Layer Protocol IDentifiers (NLPID)	OPTIONAL
[ac]	The High-level Data Link Control (HDLC) framing option applicable to Relayed PPPoA interfaces using this Relayed PPPoA profile. Before relaying the encapsulated PPP frames over the PPPoA link, make sure that the address and control field (0xFF03) is always in front of the frames (always), make sure the address and control field will never be found in front of the frames (never) or do not change the frames arriving via the PPTP tunnel (keep). By default the address and control field is never sent (compliant to RFC2364). It is recommended to keep this setting.	OPTIONAL

EXAMPLE:

```
=>pptp proflist
Profile      QoS      Encaps      AC
Relay_PPP1   default  nlpid       always
=>pptp profadd name=PPTPLink encaps=vcmux ac=never
=>pptp proflist
Profile      QoS      Encaps      AC
Relay_PPP1   default  nlpid       always
PPTPLink     default  vcmux       never
=>
```

RELATED COMMANDS:

pptp profdelete	Delete a Relayed PPPoA profile.
pptp proflist	Show current Relayed PPPoA profiles.

pptp profdelete

Delete a Relayed PPPoA profile.

SYNTAX:

```
pptp profdelete   name <string>
```

name	The name for the Relayed PPPoA profile.	REQUIRED

EXAMPLE:

```
=>pptp proflist
Profile      QoS      Encaps      AC
Relay_PPP1   default  nlpid       always
PPTPLink     default  vcmux       never
=>pptp profdelete name=PPTPLink
=>pptp proflist
Profile      QoS      Encaps      AC
Relay_PPP1   default  nlpid       always
=>
```

RELATED COMMANDS:

pptp profadd	Define a new Relayed PPPoA profile.
pptp proflist	Show current Relayed PPPoA profiles.

pptp proflist

Show all current Relayed PPPoA profiles.

SYNTAX:

```
pptp proflist
```

EXAMPLE:

```
=>pptp proflist
Profile      QoS      Encaps      AC
Relay_PPP1   default  nlpid       always
PPTPLink     default  vcmux       never
=>
```

RELATED COMMANDS:

pptp profadd	Define a new Relayed PPPoA profile.
pptp profdelete	Delete a Relayed PPPoA profile.

QoSBook Commands

Contents

This chapter covers the following commands:

qosbook add

Add a Quality of Service book entry.

SYNTAX:

```
qosbook add          name = <string>
                     class = <{ubr|cbr|vbr-nrt}>
                     [tx_peakrate = <number{0-2147483}>]
                     [tx_sustrate = <number{0-2147483}>]
                     [tx_maxburst = <number{0-2147483600}>]
                     [rx_peakrate = <number{0-2147483}>]
                     [rx_sustrate = <number{0-2147483}>]
                     [rx_maxburst = <number{0-2147483600}>]
```

name	The name for the new QoS entry.	REQUIRED
class	The ATM service category. Choose between: • ubr: unspecified bit rate • cbr: constant bit rate • vbr-nrt: variable bit rate - non real time	REQUIRED
[tx_peakrate]	A number between 0 and 2147483 (Kilobits per second). Indicates the peak rate in the transmit (upstream) direction. Use tx_peakrate=0 to indicate Linerate.	OPTIONAL
[tx_sustrate]	A number between 0 and 2147483 (Kilobits per second). Indicates the sustainable rate in the transmit (upstream) direction. Only applicable in case class=vbr-nrt.	OPTIONAL
[tx_maxburst]	A number between 0 and 2147483600 (bytes per second). Indicates the maximum burst size in the transmit (upstream) direction. Only applicable in case class=vbr-nrt.	OPTIONAL
[rx_peakrate]	A number between 0 and 2147483 (Kilobits per second). Indicates the peak rate in the receive (downstream) direction. Use rx_peakrate=0 to indicate Linerate. Only applicable in an SVC environment.	OPTIONAL
[rx_sustrate]	A number between 0 and 2147483 (Kilobits per second). Indicates the sustainable rate in the receive (downstream) direction. Only applicable in an SVC environment with class=vbr-nrt.	OPTIONAL
[rx_maxburst]	A number between 0 and 2147483600 (bytes per second). Indicates the maximum burst size in the receive (downstream) direction. Only applicable in an SVC environment with class=vbr-nrt.	OPTIONAL

EXAMPLE:

```
=>qosbook list
Name     Ref Type     TX peak   sust     burst    RX peak   sust     burst
                       (Kbits)   (Kbits)  (bytes)  (Kbits)   (Kbits)  (bytes)
default  4   ubr      linerate  0        0        linerate  0        0
=>qosbook add
name=TestVBR
class=vbr-nrt
[tx_peakrate]=
[tx_sustrate]=1000
[tx_maxburst]=100
[rx_peakrate]=
[rx_sustrate]=1000
[rx_maxburst]=100

=>qosbook list
Name     Ref  Type     TX peak   sust     burst    RX peak   sust     burst
                       (Kbits)   (Kbits)  (bytes)  (Kbits)   (Kbits)  (bytes)
default  4    ubr      linerate  0        0        linerate  0        0
TestVBR  0    vbr-nrt   linerate   1000     144       linerate   1000      144
=>
```

IMPORTANT NOTE:

The SpeedTouch™ always rounds up specified burst sizes to a multiple of 48 bytes, i.e. a multiple of ATM cells.

Example
In the example above a burst size of 100 bytes is specified (tx_maxburst=100). The SpeedTouch™ will round up the burst size to the closest matching multiple of 48 bytes, as can be seen when displaying the profile via :qosbook list (burst=144).

RELATED COMMANDS:

qosbook delete	Remove a QoS book entry.
qosbook list	Show current QoS book.

qosbook delete

Remove a Quality of Service book entry.

SYNTAX:

```
qosbook delete        name = <string>
                      [force = <{no|yes}>]
```

name	The name of the QoS book entry to delete.	REQUIRED
[force]	Force deletion of the entry even if it is still in use (yes) or do not force the deletion (no). By default forced deletion is disabled.	OPTIONAL

EXAMPLE:

```
=>qosbook list
Name     Ref Type    TX peek   sust     burst    RX peek   sust     burst
                     (Kbits)   (Kbits)  (bytes)  (Kbits)   (Kbits)  (bytes)
default  24  ubr     Linerate  0        0        Linerate  0        0
voice    0   cbr     64        0        0        64        0        0
PPP3     1   ubr     6144      0        0        Linerate  0        0
=>qosbook delete name voice
=>qosbook list
Name     Ref Type    TX peek   sust     burst    RX peek   sust     burst
                     (Kbits)   (Kbits)  (bytes)  (Kbits)   (Kbits)  (bytes)
default  24  ubr     Linerate  0        0        Linerate  0        0
PPP3     1   ubr     6144      0        0        Linerate  0        0
=>
```

RELATED COMMANDS:

qosbook add	Add a QoS book entry.
qosbook list	Show current QoS book.

qosbook flush

Flush complete Quality of Service book.

Note The flush command does not impact previously saved configurations.

SYNTAX:

```
qosbook flush
```

RELATED COMMANDS:

qosbook load	Load saved or default QoS book.
qosbook save	Save current QoS book.

qosbook list

Show current Quality of Service book.

SYNTAX:

```
qosbook list
```

EXAMPLE OUTPUT:

```
=>qosbook list
Name    Ref Type    TX peek  sust     burst    RX peek  sust     burst
                    (Kbits)  (Kbits)  (bytes)  (Kbits)  (Kbits)  (bytes)
default 24  ubr     Linerate 0        0        Linerate 0        0
PPP3    1   ubr     6144     0        0        Linerate 0        0
=>qosbook add name voice class cbr tx_peakrate 64 rx_peakrate 64
=>qosbook list
Name    Ref Type    TX peek  sust     burst    RX peek  sust     burst
                    (Kbits)  (Kbits)  (bytes)  (Kbits)  (Kbits)  (bytes)
default 24  ubr     Linerate 0        0        Linerate 0        0
voice   0   cbr     64       0        0        64       0        0
PPP3    1   ubr     6144     0        0        Linerate 0        0
=>
```

RELATED COMMANDS:

qosbook add	Add a QoS book entry.
qosbook delete	Remove a QoS book entry.

Software Commands

Contents

This chapter covers the following commands:

software cleanup

Remove all unused files from the passive software subdirectory.
This command frees the passive software subdirectory from corrupted software files and configuration files. Software marked as passive software is not deleted.

SYNTAX:

```
software cleanup
```

RELATED COMMANDS:

software deletepassive	Delete the passive software.
software setpassive	Mark an uploaded file as passive software version.

software deletepassive

Delete passive software.

SYNTAX:

```
software deletepassive
```

EXAMPLE:

```
=>Software version
Active   :   Sascha.abc        Passive   :   Bene.cba
=>Software deletepassive
=>Software version
Active   :   Sascha.abc        Passive   :
=>
```

RELATED COMMANDS:

software cleanup	Remove all unused files from the passive software subdirectory.
software duplicate	Duplicate the active software as passive software.
software setpassive	Mark a file as passive software version.

software duplicate

Duplicate the active software as passive software.

SYNTAX:

```
software duplicate
```

EXAMPLE:

```
=>Software version
Active   :   Sascha.abc        Passive   :
=>Software duplicate
=>Software version
Active   :   Sascha.abc        Passive   :   Sascha.abc
=>
```

RELATED COMMANDS:

software cleanup	Remove all unused files from the passive software subdirectory.
software deletepassive	Delete passive software.
software setpassive	Mark a file as passive software version.

software setpassive

Mark a file as passive software version. Only correctly uploaded software, valid for the SpeedTouch™, can be marked as passive software.,

SYNTAX:

```
software setpassive  file = <string>
```

file	The filename (without directory path) of the software package.	REQUIRED

EXAMPLE:

```
=>Software version
Active   :   Sascha.abc        Passive   :   Bene.cba
=>Software deletepassive
=>Software version
Active   :   Sascha.abc        Passive   :
.....
(FTP file transfer or upload via the SpeedTouch™ pages of new software Sascha.xyz)
.....
=>software setpassive file=Sascha.xyz
=>Software   version
Active   :   Sascha.abc        Passive   :   Sascha.xyz
=>
```

RELATED COMMANDS:

software cleanup	Remove all unused files from the passive software subdirectory.
software deletepassive	Delete passive software.

software switch

Switch active and passive versions and reboot the SpeedTouch™.

Because rebooting implies a flush of all non-saved configurations it is highly recommended to save the current configuration if needed, e.g. via saveall or :config save before executing a software switch.

SYNTAX:

```
software switch
```

EXAMPLE:

```
=>Software version
Active   :   Sascha.abc        Passive   :   Sascha.xyz
=>software switch
.....
(after reboot and re-opening the Telnet session)
.....
=>Software version
Active   :   Sascha.xyz        Passive   :   Sascha.abc
=>
```

RELATED COMMANDS:

software version	Show active and passive software versions.
system reboot	Reboot the SpeedTouch™.

software version

Show active and passive software versions.

SYNTAX:

```
software version
```

EXAMPLE:

```
=>Software version
Active   :   Sascha.xyz        Passive   :   Sascha.abc
=>
```

RELATED COMMANDS:

software switch	Switch active and passive software versions and reboot the SpeedTouch™.

Note

System Commands

Contents

This chapter covers the following commands:

system clearpassword

Clear current SpeedTouch™ system password.

Note To avoid unrestricted and unauthorized access to the SpeedTouch™, it is highly recommended to make sure it is protected by a SpeedTouch™ system password (via :system setpassword) and to change the password on a regular basis.

SYNTAX:

```
system clearpassword
```

EXAMPLE:

```
=>system clearpassword
Security notification: Password changed, use 'saveall' to make it permanent.
=>
```

RELATED COMMANDS:

system setpassword	Set/change current system password.

system config

Show/set SpeedTouch™ system configuration parameters.

Note For a good operation of Universal Plug and Play (UPnP) and the discovery mechanism, it is highly recommended not to change the System config settings.

SYNTAX:

```
system config          [upnp = <{disabled|enabled}>]
                       [mdap = <{disabled|enabled}>]
                       [dcache = <{disabled|enabled}>]
```

[upnp]	Enable (enabled) or disable (disabled) Universal Plug and Play (UPnP) discovery. By default UPnP discovery is enabled.	OPTIONAL
[mdap]	Enable (enabled) or disable (disabled) proprietary discovery protocol. By default MDAP discovery is enabled.	OPTIONAL
[dcache]	Enable (enabled) or disable (disabled) data cache. By default data cache is enabled. For internal use only. Do not alter in any way.	OPTIONAL

EXAMPLE:

```
=>system config
upnp discovery: on
mdap discovery: on
dcache:         on
=>
```

Note Please do not change the System config settings. In case of Setup Wizard discovery problems, check whether the System config settings are listed as in the example above. If needed, execute the System config command as listed in the example below:

system flush

Flush current SpeedTouch™ system configuration, i.e. the System password and the system config settings (dcache excluded).

Note To avoid unrestricted and unauthorized access to the SpeedTouch™, it is highly recommended to make sure it is protected by a SpeedTouch™ system password (via :system setpassword) and to change the password on a regular basis.

Note The flush command does not impact previously saved configurations.

SYNTAX:

```
system flush
```

EXAMPLE:

```
=>system flush
Security notification: Password changed, use 'saveall' to make it permanent.
=>
```

system reboot

Reboot the SpeedTouch™. Non-saved configuration settings are lost after reboot.

SYNTAX:

```
system reboot
```

EXAMPLE:

```
=>system reboot
.....
(lost session connectivity due to reboot)
.....
```

system reset

Reset the SpeedTouch™ to its factory default settings and reboot the device. All user and Service Provider specific settings and all saved configuration changes are lost after reboot.

SYNTAX:

`system reset`	`no/yes = <{no|yes}>`

no/yes	Proceed with resetting the SpeedTouch™ device to its factory default settings (yes) or not (no). By default the system reboot command is discarded if no explicit positive confirmation is given.	REQUIRED

EXAMPLE:

```
=>system reset
---------------
 !! WARNING !!
---------------
The SpeedTouch(TM)610 DSL Router will be reset to factory defaults clearing all user and ISP speci
fic settings.
Connectivity with the ISP network might be lost.
Do you want to proceed ?
no/yes = no
=>
=>system reset
---------------
 !! WARNING !!
---------------
The SpeedTouch(TM)610 DSL Router will be reset to factory defaults clearing all user and ISP speci
fic settings.
Connectivity with the ISP network might be lost.
Do you want to proceed ?
no/yes = yes

.....
(lost session connectivity due to reboot)
.....
```

system setpassword

Set/change the current SpeedTouch™ system password.

Note To avoid unrestricted and unauthorized access to the SpeedTouch™, it is highly recommended to make sure it is protected by a SpeedTouch™ system password (via :system setpassword) and to change the password on a regular basis

SYNTAX:

```
system setpassword    password = {<string>|$_BOARD_SERIAL_NBR}
```

password	the system password can be set to either: • <string> A free to choose password <string>	REQUIRED

EXAMPLE:

```
=>system setpassword password=Sascha
Security notification: Password changed, use 'saveall' to make it permanent.
=>saveall
=>
```

RELATED COMMANDS:

system clearpassword	Clear current system password.

TD Commands

Contents

This chapter covers the following commands:

td call

Call a ‘Trace & Debug‘ command.

Note For qualified personnel only.

SYNTAX:

```
td call          cmd = <string>
```

cmd	Quoted 'Trace & Debug' command string.	REQUIRED

UPnP Commands

Contents

This chapter covers the following commands:

upnp config

Configure UPnP parameter(s). Currently only maxage.
This parameter allows to configure how often the SpeedTouch™ sends a notification message to advertise its presence as an Internet Gateway Device (IGD) on the network.
Setting this parameter to a low value will increase the number of packets sent over time on the network, but will make the state of the device more up to date.

SYNTAX:

`upnp config`	`[maxage = <number{60-999999}>`	
maxage	The maximum age of the ssdp advertisements (in seconds). Default is 1800 seconds.	OPTIONAL
	Note Enter the upnp config command without a value to check thee currently configured value.	

EXAMPLE:

```
=>upnp config

ssdp max-age: 1800

=>upnp config maxage=4200
=>upnp config

ssdp max-age: 4200
```

RELATED COMMANDS:

upnp flush	Flushes the upnp configuration (i.e. reset to default configuration).
upnp list	List all registered UPnP devices.

upnp flush

Flushes the UPnP configuration (i.e. reset to default configuration).

SYNTAX:

```
upnp flush
```

EXAMPLE:

```
=>upnp config

ssdp max-age: 4200

=>upnp flush
=>upnp config

ssdp max-age: 1800
```

RELATED COMMANDS:

upnp config	Configure upnp parameter(s).
upnp list	List all registered UPnP devices.

upnp list

This command lists the devices and services currently offered by the SpeedTouch™. E.g. use this command to check whether a PPP connection is properly configured and thus advertised as a PPP connection.

SYNTAX:

`upnp list`	`[verbose = <number{0-2}>]`

verbose	Verbose level. Default is 1.	OPTIONAL

EXAMPLE:

```
=>upnplist
---------------- device: IGD.xml ------------
++ Root Device: urn:schemas-upnp-org:device:InternetGatewayDevice:1
-- Sub-device 1: urn:schemas-upnp-org:device:LANDevice:1
-- Sub-device 2: urn:schemas-upnp-org:device:WANDevice:1
-- Sub-device 3: urn:schemas-upnp-org:device:WANConnectionDevice:1
-- Service 1: urn:upnp-org:serviceId:layer3f
-- Service 2: urn:upnp-org:serviceId:lanhcm
-- Service 3: urn:upnp-org:serviceId:wancic
-- Service 4: urn:upnp-org:serviceId:wandsllc:pppoa
-- Service 5: urn:upnp-org:serviceId:wanpppc:pppoa
---------------------- end -----------------
=>
```

RELATED COMMANDS:

upnp flush	Flushes the upnp configuration (i.e. reset to default configuration).
upnp config	Configure upnp parameter(s).

Wireless Commands

Introduction

The commands covered in this chapter can only be used with the SpeedTouch™570.

Contents

This chapter covers the following commands:l

wireless wepconfig

Configures Wired Equivalent Privacy (WEP) secure access.

SYNTAX:

```
wireless wepconfig          [key = <password>]
                            [encryption = <{off|on}>]
```

key	The WEP key. In case of 64 bit encryption, the 40-bits WEPkey must consist of 10 hexadecimal digits. In case of 128 bit encryption, the 104-bits WEPkey must consist of 26 hexadecimal digits.	OPTIONAL
encryption	Encryption is turned on/off	OPTIONAL

EXAMPLE:

```
=>wireless wepconfig key=0123456789abcdef0123456789 encryption=on
=>
```

wireless aclconfig

Configures the Access Control List (ACL) mode.

SYNTAX:

```
aclconfig                    [control = <{register|lock|unlock}>]
```

control	Select the Access Control List mode. Choose between: • register button mode • lock/unlock mode Default setting is Register Button mode.	OPTIONAL

DESCRIPTION:

Register	A wireless client is only added to the ACL after using the Register Button.
Lock	The ACL is locked. No new entries are allowed in the ACL.
Unlock	Every wireless client is unconditionally added to the ACL.

EXAMPLE:

```
=>wireless aclconfig
Access control list : Register Button mode
=>wireless aclconfig control=lock
=>wireless aclconfig
Access control list : ACL locked (lock/unlock mode)
=>
```

wireless acllist

Lists the ACL.

SYNTAX:

```
wireless acllist
```

EXAMPLE:

```
=>wireless acllist
Station Name    MAC address         Allowed
Test            00:02:2d:42:E5:2D : yes
Test2           01:67:D0:23:56:CD : yes
=>
```

RELATED COMMANDS:

wireless aclflush	Flushes the ACL.
wireless aclentry	Adds or deletes an entry in the ACL.

wireless aclflush

Flushes the ACL.

SYNTAX:

```
wireless aclflush
```

EXAMPLE:

```
=>wireless acllist
Station Name    MAC address         Allowed
Test            00:90:D0:01:47:DE : yes
Test2           01:67:D0:23:56:CD : yes
=>wireless aclflush
=>wireless acllist
The access control list is empty.
=>
```

RELATED COMMANDS:

wireless acllist	Lists the ACL.
wireless aclentry	Adds or deletes an entry in the ACL.

wireless aclentry

Adds or deletes an entry in the ACL.

SYNTAX:

```
wireless aclentry            hwaddr = <hardware-address>
                             action = <{allow|deny|delete}>
                             [name = <quoted string>]
```

hwaddr	The Ethernet MAC address of the ACL entry	REQUIRED
action	The action to be performed on ACL entry. Choose between: • allow • deny • delete	REQUIRED
name	The name of the station	OPTIONAL

EXAMPLE:

```
=>wireless acllist
Station Name    MAC address         Allowed
Test            00:90:D0:01:47:DE : yes
=>wireless aclentry hwaddr=01:67:d0:23:56:cd action=allow name="Test2"
=>wireless acllist
Station Name    MAC address         Allowed
Test            00:90:D0:01:47:DE : yes
Test2           01:67:D0:23:56:CD : yes
=>wireless aclentry hwaddr=01:67:d0:23:56:cd action=deny name="Test2"
=>wireless acllist
Station Name    MAC address         Allowed
Test            00:90:D0:01:47:DE : yes
Test2           01:67:D0:23:56:CD : no
=>wireless aclentry hwaddr=01:67:d0:23:56:cd action=delete name="Test2"
=>wireless acllist
Station Name    MAC address         Allowed
Test            00:90:D0:01:47:DE : yes
=>
```

RELATED COMMANDS:

wireless acllist	Lists the ACL.
wireless aclflush	Flushes the ACL.

wireless ifconfig

Configures and shows the SpeedTouch™570 Wireless LAN (WLAN) interface parameters.

SYNTAX:

```
wireless ifconfig            [ssid = <quoted string>]
                             [any = <{off|on}>]
                             [channel = <{auto|1|2|3|4|5|6|7|8|9|10|11}>]
                             [rts = <number{0-2347}>]
```

ssid	The Service Set Identifier (SSID)	OPTIONAL
any	Accept stations with SSID 'any' or blank SSID. Default is no.	OPTIONAL
channel	The communication channel number. Choose between: • auto: the best communication channel is automatically selected by the SpeedTouch™. • a number between 1 and 11.	OPTIONAL
rts	A number between 0 and 2347. It represents the value of the RTS threshold. Setting this parameter to a small value causes RTS messages to be send more often, consuming more of the available bandwidth, therefore reducing the apparent throughput of other network packets. However, the more often RTS packets are sent, the quicker the system can recover from interference or collisions. Default is 2347.	OPTIONAL

EXAMPLE:

```
=>wireless ifconfig
Wireless SSID           : SpeedTouch3A6115
Accept any SSID         : no
Communication channel   : 6 (auto)
RTS Threshold           : 2347
Short Retry Limit       : 4
Long Retry Limit        : 4
Beacon Period           : 100
=>wireless ifconfig any=on channel=3 rts=1000
=>wireless ifconfig
Wireless SSID           : SpeedTouch3A6115
Accept any SSID         : yes
Communication channel   : 3 (selected)
RTS Threshold           : 1000
Short Retry Limit       : 4
Long Retry Limit        : 4
Beacon Period           : 100
=>
```

RELATED COMMANDS:

wireless status	Shows the status of the wireless interface.
wireless scan	Scans for other APs in the area.
wireless flush	Flushes the wireless parameters.

wireless status

Shows the status of the SpeedTouch™570 WLAN interface.

SYNTAX:

```
wireless status
```

EXAMPLE:

```
=>wireless status
Wireless interface is running on channel 3 in clear-text
=>wireless ifconfig channel=auto
=>wireless status
Wireless interface is running on channel 6 in clear-text
```

RELATED COMMANDS:

wireless ifconfig	Configures and shows the wireless interface parameters.
wireless scan	Scans for other APs in the area.
wireless flush	Flushes the wireless parameters.

wireless counters

Displays the value of the SpeedTouch™570 WLAN counters.

SYNTAX:

```
wireless counters            [clear <{no|yes}>]
```

clear	Clear the wireless counters	OPTIONAL

EXAMPLE:

```
=>wireless counters
Tx Frames Count             : 497
Rx Frames Count             : 0
Station Not Allowed Count   : 0
Station Unknown Count       : 0
WEPICV Error Count          : 0
WEP Excluded Count          : 0
=>wireless counters clear=yes
=>wireless counters
Tx Frames Count             : 0
Rx Frames Count             : 0
Station Not Allowed Count   : 0
Station Unknown Count       : 0
WEPICV Error Count          : 0
WEP Excluded Count          : 0
=>
```

wireless scan

Scans for other APs in the area. An overview of all the radio channels and their respective ratings is shown.

SYNTAX:

```
wireless scan
```

EXAMPLE:

```
=>wireless scan

Channel rating
 1 : Good
 2 : Good
 3 : Good
 4 : Good
 5 : Good
 6 : Recommended
 7 : Good
 8 : Good
 9 : Good
10 : Good
11 : Good

Recommended channel : 6

=>
```

RELATED COMMANDS:

wireless ifconfig	Configures and shows the wireless interface parameters.
wireless status	Shows the status of the wireless interface.
wireless flush	Flushes the wireless parameters.

wireless flush

Flush the SpeedTouch™570 WLAN parameters, i.e. reset to the factory defaults.

SYNTAX:

```
wireless flush
```

EXAMPLE:

```
=>wireless ifconfig
Wireless SSID           : SpeedTouch3A6115
Accept any SSID         : yes
Communication channel   : 3 (selected)
RTS Threshold           : 1000
Short Retry Limit       : 4
Long Retry Limit        : 4
Beacon Period           : 100
=>wireless flush
=>wireless ifconfig
Wireless SSID           : SpeedTouch3A6115
Accept any SSID         : no
Communication channel   : 6 (auto)
RTS Threshold           : 2347
Short Retry Limit       : 4
Long Retry Limit        : 4
Beacon Period           : 100
=>
```

RELATED COMMANDS:

wireless ifconfig	Configures and shows the wireless interface parameters.
wireless status	Shows the status of the wireless interface.
wireless scan	Scans for other APs in the area.

Abbreviations

The table below lists all the abbreviations used in the CLI Guide.

Abbreviation	Description
AAL5	ATM Adaption Layer 5
ACL	Access Control List
ADSL	Asymmetric Digital Subscriber Line
AH	Authentication Header
ARP	Address Resolution Protocol
ATM	Asynchronous Transfer Mode
ATMF	ATM Forum
BGP	Border Gateway Protocol
CA	Certificate Authority
CC	Continuity Check
CEP	Certificate Enrollment Protocol
CHAP	Challenge Handshake Authentication Protocol
CIDR	Classles Inter Domain Routing
CLI	Command Line Interface
CRL	Certificate Revocation List
DHCP	Dynamic Host Configuration Protocol
DN	Distinguished Name
DNS	Domain Name System
EGP	Exterior Gateway Protocol
ESP	Encapsulating Security Payload
GRE	General Routing Encapsulation
GRP	Gateway Routing Protocol
HDLC	High-level Data Link Control
HTTP	HyperText Transfer Protocol
ICMP	Internet Control Message Protocol
IGD	Internet Gateway Device
IGMP	Internet Group Management Protocol

Abbreviation	Description
IKE	Internet Key Exchange
IMAP	Interim Mail Access Protocol
IMAP	Interactive Mail Access Protocol
IP	Internet Protocol
IPCP	Internet Protocol Control Protocol
IPCP	IP Payload Compression Protocol
IPoA	IP over ATM
IPSec	IP Security
IRC	Internet Relay Chat
ISDN	Integrated Services Digital Network
LAN	Local Area Network
LCP	Link Control Protocol
LDAP	Light-weight Directory Access Protocol
LIS	Logical IP Subnet
LLC	Logical Link Control
MAC	Medium Access Control
MD5	Message Digest 5
MER	MAC Encapsulated Routing
NAPT	Network Address and Port Translation
NAT	Network Address Translation
NBP	Name Binding Protocol
NLPID	Network Layer Protocol IDentifiers
NNTP	Network News Transfer Protocol
NTP	Network Time Protocol
OAM	Operation and Maintenance
OBC	On Board Controller
OID	Object IDentifier
PAP	Password Authentication Protocol
PIP	Packet Interception Point
PKCS	Public Key Cryptography Standard
PKI	Public Key Infrastructure

Abbreviation	Description
POP	Post Office Protocol
POTS	Plain Old Telephone Service
PPP	Point-to-Point Protocol
PPPoA	PPP over ATM
PPPoE	PPP over Ethernet
PPTP	Point-to-Point Tunneling Protocol
PSD	Power Spectral Density
PVC	Permanent Virtual Channel
RIP	Routing Information Protocol
RTMP	RouTing Maintenance Protocol
RTSP	Real Time Stream Control Protocol
SAs	Security Associations
SHDSL	Symmetric High speed Digital Subscriber Line
SMTP	Simple Mail Transfer Protocol
SNAP	Sub Network Access Protocol
SNMP	Simple Network Management Protocol
SNPP	Simple Network Paging Protocol
SNTP	Simple Network Time Protocol
SVC	Switched Virtual Channel
TCP	Transmission Control Protocol
TFTP	Trivial File Transfer Protocol
UDP	User Datagram Protocol
UPnP	Universal Plug and Play
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
VC	Virtual Channel
VCMUX	Virtual Channel MUltipleXing
VDSL	Very high speed Digital Subscriber Line
VLSM	Variable Length Subnet Masking
VP	Virtual Path
VPN	Virtual Private Networking

Abbreviation	Description
WAN	Wide Area Network
WCD	WAN Connection Device
WEP	Wired Equivalent Privacy
WLAN	Wireless LAN
WWW	World Wide Web
ZIS	Zone Information System

Supported Key Names

Contents

This chapter lists all the key names supported by the SpeedTouch™, that can be used for completing CLI command parameters.

Supported Internet Protocol (IP) Protocol Names

For more information on the listed IP protocols, see RFC1340 or www.iana.org.

Protocol name	Number	Description
icmp	1	Internet Control Message Protocol (ICMP)
igmp	2	Internet Group Management Protocol (IGMP)
ipinip	4	IP in IP (encapsulation)
tcp	6	Transmission Control Protocol (TCP)
egp	8	Exterior Gateway Protocol (EGP)
udp	17	User Datagram Protocol (UDP)
rsvp	46	Reservation Protocol
gre	47	General Routing Encapsulation (GRE)
ah	51	Authentication Header (AH)
esp	50	Encapsulating Security Payload (ESP)
vines	83	Vines
ipcomp	108	IP Payload Compression Protocol (IPCP)

Supported TCP/UDP Port Names

For more information on the listed TCP/UDP port assignments, see RFC1340 or www.iana.org.

Port name	Number	TCP	UDP	Description
echo	7	Y	Y	Echo
discard	9	Y	Y	Discard
systat	11	Y	Y	Active Users
daytime	13	Y	Y	Daytime
qotd	17	Y	Y	Quote of the Day
chargen	19	Y	Y	Character Generator
ftp-data	20	Y	Y	File Transfer (Default data)
ftp	21	Y	Y	File Transfer (Control)
telnet	23	Y	Y	Telnet
smtp	25	Y	Y	Simple Mail Transfer Protocol (SMTP)
time	37	Y	Y	Time
nicname	43	Y	Y	Who Is
dns	53	Y	Y	Domain Name System (DNS)
domain	53	Y	Y	Domain Name System (DNS)
sql*net	66	Y	Y	Oracle SQL*NET
bootps	67	Y	Y	Bootstrap Protocol Server
bootpc	68	Y	Y	Bootstrap Protocol Client
tftp	69	Y	Y	Trivial File Transfer Protocol (TFTP)
gopher	70	Y	Y	Gopher
finger	79	Y	Y	Finger
www-http	80	Y	Y	World Wide Web (WWW) HTTP
kerberos	88	Y	Y	Kerberos
rtelnet	107	Y	Y	Remote Telnet Service
pop2	109	Y	Y	Post Office Protocol (POP) - Version 2
pop3	110	Y	Y	Post Office Protocol (POP) - Version 3
sunrpc	111	Y	Y	SUN Remote Procedure Call
auth	113	Y	Y	Authentication Service
sqlserver	118	Y	Y	SQL Services

Port name	Number	TCP	UDP	Description
nntp	119	Y	Y	Network News Transfer Protocol (NNTP)
sntp	123	Y	Y	Simple Network Time Protocol (SNTP)
ntp	123	Y	Y	Network Time Protocol (NTP)
ingres-net	134	Y	Y	INGRES-NET Service
netbios-ns	137	Y	Y	NETBIOS Naming System
netbios-dgm	138	Y	Y	NETBIOS Datagram Service
netbios-ssn	139	Y	Y	NETBIOS Session Service
imap2	143	Y	Y	Interim Mail Access Protocol (IMAP) v2
sql-net	150	Y	Y	SQL-NET
pcmail-srv	158	Y	Y	PCMail Server
snmp	161	Y	Y	Simple Network Management Protocol (SNMP)
snmptrap	162	Y	Y	SNMP Trap
bgp	179	Y	Y	Border Gateway Protocol (BGP)
irc-o	194	Y	Y	Internet Relay Chat (IRC) - o
at-rtmp	201	Y	Y	AppleTalk RouTing Maintenance Protocol (RTMP)
at-nbp	202	Y	Y	AppleTalk Name Binding Protocol (NBP)
at-echo	204	Y	Y	AppleTalk Echo
at-zis	206	Y	Y	AppleTalk Zone Information System (ZIS)
ipx	213	Y	Y	
imap3	220	Y	Y	Interactive Mail Access Protocol (IMAP) v3
clearcase	371	Y	Y	ClearCase
ulistserv	372	Y	Y	UNIX Listserv
ldap	389	Y	Y	Light-weight Directory Access Protocol (LDAP)
netware-ip	396	Y	Y	Novell Netware over IP
snpp	444	Y	Y	Simple Network Paging Protocol (SNPP)
ike	500	Y	Y	ISAKMP
exec	512	Y	-	Remote process execution
biff	512	-	Y	Used by mail system to notify users of new mail received
login	513	Y	-	Remote login a la telnet
who	513	-	Y	Maintains data bases showing who's logged in to machines on a local net and the load average of the machine

Port name	Number	TCP	UDP	Description
syslog	514	-	Y	Syslog
printer	515	Y	Y	Spooler
talk	517	Y	Y	Like Tenex link, but across machine
ntalk	518	Y	Y	NTalk
utime	519	Y	Y	UNIX Time
rip	520	-	Y	Local routing process (on site); uses variant of Xerox NS Routing Information Protocol (RIP)
timed	525	Y	Y	Timeserver
netwall	533	Y	Y	For emergency broadcasts
uucp	540	Y	Y	uucpd
uucp-rlogin	540	Y	Y	uucpd remote login
new-rwho	540	Y	Y	uucpd remote who is
rtsp	554	Y	Y	Real Time Stream Control Protocol (RTSP)

Supported ICMP Type Names

For more information on the listed ICMP type names, see RFC1340 or www.iana.org.

ICMP Type name	Number	Description
echo-reply	0	Echo Reply
destination-unreachable	3	Destination Unreachable
source-quench	4	Source Quench
redirect	5	Redirect
echo-request	8	Echo
router-advertisement	9	Router Advertisement
router-solicitation	10	Router Solicitation
time-exceeded	11	Time Exceeded
parameter-problems	12	Parameter problems
timestamp-request	13	Timestamp
timestamp-reply	14	Timestamp Reply
information-request	15	Information Request
information-reply	16	Information Reply
address-mask-request	17	Address Mask Request
address-mask-reply	18	Address Mask Reply

www.speedtouch.com

500
SERIES

Built for excellence

speedtouch™
ALCATEL