

speedtouch™

SpeedTouch™

516/536/546/510_{v5}/530_{v5}

Multi-User ADSL Gateways



CLI Reference Guide

Release R5.2.7



SpeedTouch™

516/536/546

510_{v5}/530_{v5}

CLI Reference Guide

R5.2.7

Copyright

Copyright ©1999-2004 THOMSON. All rights reserved.

Passing on, and copying of this document, use and communication of its contents is not permitted without written authorization from THOMSON. The content of this document is furnished for informational use only, may be subject to change without notice, and should not be construed as a commitment by THOMSON. THOMSON assumes no responsibility or liability for any errors or inaccuracies that may appear in this document.

Thomson Telecom Belgium
Prins Boudewijnlaan, 47
B-2650 Edegem
Belgium

www.speedtouch.com

Trademarks

The following trademarks are used in this document:

- ▶ SpeedTouch™ is a trademark of THOMSON.
- ▶ Microsoft, MS-DOS, Windows, Windows NT and Internet Explorer are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
- ▶ UNIX is a registered trademark of UNIX System Laboratories, Incorporated.
- ▶ Apple and Mac OS are trademarks of Apple Computer, Incorporated, registered in the United States and other countries.
- ▶ Adobe, the Adobe logo, Acrobat and Acrobat Reader are trademarks or registered trademarks of Adobe Systems, Incorporated, registered in the United States and/or other countries.
- ▶ Netscape and Netscape Navigator are registered trademarks of Netscape Communications Corporation.
- ▶ Ethernet is a trademark of Xerox Corporation.
- ▶ "Wi-Fi" and the Wi-Fi logo are registered trademarks of the Wi-Fi Alliance. "Wi-Fi CERTIFIED", "Wi-Fi ZONE", "Wi-Fi Alliance", their respective logos and "Wi-Fi Protected Access" are trademarks of the Wi-Fi Alliance.

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

Document Information

Status: v1.0 (October 2004)

Reference: E-DOC-CTC-20040907-0010

Short Title: CLI Reference Guide ST516/536/546/510v5/530v5 R5.2.7

Contents

About this Document	15
CLI Navigation.....	17
Accessing the Command Line Interface.....	18
Basic Navigation and Manipulation.....	19
Command Line Interface Commands.....	22
Menu-driven CLI Navigation	23
Direct FTP Access.....	27
ADSL Commands.....	29
adsl config.....	30
adsl info.....	31
ATM Commands	33
atm oam cconfig.....	34
atm oam cclist	35
atm oam ccsend.....	36
atm oam config	37
atm oam mode	38
atm oam ping	39
atm oam status.....	40
AutoPVC Commands.....	41
autopvc config.....	42
autopvc info.....	44
Bridge Commands	45
bridge config.....	46
bridge flush.....	47
bridge ifadd.....	48
bridge ifattach.....	49
bridge ifconfig	50
bridge ifdelete.....	52
bridge ifdetach.....	53
bridge iflist.....	54
bridge macadd.....	55
bridge macdelete.....	56
bridge maclist.....	57

CIP Commands 59

- cip flush 60
- cip ifadd 61
- cip ifdelete 62
- cip iflist 63
- cip pvcadd 64
- cip pvcdelete 65
- cip pvclist 66

Config Commands 67

- config backup 68
- config dump 69
- config erase 70
- config flush 71
- config list 72
- config load 73
- config save 74

DHCP Commands..... 75

- dhcp client clear 76
- dhcp client config 77
- dhcp client flush 78
- dhcp client ifadd 79
- dhcp client ifattach 80
- dhcp client ifconfig 81
- dhcp client ifdelete 83
- dhcp client iflist 84
- dhcp client ifrelease 85
- dhcp client ifrenew 87
- dhcp client stats 89
- dhcp relay add 90
- dhcp relay config 91
- dhcp relay delete 92
- dhcp relay flush 93
- dhcp relay ifconfig 94
- dhcp relay iflist 95
- dhcp relay list 96
- dhcp relay stats 97
- dhcp server clear 98
- dhcp server config 99
- dhcp server flush 100
- dhcp server policy 101
- dhcp server stats 102
- dhcp server lease add 104
- dhcp server lease delete 105
- dhcp server lease flush 106
- dhcp server lease list 107
- dhcp server pool add 108
- dhcp server pool config 109
- dhcp server pool delete 111
- dhcp server pool flush 112
- dhcp server pool list 113

DNS Commands.....	115
dns add.....	116
dns clear.....	117
dns clrstats.....	118
dns delete.....	119
dns domain.....	120
dns flush.....	121
dns fwdadd.....	122
dns fwddelete.....	123
dns fwdlist.....	124
dns fwdtable.....	125
dns list.....	126
dns nslookup.....	127
dns start.....	128
dns stats.....	129
dns status.....	130
dns stop.....	131
dns toutfwd.....	132
dns troff.....	133
dns tron.....	134
Env Commands.....	135
env flush.....	136
env get.....	137
env list.....	138
env set.....	139
env unset.....	140
Eth Commands.....	141
eth config.....	142
eth ifconfig.....	143
eth iflist.....	144
ETHoA Commands.....	145
ethoa flush.....	146
ethoa ifadd.....	147
ethoa ifattach.....	148
ethoa ifconfig.....	149
ethoa ifdelete.....	150
ethoa ifdetach.....	151
ethoa iflist.....	152

Firewall Commands..... 153

- firewall assign..... 154
- firewall flush 155
- firewall list..... 156
- firewall troff 157
- firewall tron..... 158
- firewall unassign 159
- firewall chain create..... 160
- firewall chain delete..... 161
- firewall chain flush..... 162
- firewall chain list 163
- firewall rule clear..... 164
- firewall rule create..... 165
- firewall rule delete..... 168
- firewall rule flush..... 169
- firewall rule list..... 170
- firewall rule stats..... 171

IP Commands 173

ip apadd	174
ip apdelete	176
ip aplist.....	177
ip arpadd.....	178
ip arpdelete.....	179
ip arplist	180
ip config	181
ip flush.....	184
ip ifconfig	185
ip iflist.....	186
ip ifwait.....	187
ip mcadd	188
ip mcdelete	189
ip mclist.....	190
ip ping	191
ip rtadd.....	192
ip rtdelete.....	193
ip rtlist	194
ip sendto	195
ip traceroute.....	196
ip auto flush.....	198
ip auto ifadd.....	199
ip auto ifattach.....	200
ip auto ifconfig.....	201
ip auto ifdelete.....	202
ip auto ifdetach.....	203
ip auto iflist	204

IPoA Commands 205

ipoa flush	206
ipoa ifadd	207
ipoa ifattach	208
ipoa ifconfig.....	209
ipoa ifdelete	210
ipoa ifdetach	211
ipoa iflist.....	212

IPQoS Commands.....	213
ipqos config	214
ipqos list.....	216
ipqos queue clear	217
ipqos queue config.....	218
ipqos queue list	220
ipqos queue stats	221
Label Commands	223
label add.....	224
label config	225
label delete.....	227
label flush.....	228
label list.....	229
label troff.....	230
label tron	231
label chain create	232
label chain delete	233
label chain flush	234
label chain list.....	235
label rule clear	236
label rule create	237
label rule delete.....	239
label rule flush.....	240
label rule list	241
label rule stats	242
Language Commands.....	243
language config.....	244
language list	245
language remove	246

NAT Commands	247
nat applist	248
nat bind	249
nat bindlist	250
nat clear.....	251
nat config	252
nat create	253
nat defserver.....	254
nat delete	255
nat disable.....	256
nat enable	257
nat flush	258
nat list.....	259
nat multinatadd	260
nat multinatdelete	261
nat multinatlist.....	262
nat unbind.....	263
Phonebook Commands	265
phonebook add.....	266
phonebook delete.....	267
phonebook flush.....	268
phonebook list.....	269
PPPoA Commands	271
pppoa flush	272
pppoa ifadd.....	273
pppoa ifattach	274
pppoa ifconfig.....	275
pppoa ifdelete	279
pppoa ifdetach.....	280
pppoa iflist	281
pppoa rtadd	282
pppoa rtdelete	284

PPPoE Commands	285
pppoe flush	286
pppoe ifadd	287
pppoe ifattach	288
pppoe ifconfig	289
pppoe ifdelete	293
pppoe ifdetach	294
pppoe iflist	295
pppoe ifscan	296
pppoe rtadd	297
pppoe rtdelete	299
pppoe relay add	300
pppoe relay delete	301
pppoe relay flush	302
pppoe relay portlist	303
pppoe relay sesslist	304
 PPTP Commands	 305
pptp flush	306
pptp ifadd	307
pptp list	308
pptp profadd	309
pptp profdelete	310
pptp proflist	311
 QoSBook Commands	 313
qosbook add	314
qosbook config	316
qosbook delete	317
qosbook flush	318
qosbook list	319
 Script Commands	 321
script add	322
script delete	323
script flush	324
script list	325
script run	326

SNMP Commands.....	327
snmp config.....	328
snmp get.....	329
snmp getNext.....	330
snmp list.....	331
snmp walk.....	332
Software Commands.....	333
software version.....	334
software upgrade.....	335
Switch Commands	337
switch group flush.....	338
switch group list.....	339
switch group move	340
switch mirror capture.....	341
switch mirror egress	342
switch mirror ingress.....	343
System Commands	345
system clearpassword	346
system config	347
system flush.....	348
system reboot.....	349
system reset.....	350
system setpassword	351
system stats.....	352
Systemlog Commands	353
systemlog flush	354
systemlog show	355
systemlog send	356
TD Commands.....	357
td call.....	358
UPnP Commands	359
upnp config.....	360
upnp flush.....	361
upnp list	362

Abbreviations 363

Syslog Messages 367

- Auto-PVC Module..... 368
- Configuration Module..... 368
- DHCP CLient Module..... 368
- DHCP Relay Module..... 369
- DHCP Server Module 369
- Firewall Module..... 370
- HTTP Module..... 370
- Kernel Module 370
- Linestate Module..... 370
- Login Module..... 371
- NAPT Module 371
- PPP Module..... 371
- PPTP Module 372
- Routing Module..... 372
- Software Module..... 372
- UPnP Module..... 372

Supported Key Names 373

- Supported IP Protocol Names..... 374
- Supported TCP/UDP Port Names..... 375
- Supported ICMP Type Names..... 378
- Supported Syslog Facilities..... 379
- Supported Syslog Severities..... 380

About this Document

Introduction

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

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

For the fastidious user, this Reference Guide aims to be a handbook for advanced and detailed configuration and troubleshooting of the SpeedTouch™ via its character based CLI command set.

Applicability

This CLI Reference Guide covers the CLI commands of the following Digital Subscriber Line (DSL) SpeedTouch™ products:

- ▶ SpeedTouch™516(i) Multi-User ADSL Gateway (R5.2.7)
- ▶ SpeedTouch™536(i) Multi-User ADSL Gateway (R5.2.7)
- ▶ SpeedTouch™546(i) Multi-User ADSL Gateway (R5.2.7)
- ▶ SpeedTouch™510v5(i) Multi-User ADSL Gateway (R5.2.7)
- ▶ SpeedTouch™530v5(i) Multi-User ADSL Gateway (R5.2.7)



For readability, all are referred to as SpeedTouch™ throughout this document unless a specific variant is concerned.

Contents

The Reference Guide consists of two main parts:

- ▶ CLI Navigation:

This chapter familiarizes the user with the SpeedTouch™ CLI. It describes general manipulations to navigate through the CLI and to perform some operations on the CLI.

- ▶ CLI Command Description:

The other chapters describe all the available CLI commands of the SpeedTouch™ 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 necessary completed by a description of the possible impact on the user and/or the SpeedTouch™.
- ▶ 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 MacOS 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.

Symbols

The following symbols are used in this CLI Reference Guide:



A **note** provides additional information about a topic.



A **tip** provides an alternative method or shortcut to perform an action.



A **caution** warns you about potential problems or specific precautions that need to be taken.

Conventions

The following conventions are used to present the CLI commands:

- ▶ Vertical bars (|) separate alternative, mutually exclusive, elements.
- ▶ Square brackets [] indicate optional elements.
- ▶ Braces {} indicate a required choice.
- ▶ Braces within square brackets [{}] indicate a required choice within an optional element.
- ▶ <string> indicates that an alphanumerical string without blanks must be used.
- ▶ <quoted string> indicates that an alphanumerical string with blanks can be used. The string must be enclosed in quotes " ".
- ▶ Interactive input is shown in a **bold font**.
- ▶ Interactive output is shown in **this font**.
- ▶ Comments are added *in italics*. Example:

```
=>language list
CODE LANGUAGE VERSION FILENAME
en* english 4.2.0.1 <system> In this example only one language is available
```

Documentation and software updates

THOMSON continuously develops new solutions, but is also committed to improve its existing products.

For more information on THOMSON's latest technological innovations, documents and software releases, visit us at:

www.speedtouch.com

CLI Navigation

Introduction

This chapter familiarizes the user with the SpeedTouch™ CLI. It describes general manipulations:

- ▶ to navigate through the CLI
- ▶ to perform some operations on the CLI.

Contents

This chapter covers the following topics:

Accessing the Command Line Interface	18
Basic Navigation and Manipulation	19
Command Line Interface Commands	22
Menu-driven CLI Navigation	23
Direct FTP Access	27

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 and so on.

Command group navigation

From top level, you can change to a command group by executing the name of the desired command group. The example below shows every possible CLI command group for the SpeedTouch™516.

```

=>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          autopvc         bridge          cip
config        dhcp          dns             env             eth
ethoa         firewall  ip             ipoa           ipqos
label         language  nat           phonebook      pppoa
pppoe         pptp      qosbook       script         snmp
software      system   systemlog     td             upnp

=>

```

The following command groups are specific for certain variants:

- ▶ **switch** is only available for the SpeedTouch™546.

To return to top level, or to go up one level (in case of nested command groups), enter `..` at the prompt.

Example:

```

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

```

The Help Command

Execute **help** or **?** from top level to list all available commands and command groups for the SpeedTouch™.

Example for the SpeedTouch™516:

```

=>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          autopvc         bridge          cip
config        dhcp          dns             env             eth
ethoa         firewall       ip             ipoa           ipqos
label         language       nat            phonebook       pppoa
pppoe         pptp          qosbook        script          snmp
software      system        systemlog      td             upnp

=>

```

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 :

assign        : Assign a chain to an entry point.
unassign      : Clear a specific hook
list          : Shows a list of all the hooks with the chains attached.
flush         : Clears all hooks, chains and rules
tron          : Enables verbose console messaging.
troff         : Disables verbose console messaging.

Following command groups are available :

chain         rule

[firewall]=>

```

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

Executing for example **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 :

assign        : Assign a chain to an entry point.
unassign      : Clear a specific hook
list          : Shows a list of all the hooks with the chains attached.
flush         : Clears all hooks, chains and rules
tron          : Enables verbose console messaging.
troff         : Disables verbose console messaging.

Following command groups are available :

chain         rule

=>

```

Entering **help** followed by a specific command, for example **:help firewall assign** (starting from top level) or **:help assign** (executed from 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.
=>
```

Entering **: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 command groups, for commands and for options, but not for values.

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.

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

Command Line Navigation

- ▶ Press CTRL+A to go to the beginning of the command line.
- ▶ Press CTRL+L to go to the end of the command line.

Breaking off Commands

Press CTRL+G to quit a command. This can be useful in a situation where a user is prompted to enter a value which he does not know and he wants to quit the command.

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

```
[firewall]=>match
chain =
chain =
chain = "CTRL+G"
[firewall]=>
```

History of 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 Commands

Command Execution

All CLI commands are commands that operate on, or configure, the SpeedTouch™ settings.

The CLI commands can be executed:

- ▶ from top level, preceded by the name of the command group from which the command should be executed. For example **:firewall list**:

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

- ▶ from within the command group itself, using the reduced form of the command. For example **:list** at the **firewall** command group selection.

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

Entering a CLI Command

A CLI command can be entered in one of the following ways:

- 1 As a completely built-up command with all its parameters.
For example the command **:firewall assign hook=input chain=input**.

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

- 2 Just the command itself without its parameters. After pressing *Enter*, you are prompted to complete the command with the required and the optional parameters.

The example below is the equivalent of the command **:firewall assign hook=input chain=input**.

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

- ▶ For required CLI command parameter values, either enter a value or scroll through the possible values with the arrow keys.
- ▶ For optional CLI command parameters, either enter a value or simply press *Enter* without giving a value.



This manner of entering a command is not applicable to all the CLI commands. Examples are **:system config**, **:upnp config**, ...

Menu-driven CLI Navigation

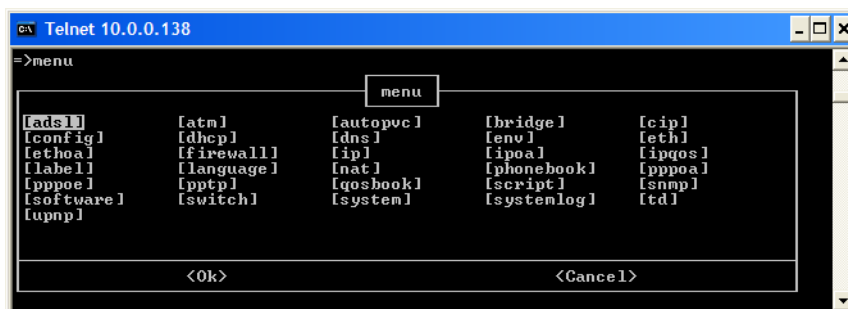
Introduction

To improve the user-friendliness of the SpeedTouch™ CLI, the CLI features a menu-driven interface. The semi-graphical menu offers an attractive and easy-to-use configuration environment for the CLI.

Calling the menu

To call the menu-driven interface, type *menu* at the CLI prompt.

Example (for the SpeedTouch™546):



Leaving the menu

To leave the menu-driven interface, use the TAB key to go to the *Cancel* field and press ENTER.



This can be done on any level of the menu, in other words there is no need to return to the top level.

Navigating through the menu

Follow the procedure below to navigate through the menu and select CLI commands:

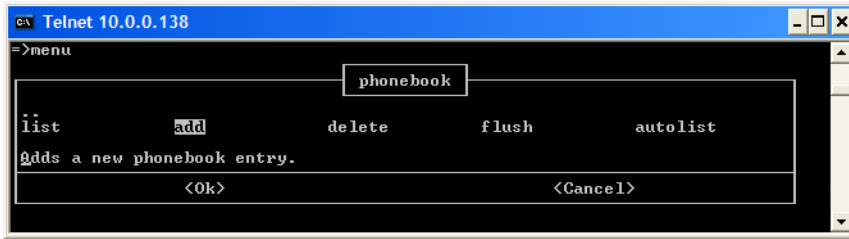
- 1 Browse through the CLI command groups with the ARROW keys:



- 2 Press ENTER to select a command group. The figure below shows the menu after the *phonebook* command group has been selected:



- Browse through the CLI commands with the ARROW keys or select .. to go back up one level:



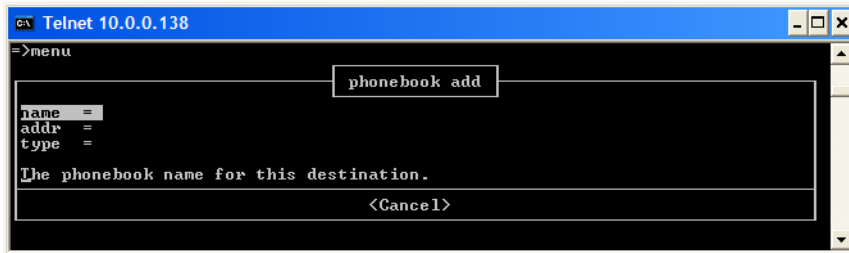
- Press ENTER to select the command. The figure below shows the menu after the **add** command has been selected:



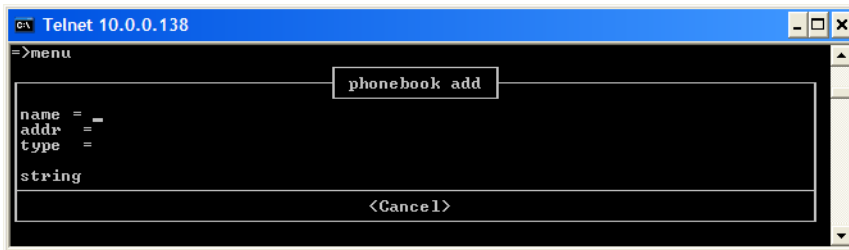
Entering parameter values

Proceed as follows to enter parameter values:

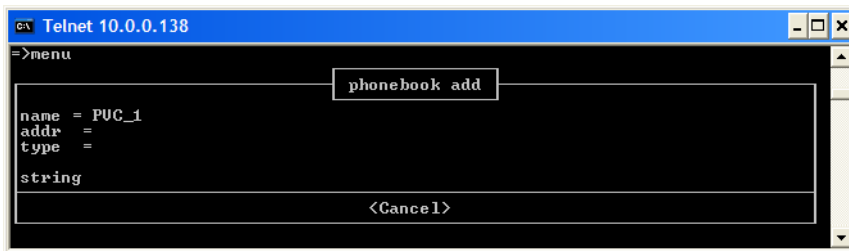
- Browse through the parameters with the ARROW keys:



- To enter a value for a parameter, press ENTER:



- Fill in the parameter value and press ENTER:



- 4 Use the ARROW keys to scroll to the next parameter:

```

c:\ Telnet 10.0.0.138
=>menu
      phonebook add
name := PUC_1
addr :=
type =
The address for this destination.
      <Cancel>
    
```

- 5 When all the necessary parameters have been entered, use the TAB key to go to the OK field and press ENTER.

```

c:\ Telnet 10.0.0.138
=>menu
      phonebook add
name := PUC_1
addr := 8*35
type := any
The protocol/service supported by this destination.
      <Ok>          <Cancel>
    
```



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

Preset parameter values

In case preset values are provided for a parameter, they are shown at the bottom of the menu.

```

c:\ Telnet 10.0.0.138
=>menu
      phonebook add
name := PUC_1
addr := 8*35
type =
<any|ethoai|pppoai|ipoa>
      <Cancel>
    
```

You can scroll through these preset values with the ARROW keys or enter the value manually.

```

c:\ Telnet 10.0.0.138
=>menu
      phonebook add
name := PUC_1
addr := 8*35
type = any
<any|ethoai|pppoai|ipoa>
      <Cancel>
    
```

CLI Commands in Service Template Files

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

In Service Template 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™. The declaration of such customization variables must be done in the [env.ini] paragraph of the Service Template file. A preset (i.e. default) value can be declared for a customization variable. Further use of these customization variables is allowed through all other paragraphs, even several times.

When a customization variable is used in a CLI command, the value of the variable must always conform to the syntax of the CLI command.

Customizing Service Template Files

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

www.speedtouch.com

Direct FTP Access

The SpeedTouch™ File System

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

The file system of the SpeedTouch™ is accessible via the File Transfer Protocol (FTP) transport protocol. This allows to transfer the SpeedTouch™ software image(s) and/or Service Template files and default settings files.

Proceed as indicated in the example below to open an FTP session to the SpeedTouch™ 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™ 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:

- ▶ all the necessary files for the SpeedTouch™ to boot correctly.
- ▶ the **active** subdirectory always contains the software image in execution, in other words the active software image.
- ▶ the **dl** directory contains the dormant software image (in other words, the passive software image). If you have made changes to the SpeedTouch™ configuration and saved them (be it via a Telnet session, via the web pages or via the Setup wizard), a user.ini Service Template file is created in the **dl** subdirectory.

In other words, after each **:saveall** or **:config save all**, the user.ini Service Template file present in the **dl** subdirectory reflects the current configuration of the SpeedTouch™.

SpeedTouch™ File System Access Rights

From the **root** directory, you can access the **active** and **dl** subdirectories.



No read/write permission is granted for the **root** directory.

The following access rights apply for the subdirectories:

- ▶ for the **active** subdirectory:
 - ▶ Listing of **active** subdirectory files (dir)
 - ▶ FTP (m)get of (multiple) **active** subdirectory files
- ▶ for the **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”.



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>
```

Quote Site Command

All the CLI commands can be executed from within an FTP session.

Only complete CLI commands (in other words, the complete command syntax with all the parameters already specified) can be executed.

Example:

To execute the command **:firewall list**, type the following at the FTP prompt:

```
ftp> quote site firewall list
200- :firewall assign hook=input chain=None
200- :firewall assign hook=sink chain=sink
200- :firewall assign hook=forward chain=forward
200- :firewall assign hook=source chain=source
200- :firewall assign hook=output chain=None
200-
200 CLI command "firewall list" executed
ftp>
```

ADSL Commands

Introduction

This chapter describes the commands of the **adsl** command group.

Contents

This chapter covers the following commands:

adsl config	Show/set the Asymmetric Digital Subscriber Line (ADSL) configuration.	30
adsl info	Show ADSL statistics and information about the SpeedTouch™ DSL line status.	31

adsl config

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

Although the command is the same for both SpeedTouch™ ADSL/Plain Old Telephone System (POTS) and SpeedTouch™ ADSL/Integrated Services Digital Network (ISDN) variants, the command features specific parameter values per variant:

SYNTAX for ADSL/POTS variants:

```
adsl config [opermode = <{multimode | multi_adsl2 | multi_readsl2
                    | multi_adsl2plus}>]
            [trace = <{off | on}>]
```

where:

opermode	The operational mode of the SpeedTouch™ modem. Choose between: <ul style="list-style-type: none"> ▶ multimode ▶ multi_adsl2 ▶ multi_readsl2 ▶ multi_adsl2plus. The default is multi_adsl2plus .	OPTIONAL
trace	Enable (on) / disable (off) ADSL tracing. The default is on .	OPTIONAL

SYNTAX for ADSL/ISDN variants:

```
adsl config [opermode = <{multimode | multi_adsl2 | multi_readsl2
                    | multi_adsl2plus}>]
            [trace = <{off | on}>]
```

where:

opermode	The operational mode of the SpeedTouch™ modem. Choose between: <ul style="list-style-type: none"> ▶ multimode ▶ multi_adsl2 ▶ multi_readsl2 ▶ multi_adsl2plus. The default is multi_adsl2plus .	OPTIONAL
trace	Enable (on) / disable (off) ADSL tracing. The default is on .	OPTIONAL

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

```
=>adsl config
ADSL configuration:
    opermode = multi_adsl2plus
    trace = on
    modemoption = 00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00
=>
```


adsl info

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

Although the same command is used for both SpeedTouch™ ADSL/POTS and SpeedTouch™ ADSL/ISDN variants, the command features specific output parameters and counters per variant.

SYNTAX:

```
adsl info
```

EXAMPLE (for a SpeedTouch™ 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

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
```


ATM Commands

Introduction

This chapter describes the commands of the **atm** command group.

Contents

This chapter covers the following commands:

<code>atm oam ccconfig</code>	Configure Operation and Maintenance (OAM) Continuity Check (CC) on the connection.	34
<code>atm oam cclist</code>	Show the current OAM CC configuration.	35
<code>atm oam ccsend</code>	Send CC activate/deactivate to the connection.	36
<code>atm oam config</code>	Configure the OAM settings.	37
<code>atm oam mode</code>	Configure the OAM data blocking mode.	38
<code>atm oam ping</code>	Send ATM loopback cells.	39
<code>atm oam status</code>	Show the OAM settings.	40

atm oam ccconfig

Configure Operation and Maintenance (OAM) Continuity Check (CC) on the connection.

SYNTAX:

```
atm oam ccconfig port = <{dsl0 | dsl1 | atm2 | atm3 | aal5 | atm5} or number>
                vpi = <number{0-15}>
                [vci = <number{0-511}>]
                [transmit = <{disabled | enabled}>]
                [receive = <{disabled | enabled}>]
                [auto = <{disabled | enabled}>]
                [span = <{segment | end2end}>]
```

where:

port	The Asynchronous Transfer Mode (ATM) port number. Choose between: <ul style="list-style-type: none"> ▶ DSL0 ▶ DSL1 ▶ ATM2 ▶ ATM3 ▶ AAL5 ▶ ATM5. Or specify a port number (dsl0 has port number 0, ..., atm5 has port number 5).	REQUIRED
vpi	A number between 0 and 15. Represents the Virtual Path Identifier (VPI)	REQUIRED
vci	A number between 0 and 511. Represents the Virtual Channel identifier (VCI). For a VP cross-connection, use VCI=0 or do not specify.	OPTIONAL
transmit	Enable or disable transmission of CC cells. The default is disabled .	OPTIONAL
receive	Enable or disable loss of continuity. The default is disabled .	OPTIONAL
auto	Enable or disable remote CC activation and deactivation. The default is disabled .	OPTIONAL
span	Select either: <ul style="list-style-type: none"> ▶ end2end: monitoring occurs on the entire VC between two ATM end stations. ▶ segment: monitoring occurs on a VC segment between the SpeedTouch™ and a first-hop ATM switch. 	OPTIONAL

RELATED COMMANDS:

atm oam cclist Show the current OAM CC configuration.

atm oam cssend Send CC activate/deactivate to the connection.

atm oam cclist

Show the current OAM CC configuration.

SYNTAX:

```
atm oam cclist
```

EXAMPLE:

```
=>atm oam cclist
PORT = 0 VPI = 15 VCI = 64 End2End Mode = Auto Segment Mode = Auto
PORT = 0 VPI = 15 VCI = 16 End2End Mode = Auto Segment Mode = Auto
PORT = 0 VPI = 0 VCI = 16 End2End Mode = Auto Segment Mode = Auto
=>
```

RELATED COMMANDS:

- atm oam ccconfig** Configure Operation and Maintenance (OAM) Continuity Check (CC) on the connection.
- atm oam ccsend** Send CC activate/deactivate to the connection.

atm oam ccsend

Send CC activate/deactivate to the connection.

SYNTAX:

```
atm oam ccsend port = <{dsl0 | dsl1 | atm2 | atm3 | aal5 | atm5} or number>
                vpi = <number{0-15}>
                [vci = <number{0-511}>]
                [span = <{segment | end2end}>]
                [action = <{activate | deactivate}>]
                [direction = <{source | sink | both}>]
```

where:

port	The ATM port number. Choose between: <ul style="list-style-type: none"> ▶ DSL0 ▶ DSL1 ▶ ATM2 ▶ ATM3 ▶ AAL5 ▶ ATM5. Or specify a port number (dsl0 has port number 0, ..., atm5 has port number 5).	REQUIRED
vpi	A number between 0 and 15. Represents the VPI.	REQUIRED
vci	A number between 0 and 511. Represents the VCI. For a VP cross-connection, use VCI=0 or do not specify.	OPTIONAL
span	Send CC action. Choose between: <ul style="list-style-type: none"> ▶ end2end ▶ segment. 	OPTIONAL
action	Enable or disable CC. The default is disabled .	OPTIONAL
direction	Indicates the direction of CC activity. Select either: <ul style="list-style-type: none"> ▶ source ▶ sink ▶ both. Note If not specified, CC is activated/deactivated for both directions.	OPTIONAL

RELATED COMMANDS:

- atm oam ccconfig** Configure Operation and Maintenance (OAM) Continuity Check (CC) on the connection.
- atm oam cclist** Show the current OAM CC configuration.

atm oam config

Configure the OAM settings.

SYNTAX:

```
atm oam config [clp = <number{0-1}>]
               [loopbackid = <string>]
```

where:

clp	The Cell Loss Priority (CLP) bit value of the OAM cells. Choose between: ▶ 0 ▶ 1. The default is 1 .	OPTIONAL
loopbackid	The loopback ID(hexadecimal string) for processing of segment loopback cells. The default is 6a 6a 6a 6a 6a 6a 6a 6a 6a 6a 6a 6a 6a 6a 6a .	OPTIONAL

EXAMPLE:

```
=>atm oam status
   OAM config dump
-----
      CLP bit value : 1
      Loopback id   : 6a 6a 6a 6a 6a 6a 6a 6a 6a 6a 6a 6a 6a 6a 6a

   OAM data blocking mode
-----
      Port dsl0: blocking
      Port dsl1: blocking
      Port atm2: blocking
      Port atm3: blocking
      Port aal5: blocking
      Port atm5: blocking
=>
```

atm oam mode

Configure the OAM data blocking mode.

SYNTAX:

```
atm oam mode port = <{dsl0 | dsl1 | atm2 | atm3 | aal5 | atm5} or number>  
blocking = <{disabled | enabled}>
```

where:

port	The port for which OAM blocking is configured. Choose between: <ul style="list-style-type: none">▶ DSL0▶ DSL1▶ ATM2▶ ATM3▶ AAL5▶ ATM5 Or specify a port number (dsl0 has port number 0, ..., atm5 has port number 5).	REQUIRED
blocking	Enable or disable the OAM data blocking mode on this port. The default is <i>enabled</i> .	REQUIRED

atm oam ping

Send ATM loopback cells.

SYNTAX:

```
atm oam ping dest = <string>
                [count = <number{1-1000000}>]
                [interval = <number{100-1000000}>]
```

where:

dest	The destination address for the request. The destination can be any phonebook entry.	REQUIRED
count	A number between 1 and 1000000. Represents the number of pings to send.	OPTIONAL
interval	A number between 100 and 1000000 (milliseconds). Represents the interval between packets.	OPTIONAL

EXAMPLE:

```
=>atm oam ping dest=Sascha count=10 interval=200
loopback: successful, sequence: 1 time: 7762 usec
loopback: successful, sequence: 2 time: 8239 usec
loopback: successful, sequence: 3 time: 11100 usec
loopback: successful, sequence: 4 time: 9384 usec
loopback: successful, sequence: 5 time: 7209 usec
loopback: successful, sequence: 6 time: 20008 usec
loopback: successful, sequence: 7 time: 9651 usec
loopback: successful, sequence: 8 time: 9593 usec
loopback: successful, sequence: 9 time: 8411 usec
loopback: successful, sequence: 10 time: 41656 usec

--- loopback statistics ---
10 loopbacks transmitted, 10 successful, 0% loss, time 600 ms
rtt min/avg/max = 7209/13301/41656
=>
```

atm oam status

Show the OAM settings.

SYNTAX:

```
atm oam status
```

EXAMPLE:

```
=>atm oam status
OAM config dump
-----
      CLP bit value : 1
      Loopback id   : 6a 6a 6a 6a 6a 6a 6a 6a 6a 6a 6a 6a 6a 6a 6a
OAM data blocking mode
-----
      Port dsl0: blocking
      Port dsl1: blocking
      Port atm2: blocking
      Port atm3: blocking
      Port aal5: blocking
      Port atm5: blocking
=>
```

AutoPVC Commands

Introduction

This chapter describes the commands of the **autopvc** command group.

Contents

This chapter covers the following commands:

<code>autopvc config</code>	Configure autopvc.	42
<code>autopvc info</code>	Show retrieved information.	44

autopvc config

Configure autopvc.

SYNTAX:

```
autopvc config [mode = <{pseudo | passive | active}>]
               [type = <{bridge | pppoerelay}>]
               [opmode = <{partial | full}>]
               [overwrite = <{disabled | enabled}>]
               [peakrate = <number{0-27786}>]
```

where:

mode	Select the autopvc mode:	OPTIONAL
	<ul style="list-style-type: none"> ▶ pseudo: enable only pseudo-ILMI (VP/VC 15/16). When connection parameters are written to the MIB, display this information on CLI or web-interface but do not use these parameters for configuration. ▶ passive: enable both ILMI (VP/VC 0/16) and pseudo-ILMI (VP/VC 15/16). When connection parameters are written to the MIB, display this information on CLI or web-interface but do not use these parameters for configuration. ▶ active: enable both ILMI (VP/VC 0/16) and pseudo-ILMI (VP/VC 15/16). When connection parameters are written to the MIB, use these parameters to configure phonebook entries, qosbook profiles and bind bridge or PPPoE interfaces on top. <p>The default is pseudo.</p>	
type	Type of autopvc:	OPTIONAL
	<ul style="list-style-type: none"> ▶ bridge ▶ pppoerelay: an ETHoA interface will be created, will be bound to the ILMI PVC and will be added to the PPPoE relay as relay port. <p>The default is bridge.</p>	
opmode	Operational autopvc mode:	OPTIONAL
	<ul style="list-style-type: none"> ▶ partial: only read the ILMI VPC and VCC MIB tables ▶ full: read all supported MIB tables. <p>The default is partial.</p>	
overwrite	Enable/disable UBR peak rate overwrite. The default is enabled .	OPTIONAL
peakrate	A number between 0 and 27786. Represents the UBR peak rate (in kilobits per second).	OPTIONAL
	Note 0 indicates the linerate.	

EXAMPLE:

```
=>autopvc config
Autopvc mode      : pseudo
Autopvc opmode   : partial
Autopvc type      : bridge
Autopvc standard : unknown
Autopvc pseudo   : unknown
UBR overwrite     : enabled
UBR peak rate    : linerate
=>
```

RELATED COMMANDS:

autopvc info Show retrieved information.

autopvc info

Show retrieved information.

SYNTAX:

```
autopvc info [table = <{Vpc | Vcc}>]
```

where:

table	Choose the autopvc table for which the information must be shown. Choose between:	OPTIONAL
	▶ Vpc	
	▶ Vcc.	

EXAMPLE:

```
=>autopvc info
Address  Type      BestEff  Par1    Par2    Par3    Par4    Par5
8.35    ubr      Enabled Tx: 451  0      0      0      0
                Rx: 7923  0      0      0      0
=>
```

RELATED COMMANDS:

[autopvc config](#) Configure autopvc.

Bridge Commands

Introduction

This chapter describes the commands of the **bridge** command group.

Contents

This chapter covers the following commands:

<code>bridge config</code>	Show/set the bridge configuration settings.	46
<code>bridge flush</code>	Flush bridge interfaces and parameters.	47
<code>bridge ifadd</code>	Create a bridged Ethernet interface.	48
<code>bridge ifattach</code>	Attach a bridged Ethernet interface.	49
<code>bridge ifconfig</code>	Configure a bridge interface.	50
<code>bridge ifdelete</code>	Delete a bridge interface.	52
<code>bridge ifdetach</code>	Detach a bridge interface.	53
<code>bridge iflist</code>	Show the current state of all or the selected bridge interfaces.	54
<code>bridge macadd</code>	Add a static MAC address to the filtering database.	55
<code>bridge macdelete</code>	Remove a MAC address from the filtering database.	56
<code>bridge maclist</code>	Show the current MAC address filtering database.	57

bridge config

Show/set the bridge configuration settings.

SYNTAX:

```
bridge config [age = <number {10 - 100000}>]
              [filter = <{no_WAN_broadcast | PPPoE_only | none}>]
```

where:

age	A number between 10 and 100000 (seconds). Represents the lifetime of a dynamically learned Medium Access Control (MAC) address. The default is 300 .	OPTIONAL
filter	The bridge filter to be applied for all Wide Area Network (WAN) bridge ports. Choose between: <ul style="list-style-type: none"> ▶ no_WAN_broadcast: broadcasts from the SpeedTouch™ itself to the WAN are filtered out, broadcasts from the Local Area Network (LAN) to the WAN are still passed through. ▶ PPPoE_only: only PPPoE broadcasts are passed through, all others are filtered out ▶ none: no broadcasts are filtered out. The default is no_WAN_broadcast .	OPTIONAL

EXAMPLE:

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


bridge flush

Flush bridge interfaces and parameters.



The flush command does not impact previously saved configurations.

SYNTAX:

```
bridge flush
```

bridge ifadd

Create a bridged Ethernet interface.

SYNTAX:

```
bridge ifadd [intf = <intfname>]
             [dest = <string>]
```

where:

intf	The bridged Ethernet interface name.	OPTIONAL
	Note If not specified, the destination name will double as interface name.	
dest	The destination address for the new interface. Typically a phonebook entry.	OPTIONAL

EXAMPLE:

```
=>bridge iflist
OBC      : Internal
          Connection State: connected
          Port: OBC      PortNr: 0      PortState: forwarding
          RX bytes: 73592      frames: 702
          TX bytes: 69549      frames: 1050      dropframes: 0
...
ethport5 : Internal
          Connection State: connected
          Port: ethport5  PortNr: 5      PortState: forwarding
          RX bytes: 2150291312 frames: 2150259736
          TX bytes: 2150201352 frames: 2147888072 dropframes: 3
=>bridge ifadd intf=myBridgeIntf
=>bridge iflist
OBC      : Internal
          Connection State: connected
          Port: OBC      PortNr: 0      PortState: forwarding
          RX bytes: 73592      frames: 702
          TX bytes: 69549      frames: 1050      dropframes: 0
...
myBridgeIntf: dest : (none)
              Retry : 10 QoS : default Encaps : llc/snap Fcs : off
              Connection State: not-connected
              Port: (Unassigned) PortNr: (Unknown) PortState: forwarding
=>
```

RELATED COMMANDS:

- bridge ifdelete** Delete a bridge interface.
- bridge iflist** Show the current state of all or the selected bridge interfaces.

bridge ifattach

Attach a bridged Ethernet interface.

SYNTAX:

```
bridge ifattach intf = <intfname>
```

where:

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      PortNr: 0      PortState: forwarding
          RX bytes: 81812      frames: 778
          TX bytes: 76463      frames: 1161      dropframes: 0
...
myBridgeIntf: dest : ATM1
             Retry : 10 QoS : default Encaps : llc/snap Fcs : off
             Connection State: not-connected
             Port: (Unassigned) PortNr: (Unknown) PortState: forwarding

=>bridge ifattach intf=myBridgeIntf
=>bridge iflist
OBC      : Internal
          Connection State: connected
          Port: OBC      PortNr: 0      PortState: forwarding
          RX bytes: 85261      frames: 807
          TX bytes: 78983      frames: 1203      dropframes: 0
...
myBridgeIntf: dest : ATM1
             Retry : 10 QoS : default Encaps : llc/snap Fcs : off
             Connection State: connected
             Port: wan0   PortNr: 6      PortState: forwarding
             RX bytes: 0      frames: 0
             TX bytes: 0      frames: 0      dropframes: 0

=>
```

RELATED COMMANDS:

bridge ifdetach Detach a bridge interface.

bridge ifconfig

Configure a bridge interface.

SYNTAX:

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

where:

intf	The name of the bridge interface to configure.	REQUIRED
dest	The destination for this interface. Typically a phonebook entry. This parameter only needs to be specified when an interface has been created without specified destination.	OPTIONAL
qos	The name of the Quality Of Service (QoS) book entry to apply on this bridge interface.	OPTIONAL
encaps	The type of encapsulation to be used for this bridge interface. Choose between: <ul style="list-style-type: none"> ▶ llc/snap ▶ vcmux. 	OPTIONAL
fcs	Whether or not to include the Ethernet FCS in the packet header on the WAN side. Choose between: <ul style="list-style-type: none"> ▶ off ▶ on. The default is <i>off</i> . Note FCS is normally left off.	OPTIONAL
portstate	The bridge portstate for this interface. Choose between: <ul style="list-style-type: none"> ▶ disabled ▶ learning ▶ forwarding. The default is <i>forwarding</i> .	OPTIONAL
retry	A number between 0 and 65535. Represents the number of times the SpeedTouch™ retries to set up a WAN connection before giving up. The default is <i>10</i> .	OPTIONAL

EXAMPLE:

```

=>bridge iflist
OBC      : Internal
          Connection State: connected
          Port: OBC      PortNr: 0      PortState: forwarding
          RX bytes: 73592      frames: 702
          TX bytes: 69549      frames: 1050      dropframes: 0
...
myBridgeIntf: dest : (none)
          Retry : 10 QoS : default Encaps : llc/snap Fcs : off
          Connection State: not-connected
          Port: (Unassigned) PortNr: (Unknown) PortState: forwarding
=>
=>bridge ifconfig intf=myBridgeIntf dest=ATM1
=>
=>bridge iflist
OBC      : Internal
          Connection State: connected
          Port: OBC      PortNr: 0      PortState: forwarding
          RX bytes: 81812      frames: 778
          TX bytes: 76463      frames: 1161      dropframes: 0
...
myBridgeIntf: dest : ATM1
          Retry : 10 QoS : default Encaps : llc/snap Fcs : off
          Connection State: not-connected
          Port: (Unassigned) PortNr: (Unknown) PortState: forwarding
=>

```

bridge ifdelete

Delete a bridge interface.

SYNTAX:

```
bridge ifdelete intf = <intfname>
```

where:

intf	The name of the interface name to be deleted.	REQUIRED
------	---	----------

EXAMPLE:

```
=>bridge iflist
OBC      : Internal
          Connection State: connected
          Port: OBC      PortNr: 0      PortState: forwarding
          RX bytes: 99930      frames: 969
          TX bytes: 93572      frames: 1443      dropframes: 0
...
ethport5 : Internal
          Connection State: connected
          Port: ethport5  PortNr: 5      PortState: forwarding
          RX bytes: 2150291312 frames: 2150259736
          TX bytes: 2150201352 frames: 2147888072 dropframes: 3

myBridgeIntf: dest : ATM1
          Retry : 10  QoS : default  Encaps : llc/snap  Fcs : off
          Connection State: connected
          Port: wan0   PortNr: 6      PortState: forwarding
          RX bytes: 0      frames: 0
          TX bytes: 0      frames: 0      dropframes: 0

=>bridge ifdelete intf=myBridgeIntf
=>bridge iflist
OBC      : Internal
          Connection State: connected
          Port: OBC      PortNr: 0      PortState: forwarding
          RX bytes: 99930      frames: 969
          TX bytes: 93572      frames: 1443      dropframes: 0
...
ethport5 : Internal
          Connection State: connected
          Port: ethport5  PortNr: 5      PortState: forwarding
          RX bytes: 2150291312 frames: 2150259736
          TX bytes: 2150201352 frames: 2147888072 dropframes: 3

=>
```

RELATED COMMANDS:

- bridge ifadd** Create a bridged Ethernet interface.
- bridge iflist** Show the current state of all or the selected bridge interfaces.

bridge ifdetach

Detach a bridge interface.

SYNTAX:

```
bridge ifdetach intf = <intfname>
```

where:

intf	The name of the bridge interface to be detached.	REQUIRED
------	--	----------

EXAMPLE:

```
=>bridge iflist
OBC      : Internal
          Connection State: connected
          Port: OBC      PortNr: 0      PortState: forwarding
          RX bytes: 85261      frames: 807
          TX bytes: 78983      frames: 1203      dropframes: 0
...
myBridgeIntf: dest : ATM1
             Retry : 10 QoS : default Encaps : llc/snap Fcs : off
             Connection State: connected
             Port: wan0 PortNr: 6      PortState: forwarding
             RX bytes: 0      frames: 0
             TX bytes: 0      frames: 0      dropframes: 0

=>bridge ifdetach intf=myBridgeIntf
=>bridge iflist
OBC      : Internal
          Connection State: connected
          Port: OBC      PortNr: 0      PortState: forwarding
          RX bytes: 88882      frames: 837
          TX bytes: 81743      frames: 1249      dropframes: 0
...
myBridgeIntf: dest : ATM1
             Retry : 10 QoS : default Encaps : llc/snap Fcs : off
             Connection State: not-connected
             Port: (Unassigned) PortNr: (Unknown) PortState: forwarding

=>
```

RELATED COMMANDS:

bridge ifattach Attach a bridged Ethernet interface.

bridge iflist

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

SYNTAX:

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

where:

intf	The name of the bridge interface for which the configuration must be shown.	OPTIONAL
Note	If no interface is specified, all bridge interfaces are shown.	

EXAMPLE:

```
=>bridge iflist
OBC      : Internal
          Connection State: connected
          Port: OBC      PortNr: 0      PortState: forwarding
          RX bytes: 99930      frames: 969
          TX bytes: 93572      frames: 1443      dropframes: 0
...
ethport5 : Internal
          Connection State: connected
          Port: ethport5  PortNr: 5      PortState: forwarding
          RX bytes: 2150291312 frames: 2150259736
          TX bytes: 2150201352 frames: 2147888072 dropframes: 3
myBridgeIntf: dest : ATM1
          Retry : 10  QoS : default  Encaps : llc/snap  Fcs : off
          Connection State: connected
          Port: wan0   PortNr: 6      PortState: forwarding
          RX bytes: 0      frames: 0
          TX bytes: 0      frames: 0      dropframes: 0
=>
```

DESCRIPTION:

- RX bytes** The number of Received bytes.
- TX bytes** The number of Transmitted bytes.
- OBC** On Board Controller: indicates the physical bridge port.

RELATED COMMANDS:

- bridge ifadd** Create a bridged Ethernet interface.
- bridge ifdelete** Delete a bridge interface.

bridge macadd

Add a static MAC address to the filtering database.

This command allows to manually add static addresses, which should normally be dynamically discovered by the bridge itself.

SYNTAX:

```
bridge macadd intf = <intfname>
                hwaddr = <hardware-address>
```

where:

intf	The name of the bridge interface for which the MAC address must be added.	REQUIRED
hwaddr	The Ethernet MAC address of the new entry.	REQUIRED

EXAMPLE:

```
=>bridge maclist
00:0d:9d:47:dd:aa -- dynamic, ethport1, 300 seconds
00:90:d0:72:88:64 -- permanent
01:00:5e:00:00:67 -- static
01:00:5e:7f:ff:fa -- static
01:80:c2:00:00:00 -- permanent
01:80:c2:00:00:01 -- permanent
01:80:c2:00:00:02 -- permanent
01:80:c2:00:00:03 -- permanent
...
01:80:c2:00:00:0e -- permanent
01:80:c2:00:00:0f -- permanent
01:80:c2:00:00:10 -- permanent
ff:ff:ff:ff:ff:ff -- permanent
=>bridge macadd intf=ethport2 hwaddr=00:80:9f:01:23:45
=>bridge maclist
00:0d:9d:47:dd:aa -- dynamic, ethport1, 300 seconds
00:80:9f:01:23:45 -- static
00:90:d0:72:88:64 -- permanent
01:00:5e:00:00:67 -- static
01:00:5e:7f:ff:fa -- static
01:80:c2:00:00:00 -- permanent
01:80:c2:00:00:01 -- permanent
01:80:c2:00:00:02 -- permanent
01:80:c2:00:00:03 -- permanent
01:80:c2:00:00:0e -- permanent
01:80:c2:00:00:0f -- permanent
01:80:c2:00:00:10 -- permanent
ff:ff:ff:ff:ff:ff -- permanent
=>
```

RELATED COMMANDS:

- bridge macadd** Add a static MAC address to the filtering database.
- bridge macdelete** Remove a MAC address from the filtering database.
- bridge maclist** Show the current MAC address filtering database.

bridge macdelete

Remove a MAC address from the filtering database.

SYNTAX:

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

where:

hwaddr	The Ethernet MAC address of the entry which must be deleted.	REQUIRED
--------	--	----------

EXAMPLE:

```
=>bridge maclist
00:0d:9d:47:dd:aa -- dynamic, ethport1
00:80:9f:01:23:45 -- static
00:90:d0:72:88:64 -- permanent
01:00:5e:00:00:67 -- static
01:00:5e:7f:ff:fa -- static
01:80:c2:00:00:00 -- permanent
01:80:c2:00:00:01 -- permanent
01:80:c2:00:00:02 -- permanent
01:80:c2:00:00:03 -- permanent
...
01:80:c2:00:00:0e -- permanent
01:80:c2:00:00:0f -- permanent
01:80:c2:00:00:10 -- permanent
ff:ff:ff:ff:ff:ff -- permanent
=>bridge macdelete hwaddr=00:80:9f:01:23:45
=>bridge maclist
00:0d:9d:47:dd:aa -- dynamic, ethport1
00:90:d0:72:88:64 -- permanent
01:00:5e:00:00:67 -- static
01:00:5e:7f:ff:fa -- static
01:80:c2:00:00:00 -- permanent
01:80:c2:00:00:01 -- permanent
01:80:c2:00:00:02 -- permanent
01:80:c2:00:00:03 -- permanent
01:80:c2:00:00:0e -- permanent
01:80:c2:00:00:0f -- permanent
01:80:c2:00:00:10 -- permanent
ff:ff:ff:ff:ff:ff -- permanent
=>
```

bridge maclist

Show the current MAC address filtering database.

SYNTAX:

```
bridge maclist
```

EXAMPLE:

```
=>bridge maclist
00:0d:9d:47:dd:aa -- dynamic, ethport1, 300 seconds
00:90:d0:72:88:64 -- permanent
01:00:5e:00:00:67 -- static
01:00:5e:7f:ff:fa -- static
01:80:c2:00:00:00 -- permanent
01:80:c2:00:00:01 -- permanent
01:80:c2:00:00:02 -- permanent
01:80:c2:00:00:03 -- permanent
01:80:c2:00:00:04 -- permanent
01:80:c2:00:00:05 -- permanent
01:80:c2:00:00:06 -- permanent
01:80:c2:00:00:07 -- permanent
01:80:c2:00:00:08 -- permanent
01:80:c2:00:00:09 -- permanent
01:80:c2:00:00:0a -- permanent
01:80:c2:00:00:0b -- permanent
01:80:c2:00:00:0c -- permanent
01:80:c2:00:00:0d -- permanent
01:80:c2:00:00:0e -- permanent
01:80:c2:00:00:0f -- permanent
01:80:c2:00:00:10 -- permanent
ff:ff:ff:ff:ff:ff -- permanent
=>
```


CIP Commands

Introduction

This chapter describes the commands of the **cip** command group.

Contents

This chapter covers the following commands:

cip flush	Flush complete Classical IP over ATM (IP oA) configuration.	60
cip ifadd	Create a Classical IP oA interface at the local side of the Logical IP Subnet (LIS).	61
cip ifdelete	Delete a Classical IP oA interface at the local side of the LIS.	62
cip iflist	Show current Classical IP oA configuration.	63
cip pvcadd	Create a PVC Address Resolution Protocol (ARP) entry.	64
cip pvcdelete	Delete a PVC ARP entry.	65
cip pvclist	Show current PVC ARP entries.	66

cip flush

Flush complete Classical IP over ATM (IP oA) configuration.



The flush command does not impact previously saved configurations.

SYNTAX:

```
cip flush
```

cip ifadd

Create a Classical IP oA 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]>]
```

where:

addr	The Classical IP oA 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:*.03': ADSL port, any address, selector 3.	OPTIONAL

Note This parameter is only applicable in an Switched Virtual Channel (SVC) environment.

In most cases, the Classical IP oA LIS is built in a Permanent Virtual Channel (PVC) environment.

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 IP oA interface at the local side of the LIS.
- cip iflist** Show current Classical IP oA configuration.

cip ifdelete

Delete a Classical IP oA interface at the local side of the LIS.

SYNTAX:

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

where:

addr	The Classical IP oA 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 IP oA interface at the local side of the Logical IP Subnet (LIS).
cip iflist	Show current Classical IP oA configuration.

cip iflist

Show current Classical IP oA configuration.

SYNTAX:

```
cip iflist
```

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
=>
```

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:

This example shows the 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 IP oA interface at the local side of the Logical IP Subnet (LIS).
cip ifdelete	Delete a Classical IP oA interface at the local side of the LIS.

cip pvcadd

Create a PVC Address Resolution Protocol (ARP) entry.



For destinations which are not RFC1577/RFC2225 compliant.

SYNTAX:

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

where:

dest	The ATM address (hardware address) of the destination host. Typically a phonebook name.	REQUIRED
destaddr	The IP address of the destination host. Typically for destinations without Inverse ATMARP support.	OPTIONAL
mtu	A number between 273 and 20000. Represents the maximum ATM Adaption Layer 5 (AAL5) packet size (in bytes) for this connection. The default is 9180 .	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 pvdelete** Delete a PVC ARP entry.
- cip pvclist** Show current PVC ARP entries.

cip pvdelete

Delete a PVC ARP entry.

SYNTAX:

```
cip pvdelete dest = <phonebook name>
```

where:

dest	The ATM address (hardware address) or name of the entry to be deleted.	REQUIRED
-------------	--	-----------------

EXAMPLE:

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

RELATED COMMANDS:

- cip pvadd** Create a PVC Address Resolution Protocol (ARP) entry.
- cip pvclist** Show current PVC ARP entries.

cip pvclist

Show current PVC ARP entries.

SYNTAX:

```
cip pvclist
```

EXAMPLE:

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

EXAMPLE:

```
=>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:

<code>cip pvccadd</code>	Create a PVC Address Resolution Protocol (ARP) entry.
<code>cip pvccdelete</code>	Delete a PVC ARP entry.

Config Commands

Introduction

This chapter describes the commands of the **config** command group.

Contents

This chapter covers the following commands:

<code>config backup</code>	Store current configuration to backup file.	68
<code>config dump</code>	Show the saved configuration file.	69
<code>config erase</code>	Erase a user configuration file.	70
<code>config flush</code>	Flush the complete current configuration.	71
<code>config list</code>	Show the current configuration set.	72
<code>config load</code>	Load complete saved (backup) or default configuration file.	73
<code>config save</code>	Save the current configuration	74

config backup

Store current configuration to backup file.

SYNTAX:

```
config backup filename = <user configuration filename>
```

where:

filename	Filename for backup file of current configuration.	REQUIRED
----------	--	----------

RELATED COMMANDS:

<code>config dump</code>	Show the saved configuration file.
<code>config list</code>	Show the current configuration set.

config dump

Show the saved configuration file.

SYNTAX:

```
config dump
```

RELATED COMMANDS:

- [config backup](#) Store current configuration to backup file.
- [config list](#) Show the current configuration set.

config erase

Erase a user configuration file.



If no filename is specified, all the user configuration files, saved in the SpeedTouch™ permanent storage, are deleted.

SYNTAX:

```
config erase [filename = <string>]
```

where:

filename	Name of the configuration file to erase.	OPTIONAL
----------	--	----------

RELATED COMMANDS:

config flush	Flush the complete current configuration.
config load	Load complete saved (backup) or default configuration file.
config save	Save the current configuration

config flush

Flush the complete current configuration.



This command does not affect saved configurations.

SYNTAX:

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

where:

flush_ip	Flush IP settings (yes) or not (no). The default is no .	OPTIONAL
Note Not keeping the IP settings could cause lost IP connectivity in the LAN.		

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** Erase a user configuration file.
- config load** Load complete saved (backup) or default configuration file.
- config save** Save the current configuration

config list

Show the current configuration set.

SYNTAX:

```
config list [templates = <{no | yes}>]
```

where:

templates	List the template files (yes) or not (no) The default is no .	OPTIONAL
-----------	---	----------

EXAMPLE:

```
=>config list templates=yes
Configurations file(s):
  user.ini
Template file(s)
  No template files present
Factory template file(s) :
  pppoe.tpl
  pppoa.tpl
  br.tpl
=>
```

RELATED COMMANDS:

- config backup Store current configuration to backup file.
- config dump Show the saved configuration file.
- config list Show the current configuration set.

config load

Load complete saved (backup) or default configuration file.

SYNTAX:

```
config load [load_ip = <{no | yes}>]
            [defaults = <{yes | no}>]
            [flush = <{yes | no}>]
            [echo = <{no | yes}>]
            [filename = <string>]
```

where:

load_ip	Load IP settings (yes) or not (no). Note Not keeping the IP settings could cause lost IP connectivity in the LAN.	OPTIONAL
defaults	Load default configuration (yes) or saved configuration (no). Note If this parameter is not specified, the saved configuration will be loaded.	OPTIONAL
flush	Flush the current configuration before loading a new configuration (yes) or not (no). Note It is advisable to flush the current configuration before loading the new configuration.	OPTIONAL
echo	Echo each command string when loaded (yes) or not (no).	OPTIONAL
filename	Configuration filename.	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
  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
=>config load flush=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
  172.16.0.0/24    0.0.0.0/0       172.16.0.5      cip1    1
=>
```

RELATED COMMANDS:

- config erase** Erase a user configuration file.
- config flush** Flush the complete current configuration.
- config save** Save the current configuration

config save

Save the current configuration

All existing configurations and modifications as entered by the user are saved.

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
```

RELATED COMMANDS:

<code>config erase</code>	Erase a user configuration file.
<code>config flush</code>	Flush the complete current configuration.
<code>config load</code>	Load complete saved (backup) or default configuration file.

DHCP Commands

Introduction

This chapter describes the commands of the **dhcp** command group.

Contents

This chapter covers the following commands:

<code>dhcp client clear</code>	Clear Dynamic Host Configuration Protocol (DHCP) client statistics.	76
<code>dhcp client config</code>	Show/set DHCP client configuration.	77
<code>dhcp client flush</code>	Flush complete DHCP client configuration and dynamic interfaces.	78
<code>dhcp client ifadd</code>	Create a DHCP lease for a specific interface.	79
<code>dhcp client ifattach</code>	Attach a DHCP lease to a dynamic interface.	80
<code>dhcp client ifconfig</code>	Show/set the configuration of DHCP lease created for a specific interface.	81
<code>dhcp client ifdelete</code>	Delete a dynamic interface.	83
<code>dhcp client iflist</code>	Show all dynamic interfaces.	84
<code>dhcp client ifrelease</code>	Release a lease attached to a dynamic interface.	85
<code>dhcp client ifrenew</code>	Renew the lease of a dynamic interface.	87
<code>dhcp client stats</code>	Show DHCP client statistics.	89
<code>dhcp relay add</code>	Add a DHCP server to the DHCP server list.	90
<code>dhcp relay config</code>	Set the relay configuration settings.	91
<code>dhcp relay delete</code>	Delete a DHCP server from the DHCP server list.	92
<code>dhcp relay flush</code>	Flush the DHCP relay settings.	93
<code>dhcp relay ifconfig</code>	Configure a relay interface.	94
<code>dhcp relay iflist</code>	Show the configuration of the relay interfaces.	95
<code>dhcp relay list</code>	Show the DHCP server list.	96
<code>dhcp relay stats</code>	Show the DHCP relay statistics.	97
<code>dhcp server clear</code>	Clear the DHCP server statistics.	98
<code>dhcp server config</code>	Show/set the DHCP server configuration settings.	99
<code>dhcp server flush</code>	Flush all the DHCP server pool and lease entries.	100
<code>dhcp server policy</code>	Show/set the DHCP server policy.	101
<code>dhcp server stats</code>	Show the DHCP server statistics.	102
<code>dhcp server lease add</code>	Assign a DHCP server lease to a DHCP host in the local network.	104
<code>dhcp server lease delete</code>	Delete a DHCP lease.	105
<code>dhcp server lease flush</code>	Flush the complete DHCP server configuration and dynamic leases.	106
<code>dhcp server lease list</code>	List the current DHCP leases.	107
<code>dhcp server pool add</code>	Add a DHCP server pool.	108
<code>dhcp server pool config</code>	Configure an existing DHCP server pool.	109
<code>dhcp server pool delete</code>	Delete a DHCP server pool.	111
<code>dhcp server pool flush</code>	Flush all the DHCP server pools.	112
<code>dhcp server pool list</code>	List the current DHCP server pools.	113

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}>]
```

where:

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

dhcp client flush

Flush complete DHCP client configuration and dynamic interfaces.



The flush command does not impact previously saved configurations.

SYNTAX:

```
dhcp client flush
```

EXAMPLE:

```
=>dhcp client iflist
NewETHoA      : [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 = <string>
```

where:

intf	The name of the dynamic interface to be created.	REQUIRED
------	--	----------

EXAMPLE:

```
=>dhcp client iflist
No dynamic interfaces defined.
=>dhcp client ifadd intf=NewEthoa
=>dhcp client iflist
NewETHoA      : [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 ifdelete** Delete a dynamic interface.
- dhcp client iflist** Show all dynamic interfaces.

dhcp client ifattach

Attach a DHCP lease to a dynamic interface.

SYNTAX:

```
dhcp client ifattach intf = <string>
```

where:

intf	The name of the dynamic interface to be attached.	REQUIRED
------	---	----------

EXAMPLE:

```
=>dhcp client iflist
NewETHoA   : [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=NewETHoA
=>dhcp client iflist
NewETHoA   : [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.lease time = 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 ifrelease Release a lease attached to a dynamic interface.

dhcp client ifrenew Renew the lease of a dynamic interface.

dhcp client ifconfig

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

SYNTAX:

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

where:

intf	The name of the dynamic interface to be configured.	REQUIRED
clientid	The client identity to be associated with the lease. Use <i>none</i> 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. The default is 7200 (2 hours). Note 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) Domain Name System (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 0 and 100. Represents the gateway route metric. The default is 1 .	OPTIONAL
dnsmetric	A number between 0 and 100. Represents the DNS route metric. The default is 1 .	OPTIONAL

EXAMPLE:

```
=>dhcp client iflist
NewETHoA      : [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=NewETHoA hostname=NewLease addr=10.0.0.10 leasetime=10800
=>dhcp client iflist
NewETHoA      : [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 %
=>
```

dhcp client ifdelete

Delete a dynamic interface.

SYNTAX:

```
dhcp client ifdelete intf = <string>
```

where:

intf	The name of the dynamic interface to be deleted.	REQUIRED
------	--	----------

EXAMPLE:

```
=>dhcp client iflist
NewETHoA : [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.lease time  = 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 NewETHoA
=>dhcp client iflist
No dynamic interfaces defined.
=>
```

RELATED COMMANDS:

- [dhcp client ifadd](#) Create a DHCP lease for a specific interface.
- [dhcp client iflist](#) Show all dynamic interfaces.

dhcp client iflist

Show all dynamic interfaces.

SYNTAX:

```
dhcp client iflist
```

EXAMPLE 1:

```
=>dhcp client iflist
NewETHoA   : [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 2:

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:

<code>dhcp client ifadd</code>	Create a DHCP lease for a specific interface.
<code>dhcp client ifdelete</code>	Delete a dynamic interface.

dhcp client ifrelease

Release a lease attached to a dynamic interface.

SYNTAX:

```
dhcp client ifrelease intf = <string>
```

where:

intf The name of the dynamic interface.

REQUIRED

EXAMPLE 1:

```
=>dhcp client iflist
NewETHoA   : [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=NewETHoA
=>dhcp client iflist
NewETHoA   : [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 2:

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 rcv : 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 rcv : 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 a dynamic interface.
- dhcp client ifrenew** Renew the lease of a dynamic interface.

dhcp client ifrenew

Renew the lease of a dynamic interface.

SYNTAX:

```
dhcp client ifrenew intf = <string>
```

where:

intf The name of the dynamic interface.

REQUIRED

EXAMPLE 1:

```
=>dhcp client iflist
NewETHoA : [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.lease time = 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=NewETHoA
=>dhcp client iflist
NewETHoA : [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.lease time = 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 2:

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:

- | | |
|------------------------------------|--|
| <code>dhcp client ifattach</code> | Attach a DHCP lease to a dynamic interface. |
| <code>dhcp client ifrelease</code> | Release a lease attached to a dynamic 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 Dynamic Host Configuration Protocol (DHCP) client statistics.

dhcp relay add

Add a DHCP server to the DHCP server list.

SYNTAX:

```
dhcp relay add  addr = <ip-address>
                [intf = <none | interface_name>]
                [giaddr = <ip-address>]
```

where:

addr	The DHCP server IP address.	REQUIRED
intf	The name of the relay interface. Tip Use <i>None</i> to indicate that no interface is specified.	OPTIONAL
giaddr	The <i>giaddr</i> field to be used in relayed DHCP packets.	OPTIONAL

RELATED COMMANDS:

[dhcp relay delete](#) Delete a DHCP server from the DHCP server list.
[dhcp relay list](#) Show the DHCP server list.

dhcp relay config

Set the relay configuration settings.

SYNTAX:

```
dhcp server config [agentinfo = <{off | on}>]
                  [agentmismatch = <{off | on}>]
                  [trace = <{off | on}>]
```

where:

agentinfo	Sets the relay agent info status (RFC3046) off or on. The default is off .	OPTIONAL
agentmismatch	Forward/Drop DHCP reply packet when a relay agent info mismatch is detected (RFC3046) (on) or not (off). The default is off .	OPTIONAL
trace	Disable (off) or enable (on) verbose console logging and generation of debug traces. The default is off .	OPTIONAL

EXAMPLE:

```
=>dhcp relay config
Agent info status : off
Drop agent info mismatch status : off
Verbose console logging : off
=>dhcp relay config agentinfo=on
=>dhcp relay config
Agent info status : on
Drop agent info mismatch status : off
Verbose console logging : off
=>
```

dhcp relay delete

Delete a DHCP server from the DHCP server list.

SYNTAX:

```
dhcp relay delete  addr = <ip-address>
                   [intf = <string>]
```

where:

addr	The DHCP server IP address.	REQUIRED
intf	The name of the relay interface.	OPTIONAL

RELATED COMMANDS:

- `dhcp relay add` Add a DHCP server to the DHCP server list.
- `dhcp relay list` Show the DHCP server list.

dhcp relay flush

Flush the DHCP relay settings.



The flush command does not impact previously saved configurations.

SYNTAX:

```
dhcp relay flush
```

EXAMPLE:

```
=>dhcp relay list
  DHCP server      Interface      giaddr
-----
  127.0.0.1
=>dhcp relay flush
=>dhcp relay list
No dynamic interfaces defined.
=>
```

dhcp relay ifconfig

Configure a relay interface.

SYNTAX:

```
dhcp relay ifconfig intf = <string>
                    [relay = <{off | on}>]
                    [maxhops = <number{0-16}>]
                    [remoteid = <password>]
                    [trusted = <{no | yes}>]
```

where:

intf	The name of the relay interface to be configured.	REQUIRED
relay	Sets the relay status.	OPTIONAL
maxhops	A number between 0 and 16. Represents the maximum number of hops allowed in the DHCP packet. The default is 4 .	OPTIONAL
remoteid	Sets the remote id as specified in RFC3046.	OPTIONAL
trusted	Drop/Forward DHCP request packet when the DHCP Relay Agent Option is enabled (with the command :dhcp relay config agentinfo=on) and the giaddr field is 0 (RFC3046).	OPTIONAL

EXAMPLE:

```
=>dhcp relay ifconfig
intf = pppoa_pppoa
[relay] = off
[maxhops] = 4
[remoteid] =
[trusted] = no
=>
```


dhcp relay iflist

Show the configuration of the relay interfaces.

SYNTAX:

```
dhcp relay iflist [intf = <string>]
```

where:

intf The name of the relay interface to be shown.

OPTIONAL

Note If not specified, all the relay interfaces will be shown.

EXAMPLE:

```
=>dhcp relay iflist
pppoa_pppoa :
  admin state = down    oper state = down
  max hops = 4
  trusted = no         remote id =

eth0 :
  admin state = up      oper state = up
  max hops = 4
  trusted = no         remote id =

=>
```

dhcp relay list

Show the DHCP server list.

SYNTAX:

```
dhcp relay list
```

EXAMPLE:

```
=>dhcp relay list
  DHCP server      Interface      giaddr
-----
  127.0.0.1        eth0          10.0.0.138
=>
```

RELATED COMMANDS:

[dhcp relay add](#)

Add a DHCP server to the DHCP server list.

[dhcp relay delete](#)

Delete a DHCP server from the DHCP server list.

dhcp relay stats

Show the DHCP relay statistics.

SYNTAX:

```
dhcp relay stats
```

EXAMPLE:

```
=>dhcp relay stats
  DHCP relay statistics
-----
Client packet relayed   :      64
Server packet relayed  :       0
Bogus relay agent      :       0
Bogus giaddr recv     :       0
Corrupt agent option   :       0
Missing agent option   :       0
Bad circuit id         :       0
Missing circuit id     :       0
=>
```

dhcp server clear

Clear the 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 the DHCP server statistics.

dhcp server config

Show/set the DHCP server configuration settings.

SYNTAX:

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

where:

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 (on) or immediately start the DHCP server (off). The default is on .	OPTIONAL
scantime	A number between 0 and 1814400 (seconds). Represents the time for which the SpeedTouch™ scans for another DHCP server to be active in the network. The default is 20 .	OPTIONAL
state	State of the DHCP server (enabled/disabled). The default is enabled .	OPTIONAL
trace	Disable (off) or enable (on) verbose console logging and generation of debug traces. The default is off .	OPTIONAL

EXAMPLE:

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

dhcp server flush

Flush all the DHCP server pool and lease entries.



The flush command does not impact previously saved configurations.

SYNTAX:

```
dhcp server flush
```

dhcp server policy

Show/set the DHCP server policy.

SYNTAX:

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

where:

verifyfirst	Enable (on) or disable (off) IP address conflict network probing before handing out an address to a client. The default is off .	OPTIONAL
trustclient	Take the IP address suggested by a DHCP client into account (on) or not (off). The default is on .	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
=>
```

dhcp server stats

Show the DHCP server statistics.

SYNTAX:

```
dhcp server stats
```

EXAMPLE:

```
=>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
Relay agent options dropped :           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:

The stat ...	indicates ...
DHCP server state	the state of the SpeedTouch™ DHCP server.
Corrupted packet rcv	the number of corrupted packets (not complaint to RFC2131) received from the LAN.
DISCOVER	the number of DHCP server discovery packets received from the LAN. These broadcasts are sent by potential DHCP clients to locate available DHCP servers.
REQUEST	the number of DHCP address lease requests received from the LAN.
DECLINE	the number of DHCP address lease requests declined.
RELEASE	the number of DHCP address release requests received from DHCP clients.
INFORM	the number of information requests received from DHCP clients.
Pure BOOTP requests	the number of BOOTP requests received from the LAN.
Other message types	the number of other messages received from the LAN.
OFFERs sent	the number of IP address offers sent in reply to DHCP requests.
ACKs sent	the number of ACKnowledgement replies sent to successfully configured DHCP clients.
NAKs sent	the number of Not-AcKnowledgement replies sent to wrongly configured DHCP clients.
Relay agent options dropped	
Lease table got full	whether the maximum number of DHCP leases is reached or not.
Ping table got full	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 (<i>dhcp server policy verifyfirst=yes</i>).
Second DHCP server seen	whether a concurrent DHCP server was found on the LAN or not.

RELATED COMMANDS:

<code>dhcp server clear</code>	Clear the DHCP server 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 | ""}>]
```

where:

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. Tip Use the command <code>:dhcp server pool list</code> 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 server 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. Note Specifying -1 makes the lease permanent.	OPTIONAL
hostname	The hostname to add to the local DNS table for this host. Use "" if 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 list` List the current DHCP leases.

dhcp server lease delete

Delete a DHCP lease.

SYNTAX:

```
dhcp server lease delete [clientid = <clientid | none>]
                        [index = <number>]
```

where:

clientid	The DHCP client identification string of the DHCP lease.	OPTIONAL
Note	If no DHCP client is specified, all DHCP clients are deleted.	
index	The index number of the entry to be deleted.	OPTIONAL
Tip	Use the command :dhcp server lease list to obtain a list of the index numbers of all current DHCP leases.	

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** Assign a DHCP server lease to a DHCP host in the local network.
- dhcp server lease list** List the current DHCP leases.

dhcp server lease flush

Flush the complete DHCP server configuration and dynamic leases.



The flush command does not impact previously saved configurations.

SYNTAX:

```
dhcp server lease flush [pool = <string>]
```

where:

pool	The name of the DHCP server pool to be flushed.	OPTIONAL
Note	If not specified, the leases of all the pool will be flushed.	

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
=>
```

dhcp server lease list

List the current DHCP leases.

SYNTAX:

```
dhcp server lease list [clientid = <clientid | none>]
                        [index = <number>]
```

where:

clientid	The DHCP client identification string of the DHCP lease. Note If not is specified, all DHCP clients are listed.	OPTIONAL
index	The index number of the entry to be shown. Note If not specified, all the entries will be shown.	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
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** Assign a DHCP server lease to a DHCP host in the local network.
- dhcp server lease delete** Delete a DHCP lease.

dhcp server pool add

Add a DHCP server pool.

SYNTAX:

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

where:

name	A name for the DHCP server pool.	OPTIONAL
Note	If not specified, the name is "dhcp_pool_x", where x is a subsequent number.	
index	The number of the pool before which you want the new pool to be added.	OPTIONAL
Note	If not specified, the pool will be added at the bottom of the 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 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 server pool.
- [dhcp server pool list](#) List the current DHCP server pools.

dhcp server pool config

Configure an existing DHCP server pool.

SYNTAX:

```
dhcp server pool config name = <string>
                        [index = <number>]
                        [intf = <string>]
                        [poolstart = <ip-address>]
                        [poolend = <ip-address>]
                        [netmask = <ip-mask(dotted or cidr)>]
                        [gateway = <ipaddress | 0>]
                        [server = <ipaddress | 0>]
                        [primdns = <ipaddress | 0>]
                        [secdns = <ipaddress | 0>]
                        [dnsmetric = <number{0-100}>]
                        [primwins = <ipaddress | 0>]
                        [secwins = <ipaddress | 0>]
                        [leasetime = <number>]
                        [unnumbered = <{no | yes}>]
                        [localgw = <{off | on}>]
```

where:

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
intf	The interface for which the pool is allowed to lease IP addresses.	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
server	The IP address of the DHCP server for DHCP clients.	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
dnsmetric	The DHCP server pool DNS route metric.	OPTIONAL
primwins	The IP address of the primary WINS server for DHCP clients.	OPTIONAL
secwins	The IP address of the secondary WINS server for DHCP clients.	OPTIONAL
leasetime	A number between 0 and 1814400 (seconds). Represents the time for which a client can use its dynamically allocated IP address. The default is 7200 . Note Specifying -1 makes the lease permanent.	OPTIONAL
unnumbered	Assign an IP address from this pool to the DHCP server or not (dynamic pools only).	OPTIONAL
localgw	Proxy for a virtual default gateway residing in same subnet of DHCP client instead of the remote peer address.	OPTIONAL

EXAMPLE:

```
=>dhcp server pool list
Pool      Start      End        Intf      State
0 LAN_Private 10.0.0.1   10.0.0.254 eth0      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        Intf      State
0 LAN_Private 10.0.0.1   10.0.0.254 eth0      USED
1 My_Pool   192.6.11.101 192.6.11.254 eth0      USED
=>
```


dhcp server pool delete

Delete a DHCP server pool.

SYNTAX:

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

where:

name	The name of the DHCP server pool to delete.	REQUIRED
Tip	Use the command <code>:dhcp server pool list</code> to see a list of all current DHCP leases.	

EXAMPLE:

```
=>dhcp server pool list
Pool      Start      End        Intf       State
0 LAN_Private 10.0.0.1   10.0.0.254 eth0       USED
1 My_Pool    192.6.11.101 192.6.11.254 eth0       USED
=>dhcp server pool delete name=My_Pool
=>dhcp server pool list
Pool      Start      End        Intf       State
0 LAN_Private 10.0.0.1   10.0.0.254 eth0       USED
=>
```

RELATED COMMANDS:

- `dhcp server pool add` Add a DHCP server pool.
- `dhcp server pool list` List the current DHCP server pools.

dhcp server pool flush

Flush all the DHCP server pools.



The flush command does not impact previously saved configurations.

SYNTAX:

```
dhcp server pool flush
```

EXAMPLE:

```
=>dhcp server pool list
Pool      Start      End      Intf      State
0 LAN_Private  10.0.0.1  10.0.0.254  eth0     USED
1 My_Pool    192.6.11.101  192.6.11.254  eth0     USED
=>dhcp server pool flush
=>dhcp server pool list
=>
```

dhcp server pool list

List the current DHCP server pools.

SYNTAX:

```
dhcp server pool list [name = <string>]
```

where:

name	The name of the DHCP server pool to be shown.	OPTIONAL
Note	If not specified, all the DHCP server pools will be shown.	

EXAMPLE:

```
=>dhcp server pool list
Pool      Start      End      Intf      State
0 LAN_Private  10.0.0.1  10.0.0.254  eth0      USED
1 My_Pool    192.6.11.101  192.6.11.254  eth0      USED
=>
```

RELATED COMMANDS:

- `dhcp server pool add` Add a DHCP server pool.
- `dhcp server pool delete` Delete a DHCP server pool.

DNS Commands

Introduction

This chapter describes the commands of the **dns** command group.

Contents

This chapter covers the following commands:

<code>dns add</code>	Add an entry to the local DNS table.	116
<code>dns clear</code>	Clear the local DNS table.	117
<code>dns clrstats</code>	Clear the DNS statistics.	118
<code>dns delete</code>	Delete an entry from the local DNS table by index.	119
<code>dns domain</code>	Set the local DNS domain name.	120
<code>dns flush</code>	Flush the complete DNS server/forwarder configuration and static entries.	121
<code>dns fwdadd</code>	Add a DNS forwarding entry.	122
<code>dns fwddelete</code>	Delete a DNS forwarding entry.	123
<code>dns fwdlist</code>	List all forwarding entries.	124
<code>dns fwdtable</code>	Show DNS forwarding table.	125
<code>dns list</code>	List the contents of the local DNS table.	126
<code>dns nslookup</code>	Look up a name or an IP address via local DNS.	127
<code>dns start</code>	Start the local DNS server and forwarder.	128
<code>dns stats</code>	Show the DNS server/forwarder statistics.	129
<code>dns status</code>	Show the current status of the DNS server/forwarder.	130
<code>dns stop</code>	Stop the local DNS server/forwarder.	131
<code>dns toutfwd</code>	Set the DNS forwarding timeout.	132
<code>dns troff</code>	Disable verbose console messaging.	133
<code>dns tron</code>	Enable verbose console messaging.	134

dns add

Add an entry to the local DNS table.

SYNTAX:

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

where:

hostname	The name of the IP host to add (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 delete** Delete an entry from the local DNS table by index.
- dns list** List the contents of the local DNS table.

dns clear

Clear the local DNS table.

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%)
=>
```

dns clrstats

Clear the 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 the DNS server/forwarder statistics.

dns delete

Delete an entry from the local DNS table by index.

SYNTAX:

```
dns delete index = <number>
```

where:

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

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 an entry to the local DNS table.
- dns list** List the contents of the local DNS table.

dns domain

Set the local DNS domain name.

SYNTAX:

```
dns domain domain = <string>
```

where:

domain	The local DNS domain name of this domain.	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%)
```

dns flush

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



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.

SYNTAX:

```
dns fwdadd dns = <ip-address>
           src = <ip-address>
           mask = <ip-mask (dotted or cidr)>
           [metric = <number{0-100}>]
           [direct = <no | yes>]
```

where:

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
metric	A number between 0 and 100. Represents the metric (weight factor) for this DNS route. the default is 1 .	OPTIONAL
direct	Selects whether DNS replies are sent directly back to the client (yes) or are being relayed by the DNS forwarder (no). The default is no .	OPTIONAL

EXAMPLE:

```
=>dns fwdlist
DNS          SRC          Metric Direct Permanent
10.0.0.138   10.0.0.2         1      no      yes
=>dns fwdadd dns=10.0.0.138 src=10.0.0.3 mask=24 direct=yes
Dns forwarding server added.
=>dns fwdlist
DNS          SRC          Metric Direct Permanent
10.0.0.138   10.0.0.2         1      no      yes
10.0.0.138   10.0.0.3         1      yes     yes
=>
```

RELATED COMMANDS:

dns fwddel Delete a DNS forwarding entry.

dns fwdlist List all forwarding entries.

dns fwdddelete

Delete a DNS forwarding entry.

SYNTAX:

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

where:

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          SRC          Metric Direct Permanent
10.0.0.138   10.0.0.2          1      no      yes
10.0.0.138   10.0.0.3          1      yes     yes
=>dns fwdddelete dns=10.0.0.138 src=10.0.0.3 mask=24 direct=1
Dns forwarding server deleted.
=>dns fwdlist
DNS          SRC          Metric Direct Permanent
10.0.0.138   10.0.0.2          1      no      yes
=>
```

RELATED COMMANDS:

- dns fwdadd** Add a DNS forwarding entry.
- dns fwdlist** List all forwarding entries.

dns fwdlist

List all forwarding entries.

SYNTAX:

```
dns fwdlist
```

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
=>
```

RELATED COMMANDS:

`dns fwdadd` Add a DNS forwarding entry.
`dns fwddelete` Delete a DNS forwarding entry.

dns fwddtable

Show DNS forwarding table.

All the currently unresolved DNS requests will be shown.

SYNTAX:

```
dns fwddtable
```

EXAMPLE:

```
=>dns fwddtable
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 list

List the contents of the local DNS table.

SYNTAX:

```
dns list
```

EXAMPLE 1:

```
=>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 2:

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 an entry to the local DNS table. |
| dns delete | Delete an entry from the local DNS table by index. |

dns nslookup

Look up a name or an IP address via local DNS.

SYNTAX:

```
dns nslookup lookup = <string>
```

where:

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
=>
```

dns start

Start the local DNS server and 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:

<code>dns status</code>	Show the current status of the DNS server/forwarder.
<code>dns stop</code>	Stop the local DNS server/forwarder.

dns stats

Show the DNS server/forwarder statistics.

SYNTAX:

```
dns stats
```

EXAMPLE:

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)
dnisd: Internet class type A request received from 10.10.1.1.
dnisd: Client.SpeedLAN.local found in local database.
dnisd: Client.SpeedLAN.local resolved into 10.0.0.3.
=>(Ping Server.SpeedLAN.local)
dnisd: Internet class type A request received from 10.10.1.1.
dnisd: Server.SpeedLAN.local found in local database.
dnisd: 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 the DNS statistics.

dns status

Show the current status of the DNS server/forwarder.

SYNTAX:

```
dns status
```

EXAMPLE:

```
=>dns status
DNS server status: Started
DNS table size           : 64, in use: 1, free: 98 %
DNS forwarding table size : 10, in use: 0, free: 100 %
DNS forwarding dns servers table size : 21, in use: 1, free: 95 %
No dns cache.
Tracing: off
=>
```

RELATED COMMANDS:

<code>dns start</code>	Start the local DNS server and forwarder.
<code>dns stop</code>	Stop the local DNS server/forwarder.

dns stop

Stop the local 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 start** Start the local DNS server and forwarder.
- dns status** Show the current status of the DNS server/forwarder.

dns toutfwd

Set the DNS forwarding timeout.

SYNTAX:

```
dns toutfwd timeout = <number>
```

where:

<p>timeout A number (of 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. The default is 15.</p>	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%)
=>
```

dns troff

Disable verbose console messaging.

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 tron Enable verbose console messaging.

dns tron

Enable verbose console messaging.

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)
dnisd: Internet class type A request received from 10.0.0.10.
dnisd: aa.aa.be is outside our domain: forward.
dnisd: forwarding request from 10.0.0.10 (1318,0x0001) to 138.203.68.61
      (try=1): 'reply to ant' mode.
dnisd: Internet class type A request received from 10.0.0.10.
dnisd: aa.aa.be is outside our domain: forward.
dnisd: forwarding request from 10.0.0.10 (1318,0x0001) to 138.203.68.11
      (try=2): 'reply to ant' mode.
dnisd: forward answer from 138.203.68.11 to 10.0.0.10 (1318,0001).
dnisd: Internet class type A request received from 10.0.0.10.
dnisd: aa.aa.be.lan unknown: return error.
.....
=>(CTRL + S)
```

RELATED COMMANDS:

[dns troff](#)

Disable verbose console messaging.

Env Commands

Introduction

This chapter describes the commands of the **env** command group.

Contents

This chapter covers the following commands:

env flush	Flush all the non-system environment variables.	136
env get	Get the current value of an environment variable.	137
env list	Show all the current environment variables.	138
env set	Create, set or change the value of a non-system environment variable.	139
env unset	Delete a non-system environment variable.	140

env flush

Flush all the non-system environment variables.

SYNTAX:

```
env flush
```

EXAMPLE :

```
=>env list
_SNTPPOLL_POST_SYNC=15
_SNTPPOLL_PRE_SYNC=15
_COMPANY_NAME=THOMSON
_COMPANY_URL=http://www.thomson.net
_PROD_NAME=SpeedTouch
_PROD_URL=http://www.speedtouch.com
...
_ETHERNET=SWITCH
_MACADDR=00-90-D0-72-01-D3
_UDN=uuid:UPnP-SpeedTouch610-1_00-90-D0-72-01-D3
_IGDX_VERSION=1.1
_WIZ_AUTOPOPUP=1
CONF_REGION=World
CONF_PROVIDER=Basic
CONF_DESCRIPTION=Routed PPPoA Packet Service configuration using always-on sessi
on connectivity.
CONF_SERVICE=Routed PPPoA - DHCP - NAPT
CONF_DATE=Configuration set by Embedded Wizard
CONF_TPVERSION=1.2.0
HOST_SETUP=auto
UPGRADE_URL=http://www.speedtouch.com/upgrade600.htm
ATM_addr=8.35
SNMP_public=public
SNMP_private=private
SNMP_contact=Service Provider
SNMP_location=Customer Premises
PPP_user=johndoe@MyISP
CONF_TEMPLATE=pppoa.tpl
CONF_USR_COMMENT=Configuration created by Embedded Wizard (profile:active/pppoe.tpl)
=>env flush
=>env list
_SNTPPOLL_POST_SYNC=15
_SNTPPOLL_PRE_SYNC=15
_COMPANY_NAME=THOMSON
_COMPANY_URL=http://www.thomson.net
_PROD_NAME=SpeedTouch
_PROD_URL=http://www.speedtouch.com
...
_ETHERNET=SWITCH
_MACADDR=00-90-D0-72-01-D3
_UDN=uuid:UPnP-SpeedTouch610-1_00-90-D0-72-01-D3
_IGDX_VERSION=1.1
_WIZ_AUTOPOPUP=1
=>
```

env get

Get the current value of an environment variable.

SYNTAX:

```
env get var = <string>
```

where:

var	The name of the environment variable.	REQUIRED
-----	---------------------------------------	----------

Tip Use the command `:env list` for a list of all environment variables.

EXAMPLE:

```
=>env get var=ATM_addr
8*35
=>
```

RELATED COMMANDS:

- `env list` Show all the current environment variables.
- `env set` Create, set or change the value of a non-system environment variable.
- `env unset` Delete a non-system environment variable.

env list

Show all the current environment variables.

SYNTAX:

```
env list
```

EXAMPLE:

```
=>env list
_COMPANY_NAME=THOMSON
_COMPANY_URL=http://www.thomson.net
_PROD_NAME=SpeedTouch
_PROD_URL=http://www.speedtouch.com
_PROD_FRIENDLY_NAME=SpeedTouch 536
_PROD_DESCRIPTION=DSL Internet Gateway Device
_PROD_NUMBER=536
_BOARD_SERIAL_NBR=0441DG02V
_PROD_SERIAL_NBR=CP0441DG02V
_FII=5.2.7.5.0
_BUILD=5.2.7.5.0
_BOOTLOADER_VERSION=1.0.8
_BUILDVARIANT=AA
_MODEMLABEL=MOD_BCM6345_V1.0.59_ADSL_PHY_A2pB015c7_reads1.d14k6
_PHYSLAYERTYPE=POTS
_BUILDNAME=ZZTVAA5.275
_PRL=35837540
_FIA=Q0
_BOARD_NAME=BANT-J
_COMPANY_ID=ALCL
_COPYRIGHT=Copyright (c) 1999-2004, THOMSON
_TPVERSION=1.2.0
_ETHERNET=SINGLE
_CHIPSET=bcm_mips
_MACADDR=00-0E-50-3F-45-A0
_UDN=uuid:UPnP-SpeedTouch536-1_00-0E-50-3F-45-A0
_IGDX_VERSION=1.1
_WIZ_AUTOPOPUP=1
CONF_REGION=---
CONF_PROVIDER=---
CONF_DESCRIPTION=Factory defaults
CONF_SERVICE=Bridging on 0/35 and 8/35
CONF_DATE=Configuration modified manually
HOST_SETUP=auto
UPGRADE_URL=http://www.speedtouch.com/upgrade500.htm
CONF_TPVERSION=1.2.0
COLUMNS=80
ROWS=24
SESSIONTIMEOUT=0
=>
```

RELATED COMMANDS:

env get	Get the current value of an environment variable.
env set	Create, set or change the value of a non-system environment variable.
env unset	Delete a non-system environment variable.

env set

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

SYNTAX:

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

where:

var	The name of the environment variable. Tip 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 translated string which defines the value of the environment variable. Note The value of system variables (built-in variables with names starting with an underscore "_", "CONF" or "HOST") cannot be changed.	REQUIRED

EXAMPLE:

For infinite TELNET time out, set the value of the variable SESSIONTIMEOUT to 0:

```
=>env set var=SESSIONTIMEOUT value=0
=>
```

RELATED COMMANDS:

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

env unset

Delete a non-system environment variable.

SYNTAX:

```
env unset var = <string>
```

where:

var	The name of the environment variable to delete.	REQUIRED
-----	---	----------

Note System variables (built-in variables with names starting with an underscore “_”, “CONF” or “HOST”) cannot be unset, changed or deleted.

EXAMPLE:

```
=>env list
_COMPANY_NAME=THOMSON multimedia
_COMPANY_URL=http://www.speedtouch.com
_PROD_NAME=SpeedTouch
.....
CONF_DATE=March 2004
HOST_SETUP=user
ATM_addr=8*35
=>env unset var=ATM_addr
=>env list
_COMPANY_NAME=THOMSON multimedia
_COMPANY_URL=http://www.speedtouch.com
_PROD_NAME=SpeedTouch
.....
CONF_DATE=March 2004
HOST_SETUP=user
=>
```

RELATED COMMANDS:

env get	Get the current value of an environment variable.
env list	Show all the current environment variables.
env set	Create, set or change the value of a non-system environment variable.

Eth Commands

Introduction

This chapter describes the commands of the **eth** command group.

Contents

This chapter covers the following commands:

<code>eth config</code>	Enable/disable the Ethernet interface.	142
<code>eth ifconfig</code>	Configure the Ethernet port.	143
<code>eth iflist</code>	Show the Ethernet port configuration and current operating status.	144

eth config

Enable/disable the Ethernet interface.

SYNTAX:

```
eth config intf = <number>
           [state = <{enabled | disabled}>]
```

where:

intf	The number of the Ethernet interface to be configured.	REQUIRED
state	Enable or disable the Ethernet interface. The default is enabled .	OPTIONAL

EXAMPLE:

The example below is for a SpeedTouch™516:

```
=>eth config
ETH Intf 1 port state = UP [forwarding]
=>eth config intf=1 state=disabled
=>eth config
ETH Intf 1 port state = DOWN [disabled]
=>
```

The example below is for a SpeedTouch™546:

```
=>eth config
ETH Intf 1 port state = UP [forwarding]
ETH Intf 2 port state = UP [forwarding]
ETH Intf 3 port state = UP [forwarding]
ETH Intf 4 port state = UP [forwarding]
=>eth config intf 4 state=disabled
=>eth config
ETH Intf 1 port state = UP [forwarding]
ETH Intf 2 port state = UP [forwarding]
ETH Intf 3 port state = UP [forwarding]
ETH Intf 4 port state = DOWN [disabled]
=>
```

RELATED COMMANDS:

eth iflist Show the Ethernet port configuration and current operating status.

eth ifconfig

Configure the Ethernet port.

SYNTAX:

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

where:

- | | | |
|------|---|----------|
| intf | The number of the Ethernet interface to be configured. | REQUIRED |
| type | The Ethernet type.
Select either: | REQUIRED |
| | <ul style="list-style-type: none"> ▶ 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. | |

or enter a number between 0 (auto) and 5 (100BaseTFD).
The default is **auto**.

Note This value should never be changed, except in case of communication problems.

RELATED COMMANDS:

- | | |
|----------------------------|--|
| eth iflist | Show the Ethernet port configuration and current operating status. |
|----------------------------|--|

eth iflist

Show the Ethernet port configuration and current operating status.

SYNTAX:

```
eth iflist
```

EXAMPLE:

The example below is for a SpeedTouch™516:

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

The example below is for a SpeedTouch™546:

```
=>eth iflist
Intf  Type          Result Type
1     auto          100BaseTFD
2     auto          100BaseTFD
3     auto          Not connected
4     auto          100BaseTFD
=>
```

DESCRIPTION:

- ▶ **Type:** Indicates the configured Ethernet communication speed and duplex mode.
- ▶ **Result type:** Indicates the effective operating status if **Type** equals **auto**. In other cases, when the Ethernet types do NOT match, **Result Type** will equal **unknown** and no Ethernet connectivity will exist.

RELATED COMMANDS:

`eth ifconfig` Configure the Ethernet port.

ETHoA Commands

Introduction

This chapter describes the commands of the **ethoa** command group.

Contents

This chapter covers the following commands:

ethoa flush	Flush the ETHoA interfaces.	146
ethoa ifadd	Create a new ETHoA interface.	147
ethoa ifattach	Attach an ETHoA interface.	148
ethoa ifconfig	Configure an ETHoA interface.	149
ethoa ifdelete	Delete an ETHoA interface.	150
ethoa ifdetach	Detach an ETHoA interface.	151
ethoa iflist	Show all or a specified ETHoA interface(s).	152

ethoa flush

Flush the ETHoA interfaces.



The flush command does not impact previously saved configurations.

SYNTAX:

```
ethoa flush
```

EXAMPLE:

```
=>ethoa iflist
Newethoa      : 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
=>ethoa flush
=>ethoa iflist
=>
```

ethoa ifadd

Create a new ETHoA interface.

SYNTAX:

```
ethoa ifadd [intf = <string>]
           [dest = <string>]
```

where:

intf	The name for the new Ethernet Over ATM (ETHoA) interface.	OPTIONAL
	Note If not specified, the destination will double as interface name.	
dest	The destination for the new ETHoA interface. This is a phonebook entry.	OPTIONAL

EXAMPLE:

```
=>ethoa iflist
Newethoa      : 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
=>ethoa ifadd intf=Moreethoa dest=Br4
=>ethoa iflist
Newethoa      : 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
Moreethoa     : dest : Br4
                Retry : 10      QoS : default      Encaps : llc/snap      Fcs : off
                Connection State : not-connected
=>
```

RELATED COMMANDS:

- ethoa ifdelete** Delete an ETHoA interface.
- ethoa iflist** Show all or a specified ETHoA interface(s).

ethoa ifattach

Attach an ETHoA interface.

SYNTAX:

```
ethoa ifattach intf = <string>
```

where:

intf	The name of the ETHoA interface to be attached.	REQUIRED
------	---	----------

EXAMPLE:

```
=>ethoa iflist
Newethoa      : 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
Moreethoa     : dest : Br4
               Retry : 10      QoS : default      Encaps : llc/snap      Fcs : off
               Connection State : not-connected
=>ethoa ifattach intf=Moreethoa
=>ethoa iflist
Newethoa      : 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
Moreethoa     : 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:

ethoa ifdetach Detach an ETHoA interface.

ethoa ifconfig

Configure an ETHoA interface.

SYNTAX:

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

where:

intf	The name of the ETHoA interface to be configured.	REQUIRED
dest	The destination for this interface. Typically a phonebook entry. Note This parameter needs only to be specified in case of an interface created without specified destination.	OPTIONAL
qos	The name of a qosbook entry defining the QoS parameters for the WAN link.	OPTIONAL
encaps	The type of encapsulation to be used for this ETHoA 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. The default is 10 .	OPTIONAL

EXAMPLE:

```
=>ethoa iflist
Moreethoa      : 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
=>ethoa ifconfig intf=Moreethoa encaps=llc/snap retry=15
=>ethoa iflist
Moreethoa      : 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
=>
```

ethoa ifdelete

Delete an ETHoA interface.

SYNTAX:

```
ethoa ifdelete intf = <string>
```

here:

intf	The name of the ETHoA interface to be deleted.	REQUIRED
------	--	----------

EXAMPLE:

```
=>ethoa iflist
Newethoa      : 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
Moreethoa     : dest : Br4
                Retry : 10      QoS : default      Encaps : llc/snap      Fcs : off
                Connection State : not-connected
=>ethoa ifdelete intf=Moreethoa
=>ethoa iflist
Newethoa      : 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:

ethoa ifadd	Create a new ETHoA interface.
ethoa iflist	Show all or a specified ETHoA interface(s).

ethoa ifdetach

Detach an ETHoA interface.

SYNTAX:

```
ethoa ifdetach intf = <string>
```

where:

intf	The name of the ETHoA interface to be detached.	REQUIRED
------	---	----------

EXAMPLE:

```
=>ethoa iflist
Newethoa      : 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
Moreethoa     : 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
=>ethoa ifdetach intf=Moreethoa
=>ethoa iflist
Newethoa      : 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
Moreethoa     : dest : Br4
               Retry : 10      QoS : default      Encaps : llc/snap      Fcs : off
               Connection State : not-connected
=>
```

RELATED COMMANDS:

ethoa ifattach Attach an ETHoA interface.

ethoa iflist

Show all or a specified ETHoA interface(s).

SYNTAX:

```
ethoa iflist [intf = <string>]
```

where:

intf	The name of the ETHoA interface.	OPTIONAL
------	----------------------------------	----------

Note If not specified, all ETHoA interfaces are shown.

EXAMPLE:

```
=>ethoa iflist
Newethoa      : 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
Moreethoa     : 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:

ethoa ifadd	Create a new ETHoA interface.
ethoa ifdelete	Delete an ETHoA interface.

Firewall Commands

Introduction

This chapter describes the commands of the **firewall** command group.

Contents

This chapter covers the following commands:

firewall assign	Assign a chain to an entry point.	154
firewall flush	Clear all the hooks, chains and rules.	155
firewall list	Show the association(s) between all hooks or a specified hook and their chain(s).	156
firewall troff	Disable verbose console messaging.	157
firewall tron	Enable verbose console messaging.	158
firewall unassign	Clear a specific hook.	159
firewall chain create	Create a new chain.	160
firewall chain delete	Delete a chain.	161
firewall chain flush	Flush all chains.	162
firewall chain list	Show a list of all current chains.	163
firewall rule clear	Clear the rule statistics.	164
firewall rule create	Create a rule.	165
firewall rule delete	Delete a rule.	168
firewall rule flush	Flush all rules created for a chain(s).	169
firewall rule list	Show a list of rules.	170
firewall rule stats	Show the rule statistics.	171

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>
```

where:

hook	The name of the entry point to which a chain must be assigned. Choose between:	REQUIRED
	<ul style="list-style-type: none"> ▶ input: The point of 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 of 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 of 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 of 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 of all outgoing traffic. At this point, it can be determined whether the packet is allowed to leave the SpeedTouch™ IP router or local host. 	
chain	The name of the chain to be used.	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 list	Show the association(s) between all hooks or a specified hook and their chain(s).
firewall unassign	Clear a specific hook.

firewall flush

Clear all the hooks, chains and rules.



The flush command does not impact previously saved configurations.

SYNTAX:

```
firewall flush
```

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
=>
```

firewall list

Show the association(s) between all hooks or a specified hook and their chain(s).

SYNTAX:

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

where:

hook	The name of the hook for which the associations must be shown. Choose between:	OPTIONAL
	<ul style="list-style-type: none">▶ input▶ sink▶ forward▶ source▶ output.	

Note If not specified, the associations for all hooks are shown.

EXAMPLE:

```
=>firewall list
firewall assign hook=input chain=None
firewall assign hook=sink chain=sink
firewall assign hook=forward chain=forward
firewall assign hook=source chain=source
firewall assign hook=output chain=None
=>firewall list hook=input
firewall assign hook=input chain=None
=>firewall list hook=forward
firewall assign hook=forward chain=forward
=>
```

RELATED COMMANDS:

firewall assign	Assign a chain to an entry point.
firewall unassign	Clear a specific hook.

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

Clear a specific hook.

SYNTAX:

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

where:

hook	The name of the hook to be cleared. Choose between:	REQUIRED
	<ul style="list-style-type: none"> ▶ input ▶ sink ▶ forward ▶ source ▶ output. 	

RELATED COMMANDS:

- firewall assign** Assign a chain to an entry point.
- firewall list** Show the association(s) between all hooks or a specified hook and their chain(s).

firewall chain create

Create a new chain.

SYNTAX:

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

where:

chain	The name of the chain to be created.	REQUIRED
-------	--------------------------------------	----------

EXAMPLE:

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

RELATED COMMANDS:

<code>firewall chain delete</code>	Delete a chain.
<code>firewall chain list</code>	Show a list of all current chains.

firewall chain delete

Delete a chain.

SYNTAX:

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

where:

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

firewall chain flush

Flush all chains.

SYNTAX:

```
firewall chain flush
```

EXAMPLE:

```
=>firewall chain list  
source, forward, sink  
=>firewall chain flush  
=>firewall chain list  
=>
```

firewall chain list

Show a list of all current chains.

SYNTAX:

```
firewall chain list
```

EXAMPLE:

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

RELATED COMMANDS:

firewall chain create	Create a new chain.
firewall chain delete	Delete a chain.

firewall rule clear

Clear the rule statistics.

SYNTAX:

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

where:

chain	The name of the chain in which the rule is to be found.	OPTIONAL
Note	If not specified, the statistics of all the chains will be cleared.	
index	The index number (determined by the position) of the rule in the chain.	OPTIONAL
Note	If not specified, the statistics for all the rules in the chain will be cleared.	

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 stats Show the rule statistics.

firewall rule create

Create a rule.

SYNTAX:

```

firewall rule create chain = <string>
                    [index = <number>]
                    [srcintf [!]= <string>]
                    [srcintfgrp [!]= <{wan | local | lan} or number>]
                    [src [!]= <ip-address>]
                    [dstintf [!]= <string>]
                    [dstintfgrp [!]= <{wan | local | lan} or number>]
                    [dst [!]= <ip-address>]
                    [tos [!]= <number{1-255}>]
                    [precedence [!]= <number{0-7}>]
                    [dscp [!]= <number{0-63}>]
                    [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>]
                    [log = <{no | yes}>]
                    action = <{accept | deny | drop | count}>
    
```



If a value is preceded by a “!”, it means NOT.
For example “srcintfgrp=!wan” means “if the source interface group is different from WAN”.

where:

chain	The name of the chain in which the rule must be inserted.	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. Note NOT applicable if used in a chain assigned to the <i>output</i> hook.	OPTIONAL
srcintfgrp	The interface group the packet should [or should NOT] arrive on. Choose between: <ul style="list-style-type: none"> ▶ wan ▶ local ▶ lan. Note NOT applicable if used in a chain assigned to the <i>output</i> hook.	OPTIONAL

src	The source IP address (range) the packet should [or should NOT] come from. (Supports cidr notation).	OPTIONAL
dstintf	The name of the interface the packet should [or should NOT] be going to. Note NOT applicable if used in a chain assigned to the <i>output</i> hook.	OPTIONAL
dstintfgrp	The interface group the packet should [or should NOT] be going to. Choose between: <ul style="list-style-type: none"> ▶ wan ▶ local ▶ lan. Note NOT applicable if used in a chain assigned to the <i>output</i> hook.	OPTIONAL
dst	The destination IP address (range) the packet should [or should NOT] be going to (supports cidr notation).	OPTIONAL
precedence	A number between 0 and 7. Represents the precedence in the IP packet (part of tos).	OPTIONAL
dscp	A number between 0 and 63. Represents the DSCP in the IP packet (part of tos).	OPTIONAL
tos	A number between 0 and 255. Represents the Type Of Service (ToS) specification which should be expected [or NOT expected] in the IP packet. The ToS numbering specification is in accordance to the latest version of <i>RFC1700: Assigned numbers</i> .	OPTIONAL
prot	The protocol (name or number) expected [or NOT expected] in the IP packet. Choose between: <ul style="list-style-type: none"> ▶ icmp ▶ igmp ▶ ipinip ▶ tcp ▶ udp ▶ ah ▶ esp ▶ ipcomp or, alternatively, specify the protocol number.	OPTIONAL
syn	Expect Transmission Control Protocol (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/User Datagram Protocol (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 375) or, alternatively, specify the port 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 375) or, alternatively, specify the port 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 375) or, alternatively, specify the port 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 375) or, alternatively, specify the port number.	OPTIONAL
icmptype	The expected [or NOT expected] Internet Control Message Protocol (ICMP) type (name or number) of the packet. Select one of the supported ICMP type names (See “ Supported ICMP Type Names ” on page 378) or, alternatively, specify the ICMP type 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 <i>RFC1700: Assigned numbers</i> .	OPTIONAL
icmpcodeend	A number between 0 and 15. Represents the ICMP code range end. Note Only applicable for ranges.	OPTIONAL
clink	The name of the chain to be parsed when this rule applies (action is ignored).	OPTIONAL
log	Logging is done when this rule applies.	OPTIONAL
action	Action to be taken when this rule applies. Choose between: <ul style="list-style-type: none"> ▶ 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, i.e. without sending an error message to the sender. ▶ count: update of statistics. Has no influence on the packet. 	REQUIRED

RELATED COMMANDS:

<code>firewall rule create</code>	Create a rule.
<code>firewall rule delete</code>	Delete a rule.
<code>firewall rule list</code>	Show a list of rules.

firewall rule delete

Delete a rule.

SYNTAX:

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

where:

chain	The name of the chain in which the rule must be deleted.	REQUIRED
index	The index number of the rule in the chain.	REQUIRED
Tip	Use the command: firewall rule list for the index number of the applicable rule.	

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 create	Create a rule.
firewall rule delete	Delete a rule.
firewall rule list	Show a list of rules.

firewall rule flush

Flush all rules created for a chain(s).



1. The flush command does not impact previously saved configurations.
2. The chain itself is not removed.

SYNTAX:

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

where:

chain	The name of the chain to be emptied.	OPTIONAL
Note	If this parameter is not specified, all rules for all chains are deleted.	

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
=>
```

firewall rule list

Show a list of rules.

SYNTAX:

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

where:

chain	The name of the chain for which the rules must be listed.	OPTIONAL
Note	If not specified, all rules for all chains are shown.	

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 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 create	Create a rule.
firewall rule delete	Delete a rule.
firewall rule list	Show a list of rules.

firewall rule stats

Show the rule statistics.

The number of packets and bytes which have passed the hooks are listed.

SYNTAX:

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

where:

chain	The name of the chain for which the statistics must be listed. Note If not specified, the statistics for the rules applicable to all chains are shown.	OPTIONAL
index	The index number of the chain's rule for which the statistics must be listed. Tip Use the command <code>:firewall rule list</code> to determine the index number of the applicable rule. Note If not specified, the statistics for all rules applicable to the specified chain are shown.	OPTIONAL

EXAMPLE 1:

```
=>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 2:

```
=>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 the rule statistics.

IP Commands

Introduction

This chapter describes the commands of the **ip** command group.

Contents

This chapter covers the following commands:

ip apadd	Assign an IP address to an IP interface.	174
ip apdelete	Remove an IP address from an IP interface.	176
ip aplist	Show a list of all configured IP addresses.	177
ip arpad	Add an entry to the ARP cache of a broadcast IP interface.	178
ip arpdelete	Remove an entry from the ARP cache.	179
ip arplist	Show the ARP cache.	180
ip config	Show/set global IP stack configuration options.	181
ip flush	Flush the complete IP configuration.	184
ip ifconfig	Configure IP interface parameters.	185
ip iflist	Show all IP interfaces.	186
ip ifwait	Wait for a status change of an IP interface.	187
ip mcadd	Add an MC address to an MC capable interface.	188
ip mcdelete	Delete an MC address to an MC capable interface.	189
ip mclist	List all MC addresses.	190
ip ping	Send ICMP ECHO_REQUEST packets.	191
ip rtadd	Add a route to the SpeedTouch™ routing table.	192
ip rtdelete	Delete a route from the routing table.	193
ip rtlist	Show the current routing table.	194
ip sendto	Send UDP packets.	195
ip traceroute	Send ICMP/UDP packets to trace the ip path.	196
ip auto flush	Flush the autoIP interfaces.	198
ip auto ifadd	Create a new autoIP interface.	199
ip auto ifattach	Select and assign a link-local address to an autoIP interface.	200
ip auto ifconfig	Configure an autoIP interface.	201
ip auto ifdelete	Delete an existing autoIP interface.	202
ip auto ifdetach	Release the link-local address for the given autoIP interface.	203
ip auto iflist	Show the autoIP interfaces.	204

ip apadd

Assign an IP address to an IP interface.

SYNTAX:

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

where:

addr	The new IP address to be added.	REQUIRED
netmask	The subnetmask associated with this address.	OPTIONAL
intf	The IP 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

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 bcastpkts:290555
IPTX bytes:839550 unicastpkts:11477 bcastpkts:0 droppkts:0
HWRX bytes:0 unicastpkts:0 bcastpkts:0
HWTX bytes:0 unicastpkts:0 bcastpkts: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 bcastpkts:2
IPTX bytes:0 unicastpkts:0 bcastpkts:0 droppkts:0
HWRX bytes:0 unicastpkts:0 bcastpkts:0
HWTX bytes:0 unicastpkts:0 bcastpkts: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 bcastpkts:0
IPTX bytes:0 unicastpkts:0 bcastpkts:0 droppkts:0
HWRX bytes:0 unicastpkts:0 bcastpkts:0
HWTX bytes:0 unicastpkts:0 bcastpkts: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 bcastpkts:290669
IPTX bytes:853114 unicastpkts:11662 bcastpkts:0 droppkts:0
HWRX bytes:0 unicastpkts:0 bcastpkts:0
HWTX bytes:0 unicastpkts:0 bcastpkts: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 bcastpkts:2
IPTX bytes:0 unicastpkts:0 bcastpkts:0 droppkts:0
HWRX bytes:0 unicastpkts:0 bcastpkts:0
HWTX bytes:0 unicastpkts:0 bcastpkts:0 droppkts:0
=>

```

RELATED COMMANDS:

- ip apdelete** Remove an IP address from an IP interface.
- ip aplist** Show a list of all configured IP addresses.

ip apdelete

Remove an IP address from an IP interface.

SYNTAX:

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

where:

addr	The IP address to be deleted.	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 broadcastpkts:0
IPTX bytes:0 unicastpkts:0 broadcastpkts:0 dropkts:0
HWRX bytes:0 unicastpkts:0 broadcastpkts:0
HWTX bytes:0 unicastpkts:0 broadcastpkts:0 dropkts: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 broadcastpkts:290555
IPTX bytes:839550 unicastpkts:11477 broadcastpkts:0 dropkts:0
HWRX bytes:0 unicastpkts:0 broadcastpkts:0
HWTX bytes:0 unicastpkts:0 broadcastpkts:0 dropkts: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 broadcastpkts:2
IPTX bytes:0 unicastpkts:0 broadcastpkts:0 dropkts:0
HWRX bytes:0 unicastpkts:0 broadcastpkts:0
HWTX bytes:0 unicastpkts:0 broadcastpkts:0 dropkts: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 broadcastpkts:290555
IPTX bytes:839550 unicastpkts:11477 broadcastpkts:0 dropkts:0
HWRX bytes:0 unicastpkts:0 broadcastpkts:0
HWTX bytes:0 unicastpkts:0 broadcastpkts:0 dropkts: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 broadcastpkts:2
IPTX bytes:0 unicastpkts:0 broadcastpkts:0 dropkts:0
HWRX bytes:0 unicastpkts:0 broadcastpkts:0
HWTX bytes:0 unicastpkts:0 broadcastpkts:0 dropkts:0
=>
```

RELATED COMMANDS:

ip apadd

Assign an IP address to an IP interface.

ip aplist

Show a list of all configured 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 bcastpkts:0
IPTX bytes:0 unicastpkts:0 bcastpkts:0 dropkts:0
HWRX bytes:0 unicastpkts:0 bcastpkts:0
HWTX bytes:0 unicastpkts:0 bcastpkts:0 dropkts: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 bcastpkts:290555
IPTX bytes:839550 unicastpkts:11477 bcastpkts:0 dropkts:0
HWRX bytes:0 unicastpkts:0 bcastpkts:0
HWTX bytes:0 unicastpkts:0 bcastpkts:0 dropkts: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 bcastpkts:2
IPTX bytes:0 unicastpkts:0 bcastpkts:0 dropkts:0
HWRX bytes:0 unicastpkts:0 bcastpkts:0
HWTX bytes:0 unicastpkts:0 bcastpkts:0 dropkts:0
=>
```

RELATED COMMANDS:

ip add

Assign an IP address to an IP interface.

ip delete

Remove an IP address from an IP interface.

ip arpadd

Add an entry to the ARP cache of a broadcast IP interface.

SYNTAX:

```
ip arpadd intf = <string>
          ip = <ip-range>
          [hwaddr = <hardware-address>]
```

where:

Parameter	Description	Requirement
intf	The IP interface name.	REQUIRED
ip	The IP address [range] of the entry to be added.	REQUIRED
hwaddr	The hardware address (the Ethernet MAC address) of the entry to be added.	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** Remove an entry from the ARP cache.
- ip arplist** Show the ARP cache.

ip arpdelete

Remove an entry from the ARP cache.

SYNTAX:

```
ip arpdelete intf = <string>
            ip = <ip-range>
            [hwaddr = <hardware-address>]
```

where:

intf	The interface name.	REQUIRED
ip	The IP address [range] of the entry to be deleted.	REQUIRED
hwaddr	The hardware address (the Ethernet MAC address) of the entry to be deleted.	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 an entry to the ARP cache of a broadcast IP interface.
- ip arplist** Show the ARP cache.

ip arplist

Show the ARP cache.

SYNTAX:

```
ip arplist
```

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
=>
```

RELATED COMMANDS:

- ip arpadd** Add an entry to the ARP cache of a broadcast IP interface.
- ip arpdelete** Remove 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}>]
```

where:

forwarding	Disable (off) or enable (on) the IP routing functionality. The default is on .	OPTIONAL
firewalling	Enable (on) or disable (off) IP firewalling (master switch). If applicable, the CLI firewall level allows configuration of the SpeedTouch™ firewall. The default is on (for security reasons). Note 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. The default is off (for security reasons).	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. The default is off (for security reasons).	OPTIONAL
netbroadcasts	Disallow (off) or allow (on) net directed broadcasts. The default is off . 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: <ul style="list-style-type: none">▶ normal: Packets to be forwarded will not be reassembled. Packets with local destination (destined for the SpeedTouch™) are reassembled.▶ always: Packets are always reassembled.▶ nat: Same behavior as <i>normal</i>, 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: <ul style="list-style-type: none">▶ off: No address checking is performed. Note For advanced users only. Under 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 (for example an address may not be entirely composed of one's or zero's).▶ dynamic: Besides the address configuration of the SpeedTouch™ itself and 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 Maximum Transmission Unit (MTU) of the outgoing interface. Note 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 IP interface parameters.

ip flush

Flush the complete IP configuration.

Dynamic configurations (for example from PPP or CIP links) remain.



The flush command does not impact previously saved configurations.



As the command **:ip flush** causes all local IP connectivity to be deleted, do not use this command during an IP based local connection, such as a Telnet CLI session, or web based CLI access.

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 bcastpkts:0
IPTX bytes:0 unicastpkts:0 bcastpkts:0 dropkts:0
HWRX bytes:0 unicastpkts:0 bcastpkts:0
HWTX bytes:0 unicastpkts:0 bcastpkts:0 dropkts: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 bcastpkts:0
IPTX bytes:0 unicastpkts:0 bcastpkts:0 dropkts:0
HWRX bytes:0 unicastpkts:0 bcastpkts:0
HWTX bytes:0 unicastpkts:0 bcastpkts:0 dropkts: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 bcastpkts:2
IPTX bytes:0 unicastpkts:0 bcastpkts:0 dropkts:0
HWRX bytes:0 unicastpkts:0 bcastpkts:0
HWTX bytes:0 unicastpkts:0 bcastpkts:0 dropkts: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 bcastpkts:0
IPTX bytes:0 unicastpkts:0 bcastpkts:0 dropkts:0
HWRX bytes:0 unicastpkts:0 bcastpkts:0
HWTX bytes:0 unicastpkts:0 bcastpkts:0 dropkts: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 bcastpkts:2
IPTX bytes:0 unicastpkts:0 bcastpkts:0 dropkts:0
HWRX bytes:0 unicastpkts:0 bcastpkts:0
HWTX bytes:0 unicastpkts:0 bcastpkts:0 dropkts:0
=>
```

ip ifconfig

Configure IP interface parameters.

SYNTAX:

```
ip ifconfig intf = <string>
    [mtu = <number{68-20000}>]
    [status = <{down | up}>]
    [hwaddr = <hardware-address>]
    [group = <{wan | local | lan}> or number]
    [linksensing = <{off | on}>]
```

where:

intf	The IP interface name.	REQUIRED
mtu	A number between 68 and 20000. Represents the maximum transmission unit, in other words, 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: <ul style="list-style-type: none"> ▶ down ▶ up. 	OPTIONAL
hwaddr	The hardware address (the Ethernet MAC address) of this IP interface.	OPTIONAL
group	The group to which this interface belongs (for example for oriented firewalling).	OPTIONAL
linksensing	The IP interface's awareness of link state transitions.	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 ETHoA 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 ETHoA 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 IP interfaces.

SYNTAX:

```
ip iflist
```

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  ETHoA 0    1500  0       0        0       UP     00:80:9f:24:ab:cf
5  cip0  0    9180  0       0        0       UP
=>
```

ip ifwait

Wait for a status change of an IP interface.

SYNTAX:

```
ip ifwait intf = <string>
        [timeout = <number{1-600000}>]
        [adminstatus = <{down | up}>]
        [operstatus = <{down | up}>]
        [linkstatus = <{down | up}>]
```

where:

intf	The IP interface name.	REQUIRED
timeout	A number between 1 and 600000 (seconds). Represents the timeout.	OPTIONAL
adminstatus	The administrative state of the interface. Choose between: <ul style="list-style-type: none"> ▶ down ▶ up. 	OPTIONAL
operstatus	The operational state of the interface. Choose between: <ul style="list-style-type: none"> ▶ down ▶ up. 	OPTIONAL
linkstatus	The link state of the interface. Choose between: <ul style="list-style-type: none"> ▶ down ▶ up. 	OPTIONAL

ip mcadd

Add an MC address to an MC capable interface.

SYNTAX:

```
ip mcadd intf = <string>
        addr = <ip-address>
        [mask = <ip-mask(dotted or cidr)>]
```

where:

intf	The IP interface name.	REQUIRED
addr	The Multicast IP address.	OPTIONAL
mask	The Multicast IP address mask associated with this address.	OPTIONAL

Note Use **0** for promiscuous mode.

RELATED COMMANDS:

ip mcdelete	Delete an MC address to an MC capable interface.
ip mclist	List all MC addresses.

ip mcdelete

Delete an MC address to an MC capable interface.

SYNTAX:

```
ip mcadd intf = <string>
        addr = <ip-address>
        [mask = <ip-mask(dotted or cidr)>]
```

where:

intf	The IP interface name.	REQUIRED
addr	The Multicast IP address.	REQUIRED
mask	The Multicast IP address mask associated with this address.	OPTIONAL
Note Use 0 for promiscuous mode.		

RELATED COMMANDS:

- `ip mcadd` Add an MC address to an MC capable interface.
- `ip mclist` List all MC addresses.

ip mclist

List all MC addresses.

SYNTAX:

```
ip mclist
```

RELATED COMMANDS:

[ip mcadd](#)

Add an MC address to an MC capable interface.

[ip mcdelete](#)

Delete an MC address to an MC capable interface.

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}>]
```

where:

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
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)>]
          [label = <string>]
          [gateway = <ip-address>]
          [intf = <string>]
          [metric = <number{0-100}>]
```

where:

dst	The destination IP address(es) for this route. Supports cidr notation.	REQUIRED
dstmsk	The destination IP address mask.	OPTIONAL
label	The name of the label.	OPTIONAL
gateway	The IP address of the next hop. Must be directly connected. Note The parameters <i>gateway</i> and <i>intf</i> are mutually exclusive.	OPTIONAL
intf	Only for special interface routes: the outgoing IP interface name. Note The parameters <i>gateway</i> and <i>intf</i> are mutually exclusive.	OPTIONAL
metric	The metric for this route (weight factor).	OPTIONAL

EXAMPLE:

```
=>ip rtlist
  Destination Label      Gateway      Intf Mtrc Status
  10.0.0.0/24           10.0.0.140  eth0  0  [UP]
  10.0.0.140/32         10.0.0.140  eth0  0  [UP]
  127.0.0.1/32          127.0.0.1   loop  0  [UP]
=>ip rtadd dst=10.10.0.0/24 src=10.0.0.0/24 gateway=10.0.0.140
=>ip rtlist
  Destination Label      Gateway      Intf Mtrc Status
  10.0.0.0/24           10.0.0.140  eth0  0  [UP]
  10.10.0.0/24          10.0.0.140  eth0  0  [UP]
  10.0.0.140/32         10.0.0.140  eth0  0  [UP]
  127.0.0.1/32          127.0.0.1   loop  0  [UP]
=>
```

RELATED COMMANDS:

- [ip rdelete](#) Delete a route from the routing table.
- [ip rtlist](#) Show the current routing table.

ip rtdelete

Delete a route from the routing table.

SYNTAX:

```
ip rtdelete  dst = <ip-address>
              [dstmsk = <ip-mask(dotted or cidr)>]
              [label = <string>]
              [gateway = <ip-address>]
              [intf = <string>]
```

where:

dst	The destination IP address(es) for this route. Supports cidr notation.	REQUIRED
dstmsk	The destination IP address mask.	OPTIONAL
label	The name of the label.	OPTIONAL
gateway	The IP address of the next hop. Must be directly connected. Note The parameters <i>gateway</i> and <i>intf</i> are mutually exclusive.	OPTIONAL
intf	Only for special interface routes: the outgoing IP interface name. Note The parameters <i>gateway</i> and <i>intf</i> are mutually exclusive.	OPTIONAL

EXAMPLE:

```
=>ip rtlist
  Destination Label      Gateway      Intf Mtrc Status
  10.0.0.0/24           10.0.0.140  eth0  0  [UP]
  10.10.0.0/24          10.0.0.140  eth0  0  [UP]
  10.0.0.140/32         10.0.0.140  eth0  0  [UP]
  127.0.0.1/32          127.0.0.1   loop  0  [UP]
=>ip rtdelete dst=10.10.0.0/24 src=10.0.0.0/24 gateway=10.0.0.140
=>ip rtlist
  Destination Label      Gateway      Intf Mtrc Status
  10.0.0.0/24           10.0.0.140  eth0  0  [UP]
  10.0.0.140/32         10.0.0.140  eth0  0  [UP]
  127.0.0.1/32          127.0.0.1   loop  0  [UP]
=>
```

RELATED COMMANDS:

- ip rtadd** Add a route to the SpeedTouch™ routing table.
- ip rtlist** Show the current routing table.

ip rtlist

Show the current routing table.

SYNTAX:

```
ip rtlist
```

EXAMPLE:

```
=>ip rtlist
      Destination Label          Gateway      Intf Mtrc Status
      10.0.0.0/24                10.0.0.140   eth0    0  [UP]
      10.0.0.140/32              10.0.0.140   eth0    0  [UP]
      127.0.0.1/32               127.0.0.1    loop    0  [UP]
=>
```

RELATED COMMANDS:

[ip rtadd](#)

Add a route to the SpeedTouch™ routing table.

[ip rtdelete](#)

Delete 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}>
```

where:

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

ip traceroute

Send ICMP/UDP packets to trace the ip path.

SYNTAX:

```
ip traceroute  addr = <ip-address>
                [count = <number{1-10}>]
                [size = <number{1-20000}>]
                [interval = <number{1000-60000}>]
                [maxhops = <number{1-255}>]
                [dstport = <number{1-65535}>]
                [maxfail = <number{0-255}>]
                [type = <{icmp | udp}>]
                [utime = <{no | yes}>]
```

where:

addr	The destination IP address.	REQUIRED
count	A number between 1 and 10. Represents the number of times to reissue a traceroute request with the same TTL. The default is 3 .	OPTIONAL
size	A number between 1 and 20000 (bytes). Represents the size of the traceroute packet(s). The default is 1 .	OPTIONAL
interval	A number between 100 and 10000000 (milliseconds). Represents the intermediate interval between two packets. The default is 1000 .	OPTIONAL
maxhops	A number between 1 and 255. Represents the upper limit on the number of routers through which a packet can pass. The default is 30 .	OPTIONAL
dstport	A number between 1 and 65535. Represents the UDP destination port number to send to.	OPTIONAL
maxfail	A number between 0 and 255. Represents the maximum number of consecutive timeouts allowed before terminating a traceroute request. The default is 5 .	OPTIONAL
type	The type of traceroute packet(s). Choose between: <ul style="list-style-type: none"> ▶ icmp ▶ udp. The default is ICMP .	OPTIONAL
utime	Display time in useconds (yes) or not (no). The default is yes .	OPTIONAL

EXAMPLE:

```
=>ip traceroute addr = 192.193.195.250 count=3 size=1 interval=1000 maxhops=30 dstport=33433  
maxfail=5 type=icmp utime=yes  
:ip traceroute addr=192.193.195.250  
ttl=1 192.193.195.250 676 us 1351 us 648 us  
=>
```

ip auto flush

Flush the autoIP interfaces.

SYNTAX:

```
ip auto flush
```


ip auto ifadd

Create a new autoIP interface.

SYNTAX:

```
ip auto ifadd intf = <string>
                [addr = <ip-address>]
```

where:

intf	The name of the IP interface for which a link-local address has to be allocated.	REQUIRED
addr	The preferred link-local IP address.	OPTIONAL

RELATED COMMANDS:

- [ip auto ifdelete](#) Delete an existing autoIP interface.
- [ip auto iflist](#) Show the autoIP interfaces.

ip auto ifattach

Select and assign a link-local address to an autoIP interface.

SYNTAX:

```
ip auto ifattach intf = <string>
```

where:

intf	The name of the IP interface for which a link-local address has to be attached.	REQUIRED
------	---	----------

RELATED COMMANDS:

ip auto ifdetach Release the link-local address for the given autoIP interface.

ip auto ifconfig

Configure an autoIP interface.

SYNTAX:

```
ip auto ifconfig intf = <string>
                [addr = <ip-address>]
                [poolstart = <ip-address>]
                [poolend = <ip-address>]
                [netmask = <ip-mask(dotted or cidr)>]
                [claim = <number{0-65535}>]
                [defence = <number{0-65535}>]
                [probe = <number{0-65535}>]
                [interval = <number{1-65535}>]
```

where:

intf	The name of the autoIP interface to configure.	REQUIRED
addr	The preferred link-local IP address.	OPTIONAL
poolstart	The start IP address of the link-local address pool.	OPTIONAL
poolend	The end IP address of the link-local address pool.	OPTIONAL
netmask	The netmask of the link-local IP address pool.	OPTIONAL
claim	A number between 0 and 65535. Represents the number of link-local address selection retries before giving up. The default is 10 .	OPTIONAL
defence	A number between 0 and 65535. Represents the number of times the link-local address is defended before releasing the address. The default is 5 .	OPTIONAL
probe	A number between 0 and 65535. Represents the number of ARP probes to be sent before accepting a link-local address. The default is 4 .	OPTIONAL
interval	A number between 1 and 65535 (seconds). Represents the time interval between two ARP probe transmissions. The default is 2 .	OPTIONAL

ip auto ifdelete

Delete an existing autoIP interface.

SYNTAX:

```
ip auto ifdelete intf = <string>
```

where:

intf	The name of the IP interface to be deleted. Typically, a phonebook entry.	REQUIRED
------	--	----------

RELATED COMMANDS:

ip auto ifadd	Create a new autoIP interface.
ip auto iflist	Show the autoIP interfaces.

ip auto ifdetach

Release the link-local address for the given autoIP interface.

SYNTAX:

```
ip auto ifdetach intf = <string>
```

where:

intf	The name of the IP interface for which a link-local address has to be detached.	REQUIRED
------	---	----------

RELATED COMMANDS:

- [ip auto ifattach](#) Select and assign a link-local address to an autoIP interface.

ip auto iflist

Show the autoIP interfaces.

SYNTAX:

```
ip auto iflist [intf = <string>]
```

where:

intf The name of the interface to be listed.

OPTIONAL

Note If no name is specified, all the autoIP interfaces are shown.

EXAMPLE:

```
=>ip auto iflist
eth0      : [CLAIMED] 169.254.138.1
           poolstart = 169.254.1.1  poolend = 169.254.254.254  netmask = 255.255.0.0
           claim : 10  defence : 5  probe : 4  interval : 2 (sec)
           probes sent = 2
           collisions = 0

=>
```

RELATED COMMANDS:

[ip auto ifadd](#) Create a new autoIP interface.

[ip auto ifdelete](#) Delete an existing autoIP interface.

IPoA Commands

Introduction

This chapter describes the commands of the **ipoa** command group.

Contents

This chapter covers the following commands:

ipoa flush	Flush the complete IP over ATM (IPoA) configuration.	206
ipoa ifadd	Create a new IPoA interface.	207
ipoa ifattach	Attach an IPoA interface.	208
ipoa ifconfig	Configure an IPoA interface.	209
ipoa ifdelete	Delete an IPoA interface.	210
ipoa ifdetach	Detach an IPoA interface.	211
ipoa iflist	Show all or a specified IPoA interface(s).	212

ipoa flush

Flush the complete IP over ATM (IPoA) configuration.



The flush command does not impact previously saved configurations.

SYNTAX:

```
ipoa flush
```


ipoa ifadd

Create a new IPoA interface.

SYNTAX:

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

where:

intf	The name for the new IPoA interface.	OPTIONAL
Note If not specified, the destination will double as interface name.		
dest	The destination for the new 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 ifdelete** Delete an IPoA interface.
- ipoa iflist** Show all or a specified IPoA interface(s).

ipoa ifattach

Attach an IPoA interface.

SYNTAX:

```
ipoa ifattach intf = <string>
```

where:

intf	The name of the IPoA interface to be attached.	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 ifdetach Detach an IPoA interface.

ipoa ifconfig

Configure an IPoA interface.

SYNTAX:

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

where:

intf	The name of the IPoA interface to be configured.	REQUIRED
dest	The destination for this interface. Typically a phonebook entry. Note 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 (qosbook) entry. If not specified, the default qosbook entry will be used.	OPTIONAL
encaps	The type of encapsulation to be used for this bridge interface. Choose between: <ul style="list-style-type: none"> ▶ llc/snap (Logical Link Control (LLC)/Sub Network Access Protocol (SNAP)) ▶ vcmux (Virtual Channel MultipleXing). 	OPTIONAL
retry	A number between 0 and 65535. Represents the number of WAN connection setup retries before giving up. The default 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
IPoA_PVC1   :   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

=>
```

ipoa ifdelete

Delete an IPoA interface.

SYNTAX:

```
ipoa ifdelete intf = <string>
```

where:

intf	The name of the IPoA interface to be deleted.	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 IPoA interface.
ipoa iflist	Show all or a specified IPoA interface(s).

ipoa ifdetach

Detach an IPoA interface.

SYNTAX:

```
ipoa ifdetach intf = <string>
```

where:

intf	The name of the IPoA interface to be detached.	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 ifattach Attach an IPoA interface.

ipoa iflist

Show all or a specified IPoA interface(s).

SYNTAX:

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

where:

intf	The name of an IPoA interface.	OPTIONAL
Note	If not specified, all IPoA interfaces are listed.	

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
=>
```

RELATED COMMANDS:

- ipoa ifadd** Create a new IPoA interface.
- ipoa ifdelete** Delete an IPoA interface.

IPQoS Commands

Introduction

This chapter describes the commands of the **ipqos** command group.

Contents

This chapter covers the following commands:

<code>ipqos config</code>	Configure IP QoS for a given destination interface for the IP QoS queues instantiation.	214
<code>ipqos list</code>	List IP QoS configuration.	216
<code>ipqos queue clear</code>	Clear IP QoS statistics.	217
<code>ipqos queue config</code>	Configure IP QoS subqueues.	218
<code>ipqos queue list</code>	List IP QoS subqueue configuration.	220
<code>ipqos queue stats</code>	Show IP QoS subqueue statistics.	221

ipqos config

Configure IP QoS for a given destination interface for the IP QoS queues instantiation.



When enabling/disabling of IP QoS, take the following into account:

- ▶ if the WAN interface (for example PPPoA, IPoA, ...) is detached at the time of enabling/disabling IP QoS, then the WAN interface has to be attached in order for the enabling/disabling of IP QoS to take effect.
- ▶ if the WAN interface is attached at the time of enabling/disabling IP QoS, then the WAN interface has to be detached and then re-attached in order for the enabling/disabling of IP QoS to take effect.

SYNTAX:

```
ipqos config dest = <string>
              [state = <{disabled | enabled}>]
              [discard = <{tail | early}>]
              [realtimerate = <number{1-100}>]
              [burstsize = <number{1-64}>]
              [maxpackets = <number{0-250}>]
              [maxbytes = <number{0-64}>]
```

where:

dest	The destination interface for the IP QoS queues instantiation. Typically, a phonebook entry.	REQUIRED
state	This parameter enables or disables IP QoS for the interface. The default is disabled .	OPTIONAL
discard	Determines the packet discard strategy in case of congestion. Choose between: <ul style="list-style-type: none"> ▶ tail: Tail Drop: arriving packets will be dropped as soon as the destination queue is in an overflow state. ▶ early: Early Packet discard: arriving packets will be dropped early according to the BLUE active queue management algorithm. The default is early .	OPTIONAL
realtimerate	A number between 1 and 100. Represents a percentage of the interface bandwidth for rate-limiting of the Real Time queue. In case of congestion, the Real Time queue will only use this percentage of the interface bandwidth when there is also traffic on the other queues. The default is 80 .	OPTIONAL
burstsize	A number between 1 and 64. Represents the realtime queue burstsize (in kilobytes) for rate limiting. The default is 2 .	OPTIONAL
maxpackets	A number between 0 and 250. Represents the maximum number of packets in all IP QoS queues instantiated for one interface. The default is 250 .	OPTIONAL
maxbytes	A number between 0 and 64 Represents the maximum size in kilobytes in all IP QoS queues instantiated for one interface. The default is 56 .	OPTIONAL

EXAMPLE:

```

=>ipqos list
Name State      Discard  Packets  Kbytes  Rate  Burstsize
ATM1 disabled  early   250     56      80%    2
=>ipqos config dest=ATM1 state=enabled
=>ipqos list
Name State      Discard  Packets  Kbytes  Rate  Burstsize
ATM1 enabled  early   250     56      80%    2
=>

```

RELATED COMMANDS:

ipqos list List IP QoS configuration.

ipqos list

List IP QoS configuration.

SYNTAX:

```
ipqos list
```

EXAMPLE:

```
=>ipqos list
Name  State   Discard  Packets  Kbytes  Rate   Burstsize
ATM1  enabled early    250      56      80%    2
=>
```

RELATED COMMANDS:

[ipqos config](#) Configure IP QoS for a given destination interface for the IP QoS queues instantiation.

ipqos queue clear

Clear IP QoS statistics.

SYNTAX:

```
ipqos queue clear
```

EXAMPLE:

Example for the SpeedTouch™516

```
=>ipqos queue stats
Name Queue      # packets # packets # packets # packets # packets Marking
      added    marked   removed  dropped   replaced
PVC_1 0          1240      0         1240      0         0        0
      1           0         0         0         0         0        0
      2           0         0         0         0         0        0
      3          234      0         234      0         0        0
=>ipqos queue clear
=>ipqos queue stats
Name Queue      # packets # packets # packets # packets # packets Marking
      added    marked   removed  dropped   replaced
PVC_1 0           0         0         0         0         0        0
      1           0         0         0         0         0        0
      2           0         0         0         0         0        0
      3           0         0         0         0         0        0
=>
```

Example for the SpeedTouch™516

```
=>ipqos queue stats
Name Queue      # packets # packets # packets # packets # packets Marking
      added    marked   removed  dropped   replaced
PVC_1 0          1240      0         1240      0         0        0
      1           0         0         0         0         0        0
      2           0         0         0         0         0        0
      3          234      0         234      0         0        0
=>
```

RELATED COMMANDS:

- ipqos queue list** List IP QoS subqueue configuration.
- ipqos queue stats** Show IP QoS subqueue statistics.

ipqos queue config

Configure IP QoS subqueues.

SYNTAX:

```
ipqos queue config  dest = <string>
                    queue = <number{0-3}>
                    [propagate = <{disabled | enabled}>]
                    [maxpackets = <number{0-250}>]
                    [maxbytes = <number{0-64}>]
                    [respackets = <number{0-250}>]
                    [resbytes = <number{0-64}>]
```

where:

dest	The destination interface for the IP QoS queues instantiation. Typically, a phonebook entry.	REQUIRED
queue	A number between 0 and 3. Represents the number of the queue, where: <ul style="list-style-type: none"> ▶ 3 is the Real Time queue ▶ 2 is the High queue ▶ 1 is the Medium queue ▶ 0 is the Best Effort queue. 	OPTIONAL
propagate	Higher priority packets will be queued in a lower priority queue, instead of being dropped, as soon as the destination queue is in overflow state. The packet will be put in a lower priority queue only once. Choose between disabled or enabled. The default is <i>disabled</i> . Note The propagate flag for the lowest priority subqueue (the Best Effort queue) has no meaning.	OPTIONAL
maxpackets	A number between 0 and 250. Represents the maximum number of packets in this queue. The default is 0 for the Real Time queue and 100 for all other queues. Note 0 means that no upper limit is enforced for the number of packets in the Real Time queue.	OPTIONAL
maxbytes	A number between 0 and 64. Represents the maximum size in kilobytes of this queue. The default is 0 for the Real Time queue and 20 for all other queues. Note 0 means that no upper limit is enforced for the number of kilobytes in the Real Time queue.	OPTIONAL
respackets	A number between 0 and 250. Represents the reserved number of packets in this queue. The default is 30 for the Real Time queue and 13 for all other queues.	OPTIONAL
resbytes	A number between 0 and 64. Represents the reserved size in kilobytes of this queue. The default is 12 for the Real Time queue and 4 for all other queues.	OPTIONAL

EXAMPLE:

```

=>ipqos queue list
Name Queue Propagate Size Size Reserved Reserved
(Packets) (Kbytes) (Packets) (Kbytes)
ATM1 0 100 20 13 4
1 disabled 100 20 13 4
2 disabled 100 20 13 4
3 disabled 0 0 30 12
=>ipqos queue config dest=ATM1 queue=2 propagate=enabled maxpackets=20 maxbytes=10
=>ipqos queue list
Name Queue Propagate Size Size Reserved Reserved
(Packets) (Kbytes) (Packets) (Kbytes)
ATM1 0 100 20 13 4
1 disabled 100 20 13 4
2 enabled 20 10 13 4
3 disabled 0 0 30 12
=>

```

ipqos queue list

List IP QoS subqueue configuration.

SYNTAX:

```
ipqos queue list
```

EXAMPLE:

```
=>ipqos queue list
Name Queue      Propagate Size      Size      Reserved Reserved
          (Packets) (Kbytes)  (Packets) (Kbytes)
ATM1  0
      1          disabled 100      20       13        4
      2          disabled 100      20       13        4
      3          disabled 0        0        30       12
=>
```

RELATED COMMANDS:

ipqos queue clear Clear IP QoS statistics.

ipqos queue stats Show IP QoS subqueue statistics.

ipqos queue stats

Show IP QoS subqueue statistics.

SYNTAX:

```
ipqos queue stats
```

EXAMPLE:

```
=>ipqos queue stats
Name Queue      # packets # packets # packets # packets # packets Marking
      added   marked   removed   dropped   replaced
PVC_1 0          1240      0          1240      0          0          0
      1           0         0           0         0          0          0
      2           0         0           0         0          0          0
      3          234      0          234      0          0          0
=>
```

RELATED COMMANDS:

ipqos queue clear	Clear IP QoS statistics.
ipqos queue list	List IP QoS subqueue configuration.

Label Commands

Introduction

This chapter describes the commands of the **label** command group.

Contents

This chapter covers the following commands:

<code>label add</code>	Create a new label.	224
<code>label config</code>	Configure a label.	225
<code>label delete</code>	Delete a label.	227
<code>label flush</code>	Delete all labels that are not in use.	228
<code>label list</code>	Show the labels.	229
<code>label troff</code>	Disable verbose console messaging.	230
<code>label tron</code>	Enable verbose console messaging.	231
<code>label chain create</code>	Create a new chain.	232
<code>label chain delete</code>	Delete a chain.	233
<code>label chain flush</code>	Flush all chains.	234
<code>label chain list</code>	Show a list of all current chains.	235
<code>label rule clear</code>	Clear statistics for a given rule or all the rules.	236
<code>label rule create</code>	Create a rule.	237
<code>label rule delete</code>	Delete a rule.	239
<code>label rule flush</code>	Flush all rules created for a chain(s).	240
<code>label rule list</code>	Show a list of rules.	241
<code>label rule stats</code>	Show statistics.	242

label add

Create a new label.

SYNTAX:

```
label add name = <string>
```

where:

name	The name of the new label.	REQUIRED
-------------	----------------------------	-----------------

EXAMPLE:

```
=>label list
Name          Class      Defclass Ackclass Ttlover  Ttl      Tosmark  Tos  Use  Trace
BestEffort    increase   4         4         disabled 0        disabled 0    0    disabled
HighPriority   increase  10        10        disabled 0        disabled 0    0    disabled
MediumPriority increase   6         6         disabled 0        disabled 0    0    disabled
RealTime      increase  14        14        disabled 0        disabled 0    0    disabled
=>label add name=Label1
=>label list
Name          Class      Defclass Ackclass Ttlover  Ttl      Tosmark  Tos  Use  Trace
BestEffort    increase   4         4         disabled 0        disabled 0    0    disabled
HighPriority   increase  10        10        disabled 0        disabled 0    0    disabled
MediumPriority increase   6         6         disabled 0        disabled 0    0    disabled
RealTime      increase  14        14        disabled 0        disabled 0    0    disabled
Label1        increase   0         0         disabled 0        disabled 0    0    disabled
=>
```

RELATED COMMANDS:

- [label delete](#) Delete a label.
- [label list](#) Show the labels.

label config

Configure a label.

SYNTAX:

```
label config name = <string>
  [classification = <{ignore | overwrite | increase}>]
  [defclass = <number{0-15}>]
  [ackclass = <number{0-15}>]
  [ttloverwrite = <{disabled | enabled}>]
  [ttl = <number{0-255}>]
  [tosmarking = <{disabled | enabled}>]
  [tos = <number{0-255}>]
  [trace = <{disabled | enabled}>]
```

where:

name	The name of the label to be configured.	REQUIRED
classification	<p>Select the method of classification, in other words determine what the Layer 3 class assignment must do with the priority of the data packet (as set by Layer 2). Choose between:</p> <ul style="list-style-type: none"> ▶ ignore: Ignore the class parameters (defclass and ackclass), but use the class as set by Layer 2 (Virtual LAN (VLAN) user priority, ATM QoS). ▶ overwrite: Change the class to defclass/ackclass, overwriting the value set by Layer 2 (VLAN user priority, ATM QoS). ▶ increase: Change the class according to defclass/ackclass, but only if the defclass value is higher than the class value already set by Layer 2. <p>The default is ignore.</p> <p>Note The class as set by Layer 2 is derived from:</p> <ul style="list-style-type: none"> ▶ the VLAN user priority (in case of VLAN or priority tagged frames) ▶ the ATM PVC QoS class (in case the packet is received from an ATM PVC). <p>For non-VLAN frames or non-PVC received data, Layer 2 sets the internal class (priority) to 4 by default.</p>	OPTIONAL
defclass	<p>A number between 0 and 15. Represents the default priority class of the assigned connection. The default is 0.</p>	OPTIONAL
ackclass	<p>A number between 0 and 15. Represents the priority class of the ACK segments of the TCP connection. The default is 0.</p>	OPTIONAL
ttloverwrite	<p>Enable or disable ttl overwrite. When on forwarding the TTL field of the IP header is decremented, an Internet Group Management Protocol (IGMP) packet with TTL=1 would be dropped. To support forwarding of IGMP packets, the TTL value can be overwritten. The default is disabled.</p>	OPTIONAL

ttl	A number between 0 and 255. The TTL value to be used for ttl overwrite. The default is 0 .	OPTIONAL
tosmarking	Enable/disable tos marking. Choose between: <ul style="list-style-type: none"> ▶ disabled ▶ enabled. The default is disabled .	OPTIONAL
tos	A number between 0 and 255. Represents the ToS specification in the IP packet (used for tosmarking). The default is 0 .	OPTIONAL
trace	Enable/disable tracing for this label. Choose between: <ul style="list-style-type: none"> ▶ disabled ▶ enabled. The default is disabled .	OPTIONAL

EXAMPLE:

```
=>label list
Name      Class      Defclass Ackclass Ttlover  Ttl      Tosmark  Tos  Use  Trace
BestEffort  increase   4         4         disabled 0         disabled 0    0    disabled
HighPriority increase  10        10        disabled 0         disabled 0    0    disabled
MediumPriority increase  6         6         disabled 0         disabled 0    0    disabled
RealTime    increase  14        14        disabled 0         disabled 0    0    disabled
Labell      increase  0         0         disabled 0         disabled 0    0    disabled
=>label config name=Labell classification=increase defclass=7 ackclass=7
=>label list
Name      Class      Defclass Ackclass Ttlover  Ttl      Tosmark  Tos  Use  Trace
BestEffort  increase   4         4         disabled 0         disabled 0    0    disabled
HighPriority increase  10        10        disabled 0         disabled 0    0    disabled
MediumPriority increase  6         6         disabled 0         disabled 0    0    disabled
RealTime    increase  14        14        disabled 0         disabled 0    0    disabled
Labell      increase  7         7         disabled 0         disabled 0    0    disabled
=>
```

label delete

Delete a label.

SYNTAX:

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

where:

name	The name of the label to be deleted.	REQUIRED
force	Force delete and cleanup references even when the label is still in use. Choose between: <ul style="list-style-type: none"> ▶ no ▶ yes. The default is no .	OPTIONAL

EXAMPLE:

```
=>label list
Name          Class      Defclass Ackclass Ttlover  Ttl      Tosmark  Tos  Use  Trace
BestEffort    increase  4         4         disabled 0        disabled 0   0    disabled
HighPriority   increase  10        10        disabled 0        disabled 0   0    disabled
MediumPriority increase  6         6         disabled 0        disabled 0   0    disabled
RealTime      increase  14        14        disabled 0        disabled 0   0    disabled
Labell        increase  7         7         disabled 0        disabled 0   0    disabled
=>label delete name=Labell force=yes
=>label list
Name          Class      Defclass Ackclass Ttlover  Ttl      Tosmark  Tos  Use  Trace
BestEffort    increase  4         4         disabled 0        disabled 0   0    disabled
HighPriority   increase  10        10        disabled 0        disabled 0   0    disabled
MediumPriority increase  6         6         disabled 0        disabled 0   0    disabled
RealTime      increase  14        14        disabled 0        disabled 0   0    disabled
=>
```

RELATED COMMANDS:

- [label add](#) Create a new label.
- [label list](#) Show the labels.

label flush

Delete all labels that are not in use.



The flush command does not impact previously saved configurations.

SYNTAX:

```
label flush
```

EXAMPLE:

In the example below, the label "Label1" is not in use:

```
=>label list
Name          Class      Defclass Ackclass Ttlover  Ttl      Tosmark  Tos  Use  Trace
BestEffort    increase   4         4         disabled 0        disabled 0    0    disabled
HighPriority   increase  10        10        disabled 0        disabled 0    0    disabled
MediumPriority increase   6         6         disabled 0        disabled 0    0    disabled
RealTime      increase  14        14        disabled 0        disabled 0    0    disabled
Label1        increase   7         7         disabled 0        disabled 0    0    disabled
=>label flush
=>label list
Name          Class      Defclass Ackclass Ttlover  Ttl      Tosmark  Tos  Use  Trace
BestEffort    increase   4         4         disabled 0        disabled 0    0    disabled
HighPriority   increase  10        10        disabled 0        disabled 0    0    disabled
MediumPriority increase   6         6         disabled 0        disabled 0    0    disabled
RealTime      increase  14        14        disabled 0        disabled 0    0    disabled
=>
```

label list

Show the labels.

SYNTAX:

```
label list [name = <string>]
```

where:

name	The name of the label to be shown.	OPTIONAL
------	------------------------------------	----------

Note If not specified, all labels will be shown.

EXAMPLE:

```
=>label list
Name          Class      Defclass Ackclass Ttlover  Ttl      Tosmark  Tos  Use  Trace
BestEffort    increase  4         4         disabled 0        disabled 0  0    disabled
HighPriority   increase  10        10        disabled 0        disabled 0  0    disabled
MediumPriority increase  6         6         disabled 0        disabled 0  0    disabled
RealTime      increase  14        14        disabled 0        disabled 0  0    disabled
=>label list name=MediumPriority
Name          Class      Defclass Ackclass Ttlover  Ttl      Tosmark  Tos  Use  Trace
MediumPriority increase  6         6         disabled 0        disabled 0  0    disabled
=>
```

RELATED COMMANDS:

- [label add](#) Create a new label.
- [label delete](#) Delete a label.

label troff

Disable verbose console messaging.

SYNTAX:

```
label troff
```

RELATED COMMANDS:

[label tron](#) Enable verbose console messaging.

label tron

Enable verbose console messaging.

SYNTAX:

```
label tron
```

RELATED COMMANDS:

[label troff](#) Disable verbose console messaging.

label chain create

Create a new chain.

SYNTAX:

```
label chain create chain = <string>
```

where:

chain	The name of the new chain.	REQUIRED
-------	----------------------------	----------

EXAMPLE:

```
=>label chain list
:label chain create chain=user_labels
:label chain create chain=_auto_labels
=>label chain create chain = my_labels
=>label chain list
:label chain create chain=my_labels
:label chain create chain=user_labels
:label chain create chain=_auto_labels
=>
```

RELATED COMMANDS:

label chain delete	Delete a chain.
label chain list	Show a list of all current chains.

label chain delete

Delete a chain.

SYNTAX:

```
label chain delete chain = <string>
```

where:

chain	The name of the chain to be deleted.	REQUIRED
-------	--------------------------------------	----------

EXAMPLE:

```
=>label chain list
:label chain create chain=my_labels
:label chain create chain=user_labels
:label chain create chain=_auto_labels
=>label chain delete chain = my_labels
=>label chain list
:label chain create chain=user_labels
:label chain create chain=_auto_labels
=>
```

RELATED COMMANDS:

- label chain create Create a new chain.
- label chain list Show a list of all current chains.

label chain flush

Flush all chains.

SYNTAX:

```
label chain flush
```

label chain list

Show a list of all current chains.

SYNTAX:

```
label chain list
```

EXAMPLE:

```
=>label chain list  
:label chain create chain=user_labels  
:label chain create chain=_auto_labels  
=>
```

RELATED COMMANDS:

label chain create	Create a new chain.
label chain delete	Delete a chain.

label rule clear

Clear statistics for a given rule or all the rules.

SYNTAX:

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

where:

chain	The name of the chain in which the rule is to be found.	OPTIONAL
	Note If not specified, the statistics for all the rules in all chains will be cleared.	
index	The index number (determined by the position) of the rule in the chain.	OPTIONAL

EXAMPLE:

```
=>label rule stats chain=user_labels index=2
Chain , index 2, packets 41, bytes 2722
=>label rule clear chain=user_labels index=2
=>label rule stats chain=user_labels index=2
Chain , index 2, packets 0, bytes 0
=>
```

RELATED COMMANDS:

[label rule stats](#) Show statistics.

label rule create

Create a rule.

SYNTAX:

```
label rule create chain = <string>
                 [index = <number>]
                 [srcintf [!]= <string>]
                 [srcintfgrp [!]= <{wan | local | lan} or number>]
                 [src [!]= <ip-range>]
                 [dst [!]= <ip-address>]
                 [tos [!]= <number{0-255}>]
                 [precedence [!]= <number{0-7}>]
                 [dscp [!]= <number{0-63}>]
                 [prot = <{<supported IP protocol name> | <number>}>]
                 [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>}>]
                 [clink = <string>]
                 [log = <{no | yes}>]
                 label = <{None | link | <string>}>
```



If a value is preceded by a "!", it means "NOT".
For example "srcintfgrp=!wan" means "if srcintfgrp is different from WAN".

where:

chain	The name of the chain in which the rule must be inserted.	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 arrive on to make this rule apply.	OPTIONAL
srcintfgrp	The interface group the packet should arrive on.	OPTIONAL
src	The source IP address (range) the packet should come from. (Supports ip/mask notation).	OPTIONAL
dst	The destination IP address (range) the packet should be going to. (Supports ip/mask notation).	OPTIONAL
tos	A number between 0 and 255. Represents the ToS specification which should be expected in the IP packet. The ToS numbering specification is in accordance to the latest version of <i>RFC1700: Assigned numbers</i> .	OPTIONAL
precedence	A number between 0 and 7. Represents the precedence in the IP packet (is part of tos).	OPTIONAL
dscp	A number between 0 and 63. Represents the DSCP in the IP packet (part of tos).	OPTIONAL

prot	The protocol (name or number) expected in the IP packet. Select one of the following protocol names: <ul style="list-style-type: none"> ▶ icmp ▶ igmp ▶ ipinip ▶ tcp ▶ udp ▶ ah ▶ esp ▶ ipcomp or, alternatively, specify the protocol number.	OPTIONAL
srcport	The TCP/UDP port (or beginning of range) the packet is coming from. Select one of the supported TCP/UDP port names (see “ Supported TCP/UDP Port Names” on page 375) or, alternatively, specify the port 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 375) or, alternatively, specify the port number.	OPTIONAL
dstport	The TCP/UDP port (or beginning of range) the packet is going to. Select one of the supported TCP/UDP port names (see “ Supported TCP/UDP Port Names” on page 375) or, alternatively, specify the port 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 375) or, alternatively, specify the port number.	OPTIONAL
clink	The name of the chain to be parsed when this rule applies.	OPTIONAL
log	Logging is done when this rule applies. Choose between no or yes.	OPTIONAL
label	Choose between: <ul style="list-style-type: none"> ▶ None ▶ link (when clink is used) ▶ label name. 	REQUIRED

RELATED COMMANDS:

label rule delete

Delete a rule.

label rule list

Show a list of rules.

label rule delete

Delete a rule.

SYNTAX:

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

where:

chain	The name of the chain in which a rule must be deleted.	REQUIRED
index	The index number of the rule in the chain.	REQUIRED
Tip	Use the command :label rule list to obtain the index number of the rule.	

EXAMPLE:

```
=>label rule list
:label rule create chain=user_labels index=0 srcintfgrp=lan prot=tcp dstport=telnet label=Test
:label rule create chain=user_labels index=1 srcintfgrp=lan prot=tcp dstport=smtp label=Test
:label rule create chain=user_labels index=2 srcintfgrp=lan prot=tcp dstport=imap3 label=Test
:label rule create chain=user_labels index=3 srcintfgrp=lan prot=tcp dstport=imap2 label=Test
:label rule create chain=user_labels index=4 srcintfgrp=lan prot=tcp dstport=http label=Test
:label rule create chain=user_labels index=5 srcintfgrp=lan prot=tcp dstport=1080 label=Test
:label rule create chain=user_labels index=6 srcintfgrp=lan prot=tcp dstport=www-http label=Test
=>label rule delete chain=user_labels index=2
=>label rule list chain=user_labels
:label rule create chain=user_labels index=0 srcintfgrp=lan prot=tcp dstport=telnet label=Test
:label rule create chain=user_labels index=1 srcintfgrp=lan prot=tcp dstport=smtp label=Test
:label rule create chain=user_labels index=2 srcintfgrp=lan prot=tcp dstport=imap2 label=Test
:label rule create chain=user_labels index=3 srcintfgrp=lan prot=tcp dstport=http label=Test
:label rule create chain=user_labels index=4 srcintfgrp=lan prot=tcp dstport=1080 label=Test
:label rule create chain=user_labels index=5 srcintfgrp=lan prot=tcp dstport=www-http label=Test
=>
```

RELATED COMMANDS:

- label rule create** Create a rule.
- label rule list** Show a list of rules.

label rule flush

Flush all rules created for a chain(s).



The flush command does not impact previously saved configurations.

SYNTAX:

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

where:

chain	The name of the chain to be flushed.	OPTIONAL
-------	--------------------------------------	----------

Note If not specified, all rules for all chains are flushed.

label rule list

Show a list of rules.

SYNTAX:

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

where:

chain	The name of the chain for which the rules must be listed.	OPTIONAL
Note	If not specified, all rules for all chains are shown.	

EXAMPLE INPUT AND OUTPUT:

```
=>label rule list chain=user_labels
:label rule create chain=user_labels index=0 srcintfgrp=lan prot=tcp dstport=telnet label=Test
:label rule create chain=user_labels index=1 srcintfgrp=lan prot=tcp dstport=smtp label=Test
:label rule create chain=user_labels index=2 srcintfgrp=lan prot=tcp dstport=imap3 label=Test
:label rule create chain=user_labels index=3 srcintfgrp=lan prot=tcp dstport=imap2 label=Test
:label rule create chain=user_labels index=4 srcintfgrp=lan prot=tcp dstport=http label=Test
:label rule create chain=user_labels index=5 srcintfgrp=lan prot=tcp dstport=1080 label=Test
:label rule create chain=user_labels index=6 srcintfgrp=lan prot=tcp dstport=www-http label=Test
=>
```

RELATED COMMANDS:

- label rule create Create a rule.
- label rule delete Delete a rule.

label rule stats

Show statistics.

SYNTAX:

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

where:

chain	The name of the chain for which the statistics must be listed.	OPTIONAL
	Note If not specified, the statistics for the rules applicable to all chains are shown.	
index	The index number of the rule for which the statistics must be listed.	OPTIONAL
	Tip Use the command <code>:label rule list</code> to obtain the index number of the rule.	
	Note If not specified, the statistics for all rules applicable to the specified chain are shown.	

EXAMPLE:

```
=>label rule stats
Chain user_labels, index 0, packets 25, bytes 3585
Chain user_labels, index 1, packets 0, bytes 0
Chain user_labels, index 2, packets 0, bytes 0
Chain user_labels, index 3, packets 0, bytes 0
Chain user_labels, index 4, packets 0, bytes 0
Chain user_labels, index 5, packets 0, bytes 0
Chain user_labels, index 6, packets 41, bytes 2722
Chain user_labels, index 7, packets 0, bytes 0
Chain user_labels, index 8, packets 0, bytes 0
Chain user_labels, index 9, packets 0, bytes 0
Chain user_labels, index 10, packets 0, bytes 0
Chain_auto_labels, index 0, packets 10, bytes 440
=>label rule stats chain=user_labels index=6
Chain , index 6, packets 41, bytes 2722
=>
```

RELATED COMMANDS:

`label rule clear` Clear statistics for a given rule or all the rules.

Language Commands

Introduction

This chapter describes the commands of the **language** command group.

Contents

This chapter covers the following commands:

language config	Select a language.	244
language list	List the available language archives.	245
language remove	Remove one or all language archives.	246

language config

Select a language.

SYNTAX:

```
language config [language = <string>]
```

where:

language	Language code: OSI language code (2 chars) for language. Example: en for english.	OPTIONAL
----------	---	----------

EXAMPLE:

```
=>language config  
language      : en  
=>
```

RELATED COMMANDS:

language list	List the available language archives.
language remove	Remove one or all language archives.

language list

List the available language archives.

SYNTAX:

```
language list
```

EXAMPLE:

```
=>language list
CODE LANGUAGE          VERSION  FILENAME
en* English           5.2.7.5.0  <system>
=>
```



The currently selected language is indicated by a "*" next to the OSI language code.

RELATED COMMANDS:

- language config Select a language.
- language remove Remove one or all language archives.

language remove

Remove one or all language archives.

SYNTAX:

```
language remove [file = <string>]
                [all = <{yes | no}>]
```

where:

file	The filename of the language archive to be removed.	OPTIONAL
all	Remove all languages archives (yes) or not (no). The default is <i>no</i> .	OPTIONAL

RELATED COMMANDS:

language config	Select a language.
language list	List the available language archives.

NAT Commands

Introduction

This chapter describes the commands of the **nat** command group.

Contents

This chapter covers the following commands:

nat applist	List NAT/PAT connection database.	248
nat bind	Create a new helper/port binding.	249
nat bindlist	List current NAT/PAT helper/port bindings.	250
nat clear	Clear NAT/PAT connection database.	251
nat config	Configure NAT parameters for IP session.	252
nat create	Create a static NAT/PAT entry.	253
nat defserver	Define the default NAT/PAT server.	254
nat delete	Delete a static NAT/PAT entry.	255
nat disable	Delete a static NAT/PAT entry.	256
nat enable	Delete a static NAT/PAT entry.	257
nat flush	Delete a static NAT/PAT entry.	258
nat list	Delete a static NAT/PAT entry.	259
nat multinatadd	Delete a static NAT/PAT entry.	260
nat multinatdelete	Delete a static NAT/PAT entry.	261
nat multinatlist	Delete a static NAT/PAT entry.	262
nat unbind	Delete a static NAT/PAT entry.	263

nat applist

List NAT/PAT connection database.

Certain protocols are 'sensitive' to Network Address Translation (NAT) / Port Address Translation (PAT) in that they do not function properly when dealing with it. This list shows which 'NAT/PAT-sensitive' applications are supported on the SpeedTouch™, that is the inherent knowledge of the SpeedTouch™ on this matter.

SYNTAX:

```
nat applist
```

EXAMPLE:

```
=>nat applist
Application  Proto DefaultPort
IP6TO4      6to4      1   OUTGOING
GRE         gre        1   INCOMING
PPTP       tcp       1723 OUTGOING INCOMING
ESP        esp        1   OUTGOING INCOMING
IKE        udp        500  OUTGOING INCOMING
SIP        udp       5060 OUTGOING INCOMING
JABBER     tcp       15222 OUTGOING
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 INCOMING
=>
```

nat bind

Create a new helper/port binding.

SYNTAX:

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

where:

application	The name of a NAT/PAT application helper. The name must be spelled exactly as listed in the application list. Tip Use the command <code>:nat applist</code> to obtain the list of applications.	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 375) or, alternatively, specify the port number.	REQUIRED
port_end	The TCP/UDP end port of the range of ports this application handler should work on. Select one of the supported TCP/UDP port names (see "Supported TCP/UDP Port Names" on page 375) or, alternatively, specify the port number.	OPTIONAL

EXAMPLE:

```
=>nat bindlist
Application Proto DefaultPort
SIP         udp      5060
...
FTP         tcp       21
IRC         tcp      6667
RAUDIO(PNA) tcp      7070
=>nat bind application=RAUDIO(PNA) port=7071
=>nat bindlist
Application Proto DefaultPort
SIP         udp      5060
...
FTP         tcp       21
IRC         tcp      6667
RAUDIO(PNA) tcp      7070
RAUDIO(PNA) tcp      7071
=>
```

RELATED COMMANDS:

- `nat bindlist` List current NAT/PAT helper/port bindings.
- `nat unbind` Delete an existing helper/port binding.

nat bindlist

List current NAT/PAT helper/port bindings.

SYNTAX:

```
nat bindlist
```

EXAMPLE:

```
=>nat bindlist
Application  Proto Port
ESP          esp   1
FTP          tcp   21
GRE          gre   1
H323        tcp   1720
IKE          udp   500
ILS          tcp   389
ILS          tcp   1002
IP6TO4      6to4  1
IRC          tcp   6660-6670
JABBER       tcp   5222
JABBER       tcp   15222
PPTP        tcp   1723
RAUDIO (PNA) tcp   7070
RTSP        tcp   554
SIP         udp   5060
=>
```

RELATED COMMANDS:

[nat bind](#)

Create a new helper/port binding.

[nat unbind](#)

Delete an existing helper/port binding.

nat clear

Clear NAT/PAT connection database.

SYNTAX:

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

where:

addr	An address-translation enabled IP address.	OPTIONAL
	Note If not specified, all IP addresses are cleared.	

nat config

Configure NAT parameters for IP session.

SYNTAX:

```
nat config [timeout_ICMP = <string>]
           [timeout_UDP = <string>]
           [timeout_TCP = <string>]
           [timeout_TCP_nego = <string>]
           [timeout_other = <string>]
           [ike_port = <{fixed | floated}>]
```

where:

timeout_ICMP	The interval of time for which an ICMP protocol session is allowed to remain valid without any activity. Syntax: <HH:MM> or <MM>.	OPTIONAL
timeout_UDP	The interval of time for which an UDP protocol session is allowed to remain valid without any activity. Syntax: <HH:MM> or <MM>.	OPTIONAL
timeout_TCP	The interval of time for which an TCP protocol session is allowed to remain valid without any activity. This timeout value applies to a TCP session during its data transfer phase. Syntax: <HH:MM> or <MM>.	OPTIONAL
timeout_TCP_nego	The interval of time for which an TCP protocol session is allowed to remain valid without any activity. This timeout value applies to a TCP session during its establishment and termination phases. Syntax :<HH:MM> or <MM>.	OPTIONAL
timeout_other	The interval of time for which an IP session for a protocol other than ICMP, UDP and TCP is allowed to remain valid without any activity. Syntax: <HH:MM> or <MM>.	OPTIONAL
ike_port	The IKE outside port number. This is the translation port number for IKEv1 (fixed to 500 or floated).	OPTIONAL

EXAMPLE:

```
=>nat config
Timeout ICMP idle       : 00h01m
Timeout UDP idle       : 00h05m
Timeout TCP idle       : 00h15m
Timeout TCP negotiation idle : 00h02m
Timeout Other idle     : 00h01m
IKE outside port number : Floated
=>
```

nat create

Create a static NAT/PAT entry.

Typically used to install specific servers behind the SpeedTouch™ NAT/PAT 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>}>]
    
```

where:

protocol	The IP protocol name (or number) of the incoming stream. Select one of the supported protocol names (see " Supported IP Protocol Names " on page 374) or ,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 NAT/PAT 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 375) or ,alternatively, specify the port number. Note 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, that is the NAT/PAT enabled WAN IP address of the SpeedTouch™. Use 0 to create a template. Such a template will then be valid for any of the SpeedTouch™ 's NAT/PAT enabled IP addresses, for example 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 375) or ,alternatively, specify the port number. Note 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-NAT/PAT routable address. Use 0 to match all foreign addresses.	OPTIONAL
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 375) or ,alternatively, specify the port number. Do not use 0 in case a foreign IP address is specified. Note Applicable for TCP and UDP protocols only. Other protocols do not need a port to be specified.	OPTIONAL

RELATED COMMANDS:

nat delete	Delete a static NAT/PAT entry.
nat list	Lists NAT/PAT connection database.

nat defserver

Define the default NAT/PAT server.

This server will handle all incoming calls.

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 the command **:nat create**.

SYNTAX:

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

where:

addr	The IP address of the server (on the 'inside') that will receive all (unknown) incoming packets.	OPTIONAL
------	--	----------

Note If not specified, the current default server is shown.

EXAMPLE:

```
=>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 NAT/PAT 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>}>]
    
```

where:

protocol	The IP protocol name (or number) of the incoming stream. Select one of the supported protocol names (see “ Supported IP Protocol Names” on page 374) or, 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 NAT/PAT 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 375) or, alternatively, specify the port number. Note 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, that is the NAT/PAT enabled WAN IP address of the SpeedTouch™ . Use 0 to delete an entry valid for any of the SpeedTouch™ 's NAT/PAT enabled IP addresses, for example 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 375) or, alternatively, specify the port number. Note 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-NAT/PAT 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 375) or, alternatively, specify the port number. Note Applicable for TCP and UDP protocols only. Other protocols do not need a port to be specified.	OPTIONAL

RELATED COMMANDS:

- nat create Create a static NAT/PAT entry.
- nat list Lists NAT/PAT connection database.

nat disable

Disable NAT or PAT on the specified SpeedTouch™ IP address.

SYNTAX:

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

where:

addr	The SpeedTouch™ IP address for which NAT/PAT 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 enable

Enable NAT/PAT on a SpeedTouch™ IP address.

nat enable

Enable NAT/PAT on a SpeedTouch™ IP address.

SYNTAX:

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

where:

addr	The IP address to use for outgoing address translation. Must be one of the own IP addresses of the SpeedTouch™.	REQUIRED
Tip	Use the command :ip aplist to obtain a list of all configured IP addresses.	
type	Enable port translation (pat) or not (none).	OPTIONAL

EXAMPLE:

```
=>ip aplist
1 eth0 Type:Ethernet HWaddr 00:80:9f:24:ab:cf BRWaddr 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 broadcastpkts:290555
IPTX bytes:839550 unicastpkts:11477 broadcastpkts:0 dropkts:0
HWRX bytes:0 unicastpkts:0 broadcastpkts:0
HWTX bytes:0 unicastpkts:0 broadcastpkts:0 dropkts: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 broadcastpkts:2
IPTX bytes:0 unicastpkts:0 broadcastpkts:0 dropkts:0
HWRX bytes:0 unicastpkts:0 broadcastpkts:0
HWTX bytes:0 unicastpkts:0 broadcastpkts:0 dropkts:0
=>nat enable addr=10.10.10.147 type=pat
=>ip aplist
1 eth0 Type:Ethernet HWaddr 00:80:9f:24:ab:cf BRWaddr 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 broadcastpkts:290555
IPTX bytes:839550 unicastpkts:11477 broadcastpkts:0 dropkts:0
HWRX bytes:0 unicastpkts:0 broadcastpkts:0
HWTX bytes:0 unicastpkts:0 broadcastpkts:0 dropkts: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 broadcastpkts:2
IPTX bytes:0 unicastpkts:0 broadcastpkts:0 dropkts:0
HWRX bytes:0 unicastpkts:0 broadcastpkts:0
HWTX bytes:0 unicastpkts:0 broadcastpkts:0 dropkts:0
=>
```

RELATED COMMANDS:

nat disable

Disable NAT or PAT on the specified SpeedTouch™ IP address.

nat flush

Flush complete NAT/PAT configuration.



The flush command does not impact previously saved configurations.

SYNTAX:

```
nat flush
```

nat list

Lists NAT/PAT connection database.

SYNTAX:

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

where:

addr	An address-translation enabled IP address.	OPTIONAL
	Note If not specified, all IP addresses are shown.	

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
=>
```

RELATED COMMANDS:

- nat create** Create a static NAT/PAT entry.
- nat delete** Delete a static NAT/PAT entry.

nat multinatadd

Add a MultiNAT configuration.

SYNTAX:

```

nat multinatadd  intf = <string>
                  inside_addr = <ip-range>
                  outside_addr = <ip-range>
                  [addroute = <{no | yes}>]
  
```

where:

Parameter	Description	Requirement
intf	The IP interface name where MultiNAT has to be applied.	REQUIRED
inside_addr	The IP address of the host behind the address translation device that will receive the incoming traffic. Typically, a private IP address.	REQUIRED
outside_addr	The apparent host IP address this application will running on. Use <i>0</i> to create a template.	REQUIRED
addroute	Add multinat label routes automatically or not. The default is yes .	OPTIONAL

EXAMPLE:

```

=>ip rtlist
      Destination Label          Gateway      Intf Mtrc Status
169.254.141.11/32      169.254.141.11 eth0 0 [UP]
255.255.255.255/32    10.0.0.138      eth0 0 [UP]
10.0.0.138/32         10.0.0.138      eth0 0 [UP]
127.0.0.1/32          127.0.0.1       loop 0 [UP]
10.0.0.0/24           10.0.0.138      eth0 0 [UP]
169.254.0.0/16        169.254.141.11 eth0 0 [UP]
224.0.0.0/4           10.0.0.138*     eth0 0 [UP]
=>nat multinatadd intf=PPPoE_1 inside_addr=10.0.0.1 outside_addr=10.10.0.138
=>nat multinatlist
Indx  Intf      Inside-address  Outside-address  Static MultiNAT
 64   PPPoE_1    10.0.0.1        10.10.0.138      Static MultiNAT
=>ip rtlist
      Destination Label          Gateway      Intf Mtrc Status
0.0.0.0/0              _from_10.0.0.1/32 10.10.0.138  PPPoE_1 0 [UP]
169.254.141.11/32     169.254.141.11 eth0 0 [UP]
255.255.255.255/32    10.0.0.138      eth0 0 [UP]
10.0.0.138/32         10.0.0.138      eth0 0 [UP]
127.0.0.1/32          127.0.0.1       loop 0 [UP]
10.0.0.0/24           10.0.0.138      eth0 0 [UP]
169.254.0.0/16        169.254.141.11 eth0 0 [UP]
224.0.0.0/4           10.0.0.138*     eth0 0 [UP]
=>
  
```

RELATED COMMANDS:

nat multinatdelete	Delete a MultiNAT configuration.
nat multinatlist	List MultiNAT configurations.

nat multinatdelete

Delete a MultiNAT configuration.

SYNTAX:

```
nat multinatdelete    index = <number>
```

where:

index	The MultiNAT index as listed by maplist.	REQUIRED
-------	--	----------

EXAMPLE:

```
=>nat multinatlist
Indx  Intf      Inside-address  Outside-address
  64  PPPoE_1    10.0.0.1       10.10.0.138    Static MultiNAT
=>nat multinatdelete index=64
=>nat multinatlist
Indx  Intf      Inside-address  Outside-address
=>
```

RELATED COMMANDS:

- nat multinatadd** Add a MultiNAT configuration.
- nat multinatlist** List MultiNAT configurations.

nat multinatlist

List MultiNAT configurations.

SYNTAX:

```
nat multinatlist [intf = <string>]
```

where:

intf	The IP interface name for which the MultiNAT configuration must be listed.	OPTIONAL
------	--	----------

Note If not specified, all MultiNAT configurations are listed.

EXAMPLE:

```
=>nat multinatlist
Indx  Intf      Inside-address  Outside-address  Static MultiNAT
  64   PPPoE_1    10.0.0.1       10.10.0.138
=>
```

RELATED COMMANDS:

nat multinatadd	Add a MultiNAT configuration.
nat multinatdelete	Delete a MultiNAT configuration.

nat unbind

Delete an existing helper/port binding.

SYNTAX:

```

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

where:

application	The name of a NAT/PAT application helper. The name must be spelled exactly as listed in the application list. Tip Use the command <code>:nat applist</code> to list the names of the NAT/PAT application helpers.	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 375) or, alternatively, specify the port number.	REQUIRED
port_end	The TCP/UDP end port of the range of ports this application handler should work on. Select one of the supported TCP/UDP port names (See “Supported TCP/UDP Port Names” on page 375) or, alternatively, specify the port number.	OPTIONAL

EXAMPLE:

```

=>nat bindlist
Application Proto Port
ESP         esp    1
...
RAUDIO(PNA) tcp    7070
RAUDIO(PNA) tcp    7072-7075
RTSP        tcp    554
SIP         udp    5060
=>nat unbind application=RAUDIO(PNA) port=7072 port_end=7075
=>nat bindlist
Application Proto Port
ESP         esp    1
...
RAUDIO(PNA) tcp    7070
RTSP        tcp    554
SIP         udp    5060
=>
    
```

RELATED COMMANDS:

- nat bind** Create a new helper/port binding.
- nat bindlist** List current NAT/PAT helper/port bindings.

Phonebook Commands

Introduction

This chapter describes the commands of the **phonebook** command group.

Contents

This chapter covers the following commands:

<code>phonebook add</code>	Add a phonebook entry.	266
<code>phonebook delete</code>	Remove a phonebook entry.	267
<code>phonebook flush</code>	Flush complete phonebook.	268
<code>phonebook list</code>	Show the current phonebook.	269

phonebook add

Add a phonebook entry.

SYNTAX:

```
phonebook add name = <string>
              addr = <atmchannel : PVC syntax is [port.]vpi.vci
                    port=dsl10 | dsl11 | ...>
              type = <{any | ethoa | pppoa | ipoa}>
```

where:

name	The name of the new phonebook entry. Two limitations apply: <ul style="list-style-type: none"> ▶ 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, for example 'DHCP_Spoof01'. 	REQUIRED
addr	The ATM address for this destination. It is composed of a VPI and a 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: <ul style="list-style-type: none"> ▶ any: All Packet Services ▶ ethoa: Bridged Ethernet, Routed Ethernet, Bridged PPPoE and Routed PPPoE ▶ pppoa: Routed PPPoA and Relayed PPPoA ▶ ipoa: Classical IPoA and Routed IPoA. 	REQUIRED

EXAMPLE:

```
=>phonebook list
Name      Type   Use   Address
PVC1     any    1     8.35
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
PVC1     any    1     8.35
Br4      ethoa  0     8.38
CIPPVC3  ipoa   1     8.82
PVC_Test pppoa  0     8.68
=>
```



The `usb_port` phonebook entry is only applicable in case of a SpeedTouch™536 variant. Do not change this entry in any way. Do not delete this entry from the phonebook!

RELATED COMMANDS:

- `phonebook delete` Remove a phonebook entry.
- `phonebook list` Show the current phonebook.

phonebook delete

Remove a phonebook entry.



This command is only applicable for phonebook entries that are not used by a packet service.

SYNTAX:

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

where:

name	The name of the phonebook entry to be deleted.	REQUIRED
Tip	Use the command :phonebook list to check whether the entry is in use (<i>Use=1</i>) or not (<i>Use=0</i>).	

EXAMPLE:

```
=>phonebook list
Name      Type   Use   Address
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
PVC1      any    1     8.35
PVC2      ethoa  0     8.36
Br4       ethoa  0     8.38
CIPPVC3   ipoa   1     8.82
=>
```

RELATED COMMANDS:

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

phonebook flush

Flush complete phonebook.



1. Phonebook entries that are in use, cannot be flushed.
2. The flush command does not impact previously saved configurations.

SYNTAX:

```
phonebook flush
```

EXAMPLE:

```
=>phonebook list
Name      Type    Use    Address
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
=>
```

phonebook list

Show the current phonebook.

SYNTAX:

```
phonebook list
```

EXAMPLE:

```
=>phonebook list
Name      Type    Use    Address
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
=>
```

RELATED COMMANDS:

- `phonebook add` Add a phonebook entry.
- `phonebook delete` Remove a phonebook entry.

PPPoA Commands

Introduction

This chapter describes the commands of the **pppoa** command group.

Contents

This chapter covers the following commands:

<code>pppoa flush</code>	Flush the current PPPoA configuration.	272
<code>pppoa ifadd</code>	Create a new PPPoA interface.	273
<code>pppoa ifattach</code>	Attach a PPPoA interface.	274
<code>pppoa ifconfig</code>	Configure a PPPoA interface.	275
<code>pppoa ifdelete</code>	Delete a PPPoA interface.	279
<code>pppoa ifdetach</code>	Detach a PPPoA interface.	280
<code>pppoa iflist</code>	Show current configuration of a specific or all PPPoA interface(s).	281
<code>pppoa rtadd</code>	Automatically add a route configuration to the routing table.	282
<code>pppoa rtdelete</code>	Delete the route specification for a PPPoA link.	284

pppoa flush

Flush the current PPPoA configuration.



The flush command does not impact previously saved configurations.

SYNTAX:

```
pppoa flush
```

EXAMPLE:

```
=>pppoa iflist
pppoa1: dest : pppoa1
  Retry: 10 QoS default encaps VC-MUX
  mode = IP Routing
  flags = echo magic accomp mru addr route savepwd pppoa0A
  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 =
=>pppoa flush
=>pppoa iflist
=>
```

pppoa ifadd

Create a new PPPoA interface.

SYNTAX:

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

where:

intf	The name for the new PPP over ATM (PPPoA) interface. Note If not specified, the destination will double as interface name.	OPTIONAL
dest	The ATM channel to be used for this PPPoA interface. This is a phonebook entry.	OPTIONAL

EXAMPLE:

```
=>pppoa iflist
pppoa: dest : pppoa [00:00:00]
  Retry : 10 QoS default encaps VC-MUX
  mode = IP routing
  flags = echo magic accomp restart mru addr route savepwd
  dns metric = 0 trans addr = pat mru = 1500
  route : dst=0.0.0.0/0 - src=10.0.0.0/1 (metric 1)
  auth = auto user = johndoe@ISP password = *****
  admin state = up oper state = up link state = connected
  LCP : state = starting retransm = 1 term. reason =
  IPCP: state = initial retransm = 0 term. reason =

=>pppoa ifadd intf=pppoa2 dest=pppoa2
=>pppoa iflist
pppoa: dest : pppoa [00:00:00]
  Retry : 10 QoS default encaps VC-MUX
  mode = IP routing
  flags = echo magic accomp restart mru addr route savepwd
  dns metric = 0 trans addr = pat mru = 1500
  route : dst=0.0.0.0/0 - src=10.0.0.0/1 (metric 1)
  auth = auto user = johndoe@ISP password = *****
  admin state = up oper state = up link state = connected
  LCP : state = starting retransm = 1 term. reason =
  IPCP: state = initial retransm = 0 term. reason =

pppoa2: dest : pppoa2 [00:00:00]
  Retry : 10 QoS default encaps VC-MUX
  mode = IP routing
  flags = echo magic accomp restart mru addr savepwd
  dns metric = 0 mru = 1500
  auth = auto user = 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:

- pppoa ifdelete** Delete a PPPoA interface.
- pppoa iflist** Show current configuration of a specific or all PPPoA interface(s).

pppoa ifattach

Attach a PPPoA interface.

SYNTAX:

```
pppoa ifattach intf = <string>
```

where:

intf	The name of the PPPoA interface to be attached.	REQUIRED
------	---	----------

EXAMPLE:

```
=>pppoa iflist
pppoal: dest : pppoal    [00:00:00]
  Retry : 10  QoS default  encaps VC-MUX
  mode = IP routing
  flags = echo magic accomp restart mru addr savepwd
  dns metric = 0  mru = 1500
  auth = auto  user =  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 =

=>pppoa ifattach intf=pppoal
=>pppoa iflist
pppoal: dest : pppoal    [00:00:00]
  Retry : 10  QoS default  encaps VC-MUX
  mode = IP routing
  flags = echo magic accomp restart mru addr savepwd
  dns metric = 0  mru = 1500
  auth = auto  user =  password =
  admin state = up  oper state = up  link state = connected
  LCP : state = starting  retransm = 1  term. reason =
  IPCP: state = initial  retransm = 0  term. reason =

=>
```

RELATED COMMANDS:

[pppoa ifdetach](#)

Detach a PPPoA interface.

pppoa ifconfig

Configure a PPPoA interface.

SYNTAX:

```
pppoa ifconfig intf = <intfname>
                [dest = <phonebook entry>]
                [user = <string>]
                [password = <password>]
                [qos = <{default}>]
                [encaps = <{vcmux | llc}>]
                [pcomp = <{off | on}>]
                [accomp = <{on | off | negotiate}>]
                [trace = <{off | on}>]
                [auth = <{pap | chap | auto}>]
                [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 = <{cidr | dotted | none}>]
                [pool = <{none}>]
                [savepwd = <{off | on}>]
                [demanddial = <{off | on}>]
                [primdns = <ip-address>]
                [secdns = <ip-address>]
                [dnsmetric = <number{0-100}>]
                [idle = <number{0-1000000}>]
                [idletrigger = <{RxTx | Rx | Tx}>]
                [addrtrans = <{none | pat}>]
                [unnumbered = <{off | on}>]
```

where:

intf	The name of the PPPoA interface to be configured.	REQUIRED
dest	The destination for this PPPoA interface. Typically, a phonebook entry.	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 qosbook entry defining the QoS parameters for the WAN link.	OPTIONAL
encaps	The WAN protocol encapsulation to be used on this interface. Choose between: <ul style="list-style-type: none"> ▶ vcmux ▶ llc. The default is vcmux .	OPTIONAL

pcomp	Try (on) or do not try (off) to negotiate PPPoA protocol compression (LCP PCOMP). The default is off .	OPTIONAL
accomp	Try (on), do never try (off) or negotiate (negotiate) to negotiate PPPoA address & control field compression (LCP ACCOMP). In most cases, LCP ACCOMP should not be disabled nor negotiated, i.e. the address field FF-03 should not be sent over ATM. The default is on . If the accomp parameter is set to 'negotiate', the local side of the PPPoA connection demands to do ACCOMP and adapts itself to the result of this negotiation.	OPTIONAL
trace	Enable (on) or disable (off) verbose console logging. The default is off .	OPTIONAL
auth	Select the authentication protocol. Choose between: <ul style="list-style-type: none"> ▶ pap: Password Authentication Protocol (PAP) authentication will be forced ▶ chap: Challenge Handshake Authentication Protocol (CHAP) authentication will be forced ▶ auto: CHAP authentication will be used. If CHAP authentication is not successful, PAP authentication will be used instead. The default is auto .	OPTIONAL
restart	Automatically restart the connection when Link Control Protocol (LCP) link goes down (on) or do not restart automatically (off). The default is on .	OPTIONAL
retryinterval	A number between 0 and 65535 (seconds). Represents the intermediate interval between two retries to establish the connection on ATM level. The default is 10 .	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. The default is off .	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). The default is off .	OPTIONAL
echo	Send LCP echo requests at regular intervals (on) or not (off). The default is on .	OPTIONAL
mru	A number between 293 and 8192. Represents the maximum packet size the SpeedTouch™ should negotiate to be able to receive. The default is 1500 .	OPTIONAL
laddr	The local IP address of the peer-to-peer connection. Specifying a local IP address forces the remote side of the PPPoA link (if it allows to) to accept this IP address as the SpeedTouch™ PPPoA 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 PPPoA link (if it allows to) to accept this IP address as its PPPoA 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 PPPoA 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, that is 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 Classless Inter Domain Routing (CIDR) format (cidr). The default is <i>cidr</i> .	OPTIONAL
pool	The name of the free DHCP server pool to which the acquired IP subnet must be assigned.	OPTIONAL
savepwd	Save password (on), if supplied, or do not save the password (off). The default is <i>on</i> .	OPTIONAL
demanddial	Enable (on) or disable (off) the dial-on-demand feature. Nothing happens until packets are sent to this PPP interface. The default is <i>off</i> .	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
dnsmetric	A number between 0 and 100. Represents the DNS route metric to be used for the negotiated DNS servers. The default is <i>0</i> .	OPTIONAL
idle	A number between 0 and 1000000 (seconds). Represents after how many seconds an idle link goes down. The default is <i>0</i> .	OPTIONAL
idletrigger	Consider the link being idle if no traffic is sent and/or received during the idle time. Choose between: <ul style="list-style-type: none"> ▶ RxTx: The idle time period restarts when a packet is transmitted or received. ▶ Rx: The idle time period restarts when a packet is received. Transmitted packets are ignored ▶ Tx: The idle time period restarts when a packet is transmitted. Received packets are ignored. The default is RxTx .	OPTIONAL
addrtrans	Automatically enable address translation for the IP address of this link (pat) or do not use address translation (none). The default is <i>none</i> .	OPTIONAL
unnumbered	Takes the local IP address from the <i>laddr</i> field and remote IP address from the IP address pool assigned to the incoming PPPoA link (on) or not (off). The default is <i>off</i> . Note 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

EXAMPLE:

```
=>ppoa iflist
myPPPoA: dest : ATM1      [00:00:00]
  Retry : 10  QoS default  encaps VC-MUX
  mode = IP routing
  flags = echo magic accomp restart mru addr savepwd
  dns metric = 0  mru = 1500
  auth = auto  user = johndoe 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 =
=>
```


pppoa ifdelete

Delete a PPPoA interface.

SYNTAX:

```
pppoa ifdelete intf = <intfname>
```

where:

intf	The name of the PPPoA interface to be deleted.	REQUIRED
------	--	----------

EXAMPLE:

```
=>pppoa iflist
pppoa_pppoa: dest : pppoa      [00:00:00]
  Retry : 10  QoS default  encaps VC-MUX
  mode = IP routing
  flags = echo magic accomp restart mru addr route savepwd
  dns metric = 0  trans addr = pat  mru = 1500
  route : dst=0.0.0.0/0 - src=10.0.0.0/1 (metric 1)
  auth = auto  user = johndoe@ISP  password = *****
  admin state = up    oper state = down    link state = retrying
  LCP : state = starting  retransm = 1  term. reason =
  IPCP: state = initial  retransm = 0  term. reason =
pppoa2: dest : pppoa2      [00:00:00]
  Retry : 10  QoS default  encaps VC-MUX
  mode = IP routing
  flags = echo magic accomp restart mru addr savepwd
  dns metric = 0  mru = 1500
  auth = auto  user = 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 =
=>pppoa ifdelete intf=pppoa2
=>pppoa iflist
pppoa_pppoa: dest : pppoa      [00:00:00]
  Retry : 10  QoS default  encaps VC-MUX
  mode = IP routing
  flags = echo magic accomp restart mru addr route savepwd
  dns metric = 0  trans addr = pat  mru = 1500
  route : dst=0.0.0.0/0 - src=10.0.0.0/1 (metric 1)
  auth = auto  user = johndoe@ISP  password = *****
  admin state = up    oper state = down    link state = retrying
  LCP : state = starting  retransm = 1  term. reason =
  IPCP: state = initial  retransm = 0  term. reason =
=>
```

RELATED COMMANDS:

- pppoa ifadd** Create a new PPPoA interface.
- pppoa iflist** Show current configuration of a specific or all PPPoA interface(s).

pppoa ifdetach

Detach a PPPoA interface.

SYNTAX:

```
pppoa ifdetach intf = <intfname>
```

where:

intf	The name of the PPPoA interface to be detached.	REQUIRED
------	---	----------

EXAMPLE:

```
=>pppoa iflist
PPPoA_1: dest : PVC_1      [00:00:00]
  Retry : 10 QoS default  encaps VC-MUX
  mode = IP routing
  flags = echo magic accomp restart mru addr route savepwd
  dns metric = 0  trans addr = pat  mru = 1500
  route : dst=0.0.0.0/0 - src=0.0.0.0/0 (metric 1)
  auth = auto  user = johndoe  password = *****
  admin state = up  oper state = up  link state = connected
  LCP : state = starting  retransm = 1  term. reason =
  IPCP: state = initial  retransm = 0  term. reason =

=>pppoa ifdetach intf=PPPoA_1
=>pppoa iflist
=>pppoa iflist
PPPoA_1: dest : PVC_1      [00:00:00]
  Retry : 10 QoS default  encaps VC-MUX
  mode = IP routing
  flags = echo magic accomp restart mru addr route savepwd
  dns metric = 0  trans addr = pat  mru = 1500
  route : dst=0.0.0.0/0 - src=0.0.0.0/0 (metric 1)
  auth = auto  user = johndoe  password = *****
  admin state = up  oper state = down  link state = not-connected
  LCP : state = starting  retransm = 1  term. reason =
  IPCP: state = initial  retransm = 0  term. reason =

=>
```

RELATED COMMANDS:

pppoa ifattach

Attach a PPPoA interface.

pppoa iflist

Show current configuration of a specific or all PPPoA interface(s).

SYNTAX:

```
pppoa iflist [intf = <string>]
```

where:

intf	The name of the PPPoA interface for which the configuration must be shown.	OPTIONAL
-------------	--	----------

Note If not specified, all PPPoA interfaces are shown.

EXAMPLE:

```
=>pppoa iflist
pppoa: dest : pppoa      [00:00:00]
      Retry : 10  QoS default  encaps VC-MUX
      mode = IP routing
      flags = echo magic accomp restart mru addr route savepwd
      dns metric = 0  trans addr = pat  mru = 1500
      route : dst=0.0.0.0/0 - src=10.0.0.0/1 (metric 1)
      auth = auto  user = johndoe@ISP  password = *****
      admin state = up    oper state = up    link state = connected
      LCP : state = starting  retransm = 1  term. reason =
      IPCP: state = initial  retransm = 0  term. reason =

pppoa2: dest : pppoa2    [00:00:00]
      Retry : 10  QoS default  encaps VC-MUX
      mode = IP routing
      flags = echo magic accomp restart mru addr savepwd
      dns metric = 0  mru = 1500
      auth = auto  user =      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:

pppoa ifadd	Create a new PPPoA interface.
pppoa ifdelete	Delete a PPPoA interface.
pppoa iflist	Show current configuration of a specific or all PPPoA interface(s).

pppoa rtadd

Automatically add a route configuration to the routing table.

The route will be added when the specified PPPoA 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.

SYNTAX:

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

where:

intf	The name of the PPPoA 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: <ul style="list-style-type: none"> ▶ Any remote destination is reachable, i.e. the PPPoA connection acts as default route (<i>dstmsk=0</i>) ▶ Only the remote (sub)net is reachable (<i>dstmsk=1</i>). The actual destination mask will be the default netmask applicable for destination IP address ▶ Only the single remote host is reachable (<i>dstmsk=32</i>). Any valid (contiguous) netmask in case of Variable Length Subnet Masking (VLSM). 	OPTIONAL
label	The name of the label.	OPTIONAL
src	The source IP address specification (in other words, who can use this link).	OPTIONAL
srcmsk	The source IP mask. Depending on the source netmask: <ul style="list-style-type: none"> ▶ Everybody is allowed to use this PPPoA connection (<i>srcmsk=0</i>) ▶ Only members of the same subnet as the host which opened the PPPoA connection are allowed to use the PPPoA connection (<i>srcmsk=1</i>). The actual destination mask will be the netmask applicable for the IP address of the host which opened the PPPoA connection. ▶ Only the host which opened the PPPoA connection is allowed to use the PPPoA connection (<i>srcmsk=32</i>). Any valid (contiguous) netmask in case of VLSM. 	OPTIONAL
metric	A number between 0 and 100. Represents the route metric (the cost factor) of the route. Practically, the cost is determined by the hop count.	OPTIONAL

EXAMPLE:

```
=>pppoe iflist
pppoe1: dest : pppoe      [00:00:00]
  Retry : 10  QoS default  encaps VC-MUX
  mode = IP routing
  flags = echo magic accomp restart mru addr route savepwd
  dns metric = 0  trans addr = pat  mru = 1500
  route : dst=0.0.0.0/0 - src=10.0.0.0/1 (metric 1)
  auth = auto  user = johndoe@ISP  password = *****
  admin state = up    oper state = down    link state = not-connected
  LCP : state = starting  retransm = 1  term. reason =
  IPCP: state = initial  retransm = 0  term. reason =

=>pppoe rtadd intf=pppoe1 dst=172.16.0.5 dstmsk=24 src=10.0.0.2 srcmask=24
=>pppoe iflist
pppoe1: dest : pppoe      [00:00:00]
  Retry : 10  QoS default  encaps VC-MUX
  mode = IP routing
  flags = echo magic accomp restart mru addr route savepwd
  dns metric = 0  trans addr = pat  mru = 1500
  route = 10.0.0.2/24 - 172.16.0.5/24 (metric 1)
  auth = auto  user = johndoe@ISP  password = *****
  admin state = up    oper state = down    link state = not-connected
  LCP : state = starting  retransm = 1  term. reason =
  IPCP: state = initial  retransm = 0  term. reason =

=>
```

RELATED COMMANDS:

pppoe rtdelete Delete the route specification for a PPPoA link.

pppoa rtdelete

Delete the route specification for a PPPoA link.

SYNTAX:

```
pppoa rtdelete intf = <intfname>
```

where:

intf	The PPPoA interface name for which to delete the route settings.	REQUIRED
------	--	----------

EXAMPLE:

```
=>pppoa iflist
pppoa: dest : pppoa      [00:00:00]
  Retry : 10  QoS default  encaps VC-MUX
  mode = IP routing
  flags = echo magic accomp restart mru addr route savepwd
  dns metric = 0  trans addr = pat  mru = 1500
  route = 10.0.0.2/24 - 172.16.0.5/24 (metric 1)
  auth = auto  user = johndoe@ISP  password = *****
  admin state = up  oper state = down  link state = not-connected
  LCP : state = starting  retransm = 1  term. reason =
  IPCP: state = initial  retransm = 0  term. reason =

=>pppoa rtdelete intf=pppoa1
=>pppoa iflist
pppoa_pppoa: dest : pppoa      [00:00:00]
  Retry : 10  QoS default  encaps VC-MUX
  mode = IP routing
  flags = echo magic accomp restart mru addr route savepwd
  dns metric = 0  trans addr = pat  mru = 1500
  auth = auto  user = johndoe@ISP  password = *****
  admin state = up  oper state = down  link state = not-connected
  LCP : state = starting  retransm = 1  term. reason =
  IPCP: state = initial  retransm = 0  term. reason =

=>
```

RELATED COMMANDS:

[pppoa rtadd](#)

Automatically add a route configuration to the routing table.

PPPoE Commands

Introduction

This chapter describes the commands of the **pppoe** command group.

Contents

This chapter covers the following commands:

pppoe flush	Flush the current PPPoE configuration.	286
pppoe ifadd	Create a new PPPoE interface.	287
pppoe ifattach	Attach a PPPoE interface.	288
pppoe ifconfig	Configure a PPPoE interface.	289
pppoe ifdelete	Delete a PPPoE interface.	293
pppoe ifdetach	Detach a PPPoE interface.	294
pppoe iflist	Show current configuration of all or a specified PPPoE interface(s).	295
pppoe ifscan	Scan a PPPoE interface for available Access Concentrator names and Service Names.	296
pppoe rtadd	Automatically add a route configuration to the routing table.	297
pppoe rtdelete	Delete the route specification for a PPPoE link.	299
pppoe relay add	Add an Ethernet port to the PPPoE relay list.	300
pppoe relay delete	Delete an Ethernet port from the PPPoE relay agent list.	301
pppoe relay flush	Remove all Ethernet ports from the PPPoE relay agent list and terminate all sessions.	302
pppoe relay portlist	List all Ethernet ports added to the PPPoE relay agent list.	303
pppoe relay sesslist	List all active PPPoE relay sessions.	304

pppoe flush

Flush the current PPPoE configuration.



The flush command does not impact previously saved configurations.

SYNTAX:

```
pppoe flush
```

EXAMPLE:

```
=>pppoe iflist
PPPoE1: dest : PPPoE1
  Retry: 10 QoS default encaps VC-MUX
  mode = IP Routing
  flags = echo magic accomp mru addr route savepwd PPPoEOA
  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 =
=>pppoe flush
=>pppoe iflist
=>
```


pppoe ifadd

Create a new PPPoE interface.

SYNTAX:

```
pppoe ifadd [intf = <string>]
           [dest = <RELAY | phonebook entry>]
```

where:

intf	The name for the new PPP over Ethernet (PPPoE) interface.	OPTIONAL
Note	If not specified, the destination parameter must be specified. In this case the name of the destination will double as interface name.	
dest	The Ethernet port to be used for this PPPoE interface (e.g. an ETHoA name).	OPTIONAL

EXAMPLE:

```
=>pppoe iflist
myPPPoE: dest : myETHoA [local disconnect] [00:00:00]
  Retry : 10
  mode = IP routing
  flags = echo magic accomp restart mru addr savepwd
  dns metric = 0 mru = 1492
  auth = auto user = johndoe password = *****
  admin state = up oper state = down link state = connected
  LCP : state = reqsent retransm = 1 term. reason =
  IPCP: state = initial retransm = 0 term. reason =
  acname : --- service : ---

=>pppoe ifadd intf=yourPPPoE
=>pppoe iflist
myPPPoE: dest : myETHoA [local disconnect] [00:00:00]
  Retry : 10
  mode = IP routing
  flags = echo magic accomp restart mru addr savepwd
  dns metric = 0 mru = 1492
  auth = auto user = johndoe password = *****
  admin state = up oper state = down link state = connected
  LCP : state = reqsent retransm = 1 term. reason =
  IPCP: state = initial retransm = 0 term. reason =
  acname : --- service : ---

yourPPPoE: dest : [00:00:00]
  Retry : 10
  mode = IP routing
  flags = echo magic accomp restart mru addr savepwd
  dns metric = 0 mru = 1492
  auth = auto user = password =
  admin state = down oper state = down link state = not-connected
  LCP : state = initial retransm = 9 term. reason =
  IPCP: state = initial retransm = 0 term. reason =
  acname : --- service : ---

=>
```

RELATED COMMANDS:

- pppoe ifdelete** Delete a PPPoE interface.
- pppoe iflist** Show current configuration of all or a specified PPPoE interface(s).

pppoe ifattach

Attach a PPPoE interface.

SYNTAX:

```
pppoe ifattach intf = <string>
```

where:

intf	The name of the PPPoE interface to be attached.	REQUIRED
------	---	----------

EXAMPLE:

```
=>pppoe iflist
myPPPoE: dest : myETHoA [local disconnect] [00:00:00]
  Retry : 10
  mode = IP routing
  flags = echo magic accomp restart mru addr savepwd
  dns metric = 0 mru = 1492
  auth = auto user = johndoe password = *****
  admin state = down oper state = down link state = not-connected
  LCP : state = initial retransm = 9 term. reason = User kill
  IPCP: state = initial retransm = 0 term. reason =
  acname : --- service : ---

=>pppoe ifattach intf=myPPPoE
=>pppoe iflist
myPPPoE: dest : myETHoA [local disconnect] [00:00:00]
  Retry : 10
  mode = IP routing
  flags = echo magic accomp restart mru addr savepwd
  dns metric = 0 mru = 1492
  auth = auto user = johndoe password = *****
  admin state = up oper state = down link state = connected
  LCP : state = reqsent retransm = 1 term. reason =
  IPCP: state = initial retransm = 0 term. reason =
  acname : --- service : ---

=>
```

RELATED COMMANDS:

pppoe ifdetach

Detach a PPPoE interface.

pppoe ifconfig

Configure a PPPoE interface.

SYNTAX:

```
pppoe ifconfig intf = <string>
                [dest = <string>]
                [user = <string>]
                [password = <password>]
                [acname = <quoted string>]
                [servicename = <quoted string>]
                [pcomp = <{off | on}>]
                [accomp = <{on | off | negotiate}>]
                [trace = <{off | on}>]
                [concentrator = <{off | on}>]
                [auth = <{pap | chap | auto}>]
                [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 = <{cidr | dotted | none}>]
                [pool = <{none}>]
                [savepwd = <{off | on}>]
                [demanddial = <{off | on}>]
                [primdns = <ip-address>]
                [secdns = <ip-address>]
                [dnsmetric = <number{0-100}>]
                [idle = <number{0-1000000}>]
                [idletrigger = <{RxTx | Rx | Tx}>]
                [addrtrans = <{none | pat}>]
                [unnumbered = <{off | on}>]
```

where:

intf	The name of the PPPoE interface to be configured.	REQUIRED
dest	The destination for this PPPoE interface. Typically, a phonebook entry.	OPTIONAL
user	The user name for remote PAP/CHAP authentication.	OPTIONAL
password	The password for remote PAP/CHAP authentication.	OPTIONAL
acname	The Access Concentrator name for a PPPoE connection.	OPTIONAL
	Tip Use the command :pppoe ifscan to see the names of available access concentrators, if any.	

servicename	The Service Name for a PPPoE connection. Tip Use the command <code>:pppoe ifscan</code> to see the available service names, if any.	OPTIONAL
pcomp	Try (on) or do not try (off) to negotiate PPPoE protocol compression (LCP PCOMP). The default is off .	OPTIONAL
accomp	Try (on), do never try (off) or negotiate (negotiate) to negotiate PPPoE address & control field compression (LCP ACCOMP). In most cases, LCP ACCOMP should not be disabled nor negotiated, that is the address field FF-03 should not be sent over ATM. The default is on . If the accomp parameter is set to 'negotiate', the local side of the PPPoE connection demands to do ACCOMP and adapts itself to the result of this negotiation.	OPTIONAL
trace	Enable (on) or disable (off) verbose console logging. The default is off .	OPTIONAL
concentrator	The access concentrator is on this side of the PPPoE connection. Choose between: <ul style="list-style-type: none"> ▶ on: the PPPoE connection is terminated on the Access Concentrator (here the SpeedTouch™ itself) ▶ off: the SpeedTouch™ is PPPoE client. The default is off .	OPTIONAL
auth	Select the authentication protocol. Choose between: <ul style="list-style-type: none"> ▶ pap: PAP authentication will be forced. ▶ chap: CHAP authentication will be forced. ▶ auto: CHAP authentication will be used. If CHAP authentication is not successful, PAP authentication will be used instead. The default is auto .	OPTIONAL
restart	Automatically restart the connection when LCP link goes down (on) or do not restart automatically (off). The default is on .	OPTIONAL
retryinterval	A number between 0 and 65535 (seconds). Represents the intermediate interval between two retries to establish the connection on ATM level. The default is 10 .	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. The default is off .	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). The default is off .	OPTIONAL
echo	Send LCP echo requests at regular intervals (on) or not (off). The default is on .	OPTIONAL
mru	A number between 293 and 8192. Represents the maximum packet size the SpeedTouch™ should negotiate to be able to receive. The default is 1492 .	OPTIONAL

laddr	The local IP address of the peer-to-peer connection. Specifying a local IP address forces the remote side of the PPPoE link (if it allows to) to accept this IP address as the SpeedTouch™ PPPoE 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 PPPoE link (if it allows to) to accept this IP address as its PPPoE 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 PPPoE 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 CIDR format (cidr). The default is <i>cidr</i> .	OPTIONAL
pool	The name of the free DHCP server pool to which the acquired IP subnet must be assigned.	OPTIONAL
savepwd	Save password (on), if supplied, or do not save the password (off). The default is <i>on</i> .	OPTIONAL
demanddial	Enable (on) or disable (off) the dial-on-demand feature. The default is <i>on</i> .	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. Note 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. Note If not specified, the SpeedTouch™ will accept any IP address.	OPTIONAL
dnsmetric	A number between 1 and 100. Represents the DNS route metric to be used for the negotiated DNS servers. The default is <i>0</i> .	OPTIONAL
idle	A number between 0 and 1000000 (seconds). Represents after how many seconds an idle link goes down. The default is <i>0</i> .	OPTIONAL
idletrigger	Consider the link being idle if no traffic is sent and/or received during the idle time. Choose between: <ul style="list-style-type: none"> ▶ RxTx: The idle time period restarts when a packet is transmitted or received. ▶ Rx: The idle time period restarts when a packet is received. Transmitted packets are ignored. ▶ Tx: The idle time period restarts when a packet is transmitted. Received packets are ignored. The default is <i>RxTx</i> .	OPTIONAL
addrtrans	Automatically enable address translation for the IP address of this link (pat) or do not use address translation (none). The default is <i>none</i> .	OPTIONAL

unnumbered Takes the local IP address from the *laddr* field and remote IP address from the IP address pool assigned to the incoming PPPoE link (on) or not (off). OPTIONAL
The default is **off**.

Note 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.

EXAMPLE:

```
=>pppoe iflist
myPPPoE: dest : myETHoA      [00:00:00]
  Retry : 10
  mode = IP routing
  flags = echo magic accomp restart mru addr savepwd
  dns metric = 0 mru = 1492
  auth = auto user = johndoe 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 =
  acname : --- service : ---
=>
```

pppoe ifdelete

Delete a PPPoE interface.

SYNTAX:

```
pppoe ifdelete intf = <string>
```

where:

```
intf The name of the PPPoE interface to be deleted.
```

EXAMPLE:

```
=>pppoe iflist
myPPPoE: dest : myETHoA [00:00:00]
  Retry : 10
  mode = IP routing
  flags = echo magic accomp restart mru addr savepwd
  dns metric = 0 mru = 1492
  auth = auto user = johndoe password = *****
  admin state = up oper state = down link state = connected
  LCP : state = reqsent retransm = 6 term. reason =
  IPCP: state = initial retransm = 0 term. reason =
  acname : --- service : ---

yourPPPoE: dest : RELAY [00:00:00]
  Retry : 10
  mode = IP routing
  flags = echo magic accomp restart mru addr savepwd
  dns metric = 0 mru = 1492
  auth = auto user = janedoe password = *****
  admin state = up oper state = down link state = connected
  LCP : state = reqsent retransm = 1 term. reason =
  IPCP: state = initial retransm = 0 term. reason =
  acname : --- service : ---
=>pppoe ifdelete intf=yourPPPoE
=>pppoe iflist
myPPPoE: dest : myETHoA [00:00:00]
  Retry : 10
  mode = IP routing
  flags = echo magic accomp restart mru addr savepwd
  dns metric = 0 mru = 1492
  auth = auto user = johndoe password = *****
  admin state = up oper state = down link state = connected
  LCP : state = reqsent retransm = 2 term. reason =
  IPCP: state = initial retransm = 0 term. reason =
  acname : --- service : ---

=>
```

RELATED COMMANDS:

- pppoe ifadd** Create a new PPPoE interface.
- pppoe iflist** Show current configuration of all or a specified PPPoE interface(s).

pppoe ifdetach

Detach a PPPoE interface.

SYNTAX:

```
pppoe ifdetach intf = <string>
```

where:

intf	The name of the PPPoE interface.	REQUIRED
------	----------------------------------	----------

EXAMPLE:

```
=>pppoe iflist
myPPPoE: dest : myETHoA [local disconnect] [00:00:00]
  Retry : 10
  mode = IP routing
  flags = echo magic accomp restart mru addr savepwd
  dns metric = 0 mru = 1492
  auth = auto user = johndoe password = *****
  admin state = up oper state = down link state = connected
  LCP : state = stopped retransm = 10 term. reason =
  IPCP: state = initial retransm = 0 term. reason =
  acname : --- service : ---

=>pppoe ifdetach intf=myPPPoE
=>pppoe iflist
myPPPoE: dest : myETHoA [local disconnect] [00:00:00]
  Retry : 10
  mode = IP routing
  flags = echo magic accomp restart mru addr savepwd
  dns metric = 0 mru = 1492
  auth = auto user = johndoe password = *****
  admin state = down oper state = down link state = not-connected
  LCP : state = initial retransm = 9 term. reason = User kill
  IPCP: state = initial retransm = 0 term. reason =
  acname : --- service : ---

=>
```

RELATED COMMANDS:

pppoe ifattach

Attach a PPPoE interface.

pppoe iflist

Show current configuration of all or a specified PPPoE interface(s).

SYNTAX:

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

where:

intf	The name of the PPPoE interface.	OPTIONAL
------	----------------------------------	----------

Note If not specified, all PPPoE interfaces are shown.

EXAMPLE :

```
=>pppoe iflist
myPPPoE: dest : myETHoA      [00:00:00]
  Retry : 10
  mode = IP routing
  flags = echo magic accomp restart mru addr savepwd
  dns metric = 0 mru = 1492
  auth = auto user = johndoe password = *****
  admin state = up oper state = down link state = connected
  LCP : state = reqsent retransm = 6 term. reason =
  IPCP: state = initial retransm = 0 term. reason =
  acname : --- service : ---

yourPPPoE: dest : RELAY      [00:00:00]
  Retry : 10
  mode = IP routing
  flags = echo magic accomp restart mru addr savepwd
  dns metric = 0 mru = 1492
  auth = auto user = janedoe password = *****
  admin state = up oper state = down link state = connected
  LCP : state = reqsent retransm = 1 term. reason =
  IPCP: state = initial retransm = 0 term. reason =
  acname : --- service : ---

=>
```

RELATED COMMANDS:

pppoe ifadd	Create a new PPPoE interface.
pppoe ifdelete	Delete a PPPoE interface.

pppoe ifscan

Scan a PPPoE interface for available Access Concentrator names and Service Names.



Use the command **:pppoe ifdetach** for this interface before performing a scan on it.

SYNTAX:

```
pppoe ifscan intf = <string>
                [time = <number{0-36000}>]
```

where:

intf	The name of the Routed PPPoE interface to be scanned.	REQUIRED
time	A number between 0 and 36000 (seconds). Represents the time to scan for services.	OPTIONAL

EXAMPLE:

```
=>pppoe iflist
myPPPoE: dest : myETHoA      [00:00:00]
  Retry : 10
  mode = IP routing
  flags = echo magic accomp restart mru addr savepwd
  dns metric = 0  mru = 1492
  auth = auto  user = johndoe  password = *****
  admin state = up  oper state = down  link state = connected
  LCP : state = reqsent  retransm = 6  term. reason =
  IPCP: state = initial  retransm = 0  term. reason =
  acname : ---  service : ---
=>pppoe ifscan intf=myPPPoE time=45
      Service Name                Access Concentrator

Done !
=>
```

pppoe rtadd

Automatically add a route configuration to the routing table.

The route is added when the specified PPPoE 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.



Use the command **:pppoe ifdetach** for this interface prior to configuring routes.

SYNTAX:

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

where:

intf	The name of the PPPoE 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: <ul style="list-style-type: none"> ▶ Any remote destination is reachable, i.e. the PPPoE connection acts as default route (<i>dstmsk=0</i>) ▶ Only the remote (sub)net is reachable (<i>dstmsk=1</i>). The actual destination mask will be the default netmask applicable for destination IP address ▶ Only the single remote host is reachable (<i>dstmsk=32</i>). Any valid (contiguous) netmask in case of VLSM. 	OPTIONAL
label	The name of the label.	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: <ul style="list-style-type: none"> ▶ Everybody is allowed to use this PPPoE connection (<i>dstmsk=0</i>) ▶ Only members of the same subnet as the host which opened the PPPoE connection are allowed to use the PPPoE connection (<i>dstmsk=1</i>). The actual destination mask will be the netmask applicable for the IP address of the host which opened the PPPoE connection. ▶ Only the host which opened the PPPoE connection is allowed to use the PPPoE connection (<i>dstmsk=32</i>). 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.	OPTIONAL

EXAMPLE:

```
=>pppoe iflist
myPPPoE: dest : myETHoA [local disconnect] [00:00:00]
  Retry : 10
  mode = IP routing
  flags = echo magic accomp restart mru addr savepwd
  dns metric = 0 mru = 1492
  auth = auto user = johndoe password = *****
  admin state = down oper state = down link state = not-connected
  LCP : state = initial retransm = 9 term. reason = User kill
  IPCP: state = initial retransm = 0 term. reason =
  acname : --- service : ---

=>pppoe rtadd intf=myPPPoE dst=0.0.0.0/0 src=0.0.0.0/0
=>pppoe iflist
myPPPoE: dest : myETHoA [local disconnect] [00:00:00]
  Retry : 10
  mode = IP routing
  flags = echo magic accomp restart mru addr route savepwd
  dns metric = 0 mru = 1492
  route : dst=0.0.0.0/0 - src=0.0.0.0/0 (metric 1)
  auth = auto user = johndoe password = *****
  admin state = down oper state = down link state = not-connected
  LCP : state = initial retransm = 9 term. reason = User kill
  IPCP: state = initial retransm = 0 term. reason =
  acname : --- service : ---

=>
```

RELATED COMMANDS:

pppoe rtdelete

Delete the route specification for a PPPoE link.

pppoe rtdelete

Delete the route specification for a PPPoE link.



Use the command **:pppoe ifdetach** for this interface prior to deleting route configurations.

SYNTAX:

```
pppoe rtdelete intf = <string>
```

where:

intf	The PPPoE interface name for which to delete the route settings.	REQUIRED
------	--	----------

EXAMPLE:

```
=>pppoe iflist
myPPPoE: dest : myETHoA [local disconnect] [00:00:00]
  Retry : 10
  mode = IP routing
  flags = echo magic accomp restart mru addr savepwd
  dns metric = 0 mru = 1492
  route = 10.0.0.2/24 - 172.16.0.5/24 (metric 1)
  auth = auto user = johndoe password = *****
  admin state = down oper state = down link state = not-connected
  LCP : state = initial retransm = 9 term. reason = User kill
  IPCP: state = initial retransm = 0 term. reason =
  acname : --- service : ---

=>pppoe rtdelete intf=myPPPoE
=>pppoe iflist
myPPPoE: dest : myETHoA [local disconnect] [00:00:00]
  Retry : 10
  mode = IP routing
  flags = echo magic accomp restart mru addr savepwd
  dns metric = 0 mru = 1492
  auth = auto user = johndoe password = *****
  admin state = down oper state = down link state = not-connected
  LCP : state = initial retransm = 9 term. reason = User kill
  IPCP: state = initial retransm = 0 term. reason =
  acname : --- service : ---

=>
```

RELATED COMMANDS:

pppoe rtadd

Automatically add a route configuration to the routing table.

pppoe relay add

Add an Ethernet port to the PPPoE relay list.

SYNTAX:

```
pppoe relay add port = <string>
```

where:

port	The Ethernet port to be added to the PPPoE relay agent list.	REQUIRED
------	--	----------

EXAMPLE:

```
=>pppoe relay portlist
=>pppoe relay add port=eth0
=>pppoe relay portlist
# Port HWaddr Status
1 eth0 00-0e-50-05-00-3a UP
=>
```

RELATED COMMANDS:

[pppoe relay delete](#)

Delete an Ethernet port from the PPPoE relay agent list.

[pppoe relay portlist](#)

List all Ethernet ports added to the PPPoE relay agent list.

pppoe relay delete

Delete an Ethernet port from the PPPoE relay agent list.

SYNTAX:

```
pppoe relay delete port = <string>
```

where:

port	The Ethernet port to be deleted from the PPPoE relay agent list.	REQUIRED
------	--	----------

EXAMPLE:

```
=>pppoe relay portlist
# Port HWaddr Status
1 eth0 00-0e-50-05-00-3a UP
=>pppoe relay delete port=eth0
=>pppoe relay portlist
=>
```

RELATED COMMANDS:

- pppoe relay add** Add an Ethernet port to the PPPoE relay list.
- pppoe relay portlist** List all Ethernet ports added to the PPPoE relay agent list.

pppoe relay flush

Remove all Ethernet ports from the PPPoE relay agent list and terminate all sessions.

SYNTAX:

```
pppoe relay flush
```


pppoe relay portlist

List all Ethernet ports added to the PPPoE relay agent list.

SYNTAX:

```
pppoe relay portlist
```

EXAMPLE:

```
=>pppoe relay portlist
# Port HWaddr Status
1 eth0 00-0e-50-05-00-3a UP
=>
```

RELATED COMMANDS:

- pppoe relay add** Add an Ethernet port to the PPPoE relay list.
- pppoe relay delete** Delete an Ethernet port from the PPPoE relay agent list.

pppoe relay sesslist

List all active PPPoE relay sessions.

SYNTAX:

```
pppoe relay sesslist
```

PPTP Commands

Introduction

This chapter describes the commands of the **pptp** command group.

Contents

This chapter covers the following commands:

pptp flush	Flush complete PPTP configuration.	306
pptp ifadd	Add a Point-to-Point Tunneling Protocol (PPTP) profile.	307
pptp list	Show current PPTP configuration.	308
pptp profadd	Define a new PPTP profile.	309
pptp profdelete	Delete a PPTP profile.	310
pptp proflist	Show all current PPTP profiles.	311

pptp flush

Flush complete PPTP configuration.



The flush command does not impact previously saved configurations.

SYNTAX:

```
pptp flush
```

EXAMPLE:

```
=>pptp proflist
Profile      QoS      Encaps      AC
Relay_PPP1  default  nlpid      always
=>pptp flush
=>pptp proflist
=>
```

pptp ifadd

Add a Point-to-Point Tunneling Protocol (PPTP) profile.



For backwards compatibility with previous release, use profiles instead.

SYNTAX:

```
pptp ifadd dest = <string>
           [rate = <number{10-10000}>]
           [encaps = <{vcmux | nlpid}>]
           [ac = <{never | always | keep}>]
```

where:

dest	The WAN destination for this PPTP tunnel. Typically a phonebook entry.	REQUIRED
rate	A number between 10 and 10000. Represents the transmission speed in bits/s for the WAN link.	OPTIONAL
encaps	The type of WAN encapsulation. Choose between: <ul style="list-style-type: none"> ▶ vcmux ▶ nlpid (Network Layer Protocol IDentifiers (NLPID)). The default is vcmux .	OPTIONAL
ac	The High-level Data Link Control (HDLC) framing option applicable to PPTP interfaces using this PPTP profile. Choose between: <ul style="list-style-type: none"> ▶ always: 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. ▶ never: Before relaying the encapsulated PPP frames over the PPPoA link, make sure the address and control field will never be found in front of the frames. ▶ keep: Do not change the frames arriving via the PPTP tunnel. The default is never (compliant to RFC2364). Note It is recommended to keep this setting.	OPTIONAL

EXAMPLE:

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

pptp list

Show current PPTP configuration.

SYNTAX:

```
pptp list
```

EXAMPLE:

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

pptp profadd

Define a new PPTP profile.

SYNTAX:

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

where:

name	The name for the PPTP profile.	REQUIRED
qos	The name of the qosbook entry, containing the settings for this profile. This parameter never needs to be specified.	OPTIONAL
encaps	The type of WAN encapsulation. Choose between: <ul style="list-style-type: none"> ▶ vc mux ▶ nlpid. The default is vc mux .	OPTIONAL
ac	The HDLC framing option applicable to PPTP interfaces using this PPTP profile. Choose between: <ul style="list-style-type: none"> ▶ always: 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. ▶ never: Before relaying the encapsulated PPP frames over the PPPoA link, make sure the address and control field will never be found in front of the frames. ▶ keep: Do not change the frames arriving via the PPTP tunnel. The default is never (compliant to RFC2364). Note 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  vc mux      never
=>
```

RELATED COMMANDS:

- pptp profdelete** Delete a PPTP profile.
- pptp proflist** Show all current PPTP profiles.

pptp profdelete

Delete a PPTP profile.

SYNTAX:

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

where:

name	The name for the PPTP 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:

<code>pptp profadd</code>	Define a new PPTP profile.
<code>pptp proflist</code>	Show all current PPTP profiles.

pptp proflist

Show all current PPTP 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 PPTP profile.
- pptp profdelete** Delete a PPTP profile.

QoSBook Commands

Introduction

This chapter describes the commands of the **qosbook** command group.

Contents

This chapter covers the following commands:

<code>qosbook add</code>	Add a QoS book entry.	314
<code>qosbook config</code>	Set or change QoS book config parameters.	316
<code>qosbook delete</code>	Remove a QoS book entry.	317
<code>qosbook flush</code>	Flush the complete QoS book.	318
<code>qosbook list</code>	Show current Quality of Service book.	319

qosbook add

Add a QoS book entry.

SYNTAX:

```
qosbook add name = <string>
            class = <{ubr | cbr | vbr-rt | vbr-nrt}>
            [tx_peakrate = <number{0-27786}>]
            [tx_sustrate = <number{0-27786}>]
            [tx_maxburst = <number{0-12240}>]
            [rx_peakrate = <number{0-27786}>]
            [rx_sustrate = <number{0-27786}>]
            [rx_maxburst = <number{0-12240}>]
            [framediscard = <{enabled | disabled}>]
```

where:

name	The name for the new QoS entry.	REQUIRED
class	The ATM service category. Choose between: <ul style="list-style-type: none"> ▶ ubr: unspecified bit rate ▶ cbr: constant bit rate ▶ vbr-rt: variable bit rate - real time ▶ vbr-nrt: variable bit rate - non real time. 	REQUIRED
tx_peakrate	A number between 0 and 27786 (Kilobits per second). Indicates the peak rate (in kilobits per second) in the transmit (upstream) direction. Use 0 to indicate linerate for UBR.	OPTIONAL
tx_sustrate	A number between 0 and 27786 (Kilobits per second). Indicates the sustainable rate (in kilobits per second) in the transmit (upstream) direction (VBR only).	OPTIONAL
tx_maxburst	A number between 0 and 12240 (bytes per second). Indicates the maximum burst size (in bytes) in the transmit (upstream) direction (VBR only).	OPTIONAL
rx_peakrate	A number between 0 and 27786 (Kilobits per second). Indicates the peak rate (in kilobits per second) in the receive (downstream) direction. Use 0 to indicate linerate for UBR. If not specified, copy of the transmit peak rate (ATMF only).	OPTIONAL
rx_sustrate	A number between 0 and 27786 (Kilobits per second). Indicates the sustainable rate (in kilobits per second) in the receive (downstream) direction. If not specified, copy of the transmit sustainable rate (VBR ATMF only).	OPTIONAL
rx_maxburst	A number between 0 and 12240 (bytes per second). Indicates the maximum burst size (in bytes) in the receive (downstream) direction. If not specified, copy of the transmit maximum burst size (VBR ATMF only).	OPTIONAL
framediscard	Enable/disable frame discard. The default is disabled .	OPTIONAL

EXAMPLE:

```

=>qosbook list
Name      Ref Type      TX peak  sust      burst     RX peak  sust      burst     framediscard
      (Kbits) (Kbits) (bytes) (Kbits) (Kbits) (bytes)
default 3   ubr      linerate 0          0          linerate 0          0          disabled
=>qosbook add name=TestVBR class=vbr-nrt tx_peakrate=1500 tx_sustrate=1000 tx_maxburst=100
rx_sustrate=1000 rx_maxburst=100 framediscard=enabled
=>qosbook list
Name      Ref Type      TX peak  sust      burst     RX peak  sust      burst     framediscard
      (Kbits) (Kbits) (bytes) (Kbits) (Kbits) (bytes)
default 3   ubr      linerate 0          0          linerate 0          0          disabled
TestVBR 0   vbr-nrt 1500     1000     144       1500     1000     144       enabled
=>

```

IMPORTANT


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 the command **:qosbook list** (burst=144).

RELATED COMMANDS:

- qosbook delete** Remove a QoS book entry.
- qosbook list** Show current Quality of Service book.

qosbook config

Set or change QoS book config parameters.

SYNTAX:

```
qosbook config [format = <{bytes | cells}>]
```

where:

format	The input/output format of the QoS book. Choose between:	OPTIONAL
	<ul style="list-style-type: none"> ▶ bytes ▶ cells. 	
	The default is bytes .	

EXAMPLE:

```
=>qosbook list
Name      Ref Type      TX peak  sust      burst      RX peak  sust      burst      framediscard
          (Kbits) (Kbits) (bytes) (Kbits) (Kbits) (bytes)
default 3  ubr      linerate 0      0      linerate 0      0      disabled
=>qosbook config
qosbook format in bytes
=>qosbook config format=cells
=>qosbook config
qosbook format in cells
=>qosbook list
Name      Ref Type      TX peak  sust      burst      RX peak  sust      burst      framediscard
          (cps) (cps) (cells) (cps) (cps) (cells)
default 3  ubr      linerate 0      0      linerate 0      0      disabled
=>
```

qosbook delete

Remove a QoS book entry.

SYNTAX:

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

where:

name	The name of the QoS book entry to be deleted.	REQUIRED
force	Force deletion of the entry even if it is still in use (yes) or do not force the deletion (no). The default is no .	OPTIONAL

EXAMPLE:

```
=>qosbook list
Name      Ref Type      TX peak  sust      burst     RX peak  sust      burst     framediscard
      (Kbits) (Kbits) (bytes) (Kbits) (Kbits) (bytes)
default 3   ubr      linerate 0          0          linerate 0          0          disabled
TestVBR 0   vbr-nrt 1500     1000     144       1500     1000     144       enabled
=>qosbook delete name=TestVBR
=>qosbook list
Name      Ref Type      TX peak  sust      burst     RX peak  sust      burst     framediscard
      (Kbits) (Kbits) (bytes) (Kbits) (Kbits) (bytes)
default 3   ubr      linerate 0          0          linerate 0          0          disabled
=>
```

RELATED COMMANDS:

- qosbook add** Add a QoS book entry.
- qosbook list** Show current Quality of Service book.

qosbook flush

Flush the complete QoS book.



The flush command does not impact previously saved configurations.

SYNTAX:

```
qosbook flush
```


qosbook list

Show current Quality of Service book.

SYNTAX:

```
qosbook list
```

EXAMPLE:

```
=>qosbook list
Name      Ref Type      TX peak  sust      burst      RX peak  sust      burst      framediscard
          (Kbits) (Kbits)  (bytes)  (Kbits)  (Kbits)  (bytes)
default 3   ubr      linerate 0          0          linerate 0          0          disabled
TestVBR 0   vbr-nrt 1500     1000     144       1500     1000     144       enabled
=>
```

RELATED COMMANDS:

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

Script Commands

Introduction

This chapter describes the commands of the **script** command group.



Scripting is not a general purpose mechanism but is only used in the autoPVC/ILMI mechanism.



It is not recommended to change the default scripts.

Contents

This chapter covers the following commands:

<code>script add</code>	Add a line to a script.	322
<code>script delete</code>	Delete a complete script or a line from a script.	323
<code>script flush</code>	Flush all the scripts.	324
<code>script list</code>	List the script(s).	325
<code>script run</code>	Run a script.	326

script add

Add a line to a script.

SYNTAX:

```
script add name = <string>
           [index = <number>]
           command = <quoted string>
```

where:

name	Name of script.	REQUIRED
index	Line number. Note Use 0 to add.	OPTIONAL
command	Command.	REQUIRED

RELATED COMMANDS:

- [script delete](#) Delete a complete script or a line from a script.
- [script list](#) List the script(s).

script delete

Delete a complete script or a line from a script.

SYNTAX:

```
script delete name = <string>
               [index = <number>]
```

where:

name	Name of the script to be deleted.	REQUIRED
	Tip Use the command :script list to obtain the name of the script.	
index	Line number to be deleted.	OPTIONAL
	Note If not is specified, the complete script will be deleted.	

RELATED COMMANDS:

- script add Add a line to a script.
- script list List the script(s).

script flush

Flush all the scripts.



The flush command does not impact previously saved configurations.

SYNTAX:

```
script flush
```

script list

List the script(s).

SYNTAX:

```
script list [name = <string>]
```

where:

name	Name of the script to be shown.	OPTIONAL
	Note If not specified, all the scripts are listed.	

EXAMPLE

Some of the default scripts are shown below:

```
=>script list
Script: autopvc_add_qos
  0: qosbook_add name _auto_$1_$2 class $3 tx_peakrate $4 tx_sustrate $5 tx_maxburst $6
    rx_peakrate $4 rx_sustrate $5 rx_maxburst $6 dynamic yes
...
Script: autopvc_add_bridge
  0: qosbook_add name _auto_$1_$2 class $3 tx_peakrate $4 tx_sustrate $5 tx_maxburst $6
    rx_peakrate $4 rx_sustrate $5 rx_maxburst $6 dynamic yes
  1: phonebook add name _auto_$1_$2 addr $1.$2 type any dynamic yes
  2: bridge ifadd intf _auto_$1_$2 dest _auto_$1_$2
  3: bridge ifconfig intf _auto_$1_$2 qos _auto_$1_$2
  4: bridge ifattach intf _auto_$1_$2
...
Script: autopvc_delete_bridge
  0: bridge ifdetach intf _auto_$1_$2
  1: bridge ifdelete intf _auto_$1_$2
  2: phonebook delete name _auto_$1_$2
  3: qosbook delete name _auto_$1_$2
...
Script: autopvc_add_pppoerelay
  0: qosbook_add name _auto_$1_$2 class $3 tx_peakrate $4 tx_sustrate $5 tx_maxburst $6
    rx_peakrate $4 rx_sustrate $5 rx_maxburst $6 dynamic yes
  1: phonebook add name _auto_$1_$2 addr $1.$2 type any dynamic yes
  2: ethoa ifadd intf _auto_$1_$2 dest _auto_$1_$2
  3: ethoa ifconfig intf _auto_$1_$2 qos _auto_$1_$2
  4: ethoa ifattach intf _auto_$1_$2
  5: ip ifwait intf _auto_$1_$2 timeout 15 adminstatus up
  6: pppoe relay add port _auto_$1_$2
...
=>
```

RELATED COMMANDS:

- script add** Add a line to a script.
- script delete** Delete a complete script or a line from a script.

script run

Run a script.

SYNTAX:

```
script run name = <string>
          [par1 = <string>]
          [par2 = <string>]
          [par3 = <string>]
          [par4 = <string>]
          [par5 = <string>]
          [par6 = <string>]
          [par7 = <string>]
          [par8 = <string>]
          [par9 = <string>]
```

where:

name	Name of the script to be run.	REQUIRED
	Tip Use the command :script list to obtain the name of the script.	
par1 ... par9	Parameters to be used in the script.	OPTIONAL

SNMP Commands

Introduction

This chapter describes the commands of the **snmp** command group.

Contents

This chapter covers the following commands:

<code>snmp config</code>	Show/set global Simple Network Management Protocol (SNMP) parameters.	328
<code>snmp get</code>	Get from the supplied SNMP Object Identifier (OID).	329
<code>snmp getNext</code>	GetNext from the supplied SNMP OID.	330
<code>snmp list</code>	List all SNMP global parameters.	331
<code>snmp walk</code>	Walk from the supplied SNMP OID.	332

snmp config

Show/set global Simple Network Management Protocol (SNMP) parameters.

SYNTAX:

```
snmp config [RWCommunity = <string>]
            [ROCommunity = <string>]
            [sysContact = <quoted string>]
            [sysName = <quoted string>]
            [sysLocation = <quoted string>]
```

where:

RWCommunity	The read-write community name.	OPTIONAL
ROCommunity	The read-only community name.	OPTIONAL
	Note At least one of the communities must be specified, otherwise SNMP is disabled.	
sysContact	The SNMP system contact. The default is Service Provider .	OPTIONAL
sysName	The SNMP system name. The default is SpeedTouchXXX (where XXX is the variant).	OPTIONAL
sysLocation	The SNMP system location. The default is Customer Premises .	OPTIONAL

EXAMPLE:

```
=>snmp list

NO COMMUNITY NAMES SPECIFIED, SNMP IS SWITCHED OFF
-- Specify at least one community name to enable snmp --

SNMP System Contact      : Service Provider
SNMP System Name         : SpeedTouch 546
SNMP System Location     : Customer Premises
=>snmp config RWCommunity=ReadWrite
=>snmp list

Read-write SNMP community name : *****
Read-only SNMP community name  : not specified
SNMP System Contact          : Service Provider
SNMP System Name             : SpeedTouch 546
SNMP System Location         : Customer Premises
=>
```

snmp get

Get from the supplied SNMP Object Identifier (OID).
 For example get ObjectID=.1.3.6.1.2.1.1.1.0.

SYNTAX:

```
snmp get [ObjectID = <string>]
```

where:

<p>ObjectID The Object Identifier. Object ID to get from ... must include the instance which is 0 for scalar objects. For example .1.3.6.1.2.1.1.1.0 sysDescription.</p> <p>Note If not specified, the sysDescription OID .1.3.6.1.2.1.1.1.0 is assumed. Its value is SpeedTouch™.</p>	<p>OPTIONAL</p>
--	-----------------

EXAMPLE:

```
=>snmp get
VB_octetStr .1.3.6.1.2.1.1.1.0                    SpeedTouch™ 516
=>
```

RELATED COMMANDS:

<p>snmp getNext</p> <p>snmp walk</p>	<p>GetNext from the supplied SNMP OID.</p> <p>Walk from the supplied SNMP OID.</p>
--------------------------------------	--

snmp getNext

getNext from the supplied SNMP OID.

SYNTAX:

```
snmp getNext [ObjectId = <string>]
```

where:

ObjectId	The Object Identifier. Object id to getNext from. For example .1.3.6.1.2.1.1 system returns sysDescription.	OPTIONAL
----------	---	----------

EXAMPLE:

```
=>snmp getNext ObjectId=.1.3.6.1.2.1.1.4.0
VB_octetStr .1.3.6.1.2.1.1.5.0 Sascha
=>
```

RELATED COMMANDS:

snmp get	Get from the supplied SNMP Object Identifier (OID).
snmp walk	Walk from the supplied SNMP OID.

snmp list

List all SNMP global parameters.

SYNTAX:

```
snmp list
```

EXAMPLE:

```
=>snmp list
Read-write SNMP community name : private
Read-only SNMP community name : public
SNMP System Contact      : Service Provider
SNMP System Name         : SpeedTouch516
SNMP System Location     : Customer Premises
=>
```

snmp walk

Walk from the supplied SNMP OID.

SYNTAX:

```
snmp walk [ObjectId = <string>]
```

where:

ObjectID	The Object Identifier. Object ID to walk from. For example .1.3.6.1.2.1.1 system walks the system group.	OPTIONAL
----------	--	----------

EXAMPLE:

```
=>snmp walk ObjectId=.1.3.6.1.2.1.1
VB_octetStr .1.3.6.1.2.1.1.1.0      SpeedTouch™ 516
VB_objId    .1.3.6.1.2.1.1.2.0      .1.3.6.1.4.1.637.61.2
VB_timeTicks .1.3.6.1.2.1.1.3.0    2927636
VB_octetStr .1.3.6.1.2.1.1.4.0      Service Provider
VB_octetStr .1.3.6.1.2.1.1.5.0      Sascha
VB_octetStr .1.3.6.1.2.1.1.6.0      Customer Premises
VB_integer  .1.3.6.1.2.1.1.7.0      72
=>
```

RELATED COMMANDS:

<code>snmp get</code>	Get from the supplied SNMP Object Identifier (OID).
<code>snmp getNext</code>	GetNext from the supplied SNMP OID.

Software Commands

Introduction

This chapter describes the commands of the **software** command group.

Contents

This chapter covers the following commands:

<code>software version</code>	Display the software version.	334
<code>software upgrade</code>	Reboot the modem to initiate the SW upgrade.	335

software version

Display the software version.

SYNTAX:

```
software version
```

EXAMPLE:

```
=>software version  
Flash image : 5.2.7.5.0  
Build name  : ZZTVAA5.275  
=>
```


software upgrade

Reboot the modem to initiate the SW upgrade.

New software available on a remote LAN host will be uploaded to the modem.

SYNTAX:

```
software upgrade
```


Switch Commands

Introduction

This chapter describes the commands of the **switch** command group.



These commands are only applicable for the SpeedTouch™546.

Contents

This chapter covers the following commands:

switch group flush	Set all ports to the default settings.	338
switch group list	List all configured groups.	339
switch group move	Move a specified port to a specified group.	340
switch mirror capture	Define the specified port to be the Mirror Capture Port.	341
switch mirror egress	Enable or disable the specified port to be the Mirror Egress Port.	342
switch mirror ingress	Enable or disable the specified port to be the Mirror Ingress Port.	343

switch group flush

Set all ports to the default settings.



All the ports are put in group 0.

SYNTAX:

```
switch group flush
```

EXAMPLE:

```
=>switch group list
Group 0 Ports: 4
Group 1 Ports: 1
Group 2 Ports: 2 3
=>switch group flush
=>switch group list
Group 0 Ports: 1 2 3 4
=>
```

RELATED COMMANDS:

switch group list

List all configured groups.

switch group move

Move a specified port to a specified group.

switch group list

List all configured groups.

SYNTAX:

```
switch group list
```

EXAMPLE:

```
=>switch group list  
Group 0 Ports: 4  
Group 1 Ports: 1  
Group 2 Ports: 2 3
```

RELATED COMMANDS:

switch group flush

Set all ports to the default settings.

switch group move

Move a specified port to a specified group.

switch group move

Move a specified port to a specified group.

SYNTAX:

```
switch group move  group = <number{0-4}>
                   port = <number{1-4}>
```

where:

group	The group ID to which the port must be moved.	REQUIRED
port	The port to be moved.	REQUIRED

EXAMPLE:

```
=>switch group list
Group 0 Ports: 1 2 3 4
=>switch group move group=3 port=1
=>switch group list
Group 0 Ports: 2 3 4
Group 3 Ports: 1
```

RELATED COMMANDS:

switch group flush	Set all ports to the default settings.
switch group list	List all configured groups.

switch mirror capture

Define the specified port to be the Mirror Capture Port.



Only one port can be the Mirror Capture Port at any one time.

SYNTAX:

```
switch mirror capture port = <number{1-4}>
```

where:

port	The port to be the Mirror Capture Port.	REQUIRED
Note	If no port number is specified, then the port number of the Mirror Capture Port is shown.	

EXAMPLE:

```
=>switch mirror capture port=2
=>switch mirror capture
Mirror capture port = 2
```

RELATED COMMANDS:

- [switch mirror egress](#) Enable or disable the specified port to be the Mirror Egress Port.
- [switch mirror ingress](#) Enable or disable the specified port to be the Mirror Ingress Port.

switch mirror egress

Enable or disable the specified port to be the Mirror Egress Port.



Only one port can be the Mirror Egress Port at any one time.
But a port can be the Mirror Egress Port and the Mirror Ingress Port at the same time.

SYNTAX:

```
switch mirror egress port = <number{1-4}>
                        [state = <{enabled | disabled}>]
```

where:

port	The port to be the Mirror Egress Port.	REQUIRED
Note	If no port number is specified, then the port number of the current Mirror Egress Port is shown.	
state	This parameter permits to enable/disable a port as Mirror Egress Port.	OPTIONAL

EXAMPLE:

Here port 1 is enabled as Mirror Egress Port

```
=>switch mirror egress port=1
=>switch mirror egress
Egress mirror port = 1
```

Here port 1 is disabled as Mirror Egress Port

```
=>switch mirror egress
Egress mirror port = 1
=>switch mirror egress port=1 state=disabled
=>switch mirror egress
=>
```

RELATED COMMANDS:

[switch mirror capture](#)

Define the specified port to be the Mirror Capture Port.

[switch mirror ingress](#)

Enable or disable the specified port to be the Mirror Ingress Port.

switch mirror ingress

Enable or disable the specified port to be the Mirror Ingress Port.



Only one port can be the Mirror Ingress Port at any one time.
But a port can be the Mirror Egress Port and the Mirror Ingress Port at the same time.

SYNTAX:

```
switch mirror ingress port = <number{1-4}>
                        [state = <{enabled | disabled}>]
```

where:

port	The port to be the Mirror Ingress Port.	REQUIRED
Note	If no port number is specified, then the port number of the current Mirror Ingress Port is shown.	
state	This parameter permits to enable/disable a port as Mirror Ingress Port.	OPTIONAL

EXAMPLE:

Here port 2 is enabled as Mirror Ingress Port

```
=>switch mirror ingress port=2
=>switch mirror ingress
Ingress mirror port = 2
```

Here port 2 is disabled as Mirror Ingress Port

```
=>switch mirror ingress
Ingress mirror port = 2
=>switch mirror ingress port=2 state=disabled
=>switch mirror ingress
=>
```

RELATED COMMANDS:

- [switch mirror capture](#) Define the specified port to be the Mirror Capture Port.
- [switch mirror egress](#) Enable or disable the specified port to be the Mirror Egress Port.

System Commands

Introduction

This chapter describes the commands of the **system** command group.

Contents

This chapter covers the following commands:

<code>system clearpassword</code>	Clear the current SpeedTouch™ system password.	346
<code>system config</code>	Show/set SpeedTouch™ system configuration parameters.	347
<code>system flush</code>	Flush current SpeedTouch™ system configuration.	348
<code>system reboot</code>	Reboot the SpeedTouch™.	349
<code>system reset</code>	Reset the SpeedTouch™ to its factory default settings and reboot the device.	350
<code>system setpassword</code>	Set/change the current SpeedTouch™ system password.	351
<code>system stats</code>	Show/set the SpeedTouch™ CPU and memory statistics.	352

system clearpassword

Clear the current SpeedTouch™ system password.



To avoid unrestricted and unauthorized access to the SpeedTouch™, it is highly recommended to make sure the SpeedTouch™ is protected by a system password (can be set via the command **: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 the current SpeedTouch™ system password.

system config

Show/set SpeedTouch™ system configuration parameters.



For a good operation of 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}>]
              [drst = <{disabled | enabled}>]
              [led = <{green | red | orange | flash | off}>]
              [digestauth = <{disabled | enabled}>]
              [strictusername = <{disabled | enabled}>]
              [dcache = <{disabled | enabled}>]
```

where:

upnp	Enable or disable UPnP discovery. The default is enabled .	OPTIONAL
mdap	Enable or disable proprietary discovery protocol. The default is enabled .	OPTIONAL
drst	Enable or disable DrSpeedTouch access. The default is enabled .	OPTIONAL
led	Set the SpeedTouch™ system LED color. Choose between: <ul style="list-style-type: none"> ▶ green: solid green ▶ red: solid red ▶ orange: solid orange ▶ flash: toggle between green and orange ▶ off: LED is off. The default is green .	OPTIONAL
digestauth	Enable or disable HTTP digest authentication. The default is disabled .	OPTIONAL
strictusername	Enable or disable strict username check. The default is disabled .	OPTIONAL
dcache	Enable or disable data cache. The default is enabled .	OPTIONAL
Note For internal use only. Do NOT alter in any way.		

EXAMPLE:

```
=>system config
upnp discovery      : enabled
mdap discovery      : enabled
drst support        : enabled
digest authentication : disabled
strict username     : disabled
=>
```

system flush

Flush current SpeedTouch™ system configuration.

The system password and the system configuration settings (except dcache) will be flushed.



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



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  factory yes/no = <{yes | no}>
               proceed no/yes = <{no | yes}>
```

where:

factory yes/no	Choose between: <ul style="list-style-type: none"> ▶ yes: the SpeedTouch™ will be reset to factory default settings. All the user and ISP specific settings will be deleted. Note Connectivity with the ISP network might be lost. ▶ no: only the user specific settings will be deleted. 	REQUIRED
proceed no/yes	Confirmation for resetting the SpeedTouch™. Choose between: <ul style="list-style-type: none"> ▶ yes: the SpeedTouch™ will be reset to default settings and will be rebooted. ▶ no: the reset command will be discarded. 	REQUIRED

EXAMPLE:

```
=>system reset
-----
!! WARNING !!
-----
The modem will be reset to (factory) defaults clearing all user (and ISP) settings.
Specifying <factory=yes> deletes user and ISP specific settings.
                        Connectivity with the ISP network might be lost.
    <factory=no> deletes user specific settings only.
factory yes/no = no
proceed no/yes = no
:system reset factory yes/no=no proceed no/yes=no
=>
=>system reset
-----
!! WARNING !!
-----
The modem will be reset to (factory) defaults clearing all user (and ISP) settings.
Specifying <factory=yes> deletes user and ISP specific settings.
                        Connectivity with the ISP network might be lost.
    <factory=no> deletes user specific settings only.
factory yes/no = yes
proceed no/yes = yes
:system reset factory yes/no=yes proceed no/yes=yes

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


system setpassword

Set/change the current SpeedTouch™ system password.



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

SYNTAX:

```
system setpassword [userid = <string>]
                  password = <password>
```

where:

userid	The new access userid.	OPTIONAL
password	The new access password.	REQUIRED

EXAMPLE:

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

RELATED COMMANDS:

system clearpassword Clear the current SpeedTouch™ system password.

system stats

Show/set the SpeedTouch™ CPU and memory statistics.

SYNTAX:

```
system stats [reset = <{no | yes}>]
```

where:

reset	Reset CPU statistics.	OPTIONAL
-------	-----------------------	----------

EXAMPLE:

```
=>system stats
Memory statistics:
-----
CHIP memory      total/used/free/min (in KB): 891/517/374/374
Application memory total/used/free/min (in KB): 2978/1592/1385/1323
=>
```

DESCRIPTION:

CHIP memory	Memory used by the CPU (first MB from the RAM) – not cached since it has to be realtime.
Application memory	Memory used by the applications.
min	The least amount of free memory detected during the uptime of the SpeedTouch™.

Systemlog Commands

Introduction

This chapter describes the commands of the **systemlog** command group.

Contents

This chapter covers the following commands:

systemlog flush	Flush all messages in the internal SpeedTouch™ Syslog message buffer.	354
systemlog show	Show syslog messages in the internal SpeedTouch™ Syslog message buffer.	355
systemlog send	Send syslog messages from the internal SpeedTouch™ Syslog message buffer to a specified local or remote syslog server host.	356

systemlog flush

Flush all messages in the internal SpeedTouch™ Syslog message buffer.

SYNTAX:

```
systemlog flush
```

systemlog show

Show syslog messages in the internal SpeedTouch™ Syslog message buffer.

SYNTAX:

```
systemlog show [fac = <supported facility name>]
               [sev = <supported severity name>]
               [hist = <{no | yes}>]
```

where:

fac	Specify the facility name of the syslog messages to be shown. Use one of the supported facility names (see "Supported Syslog Facilities" on page 379). Note If not specified, the messages of all the facilities will be shown.	OPTIONAL
sev	Specify the lowest priority severity of the syslog messages to be shown. Only the messages with a severity as specified or higher will be shown. Use one of the supported severity names (see "Supported Syslog Severities" on page 380). Note If not specified, the messages of all the facilities will be shown.	OPTIONAL
hist	Show messages over several SpeedTouch™ reboots (yes) or show only messages since latest startup (no). Note If not specified, only the recent messages will be shown.	OPTIONAL

EXAMPLE:

```
=>syslog msgbuf show fac=kern sev=emerg hist=yes
<0> SysUpTime: 14:45:43 KERNEL Controlled restart (after internal error or explicit system reboot)
<0> SysUpTime: 02:58:18 KERNEL Controlled restart (after internal error or explicit system reboot)
<0> SysUpTime: 04 days 04:52:37 KERNEL Controlled restart (after internal error or explicit system reboot)
<0> SysUpTime: 00:00:41 KERNEL Controlled restart (after internal error or explicit system reboot)
=>syslog msgbuf show fac=kern sev=warning hist=yes
<4> SysUpTime: 00:00:00 KERNEL Cold restart
<0> SysUpTime: 14:45:43 KERNEL Controlled restart (after internal error or explicit system reboot)
<4> SysUpTime: 00:00:00 KERNEL Warm restart
<0> SysUpTime: 02:58:18 KERNEL Controlled restart (after internal error or explicit system reboot)
<4> SysUpTime: 00:00:00 KERNEL Warm restart
<0> SysUpTime: 04 days 04:52:37 KERNEL Controlled restart (after internal error or explicit system reboot)
<4> SysUpTime: 00:00:00 KERNEL Warm restart
<0> SysUpTime: 00:00:41 KERNEL Controlled restart (after internal error or explicit system reboot)
=>
```

systemlog send

Send syslog messages from the internal SpeedTouch™ Syslog message buffer to a specified local or remote syslog server host.



There will be no notification on whether the host has received the messages or not.

SYNTAX:

```
systemlog send [fac = <supported facility name>]
               [sev = <supported severity name>]
               [hist = <{no | yes}>]
               dest = <ip-address>
```

where:

fac	Specify the facility name of the syslog messages to be shown. Use one of the supported facility names (see “ Supported Syslog Facilities” on page 379). Note If not specified, the messages of all the facilities will be shown.	OPTIONAL
sev	Specify the lowest priority severity of the syslog messages to be shown. Only the messages with a severity as specified or higher will be shown. Use one of the supported severity names (see “ Supported Syslog Severities” on page 380). Note If not specified, the messages of all the facilities will be shown.	OPTIONAL
hist	Show messages over several SpeedTouch™ reboots (yes) or show only messages since latest startup (no). Note If not specified, only the recent messages will be shown.	OPTIONAL
dest	The IP address of the remote host on the local or remote network, i.e. the collector's IP address, to send the syslog messages to.	REQUIRED

TD Commands

Introduction

This chapter describes the commands of the **td** command group.

Contents

This chapter covers the following commands:

<code>td call</code>	Call a 'Trace & Debug' command.	358
----------------------	---------------------------------	-----

td call

Call a 'Trace & Debug' command.



For qualified personnel only.

SYNTAX:

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

where:

cmd	Quoted 'Trace & Debug' command string.
-----	--

REQUIRED

UPnP Commands

Introduction

This chapter describes the commands of the **upnp** command group.

Contents

This chapter covers the following commands:

<code>upnp config</code>	Configure UPnP parameter(s).	360
<code>upnp flush</code>	Flush the UPnP configuration.	361
<code>upnp list</code>	List the devices and services currently offered by the SpeedTouch™.	362

upnp config

Configure UPnP parameter(s).

SYNTAX:

```
upnp config [maxage = <number{60-999999}>]
            [defcservice = <string>]
            [writemode = <{full | natonly | readonly}>]
            [safenat = <{disabled | enabled}>]
            [preferredaddress = <ip-address>]
```

where:

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. The default is 1800 .	OPTIONAL
defcservice	This parameter allows to configure the connection service to be used by the DrSpeedTouch™ application during the troubleshooting process (only when several connection services are configured on the SpeedTouch™, else only the default connection service).	OPTIONAL
writemode	Choose the set of rules to limit remote access from UPnP. Choose between: <ul style="list-style-type: none"> ▶ full: the host will accept all the UPnP SET and GET actions. ▶ natonly: GET and NAT related SET actions will be accepted, all other actions will be ignored. ▶ readonly: the UPnP control point will only be able to retrieve information, all the SET actions are ignored. The default is full .	OPTIONAL
safenat	Enable or disable check on safe NAT entries. If this check is enabled, all NAT create/delete requests for a LAN side IP address different from the source IP address of the UPnP message will be discarded. The default is disable .	OPTIONAL
preferredaddress	Preferred IPaddress for UPnP advertisements (enter 0.0.0.0 for none).	OPTIONAL

EXAMPLE:

```
=>upnp config
ssdp max-age           : 1800
default connection service :
write mode             : full
safe nat entries       : disabled
preferred address      :
multicast interface(s)  : eth0 (10.0.0.138)
=>
```

upnp flush

Flush the UPnP configuration.

The configuration will be reset to defaults.

SYNTAX:

```
upnp flush
```

upnp list

List the devices and services currently offered by the SpeedTouch™.



Use this command to check whether a PPP connection is properly configured and thus advertised as a PPP service.

SYNTAX:

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

where:

verbose	Verbose level. The default is 1 .	OPTIONAL
---------	---	----------

EXAMPLE:

```
=>upnp list
Advertising UPnP devices on address: 10.0.0.138
----- device: IGD.xml -----
++ Root Device: urn:schemas-upnp-org:device:InternetGatewayDevice: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:PVC_1
-- Service 5: urn:upnp-org:serviceId:wanpppc:PPPoA_1
----- end -----
----- device: DRST.xml -----
++ Root Device: urn:schemas-upnp-org:device:DRST_Device:1
-- Service 1: urn:upnp-org:serviceId:layer3f_DRST
-- Service 2: urn:upnp-org:serviceId:devinfo_DRST
-- Service 3: urn:upnp-org:serviceId:lanhcm_DRST
-- Service 4: urn:upnp-org:serviceId:lanelc_DRST
-- Service 5: urn:upnp-org:serviceId:wancic_DRST
-- Service 6: urn:upnp-org:serviceId:wandsllc_DRST:PVC_1
-- Service 7: urn:upnp-org:serviceId:wanpppc_DRST:PPPoA_1
----- end -----
=>
```

Abbreviations

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

Abbreviation	Description
AAL5	ATM Adaption Layer 5
ACL	Access Control List
ADSL	Asymmetric Digital Subscriber Line
AES	Advanced Encryption System
AF	Assured Forwarding
AH	Authentication Header
AP	Access Point
ARP	Address Resolution Protocol
ATM	Asynchronous Transfer Mode
ATMF	ATM Forum
BGP	Border Gateway Protocol
BSSID	Base Service Set IDentifier
CA	Certificate Authority
CAPI	Common ISDN Application Interface
CC	Continuity Check
CEP	Certificate Enrollment Protocol
CHAP	Challenge Handshake Authentication Protocol
CIDR	Classless Inter Domain Routing
CLI	Command Line Interface
CLP	Cell Loss Priority
CO	Central Office
CPE	Customer Premises Equipment
CRL	Certificate Revocation List
CTD	Conformance Traffic Descriptor
DHCP	Dynamic Host Configuration Protocol
DN	Distinguished Name
DNS	Domain Name System
DPD	Dead Peer Detection
DSCP	Differentiated Services Code Point
DSD	Differentiated Service Delivery
DSL	Digital Subscriber Line
ECN	Explicit Congestion Notification
EF	Expedited Forwarding
EGP	Exterior Gateway Protocol
ESP	Encapsulating Security Payload

Abbreviation	Description
ETHoA	Ethernet over ATM
FCS	Frame Check Sequence
FTP	File Transfer Protocol
GRE	General Routing Encapsulation
GRP	Gateway Routing Protocol
HDLC	High-level Data Link Control
HTTP	HyperText Transfer Protocol
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System
IGD	Internet Gateway Device
IGMP	Internet Group Management Protocol
IKE	Internet Key Exchange
IMAP	Internet Message Access Protocol
IP	Internet Protocol
IP oA	IP over ATM
IPCP	Internet Protocol Control Protocol
IPsec	IP Security
IRC	Internet Relay Chat
ISDN	Integrated Services Digital Network
ISI	Independent Set ID
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
MC	MultiCast
MD5	Message Digest 5
MLP	Multi-Level Password
MTU	Maximum Transmission Unit
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

Abbreviation	Description
PAP	Password Authentication Protocol
PAT	Port Address Translation
PBX	Private Branch Exchange
PFS	Perfect Forward Secrecy
PIP	Packet Interception Point
PKCS	Public Key Cryptography Standard
PKI	Public Key Infrastructure
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
PSK	Pre-Shared Key
PVC	Permanent Virtual Channel
QoS	Quality of Service
RIP	Routing Information Protocol
RTMP	RouTing Maintenance Protocol
RTSP	Real Time Streaming Protocol
RTT	Round Trip Time
SAs	Security Associations
SFTP	Secure File Transfer Protocol
SHDSL	Single Pair High-speed Digital Subscriber Line
SIP	Session Initiation Protocol
SLA	Service Level Agreement
SMTP	Simple Mail Transfer Protocol
SNAP	Sub Network Access Protocol
SNMP	Simple Network Management Protocol
SNPP	Simple Network Paging Protocol
SNR	Signal-to-Noise Ratio
SNTP	Simple Network Time Protocol
SSH	Secure SHell
SSID	Service Set IDentifier
SVC	Switched Virtual Channel
TCP	Transmission Control Protocol
TFTP	Trivial File Transfer Protocol
TKIP	Temporal Key Integrity Protocol
ToS	Type of Service

Abbreviation	Description
TTL	Time To Live
UA	User Agent
UDP	User Datagram Protocol
ULP	Upper Layer Protocol
UPnP	Universal Plug and Play
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
USB	Universal Serial Bus
VC	Virtual Channel
VCI	Virtual Channel Identifier
VCMUX	Virtual Channel MULTipleXing
VDSL	Very high speed Digital Subscriber Line
VLAN	Virtual Local Area Network
VLSM	Variable Length Subnet Masking
VP	Virtual Path
VPI	Virtual Path Identifier
VPN	Virtual Private Networking
WAN	Wide Area Network
WDS	Wireless Distribution System
WEP	Wired Equivalent Privacy
WFQ	Weighted Fair Queueing
WINS	Windows Internet Naming Service
WLAN	Wireless LAN
WPA	Wi-Fi Protected Access
WRR	Weighted Round Robin
WWW	World Wide Web
ZIS	Zone Information System

Syslog Messages

Introduction

This chapter lists the different Syslog messages.

Contents

This chapter covers the following Syslog messages:

Auto-PVC Module	368
Configuration Module	368
DHCP Client Module	368
DHCP Relay Module	369
DHCP Server Module	369
Firewall Module	370
HTTP Module	370
Kernel Module	370
Linestate Module	370
Login Module	371
NAPT Module	371
PPP Module	371
PPTP Module	372
Routing Module	372
Software Module	372
UPnP Module	372

Auto-PVC Module

Facility	Severity	Contents
LOCAL5	WARNING	AUTOPVC script <script_name> failed
LOCAL5	WARNING	AUTOPVC script <script_name> failed
LOCAL5	WARNING	AUTOPVC script <script_name> (name1, qosb_name) failed
LOCAL5	WARNING	AUTOPVC script <script_name> (name1, qosb_name, name2) failed
LOCAL5	WARNING	AUTOPVC script 'autopvc_change_qos (itable[i].intf, name1, qosb_name) failed
LOCAL5	WARNING	AUTOPVC script <script_name> (name1, name2) failed

Configuration Module

Facility	Severity	Contents
USER	INFO	CONFIGURATION saved after running Embedded Setup Wizard
USER	INFO	CONFIGURATION saved by user <user_id>
USER	INFO	CONFIGURATION backup by user to file <filename>
USER	INFO	CONFIGURATION <conf_version> upgraded to version <version>
KERN	INFO	SYSTEM reset by user <user_id> to factory defaults: user settings deleted

DHCP Client Module

Facility	Severity	Contents
LOCAL2	WARNING	DHCP lease ip-address <ip-address> bound to intf <intf_id>
LOCAL2	WARNING	DHCP intf <intf_id> renews lease ip-address <ip-address>
LOCAL2	WARNING	DHCP intf <intf_id> rebinds lease ip-address <ip-address> from server (<ip-address>)
LOCAL2	WARNING	DHCP offer received from <ip-address> (can be relay agent) for intf <intf_id>
LOCAL2	WARNING	DHCP server (<ip-address>) offers <ip-address> to intf <intf_id>
LOCAL2	WARNING	DHCP unable to configure ip address: <ip-address> (bootp-reply)
LOCAL2	WARNING	DHCP bootp lease ip-address <ip-address> bound to intf <intf_id> from server (<ip-address>)
LOCAL2	WARNING	DHCP <ip-address> already configured on intf <intf_id>: failure
LOCAL2	WARNING	DHCP <ip-address> (<ip-address>) set on intf <intf_id>: {faillurellok}
LOCAL2	WARNING	DHCP <ip-address> deleted: {faillurellok}

DHCP Relay Module

Facility	Severity	Contents
LOCAL2	WARNING	DHCP relay: Dropping boot rqs on interface <intf_id> due to invalid giaddr for server (<ip-address>)
LOCAL2	WARNING	DHCP relay: Dropping boot reply with invalid relay agent option from <intf_id>
LOCAL2	WARNING	DHCP relay: Dropping boot request containing the relay agent option from <intf_id>
LOCAL2	WARNING	DHCP relay: Dropping boot reply to unknown interface from <intf_id>
LOCAL2	WARNING	DHCP relay: Dropping boot reply to inactive interface <intf_id>
LOCAL2	WARNING	DHCP relay: Dropping boot reply to inactive interface <intf_id>
LOCAL2	WARNING	DHCP relay: Dropping boot request packet with spoofed giaddr field from <intf_id>
LOCAL2	WARNING	DHCP relay: Dropping boot request received on unknown interface from <intf_id>
LOCAL2	WARNING	DHCP relay: Dropping boot request on inactive interface <intf_id>
LOCAL2	WARNING	DHCP relay: Dropping boot request with invalid hops field on interface <intf_id>
LOCAL2	WARNING	DHCP relay: Dropping boot request with invalid giaddr on interface <intf_id>

DHCP Server Module

Facility	Severity	Contents
LOCAL2	WARNING	DHCP server: %s cannot be send due to invalid server identifier
LOCAL2	WARNING	DHCP server: DHCPACK cannot be send due to invalid server identifier
LOCAL2	WARNING	DHCP server: DHCPNAK cannot be send due to invalid server identifier
LOCAL2	WARNING	DHCP Auto DHCP: server detected on LAN, own dhcp server disabled
LOCAL2	WARNING	DHCP Auto DHCP: no server detected on LAN, SpeedTouch server started
LOCAL2	WARNING	DHCP Auto DHCP: search for DHCP server stopped
LOCAL2	WARNING	DHCP server up
LOCAL2	WARNING	DHCP server went down

Firewall Module

Facility	Severity	Contents
AUTH	WARNING	FIREWALL Hook: <hookname> Rule ID:<rule_id> Protocol: ICMP Src_ip: <ip_address> Dst_ip: <ip_address> ICMP message type: <message_type_name message_type_id > Action: <action>
AUTH	WARNING	FIREWALL Hook: <hookname> Rule ID:<rule_id> Protocol: <protocol_name> Src_ip_port: <ip-address:ip_port> Dst_ip_port: <ip-address:ip_port> Action: <action>

HTTP Module

Facility	Severity	Contents
AUTH	NOTICE	LOGOUT User <user_id> logged out on <connection_type> (<ip-address>)
AUTH	NOTICE	LOGOUT User <user_id> logged out on <connection_type>
AUTH	NOTICE	LOGOUT <session_name> session of user <user_id> killed (<ip-address>)
AUTH	NOTICE	LOGOUT <session_name> session of user <user_id> killed
AUTH	NOTICE	LOGIN User <user_id> tried to login on <connection_type> (from <ip-address>)
AUTH	NOTICE	LOGIN User <user_id> logged in on <connection_type> (from <ip-address>)
AUTH	NOTICE	LOGIN User logged in on <connection_type> (<ip-address>)
AUTH	NOTICE	LOGIN User <user_id> tried to log in on <connection_type>

Kernel Module

Facility	Severity	Contents
KERN	WARNING	KERNEL cold reset
KERN	WARNING	KERNEL warm reset
KERN	EMERG	KERNEL Controlled restart (after internal error or explicit system reboot)

Linestate Module

Facility	Severity	Contents
LOCAL5	NOTICE	xDSL linestate up (downstream: <bitrate_in> kbit/s, upstream: <bitrate_out> kbit/s)
LOCAL5	NOTICE	xDSL linestate down

Login Module

Facility	Severity	Contents
AUTH	NOTICE	LOGIN User <username> logged <in/out> on telnet (<ip address>)
AUTH	NOTICE	LOGIN User <username> logged in on http (<ip-address>)

NAPT Module

Facility	Severity	Contents
LOCAL4	INFO	NAPT Protocol: <TCP UDP ICMP> Open port: <port> Helper: <app_name> => <""failed"" ""ok"">

PPP Module

Facility	Severity	Contents
LOCAL0	WARNING	PPP Link up (<intf name>)
LOCAL0	WARNING	PPP Link down (<intf name>)
AUTH	ERROR	PPP PAP authentication failed (<intf name>) [protocol reject]
AUTH	INFO	PPP PAP on intf <intf_id> no response to PAP authenticate-request
AUTH	NOTICE	PPP PAP remote user <remote_user_name> successful authenticated
AUTH	ERROR	PPP PAP authentication for remote user <remote_user> failed
AUTH	DEBUG	PPP PAP Authenticate Ack received
AUTH	DEBUG	PPP PAP Authenticate Nack received
AUTH	DEBUG	PPP PAP Authenticate Request sent
AUTH	ERROR	PPP CHAP authentication failed (<intf name>)
AUTH	ERROR	PPP CHAP authentication failed [protocol reject(server)]
AUTH	ERROR	PPP CHAP authentication failed [protocol reject(client)]
AUTH	DEBUG	PPP CHAP Receive challenge (rhost = <hostname>)
AUTH	INFO	PPP CHAP Chap receive success : authentication ok
AUTH	DEBUG	PPP CHAP Challenge Send (Id = <challenge_id>)
AUTH	DEBUG	PPP CHAP Send status response: {ack nack}

PPTP Module

Facility	Severity	Contents
LOCAL0	WARNING	PPTP tunnel (<Pbname>) up:(<ip addr>)
LOCAL0	WARNING	PPTP tunnel (<Pbname>) down:(<ip addr>)

Routing Module

Facility	Severity	Contents
SECURITY	INFO	LABEL Rule Id:<rule_id> Protocol: ICMP Src_ip: <ip_address> Dst_ip: <ip_address> ICMP message type: <message_type_name message_type_id > Label: <label_name>
SECURITY	INFO	LABEL Rule Id:<rule_id> Protocol: <protocol_name> Src_ip: <ip_address> Dst_ip: <ip_address> Label: <label_name>

Software Module

Facility	Severity	Contents
KERN	INFO	SOFTWARE Copying all data files from <file_location> to <file_location>
KERN	INFO	SOFTWARE Switchover going down in <number> seconds
KERN	INFO	SOFTWARE No passive software found, duplicating active software
KERN	INFO	SOFTWARE Duplication of active software failed

UPnP Module

Facility	Severity	Contents
WARNING	SECURITY	UPnP<ActionName> refused for ip=<ip_address>
NOTICE	SECURITY	UPnP <ActionName> (<Error_string>) for ip=<ip_address>

Supported Key Names

Contents

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

Contents

This chapter covers the following supported Key names:

Supported IP Protocol Names	374
Supported TCP/UDP Port Names	375
Supported ICMP Type Names	378
Supported Syslog Facilities	379
Supported Syslog Severities	380

Supported 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
udp	17	User Datagram Protocol (UDP)
rsvp	46	Resource Reservation Protocol (RSVP)
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
rsh	101	Y	Y	Remote Shell
rlogin	102	Y	Y	Remote Login
rsh	102	Y	Y	Remote Shell
rlogin	102	Y	Y	Remote Login
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
nntp	119	Y	Y	Network News Transfer Protocol (NNTP)
ntp	123	Y	Y	Network Time Protocol (NTP)

Port name	Number	TCP	UDP	Description
sntp	123	Y	Y	Simple Network Time Protocol (SNTP)
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	Internet Message 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	Novell
imap3	220	Y	Y	Internet Message Access Protocol (IMAP) v3
clearcase	371	Y	Y	ClearCase
ulistserv	372	Y	Y	UNIX Listserv
ldap	389	Y	Y	Lightweight 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
biff	512	-	Y	Used by mail system to notify users of new mail received
exec	512	Y	-	Remote process execution
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
syslog	514	-	Y	Syslog
printer	515	Y	Y	Spooler
talk	517	Y	Y	Like Tenex link, but across machine
ntalk	518	Y	Y	NTalk

Port name	Number	TCP	UDP	Description
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
new-rwho	540	Y	Y	uucpd remote who is
uucp	540	Y	Y	uucpd
uucp-rlogin	540	Y	Y	uucpd remote login
rtsp	554	Y	Y	Real Time Streaming Protocol (RTSP)
whoami	565	Y	Y	whoami
ipcserver	600	Y	Y	SUN IPC Server
doom	666	Y	Y	Doom ID Software
ils	1002	Y	Y	Internet Locator Service (ILS)
h323	1720	Y	Y	H323 Host Call Secure
nfsd	2049	Y	Y	NFS daemon
sip	5060	Y	Y	Session Initiation Protocol (SIP)
xwindows	6000	Y	Y	X windows
irc-u	6667	Y	Y	Internet Relay Chat (IRC) Protocol
realaudio	7070	Y	Y	realaudio
httpproxy	8080	Y	Y	HTTP Proxy

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

Supported Syslog Facilities

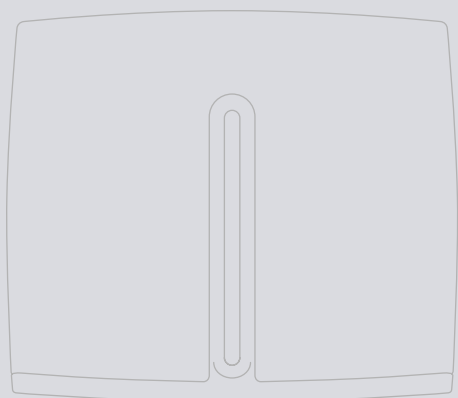
For more information on the listed Syslog facilities, see RFC3164.

Facility Name	Hierarchy Code	Syslog facility (listed according to decreasing importance)
kern	0	Kernel messages
user	8	User-level messages
mail	16	Mail system
daemon	24	System daemons
auth	32	Authorization messages
syslog	40	Syslog daemon messages
lpr	48	Line printer subsystem
news	56	Network news subsystem
uucp	64	UUCP subsystem
cron	72	Clock daemon
security	80	Security messages
ftp	88	FTP daemon
ntp	96	NTP subsystem
audit	104	Log audit
logalert	112	Log alert
clock	120	Clock daemon
local0 local1 local2 local3 local4 local5 local6 local7	128 136 144 152 160 168 176 184	Local use messages
all	-	All facilities (SpeedTouch™ specific facility parameter value.

Supported Syslog Severities

For more information on the listed Syslog severities, see RFC3164.

Severity Name	Hierarchy Code	Syslog severity (listed according to decreasing importance)
emerg	0	Emergency conditions, system unusable
alert	1	Alert conditions, immediate action is required
crit	2	Critical conditions
err	3	Error conditions
warning	4	Warning conditions
notice	5	Normal but significant conditions
info	6	Informational messages
debug	7	Debug-level messages



Need more help?

Additional help is available online at www.speedtouch.com