

SPEED TOUCH 570

CLI Reference Guide

Wireless ADSL Router



Status Released

Change Note BD F aa 39813

Short Title CD-RG AST570 CLI

All rights reserved. Passing on and copying of this document, use and communication of its contents not permitted without written authorization from Alcatel.

Contents

Preface	5
1 ADSL Commands	21
2 Bridge Commands	23
3 CIP Commands	39
4 Config Commands	49
5 DHCP Commands	55
6 DNS Commands	93
7 Firewall Commands	115
8 IP Commands	141
9 MER Commands	163
10 NAT Commands	173
11 Phonebook Commands	189
12 PPP Commands	199
13 PPTP Commands	217
14 Software Commands	225
15 System Commands	231
16 TD Commands	239
17 Wireless Commands	241
Index	265

Preface

Welcome to the **Alcatel SpeedTouch™ 570 Command Line Interface Reference Guide** !

This Reference Guide aims to give the fastidious user a concise, practical and easy to use document for configuring the **SpeedTouch™ 570** via its character based Command Line Interface.

Although the **SpeedTouch™ 570** Web interface is adequate enough for most users, access via the CLI may be still important for advanced and detailed configuration and troubleshooting.

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

▶ **Alcatel SpeedTouch™ 570**

The Reference Guide consists of three main parts:

▶ **Part 1 : CLI Navigation**

This part is meant to make the user familiar with the use and operation of the **SpeedTouch™ 570** CLI. Next to describing the various access methods to the CLI, this part will describe in brief some general manipulations to navigate through and to perform some operations on the CLI.

▶ **Part 2 : CLI Command Description**

This part forms the main part of this Reference Guide. Here all available CLI commands of the **SpeedTouch™ 570** products are alphabetically described per group selection.

Each command is described in a systematic manner:

- The full name of the CLI command (including the group selection)
- A short description of the CLI command, if needed completed by a description of the possible impact on the user and/or the **SpeedTouch™ 570**
- 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.

▶ **Part 3 : CLI Command Index**

This part allows the user to look up a command alphabetically in its incomplete form.

▶ **Trademarks**

The following trademarks are used in this document:

- Speed Touch™ is a trademark of the Alcatel Company
- Netscape® and Netscape Navigator® are registered trademarks of Netscape Communications Corporation
- Windows™ and Internet Explorer™ are trademarks of Microsoft Corporation
- Apple® and Mac®OS are registered trademarks of Apple Computer Inc.
- UNIX® is a registered trademark of UNIX System Laboratories, Inc.
- Ethernet™ is a trademark of Xerox Corporation.

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

▶ **Updates**

Due to the continuous evolution of the Alcatel DSL technology, existing products are regularly upgraded. Alcatel documentation changes accordingly.

For more information on the newest technological changes and documents, please consult the Alcatel web site at following URL:

<http://www.alcatel.com>
<http://www.alcateldsl.com>

▶ **Directive**

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

- disassemble, de-compile, reverse engineer, trace or otherwise analyze the equipment, its content, operation, or functionality, or otherwise attempt to derive source code (or the underlying ideas, algorithms, structure or organization) from the equipment or from any other information provided by Alcatel, except to the extent that this restriction is expressly prohibited by local law;
- copy, rent, loan, re-sell, sublicense, or otherwise transfer or distribute the equipment to others;
- modify, adapt or create a derivative work of the equipment;
- remove from any copies of the equipment any product identification, copyright or other notices;
- disseminate performance information or analysis (including, without limitation, benchmarks) from any source relating to the equipment.

Such acts not expressly approved by Alcatel will result in the loss of product warranty and will invalidate the user's authority to operate this equipment.

Alcatel SpeedTouch™ 570

CLI Navigation

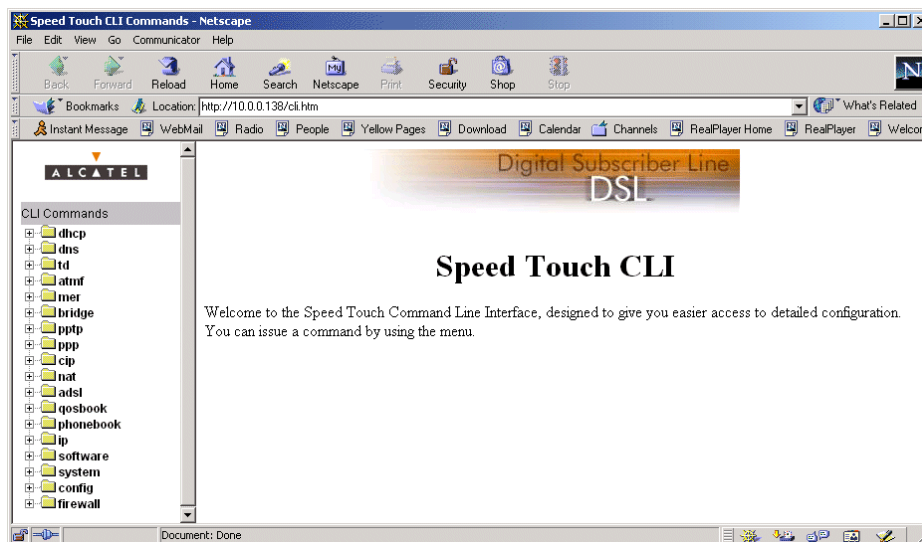
Accessing the Command Line Interface

Users can access the Command Line Interface via:

- ▶ The **SpeedTouch™ 570** CLI web pages
This requires that TCP/IP connectivity exists between the host from which the web browser is opened and the **SpeedTouch™ 570**
- ▶ A Telnet session
This requires that TCP/IP connectivity exists between the host from which the Telnet session is opened and the **SpeedTouch™ 570**.

Access via the Web Pages


The **SpeedTouch™ 570** CLI is accessible via its web interface. Browse to the **SpeedTouch™ 570** web pages and click **CLI** in the left frame. As a result the CLI web menu is opened in a new browser window:



You can open the CLI web pages directly by pointing the browser to the following URL: `http://10.0.0.138/cli.htm` (in which the `10.0.0.138` IP address should be replaced by the actual **SpeedTouch™ 570** IP address if needed).

All CLI groups and commands are placed in a menu. You can open a group by clicking the **+** mark next to a group name, or clicking the group name.

Clicking on a command name will execute it. Commands without parameters are indicated with  and are executed immediately.

Commands which require additional parameters are indicated with . After you configured all parameters you must click **Apply** to execute the command.

Navigation and Manipulation

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

Command group Navigation

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

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

EXAMPLE:

```
=>help
Following commands are available :
help      : Displays this help information
?         : Displays this help information
exit      : Exits this shell.
..        : Exits group selection.

Following command groups are available :
wireless  dhcp      dns      td        mer
bridge    pptp      ppp      cip       nat
adsl      qosbook   phonebook ip        software
system    config    firewall

=>
```

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

EXAMPLE:

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

The Help Command

Execute **help** from top level to list all available command groups for the **SpeedTouch™ 570**.

EXAMPLE:

```
=>help
Following commands are available :
help      : Displays this help information
?         : Displays this help information
exit      : Exits this shell.
..        : Exits group selection.

Following command groups are available :
wireless  dhcp      dns      td        mer
bridge    pptp      ppp      cip       nat
adsl      qosbook   phonebook ip        software
system    config    firewall

=>
```

You can execute the **help** 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]=>help
Following commands are available :

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

Following command groups are available :

chain     rule

[firewall]=>
```

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

EXAMPLE:

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

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

Following command groups are available :

chain        rule

=>
```

Entering **help** followed by a specific command, e.g. **help firewall assign** (starting from top level) or **help assign** (e.g. on 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.

=>
```

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 example, entering **a** at the firewall command group selection, followed by a **"Tab"** stroke results in the full **assign** command being completed. Entering **firewall a** from top level gives the same result.

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

EXAMPLE:

```
=>firewall
[firewall]=>a "Tab"
[firewall]=>assign
```

Going to the beginning or end of the Command Line

Go to the beginning of the Command Line by pressing **"Ctrl+A"**; to go to the end of the Command Line press **"Ctrl+E"**.

In the following example, the first || indicates the position of the cursor after pressing **"Ctrl+A"**, the second || the position of the cursor after pressing **"Ctrl+E"**.

EXAMPLE:

```
=>||list||
```

Breaking off Commands

You can break off a command by pressing **"Ctrl+G"**. This can be useful in a situation where a user is prompted to enter a value which it does not know and wants to abort the command. Instead of being prompted over and over again for the same value, this allows to break of the command.

In the example below **"Ctrl+G"** is pressed after the third prompt 'vpi ='. The command is broken of and the user returns to the command line.

EXAMPLE:

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

History of Commands

To retake previous commands press the up arrow "↑" and come back to more recent commands with the down arrow "↓". Press "Enter (↵)" to select and execute the retaken 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]=> "↑"
[firewall]=>:firewall list
```

Command Line Interface Top Level Structure

The following command groups are available:

- ▶ **adsl**
- ▶ **bridge**
- ▶ **cip**
- ▶ **config**
- ▶ **dhcp**
- ▶ **dns**
- ▶ **firewall**
- ▶ **ip**
- ▶ **mer**
- ▶ **nat**
- ▶ **phonebook**
- ▶ **ppp**
- ▶ **pptp**
- ▶ **software**
- ▶ **system**
- ▶ **td**
- ▶ **wireless**

Command Line Interface Commands

All CLI commands are commands that operate on, or configure, the **SpeedTouch™ 570**.

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

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

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

EXAMPLE:

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

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

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

EXAMPLE:

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

Alcatel SpeedTouch™ 570

CLI Command Description

1 ADSL Commands

The `adsl` command group is only applicable to the **SpeedTouch™ Pro with Firewall** ADSL/POTS variant, NOT to the **SpeedTouch™ Pro with Firewall** ADSL/ISDN and **SpeedTouch™ Pro with Firewall** SHDSL variant.

adsl (to access the ADSL level)
adsl info

adsl info

Show ADSL/POTS statistics and information about the **SpeedTouch™ Pro with Firewall** status.

SYNTAX:

```
adsl info
```

EXAMPLE:

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

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

Margin [dB]          :          Downstream  Upstream
Attenuation [dB]    :          31          31
                    :          26          13

Available Bandwidth
  Downstream         :          Cells      Kbits
  Upstream           :          2641      1014
                    :          301       115

Transfer statistics
  Total since power On
    Downstream       :          Cells      Kbits
    Upstream         :          185670    71297
                    :          10254     3937
  Current Connection
    Downstream       :          185668     71296
    Upstream         :          N/Avail    N/Avail

Errors
  Received FEC      :          0
  Received CRC      :          0
  Received HEC      :          0

=>
```

2 Bridge Commands

bridge (to access the Bridge level)
bridge config
bridge flush
bridge ifadd
bridge ifattach
bridge ifconfig
bridge ifdelete
bridge ifdetach
bridge iflist
bridge load
bridge macadd
bridge macdelete
bridge maclist
bridge save

bridge config

Show/set bridge aging policy.

SYNTAX:

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

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

EXAMPLE:

<pre>=>bridge config Aging : 300 =>bridge config age=600 =>bridge config Aging : 600 =></pre>

bridge flush

Flush complete bridging configuration.

The flush command does not impact previously saved configurations.

SYNTAX:

```
bridge flush
```

EXAMPLE:

```
=>bridge iflist
OBC      : Internal
          Connection State: connected   Port:OBC      PortState: forwarding
          RX bytes: 75783      frames: 572
          TX bytes: 82768372   frames: 341221 dropframes: 0
eth0     : Internal
          Connection State: connected   Port:eth0     PortState: forwarding
          RX bytes: 156344216 frames: 5899238
          TX bytes: 75689      frames: 425    dropframes: 5558017
Br1      : dest : Br1
          Retry: 10    QoS: default   Encaps: llc/snap Fcs: off
          Connection State: connected   Port:wano     PortState: forwarding
          RX bytes: 75      frames: 12
          TX bytes: 30246    frames: 91    dropframes: 0
Br2      : dest : Br2
          Retry: 10    QoS: default   Encaps: llc/snap Fcs: off
          Connection State: connected   Port:wano1    PortState: forwarding
          RX bytes: 167356345 frames: 7453312
          TX bytes: 64234246 frames: 2846491    dropframes: 0
=>bridge flush
=>bridge iflist
OBC      : Internal
          Connection State: connected   Port:OBC      PortState: forwarding
          RX bytes: 75783      frames: 572
          TX bytes: 82908667   frames: 341735 dropframes: 0
eth0     : Internal
          Connection State: connected   Port:eth0     PortState: forwarding
          RX bytes: 156553257 frames: 5904070
          TX bytes: 75689      frames: 425    dropframes: 5562335
=>
```

RELATED COMMANDS:

bridge load

Load saved or default bridge configuration.

bridge save

Save current bridge configuration.

bridge ifadd

Create a bridge interface.

SYNTAX:

bridge ifadd	intf = <string> dest = <phonebook entry>
---------------------	---

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

EXAMPLE:

```
=>bridge iflist
OBC      : Internal
          Connection State: connected      Port: OBC      PortState: forwarding
          RX bytes: 75783      frames: 572
          TX bytes: 82768372   frames: 341221   dropframes: 0
eth0     : Internal
          Connection State: connected      Port: eth0     PortState: forwarding
          RX bytes: 156344216  frames: 5899238
          TX bytes: 75689      frames: 425     dropframes: 5558017

=>phonebook list
Name     Type    Use  Address
Br1      bridge  0    8.35
Br2      bridge  0    8.36
CIPPVC3  cip     1    8.82
CIPPVC4  cip     1    8.83
=>bridge ifadd intf=NewBridge dest=Br1
=>bridge iflist
OBC      : Internal
          Connection State: connected      Port: OBC      PortState: forwarding
          RX bytes: 75783      frames: 572
          TX bytes: 82843610   frames: 341554   dropframes: 0
eth0     : Internal
          Connection State: connected      Port: eth0     PortState: forwarding
          RX bytes: 156472129  frames: 5903256
          TX bytes: 75689      frames: 425     dropframes: 5561702
NewBridge : dest : Br1
          Retry: 10    QoS: default    Encaps: llc/snap  Fcs: off
          Connection State: not-connectedPort: (Unassigned) PortState: forwarding
=>
```

RELATED COMMANDS:

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

bridge ifattach

Attach (i.e. connect) a bridge interface.

SYNTAX:

bridge ifattach	intf = <ifname>	
<i>intf</i>	The name of the bridge interface to attach.	REQUIRED

EXAMPLE:

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

RELATED COMMANDS:

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

bridge ifconfig

Configure a bridge interface.

SYNTAX:

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

<i>intf</i>	The name of the bridge interface to configure.	REQUIRED
[<i>dest</i>]	The destination for this interface. Typically a phonebook entry. This parameter needs only to be specified in case of an interface created without specified destination.	OPTIONAL
[<i>qos</i>]	The name of a configured Quality Of Service book entry. This parameter never needs to be specified.	OPTIONAL
[<i>encaps</i>]	The type of encapsulation to be used for this bridge interface. Choose between: <ul style="list-style-type: none"> ▪ llc/snap ▪ vcmux 	OPTIONAL
[<i>fcs</i>]	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 	OPTIONAL
[<i>portstate</i>]	The bridge portstate for this interface. Choose between: <ul style="list-style-type: none"> ▪ disabled ▪ learning ▪ forwarding 	OPTIONAL
[<i>retry</i>]	A number between 0 and 65535. Represents the number of WAN connection setup retries before giving up. By default the retry value is 10.	OPTIONAL

EXAMPLE:

```

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

```

RELATED COMMANDS:

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

bridge ifdelete

Delete a bridge interface.

```
bridge ifdelete      intf = <ifname>
```

intf The name of the interface name to delete. REQUIRED

EXAMPLE:

```

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

```

RELATED COMMANDS:

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

bridge ifdetach

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

SYNTAX:

bridge ifdetach	intf = <ifname>
------------------------	------------------------------

<i>intf</i>	The name of the bridge interface to detach.	REQUIRED
-------------	---	----------

EXAMPLE:

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

RELATED COMMANDS:

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

bridge iflist

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

SYNTAX:

bridge iflist	[intf = <ifname>]
----------------------	--------------------------------

[intf]	The name of the bridge interface to show the configuration of.	OPTIONAL
	If not specified all bridge interfaces are shown.	

EXAMPLE OUTPUT:

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

DESCRIPTION:

'RX bytes' indicates the number of Received bytes, 'TX bytes' the number of Transmitted bytes. OBC is short for On Board Controller and indicates the physical bridge port.

RELATED COMMANDS:

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

bridge load

Load saved (or default) bridge configuration.

Execute **bridge flush** prior to **bridge load**.

SYNTAX:

bridge load	[defaults = <yes no>]
--------------------	------------------------------------

[defaults]	Load factory defaults (yes) or saved configuration (no). Not specifying this parameter loads the saved configuration	OPTIONAL
------------	---	----------

EXAMPLE:

```
=>bridge iflist
OBC      : Internal
          Connection State: connected   Port:OBC      PortState: forwarding
          RX bytes: 75783      frames: 572
          TX bytes: 82768372   frames: 341221 dropframes: 0
eth0     : Internal
          Connection State: connected   Port:eth0     PortState: forwarding
          RX bytes: 156344216  frames: 5899238
          TX bytes: 75689      frames: 425    dropframes: 5558017
=>bridge ifadd intf=Br1 dest=Br1
=>bridge save
=>bridge flush
=>bridge iflist
OBC      : Internal
          Connection State: connected   Port:OBC      PortState: forwarding
          RX bytes: 75783      frames: 572
          TX bytes: 82908667   frames: 341735 dropframes: 0
eth0     : Internal
          Connection State: connected   Port:eth0     PortState: forwarding
          RX bytes: 156553257  frames: 5904070
          TX bytes: 75689      frames: 425    dropframes: 5562335
=>bridge load defaults=no
=>bridge iflist
OBC      : Internal
          Connection State: connected   Port:OBC      PortState: forwarding
          RX bytes: 75783      frames: 572
          TX bytes: 82768372   frames: 341221 dropframes: 0
eth0     : Internal
          Connection State: connected   Port:eth0     PortState: forwarding
          RX bytes: 156344216  frames: 5899238
          TX bytes: 75689      frames: 425    dropframes: 5558017
Br1      : dest : Br1
          Retry: 10    QoS: default    Encaps: llc/snap  Fcs: off
          Connection State: connected   Port:wano0    PortState: forwarding
          RX bytes: 75      frames: 12
          TX bytes: 30246    frames: 91     dropframes: 0
=>
```

RELATED COMMANDS:

bridge flush	Flush complete bridge configuration.
bridge save	Save current bridge configuration.

bridge macadd

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

SYNTAX:

bridge macadd	intf = <ifname>	
	hwaddr = <hardware-address>	

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

EXAMPLE:

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

RELATED COMMANDS:

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

bridge macdelete

Remove a MAC address from the filtering database.

SYNTAX:

bridge macdelete	hwaddr = <hardware-address>	
<i>hwaddr</i>	The MAC address of the entry to delete.	REQUIRED

EXAMPLE:

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

RELATED COMMANDS:

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

bridge maclist

Show current MAC address filtering database.

SYNTAX:

```
bridge maclist
```

EXAMPLE:

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

RELATED COMMANDS:

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

bridge save

Save current bridge configuration.

SYNTAX:

```
bridge save
```

EXAMPLE:

```

=>bridge iflist
OBC      : Internal
          Connection State: connected   Port:OBC      PortState: forwarding
          RX bytes: 75783      frames: 572
          TX bytes: 82768372   frames: 341221 dropframes: 0
eth0     : Internal
          Connection State: connected   Port:eth0     PortState: forwarding
          RX bytes: 156344216 frames: 5899238
          TX bytes: 75689      frames: 425     dropframes: 5558017
=>bridge ifadd intf=Br1 dest=Br1
=>bridge save
=>bridge flush
=>bridge iflist
OBC      : Internal
          Connection State: connected   Port:OBC      PortState: forwarding
          RX bytes: 75783      frames: 572
          TX bytes: 82908667   frames: 341735 dropframes: 0
eth0     : Internal
          Connection State: connected   Port:eth0     PortState: forwarding
          RX bytes: 156553257 frames: 5904070
          TX bytes: 75689      frames: 425     dropframes: 5562335
=>bridge load defaults=no
=>bridge iflist
OBC      : Internal
          Connection State: connected   Port:OBC      PortState: forwarding
          RX bytes: 75783      frames: 572
          TX bytes: 82768372   frames: 341221 dropframes: 0
eth0     : Internal
          Connection State: connected   Port:eth0     PortState: forwarding
          RX bytes: 156344216 frames: 5899238
          TX bytes: 75689      frames: 425     dropframes: 5558017
Br1      : dest : Br1
          Retry: 10   QoS: default   Encaps: llc/snap   Fcs: off
          Connection State: connected   Port:wan0     PortState: forwarding
          RX bytes: 75      frames: 12
          TX bytes: 30246   frames: 91      dropframes: 0
=>

```

RELATED COMMANDS:

bridge flush

Flush complete bridge configuration.

bridge load

Load saved or default bridge configuration.

3 CIP Commands

cip (to access the CIP level)
cip flush
cip ifadd
cip ifdelete
cip iflist
cip load
cip pvcadd
cip pvdelete
cip pvclist
cip save

cip flush

Flush complete CIP configuration.

The flush command does not impact previously saved configurations.

SYNTAX:

```
cip flush
```

EXAMPLE:

```
=>cip iflist
cip0      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 flush
=>cip iflist
=>
```

RELATED COMMANDS:

cip load

Load saved or default CIP configuration.

cip save

Save current CIP configuration.

cip ifadd

Create a CIP 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]>]
------------------	---

<i>addr</i>	The CIP interface's local IP address in the LIS.	REQUIRED
<i>netmask</i>	The LIS's subnetmask.	OPTIONAL
<i>uniaddr</i>	The UNI-address/port specification for incoming connections, e.g. 'A0:*.04': ADSL port, any address, selector 3. Only applicable in an SVC environment.	OPTIONAL

EXAMPLE:

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

RELATED COMMANDS:

cip ifdelete	Delete a CIP interface.
cip ifadd	Show current CIP configuration.

cip ifdelete

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

SYNTAX:

<i>cip ifdelete</i>	<i>addr = <ip-address></i>
----------------------------	---

<i>addr</i>	The CIP 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:

<i>cip ifadd</i>	Create a CIP interface.
<i>cip iflist</i>	Show current CIP configuration.

cip iflist

Show current CIP configuration.

SYNTAX:

```
cip iflist
```

EXAMPLE OUTPUT:

```

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

```

DESCRIPTION:

inarp_reqs_in/inarp_reqs_out : Incoming/outgoing inverse ARP requests

inarp_repl_in/inarp_repl_out : Incoming/outgoing inverse ARP replies

inarp_inv_in/inarp_inv_out : Incoming/outgoing invalid inverse ARP messages

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

```

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

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

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

```

RELATED COMMANDS:

cip ifadd Create a CIP interface.
cip ifdelete Delete a CIP interface.

cip load

Load saved (or default) CIP configuration.

Execute **cip flush** prior to **cip load**.

SYNTAX:

cip load	[defaults = <yes/no>]
-----------------	------------------------------------

<i>[defaults]</i>	Load factory defaults (yes) or saved configuration (no). Not specifying this parameter loads the saved configuration	OPTIONAL
-------------------	---	----------

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 save
=>cip flush
=>cip iflist
=>cip load defaults=yes
=>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

=>cip flush
=>cip iflist
=>cip load defaults=no
=>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

=>
```

RELATED COMMANDS:

cip flush	Flush complete CIP configuration.
cip save	Save current CIP configuration.

cip pvcadd

Create a PVC ARP entry for destinations which are not RFC 1577/RFC2225 compliant.

SYNTAX:

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

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

EXAMPLE:

```
=>phonebook list
Name      Type   Use  Address
Br1       bridge 1    8.35
Br2       bridge 1    8.36
Br3       bridge 1    8.37
Br4       bridge 0    8.38
RELAY_PPP1 ppp    0    8.48
RELAY_PPP2 ppp    0    8.49
RELAY_PPP3 ppp    0    8.50
RELAY_PPP4 ppp    0    8.51
PPP1      ppp    1    8.64
PPP2      ppp    1    8.65
PPP3      ppp    1    8.66
DHCP_SPOOF ppp    1    8.67
CIPPVC1   cip    0    8.80
CIPPVC2   cip    0    8.81
CIPPVC3   cip    0    8.82
CIPPVC4   cip    0    8.83
=>cip pvclist
=>cip pvcadd dest CIPPVC1 destaddr 172.16.1.2 mtu 546
=>cip pvclist
CIPPVC1_____atmport = 0      vpi = 8      vci = 80    dest_ip = 172.16.1.2
                encaps = llc    mtu = 546
=>
```

RELATED COMMANDS:

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

cip pvdelete

Delete a PVC ARP entry.

SYNTAX:

<i>cip pvdelete</i>	<i>dest = <phonebookname></i>
----------------------------	--

<i>dest</i>	Typically a phonebook entry name. Represents the ATM address (hardware address) or name of the entry to delete.	REQUIRED
-------------	--	----------

EXAMPLE:

<pre>=>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 =></pre>
--

RELATED COMMANDS:

<i>cip pvadd</i>	Create a PVC ARP entry.
<i>cip pvclist</i>	Show current PVC ARP entries.

cip pvclist

Show current PVC ARP entries.

SYNTAX:

```
cip pvclist
```

EXAMPLE OUTPUT:

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

EXAMPLE INPUT/OUTPUT IN A NETWORKED ENVIRONMENT:

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

RELATED COMMANDS:

cip pvdelete

Delete a PVC ARP entry.

cip pvadd

Create a PVC ARP entry.

cip save

Save current CIP configuration.

SYNTAX:

```
cip save
```

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 save
=>cip flush
=>cip iflist
=>cip load defaults=no
=>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

=>

```

RELATED COMMANDS:

cip flush

Flush complete CIP configuration.

cip load

Load saved or default CIP configuration.

4 Config Commands

config (to access the Config level)

config erase

config flush

config load

config reset

config save

config erase

Physically remove all saved configurations. Wireless settings are not affected.

SYNTAX:

```
config erase
```

EXAMPLE:

```
=>config erase
```

RELATED COMMANDS:

config flush
config load
config reset
config save

Flush complete runtime configuration.
Load complete saved or default configuration.
Flush current configuration and restore default configuration.
Save complete runtime configuration.

config flush

Flush complete current configuration without affecting saved configurations.

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

Wireless settings are not affected.

SYNTAX:

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

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

EXAMPLE:

```
=>ip rtlist
  Destination      Source           Gateway          Intf            Mtrc
  10.0.0.0/24      10.0.0.0/24     10.0.0.140      eth0            0
  172.16.0.5/32   0.0.0.0/0       172.16.0.5      cip1            0
  0.0.0.140/32    0.0.0.0/0       10.0.0.140      eth0            0
  127.0.0.1/32    0.0.0.0/0       127.0.0.1       loop            0
  10.0.0.0/24     0.0.0.0/0       10.0.0.140      eth0            0
  172.16.0.0/24   0.0.0.0/0       172.16.0.5      cip1            1
=>config flush flush_ip=no
=>ip rtlist
  Destination      Source           Gateway          Intf            Mtrc
  10.0.0.0/24      10.0.0.0/24     10.0.0.140      eth0            0
  10.0.0.140/32    0.0.0.0/0       10.0.0.140      eth0            0
  127.0.0.1/32    0.0.0.0/0       127.0.0.1       loop            0
  10.0.0.0/24     0.0.0.0/0       10.0.0.140      eth0            0
=>config flush flush_ip=yes

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

RELATED COMMANDS:

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

config load

Load complete saved or default configuration. Execute **config flush** prior to **config load**.

In case the saved configuration is loaded (defaults=no) this combines all load commands: **bridge load, cip load, dhcp client load, dhcp server load, dns load, firewall load, firewall rule load, mer load, nat load, phonebook load, ppp load, pptp load, qosbook load, system load** and optionally **ip load**.

Wireless settings are not affected.

SYNTAX:

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

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

EXAMPLE:

```

=>ip rtlist
  Destination      Source           Gateway          Intf    Mtrc
  10.0.0.0/24      10.0.0.0/24     10.0.0.140      eth0    0
  172.16.0.5/32   0.0.0.0/0       172.16.0.5      cip1    0
  0.0.0.140/32    0.0.0.0/0       10.0.0.140      eth0    0
  127.0.0.1/32    0.0.0.0/0       127.0.0.1       loop    0
  10.0.0.0/24     0.0.0.0/0       10.0.0.140      eth0    0
  172.16.0.0/24   0.0.0.0/0       172.16.0.5      cip1    1
=>config flush flush_ip=no
=>ip rtlist
  Destination      Source           Gateway          Intf    Mtrc
  10.0.0.0/24      10.0.0.0/24     10.0.0.140      eth0    0
  10.0.0.140/32    0.0.0.0/0       10.0.0.140      eth0    0
  127.0.0.1/32    0.0.0.0/0       127.0.0.1       loop    0
  10.0.0.0/24     0.0.0.0/0       10.0.0.140      eth0    0
=>config load load_ip=yes
=>ip rtlist
  Destination      Source           Gateway          Intf    Mtrc
  10.0.0.0/24      10.0.0.0/24     10.0.0.140      eth0    0
  10.0.0.140/32    0.0.0.0/0       10.0.0.140      eth0    0
  172.16.0.5/32   0.0.0.0/0       172.16.0.5      cip1    0
  127.0.0.1/32    0.0.0.0/0       127.0.0.1       loop    0
  10.0.0.0/24     0.0.0.0/0       10.0.0.140      eth0    0
  172.16.0.0/24   0.0.0.0/0       172.16.0.5      cip1    1
=>

```

RELATED COMMANDS:

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

config reset

Flush current runtime configuration, including the wireless settings and restore factory default configuration. Optionally the runtime, saved IP configuration can be preserved.

SYNTAX:

```
config reset [keep_ip = <{no|yes}>]
```

<code>[load_ip]</code>	Load IP settings (yes) or not (no). Not keeping the IP settings could cause lost IP connectivity in the LAN.	OPTIONAL
------------------------	---	----------

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 save
=>config reset
=>dns list
Domain: lan
Nr.      Hostname      IP Address
0        SpeedTouch   *.*.*.*
Total Table Size: 73 entries
Amount used: 1 (1%)
=>config flush
=>config load
=>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:

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

config save

Save all existing configurations and modifications entered by the user.

This combines all save commands: **atmf save, bridge save, cip save, config save, dhcp client save, dhcp server save, dns save, firewall chain save, firewall save, ip save, mer save, nat save, phonebook save, ppp save, pptp save, qosbook save, system save.**

SYNTAX:

```
config save
```

EXAMPLE:

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

RELATED COMMANDS:

config erase
config flush
config load
config reset

Physically remove all saved configurations.

Flush complete current configuration.

Load complete saved or default configuration.

Flush current configuration and restore default configuration.

5 DHCP Commands

dhcp (to access the DHCP level)
dhcp client (to access the DHCP Client level)
dhcp client clear
dhcp client config
dhcp client flush
dhcp client ifadd
dhcp client ifattach
dhcp client ifconfig
dhcp client ifdelete
dhcp client iflist
dhcp client ifrelease
dhcp client ifrenew
dhcp client load
dhcp client save
dhcp client stats
dhcp server (to access the DHCP Server level)
dhcp server add
dhcp server client
dhcp server clrstats
dhcp server config
dhcp server delete
dhcp server flush
dhcp server list
dhcp server load
dhcp server policy
dhcp server save
dhcp server spoof
dhcp server start

dhcp server stats
dhcp server status
dhcp server stop
dhcp server troff
dhcp server tron

dhcp client clear

Clear DHCP client statistics.

SYNTAX:

```
dhcp client clear
```

EXAMPLE:

```

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

```

RELATED COMMANDS:

dhcp client stats Show DHCP client statistics.

dhcp client config

Show/set DHCP client configuration.

SYNTAX:

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

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

EXAMPLE:

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

RELATED COMMANDS:

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

dhcp client flush

Flush complete DHCP client configuration and dynamic interfaces.
The flush command does not impact previously saved configurations.

SYNTAX:

```
dhcp client flush
```

EXAMPLE:

```
=>dhcp client iflist
NewMer      : [SELECTING]
              flags= uc
              IP address   : 10.0.0.10
              HW address   : 0:90:d0:01:47:de
              DHCP server  : 255.255.255.255
              hostname     : NewLease
              req.leasetime= 10800 s
              trying to get a lease for 8 min, 32 sec
              transmission of DISCOVER in 57 sec
              retransmissiontimeout: 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.
=>
```

RELATED COMMANDS:

dhcp client load

Load saved or default DHCP client configuration and dynamic interfaces.

dhcp client save

Save current DHCP client configuration and dynamic interfaces.

dhcp client ifadd

Create a DHCP lease for a specific interface.

SYNTAX:

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

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

EXAMPLE:

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

RELATED COMMANDS:

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

dhcp client ifattach

Attach a DHCP lease to a dynamic interface. Firstly create the interface with the **dhcp client ifadd** command.

SYNTAX:

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

<i>intf</i>	The name of the dynamic interface.	REQUIRED
-------------	------------------------------------	----------

EXAMPLE:

```
=>dhcp client iflist
NewMer      : [INIT]
              flags= uc
              IP address   : 0.0.0.0
              HW address   : 00:90:d0:01:47:de
              DHCP server  : 255.255.255.255
Number of leases: 1
Total size of table: 19,  in use: 1, free:94 %
=>dhcp client ifattach intf=NewMer
=>dhcp client iflist
NewMer      : [SELECTING]
              flags= uc
              IP address   : 10.0.0.10
              HW address   : 0:90:d0:01:47:de
              DHCP server  : 255.255.255.255
              hostname     : NewLease
              req.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 ifadd	Create a DHCP lease for a specific interface.
dhcp client ifconfig	Configure a DHCP lease created for a specific interface.
dhcp client ifrelease	Release a lease attached to a dynamic interface.

dhcp client ifconfig

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

Execute the **dhcp client ifrelease** command prior to configuring it.

SYNTAX:

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

<i>intf</i>	The name of the dynamic interface to be configured.	REQUIRED
<i>[clientid]</i>	The client identity to be associated with the lease.	OPTIONAL
<i>[hostname]</i>	The host name of the client to be associated with the lease.	OPTIONAL
<i>[addr]</i>	The preferred dynamic IP address.	OPTIONAL
<i>[leasetime]</i>	A number between 0 and 1814400 (seconds). Represents the preferred time the client wants to use an address. By default the leasetime is 7200 seconds (2 hours). Specifying -1 makes the lease permanent.	OPTIONAL
<i>[addrtrans]</i>	Automatically enable address translation for this dynamic interface (pat) or not (none).	OPTIONAL
<i>[dns]</i>	Request (and accept) DNS server IP addresses (on) or not (off).	OPTIONAL
<i>[gateway]</i>	Request (and accept) gateway IP addresses (on) or not (off).	OPTIONAL

EXAMPLE:

```

=>dhcp client iflist
NewMer      : [INIT]
             flags= uc
             IP address   : 0.0.0.0
             HW address   : 00:90:d0:01:47:de
             DHCP server  : 255.255.255.255
Number of leases: 1
Total size of table: 19,  in use: 1, free:94 %
=>dhcp client ifconfig intf=NewMer hostname=NewLease addr=10.0.0.10 leasetime=10800
=>dhcp client iflist
NewMer      : [INIT]
             flags= uc
             IP address   : 10.0.0.10
             HW address   : 00:90:d0:01:47:de
             DHCP server  : 255.255.255.255
             hostname    : NewLease
             req.leasetime= 10800 s
Number of leases: 1
Total size of table: 19,  in use: 1, free:94 %
=>

```

RELATED COMMANDS:

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

dhcp client ifdelete

Delete a dynamic interface.

SYNTAX:

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

intf The name of the dynamic interface. REQUIRED

EXAMPLE:

```
=>dhcp client iflist
NewMer      : [SELECTING]
              flags= uc
              IP address   : 10.0.0.10
              HW address   : 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
              retransmissiontimeout: 64
              nbr of retransmissions: 14
Number of leases: 1
Total size of table: 19,  in use: 1, free:94 %
=>dhcp client ifdelete intf NewMer
=>dhcp client iflist
No dynamic interfaces defined.
=>
```

RELATED COMMANDS:

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

dhcp client iflist

Show all dynamic interfaces.

SYNTAX:

```
dhcp client iflist
```

EXAMPLE:

```
=>dhcp client iflist
NewMer      : [INIT]
             flags= uc
             IP address   : 0.0.0.0
             HW address   : 0:90:d0:01:47:de
             DHCP server  : 255.255.255.255
Number of leases: 1
Total size of table: 19,  in use: 1, free:94 %
=>
```

EXAMPLE INPUT/OUTPUT IN A NETWORKED ENVIRONMENT:

The **SpeedTouch™ 570** is configured as DHCP client on its Ethernet interface eth0.

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

RELATED COMMANDS:

dhcp client ifadd

Create a DHCP lease for a specific interface.

dhcp client ifdelete

Delete a dynamic interface.

dhcp client ifrelease

Release a lease attached to a dynamic interface.

SYNTAX:

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

<i>intf</i>	The name of the dynamic interface.	REQUIRED
-------------	------------------------------------	----------

EXAMPLE:

```
=>dhcp client iflist
NewMer      : [SELECTING]
              flags= uc
              IP address   : 10.0.0.10
              HW address   : 0:90:d0:01:47:de
              DHCP server  : 255.255.255.255
              hostname     : NewLease
              req.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 ifattach intf=NewMer
=>dhcp client iflist
NewMer      : [INIT]
              flags= uc
              IP address   : 0.0.0.0
              HW address   : 00:90:d0:01:47:de
              DHCP server  : 255.255.255.255
              hostname     : NewLease
              req.lease time= 10800 s
Number of leases: 1
Total size of table: 19, in use: 1, free:94 %
=>
```

EXAMPLE INPUT/OUTPUT IN A NETWORKED ENVIRONMENT:

The **SpeedTouch™ 570** 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, 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: 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: 18, in use: 1, free: 94 %
=>
```

RELATED COMMANDS:

dhcp client ifattach
dhcp client ifconfig
dhcp client ifdelete

Attach a DHCP lease to an interface.
Configure a DHCP lease created for a specific interface.
Delete a dynamic interface.

dhcp client ifrenew

Renew the lease of a dynamic interface.

SYNTAX:

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

intf The name of the dynamic interface. REQUIRED

EXAMPLE:

```

=>dhcp client iflist
NewMer      : [BOUND]
             flags = uc
             IP address   : 10.0.0.10
             HW address   : 00:90:d0:01:47:de
             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=NewMer
=>dhcp client iflist
NewMer      : [RENEWING]
             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 INPUT/OUTPUT IN A NETWORKED ENVIRONMENT:

The **SpeedTouch™ 570** 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: 19,  in use: 1, free: 94 %
=>(CTRL + Q)
.....
STATE IDLE !
STATE ACTIVATE !
dhcc: intf 1 renews lease 10.0.0.3.
dhcc: intf 1 requests 10.0.0.3 from 10.10.1.1
dhcc: 10.10.1.1 acks 10.0.0.3 to intf 1.
dhcc: lease 10.0.0.3 bound to intf 1.
STATE IDLE !
STATE ACTIVATE !
.....
=>(CTRL + S)

```

RELATED COMMANDS:

dhcp client ifadd

Create a DHCP lease for a specific interface.

dhcp client ifattach

Attach a DHCP lease to an interface.

dhcp client load

Load saved DHCP client configuration and dynamic interfaces.
Execute **dhcp client flush** prior to **dhcp client load**.

SYNTAX:

```
dhcp client load
```

EXAMPLE:

```
=>dhcp client iflist
NewMer      : [SELECTING]
              flags = uc
              IP address   : 10.0.0.10
              HW address   : 00:90:d0:01:47:de
              DHCP server  : 255.255.255.255
              hostname     : NewLease
              req.lease time= 10800s
              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 save
=>dhcp client flush
=>dhcp client iflist
No dynamic interfaces defined.
=>dhcp client load
=>dhcp client iflist
NewMer      : [REBOOTING]
              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= 10800s
              trying to get a lease for 2 sec
              transmission of REQUEST in 2 sec
              retransmission timeout: 4
              retransmissions left before reinitializing : 2
Number of leases: 1
Total size of table: 19, in use: 1, free: 94 %
=>
```

RELATED COMMANDS:

dhcp client flush

Delete all dynamic interfaces.

dhcp client save

Save current DHCP client configuration and dynamic interfaces.

dhcp client save

Save current DHCP client configuration and dynamic interfaces.

SYNTAX:

```
dhcp client save
```

EXAMPLE:

```
=>dhcp client iflist
NewMer      : [SELECTING]
              flags = uc
              IP address   : 10.0.0.10
              HW address   : 00:90:d0:01:47:de
              DHCP server  : 255.255.255.255
              hostname     : NewLease
              req.lease time= 10800s
              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 save
=>dhcp client flush
=>dhcp client iflist
No dynamic interfaces defined.
=>dhcp client load
=>dhcp client iflist
NewMer      : [REBOOTING]
              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= 10800s
              trying to get a lease for 2 sec
              transmission of REQUEST in 2 sec
              retransmission timeout: 4
              retransmissions left before reinitializing : 2
Number of leases: 1
Total size of table: 19, in use: 1, free: 94 %
=>
```

RELATED COMMANDS:

dhcp client flush
dhcp client load

Flush complete DHCP client configuration and dynamic interfaces.
Load saved or default DHCP client configuration and dynamic interfaces.

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         :           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 clear Clear DHCP client statistics.

dhcp server add

Assign a static IP address to a host in the local network. This address is allocated on a permanent basis, and is excluded from the pool of addresses used by the **SpeedTouch™ 570** DHCP server.

SYNTAX:

```
dhcp server add      clientid = <client-id>
                    addr = <ip-address>
                    [leasetime = <number>]
                    [hostname = <hostname>]
```

<i>clientid</i>	The DHCP client's MAC address.	REQUIRED
<i>addr</i>	The IP address for this DHCP host.	REQUIRED
<i>[leasetime]</i>	A number between 0 and 1814400 (seconds). Represents the preferred time the client wants to use an address. By default the leasetime is 7200 seconds (2 hours). Specifying -1 makes the lease permanent.	OPTIONAL
<i>[hostname]</i>	The hostname to add to the local DNS table for this host.	OPTIONAL

EXAMPLE:

```
=>dhcp server list
Leases:
Lease  0:    01:00:A0:24:AE:66:E1
        Hostname  = Default
        ip address : 10.0.0.8
        expires in : 1 h, 16 min, 20 sec
        lease is  being used.
Total size of table: 36, in use: 1 free: 97 %
=>dhcp server add clientid=01:23:45:67:89:ab addr=10.0.0.1 leasetime=60 hostname=NewLease
=>dhcp server list
Leases:
Lease  0:    01:00:A0:24:AE:66:E1
        Hostname  = Default
        ip address : 10.0.0.8
        expires in : 1 h, 15 min, 32 sec
        lease is  being used.
Lease  1:    01:23:45:67:89:AB
        Hostname  = NewLease
        ip address : 10.0.0.1
        expires in : 23 sec
        lease is  being used.
Total size of table: 36, in use: 2 free: 94 %
=>
```

RELATED COMMANDS:

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

dhcp server client

Set the AutoDHCP client time-out in startup phase. Only applicable in AutoDHCP mode (See **dhcp server policy** command).

SYNTAX:

dhcp server client	timeout = <number>
---------------------------	---------------------------------

<i>timeout</i>	A number between 0 and 1814400 (seconds). Represents the time to look for another DHCP server. Specifying '-1' will make the timeout infinite: the SpeedTouch™ 570 will remain client. By default the timeout is 20 seconds.	REQUIRED
----------------	--	----------

EXAMPLE:

```
=>dhcp server status
DHCP Server Status:      Running
Current configuration:
    Address Range:  10.0.0.1 ... 10.255.255.254
.....
Start-up client parameters:
    Timeout:  20 sec
Tracing: off
Memory usage:
    Leases: total: 36, in use: 7 free: 80 %
=>dhcp server client timeout=15
=>dhcp server status
DHCP Server Status:      Running
Current configuration:
    Address Range:  10.0.0.1 ... 10.255.255.254
.....
Start-up client parameters:
    Timeout:  15 sec
Tracing: off
Memory usage:
    Leases: total: 36, in use: 7 free: 80 %
=>
```

RELATED COMMANDS:

dhcp server policy	Set DHCP server policy.
dhcp server start	Start DHCP server.
dhcp server status	Show current DHCP server configuration.
dhcp server stop	Stop DHCP server.

dhcp server clrstats

Clear **SpeedTouch™ 570** DHCP server statistics.

SYNTAX:

```
dhcp server clrstats
```

EXAMPLE:

```
=>dhcp server stats
DHCP server statistics:
Corrupted packet recv :      0
DISCOVER                :     9575
REQUEST                 :     121
DECLINE                 :      0
RELEASE                 :      0
INFORM                  :     13
Pure BOOTP REQUESTS    :      2
Other message types    :      0
OFFERs sent           :     9552
ACKs sent            :     121
NAKs sent               :      0
Lease table got full   : no
Ping table got full    : no
Second DHCP server seen : no
=>dhcp server clrstats
=>dhcp server stats
DHCP server statistics:
Corrupted packet recv :      0
DISCOVER                :      0
REQUEST                 :      0
DECLINE                 :      0
RELEASE                 :      0
INFORM                  :      0
Pure BOOTP REQUESTS    :      0
Other message types    :      0
OFFERs sent           :      0
ACKs sent            :      0
NAKs sent               :      0
Lease table got full   : no
Ping table got full    : no
Second DHCP server seen : no
=>
```

RELATED COMMANDS:

dhcp server stats Show DHCP server statistics.

dhcp server config

Set **SpeedTouch™ 570** DHCP server configuration.

Execute **dhcp server status** to see the actual status and configuration.

SYNTAX:

dhcp server config	<pre>[beginrange = <ip-address>] [endrange = <ip-address>] [netmask = <ip-address>] [leasetime = <number>] [gateway = <ip-address 0>] [dnsaddr = <ip-address>]</pre>
---------------------------	--

<i>beginrange</i>	The lowest IP address in the DHCP address range to use for leasing. Default value of this parameter is 10.0.0.1.	OPTIONAL
<i>endrange</i>	The highest IP address in the DHCP address range to use for leasing. Default value of this parameter is 10.255.255.254.	OPTIONAL
<i>netmask</i>	The applicable netmask for the DHCP leases.	OPTIONAL
<i>leasetime</i>	A number between 0 and 1814400 (seconds). Represents the time for which a client can use its dynamically allocated IP address. By default the leasetime is 2 hours (7200 seconds). Specifying -1 makes the lease permanent.	OPTIONAL
<i>gateway</i>	The IP address of the gateway for DHCP clients.	OPTIONAL
<i>dnsaddr</i>	The IP address of the DNS server for DHCP clients. Entering '0' sets the SpeedTouch™ 570 as DNS server.	OPTIONAL

EXAMPLE:

```

=>dhcp server status
DHCP Server Status:      Running
Current configuration:
    Address Range:  10.0.0.1 ... 10.255.255.254
    Netmask:255.0.0.0
    Lease time:    10800 seconds
    Gateway (default router):  10.0.0.138
    DNS server:    10.0.0.1
    Domain name:   lan
Policies:
    Verify first: no
    Trust client: yes
    Spoofing: no
    Start as client: yes
Spoofing parameters:
    Failure timeout (!DoD): 4 sec
    Failure lease time (!DoD): 60 sec
    Temp. lease time (DoD): 10 sec
Start-up client parameters:
    Timeout: 15 sec
Tracing:off
Memory usage:
    Leases: total: 36, in use: 7 free: 80 %
=>dhcp server config beginrange=172.16.0.2endrange=172.16.0.122netmask=255.0.0.0
    leasetime=21600 gateway=172.16.0.1 dnsaddr=172.16.0.254
=>dhcp server status
DHCP Server Status:      Running
Current configuration:
    Address Range:  172.16.0.2 ... 172.16.0.122
    Netmask:255.0.0.0
    Lease time:    21600 seconds
    Gateway (default router):  172.16.0.1
    DNS server:    172.16.0.254
    Domain name:   lan
Policies:
    Verify first: no
    Trust client: yes
    Spoofing: no
    Start as client: yes
Spoofing parameters:
    Failure timeout (!DoD): 4 sec
    Failure lease time (!DoD): 60 sec
    Temp. lease time (DoD): 10 sec
Start-up client parameters:
    Timeout: 15 sec
Tracing:off
Memory usage:
    Leases: total: 36, in use: 7 free: 80 %
=>

```

RELATED COMMANDS:

dhcp server status

Show current DHCP server configuration.

dhcp server delete

Delete a DHCP lease.

SYNTAX:

```
dhcp server delete index = <number>
```

<i>index</i>	The index number of the entry to be deleted. Execute dhcp server list to see a list of the index numbers of all current DHCP leases.	REQUIRED
--------------	--	----------

EXAMPLE:

```
=>dhcp server list
Leases:
Lease 0: 01:00:A0:24:AE:66:E1
    Hostname = Default
    ip address : 10.0.0.8
    expires in : 1 h, 16 min, 20 sec
    lease is being used.
Lease 1: 01:23:45:67:89:AB
    Hostname = NewLease
    ip address : 10.0.0.1
    expires in : 23 sec
    lease is being used.
Total size of table: 36, in use: 2 free: 94 %
=>dhcp server delete index=1
=>dhcp server list
Leases:
Lease 0: 01:00:A0:24:AE:66:E1
    Hostname = Default
    ip address : 10.0.0.8
    expires in : 1 h, 15 min, 32 sec
    lease is being used.
Total size of table: 36, in use: 1 free: 97 %
=>
```

RELATED COMMANDS:

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

dhcp server flush

Flush complete DHCP server configuration and dynamic leases.
The flush command does not impact previously saved configurations.

SYNTAX:

```
dhcp server flush
```

EXAMPLE:

```
=>dhcp server list
Leases:
Lease 2: 01:52:41:53:20:A0:1B:A7:EB:AD:3C:C0:01:01:00:00:00
  ip address: 10.0.7.79
  expires in: 11 sec
  lease is not being used.
Lease 1: 01:52:41:53:20:20:4D:0D:CB:03:40:C0:01:01:00:00:00
  ip address: 10.0.7.62
  Spoofed lease from 2: DHCP_SPOOF
  Assigned (temporary) private ip address.
  expires in: 1 min, 39 sec
  lease is not being used.
Lease 0: 01:00:A0:24:AE:66:E1
  Hostname = Default
  ip address : 10.0.0.8
  expires in : 1 h, 16 min, 20 sec
  lease is being used.
Lease 3: 01:23:55:67:89:AB
  Hostname = Tempo
  ip address : 10.0.0.1
  never expires!
  lease is not being used.
Total size of table: 36, in use: 4 free: 89 %
=>dhcp server flush
=>dhcp server list
No active leases
Total size of table: 36, in use: 0 free: 100 %
=>
```

RELATED COMMANDS:

- | | |
|-------------------------|---|
| dhcp server load | Load saved or default DHCP server configuration and permanent leases. |
| dhcp server save | Save current DHCP server configuration and permanent leases. |

dhcp server list

List current DHCP leases, indicated by their index number.

SYNTAX:

```
dhcp server list
```

EXAMPLE OUTPUT:

```

=>dhcp server list
Leases:
Lease 2: 01:52:41:53:20:50:6D:C0:40:02:32:C0:01:01:00:00:00
ip address: 10.0.7.142
expires in: 3 sec
lease is not being used.
Lease 3: 01:52:41:53:20:A0:1B:A7:EB:AD:3C:C0:01:01:00:00:00
ip address: 10.0.7.143
expires in: 17 sec
lease is not being used.
Lease 5: 01:52:41:53:20:F0:90:8F:09:E1:35:BE:01:01:00:00:00
ip address: 10.0.7.144
expires in: 55 sec
lease is not being used.
Lease 6: 01:52:41:53:20:30:F4:89:5F:9B:44:C0:01:01:00:00:00
ip address: 10.0.7.145
expires in: 1 min, 6 sec
lease is not being used.
Lease 1: 01:52:41:53:20:20:4D:0D:CB:03:40:C0:01:01:00:00:00
ip address: 10.0.7.62
Spoofed lease from 2: DHCP_SPOOF
Assigned (temporary) private ip address.
expires in: 1 min, 57 sec
lease is not being used.
Lease 0: 01:00:A0:24:AE:66:E1
Hostname= Default
ip address: 10.0.0.8
expires in: 1 h, 17 min, 21 sec
lease is being used.
Lease 4: 01:23:55:67:89:AB
Hostname = Tempo
ip address : 10.0.0.1
never expires!
lease is not being used.
Total size of table: 36, in use: 7 free: 80 %
=>

```

RELATED COMMANDS:

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

dhcp server load

Load saved (or default) DHCP server configuration and permanent leases.

SYNTAX:

```
dhcp server load [defaults = <yes|no>]
```

<i>[defaults]</i>	Load factory defaults (yes) or saved configuration (no). Not specifying this parameter loads the saved configuration.	OPTIONAL
-------------------	--	----------

EXAMPLE:

```
=>dhcp server list
Leases:
Lease 1: 01:52:41:53:20:20:4D:0D:CB:03:40:C0:01:01:00:00:00
    ip address: 10.0.7.62
    Spoofed lease from 2: DHCP_SPOOF
    Assigned (temporary) private ip address.
    expires in: 1 min, 57 sec
    lease is being used.
Lease 0: 01:00:A0:24:AE:66:E1
    Hostname= Default
    ip address: 10.0.0.8
    expires in: 1 h, 17 min, 21 sec
    lease is being used.
Lease 4: 01:23:55:67:89:AB
    Hostname = Tempo
    ip address : 10.0.0.1
    never expires!
    lease is being used.
Total size of table: 36, in use: 3 free: 92 %
=>dhcp server save
=>dhcp server flush
=>dhcp server load
=>dhcp server list
Leases:
Lease 4: 01:23:55:67:89:AB
    Hostname = Tempo
    ip address : 10.0.0.1
    never expires!
    lease is not being used.
Total size of table: 36, in use: 1 free: 97 %
=>
```

RELATED COMMANDS:

dhcp server flush

Flush current DHCP server configuration and dynamic leases.

dhcp server load

Load saved or default DHCP server configuration and permanent leases.

dhcp server policy

Set **SpeedTouch™ 570** DHCP server policy.

SYNTAX:

dhcp server policy	[verifyfirst = <yes no>] [trustclient = <yes no>] [spoofing = <yes no>] [client = <yes no>]
---------------------------	--

<i>[verifyfirst]</i>	Probe the network for conflicting IP addresses before giving a suggested IP address to the requesting DHCP client (yes) or not (no).	OPTIONAL
<i>[trustclient]</i>	Take the IP address suggested by a DHCP client into account (yes) or not (no).	OPTIONAL
<i>[spoofing]</i>	Allow a remote DHCP server to hand out IP addresses negotiated by PPP on WAN side (yes) or not (no). DHCP spoofing is used to relay local DHCP requests to an external PPP connection having a specific IP address negotiation mechanism. DHCP replies are in turn generated by the DHCP server based on the IP address information received by the PPP link.	OPTIONAL
<i>[client]</i>	Allow the SpeedTouch™ 570 DHCP server 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 the DHCP server (yes) or immediately start the DHCP server (no).	OPTIONAL

EXAMPLE:

```
=>dhcp server status
DHCP Server Status:      Running
Current configuration:
.....
Policies:
    Verify first: no
    Trust client: yes
    Spoofing: no
    Start as client: yes
.....
=>dhcp server policy verifyfirst=yes trustclient=no spoofing=yes client=no
=>dhcp server status
DHCP Server Status:      Running
Current configuration:
.....
Policies:
    Verify first: yes
    Trust client: no
    Spoofing: yes
    Start as client: no
.....
=>
```

RELATED COMMANDS:

dhcp server status Show current DHCP server configuration.

dhcp server save

Save complete **SpeedTouch™ 570** DHCP server configuration and permanent DHCP leases.

SYNTAX:

```
dhcp server save
```

EXAMPLE:

```
=>dhcp server list
Leases:
Lease 0: 01:00:A0:24:AE:66:E1
    Hostname= Default
    ip address: 10.0.0.8
    expires in: 1 h, 57 min, 9 sec
    lease is being used.
Lease 1: 01:23:55:67:89:AB
    Hostname= Tempo
    ip address: 10.0.0.1
    never expires!
    lease is not being used.
Total size of table: 36, in use: 2 free: 94 %
=>dhcp server save
=>dhcp server flush
=>dhcp server list
No active leases
Total size of table: 36, in use: 0 free: 100 %
=>dhcp server load defaults=no
=>dhcp server list
Leases:
Lease 0: 01:00:A0:24:AE:66:E1
    Hostname= Default
    ip address: 10.0.0.8
    expires in: 1 h, 58 min, 55 sec
    lease is being used.
Lease 1: 01:23:55:67:89:AB
    Hostname= Tempo
    ip address: 10.0.0.1
    never expires!
    lease is not being used.
Total size of table: 36, in use: 2 free: 94 %
=>
```

RELATED COMMANDS:

dhcp server flush
dhcp server load

Flush complete DHCP server configuration and dynamic leases
Load saved or default DHCP server configuration and permanent leases.

dhcp server spoof

Set DHCP spoofing parameters. Only applicable in case of a PPP-to-DHCP Spoofing connection. (See **dhcp server policy** command).

SYNTAX:

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

<i>[failtime]</i>	<p>A number between 0 and 1814400 (seconds). Represents the time to wait for a PPP link to successfully negotiate an IP address. This parameter determines how long the SpeedTouch™ 570 should try to set up a PPP connection before returning to normal DHCP mode, i.e. in case the PPP connection cannot be established within the time lapse determined by failtime, the SpeedTouch™ 570 DHCP server will allocate an local private IP address to the DHCP client. By default the failtime is 4 seconds.</p>	OPTIONAL
<i>[errorlt]</i>	<p>A number between 0 and 1814400 (seconds). Represents the leasetime of the private address issued when a PPP link fails. In case the PPP link fails after failtime has elapsed, this parameter determines how long the private DHCP lease must be maintained before retrying to set up the PPP link again. By default the error lease time is 60 seconds.</p>	OPTIONAL
<i>[dodlt]</i>	<p>A number between 0 and 1814400 (seconds). Represents the leasetime of the temporary private IP address in case of a dial-on-demand PPP link. In case of a dial-on-demand PPP link, this parameter determines the interval at which the the temporary DHCP lease must be maintained before checking whether a public IP address negotiated by a triggered PPP link is available. By default the dial-on-demand lease time is 10 seconds.</p>	OPTIONAL

EXAMPLE:

```
=>dhcp server status
DHCP Server Status: Running
Current configuration:
.....
Spoofing parameters:
    Failure timeout (!DoD): 4 sec
    Failure lease time (!DoD): 60 sec
    Temp. lease time (DoD): 10 sec
.....
=>dhcp server spoof failtime=8 errorlt=120 dodlt=20
=>dhcp server status
DHCP Server Status: Running
Current configuration:
.....
Spoofing parameters:
    Failure timeout (!DoD): 8 sec
    Failure lease time (!DoD): 120 sec
    Temp. lease time (DoD): 20 sec
.....
=>
```

RELATED COMMANDS:

dhcp server policy
dhcp server status

Set DHCP server policy.
Show current DHCP server configuration.

dhcp server start

Start **SpeedTouch™ 570** DHCP server.

SYNTAX:

```
dhcp server start
```

EXAMPLE:

```
=>dhcp server status
DHCP Server Status: Stopped
Current configuration:
.....
=>dhcp server start
=>dhcp server status
DHCP Server Status: Searching for server...
Current configuration:
.....
=>
=>dhcp server status
DHCP Server Status: Running
Current configuration:
.....
=>
```

RELATED COMMANDS:

dhcp server status

Show current DHCP server configuration.

dhcp server stop

Stop DHCP server.

dhcp server stats

Show **SpeedTouch™ 570** DHCP server statistics.

SYNTAX:

```
dhcp server stats
```

EXAMPLE OUTPUT:

```
=>dhcp server stats
DHCP server statistics:
Corrupted packet recv  :          0
DISCOVER                :        9575
REQUEST                 :        121
DECLINE                 :          0
RELEASE                 :          0
INFORM                  :         13
Pure BOOTP REQUESTS    :          2
Other message types     :          0
OFFERs sent             :        9552
ACKs sent                :        121
NAKs sent                :          0
Lease table got full    : no
Ping table got full     : no
Second DHCP server seen : no
=>
```

DESCRIPTION:

<i>Corrupted packet recv</i>	Indicates the number of corrupted packets (not complaint to RFC2131) were received from the LAN.
<i>DISCOVER</i>	Indicates the number of DHCP server discovery packets were received from the LAN. These broadcasts are sent by potential DHCP clients to locate available DHCP servers.
<i>REQUEST</i>	Indicates the number of DHCP address lease requests were received from the LAN.
<i>DECLINE</i>	Indicates the number of DHCP address lease requests that were declined.
<i>RELEASE</i>	Indicates the number of DHCP address release requests that were received from DHCP clients.
<i>INFORM</i>	Indicates the number of information requests that were received from DHCP clients.
<i>Pure BOOTP requests</i>	Indicates the number of BOOTP requests that were received from the LAN.
<i>OFFERs sent</i>	Indicates the number of IP address offers were sent in reply to DHCP requests.

<i>ACKs sent</i>	Indicates the number of ACKnowledgement replies were sent to successfully configured DHCP clients.
<i>NAKs sent</i>	Indicates the number of Not-AcKnowledge replies were sent to wrongly configured DHCP clients.
<i>Lease table got full</i>	Indicates whether the maximum number of DHCP leases is reached or not.
<i>Ping table got full</i>	Indicates whether the history list of IP address pings got full or not. These pings are sent by the SpeedTouch™ 570 DHCP server to verify whether the IP address is already in use on the LAN or not. (dhcp server policy verifyfirst=yes)
<i>Second DHCP server seen</i>	Indicates whether a concurrent DHCP server was found on the LAN or not.

RELATED COMMANDS:

dhcp server clrstats Clear DHCP server statistics.

dhcp server status

Show current DHCP server configuration.

SYNTAX:

```
dhcp server status
```

EXAMPLE:

```
=>dhcp server status
DHCP Server Status:      Client
Current configuration:
    Address Range:  10.0.0.1 ... 10.255.255.254
    Netmask:255.0.0.0
    Lease time:     7200 seconds
    Gateway (default router): 10.0.0.1 (auto)
    DNS server:    10.0.0.1 (auto)
    Domain name:   office.lan
Policies:
    Verify first: no
    Trust client: yes
    Spoofing:     no
    Start as client: yes
Spoofing parameters:
    Failure timeout (!DoD): 4 sec
    Failure lease time (!DoD): 60 sec
    Temp. lease time (DoD): 10 sec
Start-up client parameters:
    Timeout: 20 sec
Tracing:off
Memory usage:
    Leases: total: 36, in use: 7 free: 80 %
=>
```

RELATED COMMANDS:

dhcp server stop	Stop DHCP server.
dhcp server start	Start DHCP server.
dhcp server policy	Set DHCP server policy.
dhcp server spoofing	Set spoofing parameters.

dhcp server stop

Stop SpeedTouch™ 570 DHCP server.

SYNTAX:

```
dhcp server stop
```

EXAMPLE:

```
=>dhcp server status
DHCP Server Status: Running
Current configuration:
.....
=>dhcp server stop
=>dhcp server status
DHCP Server Status: Stopped
Current configuration:
.....
=>
```

RELATED COMMANDS:

dhcp server start

Start DHCP server.

dhcp server status

Show current DHCP server configuration.

dhcp server troff

Disable verbose console logging. No debug traces are generated anymore.

SYNTAX:

```
dhcp server troff
```

EXAMPLE:

```
=>dhcp server status
DHCP Server Status:      Running
Current configuration:
.....
Tracing:on
.....
=>dhcp server troff
=>dhcp server status
DHCP Server Status:      Running
Current configuration:
.....
Tracing:off
.....
=>
```

RELATED COMMANDS:

dhcp server status

Show current DHCP server configuration.

dhcp server tron

Enable verbose console logging.

dhcp server tron

Enable verbose console logging. Debug traces are generated.

SYNTAX:

```
dhcp server tron
```

EXAMPLE:

```
=>dhcp server status
DHCP Server Status:      Running
Current configuration:
.....
Tracing:off
.....
=>dhcp server tron
=>dhcp server status
DHCP Server Status:      Running
Current configuration:
.....
Tracing:on
.....
=>
```

RELATED COMMANDS:

dhcp server status	Show current DHCP server configuration
dhcp server troff	Disable verbose console logging.

6 DNS Commands

dns (to access the DNS level)

dns add

dns clear

dns clrstats

dns delete

dns domain

dns flush

dns fwdadd

dns fwddelete

dns fwdlist

dns fwdtable

dns list

dns load

dns nslookup

dns save

dns start

dns stats

dns status

dns stop

dns toutfwd

dns troff

dns tron

dns add

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

SYNTAX:

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

<i>hostname</i>	The name of the IP host (without the (sub)domain name).	REQUIRED
<i>[addr]</i>	The IP address of the host (without mask). In case this parameter is not specified the hostname applies to the SpeedTouch™ 570 itself.	OPTIONAL

EXAMPLE:

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

RELATED COMMANDS:

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

dns clear

Delete current DNS entries.

SYNTAX:

```
dns clear
```

EXAMPLE:

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

RELATED COMMANDS:

dns list

List current DNS entries.

dns clrstats

Clear DNS statistics.

SYNTAX:

```
dns clrstats
```

EXAMPLE:

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

Total number of packets received :           4

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

Total number of packets received :           0

=>
```

RELATED COMMANDS:

dns stats

Show DNS server/forwarder statistics.

dns delete

Delete a DNS entry.

SYNTAX:

dns delete	index = <number>
-------------------	-------------------------------

<i>index</i>	The index number of the entry to be deleted. Execute dns list to see a list of the index numbers of all current DNS entries.	REQUIRED
--------------	--	----------

EXAMPLE:

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

RELATED COMMANDS:

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

dns domain

Set local DNS (sub)domain name.

SYNTAX:

dns domain	domain = <string>
-------------------	--------------------------------

<i>domain</i>	The local DNS (sub)domain name.	REQUIRED
---------------	---------------------------------	----------

EXAMPLE:

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

RELATED COMMANDS:

dns list	List current DNS entries.
-----------------	---------------------------

dns flush

Flush complete **SpeedTouch™ 570** 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%)
=>
```

RELATED COMMANDS:

dns save

Save current DNS server/forwarder configuration and static entries.

dns load

Load saved or default DNS server/forwarder configuration and static entries.

dns fwdadd

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

SYNTAX:

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

<i>dns</i>	The IP address of the (remote) DNS server.	REQUIRED
<i>src</i>	The source IP address (pool) of the host(s) using this DNS server.	REQUIRED
<i>mask</i>	The appropriate source IP (sub)netmask.	REQUIRED
<i>[direct]</i>	Determines whether DNS replies are sent directly back to the client (1) or relayed by the SpeedTouch™ 570 DHCP server's DNS forwarder (0) in case of PPP-to-DHCP spoofing connections.	OPTIONAL

EXAMPLE:

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

RELATED COMMANDS:

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

dns fwdddelete

Delete a DNS forwarding entry.

SYNTAX:

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

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

EXAMPLE:

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

RELATED COMMANDS:

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

dns fwdlist

Show current DNS forwarding entries.

SYNTAX:

```
dns fwdlist
```

EXAMPLE OUTPUT:

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

RELATED COMMANDS:

<i>dns fwdadd</i>	Add a DNS forwarding entry.
<i>dns fwddelete</i>	Delete a DNS forwarding entry.
<i>dns fwdtable</i>	Show DNS forwarding table.

dns fwdtable

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

SYNTAX:

```
dns fwdtable
```

EXAMPLE OUTPUT:

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

RELATED COMMANDS:

dns fwdlist

Show current DNS forwarding entries.

dns list

Show current DNS entries.

SYNTAX:

```
dns list
```

EXAMPLE OUTPUT:

```

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

```

EXAMPLE INPUT/OUTPUT IN A NETWORKED ENVIRONMENT:

The **SpeedTouch™ 570** is configured as DNS server.

```

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

```

RELATED COMMANDS:

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

dns load

Load saved or default **SpeedTouch™ 570** DNS server/forwarder configuration and static DNS entries.

Execute **dns flush** prior to **dns load**.

SYNTAX:

dns load	[defaults = <yes no>]
-----------------	------------------------------------

<i>[defaults]</i>	Load factory defaults (yes) or saved configuration (no). Not specifying this parameter loads the saved configuration	OPTIONAL
-------------------	---	----------

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
Total Table Size: 73 entries
Amount used: 4 (5%)
=>dns save
=>dns flush
=>dns list
Domain: office.home.lan
Nr.      Hostname      IP Address

Total Table Size: 73 entries
Amount used: 0 (0%)
=>dns load defaults=yes
=>dns list
Domain: lan
Nr.      Hostname      IP Address
0        SpeedTouch    *.*.*.*
Total Table Size: 73 entries
Amount used: 1 (1%)
=>dns load defaults=no
=>dns flush
=>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
Total Table Size: 73 entries
Amount used: 4 (5%)
=>
```

RELATED COMMANDS:

dns flush	Flush complete DNS server/forwarder configuration and static entries.
dns save	Save current DNS server/forwarder configuration and static entries

dns nslookup

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

SYNTAX:

dns nslookup	lookup = <string>
---------------------	--------------------------------

<i>lookup</i>	The DNS hostname or IP address to query.	REQUIRED
---------------	--	----------

EXAMPLE:

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

RELATED COMMANDS:

dns list	List current DNS entries.
-----------------	---------------------------

dns save

Save current **SpeedTouch™ 570** DNS server/forwarder configuration and static entries.

SYNTAX:

```
dns save
```

EXAMPLE:

```
=>dns fwdlist
DNS forwarding servers:
DNS          SRC          MASK          Direct
10.0.0.138   10.0.0.2      255.255.255.0 yes
10.0.0.138   10.0.0.4      255.255.255.0 no
10.0.0.138   10.0.0.3      255.255.255.0 yes
=>dns save
=>dns flush
=>dns fwdlist
DNS forwarding servers:
DNS          SRC          MASK          Direct
=>dns load
=>dns fwdlist
DNS forwarding servers:
DNS          SRC          MASK          Direct
10.0.0.138   10.0.0.2      255.255.255.0 yes
10.0.0.138   10.0.0.4      255.255.255.0 no
10.0.0.138   10.0.0.3      255.255.255.0 yes
=>
```

RELATED COMMANDS:

dns flush

Flush complete DNS server/forwarder configuration and dynamic entries.

dns load

Load saved or default DNS server/forwarder configuration and static entries.

dns start

Start **SpeedTouch™ 570** DNS server/forwarder.

SYNTAX:

```
dns start
```

EXAMPLE:

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

RELATED COMMANDS:

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

dns stats

Show **SpeedTouch™ 570** DNS server/forwarder statistics.

SYNTAX:

```
dns stats
```

EXAMPLE INPUT/OUTPUT IN A NETWORKED ENVIRONMENT:

The **SpeedTouch™ 570** 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)
dnssd: Internet class type A request received from 10.10.1.1.
dnssd: Client.SpeedLAN.local found in local database.
dnssd: Client.SpeedLAN.local resolved into 10.0.0.3.
=>(Ping Server.SpeedLAN.local)
dnssd: Internet class type A request received from 10.10.1.1.
dnssd: Server.SpeedLAN.local found in local database.
dnssd: Server.SpeedLAN.local resolved into 10.0.0.3.
=>(CTRL + S)
=>dns stats
DNS Statistics:
Corrupted packets recv      :      0
Local questions resolved    :      3
Local neg answers sent      :      0
Total DNS packets fwd       :      0
External answers recv       :      0
Fwd table full, discard     :      0
Spurious answers           :      0
Unknown query types         :      0
Total number of packets received :      3
-----
=>

```

RELATED COMMANDS:

dns clrstats

Clear DNS server/forwarder statistics.

dns status

Show **SpeedTouch™ 570** DNS server/forwarder configuration.

SYNTAX:

```
dns status
```

EXAMPLE OUTPUT:

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

RELATED COMMANDS:

dns flush	Flush complete DNS server/forwarder configuration and dynamic entries.
dns load	Load saved or default DNS server/forwarder configuration and static entries.
dns save	Save current DNS server/forwarder configuration and static entries.

dns stop

Stop **SpeedTouch™ 570** DNS server/forwarder.

SYNTAX:

```
dns stop
```

EXAMPLE:

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

RELATED COMMANDS:

dns status

Show DNS server/forwarder configuration.

dns start

Start DNS server/forwarder.

dns toutfwd

Set DNS forwarding timeout.

SYNTAX:

dns toutfwd	timeout = <number>
--------------------	---------------------------------

<i>timeout</i>	A number (seconds). Represents the query forwarding timeout. This parameter determines how long the SpeedTouch™ 570 DNS server should try to contact a (remote) DNS server before (temporarily) declaring the DNS requests unresolved. By default the timeout is 15 seconds.	REQUIRED
----------------	--	----------

EXAMPLE:

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

RELATED COMMANDS:

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

dns troff

Disable verbose console messaging. No debug traces are generated.

SYNTAX:

```
dns troff
```

EXAMPLE:

```

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

```

RELATED COMMANDS:

dns fwdtable

Show DNS forwarding table.

dns fwdlist

Show current DNS forwarding entries..

dns status

Show DNS server/forwarder configuration.

dns tron

Enable verbose console messaging.

dns tron

Enable verbose console messaging. Debug traces are generated.

SYNTAX:

```
dns tron
```

EXAMPLE:

```

=>dns status
DNS server status: Started
DNS table size           :    73,   in use:    4,   free: 94 %
DNS forwarding table size :    10,   in use:    0,   free:100 %
DNS forwarding dns servers table size :    25,   in use:    4,   free: 84 %
No dns cache.
Tracing: off
=>dns tron
Tracing on.
=>dns status
DNS server status: Started
DNS table size           :    73,   in use:    4,   free: 94 %
DNS forwarding table size :    10,   in use:    0,   free:100 %
DNS forwarding dns servers table size :    25,   in use:    4,   free: 84 %
No dns cache.
Tracing: on
=>(CTRL + Q)
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 fwdtable	Show DNS forwarding table.
dns fwdlist	Show current DNS forwarding entries..
dns status	Show DNS server/forwarder configuration.
dns troff	Disable verbose console messaging.

7 Firewall Commands

firewall (to access the Firewall level)
firewall assign
firewall flush
firewall list
firewall load
firewall match
firewall save
firewall troff
firewall tron
firewall chain (to access the Firewall Chain level)
firewall chain create
firewall chain delete
firewall chain list
firewall chain load
firewall chain save
firewall rule (to access the Firewall Rule level)
firewall rule clear
firewall rule create
firewall rule delete
firewall rule flush
firewall rule list
firewall rule stats

firewall assign

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

SYNTAX:

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

hook	<p>The entry point's name to assign a chain to.</p> <p>Choose between:</p> <ul style="list-style-type: none"> ▪ input : The point off all incoming traffic. At this point it can be determined whether the packet is allowed to reach the SpeedTouch™ 570 IP router or local host. ▪ sink : The point off all traffic destined to the SpeedTouch™ 570 IP router itself. At this point it can be determined whether the packet is allowed to address the local host. ▪ forward : The point off all traffic to be forwarded by the SpeedTouch™ 570 IP router. At this point it can be determined whether the packet is allowed to be handled, i.e. routed. ▪ source : The point off all traffic sourced by the SpeedTouch™ 570 IP router. At this point it can be determined whether the packet is allowed to leave the local host. ▪ output : The point off all outgoing traffic. At this point it can be determined whether the packet is allowed to leave the SpeedTouch™ 570 IP router or local host. 	REQUIRED
chain	The name of the chain to use.	REQUIRED

EXAMPLE:

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

RELATED COMMANDS:

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

firewall flush

Flush all associations between a hook and its chain(s). The chain itself is not removed. The flush command does not impact previously saved configurations.

SYNTAX:

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

<code>[hook]</code>	the name of the hook to clear. Choose between: <ul style="list-style-type: none"> ▪ input ▪ sink ▪ forward ▪ source ▪ output. In case this parameter is not specified all hooks are cleared.	OPTIONAL
---------------------	--	----------

EXAMPLE:

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

RELATED COMMANDS:

firewall assign Assign a chain to an entry point.

firewall list

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

SYNTAX:

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

<i>[hook]</i>	<p>the name of the hook to show the associations for.</p> <p>Choose between:</p> <ul style="list-style-type: none"> ▪ input ▪ sink ▪ forward ▪ source ▪ output. <p>In case this parameter is not specified the associations for all hooks are shown.</p>	OPTIONAL
---------------	---	----------

EXAMPLEINPUT/OUTPUT:

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

RELATED COMMANDS:

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

firewall load

Load saved (or default) firewall configuration.

Execute **firewall flush** prior to **firewall load**.

SYNTAX:

firewall load	[file = <string>] [defaults = <yes no>]
----------------------	--

<i>[file]</i>	The name of the firewall configuration to be loaded. Not specifying this parameter loads the default configuration	OPTIONAL
<i>[defaults]</i>	Load factory defaults (yes) or saved configuration (no). Not specifying this parameter loads the saved configuration	OPTIONAL

EXAMPLE:

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

RELATED COMMANDS:

firewall save	Save current firewall configuration.
firewall flush	Clear associations for all or a selected entry point(s).

firewall match

Match a specified IP packet. Used to match an IP packet against a chain in order to determine what the reaction of the firewall would be.

This command can be considered as being the same as the **firewall rule create** command, but without the action to be taken.

SYNTAX:

```

firewall match      chain = <string>
                    [srcintf = <string>]
                    [src = <ip-address>]
                    [srcbridgeport = <number>]
                    [dstintf = <string>]
                    [dst = <ip-address>]
                    [tos = <number>]
                    [prot = <{tcp|udp|icmp|protocol}>]
                    [syn = <yes|no>]
                    [urg = <yes|no>]
                    [ack = <yes|no>]
                    [srcport = <{ftp|ftp-data|telnet|mail|smtp|dns|domain|tftp|port}>]
                    [dstport = <{ftp|ftp-data|telnet|mail|smtp|dns|domain|tftp|port}>]
                    [icmptype = <{echo-reply|destination-unreachable|source-quench|
                                redirect|echo-request|router-advertisement|
                                router-solicitation|time-exceeded|parameter-problems|
                                timestamp-request|timestamp-reply|
                                information-request|information-reply|
                                address-mask-request|address-mask-reply|
                                icmpnumber}>]
                    [icmpcode = <number{0-15}>]

```

<i>chain</i>	The name of the chain to match the packet against.	REQUIRED
<i>[srcintf]</i>	The name of the interface the virtual packet arrived on.	OPTIONAL
<i>[src]</i>	The source IP address the virtual packet is coming from.	OPTIONAL
<i>[srcbridgeport]</i>	A number between 0 and 6. Represents the bridge port the virtual packet arrived on. Execute bridge iflist for a list of available bridge ports.	OPTIONAL
<i>[dstintf]</i>	The name of the interface the virtual packet is going to.	OPTIONAL
<i>[dst]</i>	The destination IP address the virtual packet is going to.	OPTIONAL
<i>[tos]</i>	A number between 0 and 255. Represents the Type Of Service specification which should be expected [or NOT expected] in the IP packet. The Type of Service numbering specification is in accordance to the latest version of RFC1700: Assigned numbers.	OPTIONAL

<i>[prot]</i>	<p>The protocol (name or number) in the virtual IP packet. Choose between:</p> <ul style="list-style-type: none"> ▪ <i>tcp</i> ▪ <i>udp</i> ▪ <i>icmp</i> <p>Or specify the protocol number in accordance to the latest version of RFC1700: Assigned numbers.</p>	OPTIONAL
<i>[syn]</i>	Set the TCP SYN flag (yes) or not (no).	OPTIONAL
<i>[urg]</i>	Set the TCP URG flag (yes) or not (no).	OPTIONAL
<i>[ack]</i>	Set the TCP ACK flag (yes) or not (no).	OPTIONAL
<i>[srcport]</i>	<p>The TCP/UDP port the virtual packet is coming from. Choose between:</p> <ul style="list-style-type: none"> ▪ <i>ftp</i> ▪ <i>ftp-data</i> ▪ <i>telnet</i> ▪ <i>mail</i> ▪ <i>smtp</i> ▪ <i>dns</i> ▪ <i>domain</i> ▪ <i>fttp</i> <p>Or specify the port number in accordance to the latest version of RFC1700: Assigned numbers.</p>	OPTIONAL
<i>[dstport]</i>	<p>The TCP/UDP port the virtual packet is going to. Choose between:</p> <ul style="list-style-type: none"> ▪ <i>ftp</i> ▪ <i>ftp-data</i> ▪ <i>telnet</i> ▪ <i>mail</i> ▪ <i>smtp</i> ▪ <i>dns</i> ▪ <i>domain</i> ▪ <i>fttp</i> <p>Or specify the port number in accordance to the latest version of RFC1700: Assigned numbers.</p>	OPTIONAL

- [icmp^ttype]* The ICMP (Internet Control Message Protocol) type (name or number) of the virtual packet. OPTIONAL
Choose between:
- *echo-reply*
 - *destination-unreachable*
 - *source-quench*
 - *redirect*
 - *echo-request*
 - *router-advertisement*
 - *router-solicitation*
 - *time-exceeded*
 - *parameter-problems*
 - *timestamp-request*
 - *timestamp-reply*
 - *information-request*
 - *information-reply*
 - *address-mask-request*
 - *address-mask-reply*
- Or specify the ICMP type number in accordance to the latest version of RFC1700: Assigned numbers.
- [icmp^tcode]* A number between 0 and 15. OPTIONAL
Represents the ICMP code of the virtual packet as specified in the latest version of RFC1700: Assigned number.

EXAMPLE INPUT/OUTPUT:

```
=>firewall rule list chain=Telnet
:firewall rule create chain=Telnet index=0 srcintfgrp=lan src=10.0.0.0/8
  dst=200.200.200.1/32 prot=tcp srcport=1024 srcportend=65535dstport=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 match chain=Telnet src=200.200.200.1 dst=10.0.0.1 ack srcport=23 dstport=1023
Packet was ACCEPTED
=>
```

RELATED COMMANDS:

firewall rule create Create a firewall rule.

firewall save

Save all modifications entered by : **firewall assign**.

This command only saves the association between hook(s) and chain(s), set by the **firewall assign** command, not the information about chains, rules and their parameters .

SYNTAX:

firewall save	[file = <string>]
----------------------	--------------------------------

<i>[file]</i>	A name for the current firewall configuration file to be saved. The name is limited to 9 characters. This parameter allows multiple firewall configurations to be saved under different names. In case this parameter is not specified the configuration is saved as single configuration.	OPTIONAL
---------------	---	----------

EXAMPLE:

```
=>firewall list
assign hook=sink chain=sink
assign hook=forward chain=forward
assign hook=source chain=source
=>firewall assign hook input chain Test
=>firewall list
assign hook=input chain=Test
assign hook=sink chain=sink
assign hook=forward chain=forward
assign hook=source chain=source
=>firewall save file=TestCnfg
=>firewall flush
=>firewall list
=>firewall load file=TestCnfg
=>firewall list
assign hook=input chain=Test
assign hook=sink chain=sink
assign hook=forward chain=forward
assign hook=source chain=source
=>
```

RELATED COMMANDS:

firewall load	Load a saved or default firewall configuration.
firewall flush	Clear associations for all or a selected entry point(s).
firewall chain save	Save current chain(s) configuration.

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 chain create

Create a new chain.

SYNTAX:

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

<i>chain</i>	The name of the chain to create.	REQUIRED
--------------	----------------------------------	----------

EXAMPLE:

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

RELATED COMMANDS:

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

firewall chain delete

Delete a chain.

SYNTAX:

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

<i>chain</i>	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:

RELATED COMMANDS:

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

firewall chain list

Show a list of all current chains.

SYNTAX:

```
firewall chain list
```

EXAMPLEINPUT/OUTPUT:

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

RELATED COMMANDS:

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

firewall chain load

Load saved or default chain(s) configuration (with related rules).

SYNTAX:

```
firewall chain load [file = <string>]
                    [defaults = <yes|no>]
```

[file]	The name of the chain configuration to be loaded. Not specifying this parameter loads the default configuration	OPTIONAL
[defaults]	Load factory defaults (yes) or saved configuration (no). Not specifying this parameter loads the saved configuration	OPTIONAL

EXAMPLE:

```
=>firewall flush
=>firewall load
=>firewall chain list
source, forward, sink
=>firewall flush
=>firewall chain load file TelConfig
=>firewall chain list
sink, forward, source, Telnet
=>
```

RELATED COMMANDS:

firewall assign	Assign a chain to an entry point.
firewall chain create	Create a chain.
firewall chain delete	Delete a chain.
firewall chain list	Show a list of all current chains.
firewall chain save	Save current chain(s) configuration.

firewall chain save

Save all modifications entered by : **firewall chain create**, **firewall chain delete** .

This command saves the information about chains, rules and their parameters. This is different from **firewall save** which saves only the association between hook(s) and chain(s), set by the **firewall assign** command.

SYNTAX:

```
firewall chain save [file = <string>]
```

<i>[file]</i>	A name for the current chain configuration file to be saved. The name is limited to 9 characters. This parameter allows multiple chain configurations to be saved under different names. In case this parameter is not specified the configuration is saved as single configuration.	OPTIONAL
---------------	---	----------

EXAMPLE:

```
=>firewall flush
=>firewall chain load file TelConfig
=>firewall chain list
Telnet, source, forward, sink
=>firewall chain create chain NewTel
=>firewall chain list
NewTel, Telnet, source, forward, sink
=>firewall chain save file NewTelcnf
=>firewall flush
=>firewall chain load file NewTelcnf
=>firewall chain list
sink, forward, source, Telnet, NewTel
=>
```

RELATED COMMANDS:

firewall assign	Assign a chain to an entry point.
firewall chain create	Create a chain.
firewall chain delete	Delete a chain.
firewall chain list	Show a list of all current chains.
firewall chain load	Load saved or default chain configuration(s).

firewall rule clear

Clear statistics for a given rule.

SYNTAX:

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

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

EXAMPLE:

```
=>firewall rule stats
ChainTelnet, index0, packets 0, bytes 0
ChainTelnet, index1, packets 0, bytes 0
ChainTelnet, index2, packets 0, bytes 0
Chainsource, index0, packets 203, bytes 15229
Chainsource, index1, packets 0, bytes 0
Chainsource, index2, packets 0, bytes 0
Chainforward,index0, packets 0, bytes 0
Chainsink, index0, packets 202, bytes 10159
Chainsink, index1, packets 0, bytes 0
Chainsink, index2, packets 0, bytes 0
=>firewall rule clear chain=source index=0
=>firewall rule stats
ChainTelnet, index0, packets 0, bytes 0
ChainTelnet, index1, packets 0, bytes 0
ChainTelnet, index2, packets 0, bytes 0
Chainsource, index0, packets 11, bytes 559
Chainsource, index1, packets 0, bytes 0
Chainsource, index2, packets 0, bytes 0
Chainforward,index0, packets 0, bytes 0
Chainsink, index0, packets 409, bytes 21535
Chainsink, index1, packets 0, bytes 0
Chainsink, index2, packets 0, bytes 0
=>
```

RELATED COMMANDS:

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

firewall rule create

Create a rule.

SYNTAX:

```

firewall rule create chain = <string>
[index = <number>]
[srcintf [!]= <string>]
[srcintfgrp [!]= <{wan|local|lan}>]
[srcbridgeport [!]= <number>]
[src [!]= <ip-address>]
[srcmsk = <ip-mask(dotted or cidr)>]
[dstintf [!]= <string>]
[dstintfgrp [!]= <{wan|local|lan}>]
[dst [!]= <ip-address>]
[dstmsk = <ip-mask(dotted or cidr)>]
[tos [!]= <number{1-255}>]
[prot [!]= <{tcp|udp|icmp|protocol}>]
[syn <yes|no>]
[urg <yes|no>]
[ack <yes|no>]
[srcport [!]= <{ftp|ftp-data|telnet|mail|smtp|dns|domain|tftp|port}>]
[srcportend = <{ftp|ftp-data|telnet|mail|smtp|dns|domain|tftp|port}>]
[dstport [!]= <{ftp|ftp-data|telnet|mail|smtp|dns|domain|tftp|port}>]
[dstportend = <{ftp|ftp-data|telnet|mail|smtp|dns|domain|tftp|port}>]
[icmptype [!]= <{echo-reply|destination-unreachable|source-quench|
redirect|echo-request|router-advertisement|
router-solicitation|time-exceeded|parameter-problems|
timestamp-request|timestamp-reply|
information-request|information-reply|
address-mask-request|address-mask-reply|
icmpnumber}>]
[icmpcode [!]= <number{0-15}>]
[icmpcodeend = <number{0-15}>]
[clink = <string>]
action = <{accept|deny|drop|count}>

```

<i>chain</i>	The name of the chain to insert the rule in.	REQUIRED
[<i>index</i>]	The number of the rule before which the new rule must be added.	OPTIONAL
[<i>srcintf</i>]	The name of the interface the packet should [or should NOT] arrive on to make this rule apply. (NOT applicable if used in a chain assigned to the output hook)	OPTIONAL
[<i>srcintfgrp</i>]	The interface group the packet should [or should NOT] arrive on. Choose between: <ul style="list-style-type: none"> ▪ wan ▪ local ▪ lan (NOT applicable if used in a chain assigned to the output hook)	OPTIONAL

<i>[srcbridgeport]</i>	A number between 0 and 6. Represents the bridge port the virtual packet should [or should NOT] arrive on. Execute bridge iflist for a list of available bridge ports.	OPTIONAL
<i>[src]</i>	The source IP address (range) the packet should [or should NOT] come from. (Supports cidr notation).	OPTIONAL
<i>[srcmsk]</i>	The source IP address mask defining the range (see src).	OPTIONAL
<i>[dstintf]</i>	The name of the interface the packet should [or should NOT] be going to. (NOT applicable if used in a chain assigned to the input hook)	OPTIONAL
<i>[dstintfgrp]</i>	The interface group the packet should [or should NOT] be going to. Choose between: <ul style="list-style-type: none"> ▪ wan ▪ local ▪ lan (NOT applicable if used in a chain assigned to the input hook)	OPTIONAL
<i>[dst]</i>	The destination IP address (range) the packet should [or should NOT] be going to. (supports cidr notation).	OPTIONAL
<i>[dstmsk]</i>	The destination IP address mask defining the range (see dst).	OPTIONAL
<i>[tos]</i>	A number between 0 and 255. Represents the Type Of Service specification which should be expected [or NOT expected] in the IP packet. The Type of Service numbering specification is in accordance to the latest version of RFC1700: Assigned numbers.	OPTIONAL
<i>[prot]</i>	The protocol (name or number) in the IP packet expected [or NOT expected] in the IP packet. Choose between: <ul style="list-style-type: none"> ▪ tcp ▪ udp ▪ icmp Or specify the protocol number in accordance to the latest version of RFC1700: Assigned numbers.	OPTIONAL
<i>[syn]</i>	Expect TCP SYN flag set (yes) or not (no). In combination with TCP ACK this allows selection of incoming versus outgoing TCP connections.	OPTIONAL
<i>[urg]</i>	Expect TCP URG flag set (yes) or not (no).	OPTIONAL
<i>[ack]</i>	Expect TCP ACK flag set (yes) or not (no).	OPTIONAL

<i>[srcport]</i>	<p>The TCP/UDP port (or beginning of range) the packet should [or should NOT] be from.</p> <p>Choose between:</p> <ul style="list-style-type: none"> ▪ ftp ▪ ftp-data ▪ telnet ▪ mail ▪ smtp ▪ dns ▪ domain ▪ tftp <p>Or specify the port number in accordance to the latest version of RFC1700: Assigned numbers.</p>	OPTIONAL
<i>[srcportend]</i>	<p>The source TCP/UDP port range end (inclusive). (Only applicable for ranges)</p>	OPTIONAL
<i>[dstport]</i>	<p>The TCP/UDP port (or beginning of range) the packet should [or should NOT] be going to.</p> <p>Choose between:</p> <ul style="list-style-type: none"> ▪ ftp ▪ ftp-data ▪ telnet ▪ mail ▪ smtp ▪ dns ▪ domain ▪ tftp <p>Or specify the port number in accordance to the latest version of RFC1700: Assigned numbers.</p>	OPTIONAL
<i>[dstportend]</i>	<p>The destination TCP/UDP port range end (inclusive). (Only applicable for ranges)</p>	OPTIONAL
<i>[icmptype]</i>	<p>The expected [or NOT expected] ICMP type (name or number) of the packet.</p> <p>Choose between:</p> <ul style="list-style-type: none"> ▪ <i>echo-reply</i> ▪ <i>destination-unreachable</i> ▪ <i>source-quench</i> ▪ <i>redirect</i> ▪ <i>echo-request</i> ▪ <i>router-advertisement</i> ▪ <i>router-solicitation</i> ▪ <i>time-exceeded</i> ▪ <i>parameter-problems</i> ▪ <i>timestamp-request</i> ▪ <i>timestamp-reply</i> ▪ <i>information-request</i> ▪ <i>information-reply</i> ▪ <i>address-mask-request</i> ▪ <i>address-mask-reply</i> <p>Or specify the ICMP type number in accordance to the latest version of RFC1700: Assigned numbers.</p>	OPTIONAL

<i>[icmpcode]</i>	A number between 0 and 15. Represents the expected [or NOT expected] ICMP code (or beginning of range) of the packet as specified in the latest version of RFC1700: Assigned number.	OPTIONAL
<i>[icmpcodeend]</i>	A number between 0 and 15. Represents the ICMP code range end. Only applicable for ranges.	OPTIONAL
<i>[clink]</i>	The name of the chain to be parsed when this rule applies. (action is ignored).	OPTIONAL
<i>action</i>	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, that is, without sending an error message to the sender. ▪ count : update of statistics. Has no influence on the packet. 	REQUIRED

EXAMPLE:

```
=>firewall rule list chain=Telnet
=>firewall rule create chain=telnet src=10.0.0.0/8 dst=200.200.200.1 srcintfgrp=lan
  prot=tcp srcport=1024 srcportend=65535dstport=23
  action=accept
=>firewall rule create chain=telnet src=200.200.200.1 dst=10.0.0.0/8 srcintfgrp=wan
  prot=tcp srcport=23 dstport=1024 dstportend=65535
  action=accept
=>firewall rule create chain=telnet
  action=drop
=>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=65535dstport=telnet
  action=accept
:firewall rule create chain=Telnet index=1 srcintfgrp=wan src=200.200.200.1/32
  dst=10.0.0.0/8 prot=tcp srcport=telnet dstport=1024 dstportend=65535
  action=accept
:firewall rule create chain=Telnet index=2
  action=drop
=>
```

RELATED COMMANDS:

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

firewall rule delete

Delete a rule.

SYNTAX:

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

<i>chain</i>	The name of the chain in which to delete the rule.	REQUIRED
<i>index</i>	The index number of the rule in the chain. Execute firewall rule list first to determine the index number of the applicable rule.	REQUIRED

EXAMPLE:

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

RELATED COMMANDS:

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

firewall rule flush

Flush all rules created for a chain(s). The chain itself is not removed. The flush command does not impact previously saved configurations.

SYNTAX:

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

<i>[chain]</i>	The name of the chain to empty. In case this parameter is not specified all rules for all chains are deleted.	OPTIONAL
----------------	--	----------

EXAMPLE:

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

RELATED COMMANDS:

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

firewall rule list

Show a list of rules.

SYNTAX:

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

<i>[chain]</i>	The name of the chain to list the rules of. In case this parameter is not specified all rules for all chains are shown.	OPTIONAL
----------------	--	----------

EXAMPLE INPUT AND OUTPUT:

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

RELATED COMMANDS:

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

firewall rule stats

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

SYNTAX:

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

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

EXAMPLE OUTPUT:

```
=>firewall rule list chain=Test
:firewall rule create chain=Test index=0 srcintfgrp=lan src=200.200.0.1/32
  dst=200.200.0.2/32 prot=udp srcport=0 srcportend=65535dstport=telnet
  action=deny
=>firewall rule clear
=>firewall rule stats
Chainsink, index0, packets 43, bytes 1743
Chainsink, index1, packets 0, bytes 0
Chainsink, index2, packets 0, bytes 0
Chainsink, index3, packets 0, bytes 0
Chainforward,index0, packets 0, bytes 0
Chainsource, index0, packets 43, bytes 1977
Chainsource, index1, packets 0, bytes 0
Chainsource, index2, packets 0, bytes 0
ChainTest, index0, packets 0, bytes 0
=>firewall rule stats
Chainsink, index0, packets 104, bytes 6143
Chainsink, index1, packets 0, bytes 0
Chainsink, index2, packets 0, bytes 0
Chainsink, index3, packets 0, bytes 0
Chainforward,index0, packets 0, bytes 0
Chainsource, index0, packets 43, bytes 1977
Chainsource, index1, packets 0, bytes 0
Chainsource, index2, packets 0, bytes 0
ChainTest, index0, packets 44, bytes 21032
=>
```

DESCRIPTION:

The statistics for the 'Test' chain are the result of sending udp packets to the **SpeedTouch™ 570**. The chain 'Test' is assigned to the hook 'input' and prohibits the sending of udp packets from one host to another.

EXAMPLE INPUT/OUTPUT IN A NETWORKED ENVIRONMENT:

The **SpeedTouch™ 570** is configured as DHCP client on its Ethernet interface eth0.

```
=>firewall rule list chain=Sending
:firewall rule create chain=Sendingindex=0 srcintfgrp=lan src=10.0.0.3/32
  dst=10.10.1.1/32prot=icmp action=count
:firewall rule create chain=Sendingindex=1 srcintfgrp=lan src=10.10.1.1/32
  dst=10.0.0.3/32 prot=icmp action=count
=>firewall rule stats
Chainsource, index0, packets 0, bytes 0
Chainsource, index1, packets 0, bytes 0
Chainsource, index2, packets 0, bytes 0
Chainsource, index3, packets 0, bytes 0
Chainforward,index0, packets 0, bytes 0
Chainsink, index0, packets 0, bytes 0
Chainsink, index1, packets 144, bytes 5844
Chainsink, index2, packets 0, bytes 0
Chainsink, index3, packets 0, bytes 0
Chainsink, index4, packets 0, bytes 0
Chainsink, index5, packets 0, bytes 0
Chainsending,index0, packets 0, bytes 0
Chainsending,index1, packets 0, bytes 0
=>firewall rule clear
=>(Ping from server 10.10.1.1 to client 10.0.0.3)
=>firewall rule stats
Chainsource, index0, packets 0, bytes 0
Chainsource, index1, packets 0, bytes 0
Chainsource, index2, packets 0, bytes 0
Chainsource, index3, packets 0, bytes 0
Chainforward,index0, packets 0, bytes 0
Chainsink, index0, packets 0, bytes 0
Chainsink, index1, packets 42, bytes 1782
Chainsink, index2, packets 0, bytes 0
Chainsink, index3, packets 0, bytes 0
Chainsink, index4, packets 0, bytes 0
Chainsink, index5, packets 0, bytes 0
Chainsending,index0, packets 4, bytes 240
Chainsending,index1, packets 4, bytes 240
=>
```

RELATED COMMANDS:

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

8 IP Commands

ip (to access the IP level)

ip apadd

ip apdelete

ip aplist

ip arpadd

ip arpdelete

ip arplist

ip config

ip flush

ip ifconfig

ip iflist

ip load

ip ping

ip rtadd

ip rtdelete

ip rtlist

ip save

ip sendto

ip apadd

Assign an IP address to an interface.

SYNTAX:

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

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

EXAMPLE:

```

=>ip aplist
1  eth0      Type:EthernetHWaddr 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
=>ip apadd addr=10.0.0.2 netmask=255.255.255.0 intf=eth0 addrtrans=pat addroute=yes
=>ip aplist
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
1  eth0      Type:EthernetHWaddr 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 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 apdelete
ip aplist

Remove an IP address from an interface.
 Show current IP addresses.

ip apdelete

Remove an IP address from an interface.

SYNTAX:

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

<i>addr</i>	The IP address to delete.	REQUIRED
-------------	---------------------------	----------

EXAMPLE:

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

RELATED COMMANDS:

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

ip aplist

Show a list of all configured IP addresses.

SYNTAX:

```
ip aplist
```

EXAMPLE:

```

=>ip aplist
2  eth0      Type:EthernetHWaddr 00:80:9f:24:ab:cf BRHWaddr  ff:ff:ff:ff:ff:ff
   inet addr:10.0.0.138      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:EthernetHWaddr 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 aplist

Add an IP address to an interface.

ip adelete

Remove an IP address from an interface.

ip arpadd

Add a static entry to the **SpeedTouch™ 570** ARP cache.

SYNTAX:

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

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

EXAMPLE:

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

RELATED COMMANDS:

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

ip arpdelete

Remove an entry from the SpeedTouch™ 570 ARP cache.

SYNTAX:

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

<i>intf</i>	The interface name.	REQUIRED
<i>ip</i>	The IP address.	REQUIRED
<i>[hwaddr]</i>	The hardware address.	OPTIONAL

EXAMPLE:

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

RELATED COMMANDS:

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

ip arplist

Show the **SpeedTouch™ 570** ARP cache.

SYNTAX:

```
ip arplist
```

EXAMPLE OUTPUT:

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

RELATED COMMANDS:

ip arpadd

Add a static entry to the ARP cache.

ip arpdelete

Delete an entry from the ARP cache.

ip config

Show/set global IP stack configuration options.

SYNTAX:

ip config	<pre>[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}>]</pre>	
[forwarding]	Disable (off) or enable (on) the IP routing functionality.	OPTIONAL
[firewalling]	Enable (on) or disable (off) IP firewalling (master switch). If applicable the CLI firewall level allows configuration of the SpeedTouch™ 570 firewall. For security reasons this parameter is enabled per default. It is strongly recommended never to disable the SpeedTouch™ 570 firewall.	OPTIONAL
[redirects]	Disable (off) or enable (on) the sending of ICMP redirect messages. A router can send a redirect message in case a shorter path than the path followed is discovered. For security reasons this parameter is disabled per default.	OPTIONAL
[sourcerouting]	Disallow (off) or allow (on) IP source routed packets. IP source routed packets are packets with the route to follow specified in the header. For security reasons this parameter is disabled per default.	OPTIONAL
[netbroadcasts]	Disallow (off) or allow (on) net directed broadcasts. This parameter is per default disabled. In case netbroadcasts are allowed no traces of netbroadcasts are generated.	OPTIONAL
[ttl]	A number between 0 and 255. Represents the default time-to-live (ttl) for locally generated IP packets. This parameter determines the number of hop-counts the IP packet may pass before it is dropped. Generally the time-to-live is 64 hop-counts. By limiting the time-to-live continuous circulation of IP packets on the network without ever reaching a destination is avoided.	OPTIONAL
[fraglimit]	A number between 1 and 1024. Represents the maximum number of IP packet fragments waiting for completion. Generally the fragmentation limit is 64. By limiting the fragmentation limit the depletion of the buffer is avoided.	OPTIONAL

- [defragmode]* Define which packets are reassembled under which circumstances. OPTIONAL
Choose between:
- **normal**
Packets to be forwarded will not be reassembled.
Packets with local destination, i.e. destined for the **SpeedTouch™ 570**, are reassembled.
 - **always**
Packets are always reassembled.
 - **nat**
Same behaviour as **normal** except for packets to be forwarded through the NAT engine.
Packets on which address translation is performed are reassembled as the NAT engine requires the entire packet.
- [addrcheck]* Set the level of IP address checks. OPTIONAL
Choose between:
- **off**
No address checking is performed.
For advanced users only; in normal circumstances there should always be some kind of address checking.
 - **own**
Minimum level of checking.
Only the address configuration on the **SpeedTouch™ 570** is checked.
 - **static**
Checking of the address configuration of the **SpeedTouch™ 570** and also of traffic: addresses of incoming packets; this checking is related to constants (e. g. an address may not be entirely composed of one's or zero's).
 - **dynamic**
Besides the address configuration of the **SpeedTouch™ 570** itself, and besides the checking of traffic on a constants level, additional checking is performed on the IP addresses that are determined by the configuration, more specifically by the network.
- [mssclamping]* Disable (off) or enable (on) mss clamping for low mtu interfaces. OPTIONAL
Mss clamping assures that the size of a TCP packet never exceeds the available mtu of the outgoing interface.
It is recommended not to disable this parameter.

EXAMPLE:

```
=>ip config
Forwarding on
Firewalling off
Sendredirects off
Soucerouting 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
Soucerouting on
NetBroadcasts off
Default TTL 64
Fraglimit 64 fragments
Fragcount currently 0 fragments
Defragment mode : nat
Address checks : static
Mss clamping : on
=>
```

RELATED COMMANDS:

ip ifconfig

Configure interface parameters.

ip flush

Flush complete IP configuration. Dynamic configurations (e.g. from PPP or CIP links) remain. The flush command does not impact previously saved configurations.

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

SYNTAX:

```
ip flush
```

EXAMPLE:

```
=>ip aplist
3   cipl      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      broadcastpkts:0
    IPTX bytes:0      unicastpkts:0      broadcastpkts:0      dropkts:0
    HWRX bytes:0      unicastpkts:0      broadcastpkts:0      dropkts:0
    HWTX bytes:0      unicastpkts:0      broadcastpkts: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      broadcastpkts:0
    IPTX bytes:0      unicastpkts:0      broadcastpkts:0      dropkts:0
    HWRX bytes:0      unicastpkts:0      broadcastpkts:0      dropkts: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      dropkts:0
    HWTX bytes:0      unicastpkts:0      broadcastpkts:0      dropkts:0
=>ip flush
=>ip aplist
3   cipl      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      broadcastpkts:0
    IPTX bytes:0      unicastpkts:0      broadcastpkts:0      dropkts:0
    HWRX bytes:0      unicastpkts:0      broadcastpkts:0      dropkts: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      dropkts:0
    HWTX bytes:0      unicastpkts:0      broadcastpkts:0      dropkts:0
=>
```

RELATED COMMANDS:

ip load Load saved or default IP configuration.
ip save Save current IP configuration.

ip ifconfig

Configure interface parameters.

SYNTAX:

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

<i>intf</i>	The IP interface name.	REQUIRED
[<i>mtu</i>]	A number between 293 and 20000. Represents the maximum transmission unit, i.e. the maximum packet size (including IP header) to use on this interface. The default value depends on the connection and packet service for which the interface was created.	OPTIONAL
[<i>status</i>]	The administrative status of the interface. Choose between: <ul style="list-style-type: none"> ▪ down ▪ up 	OPTIONAL
[<i>hwaddr</i>]	The hardware address (e.g. the Ethernet MAC address) of this interface.	OPTIONAL
[<i>group</i>]	The group this interface belongs to (e.g. for oriented firewalling).	OPTIONAL

EXAMPLE:

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

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

RELATED COMMANDS:

ip config Show/set global IP stack configuration options.

ip iflist

Show all current interfaces.

SYNTAX:

```
ip iflist
```

EXAMPLE OUTPUT:

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

RELATED COMMANDS:

ip ifconfig

Configure interface parameters.

ip load

Load saved (or default) IP configuration.

Execute **ip flush** prior to **ip load**.

SYNTAX:

```
ip load [defaults = <yes|no>]
```

[defaults] Load factory defaults (yes) or saved configuration (no). OPTIONAL
Not specifying this parameter loads the saved configuration

EXAMPLE:

```

=>ip rtlist
  Destination      Source           Gateway          Intf            Mtrc
  192.16.11.0/24   192.16.11.0/24  192.16.11.140   eth0            0
  172.16.1.1/32   0.0.0.0/0       172.16.1.1      cip0            0
  192.16.11.140/32 0.0.0.0/0       192.16.11.140   eth0            0
  127.0.0.1/32    0.0.0.0/0       127.0.0.1       loop            0
  172.16.1.0/24   0.0.0.0/0       172.16.1.1      cip0            0
  192.16.11.0/24   0.0.0.0/0       192.16.11.140   eth0            0
=>ip save
=>ip flush
=>ip rtlist
  Destination      Source           Gateway          Intf            Mtrc
  172.16.1.1/32    0.0.0.0/0       172.16.1.1      cip0            0
  127.0.0.1/32    0.0.0.0/0       127.0.0.1       loop            0
  172.16.1.0/24   0.0.0.0/0       172.16.1.1      cip0            0
=>ip load defaults=yes
=>ip rtlist
  Destination      Source           Gateway          Intf            Mtrc
  10.0.0.0/8       10.0.0.0/8      10.0.0.138      eth0            0
  255.255.255.255/32 0.0.0.0/0       10.0.0.138      eth0            0
  10.0.0.138/32    0.0.0.0/0       10.0.0.138      eth0            0
  172.16.1.1/32    0.0.0.0/0       172.16.1.1      cip0            0
  127.0.0.1/32    0.0.0.0/0       127.0.0.1       loop            0
  172.16.1.0/24   0.0.0.0/0       172.16.1.1      cip0            0
  10.0.0.0/8       0.0.0.0/0       10.0.0.138      eth0            0
  0.0.0.0/0        0.0.0.0/0       10.0.0.138      eth0            0
=>ip load
=>ip rtlist
  Destination      Source           Gateway          Intf            Mtrc
  192.16.11.0/24   192.16.11.0/24  192.16.11.140   eth0            0
  172.16.1.1/32   0.0.0.0/0       172.16.1.1      cip0            0
  192.16.11.140/32 0.0.0.0/0       192.16.11.140   eth0            0
  127.0.0.1/32    0.0.0.0/0       127.0.0.1       loop            0
  172.16.1.0/24   0.0.0.0/0       172.16.1.1      cip0            0
  192.16.11.0/24   0.0.0.0/0       192.16.11.140   eth0            0
=>

```

RELATED COMMANDS:

ip flush Flush complete IP configuration.
ip save Save current IP configuration.

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}>]
----------------	--

<i>addr</i>	The destination IP address.	REQUIRED
<i>[count]</i>	A number between 1 and 1000000. Represents the number of pings to send.	OPTIONAL
<i>[size]</i>	A number between 1 and 20000 (bytes). Represents the size of the ping packet(s).	OPTIONAL
<i>[interval]</i>	A number between 100 and 10000000 (milliseconds). Represents the intermediate interval between two sent ICMP packets.	OPTIONAL
<i>[listen]</i>	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™ 570** routing table.

SYNTAX:

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

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

EXAMPLE:

```

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

```

RELATED COMMANDS:

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

ip rtdelete

Delete a route from the **SpeedTouch™ 570** routing table.

SYNTAX:

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

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

EXAMPLE:

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

RELATED COMMANDS:

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

ip rtlist

Show current **SpeedTouch™ 570** routing table.

SYNTAX:

```
ip rtlist
```

EXAMPLE OUTPUT:

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

RELATED COMMANDS:

ip rtadd

Add a route to the routing table.

ip rtdelete

Remove a route from the routing table.

ip save

Save current IP configuration.

SYNTAX:

ip save

EXAMPLE:

```

=>ip rtlist
  Destination      Source          Gateway         Intf           Mtrc
  192.16.11.0/24   192.16.11.0/24 192.16.11.140 eth0           0
  172.16.1.1/32    0.0.0.0/0       172.16.1.1     cip0           0
  192.16.11.140/32 0.0.0.0/0       192.16.11.140 eth0           0
  127.0.0.1/32     0.0.0.0/0       127.0.0.1      loop           0
  172.16.1.0/24    0.0.0.0/0       172.16.1.1     cip0           0
  192.16.11.0/24   0.0.0.0/0       192.16.11.140 eth0           0
=>ip save
=>ip flush
=>ip rtlist
  Destination      Source          Gateway         Intf           Mtrc
  172.16.1.1/32    0.0.0.0/0       172.16.1.1     cip0           0
  127.0.0.1/32     0.0.0.0/0       127.0.0.1      loop           0
  172.16.1.0/24    0.0.0.0/0       172.16.1.1     cip0           0
=>ip load
=>ip rtlist
  Destination      Source          Gateway         Intf           Mtrc
  192.16.11.0/24   192.16.11.0/24 192.16.11.140 eth0           0
  172.16.1.1/32    0.0.0.0/0       172.16.1.1     cip0           0
  192.16.11.140/32 0.0.0.0/0       192.16.11.140 eth0           0
  127.0.0.1/32     0.0.0.0/0       127.0.0.1      loop           0
  172.16.1.0/24    0.0.0.0/0       172.16.1.1     cip0           0
  192.16.11.0/24   0.0.0.0/0       192.16.11.140 eth0           0
=>

```

RELATED COMMANDS:

ip flush

Flush complete IP configuration.

ip load

Load saved or default IP configuration.

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}>
------------------	--

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

EXAMPLE:

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

RELATED COMMANDS:

ip ping

Send ICMP ECHO_REQUEST packets.

9 MER Commands

mer (to access the MER level)
mer flush
mer ifadd
mer ifattach
mer ifconfig
mer ifdelete
mer ifdetach
mer iflist
mer load
mer save

mer flush

Flush complete MER configuration.

The flush command does not impact previously saved configurations.

SYNTAX:

```
mer flush
```

EXAMPLE:

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

=>mer flush
=>mer iflist
=>
```

RELATED COMMANDS:

mer load

Load saved or default MER configuration.

mer save

Save current MER configuration.

mer ifadd

Create a new MER interface.

SYNTAX:

mer ifadd	[<i>intf</i> = <string>] [<i>dest</i> = <phonebook entry>]
------------------	---

[<i>intf</i>]	The name for the new MER interface. If not specified, the destination parameter must be specified. In this case the name of the destination will double as interface name.	OPTIONAL
[<i>dest</i>]	The destination for the new MER interface. Typically, an phonebook entry.	OPTIONAL

EXAMPLE:

```

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

```

RELATED COMMANDS:

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

mer ifattach

Attach (i.e. connect) a MER interface.

SYNTAX:

mer ifattach	intf = <ifname>
---------------------	------------------------------

<i>intf</i>	The name of the MER interface to attach.	REQUIRED
-------------	--	----------

EXAMPLE:

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

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

=>
```

RELATED COMMANDS:

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

mer ifconfig

Configure a MER interface.

SYNTAX:

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

<i>intf</i>	The name of the MER interface to configure.	REQUIRED
<i>[dest]</i>	The destination for this interface. Typically a phonebook entry. This parameter needs only to be specified in case of an interface created without specified destination.	OPTIONAL
<i>[qos]</i>	The name of a configured Quality Of Service book entry. This parameter never needs to be specified.	OPTIONAL
<i>[encaps]</i>	The type of encapsulation to be used for this bridge interface. Choose between: <ul style="list-style-type: none"> ▪ llc/snap ▪ vcmux 	OPTIONAL
<i>[retry]</i>	A number between 0 and 65535. Represents the number of WAN connection setup retries before giving up. By default the retry value is 10.	OPTIONAL

EXAMPLE:

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

RELATED COMMANDS:

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

mer ifdelete

Delete a MER interface.

SYNTAX:

mer ifdelete	intf = <ifname>
---------------------	------------------------------

<i>intf</i>	The name of the MER interface.	REQUIRED
-------------	--------------------------------	----------

EXAMPLE:

```

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

```

RELATED COMMANDS:

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

mer ifdetach

Detach a MER interface.

SYNTAX:

mer ifdetach	intf = <ifname>
---------------------	------------------------------

<i>intf</i>	The name of the MER interface.	REQUIRED
-------------	--------------------------------	----------

EXAMPLE:

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

RELATED COMMANDS:

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

mer iflist

Show all or a specified MER interface(s).

SYNTAX:

mer iflist	[intf = <ifname>]
-------------------	--------------------------------

<i>[intf]</i>	The name of the MER interface. If not specified all MER interfaces are listed.	OPTIONAL
---------------	---	----------

EXAMPLE OUTPUT:

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

RELATED COMMANDS:

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

mer load

Load saved (or default) MER configuration.

Execute **mer flush** prior to **mer load**.

SYNTAX:

mer load	[defaults <yes no>]
-----------------	----------------------------------

<i>[defaults]</i>	Load factory defaults (yes) or saved configuration (no). Not specifying this parameter loads the saved configuration	OPTIONAL
-------------------	---	----------

EXAMPLE:

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

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

=>
```

RELATED COMMANDS:

mer flush	Flush complete MER configuration.
mer save	Save current MER configuration.

mer save

Save current MER configuration.

SYNTAX:

```
mer save
```

EXAMPLE:

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

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

=>
```

RELATED COMMANDS:

mer flush

Flush complete MER configuration.

mer load

Load saved or default MER configuration.

10 NAT Commands

nat (to access the NAT level)
nat applist
nat bind
nat bindlist
nat create
nat defserver
nat delete
nat disable
nat enable
nat flush
nat list
nat load
nat save
nat unbind

nat applist

List available NAPT protocol helpers.

Certain protocols are 'sensitive' to NAPT in that they do not function properly when dealing with it. This list shows which 'NAPT-sensitive' applications are supported on the **SpeedTouch™ 570**, i.e. the inherent knowledge of the **SpeedTouch™ 570** on this matter.

SYNTAX:

```
nat applist
```

EXAMPLE OUTPUT:

```
=>nat applist
Application Proto DefaultPort
H254        tcp      0
H323        tcp      1720
RAUDIO(PNA) tcp      7070
RTSP        tcp      554
IRC         tcp      6667
FTP         tcp      21
=>
```

RELATED COMMANDS:

nat bind

Create a new helper/port binding.

nat bindlist

List current NAPT helper/port bindings.

nat unbind

Delete an existing helper/port binding.

nat bind

Create a new helper/port binding.

SYNTAX:

nat bind	application = <string> port = <TCP/UDP service name or port number>
-----------------	--

<i>application</i>	The name of a NAT application helper. The name must be spelled exactly as listed in the application list (nat applist).	REQUIRED
<i>port</i>	The port number this application handler should work on.	REQUIRED

EXAMPLE INPUT:

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

RELATED COMMANDS:

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

nat bindlist

List current NAT helper/port bindings.

SYNTAX:

```
nat bindlist
```

EXAMPLE OUTPUT:

```
=>nat bindlist
Application  Proto  Port
RAUDIO(PNA) tcp    7071
H323         tcp    1720
FTP          tcp    21
RTSP         tcp    554
IRC          tcp    6667
RAUDIO(PNA) tcp    7070
=>
```

RELATED COMMANDS:

nat applist

List available NAT protocol helpers.

nat bind

Create a new NAT helper/port binding.

nat unbind

Delete an existing helper/port binding.

nat create

Create a static NAPT entry. Typically used to install specific servers behind the **SpeedTouch™ 570's** NAPT device.

SYNTAX:

nat create	protocol = <IP protocol name or number> inside_addr = <ip-address> [inside_port = <TCP/UDP service name or port number>] outside_addr = <ip-address> [outside_port = <TCP/UDP service name or port number>]
-------------------	--

<i>protocol</i>	The IP protocol name (or number) of the incoming stream.	REQUIRED
<i>inside_addr</i>	The IP address of the local host (intended to receive the incoming traffic) behind the SpeedTouch™ 570's NAPT device. Typically, a private IP address.	REQUIRED
<i>[inside_port]</i>	The port number of the application on the local host. Applicable for TCP and UDP protocols. All other protocols do not need a port to be specified.	OPTIONAL
<i>outside_addr</i>	The apparent host IP address this application is running on, i.e. the NAPT enabled WAN IP address of the SpeedTouch™ 570 . Use '0' to create a template. Such template will then be valid for any of SpeedTouch™ 570's NAPT enabled IP addresses, e.g. also dynamically assigned/negotiated IP addresses.	REQUIRED
<i>[outside_port]</i>	The apparent port number this application is running on. Applicable for TCP and UDP protocols. All other protocols do not need a port to be specified.	OPTIONAL

EXAMPLE:

```

=>nat list
=>ip aplist
1 eth0      Type:EthernetHWaddr 00:80:9f:24:ab:cf BRHWaddr ff:ff:ff:ff:ff:ff
  inet addr:10.10.10.147 Bcast: 10.10.10.255 Mask:255.0.0.0
  UP RUNNING pat MTU:1500 ReasmMAX:65535 Group:2
  IPRX bytes:19791886 unicastpkts:11341 brcastpkts:290555
  IPTX bytes:839550 unicastpkts:11477 brcastpkts:0 droppkts:0
  HWRX bytes:0 unicastpkts:0 brcastpkts:0
  HWTX bytes:0 unicastpkts:0 brcastpkts:0 droppkts:0
5 cip0      Type:ATM
  inet addr:172.16.0.5 Bcast: 127.16.0.255 Mask:255.255.255.0
  UP RUNNING MTU:1500 ReasmMAX:65535 Group:0
  IPRX bytes:0 unicastpkts:0 brcastpkts:0
  IPTX bytes:0 unicastpkts:0 brcastpkts:0 droppkts:0
  HWRX bytes:0 unicastpkts:0 brcastpkts:0
  HWTX bytes:0 unicastpkts:0 brcastpkts:0 droppkts:0
0 loop      Type:0
  inet addr:127.0.0.1 Bcast:127.255.255.255 Mask:255.0.0.0
  UP RUNNING MTU:1500 ReasmMAX:65535 Group:1
  IPRX bytes:116 unicastpkts:0 brcastpkts:2
  IPTX bytes:0 unicastpkts:0 brcastpkts:0 droppkts:0
  HWRX bytes:0 unicastpkts:0 brcastpkts:0
  HWTX bytes:0 unicastpkts:0 brcastpkts:0 droppkts:0
=>nat enable addr=172.16.0.5 type=pat
=>nat create protocol=tcp inside_addr=10.0.0.1 inside_port=80 outside_addr=172.16.0.5
  outside_port=1080
=>nat list
Indx Prot Inside-addr:PortOutside-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
=>

```

RELATED COMMANDS:

nat delete	Delete a static NAT entry.
nat disable	Disable NAT on the specified IP address.
nat enable	Enable NAT on one of the devices own IP addresses.
nat list	List NAT connection database.

nat defserver

Define the default server behind the **SpeedTouch™ 570** NAPT device that receives all (unknown) incoming packets.

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

SYNTAX:

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

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

EXAMPLEINPUT/OUTPUT:

<pre>=>nat defserver Default server is undefined =>nat defserver addr=10.0.0.1 =>nat defserver Default server is 10.0.0.1 =></pre>
--

nat delete

Delete a static NATP entry.

SYNTAX:

```
nat delete          protocol = <IP protocol name or number>
                    inside_addr = <ip-address>
                    [inside_port = <TCP/UDP service name or port number>]
                    outside_addr = <ip-address>
                    [outside_port = <TCP/UDP service name or port number>]
```

<i>protocol</i>	The IP protocol name (or number) of the NAT entry.	REQUIRED
<i>inside_addr</i>	The IP address of the NAT entry.	REQUIRED
<i>[inside_port]</i>	The port number of the NAT entry.	OPTIONAL
<i>outside_addr</i>	The apparent host IP address of the NAT entry.	REQUIRED
<i>[outside_port]</i>	The apparent port number of the NAT entry.	OPTIONAL

EXAMPLE:

```
=>nat list
Indx Prot Inside-addr:PortOutside-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 delete protocol=tcp inside_addr=10.0.0.138 inside_port=80 outside_addr=172.16.0.5
outside_port 1080
=>nat list
Indx Prot Inside-addr:PortOutside-addr:Port Foreign-addr:Port Flgs Expir State Control
1 17 10.0.0.138:138 10.0.0.140:138 10.0.0.20:138 11 20 10
2 17 10.0.0.138:137 10.0.0.140:137 10.0.0.254:137 11 20 10
3 17 10.0.0.138:7938 10.0.0.140:7938 10.0.0.96:4756 11 20 10
4 17 10.0.0.138:513 10.0.0.140:513 10.0.0.109:513 11 20 10
5 17 10.0.0.138:111 10.0.0.140:111 10.0.0.96:4756 11 20 10
=>
```

RELATED COMMANDS:

- nat create** Create a static NATP entry.
- nat disable** Disable NATP on one of the **SpeedTouch™ 570** IP addresses.
- nat enable** Enable NATP on one of the **SpeedTouch™ 570** IP addresses.
- nat list** List NATP connection database.

nat disable

Disable NAT on a **SpeedTouch™ 570** IP address.

SYNTAX:

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

<i>addr</i>	One of SpeedTouch™ 570 's IP addresses one which NAT is enabled.	REQUIRED
-------------	---	----------

EXAMPLE:

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

RELATED COMMANDS:

nat create	Create a static NAT entry.
nat delete	Delete a static NAT entry.
nat enable	Enable NAT on one of the SpeedTouch™ 570 IP addresses.
nat list	List NAT connection database.

nat enable

Enable NAT on a **SpeedTouch™ 570** IP address.

SYNTAX:

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

<i>addr</i>	The SpeedTouch™ 570 IP address on which NAT must be applied.	REQUIRED
<i>[type]</i>	Enable port translation (pat) or not (none).	OPTIONAL

EXAMPLE:

```
=>ip aplist
1 eth0 Type:EthernetHWaddr 00:80:9f:24:ab:cf BRHWaddr ff:ff:ff:ff:ff:ff
  inet addr:10.10.10.147 Bcast: 10.10.10.255 Mask:255.0.0.0
  UP RUNNING MTU:1500 ReasmMAX:65535 Group:2
  IPRX bytes:19791886 unicastpkts:11341 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
=>nat enable addr=10.10.10.147 type=pat
=>ip aplist
1 eth0 Type:EthernetHWaddr 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:

nat create	Create a static NAT entry.
nat delete	Delete a static NAT entry.
nat disable	Disable NAT on one of the SpeedTouch™ 570 IP addresses.
nat list	List NAT connection database.

nat flush

Flush complete NAPT configuration.

The flush command does not impact previously saved configurations.

SYNTAX:

```
nat flush
```

EXAMPLE:

```

=>nat list
Indx Prot Inside-addr:PortOutside-addr:Port Foreign-addr:Port Flgs Expir StateControl
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
=>nat bindlist
Application Proto Port
RAUDIO(PNA) tcp 7071
H323 tcp 1720
FTP tcp 21
RTSP tcp 554
IRC tcp 6667
RAUDIO(PNA) tcp 7070
=>nat flush
=>nat list
Indx Prot Inside-addr:PortOutside-addr:Port Foreign-addr:Port Flgs Expir StateControl
=>nat bindlist
=>

```

RELATED COMMANDS:

nat load

Load saved or default NAPT configuration.

nat save

Save current NAPT configuration.

nat list

Show NAPT connection database.

SYNTAX:

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

<i>[addr]</i>	The SpeedTouch™ 570 IP address for which the NAPT connection database must be shown. In case the parameter is not specified the NAPT connection database for all IP addresses is shown.	OPTIONAL
---------------	---	----------

EXAMPLEINPUT/OUTPUT:

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

RELATED COMMANDS:

nat create	Create a static NAPT entry.
nat delete	Delete a static NAPT entry.
nat disable	Disable NAPT on one of the SpeedTouch™ 570 IP addresses.
nat enable	Enable NAPT on one of the SpeedTouch™ 570 IP addresses.

nat load

Load saved (or default) NAT configuration.

Execute **nat flush** prior to **nat load**.

SYNTAX:

```
nat load [defaults = <yes|no>]
```

[defaults] Load factory defaults (yes) or saved configuration (no). OPTIONAL
Not specifying this parameter loads the saved configuration

EXAMPLE:

```

=>nat list
Indx Prot Inside-addr:PortOutside-addr:Port Foreign-addr:Port Flgs Expir StateControl
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
=>nat save
=>nat flush
=>nat list
Indx Prot Inside-addr:PortOutside-addr:Port Foreign-addr:Port Flgs Expir StateControl
=>nat load
=>nat list
Indx Prot Inside-addr:PortOutside-addr:Port Foreign-addr:Port Flgs Expir StateControl
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 flush Flush complete NAT configuration.
nat save Save current NAT configuration.

nat save

Save current NAPT configuration.

SYNTAX:

```
nat save
```

EXAMPLE:

```
=>nat list
Indx Prot Inside-addr:PortOutside-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
=>nat save
=>nat flush
=>nat list
Indx Prot Inside-addr:PortOutside-addr:Port Foreign-addr:Port Flgs Expir State Control
=>nat load
=>nat list
Indx Prot Inside-addr:PortOutside-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 flush

Flush complete NAPT configuration.

nat load

Load saved or default NAPT configuration.

nat unbind

Delete an existing helper/port binding.

SYNTAX:

nat unbind	application = <string> port = <TCP/UDP service name or port number>
-------------------	--

<i>application</i>	The name of a NAT application helper. The name must be spelled exactly as listed in the application list (nat applist).	REQUIRED
<i>port</i>	The port number this application handler should work on.	REQUIRED

EXAMPLE:

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

RELATED COMMANDS:

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

11 Phonebook Commands

phonebook (to access the Phonebook level)

phonebook add

phonebook autolist

phonebook delete

phonebook flush

phonebook list

phonebook load

phonebook save

phonebook add

Add a phonebook entry.

The number of entries is limited to 64. The number of active connections is limited to 12, but more may be configured at the same time.

SYNTAX:

phonebook add	name = <string> addr = <[port.]vpi.vci> type = <{any bridge ppp cip ans pptp}>
----------------------	---

<i>name</i>	<p>A free to choose phonebook name for the destination. REQUIRED</p> <p>Two limitations apply:</p> <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, e.g. 'DHCP_Spoof01'.
<i>addr</i>	<p>The ATM address for this destination. REQUIRED</p> <p>It is composed of a Virtual Path Identifier (VPI) and a Virtual Channel Identifier (VCI) identifying ATM virtual channels.</p> <p>In most cases the values are provided by the Service Provider.</p> <p>Accepted VPI: a number between 0 and 15 Accepted VCI: a number between 0 and 511.</p>
<i>type</i>	<p>The Connection Service supported by the destination. REQUIRED</p> <p>Choose between:</p> <ul style="list-style-type: none"> ▪ any (All Packet Services) ▪ bridge (Bridging, Routed Ethernet, Bridged PPPoE, Routed PPPoE) ▪ ppp (Routed PPPoA and Relayed PPPoA) ▪ cip (Classical IP & IP Routing) ▪ ans (ATM Name Service) ▪ pptp (Relayed PPPoA, PPPoA-to-PPTP Relaying).

EXAMPLE:

```
=>phonebook list
Name      Type    Use  Address
PVC1      any     1    8.35
PVC2      bridge 0    8.36
Br3       bridge 0    8.36
Br4       bridge 0    8.38
CIPPVC3   cip     1    8.82
CIPPVC4   cip     1    8.83
=>phonebook add name=Alcatel addr=8.68 type=ppp
=>phonebook list
Name      Type    Use  Address
PVC1      any     1    8.35
PVC2      bridge 0    8.36
Br3       bridge 0    8.36
Br4       bridge 0    8.38
CIPPVC3   cip     1    8.82
CIPPVC4   cip     1    8.83
Alcatel   ppp     0    8.68
=>
```

RELATED COMMANDS:

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

phonebook autolist

Show auto PVCs, if supported by the Central Office DSLAM. (Only applicable for Alcatel ASAM DSLAMs).

SYNTAX:

```
phonebook autolist
```

EXAMPLEINPUT/OUTPUT:

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

RELATED COMMANDS:

phonebook list Show current phonebook.

phonebook delete

Remove an unused phonebook entry.

SYNTAX:

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

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

EXAMPLE:

```
=>phonebook list
Name      Type      Use      Address
PVC1      any        1        8.35
PVC2      bridge    0        8.36
Br3       bridge    0        8.36
Br4       bridge    0        8.38
CIPPVC3   cip        1        8.82
CIPPVC4   cip        1        8.83
Alcatel   ppp        0        8.68
=>phonebook delete name=Alcatel
=>phonebook list
Name      Type      Use      Address
PVC1      any        1        8.35
PVC2      bridge    0        8.36
Br3       bridge    0        8.36
Br4       bridge    0        8.38
CIPPVC3   cip        1        8.82
CIPPVC4   cip        1        8.83
=>
```

RELATED COMMANDS:

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

phonebook flush

Flush complete phonebook.

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      bridge   0        8.36
Br3       bridge   0        8.36
Br4       bridge   0        8.38
CIPPVC3   cip       1        8.82
CIPPVC4   cip       1        8.83
Alcatel   ppp       0        8.68
=>phonebook flush
=>phonebook list
Name      Type      Use      Address
=>
```

RELATED COMMANDS:

phonebook load

Load saved or default phonebook.

phonebook save

Save current phonebook.

phonebook list

Show current phonebook.

SYNTAX:

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

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

EXAMPLEINPUT/OUTPUT:

```

=>phonebook list
Name      Type    Use  Address
PVC1      any     1    8.35
PVC2      bridge 0    8.36
Br3       bridge 0    8.36
Br4       bridge 0    8.38
CIPPVC3   cip     1    8.82
CIPPVC4   cip     1    8.83
Alcatel   ppp     0    8.68
=>

```

RELATED COMMANDS:

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

phonebook load

Load saved (or default) phonebook.

Execute **phonebook flush** prior to **phonebook load**.

SYNTAX:

```
phonebook load [defaults <yes|no>]
```

[defaults]

Load factory defaults (yes) or saved configuration (no).
Not specifying this parameter loads the saved configuration

OPTIONAL

EXAMPLE:

```
=>phonebook list
Name      Type      Use      Address
PVC2      bridge   0        8.36
Br4       bridge   0        8.38
CIPPVC4   cip      1        8.83
Alcatel   ppp      1        8.68
=>phonebook save
=>phonebook flush
=>phonebook list
Name      Type      Use      Address
=>phonebook load defaults=yes
=>phonebook list
Name      Type      Use      Address
Br1       bridge   0        8.35
Br2       bridge   0        8.36
Br3       bridge   0        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
DIAL_PPP1 ppp      0        8.64
DIAL_PPP2 ppp      0        8.65
DIAL_PPP3 ppp      0        8.66
DHCP_SPOOF ppp      0        8.67
CIPPVC1   cip      0        8.80
CIPPVC2   cip      0        8.81
CIPPVC3   cip      1        8.82
CIPPVC4   cip      1        8.83
=>phonebook load defaults=no
=>phonebook list
Name      Type      Use      Address
PVC2      bridge   0        8.36
Br4       bridge   0        8.38
CIPPVC4   cip      1        8.83
Alcatel   ppp      0        8.68
=>phonebook save
```

RELATED COMMANDS:

phonebook flush Flush complete phonebook.
phonebook save Save current phonebook.

phonebook save

Save current phonebook.

SYNTAX:

```
phonebook save
```

EXAMPLE:

```
=>phonebook list
Name      Type      Use  Address
PVC2      bridge   0    8.36
Br4       bridge   0    8.38
CIPPVC4   cip       1    8.83
Alcatel   ppp       1    8.68
=>phonebook save
=>phonebook flush
=>phonebook list
Name      Type      Use  Address
=>phonebook load
=>phonebook list
Name      Type      Use  Address
PVC2      bridge   0    8.36
Br4       bridge   0    8.38
CIPPVC4   cip       1    8.83
Alcatel   ppp       0    8.68
=>phonebook save
```

RELATED COMMANDS:

phonebook flush
phonebook load

Flush complete phonebook.
Load saved or default phonebook.

12 PPP Commands

ppp (to access the PPP level)
ppp flush
ppp ifadd
ppp ifattach
ppp ifconfig
ppp ifdelete
ppp ifdetach
ppp iflist
ppp load
ppp rtadd
ppp rtdelete
ppp save

ppp flush

Flush complete PPP configuration. The flush command does not impact previously saved configurations.

SYNTAX:

```
ppp flush
```

EXAMPLE:

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

RELATED COMMANDS:

ppp load

Load saved or default PPP configuration.

ppp save

Save current PPP configuration.

ppp ifadd

Create a new PPP interface.

SYNTAX:

ppp ifadd	[intf = <string>] [dest = <phonebook entry>] [encaps = <{vcmux llc}>] [speed = <number{4800–1000000}>]
------------------	---

<i>[intf]</i>	The name for the new PPP interface. If not specified, the destination parameter must be specified. In this case the name of the destination will double as interface name.	OPTIONAL
<i>[dest]</i>	The destination for the new PPP interface. Typically, an phonebook entry.	OPTIONAL
<i>[encaps]</i>	The type of encapsulation to be used for this PPP interface. Choose between: <ul style="list-style-type: none"> ▪ vcmux ▪ llc/snap 	OPTIONAL
<i>[speed]</i>	A number between 4800 and 10000000 (bits per second). Represents the speed of the peer-to-peer connection. Use for backward compatibility. Use Quality Of Service instead.	OPTIONAL

EXAMPLE:

```

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

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

```

RELATED COMMANDS:

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

ppp ifattach

Attach (i.e. connect) a PPP interface.

SYNTAX:

ppp ifattach	intf = <ifname>	
<i>intf</i>	The name of the PPP interface to attach.	REQUIRED

EXAMPLE:

```
=>ppp iflist
PPP1:  dest : PPP1
      Retry: 10  QoS default encaps LLC
      mode = IP Routing
      flags= echo magicaccomp restart mru  addr routesavepwd PPPOE
      transaddr = pat  mru = 1492
      route=      0.0.0.0/0 -      0.0.0.0/0 (metric 0)
      user name = guest  password= *****
      adminstate= down  oper state= down  link state= not-connected
      LCP  : state= initial retransm= 10 term.reason =
      IPCP : state= initial retransm= 0 term.reason =
=>ppp ifattach =intf=PPP1
=>ppp iflist
PPP1:  dest : PPP1
      Retry: 10  QoS default encaps LLC
      mode = IP Routing
      flags= echo magicaccomp restart mru  addr routesavepwd PPPOA
      transaddr = pat  mru = 1492
      route=      0.0.0.0/0 -      0.0.0.0/0 (metric 0)
      user name = guest  password= *****
      adminstate= up    oper state= down  link state= connected
      LCP  : state= reqsent retransm= 10 term.reason =
      IPCP : state= initial retransm= 10 term.reason =
=>ppp iflist
PPP1:  dest : PPP1
      Retry: 10  QoS default encaps LLC
      mode = IP Routing
      flags= echo magicaccomp restart mru  addr routesavepwd PPPOA
      transaddr = pat  mru = 1492
      route=      0.0.0.0/0 -      0.0.0.0/0 (metric 0)
      user name = guest  password= *****
      adminstate= up    oper state= up    link state= connected
      LCP  : state= opened retransm= 0 term.reason =
      IPCP : state= opened retransm= 10 term.reason =
      acname : , service = .
=>
```

RELATED COMMANDS:

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

ppp ifconfig

Configure a PPP interface. As the PPP interface to be configured may not be connected at the time of configuration, execute **ppp ifdetach** prior to executing the **ppp ifconfig** command.

SYNTAX:

ppp ifconfig	<pre> intf = <ifname> [dest = <phonebook entry>] [user = <string>] [password = <string>] [qos = <string>] [proto = <{pppoe pppoe}>] [acname = <string>] [servicename = <string>] [encaps = <{vcmux llc}>] [pcomp = <{off on}>] [accomp = <{on off negotiate}>] [trace = <{off on}>] [pap = <{off on}>] [restart = <{off on}>] [retryinterval = <number{0-65535}>] [passive = <{off on}>] [silent = <{off on}>] [echo = <{off on}>] [mru = <number{293-8192}>] [laddr = <ip-address>] [raddr = <ip-address>] [savepwd = <{off on}>] [demanddial = <{off on}>] [primdns = <ip-address>] [secdns = <ip-address>] [idle = <number{0-1000000}>] [addrtrans = <{none pat}>] [unnumbered = <{off on}>] [poolstart = <ip-address>] [poolend = <ip-address>] [status = <{down up}>] </pre>
---------------------	--

<i>intf</i>	The name of the PPP interface to configure.	REQUIRED
<i>[dest]</i>	The destination for this PPP interface. Typically, a phonebook entry. Use: <ul style="list-style-type: none"> ▪ PPPoA (ppp) phonebook entries For the Routed PPPoA (PPP & IP Routing) packet service. ▪ ETHoA (bridge) phonebook entries For the Routed PPPoE packet service. 	OPTIONAL
<i>[user]</i>	The user name for remote PAP/CHAP authentication.	OPTIONAL
<i>[password]</i>	The password for remote PAP/CHAP authentication.	OPTIONAL

[qos]	The name of a configured Quality Of Service book entry. This parameter never needs to be specified.	OPTIONAL
[proto]	The encapsulation method for the PPP frames, i.e. the applicable packet service for the connection. Select: <ul style="list-style-type: none"> ▪ pppoa For a Routed PPPoA (PPP & IP Routing) connections. ▪ pppoe For a Routed PPPoE connection. Per default the PPPoA protocol applies.	OPTIONAL
[acname]	The Access Concentrator name for a Routed PPPoE connection. This parameter is applicable only for Routed PPPoE PPP interfaces (proto=pppoe). Execute the ppp ifscan command to see the names of available access concentrators, if any.	OPTIONAL
[servicename]	The Service Name for a Routed PPPoE connection. This parameter is applicable only for Routed PPPoE PPP interfaces (proto=pppoe). Execute the ppp ifscan command to see the available service names, if any.	OPTIONAL
[encaps]	The type of encapsulation to be used for this PPP interface. Choose between: <ul style="list-style-type: none"> ▪ vcmux (default) Standard encapsulation method for PPPoA (ppp) frames. ▪ llc Standard encapsulation method for ETHoA (bridge) frames. 	OPTIONAL
[pcomp]	Try (on) or do not try (off) to negotiate PPP protocol compression (LCP PCOMP). Per default the negotiation is disabled (off).	OPTIONAL
[accomp]	Try (on), do never try (off) or negotiate (negotiate) to negotiate PPP address & control field compression (LCP ACCOMP). In the very most cases LCP ACCOMP should not be disabled nor negotiated, i.e. the address field FF-03 should not be sent over ATM. Therefore by default this parameter is enabled (on). In case the accomp parameter is set 'negotiate' the local side of the PPP connection demands to do ACCOMP and adapts itself to the result of this negotiation.	OPTIONAL
[trace]	Enable (on) or disable (off) verbose console logging. By default tracing is disabled (off).	OPTIONAL
[pap]	Force PAP based authentication (on) or use CHAP based authentication, if available (off). For security reasons PAP negotiation is disabled (off) per default.	OPTIONAL
[restart]	Automatically restart the connection when LCP link goes down (on) or do not restart automatically (off). By default restart is disabled (off).	OPTIONAL

<i>[retryinterval]</i>	A number between 0 and 65535 (seconds). Represents the intermediate interval between two retries to establish the connection on ATM level.. Only applicable in an SVC environment.	OPTIONAL
<i>[passive]</i>	Put the link in listening state in case LCP times out (on) or not (off). This parameter allows to determine whether the link should be left open to wait for incoming messages from the remote side after 10 unsuccessful tries to establish the connection or not. Per default the listening state is disabled.	OPTIONAL
<i>[silent]</i>	Do not send anything at startup and just listen for incoming LCP messages (on) or retry up to 10 times to establish the connection (off). Per default the silent state is disabled.	OPTIONAL
<i>[echo]</i>	Send LCP echo requests at regular intervals (on) or not (off). Per default the sending of LCP echo requests is enabled.	OPTIONAL
<i>[mru]</i>	A number between 293 and 8192. Represents the maximum packet size the SpeedTouch™ 570 should negotiate to be able to receive.	OPTIONAL
<i>[laddr]</i>	The local IP address of the peer-to-peer connection. Specifying a local IP address forces the remote side of the PPP link (if it allows to) to accept this IP address as the SpeedTouch™ 570 PPP session IP address. If not specified, the SpeedTouch™ 570 will accept any IP address. Typically the local IP address parameter is not specified.	OPTIONAL
<i>[raddr]</i>	The remote IP address of the peer-to-peer connection. Specifying a remote IP address forces the remote side of the PPP link (if it allows to) to accept this IP address as its PPP session IP address. If not specified, the SpeedTouch™ 570 will accept any IP address. Typically the remote IP address parameter is not specified.	OPTIONAL
<i>[savepwd]</i>	Save password (on), if supplied, or do not save the password (off). Per default the saving of the password is disabled.	OPTIONAL
<i>[demanddial]</i>	Enable (on) or disable (off) the dial-on-demand feature.	OPTIONAL
<i>[primdns]</i>	The IP address of the primary DNS server. In case a primary DNS server is specified the SpeedTouch™ 570 will negotiate this IP address with the remote side. If not specified, the SpeedTouch™ 570 will accept any IP address.	OPTIONAL
<i>[secdns]</i>	The IP address of the (optional) secondary DNS server. In case a secondary DNS server is specified the SpeedTouch™ 570 will negotiate this IP address with the remote side. If not specified, the SpeedTouch™ 570 will accept any IP address.	OPTIONAL
<i>[idle]</i>	A number between 1 and 1000000 (seconds). Represents after how many seconds an idle link goes down.	OPTIONAL

<code>[addrtrans]</code>	Automatically enable address translation for the IP address of this link (<code>pat</code>) or do not use address translation (<code>none</code>).	OPTIONAL
<code>[unnumbered]</code>	Takes the local IP address from 'laddr' field and remote IP address from the IP address pool assigned to the incoming PPP link. In case the unnumbered parameter is disabled the same IP address is used for each connection on the server side, thus reducing the number of IP addresses used.	OPTIONAL
<code>[poolstart]</code>	The lower bound of the IP address pool assigned to the incoming PPP link.	OPTIONAL
<code>[poolend]</code>	The upper bound of the IP address pool assigned to the incoming PPP link.	OPTIONAL
<code>[status]</code>	Force automatically to attach the PPP interface (<code>up</code>) or use the regular ppp ifattach command (<code>down</code>). Per default the startup status is down (recommended).	OPTIONAL

EXAMPLE:

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

RELATED COMMANDS:

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

ppp ifdelete

Delete a PPP interface.

SYNTAX:

ppp ifdelete	intf = <ifname>
---------------------	------------------------------

intf The name of the PPP interface to delete.

EXAMPLE:

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

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

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

=>
```

RELATED COMMANDS:

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

ppp ifdetach

Detach a PPP interface.

SYNTAX:

ppp ifdetach	intf = <ifname>	
<i>intf</i>	The name of the PPP interface.	REQUIRED

EXAMPLE:

```
=>ppp iflist
PPP1:  dest : PPP1
      Retry: 10  QoS default encaps LLC
      mode = IP Routing
      flags= echo magicaccomp restart mru  addr routesavepwd PPPOA
      transaddr = pat      mru = 1492
      route=      0.0.0.0/0 -      0.0.0.0/0 (metric 0)
      user name = guest  password= *****
      adminstate= up      oper state= up      link state= connected
      LCP  : state= opened retransm= 0 term.reason =
      IPCP : state= opened retransm= 10 term.reason =
      acname : , service = .
=>ppp ifdetach =intf=PPP1
=>ppp iflist
PPP1:  dest : PPP1
      Retry: 10  QoS default encaps LLC
      mode = IP Routing
      flags= echo magicaccomp restart mru  addr routesavepwd PPPOE
      transaddr = pat      mru = 1492
      route=      0.0.0.0/0 -      0.0.0.0/0 (metric 0)
      user name = guest  password= *****
      adminstate= down    oper state= down    link state= not-connected
      LCP  : state= initial retransm= 10 term.reason =
      IPCP : state= initial retransm= 0 term.reason =
=>
```

RELATED COMMANDS:

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

ppp ifscan

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

Execute the **ppp ifdetach** command for this interface prior to perform a scan on it.

SYNTAX:

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

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

EXAMPLE:

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

Done !
=>
```

RELATED COMMANDS:

ppp ifconfig Configure a PPP interface.

ppp load

Load saved (or default) PPP configuration.

Execute **ppp flush** prior to **ppp load**.

SYNTAX:

```
ppp load [defaults = <yes|no>]
```

<i>[defaults]</i>	Load factory defaults (yes) or saved configuration (no). Not specifying this parameter loads the saved configuration	OPTIONAL
-------------------	---	----------

EXAMPLE:

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

RELATED COMMANDS:

ppp flush	Flush complete PPP configuration.
ppp save	Save current PPP configuration.

ppp rtadd

Automatically add a route configuration to the routing table in case the specified PPP 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.

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

SYNTAX:

ppp rtadd	intf = <ifname> dst = <ip-address> [dstmsk = <ip-mask(dotted or cidr)>] [src = <ip-address>] [srcmsk = <ip-mask(dotted or cidr)>] [metric = <number{0-100}>]	
<i>intf</i>	The name of the PPP interface.	REQUIRED
<i>dst</i>	The destination IP address specification for the route to be added when the link comes up.	REQUIRED
<i>[dstmsk]</i>	The destination IP mask. Depending on the destination netmask: <ul style="list-style-type: none"> ▪ Any remote destination is reachable, i.e. the PPP connection acts as default route (dstmsk=0) ▪ Only the remote (sub)net is reachable (dstmsk=1) The actual destination mask will be the default netmask applicable for destination IP address ▪ Only the single remote host is reachable (dstmsk=32) ▪ Any valid (contiguous) netmask in case of VLSM. 	OPTIONAL
<i>[src]</i>	The source IP address specification for the route to be added when the link comes up.	OPTIONAL
<i>[srcmsk]</i>	The source IP mask. Depending on the source netmask: <ul style="list-style-type: none"> ▪ Everybody is allowed to use this PPP connection (dstmsk=0) ▪ Only members of the same subnet as the host which opened the PPP connection are allowed to use the PPP connection (dstmsk=1) The actual destination mask will be the netmask applicable for the IP address of the host which opened the PPP connection. ▪ Only the host which opened the PPP connection is allowed to use the PPP connection. (dstmsk=32) ▪ Any valid (contiguous) netmask in case of VLSM. 	OPTIONAL
<i>[metric]</i>	The route metric, i. e. the cost factor of the route. Practically, the cost is determined by the hop count. It is recommended not to use this parameter.	OPTIONAL

EXAMPLE:

```

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

```

RELATED COMMANDS:

ppp rtdelete

Delete the route specification for an upcoming PPP link.

ppp rtdelete

Delete the route specification for a PPP link.

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

SYNTAX:

ppp rtdelete	intf = <ifname>	
<i>intf</i>	The PPP interface name for which to delete the route settings.	REQUIRED

EXAMPLE:

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

RELATED COMMANDS:

ppp rtadd Configure a route specification for an upcoming PPP link.

ppp save

Save current PPP configuration.

SYNTAX:

```
ppp save
```

EXAMPLE:

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

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

=>
```

RELATED COMMANDS:

ppp flush
ppp load

Flush complete PPP configuration.
Load saved or default PPP configuration.

13 PPTP Commands

pptp (to access the PPTP level)

pptp flush

pptp list

pptp load

pptp profadd

pptp profdelete

pptp proflist

pptp save

pptp flush

Flush complete PPTP configuration.

The flush command does not impact previously saved configurations.

SYNTAX:

```
pptp flush
```

EXAMPLE:

```
=>pptp profadd name=Relay PPP1 encaps=nlpid ac=always
=>pptp proflist
Profile           QoS           Encaps         AC
Relay PPP1       default      nlpid          always
=>pptp flush
=>pptp proflist
=>
```

RELATED COMMANDS:

pptp load

Load saved or default PPTP configuration.

pptp save

Save current PPTP configuration.

pptp list

Show current PPTP configuration.

SYNTAX:

```
pptp list
```

EXAMPLEINPUT/OUTPUT:

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

pptp load

Load saved (or default) PPTP configuration.

Execute **pptp flush** prior to **pptp load**.

SYNTAX:

```
pptp load [defaults = <yes|no>]
```

<i>[defaults]</i>	Load factory defaults (yes) or saved configuration (no). Not specifying this parameter loads the saved configuration	OPTIONAL
-------------------	---	----------

EXAMPLE:

```
=>pptp proflist
Profile           QoS           Encaps         AC
Relay_PPP1       default      nlpid          always
PPTPLink         default      vcmux          never
=>pptp save
=>pptp flush
=>pptp proflist
=>pptp load defaults=yes
Profile           QoS           Encaps         AC
default          default      vcmux          never
=>pptp load defaults=no
=>pptp proflist
Profile           QoS           Encaps         AC
Relay_PPP1       default      nlpid          always
PPTPLink         default      vcmux          never
=>
```

RELATED COMMANDS:

pptp flush	Flush complete PPTP configuration.
pptp save	Save current PPTP configuration.

pptp profadd

Define a new PPTP profile.

SYNTAX:

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

<i>name</i>	The name for the PPTP profile.	REQUIRED
<i>[qos]</i>	The name of the Quality Of Service book entry. This parameter never needs to be specified.	OPTIONAL
<i>[encaps]</i>	The type of encapsulation applicable to Relayed PPPoA interfaces using this PPTP profile. Choose between: <ul style="list-style-type: none"> ▪ vcmux ▪ nlpid 	OPTIONAL
<i>[ac]</i>	The HDLC framing option applicable to Relayed PPPoA interfaces using this PPTP profile. Before relaying the encapsulated PPP frames over the PPPoA link, make sure that the address and control field (0xFF03) is always in front of the frames (always), make sure the address and control field will never be found in front of the frames (never) or do not change the frames arriving via the PPTP tunnel (keep). By default the address and control field is never sent (compliant to RFC2364). It is recommended to keep this setting.	OPTIONAL

EXAMPLE:

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

RELATED COMMANDS:

pptp profdelete	Delete a PPTP profile.
pptp proflist	Show current PPTP profiles.

pptp profdelete

Delete a PPTP profile.

SYNTAX:

pptp profdelete	name <string>
------------------------	----------------------

<i>name</i>	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:

pptp profadd	Define a new PPTP profile.
pptp proflist	Show current PPTP profiles.

pptp proflist

Show all current PPTP profiles.

```
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.

pptp save

Save current PPTP configuration.

SYNTAX:

```
pptp save
```

EXAMPLE:

```
=>pptp proflist
Profile           QoS           Encaps         AC
Relay_PPP1       default      nlpid         always
PPTPLink         default      vcmux         never
=>pptp save
=>pptp flush
=>pptp proflist
=>pptp load defaults=no
=>pptp proflist
Profile           QoS           Encaps         AC
Relay_PPP1       default      nlpid         always
PPTPLink         default      vcmux         never
=>
```

RELATED COMMANDS:

pptp flush

Flush complete PPTP configuration.

pptp load

Load saved or default PPTP configuration.

14 Software Commands

software (to access the Software level)

software cleanup

software deletepassive

software setpassive

software switch

software version

software cleanup

Remove all unused files from the passive software subdirectory.

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

SYNTAX:

```
software cleanup
```

EXAMPLE:

```
=>software cleanup  
=>
```

RELATED COMMANDS:

software deletepassive
software setpassive

Delete the passive software.

Mark an uploaded file as passive software version.

software deletepassive

Delete passive software.

SYNTAX:

```
software deletepassive
```

EXAMPLE:

```
=>Software version
Active : Saschal.004           Passive : Benel.003
=>software deletepassive
=>Software version
Active : Saschal.004           Passive : _____
=>
```

RELATED COMMANDS:

software cleanup

Remove all unused files from the passive software subdirectory.

software setpassive

Mark a file as passive software version.

software setpassive

Mark a file as passive software version. Only correctly uploaded software, valid for the **SpeedTouch™ 570** can be marked as passive software.

SYNTAX:

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

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

EXAMPLE:

```
=>Software version
Active : Saschal.003                      Passive : Bene1.003
=>Software deletepassive
=>Software version
Active : Saschal.003                      Passive :
.....
(FTP file transfer or upload via the SpeedTouch™ 570 pages of new software Saschal.008)
.....
=>software setpassive file=Saschal.008
=>Software version
Active : Saschal.003                      Passive : Saschal.008
=>
```

RELATED COMMANDS:

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

software switch

Switch active and passive versions and reboot the **Speed Touch™ 570**.

Because rebooting implies a flush of all non-saved configurations it is highly recommended to save the current configuration if needed, e.g. by executing the **config save** command prior to executing a software switch.

SYNTAX:

```
software switch
```

EXAMPLE:

```
=>Software version
Active : Saschal.003           Passive : Saschal.008
=>software switch
.....
(after reboot and re-opening the Telnet session)
.....
=>Software version
Active : Saschal.008           Passive : Saschal.003
=>
```

RELATED COMMANDS:

software version
system reboot

Show active and passive software versions.
Reboot the **Speed Touch™ 570**.

software version

Show active and passive software versions.

SYNTAX:

```
software version
```

EXAMPLE:

```
=>Software version  
Active   : Sascha1.008           Passive  : Sascha1.003  
=>
```

RELATED COMMANDS:

software switch

Switch active and passive software versions and reboot the **Speed Touch™ 570**.

15 System Commands

system (to access the System level)

system clearpassword

system flush

system load

system reboot

system save

system setpassword

system clearpassword

Clear current **SpeedTouch™ 570** system password.

To avoid unrestricted and unauthorized access to the **SpeedTouch™ 570** it is highly recommended always to make sure that it is protected by a **SpeedTouch™ 570** system password (by executing **system setpassword**) and to change the password regularly.

SYNTAX:

```
system clearpassword
```

EXAMPLE:

```
=>system clearpassword  
=>
```

RELATED COMMANDS:

system setpassword Set/change current system password.

system flush

Flush current **SpeedTouch™ 570** system configuration, i.e. the System password.
The flush command does not impact previously saved configurations.

To avoid unrestricted and unauthorized access to the **SpeedTouch™ 570** it is highly recommended always to make sure that it is protected by a **SpeedTouch™ 570** system password (by executing **system setpassword**) and to change the password regularly.

SYNTAX:

```
system flush
```

EXAMPLE:

```
=>system flush  
=>
```

RELATED COMMANDS:

system load
system save

Load saved or default system configuration.
Save current system configuration.

system load

Load saved (or default) system configuration.

Execute **system flush** prior to **system load**.

In most cases loading the default system configuration causes the **SpeedTouch™ 570** system password to be CLEARED.

Therefore, to avoid unrestricted and unauthorized access to the **SpeedTouch™ 570** it is highly recommended always to make sure that it is protected by a **SpeedTouch™ 570** system password (by executing **system setpassword**) and to change the password regularly.

SYNTAX:

system load	[defaults = <yes no>]
--------------------	------------------------------------

<i>[defaults]</i>	Load factory defaults (yes) or saved configuration (no). Not specifying this parameter loads the saved configuration	OPTIONAL
-------------------	---	----------

EXAMPLE:

<pre>=>system load defaults=no =></pre>

RELATED COMMANDS:

system flush

Flush complete system configuration.

system save

Save current system configuration.

system reboot

Reboot the **SpeedTouch™ 570**.

Because rebooting implies a flush of all non-saved configurations it is highly recommended to save the current configuration by executing **config save**.

To avoid unrestricted and unauthorized access to the **SpeedTouch™ 570** it is highly recommended always to make sure that it is protected by a **SpeedTouch™ 570** system password (by executing **system setpassword**) and to save it (by executing **system save**) prior to executing this command.

SYNTAX:

```
system reboot
```

EXAMPLE:

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

system save

Save current system configuration, i.e. the System password.

To avoid unrestricted and unauthorized access to the **SpeedTouch™ 570** it is highly recommended always to make sure that it is protected by a **SpeedTouch™ 570** system password (by executing **system setpassword**) and to save it (by executing **system save**) prior to executing this command.

SYNTAX:

```
system save
```

EXAMPLE:

```
=>system save  
=>
```

RELATED COMMANDS:

system load
system flush

Load saved or default system configuration.
Flush complete system configuration.

16 TD Commands

td (to access this level)
td call

td call

Call a 'Trace & Debug' command. For qualified personnel only.

SYNTAX:

<i>td call</i>	<i>cmd = <string></i>
-----------------------	------------------------------------

cmd

The quoted trace & debug command string.

REQUIRED

17 Wireless Commands

wireless (to access the Wireless level)
wireless beacon_period
wireless channel
wireless counters
wireless deletemac
wireless disablecontrol
wireless enablecontrol
wireless filtconfig
wireless flush
wireless fragment_thres
wireless listacl
wireless long_retry_lim
wireless params
wireless random
wireless reset
wireless rts_threshold
wireless short_retry_lim
wireless ssid
wireless startWEP
wireless stopWEP
wireless status
wireless wepkey

wireless beacon_period

Set the value of the beacon period.

Beacon packets contain timing information and hop patterns information to be broadcasted over the WLAN. It allows any (other) wireless station (access point and client) to synchronize their internal timer and in order to hop, i.e. change to another frequency, at the correct time

SYNTAX:

```
wireless beacon_period beacon_period = <number{1-65535}>
```

<i>beacon_period</i>	A number between 1 and 65535 (milliseconds). Indicates the duration between beacon packets, which are used by IEEE802.11b systems to synchronize the 'hops'. By default the beacon period is one-half of the dwell period, i.e. 80 milliseconds, so that two beacon packets are transmitted per hop dwell period.	REQUIRED
----------------------	---	----------

EXAMPLE:

```
=>wireless params
RTS Threshold      : 2347
Short Retry Limit  : 8
Long Retry Limit   : 4
Fragmentation Threshold : 2346
Beacon Period      : 80
=>wireless beacon_period beacon_period = 160
=>wireless params
RTS Threshold      : 2347
Short Retry Limit  : 8
Long Retry Limit   : 4
Fragmentation Threshold : 2346
Beacon Period      : 160
=>
```

RELATED COMMANDS:

- wireless params** Show **SpeedTouch™ 570** wireless communication configuration.
- wireless counters** Show **SpeedTouch™ 570** wireless communication counters.

wireless channel

Set WLAN radio channel.

SYNTAX:

```
wireless channel channel = <number{1-13}>
```

<i>channel</i>	A number between 1 and 13. Indicates the radio channel number for the WLAN which identifies the frequency on which the wireless connectivity relies. The allowed range is dependent of the regulation domain where the SpeedTouch™570 is used. By default the radio channel is 11 (2.462GHz).	REQUIRED
----------------	---	----------

EXAMPLE:

```
=>wireless status
Wireless SSID      :WLAN_Alcatel
Wireless channel   :6
Regulation domain  :ETSI
WEP encryption     :Off
Association Control :Off
=>wireless channel channel=1
=>wireless status
Wireless SSID      :WLAN_Alcatel
Wireless channel   :1
Regulation domain  :ETSI
WEP encryption     :Off
Association Control :Off
=>
```

RELATED COMMANDS:

wireless status Show **SpeedTouch™ 570** WLAN configuration.

wireless counters

Show some wireless communication counters.

SYNTAX:

```
wireless counters
```

EXAMPLE:

```
=>wireless counters
Transmitted Frag Count : 22954
Received Fragment Count : 23585
WEP Undecryptable Count : 0
WEPICV Error Count : 0
WEP Excluded Count : 0
Failed Count : 20
FCS Error Count : 129
RTS Success Count : 0
RTS Failure Count : 0
ACK Failure Count : 2152
=>
```

RELATED COMMANDS:

wireless params

Show **SpeedTouch™ 570** wireless communication configuration.

wireless deletemac

Delete a WLAN client in the access control list.

This command only applies in case association control is enabled.

SYNTAX:

```
wireless deletemac deletemac = <string>
```

<code>deletemac</code>	The MAC address as listed in the access control list of the client to delete.	REQUIRED
------------------------	---	----------

EXAMPLE:

```
=>wireless listacl
   MAC address      Authorization
01:23:45:67:89:AB   true
02:02:03:04:05:06   true
=>wireless deletemac deletemac=01:23:45:67:89:AB
=>wireless listacl
   MAC address      Authorization
02:02:03:04:05:06   true
=>
```

RELATED COMMANDS:

wireless listacl	Show access control list.
wireless enablecontrol	Enable association control.
wireless disablecontrol	Disable association control.
wireless filtconfig	Show WLAN client configuration.
wireless flush	Flush complete access control list.

wireless disablecontrol

Disable association control.

SYNTAX:

```
wireless disablecontrol
```

EXAMPLE:

```
=>wireless status
Wireless SSID      :WLAN_Alcatel
Wireless channel   :1
Regulation domain  :ETSI
WEP encryption     :Off
Association Control :On
=>wireless disablecontrol
The Association Control is Off
=>wireless status
Wireless SSID      :WLAN_Alcatel
Wireless channel   :1
Regulation domain  :ETSI
WEP encryption     :Off
Association Control :Off
=>
```

RELATED COMMANDS:

wireless listacl	Show access control list.
wireless enablecontrol	Enable association control.
wireless deletemac	Delete a WLAN client in the access control list.
wireless filtconfig	Show WLAN client configuration.
wireless flush	Flush complete access control list.
wireless status	Show SpeedTouch™ 570 WLAN configuration.

wireless enablecontrol

Enable association control.

SYNTAX:

```
wireless enablecontrol
```

EXAMPLE:

```
=>wireless status
Wireless SSID      :WLAN_Alcatel
Wireless channel   :1
Regulation domain  :ETSI
WEP encryption     :Off
Association Control :Off
=>wireless enablecontrol
The Association Control is On
=>wireless status
Wireless SSID      :WLAN_Alcatel
Wireless channel   :1
Regulation domain  :ETSI
WEP encryption     :Off
Association Control :On
=>
=>
```

RELATED COMMANDS:

wireless listacl	Show access control list.
wireless disablecontrol	Disable association control.
wireless filtconfig	Show WLAN client configuration.
wireless deletemac	Delete a WLAN client in the access control list.
wireless flush	Flush complete access control list.
wireless status	Show SpeedTouch™ 570 WLAN configuration.

wireless filtconfig

Show/set association control of a WLAN client.

This command only applies in case association control is enabled.

SYNTAX:

wireless filtconfig	MACaddress = <string> [authorization = <string>]
----------------------------	---

MACaddress	The MAC address of the WLAN client.	REQUIRED
[authorization]	The client is allowed to join the SpeedTouch™ 570 WLAN (true) or will never be allowed to join (false).	OPTIONAL

EXAMPLE:

```
=>wireless listacl
    MAC address      Authorization
01:23:45:67:89:AB   true
02:02:03:04:05:06   true
=>wireless filtconfig MACaddress=01:23:45:67:89:AB authorization=false
=>wireless listacl
    MAC address      Authorization
01:23:45:67:89:AB   false
02:02:03:04:05:06   true
=>
```

RELATED COMMANDS:

wireless listacl	Show access control list.
wireless disablecontrol	Disable association control.
wireless enablecontrol	Enable association control.
wireless deletemac	Delete a WLAN client in the access control list.
wireless flush	Flush complete access control list.

wireless flush

Flush complete access control list, i.e. delete all WLAN clients from the access control list.
This command only applies in case association control is enabled.

SYNTAX:

```
wireless flush
```

EXAMPLE:

```
=>wireless listacl
   MAC address      Authorization
01:23:45:67:89:AB   true
02:02:03:04:05:06   true
=>wireless flush
=>wireless listacl
   MAC address      Authorization
=>
```

RELATED COMMANDS:

wireless listacl	Show access control list.
wireless disablecontrol	Disable association control.
wireless enablecontrol	Enable association control.
wireless deletemac	Delete a WLAN client in the access control list.

wireless fragment_thres

Set the value of the fragmentation threshold.

Fragmentation is a procedure used to subdivide a data packet into smaller packets to enable efficient use of the wireless bandwidth. This is necessary in order to allow large size packets to traverse the network with constraints on maximum packet size.

Throughput will generally be lower for fragmented packets, since the fixed packet overhead will consume a higher portion of the bandwidth.

SYNTAX:

```
wireless fragment_thres fragment_thres = <number{256-2346}>
```

<i>fragment_thres</i>	A number between 256 and 2346 (bytes). Indicates the threshold above which a packet should be fragmented before transmitting it over the WLAN. By default the fragmentation threshold is 2346 bytes.	REQUIRED
-----------------------	--	----------

EXAMPLE:

```
=>wireless params
RTS Threshold      : 2347
Short Retry Limit  : 8
Long Retry Limit   : 4
Fragmentation Threshold : 2346
Beacon Period      : 80
=>wireless fragment_thres fragment_thres = 640
=>wireless params
RTS Threshold      : 2347
Short Retry Limit  : 8
Long Retry Limit   : 4
Fragmentation Threshold : 640
Beacon Period      : 80
=>
```

RELATED COMMANDS:

wireless params	Show SpeedTouch™ 570 wireless communication configuration.
wireless counters	Show SpeedTouch™ 570 wireless communication counters.

wireless listacl

Show access control list.

When Association Control is enabled, this command shows the access control list. When it is disabled , the command shows all currently associated WLAN clients.

SYNTAX:

```
wireless listacl
```

EXAMPLE:

```
=>wireless listacl
      MAC address      Authorization
01:23:45:67:89:AB      true
02:02:03:04:05:06      true
=>
```

RELATED COMMANDS:

wireless disablecontrol	Disable association control.
wireless enablecontrol	Enable association control.
wireless deletemac	Delete a WLAN client in the access control list.
wireless flush	Flush complete access control list.

wireless long_retry_lim

Set the value of the long retry limit.

SYNTAX:

```
wireless long_retry_lim long_retry_lim = <number{1-255}>
```

<i>long_retry_lim</i>	A number between 1 and 255. Indicates the maximum number of transmission attempts that shall be made of a frame (with a length which is more than rts_threshold), in case the acknowledgment is not received in time. By default the long retry limit is 4.	REQUIRED
-----------------------	---	----------

EXAMPLE:

```
=>wireless params
RTS Threshold      : 2347
Short Retry Limit  : 8
Long Retry Limit   : 4
Fragmentation Threshold : 2346
Beacon Period     : 80
=>wireless long_retry_lim long_retry_lim = 8
=>wireless params
RTS Threshold      : 2347
Short Retry Limit  : 8
Long Retry Limit   : 8
Fragmentation Threshold : 2346
Beacon Period     : 80
=>
```

RELATED COMMANDS:

wireless params	Show SpeedTouch™ 570 wireless communication configuration.
wireless counters	Show SpeedTouch™ 570 wireless communication counters.
wireless rts_threshold	Set the RTS_Threshold.
wireless short_retry_lim	Set the short retry limit.

wireless params

Show **SpeedTouch™ 570** wireless communication configuration.

SYNTAX:

```
wireless params
```

EXAMPLE:

```
=>wireless params
RTS Threshold      : 2347
Short Retry Limit  : 8
Long Retry Limit   : 4
Fragmentation Threshold : 2346
Beacon Period      : 80
=>
```

RELATED COMMANDS:

wireless counters

Show **SpeedTouch™ 570** wireless communication counters.

wireless random

Generate a random hexadecimal value for the WEP key.

SYNTAX:

```
wireless random
```

EXAMPLE:

```
=>wireless random  
The random key = 41:e6:27:d4:7d  
=>
```

RELATED COMMANDS:

wireless startWEP

Start WEP encryption.

wireless stopWEP

Stop WEP encryption.

wireless wepkey

Set a 40-bits WEP key yourself.

wireless reset

Reset **SpeedTouch™ 570** WLAN configuration to defaults.

wireless reset

Reset **SpeedTouch™ 570** wireless parameters to default values.

SYNTAX:

```
wireless reset
```

EXAMPLE:

```
=>wireless status
Wireless SSID      :WLAN_Alcatel
Wireless channel   :6
Regulation domain  :ETSI
WEP encryption     :Off
Association Control :Off
=>wireless reset
=>wireless status
Wireless SSID      :Alcatel012345
Wireless channel   :11
Regulation domain  :ETSI
WEP encryption     :Off
Association Control :On
=>
```

RELATED COMMANDS:

wireless status

Show **SpeedTouch™ 570** WLAN configuration.

wireless rts_threshold

Set the value of the RTS Threshold.

The **SpeedTouch™ 570** wireless access point supports the Medium Reservation Mechanism using the RTS and CTS protocol, based on the length of the wireless message that is to be transmitted. When a wireless message is longer than the RTS Threshold, the **SpeedTouch™ 570** will send a Request-To-Send (RTS) message to the WLAN client and defer transmission until the WLAN client has responded with a Clear-To-Send (CTS) message. This CTS message announces all other clients they should defer transmissions until the WLAN client who sent the RTS message has finished its transmission. The **SpeedTouch™ 570** then sends the data and the client acknowledges all transmitted packets by sending a short ACK message.

SYNTAX:

```
wireless rts_threshold rts_threshold = <number{0-2347}>
```

<i>rts_threshold</i>	A number between 0 and 2437 (bytes). Indicates the RTS Threshold. Setting this parameter to a small value causes RTS messages to be sent more often, consuming more of the available bandwidth, therefore reducing the apparent throughput of other network packets. However, the more often RTS packets are sent, the quicker the system can recover from interference or collisions. By default the RTS Threshold is 2437 bytes.	REQUIRED
----------------------	---	----------

EXAMPLE:

```
=>wireless params
RTS Threshold      : 2347
Short Retry Limit  : 8
Long Retry Limit   : 4
Fragmentation Threshold : 2346
Beacon Period      : 80
=>wireless rts_threshold rts_threshold = 640
=>wireless params
RTS Threshold      : 640
Short Retry Limit  : 8
Long Retry Limit   : 8
Fragmentation Threshold : 2346
Beacon Period      : 80
=>
```

RELATED COMMANDS:

wireless params	Show SpeedTouch™ 570 wireless communication configuration.
wireless counters	Show SpeedTouch™ 570 wireless communication counters.
wireless long_retry_lim	Set the long retry limit.
wireless short_retry_lim	Set the short retry limit.

wireless short_retry_lim

Set the value of the short retry limit.

SYNTAX:

```
wireless long_retry_lim long_retry_lim = <number{1-255}>
```

<i>short_retry_lim</i>	A number between 1 and 255. Indicates the maximum number of transmission attempts that shall be made of a frame (with a length which is less or equal to <i>rts_threshold</i>), in case the acknowledgment is not received in time. By default the short retry limit is 8.	REQUIRED
------------------------	---	----------

EXAMPLE:

```
=>wireless params
TS Threshold      : 2347
Short Retry Limit : 8
Long Retry Limit  : 4
Fragmentation Threshold : 2346
Beacon Period    : 80
=>wireless short_retry_lim short_retry_lim = 16
=>wireless params
TS Threshold      : 2347
Short Retry Limit : 16
Long Retry Limit  : 4
Fragmentation Threshold : 2346
Beacon Period    : 80
=>
```

RELATED COMMANDS:

wireless params	Show SpeedTouch™ 570 wireless communication configuration.
wireless counters	Show SpeedTouch™ 570 wireless communication counters.
wireless rts_threshold	Set the RTS_Threshold.
wireless short_retry_lim	Set the short retry limit.

wireless ssid

Set **SpeedTouch™ 570** WLAN Service Set Identifier.

SYNTAX:

wireless ssid	ssid = <string>
----------------------	------------------------------

<i>ssid</i>	A string between 1 and 32 characters.	REQUIRED
-------------	---------------------------------------	----------

EXAMPLE:

```
=>wireless status
Wireless SSID      :WLAN_Alcatel
Wireless channel   :6
Regulation domain  :ETSI
WEP encryption     :Off
Association Control :Off
=>wireless ssid=WLAN_Sascha
=>wireless status
Wireless SSID      :WLAN_Sascha
Wireless channel   :6
Regulation domain  :ETSI
WEP encryption     :Off
Association Control :Off
=>
```

RELATED COMMANDS:

wireless status Show **SpeedTouch™ 570** WLAN configuration.

wireless startWEP

Start WEP encryption. Prior to this command a WEP key must have been set.

SYNTAX:

```
wireless startWEP
```

EXAMPLE:

```
=>wireless status
Wireless SSID      :WLAN_Alcatel
Wireless channel   :6
Regulation domain  :ETSI
WEP encryption     :Off
Association Control :Off
=>wireless startWEP
=>wireless status
Wireless SSID      :WLAN_Alcatel
Wireless channel   :6
Regulation domain  :ETSI
WEP encryption     :On
Association Control :Off
=>
```

RELATED COMMANDS:

wireless random	generate a random WEP key
wireless stopWEP	Stop WEP encryption.
wireless wepkey	Set a 40-bits WEP key yourself.
wireless reset	Reset SpeedTouch™ 570 WLAN configuration to defaults.

wireless stopWEP

Stop WEP encryption.

SYNTAX:

```
wireless stopWEP
```

EXAMPLE:

```
=>wireless status
Wireless SSID      :WLAN_Alcatel
Wireless channel   :6
Regulation domain  :ETSI
WEP encryption     :On
Association Control :Off
=>wireless startWEP
=>wireless status
Wireless SSID      :WLAN_Alcatel
Wireless channel   :6
Regulation domain  :ETSI
WEP encryption     :Off
Association Control :Off
=>
```

RELATED COMMANDS:

wireless random	generate a random WEP key
wireless startWEP	Start WEP encryption.
wireless wepkey	Set a 40-bits WEP key yourself.
wireless reset	Reset SpeedTouch™ 570 WLAN configuration to defaults.

wireless status

Show **SpeedTouch™ 570** WLAN configuration.

SYNTAX:

```
wireless status
```

EXAMPLE:

```
=>wireless status
Wireless SSID      :WLAN_Alcatel
Wireless channel   :6
Regulation domain  :ETSI
WEP encryption     :Off
Association Control :Off
=>
```

RELATED COMMANDS:

wireless ssid	Set SpeedTouch™ 570 WLAN SSID.
wireless channel	Set SpeedTouch™ 570 WLAN radio channel.
wireless startWEP	Start WEP encryption.
wireless stopWEP	Stop WEP encryption.
wireless disablecontrol	Disable association control.
wireless enablecontrol	Enable association control.
wireless reset	Reset SpeedTouch™ 570 WLAN configuration to defaults.

wireless wepkey

Set a user defined 40-bits value for the WEP key.

SYNTAX:

```
wireless wepkey    wepkey = <string>
```

wepkey = <string> The 40-bits key, entered as XX:XX:XX:XX:XX REQUIRED

EXAMPLE:

```
=>wireless wepkey wepkey = 1:22:33:4:ab  
=>
```

RELATED COMMANDS:

wireless random	generate a random WEP key
wireless startWEP	Start WEP encryption.
wireless stopWEP	Stop WEP encryption.
wireless reset	Reset SpeedTouch™ 570 WLAN configuration to defaults.

Alcatel SpeedTouch™ 570

CLI Command Index

Index

A

add
 dhcp server add, 73
 dns add, 94
 phonebook add, 190
apadd, ip apadd, 142
apdelete, ip apdelete, 144
aplist, ip aplist, 145
applist, nat applist, 174
arpadd, ip arpadd, 146
arpdelete, ip arpdelete, 147
arplist, ip arplist, 148
assign, firewall assign, 116
autolist, phonebook autolist, 192

B

beacon_period, wireless beacon_period, 242
bind, nat bind, 175
bindlist, nat bindlist, 176

C

call, td call, 240
channel, wireless channel, 243
cleanup, software cleanup, 226
clear
 dhcp client clear, 57
 dns clear, 95
 firewall rule clear, 131
clearpassword, system clearpassword, 232
client, dhcp server client, 74

clrstats
 dhcp server clrstats, 75
 dns clrstats, 96
config
 bridge config, 24
 dhcp client config, 58
 dhcp server config, 76
 ip config, 149
counters, wireless counters, 244
create
 firewall chain create, 126
 firewall rule create, 132
 nat create, 177

D

defserver, nat defserver, 179
delete
 dhcp server delete, 78
 dns delete, 97
 firewall chain delete, 127
 firewall rule delete, 136
 nat delete, 180
 phonebook delete, 193
deletemac, wireless deletemac, 245
deletpassive, software deletpassive, 227
disable, nat disable, 181
disablecontrol, wireless disablecontrol, 246
domain, dns domain, 98

E

enable, nat enable, 182
enablecontrol, wireless enablecontrol, 247
erase, config erase, 50

F

filtconfig, wireless filtconfig, 248

flush

- bridge flush, 25
- cip flush, 40
- config flush, 51
- dhcp client flush, 59
- dhcp server flush, 79
- dns flush, 99
- firewall flush, 117
- firewall rule flush, 137
- ip flush, 152
- mer flush, 164
- nat flush, 183
- phonebook flush, 194
- ppp flush, 200
- pptp flush, 218
- system flush, 233
- wireless flush, 249

fragment_thres, wireless fragment_thres, 250

fwddadd, dns fwddadd, 100

fwdddelete, dns fwdddelete, 101

fwddlist, dns fwddlist, 102

fwddtable, dns fwddtable, 103

H

help, 12

I

ifadd

- bridge ifadd, 26
- cip ifadd, 41
- dhcp client ifadd, 60
- mer ifadd, 165
- ppp ifadd, 201

ifattach

- bridge ifattach, 27
- dhcp client ifattach, 61
- mer ifattach, 166
- ppp ifattach, 203

ifconfig

- bridge ifconfig, 28
- dhcp client ifconfig, 62
- ip ifconfig, 153
- mer ifconfig, 167
- ppp ifconfig, 204

ifdelete

- bridge ifdelete, 30
- cip ifdelete, 42
- dhcp client ifdelete, 64
- mer ifdelete, 168
- ppp ifdelete, 208

ifdetach

- bridge ifdetach, 31
- mer ifdetach, 169
- ppp ifdetach, 209

iflist

- bridge iflist, 32
- cip iflist, 43
- dhcp client iflist, 65
- ip iflist, 154
- mer iflist, 170
- ppp iflist, 210

ifrelease, dhcp client ifrelease, 66

ifrenew, dhcp client ifrenew, 68

ifscan, ppp ifscan, 211

info, adsl info, 22

L

list

- dhcp server list, 80
- dns list, 104
- firewall chain list, 128
- firewall list, 118
- firewall rule list, 138
- nat list, 184
- phonebook list, 195
- pptp list, 219

listacl, wireless listacl, 251

load

- bridge load, 33
- cip load, 44
- config load, 52
- dhcp client load, 70
- dhcp server load, 81
- dns load, 105
- firewall chain load, 129
- firewall load, 119
- ip load, 155
- mer load, 171
- nat load, 185
- phonebook load, 196
- ppp load, 212
- pptp load, 220
- system load, 234

long_retry_lim, wireless long_retry_lim, 252

M

macadd, bridge macadd, 34

macdelete, bridge macdelete, 35

maclist, bridge maclist, 36

match, firewall match, 120

N

nslookup, dns nslookup, 106

P

params, wireless params, 253

ping, ip ping, 156

policy, dhcp server policy, 82

profadd, pptp profadd, 221

profdelete, pptp profdelete, 222

proflist, pptp proflist, 223

pvcadd, cip pvcadd, 45

pvcdelete, cip pvcdelete, 46

pvclist, cip pvclist, 47

R

random, wireless random, 254

reboot, system reboot, 235

reset

- config reset, 53

- wireless reset, 255

rtadd

- ip rtadd, 157

- ppp rtadd, 213

rtdelete

- ip rtdelete, 158

- ppp rtdelete, 215

rtlist, ip rtlist, 159

rts_threshold, wireless rts_threshold, 256

S

save

- bridge save, 37

- cip save, 48

- config save, 54

- dhcp client save, 71

- dhcp server save, 83

- dns save, 107

- firewall chain save, 130

- firewall save, 123

- ip save, 160

- mer save, 172

- nat save, 186

- phonebook save, 197

- ppp save, 216

- pptp save, 224

- system save, 236

sendto, ip sendto, 161

setpassive, software setpassive, 228

setpassword, system setpassword, 237

short_retry_lim, wireless short_retry_lim, 257

spoof, dhcp server spoof, 84
ssid, wireless ssid, 258
start
 dhcp server start, 86
 dns start, 108
startWEP, wireless startWEP, 259
stats
 dhcp client stats, 72
 dhcp server stats, 87
 dns stats, 109
 firewall rule stats, 139
status
 dhcp server status, 89
 dns status, 110
 wireless status, 261
stop
 dhcp server stop, 90
 dns stop, 111
stopWEP, wireless stopWEP, 260
switch, software switch, 229

T

toutfwd, dns toutfwd, 112
troff
 dhcp server troff, 91
 dns troff, 113
tron
 dhcp server tron, 92
 dns tron, 114

U

unbind, nat unbind, 187

V

version, software version, 230

W

wepkey, wireless wepkey, 262