

# **SPEED TOUCH 510**

## **CLI Reference Guide**

**ADSL Router Series**



Status Released

Change Note BD F aa 41610

**Short Title** CD-RG AST510 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

<b>Preface</b> .....	<b>5</b>
<b>1 ADSL Commands</b> .....	<b>23</b>
<b>2 Bridge Commands</b> .....	<b>27</b>
<b>3 CIP Commands</b> .....	<b>41</b>
<b>4 Config Commands</b> .....	<b>49</b>
<b>5 DHCP Commands</b> .....	<b>55</b>
<b>6 DNS Commands</b> .....	<b>89</b>
<b>7 Env Commands</b> .....	<b>109</b>
<b>8 Firewall Commands</b> .....	<b>117</b>
<b>9 IP Commands</b> .....	<b>139</b>
<b>10 MER Commands</b> .....	<b>159</b>
<b>11 NAT Commands</b> .....	<b>167</b>
<b>12 Phonebook Commands</b> .....	<b>181</b>
<b>13 PPP Commands</b> .....	<b>189</b>
<b>14 PPTP Commands</b> .....	<b>207</b>
<b>15 Software Commands</b> .....	<b>213</b>
<b>16 System Commands</b> .....	<b>219</b>
<b>17 TD Commands</b> .....	<b>227</b>
<b>Index</b> .....	<b>233</b>



# Preface

Welcome to the **Alcatel SpeedTouch™ 510Series Command Line Interface Reference Guide !**

For the Service Provider this Reference Guide aims to give a concise and practical document for creating a customized configuration profile file, based on the **SpeedTouch™ 510Series** CLI command set, to be used by the end-user to configure the **SpeedTouch™ 510Series** and PC(s).

For the fastidious user this Reference Guide aims to give a handbook for advanced and detailed configuration and troubleshooting of the **SpeedTouch™ 510Series** via its character based Command Line Interface.

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

- ▶ **Alcatel SpeedTouch™ 510**
- ▶ **Alcatel SpeedTouch™ 510i**

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™ 510Series** CLI. In brief some general manipulations are described 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™ 510Series** are alphabetically described per command group.

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™ 510Series**
- 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:

- SpeedTouch™ 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™ 510 Series

## CLI Navigation

---





---

# 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 :
dhcp      dns      td      mer      bridge
pptp     ppp      cip     nat      adsl
qosbook  phonebook ip     software system
config   firewall  env

=>
```

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™ 510**.

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 :
dhcp      dns          td          mer          bridge
pptp      ppp           cip         nat          adsl
qosbook   phonebook  ip         software     system
config    firewall    env

=>
```

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** (executed from the the firewall command group selection) results in a description of the syntax for the command.

**EXAMPLE:**

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

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

=>
```

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

---

## Command Completion

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

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

For example, 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.

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 'chain ='. 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** (only applicable for the ADSL/POTS variants)
- ▶ **bridge**
- ▶ **cip**
- ▶ **config**
- ▶ **dhcp**
- ▶ **dns**
- ▶ **env**
- ▶ **firewall**
- ▶ **ip**
- ▶ **mer**
- ▶ **nat**
- ▶ **phonebook**
- ▶ **ppp**
- ▶ **pptp**
- ▶ **qosbook**
- ▶ **software**
- ▶ **system**
- ▶ **td**

---

# Command Line Interface Commands

All CLI commands are commands that operate on, or configure, the **SpeedTouch™ 510** settings.

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

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

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

## EXAMPLE:

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

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

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

## EXAMPLE:

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

---

# Configuration Profile Files and the CLI

## Configuration Profile Files

The configuration profile files, used by the Setup wizard, or directly uploaded via the **SpeedTouch™ 510** web pages, consist of a set of CLI commands. Only CLI commands, which are part of the **SpeedTouch™ 510** CLI command set may be used in the .ini and .def files. For readability, configuration profile files are partitioned in paragraphs. These paragraphs reflect a CLI command group selection in the **SpeedTouch™ 510** CLI. However, due to internal processing, not all paragraph names are the same as their corresponding CLI command group selection.

In the following overview the relationship between the available **Alcatel SpeedTouch™510** CLI commands and the [`<configuration>.ini`] paragraphs is given:

- ▶ **[env.ini]**  
All commands present in the **env** CLI command group.
- ▶ **[phone.ini]**  
All commands present in the **phonebook** CLI command group.
- ▶ **[bridge.ini]**  
All commands present in the **bridge** CLI command group.
- ▶ **[mer.ini]**  
All commands present in the **mer** CLI command group.
- ▶ **[pptp.ini]**  
All commands present in the **pptp** CLI command group.
- ▶ **[ppp.ini]**  
All commands present in the **ppp** CLI command group.
- ▶ **[cip.ini]**  
All commands present in the **cip** CLI command group.
- ▶ **[ip.ini]**  
All commands present in the **ip** CLI command group.
- ▶ **[nat.ini]**  
All commands present in the **nat** CLI command group.
- ▶ **[pfilter.ini]**  
All commands present in the **firewall chain** CLI command subgroup and all commands present in the **firewall rule** CLI command subgroup.
- ▶ **[pfirewall.ini]**  
All commands present in the **firewall** CLI command group (except those part of the **firewall chain** and **firewall rule** CLI command subgroups).
- ▶ **[dhcp.ini]**  
All commands present in the **dhcp server** CLI command subgroup.
- ▶ **[dhcc.ini]**  
All commands present in the **dhcp client** CLI command subgroup.
- ▶ **[dnsd.ini]**  
All commands present in the **dns** CLI command group.

---

► **[system.ini]**

All commands present in the **system** CLI command group.

## CLI Commands in Configuration Profile Files

CLI commands in a paragraph of a configuration profile file should always be constructed in their complete form. Uncompleted CLI commands, i.e. commands in which required parameters are not specified, used in a configuration profile file will be discarded by the CLI command interpreter. This may result in a misconfigured **Alcatel SpeedTouch™510**.

In configuration profile files, the use of customization variables, allow the Setup wizard to invite the end-user to provide some input regarding the settings of the **Alcatel SpeedTouch™510**. The declaration of such variables must be done in the **[env.ini]** paragraph. Further use of the resulting variable TAG is allowed through all other paragraphs, even multiple times.

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

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

## Customizing Configurationh Profile Files

For more information on the customization possibilities of the **Alcatel SpeedTouch™510**, the Setup wizard and the configuration profile files, please check the Alcatel support pages at:

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

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

Or contact your local Alcatel Sales representative.

---

## Direct FTP Access

### The SpeedTouch™ 510 File System

The **SpeedTouch™510** permanent storage, further referred to as 'file system', exists of nonvolatile memory responsible for storing, retrieving and maintaining the **SpeedTouch™510** software image(s), configuration profile files and optionally default settings files.

The file system of the **SpeedTouch™510** is accessible via the FTP transport protocol. This allows to transfer the **SpeedTouch™510** software image(s) and/or configuration profile files and default settings files.

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

Proceed as indicated in the example below to open an FTP session to the **SpeedTouch™510** file system:

#### EXAMPLE:

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

### SpeedTouch™ 510 File System Structure

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

The 'root' contains next to the two subdirectories 'active' and 'dl' all necessary files for the **SpeedTouch™510** to boot correctly.

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

The 'dl' directory contains dormant software image. In case you have made changes to the **SpeedTouch™510** configuration and saved them, be it via a Telnet session, via the web pages or via the Setup wizard, a *user.ini* configuration profile file is created in the 'dl' subdirectory.

In other words, after each 'Save all', or *config save* call, the *user.ini* configuration profile file present in the 'dl' subdirectory reflects the current configuration of the **SpeedTouch™510**.

---

## SpeedTouch™ 510 File System Access Rights

Following access rights apply on the file system:

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

## FTP File Transfer

To allow correct file transfers the transfer mode must be set to "binary". Moreover, it is suggested to turn on the hashing option to be able to see how the file transfer proceeds:

EXAMPLE:

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

---

# Alcatel SpeedTouch™ 510 Series

## CLI Command Description

---



# 1 ADSL Commands

The **adsl** command group is only applicable to the **SpeedTouch™ 510** ADSL/POTS variant, NOT to the **SpeedTouch™ 510i** ADSL/ISDN variant.

**adsl (to access the ADSL level)**

**adsl config**

**adsl info**

## adsl config

Show/set the ADSL/POTS configuration.

SYNTAX:

<b>adsl config</b>	<b>[opermode = &lt;ansi g.dmt_annex_a g.lite multimode&gt;]</b> <b>[maxbitstertoneUS = &lt;number{10-14}&gt;]</b>
--------------------	--

<i>[opermode]</i>	The operational mode of the <b>SpeedTouch™ 510</b> modem. Choose between: <ul style="list-style-type: none"> <li>▪ ansi</li> <li>▪ g.dmt_annex_a</li> <li>▪ g.lite</li> <li>▪ multimode</li> </ul> By default the ADSL modem will start in multimode.	OPTIONAL
<i>maxbitstertoneUS</i>	A number between 10 and 14 (bits per tone). Represents the maximum number of bits which hcan be allocated to each ADSL DMT tone in the upstream direction. By default the modem will use up to 13 bits per tone.	OPTIONAL

EXAMPLE:

<pre>=&gt;adsl config ADSL configuration:       opermode = multimode       maxbitstertoneUS = 13 =&gt;</pre>
--

## adsl info

Show ADSL/POTS statistics and information about the **SpeedTouch™ 510** 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
                    :          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 macadd**  
**bridge macdelete**  
**bridge maclist**

## **bridge config**

Show/set bridge aging policy.

SYNTAX:

<b>bridge config</b>	<b>[age = &lt;number {10 - 100000}&gt;]</b>
----------------------	---

<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>=&gt;bridge config Aging : 300 =&gt;bridge config age=600 =&gt;bridge config Aging : 600 =&gt;</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:wan0     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:wan1     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
=>
```

## bridge ifadd

Create a bridge interface.

SYNTAX:

<b>bridge ifadd</b>	<b>intf = &lt;string&gt;</b> <b>dest = &lt;phonebook entry&gt;</b>
---------------------	---

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

RELATED COMMANDS:

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

## bridge ifattach

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

SYNTAX:

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

RELATED COMMANDS:

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

## bridge ifconfig

Configure a bridge interface.

SYNTAX:

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

<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"> <li>▪ llc/snap</li> <li>▪ vcmux</li> </ul>	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"> <li>▪ off</li> <li>▪ on</li> </ul>	OPTIONAL
<i>[portstate]</i>	The bridge portstate for this interface. Choose between: <ul style="list-style-type: none"> <li>▪ disabled</li> <li>▪ learning</li> <li>▪ forwarding</li> </ul>	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=TestBridge
TestBridge : dest : Br1
              Retry: 10   QoS: default   Encaps: llc/snap   Fcs: off
              Connection State: connected   Port: wan0   PortState: forwarding
              RX bytes: 75           frames: 12
              TX bytes: 30246        frames: 91   dropframes: 0
=>bridge ifconfig intf=TestBridge encaps=vcmux retry=15
=>bridge iflist intf=TestBridge
TestBridge : dest : Br1
              Retry: 15   QoS: default   Encaps: vcmux   Fcs: off
              Connection State: connected   Port: wan0   PortState: forwarding
              RX bytes: 83           frames: 13
              TX bytes: 30740        frames: 102   dropframes: 0
=>

```

## RELATED COMMANDS:

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

## bridge ifdelete

Delete a bridge interface.

**bridge ifdelete**      *intf* = <iname>

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

```

### RELATED COMMANDS:

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

## bridge ifdetach

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

SYNTAX:

<b>bridge ifdetach</b>	<b>intf = &lt;ifname&gt;</b>	
<i>intf</i>	The name of the bridge interface to detach.	REQUIRED

EXAMPLE:

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

RELATED COMMANDS:

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

## bridge iflist

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

SYNTAX:

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

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

EXAMPLE OUTPUT:

```
=>bridge iflist
OBC      : Internal
          Connection State: connected   Port:OBC      PortState: forwarding
          RX bytes: 75783      frames: 572
          TX bytes: 82768372   frames: 341221 dropframes: 0
eth0     : Internal
          Connection State: connected   Port:eth0     PortState: forwarding
          RX bytes: 156344216 frames: 5899238
          TX bytes: 75689     frames: 425     dropframes: 5558017
TestBridge : dest : Br1
          Retry: 15   QoS: default   Encaps: vcmux   Fcs: off
          Connection State: connected   Port:wano0    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:

<b>bridge ifadd</b>	Create a bridge interface.
<b>bridge ifattach</b>	Attach a created bridge interface.
<b>bridge ifconfig</b>	Configure a bridge interface.
<b>bridge ifdelete</b>	Delete a bridge interface.
<b>bridge ifdetach</b>	Detach a bridge interface.

## bridge macadd

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

### SYNTAX:

<b>bridge macadd</b>	<b>intf = &lt;ifname&gt;</b> <b>hwaddr = &lt;hardware-address&gt;</b>
----------------------	--

<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:

<b>bridge macdelete</b>	Delete a MAC address entry.
<b>bridge maclist</b>	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:

<b>bridge macadd</b>	Add a static MAC address entry.
<b>bridge maclist</b>	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.



## 3 CIP Commands

**cip (to access the CIP level)**

**cip flush**

**cip ifadd**

**cip ifdelete**

**cip iflist**

**cip pvcadd**

**cip pvcdelete**

**cip pvclist**

## ***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
=>
```

## ***cip ifadd***

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

SYNTAX:

<b><i>cip ifadd</i></b>	<b><i>addr = &lt;ip-address&gt;</i></b> <b><i>[netmask = &lt;ip-mask (dotted or cidr)&gt;]</i></b> <b><i>[uniaddr = &lt;portspec:address.selector&gt;]</i></b>
-------------------------	--

<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:

<b><i>cip ifdelete</i></b>	Delete a CIP interface.
<b><i>cip ifadd</i></b>	Show current CIP configuration.

## ***cip ifdelete***

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

SYNTAX:

<b><i>cip ifdelete</i></b>	<b><i>addr = &lt;ip-address&gt;</i></b>
----------------------------	---

<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:

<b><i>cip ifadd</i></b>	Create a CIP interface.
<b><i>cip iflist</i></b>	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 pvcadd**

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

### SYNTAX:

<b>cip pvcadd</b>	<b>dest = &lt;phonebookname&gt;</b> <b>[destaddr = &lt;ip-address&gt;]</b> <b>[mtu = &lt;number {273–20000}&gt;]</b>
-------------------	--

<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      atmpport = 0      vpi = 8      vci = 80      dest_ip = 172.16.1.2
              encaps = llc      mtu = 546
=>
```

### RELATED COMMANDS:

<b>cip pvdelete</b>	Delete a PVC ARP entry.
<b>cip pvclist</b>	Show current PVC ARP entries.

## ***cip pvdelete***

Delete a PVC ARP entry.

### SYNTAX:

<b><i>cip pvdelete</i></b>	<b><i>dest = &lt;phonebookname&gt;</i></b>
----------------------------	--

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

### EXAMPLE:

```
=>cip pvclist
CIPPVC1_____atmport = 0      vpi  = 8    vci  = 80    dest_ip = 172.16.1.2
                encaps = llc    mtu   = 546
=>cip pvdelete dest=CIPPVC1
=>cip pvclist
=>
```

### RELATED COMMANDS:

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

***cip pvclist***

Show current PVC ARP entries.

SYNTAX:

```
cip pvclist
```

EXAMPLE OUTPUT:

```
=>cip pvclist
CIPPVC1      atmpport = 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          atmpport = 0      vpi = 6      vci = 99      dest_ip = 172.16.1.3
              encaps = llc      mtu = 9180
8.50        atmpport = 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.

## 4 Config Commands

**config (to access the Config level)**

**config erase**

**config flush**

**config load**

**config save**

## **config erase**

Physically remove all saved configurations, i.e. the user configuration file, saved in the **SpeedTouch™ 510** permanent storage, is deleted.

SYNTAX:

```
config erase
```

EXAMPLE:

```
=>config erase
```

RELATED COMMANDS:

**config flush**  
**config load**  
**config save**

Flush complete runtime configuration.  
Load complete saved or default configuration.  
Save complete runtime configuration.

## config flush

Flush complete current configuration without affecting saved configurations.

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

SYNTAX:

<b>config flush</b>	<b>[flush_ip = &lt;{no yes}&gt;]</b>
---------------------	--------------------------------------

<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:

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

## config load

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

SYNTAX:

<b>config load</b>	<b>[load_ip = &lt;{no yes}&gt;]</b> <b>[defaults = {yes no}]</b>
--------------------	---

**[load\_ip]** Load IP settings (yes) or not (no). OPTIONAL  
Not keeping the IP settings could cause lost IP connectivity in the LAN.

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

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:

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

## **config save**

Save all existing configurations and modifications entered by the user.

The result of executing this command is a user.ini file saved in the **SpeedTouch™ 510** permanent storage. This file can be downloaded via the **SpeedTouch™ 510** web pages.

SYNTAX:

```
config save
```

EXAMPLE:

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

RELATED COMMANDS:

**config erase**

Physically remove all saved configurations.

**config flush**

Flush complete current configuration.

**config load**

Load complete saved or default configuration.



## 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 stats**  
**dhcp server (to access the DHCP Server level)**  
**dhcp server clear**  
**dhcp server config**  
**dhcp server policy**  
**dhcp server spoof**  
**dhcp server start**  
**dhcp server stats**  
**dhcp server stop**

**dhcp server lease (to access the server lease level)**

**dhcp server lease add**

**dhcp server lease delete**

**dhcp server lease flush**

**dhcp server lease list**

**dhcp server pool (to access the server pool level)**

**dhcp server pool add**

**dhcp server pool config**

**dhcp server pool delete**

**dhcp server pool flush**

**dhcp server pool list**

## **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.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 flush
=>dhcp client iflist
No dynamic interfaces defined.
=>
```

## **dhcp client ifadd**

Create a DHCP lease for a specific interface.

### SYNTAX:

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

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

### EXAMPLE:

```

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

```

### RELATED COMMANDS:

<b>dhcp client ifattach</b>	Attach a DHCP lease to an interface.
<b>dhcp client ifconfig</b>	Configure a DHCP lease created for a specific interface.
<b>dhcp client ifdelete</b>	Delete a dynamic interface.
<b>dhcp client iflist</b>	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:

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

## 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|none}>]
                    [hostname = <hostname|"">]
                    [addr = <ip-address>]
                    [leasetime = <number>]
                    [addrtrans = <{none|pat}>]
                    [dns = <{off|on}>]
                    [gateway = <{off|on}>]
                    [metric = <number{0-100}>]

```

<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. Use none in case no clientid should be associated with this lease.	OPTIONAL
[ <i>hostname</i> ]	The host name of the client to be associated with the lease. Use "" in case no hostname should not be associated with this lease.	OPTIONAL
[ <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
[ <i>metric</i> ]	A number between 10 and 100000. Represents the gateway route metric By default the gateway route metric is 1.	OPTIONAL

## EXAMPLE:

```

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

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

## **dhcp client ifdelete**

Delete a dynamic interface.

SYNTAX:

```
dhcp client ifdelete 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 ifdelete intf NewMer
=>dhcp client iflist
No dynamic interfaces defined.
=>
```

RELATED COMMANDS:

<b>dhcp client ifadd</b>	Create a DHCP lease for a specific interface.
<b>dhcp client ifattach</b>	Attach a DHCP lease to an interface.
<b>dhcp client ifconfig</b>	Configure a DHCP lease created for a specific interface.
<b>dhcp client iflist</b>	Show all dynamic interfaces.
<b>dhcp client ifrelease</b>	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™ 510** 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.leasetime= 10800 s
              trying to get a lease for 8 min, 32 sec
              transmission of DISCOVER in 57 sec
              retransmission timeout: 64
              nbr of retransmissions: 14
Number of leases: 1
Total size of table: 19,  in use: 1, free:94 %
=>dhcp client ifattach intf=NewMer
=>dhcp client iflist
NewMer      : [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.leasetime= 10800 s
Number of leases: 1
Total size of table: 19,  in use: 1, free:94 %
=>
```

## EXAMPLE INPUT/OUTPUT IN A NETWORKED ENVIRONMENT:

The **SpeedTouch™ 510** 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>
```

<i>intf</i>	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™ 510** 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 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 clear**

Clear **SpeedTouch™ 510** DHCP server statistics.

SYNTAX:

```
dhcp server clear
```

EXAMPLE:

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

RELATED COMMANDS:

**dhcp server stats**                    Show DHCP server statistics.

## dhcp server config

Show/set **SpeedTouch™ 510** DHCP server configuration settings.

SYNTAX:

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

<i>[autodhcp]</i>	Allow the <b>SpeedTouch™ 510</b> to present itself as DHCP client (AutoDHCP mode) at boot time and probe for another DHCP server on the network for some time before starting its own DHCP server (yes) or immediately start the DHCP server (no).	OPTIONAL
<i>[scantime]</i>	A number between 0 and 1814400 (seconds). Represents the time the <b>SpeedTouch™ 510</b> scans for another DHCP server to be active in the network. By default the scantime is 20 seconds.	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>[trace]</i>	Disable verbose console logging and generation of debug traces (off) or enable verbose console logging and generation of debug traces (on). By default tracing is disabled.	OPTIONAL

EXAMPLE:

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

RELATED COMMANDS:

**dhcp server stats** Show current DHCP server state and statistics.

## dhcp server policy

Show/set **SpeedTouch™ 510** DHCP server policy.

SYNTAX:

<b>dhcp server policy</b>	<b>[verifyfirst = &lt;off on&gt;]</b> <b>[trustclient = &lt;off on&gt;]</b>
---------------------------	--

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

EXAMPLE:

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

RELATED COMMANDS:

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

## dhcp server spoof

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

SYNTAX:

<b>dhcp server spoof</b>	<b>[failtime = &lt;number&gt;]</b> <b>[errorlt = &lt;number&gt;]</b> <b>[dodlt = &lt;number&gt;]</b>
--------------------------	--

<i>[failtime]</i>	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 <b>SpeedTouch™ 510</b> 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 <b>SpeedTouch™ 510</b> DHCP server will allocate a local private IP address to the DHCP client. By default the failtime is 4 seconds.	OPTIONAL
<i>[errorlt]</i>	A number between 0 and 1814400 (seconds). Represents the leasetime of the private address issued when a PPP link fails. In case the PPP link fails after failtime has elapsed, this parameter determines how long the private DHCP lease must be maintained before retrying to set up the PPP link again. By default the error lease time is 60 seconds.	OPTIONAL
<i>[dodlt]</i>	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 temporary DHCP lease must be maintained before checking whether a public IP address negotiated by a triggered PPP link is available. By default the dial-on-demand lease time is 10 seconds.	OPTIONAL

EXAMPLE:

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

RELATED COMMANDS:

<b>dhcp server policy</b>	Set DHCP server policy.
<b>dhcp server stats</b>	Show current DHCP server state and statistics.

## **dhcp server start**

Start Speed510Touch™ 510 DHCP server.

### SYNTAX:

```
dhcp server start
```

### EXAMPLE:

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

### RELATED COMMANDS:

**dhcp server stats**

Show current DHCP server state and statistics.

**dhcp server stop**

Stop DHCP server.

## **dhcp server stats**

Show **SpeedTouch™ 510** DHCP server statistics.

SYNTAX:

```
dhcp server stats
```

EXAMPLE OUTPUT:

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

DESCRIPTION:

<i>DHCP server state</i>	Indicates the state of the <b>SpeedTouch™ 510</b> DHCP server.
<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 <b>SpeedTouch™ 510</b> DHCP server to verify whether the IP address is already in use on the LAN or not. ( <b>dhcp server policy</b> verifyfirst=yes)
<i>Second DHCP server</i>	Indicates whether a concurrent DHCP server was found on the LAN or not.

**RELATED COMMANDS:**

**dhcp server clear** Clear DHCP server statistics.

## **dhcp server stop**

Stop SpeedTouch™ 510 DHCP server.

SYNTAX:

```
dhcp server stop
```

EXAMPLE:

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

RELATED COMMANDS:

<b>dhcp server start</b>	Start DHCP server.
<b>dhcp server stats</b>	Show current DHCP server state and statistics.

## dhcp server lease add

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

SYNTAX:

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

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

EXAMPLE:

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

RELATED COMMANDS:

<b>dhcp server lease delete</b>	Delete a DHCP lease.
<b>dhcp server lease flush</b>	Delete all DHCP leases.
<b>dhcp server lease list</b>	Show current DHCP leases.

## dhcp server lease delete

Delete a DHCP lease.

SYNTAX:

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

<i>clientid</i>	The DHCP client identification string of the DHCP lease.	REQUIRED
[ <i>index</i> ]	The index number of the entry to be deleted. Execute <b>dhcp server lease list</b> to see a list of the index numbers of all current DHCP leases.	OPTIONAL

EXAMPLE:

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

RELATED COMMANDS:

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

## **dhcp server lease flush**

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

SYNTAX:

```
dhcp server lease flush
```

EXAMPLE:

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

RELATED COMMANDS:

<b>dhcp server lease add</b>	Add a DHCP lease manually.
<b>dhcp server lease delete</b>	Delete a DHCP lease.
<b>dhcp server lease list</b>	Show current DHCP leases.

## **dhcp server lease list**

List current DHCP leases, indicated by their index number.

SYNTAX:

```
dhcp server lease list
```

EXAMPLE OUTPUT:

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

RELATED COMMANDS:

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

## dhcp server pool add

Add a DHCP server pool.

SYNTAX:

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

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

EXAMPLE:

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

RELATED COMMANDS:

<b>dhcp server pool delete</b>	Delete a DHCP pool.
<b>dhcp server pool flush</b>	Delete all DHCP pools.
<b>dhcp server pool list</b>	Show current DHCP pools.

## dhcp server pool config

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

SYNTAX:

```

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

<i>name</i>	The name of the DHCP server pool to configure.	REQUIRED
<i>[index]</i>	A number between 0 (highest priority) and the highest number (lowest priority) found in the list of existing DHCP server pools. Represents a (higher) priority for the DHCP server pool.	OPTIONAL
<i>[poolstart]</i>	The lowest IP address in the DHCP address range to use for leasing. Default value of this parameter is 0.0.0.0 (not specified), which means that the lowest IP address of the pool will be defined by the remote server via IPCP as soon as the PPP Helenius connection is established.	OPTIONAL
<i>[poolend]</i>	The highest IP address in the DHCP address range to use for leasing. Default value of this parameter is 0.0.0.0 (not specified), which means that the highest IP address of the pool will be defined by the remote server via IPCP as soon as the PPP Helenius connection is established.	OPTIONAL
<i>[netmask]</i>	The applicable netmask for the DHCP leases.	OPTIONAL
<i>[gateway]</i>	The IP address of the default gateway for the DHCP clients. Default value of this parameter is 0 (not specified), which means that the gateway IP address will be communicated by the remote server as soon as the PPP Helenius connection is established or that the <b>SpeedTouch™ 510</b> acts as the LAN default gateway.	OPTIONAL
<i>[primdns]</i>	The IP address of the primary DNS server for the DHCP clients. Default value of this parameter is 0 (not specified), which means that the IP address of the DNS server will be communicated by the remote server as soon as the PPP Helenius connection is established or that the <b>SpeedTouch™ 510</b> acts as the LAN DNS server.	OPTIONAL

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

**EXAMPLE:**

```

=>dhcp server pool list
Pool          Start          End          State      PPP
0 LAN_Private  10.0.0.1        10.255.255.254  USED
=>dhcp server pool config name=My_Pool poolstart=192.6.11.101 poolend=192.6.11.254
netmask=255.255.255 gateway=192.6.11.100 leasetime=21600
=>dhcp server pool list
Pool          Start          End          State      PPP
0 LAN_Private  10.0.0.1        10.255.255.254  USED
1 My Pool     192.6.11.101    192.6.11.254    USED
=>

```

**RELATED COMMANDS:**

**dhcp server pool list**            Show current DHCP pools.

## **dhcp server pool delete**

Delete a DHCP pool.

### SYNTAX:

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

<i>name</i>	The name of the DHCP server pool to delete. Execute <b>dhcp server pool list</b> to see a list of all current DHCP leases.	REQUIRED
-------------	---	----------

### EXAMPLE:

```
=>dhcp server pool list
Pool      Start      End      State      PPP
0 LAN_Private 10.0.0.1  10.255.255.254 USED
1 My_Pool   192.6.11.101 10.255.255.254 USED
=>dhcp server pool delete name=My_Pool
=>dhcp server pool list
Pool      Start      End      State      PPP
0 LAN_Private 10.0.0.1  10.255.255.254 USED
=>
```

### RELATED COMMANDS:

<b>dhcp server pool add</b>	Add a DHCP pool.
<b>dhcp server pool flush</b>	Delete all DHCP pools.
<b>dhcp server pool list</b>	Show current DHCP pools.

## **dhcp server pool flush**

Flush all DHCP pools

The flush command does not impact previously saved configurations.

SYNTAX:

```
dhcp server flush
```

EXAMPLE:

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

RELATED COMMANDS:

<b>dhcp server pool add</b>	Add a DHCP pool.
<b>dhcp server pool delete</b>	Delete a DHCP pool.
<b>dhcp server pool list</b>	Show current DHCP pools.

## **dhcp server pool list**

List current DHCP leases.

SYNTAX:

```
dhcp server pool list
```

EXAMPLE:

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

RELATED COMMANDS:

<b>dhcp server pool add</b>	Add a DHCP pool.
<b>dhcp server pool delete</b>	Delete a DHCP pool.
<b>dhcp server pool flush</b>	Delete all DHCP pools.

## 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 nslookup**

**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:

<b>dns add</b>	<b>hostname = &lt;string&gt;</b> <b>[addr = &lt;ip-address&gt;]</b>
----------------	--

<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 <b>SpeedTouch™ 510</b> 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:

<b>dns list</b>	List current DNS entries.
<b>dns delete</b>	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:

<b>dns delete</b>	<b>index = &lt;number&gt;</b>
-------------------	-------------------------------

<i>index</i>	The index number of the entry to be deleted. Execute <b>dns list</b> 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:

<b>dns add</b>	Add a static DNS entry.
<b>dns list</b>	List current DNS entries.

## dns domain

Set local DNS (sub)domain name.

### SYNTAX:

<b>dns domain</b>	<b>domain = &lt;string&gt;</b>
-------------------	--------------------------------

<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:

<b>dns list</b>	List current DNS entries.
-----------------	---------------------------

## **dns flush**

Flush complete **SpeedTouch™ 510** DNS server/forwarder configuration and static entries. The flush command does not impact previously saved configurations.

SYNTAX:

```
dns flush
```

EXAMPLE:

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

## dns fwdadd

Add a DNS forwarding entry. 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:

<b>dns fwdadd</b>	<b>dns = &lt;ip-address&gt;</b> <b>src = &lt;ip-address&gt;</b> <b>mask = &lt;ip-mask (dotted or cidr)&gt;</b> <b>[direct = &lt;number&gt;]</b>
-------------------	--

<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 <b>SpeedTouch™ 510</b> 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:

<b>dns fwddelete</b>	Delete a DNS forwarding entry.
<b>dns fwdlist</b>	Show current DNS forwarding entries.

## dns fwddelete

Delete a DNS forwarding entry.

SYNTAX:

<b>dns fwddelete</b>	<b>src = &lt;ip-address&gt;</b> <b>mask = &lt;ip-mask (dotted or cidr)&gt;</b> <b>[dns = &lt;ip-address&gt;]</b>
----------------------	--

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

RELATED COMMANDS:

<b>dns fwdadd</b>	Add a DNS forwarding entry.
<b>dns fwdlist</b>	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:

<b><i>dns fwdadd</i></b>	Add a DNS forwarding entry.
<b><i>dns fwddelete</i></b>	Delete a DNS forwarding entry.
<b><i>dns fwdtable</i></b>	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™ 510** is configured as DNS server.

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

RELATED COMMANDS:

**dns add**

Add a static DNS entry.

**dns delete**

Delete a DNS entry (via its index number).

## **dns nslookup**

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

SYNTAX:

<b>dns nslookup</b>	<b>lookup = &lt;string&gt;</b>
---------------------	--------------------------------

<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:

<b>dns list</b>	List current DNS entries.
-----------------	---------------------------

## **dns start**

Start **SpeedTouch™ 510** 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:

<b>dns status</b>	Show DNS server/forwarder configuration.
<b>dns stop</b>	Stop DNS server/forwarder.

## dns stats

Show **SpeedTouch™ 510** DNS server/forwarder statistics.

SYNTAX:

```
dns stats
```

EXAMPLE INPUT/OUTPUT IN A NETWORKED ENVIRONMENT:

The **SpeedTouch™ 510** is configured as DNS server.

```

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

```

RELATED COMMANDS:

**dns clrstats**

Clear DNS server/forwarder statistics.

## ***dns status***

Show **SpeedTouch™ 510** DNS server/forwarder configuration.

SYNTAX:

```
dns status
```

EXAMPLE OUTPUT:

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

## **dns stop**

Stop **SpeedTouch™ 510** 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:

<b>dns status</b>	Show DNS server/forwarder configuration.
<b>dns start</b>	Start DNS server/forwarder.

## **dns toutfwd**

Set DNS forwarding timeout.

### SYNTAX:

<b>dns toutfwd</b>	<b>timeout = &lt;number&gt;</b>
--------------------	---------------------------------

<i>timeout</i>	A number (seconds). Represents the query forwarding timeout. This parameter determines how long the <b>SpeedTouch™ 510</b> 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:

<b>dns fwdtable</b>	Show DNS forwarding table.
<b>dns fwdlist</b>	Show current DNS forwarding entries..
<b>dns fwdadd</b>	Add a DNS forwarding entry.
<b>dns fwddelete</b>	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:

<b>dns fwdtable</b>	Show DNS forwarding table.
<b>dns fwdlist</b>	Show current DNS forwarding entries..
<b>dns status</b>	Show DNS server/forwarder configuration.
<b>dns tron</b>	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:

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

## 7 Env Commands

**env (to access the Env level)**

**env def**

**env flush**

**env get**

**env list**

**env set**

**env unset**

## ***env def***

Define an environment variable.

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

SYNTAX:

<b><i>env def</i></b>
-----------------------

## env flush

Flush all non-system environment variables.

SYNTAX:

```
env flush
```

EXAMPLE:

```
=>env list
_COMPANY_NAME=Alcatel
_PROD_NAME=Speed Touch
_PROD_FRIENDLY_NAME=Speed Touch 510 ADSL Router
_PROD_NUMBER=510
_BOARD_SERIAL_NBR=012345678
_BUILD=R3.7.1.6
_BUILDNAME=Sascha3.716
_PRL=3EC18203CCAB
_BOARD_NAME=ADNT-N
_COMPANY_ID=ALCL
_COPYRIGHT=Copyright (c) 1999-2002 Alcatel
_MACADDR=00-90-D0-01-02-03
_UDN=uuid:UPnP-SpeedTouch510-1_00-90-D0-01-02-03
CONF_REGION=Alcatel Belgium
CONF_PROVIDER=ALCA_peckelbs
CONF_DESCRIPTION=Default Bridging configuration
CONF_SERVICE=Bridging on 0/35 and 8/35
CONF_DATE=07 Nov 2001
HOST_SETUP=user
VPI_VCI=8*35
Sys_password=Sascha
=>env flush
=>env list
_COMPANY_NAME=Alcatel
_PROD_NAME=Speed Touch
_PROD_FRIENDLY_NAME=Speed Touch 510 ADSL Router
_PROD_NUMBER=510
_BOARD_SERIAL_NBR=012345678
_BUILD=R3.7.1.6
_BUILDNAME=Sascha3.716
_PRL=3EC18203CCAB
_BOARD_NAME=ADNT-N
_COMPANY_ID=ALCL
_COPYRIGHT=Copyright (c) 1999-2002 Alcatel
_MACADDR=00-90-D0-01-02-03
_UDN=uuid:UPnP-SpeedTouch510-1_00-90-D0-01-02-03
=>
```

RELATED COMMANDS:

**env list**

List all environment variables.

## env get

Get the current value of a environment variable.

### SYNTAX:

<b>env get</b>	<b>var = &lt;string&gt;</b>
----------------	-----------------------------

<b>var</b>	The name of the environment variable. Execute <b>env list</b> to see a list of all environment variables.	<b>REQUIRED</b>
------------	--	-----------------

### EXAMPLE:

```
=>env list
_COMPANY_NAME=Alcatel
_PROD_NAME=Speed Touch
_PROD_FRIENDLY_NAME=Speed Touch 510 ADSL Router
_PROD_NUMBER=510
_BOARD_SERIAL_NBR=012345678
_BUILD=R3.7.1.6
_BUILDNAME=Sascha3.716
_PRL=3EC18203CCAB
_BOARD_NAME=ADNT-N
_COMPANY_ID=ALCL
_COPYRIGHT=Copyright (c) 1999-2002 Alcatel
_MACADDR=00-90-D0-01-02-03
_UDN=uuid:UPnP-SpeedTouch510-1_00-90-D0-01-02-03
_CONF_REGION=Alcatel Belgium
_CONF_PROVIDER=ALCA_peckelbs
_CONF_DESCRIPTION=Default Bridging configuration
_CONF_SERVICE=Bridging on 0/35 and 8/35
_CONF_DATE=07 Nov 2001
_HOST_SETUP=user
_VPI_VCI=8*35
Sys_password=Sascha
=>env get var=VPI_VCI
8*35
=>
```

### RELATED COMMANDS:

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

## **env list**

Show all currently available environment variables.

SYNTAX:

```
env list
```

EXAMPLE:

```
=>env list
_COMPANY_NAME=Alcatel
_PROD_NAME=Speed Touch
_PROD_FRIENDLY_NAME=Speed Touch 510 ADSL Router
_PROD_NUMBER=510
_BOARD_SERIAL_NBR=012345678
_BUILD=R3.7.1.6
_BUILDNAME=Sascha3.716
_PRL=3EC18203CCAB
_BOARD_NAME=ADNT-N
_COMPANY_ID=ALCL
_COPYRIGHT=Copyright (c) 1999-2002 Alcatel
_MACADDR=00-90-D0-01-02-03
_UDN=uuid:UPnP-SpeedTouch510-1_00-90-D0-01-02-03
CONF_REGION=Alcatel Belgium
CONF_PROVIDER=ALCA_peckelbs
CONF_DESCRIPTION=Default Bridging configuration
CONF_SERVICE=Bridging on 0/35 and 8/35
CONF_DATE=07 Nov 2001
HOST_SETUP=user
VPI_VCI=8*35
Sys_password=Sascha
=>
```

## env set

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

### SYNTAX:

<b>env set</b>	<b>var = &lt;string&gt;</b> <b>value = &lt;string&gt;</b>	
<i>var</i>	The name of the environment variable. When creating an environment variable, any name is allowed, however spaces are not allowed and the name may not start with "CONF", "HOST", an underscore "_" or the dollar sign "\$".	REQUIRED
<i>value</i>	A quoted string which defines the value of the environment variable. The value of system variables (built-in variables with names starting with an underscore "_", "CONF" or "HOST") can not be changed.	REQUIRED

### EXAMPLE:

```
=>env list
_COMPANY_NAME=Alcatel
_PROD_NAME=Speed Touch
_PROD_FRIENDLY_NAME=Speed Touch 510 ADSL Router
_PROD_NUMBER=510
_BOARD_SERIAL_NBR=012345678
_BUILD=R3.7.1.6
_BUILDNAME=Sascha3.716
_PRL=3EC18203CCAB
_BOARD_NAME=ADNT-N
_COMPANY_ID=ALCL
_COPYRIGHT=Copyright (c) 1999-2002 Alcatel
_MACADDR=00-90-D0-01-02-03
_UDN=uuid:UPnP-SpeedTouch510-1_00-90-D0-01-02-03
CONF_REGION=Alcatel Belgium
CONF_PROVIDER=ALCA_peckelbs
CONF_DESCRIPTION=Default Bridging configuration
CONF_SERVICE=Bridging on 0/35 and 8/35
CONF_DATE=07 Nov 2001
HOST_SETUP=user
=>env set var=VPI_VCI value="8*35"
=>env get var=VPI_VCI
8*35
=>env set var=VPI_VCI value="11*35"
=>env get var=VPI_VCI
11*35
=>
```

### RELATED COMMANDS:

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

## env unset

Delete a non-system environment variable.

### SYNTAX:

<b>env unset</b>	<b>var = &lt;string&gt;</b>
------------------	-----------------------------

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

### EXAMPLE:

```

=>env list
_COMPANY_NAME=Alcatel
_PROD_NAME=Speed Touch
_PROD_FRIENDLY_NAME=Speed Touch 510 ADSL Router
.....
_UDN=uuid:UPnP-SpeedTouch510-1_00-90-D0-01-02-03
CONF_REGION=Alcatel Belgium
CONF_PROVIDER=ALCA_peckelbs
CONF_DESCRIPTION=Default Bridging configuration
CONF_SERVICE=Bridging on 0/35 and 8/35
CONF_DATE=07 Nov 2001
HOST_SETUP=user
VPI_VCI=8*35
=>env unset var=VPI_VCI
=>env list
_COMPANY_NAME=Alcatel
_PROD_NAME=Speed Touch
_PROD_FRIENDLY_NAME=Speed Touch 510 ADSL Router
.....
_UDN=uuid:UPnP-SpeedTouch510-1_00-90-D0-01-02-03
CONF_REGION=Alcatel Belgium
CONF_PROVIDER=ALCA_peckelbs
CONF_DESCRIPTION=Default Bridging configuration
CONF_SERVICE=Bridging on 0/35 and 8/35
CONF_DATE=07 Nov 2001
HOST_SETUP=user
=>

```

### RELATED COMMANDS:

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



## 8 Firewall Commands

**firewall (to access the Firewall level)**  
**firewall assign**  
**firewall flush**  
**firewall list**  
**firewall match**  
**firewall troff**  
**firewall tron**  
**firewall chain (to access the Firewall Chain level)**  
**firewall chain create**  
**firewall chain delete**  
**firewall chain list**  
**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:

<b>firewall assign</b>	<b>hook = &lt;{input sink forward source output}&gt;</b> <b>chain = &lt;string&gt;</b>
------------------------	---

<i>hook</i>	The entry point's name to assign a chain to. Choose between: <ul style="list-style-type: none"> <li>▪ <b>input</b> : The point off all incoming traffic. At this point it can be determined whether the packet is allowed to reach the <b>SpeedTouch™ 510</b> IP router or local host.</li> <li>▪ <b>sink</b> : The point off all traffic destined to the <b>SpeedTouch™ 510</b> IP router itself. At this point it can be determined whether the packet is allowed to address the local host.</li> <li>▪ <b>forward</b> : The point off all traffic to be forwarded by the <b>SpeedTouch™ 510</b> IP router. At this point it can be determined whether the packet is allowed to be handled, i.e. routed.</li> <li>▪ <b>source</b> : The point off all traffic sourced by the <b>SpeedTouch™ 510</b> IP router. At this point it can be determined whether the packet is allowed to leave the local host.</li> <li>▪ <b>output</b> : The point off all outgoing traffic. At this point it can be determined whether the packet is allowed to leave the <b>SpeedTouch™ 510</b> IP router or local host.</li> </ul>	REQUIRED
<i>chain</i>	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:

<b>firewall chain create</b>	Create a chain.
<b>firewall chain list</b>	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:

<b>firewall flush</b>	<b>[hook = &lt;{input sink forward source output}&gt;]</b>
-----------------------	--

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

EXAMPLE:

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

RELATED COMMANDS:

**firewall assign** Assign a chain to an entry point.

## **firewall list**

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

SYNTAX:

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

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

EXAMPLE:

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

RELATED COMMANDS:

**firewall assign**  
**firewall flush**

Assign a chain to an entry point.  
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|ftfp|port}>]
                    [dstport = <{ftp|ftp-data|telnet|mail|smtp|dns|domain|ftfp|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 <b>bridge iflist</b> 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"><li>▪ <i>tcp</i></li><li>▪ <i>udp</i></li><li>▪ <i>icmp</i></li></ul> <p>Or specify the protocol number in accordance to the latest version of RFC1700: Assigned numbers.</p>	OPTIONAL
<i>[syn]</i>	<p>Set the TCP SYN flag (yes) or not (no).</p>	OPTIONAL
<i>[urg]</i>	<p>Set the TCP URG flag (yes) or not (no).</p>	OPTIONAL
<i>[ack]</i>	<p>Set the TCP ACK flag (yes) or not (no).</p>	OPTIONAL
<i>[srcport]</i>	<p>The TCP/UDP port the virtual packet is coming from. Choose between:</p> <ul style="list-style-type: none"><li>▪ <i>ftp</i></li><li>▪ <i>ftp-data</i></li><li>▪ <i>telnet</i></li><li>▪ <i>mail</i></li><li>▪ <i>smtp</i></li><li>▪ <i>dns</i></li><li>▪ <i>domain</i></li><li>▪ <i>tftp</i></li></ul> <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"><li>▪ <i>ftp</i></li><li>▪ <i>ftp-data</i></li><li>▪ <i>telnet</i></li><li>▪ <i>mail</i></li><li>▪ <i>smtp</i></li><li>▪ <i>dns</i></li><li>▪ <i>domain</i></li><li>▪ <i>tftp</i></li></ul> <p>Or specify the port number in accordance to the latest version of RFC1700: Assigned numbers.</p>	OPTIONAL

- [icmptype]* 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.
- [icmrcode]* 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=65535 dstport=telnet
  action=accept
:firewall rule create chain=Telnet index=1 srcintfgrp=wan src=200.200.200.1/32
  dst=10.0.0.0/8 prot=tcp srcport=telnet dstport=1024 dstportend=65535
  action=accept
:firewall rule create chain=Telnet index=2
  action=drop
=>firewall 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 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:

<b>firewall assign</b>	Assign a chain to an entry point.
<b>firewall chain delete</b>	Delete a chain.
<b>firewall chain list</b>	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:

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

## **firewall chain list**

Show a list of all current chains.

SYNTAX:

```
firewall chain list
```

EXAMPLE INPUT/OUTPUT:

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

RELATED COMMANDS:

<b>firewall assign</b>	Assign a chain to an entry point.
<b>firewall chain create</b>	Create a chain.
<b>firewall chain delete</b>	Delete a chain.

## firewall rule clear

Clear statistics for a given rule.

SYNTAX:

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

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

EXAMPLE:

```
=>firewall rule stats
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:

<b>firewall rule create</b>	Create a rule.
<b>firewall rule delete</b>	Delete a specified rule in a chain.
<b>firewall rule flush</b>	Delete all rules in a chain.
<b>firewall rule list</b>	Show a list of all (or a specified) chains' rules.
<b>firewall rule stats</b>	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)>]
[fos [!]= <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-querch|
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"> <li>▪ wan</li> <li>▪ local</li> <li>▪ lan</li> </ul> (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 <b>bridge iflist</b> 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"> <li>▪ wan</li> <li>▪ local</li> <li>▪ lan</li> </ul> (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"> <li>▪ tcp</li> <li>▪ udp</li> <li>▪ icmp</li> </ul> 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"> <li>▪ ftp</li> <li>▪ ftp-data</li> <li>▪ telnet</li> <li>▪ mail</li> <li>▪ smtp</li> <li>▪ dns</li> <li>▪ domain</li> <li>▪ tftp</li> </ul> <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"> <li>▪ ftp</li> <li>▪ ftp-data</li> <li>▪ telnet</li> <li>▪ mail</li> <li>▪ smtp</li> <li>▪ dns</li> <li>▪ domain</li> <li>▪ tftp</li> </ul> <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"> <li>▪ <i>echo-reply</i></li> <li>▪ <i>destination-unreachable</i></li> <li>▪ <i>source-quench</i></li> <li>▪ <i>redirect</i></li> <li>▪ <i>echo-request</i></li> <li>▪ <i>router-advertisement</i></li> <li>▪ <i>router-solicitation</i></li> <li>▪ <i>time-exceeded</i></li> <li>▪ <i>parameter-problems</i></li> <li>▪ <i>timestamp-request</i></li> <li>▪ <i>timestamp-reply</i></li> <li>▪ <i>information-request</i></li> <li>▪ <i>information-reply</i></li> <li>▪ <i>address-mask-request</i></li> <li>▪ <i>address-mask-reply</i></li> </ul> <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"> <li>▪ <b>accept</b> : the packet may pass.</li> <li>▪ <b>deny</b> : ICMP error destination unreachable. An error message is sent back to the sender.</li> <li>▪ <b>drop</b> : packet disappears. It is silently dropped, that is, without sending an error message to the sender.</li> <li>▪ <b>count</b> : update of statistics. Has no influence on the packet.</li> </ul>	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=65535 dstport=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=65535 dstport=telnet
  action=accept
:firewall rule create chain=Telnet index=1 srcintfgrp=wan src=200.200.200.1/32
  dst=10.0.0.0/8 prot=tcp srcport=telnet dstport=1024 dstportend=65535
  action=accept
:firewall rule create chain=Telnet index=2
  action=drop
=>

```

## RELATED COMMANDS:

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

## firewall rule delete

Delete a rule.

SYNTAX:

<b>firewall rule delete</b>	<b>chain = &lt;string&gt;</b> <b>index = &lt;number&gt;</b>
-----------------------------	--

<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 <b>firewall rule list</b> first to determine the index number of the applicable rule.	REQUIRED

EXAMPLE:

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

RELATED COMMANDS:

<b>firewall rule clear</b>	Clear statistics of a given rule.
<b>firewall rule create</b>	Create a rule.
<b>firewall rule flush</b>	Delete all rules in a chain.
<b>firewall rule list</b>	Show a list of all (or a specified) chains' rules.
<b>firewall rule stats</b>	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=65535 dstport=telnet
  action=accept
:firewall rule create chain=Telnet index=1 srcintfgrp=wan src=200.200.200.1/32
  dst=10.0.0.0/8 prot=tcp srcport=telnet dstport=1024 dstportend=65535
  action=accept
:firewall rule create chain=Telnet index=2 action=drop
=>firewall rule flush chain=Telnet
=>firewall rule list chain=Telnet
=>
```

RELATED COMMANDS:

<b>firewall rule clear</b>	Clear statistics of a given rule.
<b>firewall rule create</b>	Create a rule.
<b>firewall rule delete</b>	Delete a specified rule.
<b>firewall rule list</b>	Show a list of all (or a specified) chains' rules.
<b>firewall rule stats</b>	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=65535 dstport=telnet
action=accept
:firewall rule create chain=Telnet index=1 srcintfgrp=wan src=200.200.200.1/32
dst=10.0.0.0/8 prot=tcp srcport=telnet dstport=1024 dstportend=65535
action=accept
:firewall rule create chain=Telnet index=2 action=drop
=>firewall rule list
:firewall rule create chain=source index=0 dstintfgrp=!wan action=accept
:firewall rule create chain=source index=1 prot=udp dstport=dns action=accept
:firewall rule create chain=source index=2 prot=udp dstport=67 action=accept
:firewall rule create chain=source index=3 action=drop
:firewall rule create chain=forward index=0 srcintfgrp=wan dstintfgrp=wan action=drop
:firewall rule create chain=sink index=0 srcintf=eth0 srcbridgeport=1 action=accept
:firewall rule create chain=sink index=1 srcintfgrp=!wan action=accept
:firewall rule create chain=sink index=2 prot=udp dstport=dns action=accept
:firewall rule create chain=sink index=3 prot=udp dstport=68 action=accept
:firewall rule create chain=sink index=4 action=drop
:firewall rule create chain=Telnet index=0 srcintfgrp=lan src=10.0.0.0/8
dst=200.200.200.1/32 prot=tcp srcport=1024 srcportend=65535 dstport=telnet
action=accept
:firewall rule create chain=Telnet index=1 srcintfgrp=wan src=200.200.200.1/32
dst=10.0.0.0/8 prot=tcp srcport=telnet dstport=1024 dstportend=65535
action=accept
:firewall rule create chain=Telnet index=2 action=drop
=>
```

RELATED COMMANDS:

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

## firewall rule stats

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

SYNTAX:

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

[chain]                    The name of the chain of which the statistics must be listed.                    OPTIONAL  
In case this parameter is not specified the statistics for the rules applicable to all chains are shown.

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

EXAMPLE OUTPUT:

```
=>firewall rule list chain=Test
:firewall rule create chain=Test index=0 srcintfgrp=lan src=200.200.0.1/32
  dst=200.200.0.2/32 prot=udp srcport=0 srcportend=65535 dstport=telnet
  action=deny
=>firewall rule clear
=>firewall rule stats
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™ 510**. 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™ 510** 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=10.10.1.1/32 prot=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:

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

## 9 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 ping**  
**ip rtadd**  
**ip rtdelete**  
**ip rtlist**  
**ip sendto**

## **ip apadd**

Assign an IP address to an interface.

### SYNTAX:

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

<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 brcastpkts:290555
  IPTX bytes:839550 unicastpkts:11477 brcastpkts:0 dropkts:0
  HWRX bytes:0 unicastpkts:0 brcastpkts:0
  HWTX bytes:0 unicastpkts:0 brcastpkts: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 brcastpkts:2
  IPTX bytes:0 unicastpkts:0 brcastpkts:0 dropkts:0
  HWRX bytes:0 unicastpkts:0 brcastpkts:0
  HWTX bytes:0 unicastpkts:0 brcastpkts: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 brcastpkts:0
  IPTX bytes:0 unicastpkts:0 brcastpkts:0 dropkts:0
  HWRX bytes:0 unicastpkts:0 brcastpkts:0
  HWTX bytes:0 unicastpkts:0 brcastpkts: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 brcastpkts:290669
  IPTX bytes:853114 unicastpkts:11662 brcastpkts:0 dropkts:0
  HWRX bytes:0 unicastpkts:0 brcastpkts:0
  HWTX bytes:0 unicastpkts:0 brcastpkts: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 brcastpkts:2
  IPTX bytes:0 unicastpkts:0 brcastpkts:0 dropkts:0
  HWRX bytes:0 unicastpkts:0 brcastpkts:0
  HWTX bytes:0 unicastpkts:0 brcastpkts:0 dropkts:0
=>

```

## RELATED COMMANDS:

**ip adelete**

Remove an IP address from an interface.

**ip aplist**

Show current IP addresses.

## ip apdelete

Remove an IP address from an interface.

### SYNTAX:

<b>ip apdelete</b>	<b>addr = &lt;ip-address&gt;</b>
--------------------	----------------------------------

<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:

<b>ip apadd</b>	Add an IP address to an interface.
<b>ip aplist</b>	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:10.0.0.2          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
=>

```

RELATED COMMANDS:

**ip apadd**

Add an IP address to an interface.

**ip apdelete**

Remove an IP address from an interface.

## ip arpadd

Add a static entry to the **SpeedTouch™ 510** ARP cache.

### SYNTAX:

<b>ip arpadd</b>	<b>intf = &lt;interface name&gt;</b> <b>ip = &lt;ip-address&gt;</b> <b>[hwaddr = &lt;hardware-address&gt;]</b>
------------------	--

<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:

<b>ip arpdelete</b>	Delete an ARP entry.
<b>ip arplist</b>	Show current ARP cache.

## ip arpdelete

Remove an entry from the **SpeedTouch™ 510** ARP cache.

### SYNTAX:

<b>ip arpdelete</b>	<b>intf = &lt;interface name&gt;</b> <b>ip = &lt;ip-address&gt;</b> <b>[hwaddr = &lt;hardware-address&gt;]</b>
---------------------	--

<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
<u>eth0</u>	<u>10.0.0.2</u>	<u>00:10:a4:d0:9a:db</u>	<u>STATIC</u>
=>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
<u>eth0</u>	<u>10.0.0.2</u>	<u>00:10:a4:d0:9a:db</u>	<u>STATIC</u>
=>			

### RELATED COMMANDS:

<b>ip arpadd</b>	Add a static ARP entry.
<b>ip arplist</b>	Show current ARP cache.

## **ip arplist**

Show the **SpeedTouch™ 510** 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:

<b>ip config</b>	<pre>[forwarding = &lt;{off on}&gt;] [firewalling = &lt;{off on}&gt;] [redirects = &lt;{off on}&gt;] [sourcerouting = &lt;{off on}&gt;] [netbroadcasts = &lt;{off on}&gt;] [ttl = &lt;number{0-255}&gt;] [fraglimit = &lt;number{1-1024}&gt;] [defragmode = &lt;{normal always nat}&gt;] [addrcheck = &lt;{off own static dynamic}&gt;] [mssclamping = &lt;{off on}&gt;]</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 <b>SpeedTouch™ 510</b> firewall. For security reasons this parameter is enabled per default. It is strongly recommended never to disable the <b>SpeedTouch™ 510</b> firewall.	OPTIONAL
[redirects]	Disable (off) or enable (on) the sending of ICMP redirect messages. A router can send a redirect message in case a shorter path than the path followed is discovered. For security reasons this parameter is disabled per default.	OPTIONAL
[sourcerouting]	Disallow (off) or allow (on) IP source routed packets. IP source routed packets are packets with the route to follow specified in the header. For security reasons this parameter is disabled per default.	OPTIONAL
[netbroadcasts]	Disallow (off) or allow (on) net directed broadcasts. This parameter is per default disabled. In case netbroadcasts are allowed no traces of netbroadcasts are generated.	OPTIONAL
[ttl]	A number between 0 and 255. Represents the default time-to-live (ttl) for locally generated IP packets. This parameter determines the number of hop-counts the IP packet may pass before it is dropped. Generally the time-to-live is 64 hop-counts. By limiting the time-to-live continuous circulation of IP packets on the network without ever reaching a destination is avoided.	OPTIONAL
[fraglimit]	A number between 1 and 1024. Represents the maximum number of IP packet fragments waiting for completion. Generally the fragmentation limit is 64. By limiting the fragmentation limit the depletion of the buffer is avoided.	OPTIONAL

[defragmode]	Define which packets are reassembled under which circumstances. Choose between: <ul style="list-style-type: none"><li>▪ <b>normal</b> Packets to be forwarded will not be reassembled. Packets with local destination, i.e. destined for the <b>SpeedTouch™ 510</b>, are reassembled.</li><li>▪ <b>always</b> Packets are always reassembled.</li><li>▪ <b>nat</b> Same behaviour as <b>normal</b> 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.</li></ul>	OPTIONAL
[addrcheck]	Set the level of IP address checks. Choose between: <ul style="list-style-type: none"><li>▪ <b>off</b> No address checking is performed. For advanced users only; in normal circumstances there should always be some kind of address checking.</li><li>▪ <b>own</b> Minimum level of checking. Only the address configuration on the <b>SpeedTouch™ 510</b> is checked.</li><li>▪ <b>static</b> Checking of the address configuration of the <b>SpeedTouch™ 510</b> 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).</li><li>▪ <b>dynamic</b> Besides the address configuration of the <b>SpeedTouch™ 510</b> 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.</li></ul>	OPTIONAL
[mssclamping]	Disable (off) or enable (on) mss clamping for low mtu interfaces. Mss clamping assures that the size of a TCP packet never exceeds the available mtu of the outgoing interface. It is recommended not to disable this parameter.	OPTIONAL

## EXAMPLE:

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

## ip ifconfig

Configure interface parameters.

SYNTAX:

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

<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"> <li>▪ down</li> <li>▪ up</li> </ul>	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:c
f
2 NewMer    0    1500  0        0        0        UP      00:80:9f:24:ab:cf
5 cip0      0    9180  0        0        0        UP
=>
```

RELATED COMMANDS:

**ip ifconfig**

Configure interface parameters.

## ip ping

Send ICMP ECHO\_REQUEST packets.

SYNTAX:

<b>ip ping</b>	<b>addr = &lt;ip-address&gt;</b> <b>[count = &lt;number{1-1000000}&gt;]</b> <b>[size = &lt;number{1-20000}&gt;]</b> <b>[interval = &lt;number{100-1000000}&gt;]</b> <b>[listen = &lt;{off on}&gt;]</b>
----------------	--

<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
=>
```

RELATED COMMANDS:

**ip sendto**                      Send UDP packets.

## ip rtadd

Add a route to the **SpeedTouch™ 510** routing table.

### SYNTAX:

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

<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:

<b>ip rtdelete</b>	Remove a route from the routing table.
<b>ip rtlist</b>	Show current routing table.

## ip rtdelete

Delete a route from the **SpeedTouch™ 510** routing table.

### SYNTAX:

<b>ip rtdelete</b>	<b>dst = &lt;ip-address&gt;</b> <b>[dstmsk = &lt;ip-mask(dotted or cidr)&gt;]</b> <b>[src = &lt;ip-address&gt;]</b> <b>[srcmsk = &lt;ip-mask(dotted or cidr)&gt;]</b> <b>[gateway = &lt;ip-address&gt;]</b> <b>[intf = &lt;interface name&gt;]</b>
--------------------	---

<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:

<b>ip rtadd</b>	Add a route to the routing table.
<b>ip rtlist</b>	Show current routing table.

## **ip rtlist**

Show current **SpeedTouch™ 510** 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 rdelete**

Remove a route from the routing table.

## ip sendto

Send UDP packets.

### SYNTAX:

<b>ip sendto</b>	<b>addr = &lt;ip-address&gt;</b> <b>[count = &lt;number{1-1000000}&gt;]</b> <b>[size = &lt;number{1-20000}&gt;]</b> <b>[interval = &lt;number{100-1000000}&gt;]</b> <b>[listen = &lt;{off on}&gt;]</b> <b>[srcport = &lt;number{1-65535}&gt;]</b> <b>dstport = &lt;number{1-65535}&gt;</b>
------------------	--

<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.



## 10 MER Commands

**mer (to access the MER level)**

**mer flush**

**mer ifadd**

**mer ifattach**

**mer ifconfig**

**mer ifdelete**

**mer ifdetach**

**mer iflist**

## ***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
=>
```

## **mer ifadd**

Create a new MER interface.

### SYNTAX:

<b>mer ifadd</b>	<b>[intf = &lt;string&gt;]</b> <b>[dest = &lt;phonebook entry&gt;]</b>
------------------	---

<b>[intf]</b>	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
<b>[dest]</b>	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:

<b>mer ifattach</b>	Attach a MER interface.
<b>mer ifconfig</b>	Configure a MER interface.
<b>mer ifdelete</b>	Delete a MER interface.
<b>mer ifdetach</b>	Detach a MER interface.
<b>mer iflist</b>	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:

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

## mer ifconfig

Configure a MER interface.

SYNTAX:

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

<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"> <li>▪ llc/snap</li> <li>▪ vcmux</li> </ul>	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:

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

## **mer ifdelete**

Delete a MER interface.

SYNTAX:

<b>mer ifdelete</b>	<b>intf = &lt;ifname&gt;</b>	
<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:

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

## mer ifdetach

Detach a MER interface.

SYNTAX:

<b>mer ifdetach</b>	<b>intf = &lt;iframe&gt;</b>	
<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:

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

## **mer iflist**

Show all or a specified MER interface(s).

### SYNTAX:

<b>mer iflist</b>	<b>[intf = &lt;ifname&gt;]</b>
-------------------	--------------------------------

<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:

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

## 11 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 unbind**

## **nat applist**

List available NATP protocol helpers.

Certain protocols are 'sensitive' to NATP in that they do not function properly when dealing with it. This list shows which 'NAPT-sensitive' applications are supported on the **SpeedTouch™ 510**, i.e. the inherent knowledge of the **SpeedTouch™ 510** on this matter.

SYNTAX:

```
nat applist
```

EXAMPLE OUTPUT:

```

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

```

RELATED COMMANDS:

<b>nat bind</b>	Create a new helper/port binding.
<b>nat bindlist</b>	List current NATP helper/port bindings.
<b>nat unbind</b>	Delete an existing helper/port binding.

## nat bind

Create a new helper/port binding.

SYNTAX:

<b>nat bind</b>	<b>application = &lt;string&gt;</b> <b>port = &lt;TCP/UDP service name or port number&gt;</b>
-----------------	--

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

EXAMPLE INPUT:

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

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

## **nat bindlist**

List current NATP 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 NATP protocol helpers.

**nat bind**

Create a new NATP helper/port binding.

**nat unbind**

Delete an existing helper/port binding.

## nat create

Create a static NAT entry. Typically used to install specific servers behind the **SpeedTouch™ 510's** NAT device.

### SYNTAX:

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

<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 <b>SpeedTouch™ 510's</b> NAT 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 NAT enabled WAN IP address of the <b>SpeedTouch™ 510</b> . Use '0' to create a template. Such template will then be valid for any of <b>SpeedTouch™ 510's</b> NAT 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:Port Outside-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
=>

```

## 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™ 510** 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:

<b>nat defserver</b>	<b>[addr = &lt;ip-address&gt;]</b>
----------------------	------------------------------------

<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
---------------	---	----------

EXAMPLE INPUT/OUTPUT:

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

## nat delete

Delete a static NAT entry.

SYNTAX:

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

<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:Port      Outside-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     10.0.0.138:138    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:Port      Outside-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:

<b>nat create</b>	Create a static NAT entry.
<b>nat disable</b>	Disable NAT on one of the <b>SpeedTouch™ 510</b> IP addresses.
<b>nat enable</b>	Enable NAT on one of the <b>SpeedTouch™ 510</b> IP addresses.
<b>nat list</b>	List NAT connection database.

## nat disable

Disable NAT on a **SpeedTouch™ 510** IP address.

SYNTAX:

<b>nat disable</b>	<b>addr = &lt;ip-address&gt;</b>
--------------------	----------------------------------

<i>addr</i>	One of <b>SpeedTouch™ 510</b> '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
StateControl
1      6      10.0.0.138:80      172.16.0.5:1080      0.0.0.0:0          19   8   9
2      17     10.0.0.138:138     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
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:

<b>nat create</b>	Create a static NAT entry.
<b>nat delete</b>	Delete a static NAT entry.
<b>nat enable</b>	Enable NAT on one of the <b>SpeedTouch™ 510</b> IP addresses.
<b>nat list</b>	List NAT connection database.

## nat enable

Enable NAT on a **SpeedTouch™ 510** IP address.

SYNTAX:

<b>nat enable</b>	<b>addr = &lt;ip-address&gt;</b> <b>[type = &lt;{none pat}&gt;]</b>
-------------------	--

<i>addr</i>	The <b>SpeedTouch™ 510</b> 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 brcastpkts:290555
  IPTX bytes:839550 unicastpkts:11477 brcastpkts:0 droppkts:0
  HWRX bytes:0 unicastpkts:0 brcastpkts:0
  HWTX bytes:0 unicastpkts:0 brcastpkts:0 droppkts:0
0 loop Type:0
  inet addr:127.0.0.1 Bcast:127.255.255.255 Mask:255.0.0.0
  UP RUNNING MTU:1500 ReasmMAX:65535 Group:1
  IPRX bytes:116 unicastpkts:0 brcastpkts:2
  IPTX bytes:0 unicastpkts:0 brcastpkts:0 droppkts:0
  HWRX bytes:0 unicastpkts:0 brcastpkts:0
  HWTX bytes:0 unicastpkts:0 brcastpkts:0 droppkts:0
=>nat enable addr=10.10.10.147 type=pat
=>ip aplist
1 eth0 Type: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:

<b>nat create</b>	Create a static NAT entry.
<b>nat delete</b>	Delete a static NAT entry.
<b>nat disable</b>	Disable NAT on one of the <b>SpeedTouch™ 510</b> IP addresses.
<b>nat list</b>	List NAT connection database.

## **nat flush**

Flush complete NAT configuration.

The flush command does not impact previously saved configurations.

SYNTAX:

```
nat flush
```

EXAMPLE:

```

=>nat list
Indx Prot Inside-addr:Port      Outside-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:Port      Outside-addr:Port      Foreign-addr:Port Flgs Expir
StateControl
=>nat bindlist
=>

```

## nat list

Show NAPT connection database.

SYNTAX:

<b>nat list</b>	<b>[addr = &lt;ip-address&gt;]</b>
-----------------	------------------------------------

<i>[addr]</i>	The <b>SpeedTouch™ 510</b> 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
---------------	---	----------

EXAMPLE INPUT/OUTPUT:

```
=>nat list
Indx Prot Inside-addr:Port      Outside-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:

<b>nat create</b>	Create a static NAPT entry.
<b>nat delete</b>	Delete a static NAPT entry.
<b>nat disable</b>	Disable NAPT on one of the <b>SpeedTouch™ 510</b> IP addresses.
<b>nat enable</b>	Enable NAPT on one of the <b>SpeedTouch™ 510</b> IP addresses.

## **nat unbind**

Delete an existing helper/port binding.

### SYNTAX:

<b>nat unbind</b>	<b>application = &lt;string&gt;</b> <b>port = &lt;TCP/UDP service name or port number&gt;</b>
-------------------	--

<i>application</i>	The name of a NAPT application helper. The name must be spelled exactly as listed in the application list ( <b>nat applist</b> ).	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          OUTGOING    INCOMING
H323        tcp    1720     OUTGOING    INCOMING
RAUDIO(PNA) tcp    7070     OUTGOING
RTSP        tcp    554      OUTGOING
IRC         tcp    6667     OUTGOING
FTP         tcp    21       OUTGOING    INCOMING
=>nat bindlist
Application  Proto  Port
RAUDIO(PNA) tcp    7071
H323        tcp    1720
FTP         tcp    21
RTSP        tcp    554
IRC         tcp    6667
RAUDIO(PNA) tcp    7070
=>
=>nat unbind application=RAUDIO(PNA) port=7071
=>nat bindlist
Application  Proto  Port
H323        tcp    1720
FTP         tcp    21
RTSP        tcp    554
IRC         tcp    6667
RAUDIO(PNA) tcp    7070
=>
```

### RELATED COMMANDS:

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



## 12 Phonebook Commands

**phonebook (to access the Phonebook level)**

**phonebook add**

**phonebook autolist**

**phonebook delete**

**phonebook flush**

**phonebook list**

## 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:

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

<b>name</b>	A free to choose phonebook name for the destination. Two limitations apply: <ul style="list-style-type: none"> <li>▪ The name of a phonebook entry intended for the <b>Relayed PPPoA</b> (PPPoA-to-PPTP Relaying) packet service may not start with capital <b>P</b> or capital <b>T</b></li> <li>▪ The name of a phonebook entry intended for the <b>PPP-to-DHCP spoofing</b> packet service must start with DHCP, e.g. 'DHCP_Spoof01'.</li> </ul>	<b>REQUIRED</b>
<b>addr</b>	The ATM address for this destination. It is composed of a Virtual Path Identifier (VPI) and a Virtual Channel Identifier (VCI) identifying ATM virtual channels. In most cases the values are provided by the Service Provider. Accepted VPI: a number between 0 and 15 Accepted VCI: a number between 0 and 511.	<b>REQUIRED</b>
<b>type</b>	The Connection Service supported by the destination. Choose between: <ul style="list-style-type: none"> <li>▪ any (All Packet Services)</li> <li>▪ bridge (Bridging, Routed Ethernet, Bridged PPPoE, Routed PPPoE)</li> <li>▪ ppp (Routed PPPoA and Relayed PPPoA)</li> <li>▪ cip (Classical IP &amp; IP Routing)</li> <li>▪ ans (ATM Name Service)</li> <li>▪ pptp (Relayed PPPoA, PPPoA-to-PPTP Relaying).</li> </ul>	<b>REQUIRED</b>

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**  
**phonebook list**

Remove a phonebook entry.  
Show current phonebook.

## ***phonebook autolist***

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

SYNTAX:

```
phonebook autolist
```

EXAMPLE INPUT/OUTPUT:

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

RELATED COMMANDS:

**phonebook list**                      Show current phonebook.

## phonebook delete

Remove an unused phonebook entry.

### SYNTAX:

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

<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 <b>phonebook list</b> 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:

<b>phonebook add</b>	Add a phonebook entry.
<b>phonebook list</b>	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
=>
```

## **phonebook list**

Show current phonebook.

SYNTAX:

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

[opt]

Select output format. For internal use only.

OPTIONAL

EXAMPLE INPUT/OUTPUT:

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



## 13 PPP Commands

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

## **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
=>
```

## **ppp ifadd**

Create a new PPP interface.

SYNTAX:

<b>ppp ifadd</b>	<b>[<i>intf</i> = &lt;string&gt;] [<i>dest</i> = &lt;phonebook entry&gt;] [<i>encaps</i> = &lt;{vcmux llc}&gt;] [<i>speed</i> = &lt;number{4800–10000000}&gt;]</b>
------------------	--

<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"> <li>▪ vcmux</li> <li>▪ llc/snap</li> </ul>	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:

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

## ppp ifattach

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

SYNTAX:

<b>ppp ifattach</b>	<b>intf = &lt;ifname&gt;</b>	
<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:

<b>ppp ifadd</b>	Create a PPP interface.
<b>ppp ifconfig</b>	Configure a PPP interface.
<b>ppp ifdelete</b>	Delete a PPP interface.
<b>ppp ifdetach</b>	Detach a PPP interface.
<b>ppp iflist</b>	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:

<b>ppp ifconfig</b>	<pre> <b>intf</b> = &lt;ifname&gt; [<b>dest</b> = &lt;phonebook entry&gt;] [<b>user</b> = &lt;string&gt;] [<b>password</b> = &lt;string&gt;] [<b>qos</b> = &lt;string&gt;] [<b>proto</b> = &lt;{pppoa pppoe}&gt;] [<b>acname</b> = &lt;string&gt;] [<b>servicename</b> = &lt;string&gt;] [<b>encaps</b> = &lt;{vcmux llc}&gt;] [<b>pcomp</b> = &lt;{off on}&gt;] [<b>accomp</b> = &lt;{on off negotiate}&gt;] [<b>trace</b> = &lt;{off on}&gt;] [<b>pap</b> = &lt;{off on}&gt;] [<b>restart</b> = &lt;{off on}&gt;] [<b>retryinterval</b> = &lt;number{0-65535}&gt;] [<b>passive</b> = &lt;{off on}&gt;] [<b>silent</b> = &lt;{off on}&gt;] [<b>echo</b> = &lt;{off on}&gt;] [<b>mru</b> = &lt;number{293-8192}&gt;] [<b>laddr</b> = &lt;ip-address&gt;] [<b>raddr</b> = &lt;ip-address&gt;] [<b>netmask</b> = &lt;ip-mask(dotted or cidr)&gt;] [<b>format</b> = &lt;{dotted cidr}&gt;] [<b>pool</b> = &lt;{poolname none}&gt;] [<b>savepwd</b> = &lt;{off on}&gt;] [<b>demanddial</b> = &lt;{off on}&gt;] [<b>primdns</b> = &lt;ip-address&gt;] [<b>secdns</b> = &lt;ip-address&gt;] [<b>idle</b> = &lt;number{0-1000000}&gt;] [<b>addrtrans</b> = &lt;{none pat}&gt;] [<b>unnumbered</b> = &lt;{off on}&gt;] [<b>poolstart</b> = &lt;ip-address&gt;] [<b>poolend</b> = &lt;ip-address&gt;] [<b>status</b> = &lt;{down up}&gt;] </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"> <li>▪ PPPoA (ppp) phonebook entries For the Routed PPPoA (PPP &amp; IP Routing) packet service.</li> <li>▪ ETHoA (bridge) phonebook entries For the Routed PPPoE packet service.</li> </ul>	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"> <li>▪ pppoa For a Routed PPPoA (PPP &amp; IP Routing) connections.</li> <li>▪ pppoe For a Routed PPPoE connection.</li> </ul> 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 <b>ppp ifscan</b> 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 <b>ppp ifscan</b> 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"> <li>▪ vcmux (default) Standard encapsulation method for PPPoA (ppp) frames.</li> <li>▪ llc Standard encapsulation method for ETHoA (bridge) frames.</li> </ul>	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 <b>SpeedTouch™ 510</b> 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 <b>SpeedTouch™ 510</b> PPP session IP address. If not specified, the <b>SpeedTouch™ 510</b> 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 <b>SpeedTouch™ 510</b> will accept any IP address. Typically the remote IP address parameter is not specified.	OPTIONAL
<i>[netmask]</i>	The subnetmask associated with this address. Specifying a subnetmask forces the remote side (if it allows to) to accept this subnetmask as the PPP session subnetmask. If not specified, the <b>SpeedTouch™ 510</b> will accept any subnetmask. The <b>SpeedTouch™ 510</b> will only request/accept a subnetmask if a DHCP server pool is associated, i.e. if the <i>[pool]</i> parameter is specified.	OPTIONAL
<i>[format]</i>	The negotiated subnetmask specified in the netmask parameter is specified in the dotted format (dotted) or in CIDR format (cidr). Per default the format is CIDR.	OPTIONAL
<i>[pool]</i>	The name of the free DHCP server pool to which the acquired IP subnet must be assigned to.	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 <b>SpeedTouch™ 510</b> will negotiate this IP address with the remote side. If not specified, the <b>SpeedTouch™ 510</b> 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 <b>SpeedTouch™ 510</b> will negotiate this IP address with the remote side. If not specified, the <b>SpeedTouch™ 510</b> 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
<i>[addrtrans]</i>	Automatically enable address translation for the IP address of this link (pat) or do not use address translation (none).	OPTIONAL
<i>[unnumbered]</i>	Takes the local IP address from 'laddr' field and remote IP address from the IP address pool assigned to the incoming PPP link. In case the unnumbered parameter is disabled the same IP address is used for each connection on the server side, thus reducing the number of used IP addresses.	OPTIONAL
<i>[poolstart]</i>	The lower bound of the IP address pool assigned to the incoming PPP link.	OPTIONAL
<i>[poolend]</i>	The upper bound of the IP address pool assigned to the incoming PPP link.	OPTIONAL
<i>[status]</i>	Force automatically to attach the PPP interface (up) or use the regular <b>ppp ifattach</b> command (down). Per default the startup status is down (recommended).	OPTIONAL

## EXAMPLE:

```

=>ppp iflist
PPP1:  dest : PPP1
  Retry: 10  QoS  default encaps LLC
  mode = IP Routing
  flags= echo 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=pppoa 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:

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

## ppp ifdelete

Delete a PPP interface.

SYNTAX:

<b>ppp ifdelete</b>	<b>intf = &lt;ifname&gt;</b>
---------------------	------------------------------

*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:

<b>ppp ifadd</b>	Create a PPP interface.
<b>ppp ifattach</b>	Attach a PPP interface.
<b>ppp ifconfig</b>	Configure a PPP interface.
<b>ppp ifdetach</b>	Detach a PPP interface.
<b>ppp iflist</b>	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:

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

## ppp iflist

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

SYNTAX:

<b>ppp iflist</b>	<b>[intf = &lt;ifname&gt;]</b>
-------------------	--------------------------------

<i>intf</i>	the name of the PPP interface. In case this parameter is not specified all PPP interfaces are shown.	OPTIONAL
-------------	---	----------

EXAMPLE INPUT/OUTPUT :

```
=>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:

<b>ppp ifadd</b>	Create a PPP interface.
<b>ppp ifattach</b>	Attach a PPP interface.
<b>ppp ifconfig</b>	Configure a PPP interface.
<b>ppp ifdelete</b>	Delete a PPP interface.
<b>ppp ifdetach</b>	Detach a PPP interface.

## 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:

<b>ppp ifscan</b>	<b>intf = &lt;ifname&gt;</b> <b>[time = &lt;number{0-36000}&gt;]</b> <b>[kit = &lt;number{0-8}&gt;]</b>
-------------------	---

<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 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:

<b>ppp rtadd</b>	<b>intf = &lt;ifname&gt;</b> <b>dst = &lt;ip-address&gt;</b> <b>[dstmsk = &lt;ip-mask(dotted or cidr)&gt;]</b> <b>[src = &lt;ip-address&gt;]</b> <b>[srcmsk = &lt;ip-mask(dotted or cidr)&gt;]</b> <b>[metric = &lt;number{0-100}&gt;]</b>
------------------	---

<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"> <li>▪ Any remote destination is reachable, i.e. the PPP connection acts as default route (dstmsk=0)</li> <li>▪ Only the remote (sub)net is reachable (dstmsk=1) The actual destination mask will be the default netmask applicable for destination IP address</li> <li>▪ Only the single remote host is reachable (dstmsk=32)</li> <li>▪ Any valid (contiguous) netmask in case of VLSM.</li> </ul>	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"> <li>▪ Everybody is allowed to use this PPP connection (dstmsk=0)</li> <li>▪ 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.</li> <li>▪ Only the host which opened the PPP connection is allowed to use the PPP connection. (dstmsk=32)</li> <li>▪ Any valid (contiguous) netmask in case of VLSM.</li> </ul>	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:

<b>ppp rtdelete</b>	<b>intf = &lt;ifname&gt;</b>
---------------------	------------------------------

<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:

<b>ppp rtadd</b>	Configure a route specification for an upcoming PPP link.
------------------	---



## 14 PPTP Commands

**pptp (to access the PPTP level)**

**pptp flush**

**pptp list**

**pptp profadd**

**pptp profdelete**

**pptp proflist**

## **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
=>
```

## **pptp list**

Show current PPTP configuration.

SYNTAX:

```
pptp list
```

EXAMPLE INPUT/OUTPUT:

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

## **pptp profadd**

Define a new PPTP profile.

### SYNTAX:

<b>pptp profadd</b>	<b>name = &lt;string&gt;</b> <b>[qos = &lt;string&gt;]</b> <b>[encaps = &lt;{vcmux nlpid}&gt;]</b> <b>[ac = &lt;{never always keep}&gt;]</b>
---------------------	---

<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"> <li>▪ vc mux</li> <li>▪ nlpid</li> </ul>	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  vc mux      never
=>
```

### RELATED COMMANDS:

<b>pptp profdelete</b>	Delete a PPTP profile.
<b>pptp proflist</b>	Show current PPTP profiles.

## **pptp profdelete**

Delete a PPTP profile.

### SYNTAX:

<b>pptp profdelete</b>	<b>name &lt;string&gt;</b>
------------------------	----------------------------

<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:

<b>pptp profadd</b>	Define a new PPTP profile.
<b>pptp proflist</b>	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.

## 15 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 : Sascha3.716           Passive : Bene3.716
=>software deletepassive
=>Software version
Active : Sascha3.426           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™ 510** 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 : Sascha3.716                      Passive : Bene3.716
=>Software deletepassive
=>Software version
Active : Sascha3.716                      Passive :
.....
(FTP file transfer or upload via the SpeedTouch™ 510 pages of new software Sascha3.722)
.....
=>software setpassive file=Sascha3.722
=>Software version
Active : Sascha3.716                      Passive : Sascha3.722
=>
```

RELATED COMMANDS:

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

## software switch

Switch active and passive versions and reboot the **SpeedTouch™ 510**.

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 : Sascha3.716           Passive : Sascha3.722
=>software switch
.....
(after reboot and re-opening the Telnet session)
.....
=>Software version
Active : Sascha3.722           Passive : Sascha3.716
=>
```

RELATED COMMANDS:

**software version**  
**system reboot**

Show active and passive software versions.  
Reboot the **SpeedTouch™ 510**.

## **software version**

Show active and passive software versions.

SYNTAX:

```
software version
```

EXAMPLE:

```
=>Software version  
Active   : Sascha3.722           Passive  : Sascha3.716  
=>
```

RELATED COMMANDS:

**software switch**

Switch active and passive software versions and reboot the **SpeedTouch™ 510**.

## 16 System Commands

**system (to access the System level)**

**system clearpassword**

**system config**

**system flush**

**system reboot**

**system reset**

**system setpassword**

## ***system clearpassword***

Clear current **SpeedTouch™ 510** system password.

To avoid unrestricted and unauthorized access to the **SpeedTouch™ 510** it is highly recommended always to make sure that it is protected by a **SpeedTouch™ 510** 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 config

Configure the Universal Plug and Play and Alcatel proprietary MDAP protocol.

For a good operation of UPnP and the discovery mechanism, it is highly recommended not to change the System config settings.

SYNTAX:

<b>system config</b>	<b>[upnp = &lt;{off on}&gt;]</b> <b>[mdap = &lt;{off on}&gt;]</b>
----------------------	--

<i>[upnp]</i>	Enable (on) or disable (off) Universal Plug and Play (UPnP) discovery. By default UPnP discovery is enabled.	OPTIONAL
---------------	---	----------

<i>[mdap]</i>	Enable (on) or disable (off) Alcatel proprietary MDAP discovery. By default MDAP discovery is enabled.	OPTIONAL
---------------	---	----------

EXAMPLE:

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

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

EXAMPLE:

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

## ***system flush***

Flush current **SpeedTouch™ 510** system configuration, i.e. the System password.

System config settings are not affected by the System flush command.

The flush command does not impact previously saved configurations.

To avoid unrestricted and unauthorized access to the **SpeedTouch™ 510** it is highly recommended always to make sure that it is protected by a **SpeedTouch™ 510** system password (by executing **system setpassword**) and to change the password regularly.

SYNTAX:

```
system flush
```

EXAMPLE:

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

## ***system reboot***

Reboot the **SpeedTouch™ 510**. Non-saved configuration settings are lost after reboot.

SYNTAX:

```
system reboot
```

EXAMPLE:

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

## system reset

Reset the **SpeedTouch™ 510** to its factory default settings and reboot the device. All user and Service Provider specific settings and all saved configuration changes are lost after reboot.

### SYNTAX:

<b>system reboot</b>	<b>no/yes = &lt;{no yes}&gt;</b>	
no/yes	Proceed with resetting the <b>SpeedTouch™ 510</b> device to its factory default settings (yes) or not (no). By default the system reboot command is discarded in case of no explicit positive confirmation.	REQUIRED

### EXAMPLE:

```

=>system reboot
-----
!! WARNING !!
-----
The SpeedTouch(TM)510 ADSL Router will be reset to factory defaults clearing all user
and ISP specific settings.
Connectivity with the ISP network might be lost.
Do you want to proceed ?
no/yes = no
=>
=>system reboot
-----
!! WARNING !!
-----
The SpeedTouch(TM)510 ADSL Router will be reset to factory defaults clearing all user
and ISP specific settings.
Connectivity with the ISP network might be lost.
Do you want to proceed ?
no/yes = yes

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

```

## system setpassword

Set/change the current **SpeedTouch™ 510** system password.

To avoid unrestricted and unauthorized access to the **SpeedTouch™ 510** it is highly recommended always to make sure that it is protected by a **SpeedTouch™ 510** system password and to change it regularly.

SYNTAX:

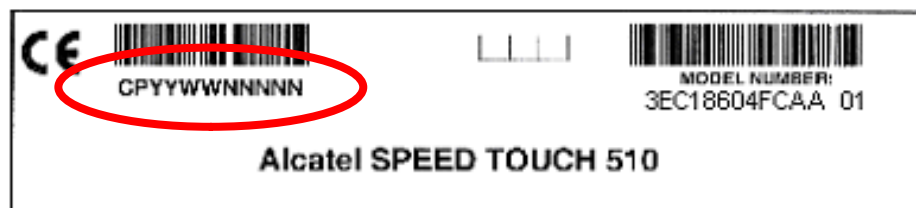
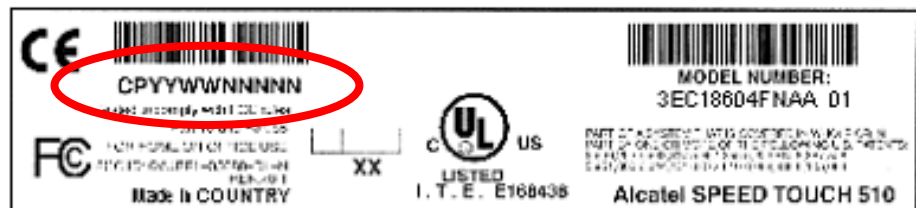
```
system setpassword password = {<string> | $_BOARD_SERIAL_NBR}
```

<i>password</i>	the system password can be set to either:	REQUIRED
	<ul style="list-style-type: none"> <li>▪ &lt;string&gt; A free to choose password &lt;string&gt;</li> <li>▪ \$_BOARD_SERIAL_NBR Equal to the <b>SpeedTouch™ 510</b> device serial number's nine numerical digits.</li> </ul>	

IMPORTANT NOTE:

### Serial number

The code serial number is printed on the marking label found on the bottom of the **SpeedTouch™ 510**:



It consists of the concatenation of the string 'CP' followed by nine digits. These nine digits incorporate the serial number.

In case the System password is set to the serial number, for authentication the serial number must be given without the preceding string 'CP'.

**DO NOT REMOVE OR COVER THIS MARKING LABEL !!!**



## 17 TD Commands

**td (to access this level)**  
**td call**  
**td prompt**

***td call***

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

## SYNTAX:

<b><i>td call</i></b>	<b><i>cmd = &lt;string&gt;</i></b> <b><i>[pwd = &lt;string&gt;]</i></b>
-----------------------	--

<i>cmd</i>	The quoted trace & debug command string.	REQUIRED
------------	--	----------

<i>pwd</i>	The 'Trace & Debug' prompt password, if required.	OPTIONAL
------------	---	----------

## ***td prompt***

Switch to Alcatel-owned 'Trace & Debug' prompt (expert mode). For qualified personnel only.

SYNTAX:

<b><i>td prompt</i></b>
-------------------------

Before entering the expert mode a DISCLAIMER is shown stipulating that the 'Trace & Debug' prompt (expert mode) is intended for qualified personnel only.

Pressing ENTER allows to return to user mode.

The 'Trace & Debug' prompt (expert mode) password is intended to be used by qualified personnel only.

The 'Trace & Debug' prompt (expert mode) password is not intended to protect the **SpeedTouch™ 510** from unrestricted and unauthorized access.

Therefore, to avoid unrestricted and unauthorized access to the **SpeedTouch™ 510** it is highly recommended always to make sure that it is protected by a **SpeedTouch™ 510** system password and to change it regularly. See the **system setpassword** command for more information.



---

# Alcatel SpeedTouch™ 510 Series

## CLI Command Index

---



# Index

## A

add  
 dhcp server lease add, 79  
 dhcp server pool add, 83  
 dns add, 90  
 phonebook add, 182

apadd, ip apadd, 140

apdelete, ip apdelete, 142

aplist, ip aplist, 143

applist, nat applist, 168

arpadd, ip arpadd, 144

arpdelete, ip arpdelete, 145

arplist, ip arplist, 146

assign, firewall assign, 118

autolist, phonebook autolist, 184

## B

bind, nat bind, 169

bindlist, nat bindlist, 170

## C

call, td call, 228

cleanup, software cleanup, 214

clear  
 dhcp client clear, 57  
 dhcp server clear, 71  
 dns clear, 91  
 firewall rule clear, 129

clearpassword, system clearpassword, 220

clrstats, dns clrstats, 92

config  
 adsl config, 24  
 bridge config, 28  
 dhcp client config, 58  
 dhcp server config, 72  
 dhcp server pool config, 84  
 ip config, 147  
 system config, 221

create  
 firewall chain create, 126  
 firewall rule create, 130  
 nat create, 171

## D

def, env def, 110

defserver, nat defserver, 173

delete  
 dhcp server lease delete, 80  
 dhcp server pool delete, 86  
 dns delete, 93  
 firewall chain delete, 127  
 firewall rule delete, 134  
 nat delete, 174  
 phonebook delete, 185

deletepassive, software deletepassive, 215

disable, nat disable, 175

domain, dns domain, 94

## E

enable, nat enable, 176

erase, config erase, 50

**F**

## flush

- bridge flush, 29
- cip flush, 42
- config flush, 51
- dhcp client flush, 59
- dhcp server lease flush, 81
- dhcp server pool flush, 87
- dns flush, 95
- env flush, 111
- firewall flush, 119
- firewall rule flush, 135
- ip flush, 150
- mer flush, 160
- nat flush, 177
- phonebook flush, 186
- ppp flush, 190
- pptp flush, 208
- system flush, 222

fwddadd, dns fwddadd, 96

fwdddelete, dns fwdddelete, 97

fwddlist, dns fwddlist, 98

fwddtable, dns fwddtable, 99

**G**

get, env get, 112

**H**

help, 11

**I**

## ifadd

- bridge ifadd, 30
- cip ifadd, 43
- dhcp client ifadd, 60
- mer ifadd, 161
- ppp ifadd, 191

## ifattach

- bridge ifattach, 31
- dhcp client ifattach, 61
- mer ifattach, 162
- ppp ifattach, 193

## ifconfig

- bridge ifconfig, 32
- dhcp client ifconfig, 62
- ip ifconfig, 151
- mer ifconfig, 163
- ppp ifconfig, 194

## ifdelete

- bridge ifdelete, 34
- cip ifdelete, 44
- dhcp client ifdelete, 64
- mer ifdelete, 164
- ppp ifdelete, 199

## ifdetach

- bridge ifdetach, 35
- mer ifdetach, 165
- ppp ifdetach, 200

## iflist

- bridge iflist, 36
- cip iflist, 45
- dhcp client iflist, 65
- ip iflist, 152
- mer iflist, 166
- ppp iflist, 201

ifrelease, dhcp client ifrelease, 66

ifrenew, dhcp client ifrenew, 68

ifscan, ppp ifscan, 202

info, adsl info, 25

**L**

## list

- dhcp server lease list, 82
- dhcp server pool list, 88
- dns list, 100
- env list, 113
- firewall chain list, 128
- firewall list, 120
- firewall rule list, 136
- nat list, 178
- phonebook list, 187
- pptp list, 209

load, config load, 52

**M**

macadd, bridge macadd, 37

macdelete, bridge macdelete, 38

maclist, bridge maclist, 39  
match, firewall match, 121

## N

nslookup, dns nslookup, 101

## P

ping, ip ping, 153  
policy, dhcp server policy, 73  
profadd, pptp profadd, 210  
profdelete, pptp profdelete, 211  
proflist, pptp proflist, 212  
prompt, td prompt, 229  
pvcadd, cip pvcadd, 46  
pvcdelete, cip pvcdelete, 47  
pvclist, cip pvclist, 48

## R

reboot, system reboot, 223  
reset, system reset, 224  
rtadd  
    ip rtadd, 154  
    ppp rtadd, 203  
rtdelete  
    ip rtdelete, 155  
    ppp rtdelete, 205  
rtlist, ip rtlist, 156

## S

save, config save, 53

sendto, ip sendto, 157  
set, env set, 114  
setpassive, software setpassive, 216  
setpassword, system setpassword, 225  
spooof, dhcp server spooof, 74  
start  
    dhcp server start, 75  
    dns start, 102  
stats  
    dhcp client stats, 70  
    dhcp server stats, 76  
    dns stats, 103  
    firewall rule stats, 137  
status, dns status, 104  
stop  
    dhcp server stop, 78  
    dns stop, 105  
switch, software switch, 217

## T

toutfwd, dns toutfwd, 106  
troff, dns troff, 107  
tron, dns tron, 108

## U

unbind, nat unbind, 179  
unset, env unset, 115

## V

version, software version, 218

