

SPEED TOUCH

CLI Reference Guide

DSL Routers



Status Released

Change Note BD F aa 34022

Short Title CD-UG STCLI

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

Contents

Preface	5
1 ADSL Commands	19
2 ATMF Commands	21
3 Bridge Commands	29
4 CIP Commands	45
5 Config Commands	55
6 DHCP Commands	61
7 DNS Commands	99
8 Firewall Commands	121
9 IP Commands	147
10 MER Commands	169
11 NAT Commands	181
12 Phonebook Commands	197
13 PPP Commands	207
14 PPTP Commands	225
15 QoSBook Commands	235
16 SHDSL Commands	243
17 Software Commands	249
18 System Commands	255
19 TD Commands	263
20 Voice Commands	267
21 Wireless Commands	273
Index	297

Preface

Welcome to the **Alcatel Speed Touch™ Command Line Interface Reference Guide** !

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

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

This CLI Reference Guide covers the CLI commands of all Alcatel's DSL Speed Touch products. Therefore it is possible that far more commands are described than available on your **Speed Touch™** product. Please ignore all CLI commands which are not applicable for your product.

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 **Speed Touch™** 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 all **Speed Touch™** products are alphabetically described per group selection.

Each command is described in a systematic manner:

- The full name of the CLI command (including the group selection)
- A short description of the CLI command, if needed completed by a description of the possible impact on the user and/or the **Speed Touch™**
- The syntax of the command with a description of each parameter
- An example to demonstrate the use of the CLI command
- A list of related CLI commands.

► **Part 3 : CLI Command Index**

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

► Trademarks

The following trademarks are used in this document:

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

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

► Updates

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

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

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

► Directive

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

- disassemble, de-compile, reverse engineer, trace or otherwise analyse 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 Speed Touch™

CLI Navigation

Accessing the Command Line Interface

Users can access the Command Line Interface via:

- ▶ A Telnet session
This requires that TCP/IP connectivity exists between the host from which the Telnet session is opened and the **Speed Touch**™
- ▶ The serial 'Console' interface
This requires a **Speed Touch**™ product which is equipped with a 'Console' port on its back panel.

As soon a session to the CLI is opened, the **Speed Touch™** banner pops up, followed by the CLI prompt.

In case of a Telnet session authentication via the **Speed Touch™** System Password might be required before access is granted.

The following figure shows an example of the **Speed Touch™** banner after opening a Telnet session and authentication.

EXAMPLE:

```
/home/doejohn{1}$ telnet 10.0.0.138
Trying 10.0.0.138...
Connected to 10.0.0.138.
Escape character is '^]'.
User :
SpeedTouch (00-90-D0-00-01-23-45)
Password : #####
```

```
*
*
*
*      ALCATEL DSL ROUTER
*
*
*      Copyright 1999-2001.
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
```

=>

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 group selection.
..            : Exits group selection.

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

=>atmf
[atmf]=>
```

To return to top level, or to descend one level (in case of nested command groups) execute **..** or **exit** (depending on the **Speed Touch™** product).

EXAMPLE:

```
=>atmf
[atmf]=>
[atmf]=>..  
=>
```

The Help Command

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

EXAMPLE:

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

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

=>
```

The available command groups depend on the particular **Speed Touch™** the CLI session is opened to, e.g. the **wireless** command group is only present for the **SpeedTouch™570** product.

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:

```
=>atmf
[atmf]=>
[atmf]=>help
Following commands are available :
add : Adds a VP or VC cross-connection between the ATMF interface and the WAN
interface.
delete : Deletes a cross-connection on the ATM-Forum interface.
list : Shows all ATM-Forum interface cross-connections.
save : Saves the ATM-Forum interface cross-connections.
flush : Flushes the ATM-Forum interface cross-connections.
load : Loads ATM-Forum interface cross-connections.

[atmf]=>
```

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

EXAMPLE:

```
=>help atmf
Following commands are available :
add : Adds a VP or VC cross-connection between the ATMF interface and the WAN
interface.
delete : Deletes a cross-connection on the ATM-Forum interface.
list : Shows all ATM-Forum interface cross-connections.
save : Saves the ATM-Forum interface cross-connections.
flush : Flushes the ATM-Forum interface cross-connections.
load : Loads ATM-Forum interface cross-connections.

=>
```

Entering **help** followed by a specific command, e.g. **help atmf add** (starting from top level) or **help add** (e.g. on the atmf command group selection) results in a description of the syntax for the command.

EXAMPLE:

```
=>help atmf add
add : Adds a VP or VC cross-connection between the ATMF interface and the WAN
interface.

    vpi = <number{0-15}>
        The Virtual Path Identifier.
    vci = <number{0-511}>
        The Virtual Channel Identifier. Use 0 for a VP crossconnect.

=>
```

Command Completion

For some **Speed Touch™** products the CLI features command completion, which means that when starting to enter a command it can be completed by pressing the **"Tab"** key.

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

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

EXAMPLE:

```
=>atmf
[atmf]=>a "Tab"
[atmf]=>add
```

Going to the beginning or end of the Command Line

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

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

EXAMPLE:

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

Breaking off Commands

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

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

EXAMPLE:

```
[atmf]=>add  
vpi =  
vpi =  
vpi = "Ctrl+G" [atmf]=>
```

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:

```
=>atmf  
[atmf]=>list  
VPI = 0 VCI = 0  
VPI = 1 VCI = 0  
VPI = 2 VCI = 0  
VPI = 3 VCI = 0  
VPI = 4 VCI = 0  
VPI = 5 VCI = 0  
VPI = 6 VCI = 0  
VPI = 7 VCI = 0  
[atmf]=> "↑"  
=>:list
```

Command Line Interface Top Level Structure

Depending on the **Speed Touch™** product the following command groups are available:

- ▶ **adsl** (only applicable for **SpeedTouch™** ADSL variants)
- ▶ **atmf** (only applicable for **SpeedTouch™** models equipped with an ATMF-25.6Mbps port)
- ▶ **bridge**
- ▶ **cip** (only applicable for **SpeedTouch™** DSL routers)
- ▶ **config**
- ▶ **dhcp**
- ▶ **dns**
- ▶ **env** (only applicable for **SpeedTouch™ 510Series**)
- ▶ **firewall** (only applicable for **SpeedTouch™ Pro with Firewall** and **SpeedTouch™ 520Series**)
- ▶ **ip**
- ▶ **mer** (only applicable for **SpeedTouch™ Pro with Firewall** and **SpeedTouch™ 520Series**)
- ▶ **nat** (only applicable for **Speed Touch™** DSL routers)
- ▶ **phonebook**
- ▶ **ppp** (only applicable for **Speed Touch™** DSL routers)
- ▶ **pptp**
- ▶ **qosbook** (only applicable for **SpeedTouch™ Pro with Firewall** and **SpeedTouch™ 520Series**)
- ▶ **shdsl** (only applicable for **SpeedTouch™** SHDSL variants)
- ▶ **software**
- ▶ **system**
- ▶ **td**
- ▶ **voice** (only applicable for the **SpeedTouch™ 700/710/720**)
- ▶ **wireless** (only applicable for the **SpeedTouch™ 570**)

Command Line Interface Commands

All CLI commands are commands that operate on, or configure, the **Speed Touch™** 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. **atmf list**).

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

When executing commands with specification of their parameter values, it is not necessary to use '='. E.g. '**atmf add vpi 0 vci 35**' is equal to '**atmf add vpi=0 vci=35**'.

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

EXAMPLE:

```
=>atmf list
VPI = 0 VCI = 0
VPI = 1 VCI = 0
VPI = 2 VCI = 0
VPI = 3 VCI = 0
VPI = 4 VCI = 0
VPI = 5 VCI = 0
VPI = 6 VCI = 0
VPI = 7 VCI = 0
=>atmf
[atmf]=>list
VPI = 0 VCI = 0
VPI = 1 VCI = 0
VPI = 2 VCI = 0
VPI = 3 VCI = 0
VPI = 4 VCI = 0
VPI = 5 VCI = 0
VPI = 6 VCI = 0
VPI = 7 VCI = 0
[atmf]=>
```

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 '**atmf add vpi 0 vci 35**'. To break of such incomplete command press "**Ctrl+G**".

EXAMPLE:

```
=>atmf add
vpi = 0
vci = 35
=>
```

Alcatel Speed Touch™

CLI Command Description

1 ADSL Commands

adsl (to access the ADSL level)
adsl info

adsl info

Show ADSL statistics and information about the **SpeedTouch™** 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
Vendor               :          0f          0f
VendorSpecific       :          ALCB       ALCB
StandardRevisionNr  :          0000       0000
                     :          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
Current Connection
Downstream            :          185668    71296
Upstream              :          N/Avail    N/Avail
Errors
Received FEC          :          0
Received CRC          :          0
Received HEC          :          0

=>
```

2 ATMF Commands

atmf (to access the ATMF level)

atmf add

atmf delete

atmf flush

atmf list

atmf load

atmf save

atmf add

Add a cross-connection between the ATMF-25 interface and the WAN interface.

SYNTAX:

atmf add	vpi = <number {0-7}> vci = <number {0-511}>
-----------------	--

vpi	A number between 0 and 7. Represents the Virtual Path identifier.	REQUIRED
vci	A number between 0 and 511. Represents the Virtual Channel identifier. Use VCI=0 for a VP cross-connection.	REQUIRED

EXAMPLE:

```
=>atmf list
VPI = 0 VCI = 0
VPI = 1 VCI = 0
VPI = 2 VCI = 0
VPI = 3 VCI = 0
VPI = 4 VCI = 0
VPI = 5 VCI = 0
VPI = 6 VCI = 0
VPI = 7 VCI = 0
=>atmf delete vpi 0 vci 0
=>atmf add vpi 0 vci 35
=>atmf list
VPI = 0 VCI = 35
VPI = 1 VCI = 0
VPI = 2 VCI = 0
VPI = 3 VCI = 0
VPI = 4 VCI = 0
VPI = 5 VCI = 0
VPI = 6 VCI = 0
VPI = 7 VCI = 0
=>
```

RELATED COMMANDS:

atmf delete	Delete a cross-connection on the ATMF-25 interface.
atmf list	Show current ATMF-25 interface configuration.

atmf delete

Delete a cross-connection on the ATMF-25 interface.

SYNTAX:

atmf delete	vpi = <number {0-7}> vci = <number {0-511}>	
vpi	A number between 0 and 7. Represents the Virtual Path identifier.	REQUIRED
vci	A number between 0 and 511. Represents the Virtual Channel identifier. Use VCI=0 for a VP cross-connection.	REQUIRED

EXAMPLE:

```
=>atmf list
VPI = 0 VCI = 35
VPI = 1 VCI = 0
VPI = 2 VCI = 0
VPI = 3 VCI = 0
VPI = 4 VCI = 0
VPI = 5 VCI = 0
VPI = 6 VCI = 0
VPI = 7 VCI = 0
=>atmf delete vpi 0 vci 35
=>atmf list
VPI = 1 VCI = 0
VPI = 2 VCI = 0
VPI = 3 VCI = 0
VPI = 4 VCI = 0
VPI = 5 VCI = 0
VPI = 6 VCI = 0
VPI = 7 VCI = 0
=>
```

RELATED COMMANDS:

atmf add	Add a cross-connection on the ATMF-25 interface.
atmf list	Show current ATMF-25 interface configuration.

atmf flush

Flush complete ATMF-25 interface configuration.

The flush command does not impact previously saved configurations.

SYNTAX:

<i>atmf flush</i>

EXAMPLE:

```
=>atmf list
VPI = 0 VCI = 35
VPI = 1 VCI = 0
VPI = 2 VCI = 0
VPI = 3 VCI = 0
VPI = 4 VCI = 0
VPI = 5 VCI = 0
VPI = 6 VCI = 0
VPI = 7 VCI = 0
=>atmf flush
=>atmf list
=>atmf load
=>atmf list
VPI = 0 VCI = 35
VPI = 1 VCI = 0
VPI = 2 VCI = 0
VPI = 3 VCI = 0
VPI = 4 VCI = 0
VPI = 5 VCI = 0
VPI = 6 VCI = 0
VPI = 7 VCI = 0
=>
```

RELATED COMMANDS:

atmf load

Load saved or default ATMF-25 interface configuration.

atmf save

Save current ATMF-25 interface configuration.

atmf list

Show all current ATMF-25 interface cross-connections .

SYNTAX:

```
atmf list
```

EXAMPLE OUTPUT:

```
=>atmf list
VPI = 0 VCI = 35
VPI = 1 VCI = 0
VPI = 2 VCI = 0
VPI = 3 VCI = 0
VPI = 4 VCI = 0
VPI = 5 VCI = 0
VPI = 6 VCI = 0
VPI = 7 VCI = 0
=>
```

RELATED COMMANDS:

atmf add

Add an ATMF-25 interface cross-connection.

atmf delete

Delete a cross-connection on the ATMF-25 interface.

atmf load

Load saved (or default) ATMF-25 interface configuration.

Execute **atmf flush** prior to **atmf load**.

SYNTAX:

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

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

EXAMPLE:

```
=>atmf list
VPI = 0 VCI = 35
=>atmf save
=>atmf flush
=>atmf load defaults yes
=>atmf list
VPI = 0 VCI = 0
VPI = 1 VCI = 0
VPI = 2 VCI = 0
VPI = 3 VCI = 0
VPI = 4 VCI = 0
VPI = 5 VCI = 0
VPI = 6 VCI = 0
VPI = 7 VCI = 0
=>atmf flush
=>atmf load defaults no
=>atmf list
VPI = 0 VCI = 35
=>
```

RELATED COMMANDS:

atmf flush	Flush complete ATMF-25 interface configuration.
atmf save	Save current ATMF-25 interface configuration.

atmf save

Save current ATMF-25 interface configuration.

SYNTAX:

atmf save

EXAMPLE:

```
>atmf list
VPI = 0 VCI = 0
VPI = 1 VCI = 0
VPI = 2 VCI = 0
VPI = 3 VCI = 0
=>atmf delete vpi 0 vci 0
=>atmf add vpi 0 vci 35
=>atmf list
VPI = 0 VCI = 35
VPI = 1 VCI = 0
VPI = 2 VCI = 0
VPI = 3 VCI = 0
=>atmf save
=>atmf flush
=>atmf list
=>atmf load
=>atmf list
VPI = 0 VCI = 35
VPI = 1 VCI = 0
VPI = 2 VCI = 0
VPI = 3 VCI = 0
=>
```

RELATED COMMANDS:

atmf flush

Flush complete ATMF-25 interface configuration.

atmf load

Load saved or default ATMF-25 interface configuration.

3 Bridge Commands

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

bridge config

Show/set bridge aging policy.

SYNTAX:

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

[age]

A number between 10 and 100000.

OPTIONAL

Represents the lifetime in seconds of a dynamically learned MAC address.

By default the aging timer is 300 seconds.

EXAMPLE:

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

bridge flush

Flush complete bridging configuration.

The flush command does not impact previously saved configurations.

SYNTAX:

bridge flush

EXAMPLE:

```
=>bridge iflist
OBC      : Internal
           Connection State : connected   Port : OBC   PortState : forwarding
           RX bytes: 75783      frames: 572
           TX bytes: 82768372   frames: 341221      dropframes: 0
eth0      : Internal
           Connection State : connected   Port : eth0   PortState : forwarding
           RX bytes: 156344216  frames: 5899238
           TX bytes: 75689      frames: 425        dropframes: 5558017
=>bridge ifadd intf NewBridge dest Br1
=>bridge iflist
OBC      : Internal
           Connection State : connected   Port : OBC   PortState : forwarding
           RX bytes: 75783      frames: 572
           TX bytes: 82843610   frames: 341554      dropframes: 0
eth0      : Internal
           Connection State : connected   Port : eth0   PortState : forwarding
           RX bytes: 156472129  frames: 5903256
           TX bytes: 75689      frames: 425        dropframes: 5561702
NewBridge : dest : Br1
           Retry : 10   QoS : default   Encaps : llc/snap   Fcs : off
           Connection State: not-connected Port : (Unassigned) PortState : forwarding
=>bridge flush
=>bridge load
=>bridge iflist
OBC      : Internal
           Connection State : connected   Port : OBC   PortState : forwarding
           RX bytes: 75783      frames: 572
           TX bytes: 82908667   frames: 341735      dropframes: 0
eth0      : Internal
           Connection State : connected   Port : eth0   PortState : forwarding
           RX bytes: 156553257  frames: 5904070
           TX bytes: 75689      frames: 425        dropframes: 5562335
=>
```

RELATED COMMANDS:

bridge load

Load saved or default bridge configuration.

bridge save

Save current bridge configuration.

bridge ifadd

Create a bridge interface.

SYNTAX:

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

EXAMPLE:

```
=>bridge iflist
OBC      : Internal
           Connection State : connected   Port : OBC   PortState : forwarding
           RX bytes: 75783      frames: 572
           TX bytes: 83293945   frames: 343261      dropframes: 0
eth0      : Internal
           Connection State : connected   Port : eth0   PortState : forwarding
           RX bytes: 157212625  frames: 5920249
           TX bytes: 75689      frames: 425        dropframes: 5576988
=>phonebook list
Name      Type   Use   Address
Br1       bridge 0    8.35
=>bridge ifadd intf NewBridge dest Br1
=>bridge iflist
OBC      : Internal
           Connection State : connected   Port : OBC   PortState : forwarding
           RX bytes: 76713      frames: 578
           TX bytes: 84056140   frames: 346238      dropframes: 0
eth0      : Internal
           Connection State : connected   Port : eth0   PortState : forwarding
           RX bytes: 158501623  frames: 5952020
           TX bytes: 76643      frames: 431        dropframes: 5605782
NewBridge : dest : Br1
           Retry : 10   QoS : default   Encaps : llc/snap   Fcs : off
           Connection State : not-connected Port : (Unassigned)PortState : forwarding
=>
```

RELATED COMMANDS:

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

bridge ifattach

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

SYNTAX:

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

EXAMPLE:

```
=>bridge iflist
OBC      : Internal
           Connection State : connected   Port : OBC   PortState : forwarding
           RX bytes: 76713      frames: 578
           TX bytes: 84571673   frames: 348117   dropframes: 0
eth0      : Internal
           Connection State : connected   Port : eth0   PortState : forwarding
           RX bytes: 159362213  frames: 5969925
           TX bytes: 76643      frames: 431     dropframes: 5621648
NewBridge : dest : Br1
           Retry : 10   QoS : default   Encaps : llc/snap   Fcs : off
           Connection State : not-connected Port : (Unassigned) PortState: forwarding
=>bridge ifattach intf NewBridge
=>bridge iflist
OBC      : Internal
           Connection State : connected   Port : OBC   PortState : forwarding
           RX bytes: 76713      frames: 578
           TX bytes: 84643057   frames: 348365   dropframes: 0
eth0      : Internal
           Connection State : connected   Port : eth0   PortState : forwarding
           RX bytes: 159468781  frames: 5973337
           TX bytes: 76643      frames: 431     dropframes: 5624791
NewBridge : dest : Br1
           Retry : 10   QoS : default   Encaps : llc/snap   Fcs : off
           Connection State : connected   Port : wan0   PortState : forwarding
           RX bytes: 0          frames: 0
           TX bytes: 30246      frames: 91     dropframes: 0
=>
```

RELATED COMMANDS:

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

bridge ifconfig

Configure a bridge interface.

SYNTAX:

bridge ifconfig	intf = <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.	OPTIONAL
<i>[encaps]</i>	The type of encapsulation to be used for this bridge interface. Choose between: <ul style="list-style-type: none"> ▪ llc/snap ▪ vcmux 	OPTIONAL
<i>[fcs]</i>	Whether or not to include the Ethernet FCS in the packet header on the WAN side. Choose between: <ul style="list-style-type: none"> ▪ off ▪ on 	OPTIONAL
<i>[portstate]</i>	The bridge portstate for this interface. Choose between: <ul style="list-style-type: none"> ▪ disabled ▪ learning ▪ forwarding 	OPTIONAL
<i>[retry]</i>	A number between 0 and 65535. Represents the number of WAN connection setup retries before giving up. By default the retry value is 10.	OPTIONAL

EXAMPLE:

```
=>bridge iflist intf NewBridge NewBridge : dest : Br1
      Retry : 10 QoS : default Encaps : llc/snap Fcs : off
      Connection State : connected   Port : wan0   PortState : forwarding
      RX bytes: 0             frames: 0
      TX bytes: 5103034       frames: 22139         dropframes: 0
=>qosbook list
Name      Ref Type      TX peek  sust      burst    RX peek  sust      burst
          (Kbits) (Kbits) (bytes) (Kbits) (Kbits) (bytes)
default 3   ubr      Linerate 0          0          Linerate 0          0
PPP3    0   ubr      6144     0          0          Linerate 0          0
voice   0   cbr      64       0          0          64       0          0
=>bridge ifconfig intf NewBridge qos PPP3 encaps vcmux retry 15
=>bridge iflist intf NewBridge
NewBridge : dest : Br1
      Retry : 15 QoS : PPP3 Encaps : vcmux Fcs : off
      Connection State : connected   Port : wan0   PortState : forwarding
      RX bytes: 0             frames: 0
      TX bytes: 6094426       frames: 26373         dropframes: 0
=>
```

RELATED COMMANDS:

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

bridge ifdelete

Delete a bridge interface.

bridge ifdelete	intf = <ifname>
------------------------	------------------------------

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

EXAMPLE:

```
=>bridge iflist
OBC      : Internal
          Connection State : connected   Port : OBC   PortState : forwarding
          RX bytes: 79742      frames: 595
          TX bytes: 88218185   frames: 362634      dropframes: 0
eth0     : Internal
          Connection State : connected   Port : eth0   PortState : forwarding
          RX bytes: 165656174  frames: 6517744
          TX bytes: 79740      frames: 448          dropframes: 6141689
NewBridge : dest : Br1
          Retry : 15   QoS : PPP3   Encaps : vcmux   Fcs : off
          Connection State : connected   Port : wan0   PortState : forwarding
          RX bytes: 0        frames: 0
          TX bytes: 6379953   frames: 27586      dropframes: 0
=>bridge ifdelete intf NewBridge
=>bridge iflist
OBC      : Internal
          Connection State : connected   Port : OBC   PortState : forwarding
          RX bytes: 79742      frames: 595
          TX bytes: 88296241   frames: 362952      dropframes: 0
eth0     : Internal
          Connection State : connected   Port : eth0   PortState : forwarding
          RX bytes: 165797004  frames: 6520010
          TX bytes: 79740      frames: 448          dropframes: 6143571
=>
```

RELATED COMMANDS:

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

bridge ifdetach

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

SYNTAX:

bridge ifdetach	intf = <ifname>
intf	The name of the bridge interface to detach.
	REQUIRED

EXAMPLE:

```
=>bridge iflist intf NewBridge
NewBridge : dest : Br1
           Retry : 10  QoS : default  Encaps : llc/snap  Fcs : off
           Connection State : connected  Port : wan0  PortState : forwarding
           RX bytes: 0      frames: 0
           TX bytes: 16363  frames: 98      dropframes: 0
=>bridge ifdetach intf NewBridge
=>bridge iflist intf NewBridge
NewBridge : dest : Br1
           Retry : 10  QoS : default  Encaps : llc/snap  Fcs : off
           Connection State : not-connected Port : (Unassigned) PortState :forwarding
=>
```

RELATED COMMANDS:

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

bridge iflist

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

SYNTAX:

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

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

EXAMPLE OUTPUT:

```
=>bridge iflist
OBC      : Internal
          Connection State : connected   Port : OBC   PortState : forwarding
          RX bytes: 0       frames: 0
          TX bytes: 1964    frames: 31      dropframes: 0
eth0     : Internal
          Connection State : connected   Port : eth0  PortState : forwarding
          RX bytes: 23660   frames: 788
          TX bytes: 0       frames: 0      dropframes: 757
=>bridge ifadd intf NewBridge dest Br1
=>bridge ifattach intf NewBridge
=>bridge iflist intf NewBridge
NewBridge : dest : Br1
          Retry : 10  QoS : default  Encaps : llc/snap  Fcs : off
          Connection State : connected   Port : wan0  PortState : forwarding
          RX bytes: 0       frames: 0
          TX bytes: 16363   frames: 98      dropframes: 0
=>
```

DESCRIPTION:

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

RELATED COMMANDS:

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

bridge load

Load saved (or default) bridge configuration.

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

SYNTAX:

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

EXAMPLE:

```
=>bridge iflist
OBC      : Internal
           Connection State : connected   Port : OBC   PortState : forwarding
           RX bytes: 2706      frames: 16
           TX bytes: 1798797   frames: 6807      dropframes: 0
eth0     : Internal
           Connection State : connected   Port : eth0  PortState : forwarding
           RX bytes: 3205514   frames: 151375
           TX bytes: 2770      frames: 16      dropframes: 144380
=>bridge ifadd intf NewBridge dest Br1
=>bridge save
=>bridge flush
=>bridge iflist
OBC      : Internal
           Connection State : connected   Port : OBC   PortState : forwarding
           RX bytes: 3029      frames: 17
           TX bytes: 2171576   frames: 8299      dropframes: 0
eth0     : Internal
           Connection State : connected   Port : eth0  PortState : forwarding
           RX bytes: 3868737   frames: 170628
           TX bytes: 3097      frames: 17      dropframes: 162141
=>bridge load defaults no
=>bridge iflist
OBC      : Internal
           Connection State : connected   Port : OBC   PortState : forwarding
           RX bytes: 3029      frames: 17
           TX bytes: 2261537   frames: 8663      dropframes: 0
eth0     : Internal
           Connection State : connected   Port : eth0  PortState : forwarding
           RX bytes: 4037236   frames: 184096
           TX bytes: 3097      frames: 17      dropframes: 175245
NewBridge : dest : Br1
           Retry : 10  QoS : default  Encaps : llc/snap  Fcs : off
           Connection State : not-connected Port : (Unassigned) PortState : forwarding
=>
```

RELATED COMMANDS:

bridge flush

Flush complete bridge configuration.

bridge save

Save current bridge configuration.

bridge macadd

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

SYNTAX:

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

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

EXAMPLE:

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

RELATED COMMANDS:

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

bridge macdelete

Remove a MAC address from the filtering database.

SYNTAX:

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

EXAMPLE:

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

RELATED COMMANDS:

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

bridge maclist

Show current MAC address filtering database.

SYNTAX:

bridge maclist

EXAMPLE:

<pre>=>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, 597 seconds 00:50:8b:31:cc:aa -- dynamic, eth0, 513 seconds 08:00:20:c1:9a:12 -- dynamic, eth0, 600 seconds 00:08:c7:c3:5f:fc -- dynamic, eth0, 349 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</pre>
--

RELATED COMMANDS:

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

bridge save

Save current bridge configuration.

SYNTAX:

```
bridge save
```

EXAMPLE:

```
=>bridge iflist
OBC      : Internal
           Connection State : connected   Port : OBC   PortState : forwarding
           RX bytes: 2706      frames: 16
           TX bytes: 1798797   frames: 6807      dropframes: 0
eth0     : Internal
           Connection State : connected   Port : eth0  PortState : forwarding
           RX bytes: 3205514   frames: 151375
           TX bytes: 2770      frames: 16      dropframes: 144380
=>bridge ifadd intf NewBridge dest Br1
=>bridge save
=>bridge flush
=>bridge iflist
OBC      : Internal
           Connection State : connected   Port : OBC   PortState : forwarding
           RX bytes: 3029      frames: 17
           TX bytes: 2171576   frames: 8299      dropframes: 0
eth0     : Internal
           Connection State : connected   Port : eth0  PortState : forwarding
           RX bytes: 3868737   frames: 170628
           TX bytes: 3097      frames: 17      dropframes: 162141
=>bridge load
=>bridge iflist
OBC      : Internal
           Connection State : connected   Port : OBC   PortState : forwarding
           RX bytes: 3029      frames: 17
           TX bytes: 2261537   frames: 8663      dropframes: 0
eth0     : Internal
           Connection State : connected   Port : eth0  PortState : forwarding
           RX bytes: 4037236   frames: 184096
           TX bytes: 3097      frames: 17      dropframes: 175245
NewBridge : dest : Br1
           Retry : 10  QoS : default  Encaps : llc/snap  Fcs : off
           Connection State : not-connected Port : (Unassigned) PortState : forwarding
=>
```

RELATED COMMANDS:

bridge flush

Flush complete bridge configuration.

bridge load

Load saved or default bridge configuration.

4 CIP Commands

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

cip flush

Flush complete CIP configuration.

The flush command does not impact previously saved configurations.

SYNTAX:

<i>cip flush</i>

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

RELATED COMMANDS:

cip load
cip save

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

cip ifadd

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

SYNTAX:

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

<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
=>
=>cip save
=>
```

RELATED COMMANDS:

cip ifdelete

Delete a CIP interface.

cip ifadd

Show current CIP configuration.

cip ifdelete

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

SYNTAX:

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

<i>addr</i>	The CIP interface's local IP address in the LIS.	REQUIRED
-------------	--	----------

EXAMPLE:

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

RELATED COMMANDS:

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

cip iflist

Show current CIP configuration.

SYNTAX:

```
cip iflist
```

EXAMPLE OUTPUT:

```

=>cip iflist
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
=>

```

DESCRIPTION:

inarp_reqs_in/inarp_reqs_out : Incoming/outgoing inverse ARP requests

inarp_repl_in/inarp_repl_out : Incoming/outgoing inverse ARP replies

inarp_inv_in/inarp_inv_out : Incoming/outgoing invalid inverse ARP messages

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

```

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

```

RELATED COMMANDS:

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

cip load

Load saved (or default) CIP configuration.

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

SYNTAX:

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

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

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

```

RELATED COMMANDS:

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

cip pvcadd

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

SYNTAX:

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

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

EXAMPLE:

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

RELATED COMMANDS:

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

cip pvcdelete

Delete a PVC ARP entry.

SYNTAX:

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

EXAMPLE:

```
=>phonebook list
Name      Type    Use  Address
Br1       bridge 1    8.35
PPP1      ppp     1    8.64
DHCP_SPOOF ppp     1    8.67
CIPPVC1   cip     0    8.80
CIPPVC2   cip     0    8.81
CIPPVC3   cip     0    8.82
=>cip pvclist
CIPPVC1   atmport = 0  vpi = 8  vci = 80  dest_ip = 172.16.1.2
          encaps = llc  mtu = 546
=>cip pvcdelete dest CIPPVC1
=>cip pvclist
=>
```

RELATED COMMANDS:

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

cip pvclist

Show current PVC ARP entries.

SYNTAX:

```
cip pvclist
```

EXAMPLE OUTPUT:

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

EXAMPLE INPUT/OUTPUT IN A NETWORKED ENVIRONMENT:

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

RELATED COMMANDS:

cip pvdelete

Delete a PVC ARP entry.

cip pvadd

Create a PVC ARP entry.

cip save

Save current CIP configuration.

SYNTAX:

```
cip save
```

EXAMPLE:

```

=>cip iflist
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
=>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
=>cip save
=>cip flush
=>cip load
=>cip iflist
cip1   addr = 172.16.0.5  mask = 255.255.255.0
      UNI address = A0:*.04
      inarp_reqs_in  =    0  inarp_repl_in  =    0  inarp_inv_in  =    0
      inarp_reqs_out =    0  inarp_repl_out =    0  inarp_inv_out =    0
=>

```

RELATED COMMANDS:

cip flush

Flush complete CIP configuration.

cip load

Load saved or default CIP configuration.

5 Config Commands

config (to access the Config level)

config erase

config flush

config load

config reset

config save

config erase

Physically remove all saved configurations.

SYNTAX:

config erase

EXAMPLE:

=>config erase

RELATED COMMANDS:

config flush	Flush complete runtime configuration.
config load	Load complete saved or default configuration.
config reset	Flush current and optionally restore default configuration.
config save	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 flush**, **dns flush**, **firewall flush**, **firewall rule flush**, **mer flush**, **nat flush**, **phonebook flush**, **ppp flush**, **pptp flush**, **qosbook flush**, **system flush** and optionally **ip flush**.

SYNTAX:

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

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

EXAMPLE:

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

RELATED COMMANDS:

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

config load

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

SYNTAX:

config load	[load_ip = <{no yes}>] [config_set = {saved default}]
--------------------	--

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

EXAMPLE:

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

RELATED COMMANDS:

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

config reset

Flush current runtime configuration and optionally restore factory default configuration. The reset command does not impact saved configurations.

SYNTAX:

config reset	[keep_ip = <{no yes}>]	
[load_ip]	Load IP settings (yes) or not (no). Not keeping the IP settings could cause lost IP connectivity in the LAN.	OPTIONAL

EXAMPLE:

```
=>dns list
Domain: lan
Nr.      Hostname      IP Address
  0      TestHost      10.0.0.140
  1      Default       10.0.0.8
Total Table Size: 73 entries
Amount used: 2 (2%)
=>config reset
=>dns list
Domain: lan
Nr.      Hostname      IP Address
  0      SpeedTouch    *.*.*.*
Total Table Size: 73 entries
Amount used: 1 (1%)
=>config flush
=>config load
=>dns list
Domain: lan
Nr.      Hostname      IP Address
  0      TestHost      10.0.0.140
  1      Default       10.0.0.8
Total Table Size: 73 entries
Amount used: 2 (2%)
=>
```

RELATED COMMANDS:

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

config save

Save all existing configurations and modifications entered by the user.

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

SYNTAX:

config save

EXAMPLE:

```
=>dns list
Domain: lan
Nr.      Hostname      IP Address
  0      Default      10.0.0.8
Total Table Size: 73 entries
Amount used: 1 (1%)
=>dns list
Domain: lan
Nr.      Hostname      IP Address
  0      Default      10.0.0.8
Total Table Size: 73 entries
Amount used: 1 (1%)
=>dns add hostname TestHost addr 172.16.1.1
=>dns list
Domain: lan
Nr.      Hostname      IP Address
  0      Default      10.0.0.8
  1      Testhost     172.16.1.1
Total Table Size: 73 entries
Amount used: 2 (2%)
=>config save
=>dns flush
=>dns load
=>dns list
Domain: lan
Nr.      Hostname      IP Address
  0      Default      10.0.0.8
  1      Testhost     172.16.1.1
Total Table Size: 73 entries
Amount used: 2 (2%)
=>
```

RELATED COMMANDS:

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

6 DHCP Commands

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

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

dhcp client clear

Clear DHCP client statistics.

SYNTAX:

```
dhcp client clear
```

EXAMPLE:

```

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

```

RELATED COMMANDS:

dhcp client stats

Show DHCP client statistics.

dhcp client config

Show/set DHCP client configuration.

SYNTAX:

<i>dhcp client config</i> <i>[trace <{off on}>]</i>

<i>[trace]</i>	Enable tracing (on) or not (off).	OPTIONAL
----------------	-----------------------------------	----------

EXAMPLE:

<pre>=>dhcp client config tracing: off =>dhcp client config trace on =>dhcp client config tracing: on =></pre>
--

RELATED COMMANDS:

<i>dhcp client ifconfig</i>	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 : 00:90:d0:01:47:de
              DHCP server: 255.255.255.255
              hostname : NewLease
              req.lease time = 10800 s
              trying to get a lease for 8 min, 32 sec
              transmission of DISCOVER in 57 sec
              retransmission timeout: 64
              nbr of retransmissions: 14
Number of leases: 1
Total size of table: 19, in use: 1, free: 94 %
=>dhcp client save
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 load
=>dhcp client iflist
NewMer      : [REBOOTING]
              flags = uc
              IP address : 10.0.0.10
              HW address : 00:90:d0:01:47:de
              DHCP server: 255.255.255.255
              hostname : NewLease
              req.lease time = 10800 s
              trying to request known lease for 2 sec
              retransmission of REQUEST in 2 sec
              retransmission timeout : 4
              retransmissions left before reinitializing : 2
Number of leases: 1
Total size of table: 19, in use: 1, free: 94 %
=>
```

RELATED COMMANDS:

dhcp client load

Load saved or default DHCP client configuration and dynamic interfaces.

dhcp client save

Save current DHCP client configuration and dynamic interfaces.

dhcp client ifadd

Create a DHCP lease for a specific interface.

SYNTAX:

<i>dhcp client ifadd</i>	<i>intf = <interface name></i>
---------------------------------	---

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

EXAMPLE:

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

RELATED COMMANDS:

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

dhcp client ifattach

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

SYNTAX:

dhcp client ifattach intf = <interface name>
--

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

EXAMPLE:

```
=>dhcp client iflist
NewMer      : [INIT]
              flags = uc
              IP address : 0.0.0.0
              HW address : 00:90:d0:01:47:de
              DHCP server: 255.255.255.255
Number of leases: 1
Total size of table: 19, in use: 1, free: 94 %
=>dhcp client ifattach intf NewMer
=>dhcp client iflist
NewMer      : [SELECTING]
              flags = uc
              IP address : 0.0.0.0
              HW address : 00:90:d0:01:47:de
              DHCP server: 255.255.255.255
              trying to get a lease for 8 sec
              transmission of DISCOVER in 4 sec
              retransmission timeout: 8
              nbr of retransmissions: 3
Number of leases: 1
Total size of table: 19, in use: 1, free: 94 %
=>
```

RELATED COMMANDS:

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

dhcp client ifconfig

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

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

SYNTAX:

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

<i>intf</i>	The name of the dynamic interface to be configured.	REQUIRED
[<i>clientid</i>]	The client identity to be associated with the lease.	OPTIONAL
[<i>hostname</i>]	The host name of the client to be associated with the lease.	OPTIONAL
[<i>addr</i>]	The preferred dynamic IP address.	OPTIONAL
[<i>leasetime</i>]	The time in seconds the client wants to use an address.	OPTIONAL
[<i>addrtrans</i>]	Automatically enable address translation for this dynamic interface (pat) or not (none).	OPTIONAL
[<i>dns</i>]	Request (and accept) DNS server IP addresses (on) or not (off).	OPTIONAL
[<i>gateway</i>]	Request (and accept) gateway IP addresses (on) or not (off).	OPTIONAL

EXAMPLE:

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

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 : 0.0.0.0
              HW address : 00:90:d0:01:47:de
              DHCP server: 255.255.255.255
              trying to get a lease for 2 h, 16 min, 54 sec
              transmission of DISCOVER in 53 sec
              retransmission timeout: 64
              nbr of retransmissions: 134
Number of leases: 1
Total size of table: 19, in use: 1, free: 94 %
=>dhcp client ifdelete intf NewMer
=>dhcp client iflist
No dynamic interfaces defined.
=>
```

RELATED COMMANDS:

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

dhcp client iflist

Show all dynamic interfaces.

SYNTAX:

```
dhcp client iflist
```

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 %
=>
```

EXAMPLE INPUT/OUTPUT IN A NETWORKED ENVIRONMENT (**SpeedTouch™** as DHCP client):

```
=>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:

<i>dhcp client ifrelease</i> <i>intf = <interface name></i>

<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 : 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 6 min, 7 sec
              transmission of DISCOVER in 13 sec
              retransmission timeout: 64
              nbr of retransmissions: 10
Number of leases: 1
Total size of table: 19, in use: 1, free: 94 %
=>dhcp client ifrelease intf NewMer
=>dhcp client iflist
NewMer      : [INIT]
              flags = uc
              IP address : 0.0.0.0
              HW address : 00:90:d0:01:47:de
              DHCP server: 255.255.255.255
              hostname   : NewLease
              req.lease time = 10800 s
Number of leases: 1
Total size of table: 19, in use: 1, free: 94 %
=>
```

EXAMPLE INPUT/OUTPUT IN A NETWORKED ENVIRONMENT (**SpeedTouch™** as DHCP client):

```

=>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 recv :          0
DECLINEs sent         :          0
RELEASEs sent         :          0
INFORMs sent          :          0
Number of dynamic interfaces: 1
Memory usage:
Table size of dyn leases: 18, in use: 1, free: 94 %
=>dhcp client ifrelease intf eth0
=>(CTRL + Q)
=>STATE ACTIVATE !
STATE IDLE !
STATE ACTIVATE !
dhcc: intf 1 releases 10.0.0.3 to server 10.10.1.1.
dhcc: 10.0.0.3 deleted: ok.
STATE IDLE !
STATE ACTIVATE !
.....
dhcc: intf 1 in init state.
n_send() broadcast triggered; To be verified
dhcc: broadcast discover on intf 1.
=>(CTRL + S)
=>dhcp client stats
DHCP client statistics:
Corrupted packet recv :          0
DECLINEs sent         :          0
RELEASEs sent         :          1
INFORMs sent          :          0
Number of dynamic interfaces: 1
Memory usage:
Table size of dyn leases: 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:

<i>dhcp client ifrenew</i> <i>intf = <interface name></i>

<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 : 00:90:d0:01:47:de
              DHCP server: 255.255.255.255
              hostname   : NewLease
              req.lease time = 10800 s
              trying to get a lease for 8 min, 13 sec
              transmission of DISCOVER in 49 sec
              retransmission timeout: 64
              nbr of retransmissions: 13
Number of leases: 1
Total size of table: 19, in use: 1, free: 94 %
=>dhcp client ifrenew intf = NewMer
=>dhcp client iflist
NewMer      : [SELECTING]
              flags = uc
              IP address : 10.0.0.10
              HW address : 00:90:d0:01:47:de
              DHCP server: 255.255.255.255
              hostname   : NewLease
              req.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 %
=>
```

EXAMPLE INPUT/OUTPUT IN A NETWORKED ENVIRONMENT (**SpeedTouch™** as DHCP client):

```

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

```

RELATED COMMANDS:

dhcp client ifadd

Create a DHCP lease for a specific interface.

dhcp client ifattach

Attach a DHCP lease to an interface.

dhcp client load

Load saved (or default) DHCP client configuration and dynamic interfaces.

Execute **dhcp client flush** prior to **dhcp client load**.

SYNTAX:

<i>dhcp client load</i>

EXAMPLE:

```
=>dhcp client iflist
NewMer      : [SELECTING]
              flags = uc
              IP address : 10.0.0.10
              HW address : 00:90:d0:01:47:de
              DHCP server: 255.255.255.255
              hostname : NewLease
              req.lease time = 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 save
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 load
=>dhcp client iflist
NewMer      : [REBOOTING]
              flags = uc
              IP address : 10.0.0.10
              HW address : 00:90:d0:01:47:de
              DHCP server: 255.255.255.255
              hostname : NewLease
              req.lease time = 10800 s
              trying to request known lease for 2 sec
              retransmission of REQUEST in 2 sec
              retransmission timeout : 4
              retransmissions left before reinitializing : 2
Number of leases: 1
Total size of table: 19, in use: 1, free: 94 %
=>
```

RELATED COMMANDS:

dhcp client flush

Delete all dynamic interfaces.

dhcp client save

Save current DHCP client configuration and dynamic interfaces.

dhcp client save

Save current DHCP client configuration and dynamic interfaces.

SYNTAX:

dhcp client save

EXAMPLE:

```
=>dhcp client iflist
NewMer      : [SELECTING]
              flags = uc
              IP address : 10.0.0.10
              HW address : 00:90:d0:01:47:de
              DHCP server: 255.255.255.255
              hostname : NewLease
              req.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 save
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 load
=>dhcp client iflist
NewMer      : [REBOOTING]
              flags = uc
              IP address : 10.0.0.10
              HW address : 00:90:d0:01:47:de
              DHCP server: 255.255.255.255
              hostname : NewLease
              req.leasetime = 10800 s
              trying to request known lease for 2 sec
              retransmission of REQUEST in 2 sec
              retransmission timeout : 4
              retransmissions left before reinitializing : 2
Number of leases: 1
Total size of table: 19, in use: 1, free: 94 %
=>
```

RELATED COMMANDS:

dhcp client flush
dhcp client load

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

dhcp client stats

Show DHCP client statistics.

SYNTAX:

```
dhcp client stats
```

EXAMPLE:

```
=>dhcp client stats
DHCP client statistics:
Corrupted packet recv      :          0
OFFERs      recv          :          0
ACKs        recv          :          0
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 %
=>
```

EXAMPLE INPUT/OUTPUT IN A NETWORKED ENVIRONMENT (**SpeedTouch™** as DHCP client):

```
=>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          :          4
REQUESTs    sent          :          1
DECLINEs    sent          :          0
RELEASEs    sent          :          0
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 clear Clear DHCP client statistics.

dhcp server add

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

SYNTAX:

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

<i>clientid</i>	The DHCP client's MAC address.	REQUIRED
<i>addr</i>	The IP address for this DHCP host.	REQUIRED
<i>[leasetime]</i>	A number between 0 and (seconds). Represents the time the host is allowed to use this address, before renewing. Not specifying this parameter makes the lease permanent. By default the lease time is 7200 seconds.	OPTIONAL
<i>[hostname]</i>	The hostname to add to the local DNS table for this host.	OPTIONAL

EXAMPLE:

```
=>dhcp server list
Leases:
Lease 0: 01:00:A0:24:AE:66:E1
    Hostname = Default
    ip address: 10.0.0.8
    expires in: 1 h, 16 min, 20 sec
    lease is being used.
Total size of table: 36, in use: 1 free: 97 %
=>dhcp server add clientid 01:23:55:67:89:ab addr 10.0.0.1 leasetime 7200
    hostname NewLease
=>dhcp server list
Leases:
Lease 0: 01:00:A0:24:AE:66:E1
    Hostname = Default
    ip address: 10.0.0.8
    expires in: 1 h, 15 min, 19 sec
    lease is being used.
Lease 1: 01:23:55:67:89:AB
    Hostname = NewLease
    ip address: 10.0.0.1
    expires in: 1 h, 59 min, 52 sec
    lease is not being used.
Total size of table: 36, in use: 2 free: 94 %
=>
```

RELATED COMMANDS:

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

dhcp server client

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

SYNTAX:

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

EXAMPLE:

```
=>dhcp server status
DHCP Server Status: Running
Current configuration:
    Address Range: 10.0.0.1 ... 10.255.255.254
.....
Policies:
    Verify first: no
Spoofing parameters:
    Failure timeout (!DoD): 4 sec
Start-up client parameters:
    Timeout: 20 sec
Tracing: off
Memory usage:
    Leases: total: 36, in use: 7 free: 80 %
=>dhcp server client timeout 15
=>dhcp server status
DHCP Server Status: Running
Current configuration:
    Address Range: 10.0.0.1 ... 10.255.255.254
.....
Policies:
    Verify first: no
Spoofing parameters:
    Failure timeout (!DoD): 4 sec
Start-up client parameters:
    Timeout: 15 sec
Tracing: off
Memory usage:
    Leases: total: 36, in use: 7 free: 80 %
=>
```

RELATED COMMANDS:

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

dhcp server clrstats

Clear **SpeedTouch™** DHCP server statistics.

SYNTAX:

<i>dhcp server clrstats</i>

EXAMPLE:

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

RELATED COMMANDS:

dhcp server stats Show DHCP server statistics.

dhcp server config

Set **SpeedTouch**™ DHCP server configuration.

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

SYNTAX:

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

EXAMPLE:

```

=>dhcp server status
DHCP Server Status: Running
Current configuration:
  Address Range: 10.0.0.1 ... 10.255.255.254
  Netmask: 255.0.0.0
  Lease time: 10800 seconds
  Gateway (default router): 10.0.0.138
  DNS server: 10.0.0.1
  Domain name: lan
Policies:
  Verify first: no
  Trust client: yes
  Spoofing: no
  Start as client: yes
Spoofing parameters:
  Failure timeout (!DoD): 4 sec
  Failure lease time (!DoD): 60 sec
  Temp. lease time (DoD): 10 sec
Start-up client parameters:
  Timeout: 15 sec
Tracing: off
Memory usage:
  Leases: total: 36, in use: 7 free: 80 %
=>dhcp server config beginrange=172.16.0.0 endrange=172.16.0.122 netmask=255.0.0.0
  leasetime=21600 gateway=172.16.1.1 dnsaddr=172.16.1.1
=>dhcp server status
DHCP Server Status: Running
Current configuration:
  Address Range: 172.16.0.0 ... 172.16.0.122
  Netmask: 255.0.0.0
  Lease time: 21600 seconds
  Gateway (default router): 172.16.1.1
  DNS server: 172.16.1.1
  Domain name: lan
Policies:
  Verify first: no
  Trust client: yes
  Spoofing: no
  Start as client: yes
Spoofing parameters:
  Failure timeout (!DoD): 4 sec
  Failure lease time (!DoD): 60 sec
  Temp. lease time (DoD): 10 sec
Start-up client parameters:
  Timeout: 15 sec
Tracing: off
Memory usage:
  Leases: total: 36, in use: 6 free: 83 %
=>

```

RELATED COMMANDS:

dhcp server status Show current DHCP server configuration.

dhcp server delete

Delete a DHCP lease.

SYNTAX:

dhcp server delete	index <number>
---------------------------	-----------------------------

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

EXAMPLE:

```
=>dhcp server list
Leases:
Lease 1: 01:23:55:67:89:AB
    Hostname = NewLease
    ip address: 10.0.0.1
    expires in: 1 h, 21 min, 9 sec
    lease is not being used.
Lease 0: 01:00:A0:24:AE:66:E1
    Hostname = Default
    ip address: 10.0.0.8
    expires in: 1 h, 31 min, 36 sec
    lease is being used.
Total size of table: 36, in use: 2 free: 94 %
=>dhcp server delete index 1
=>dhcp server list
Leases:
Lease 0: 01:00:A0:24:AE:66:E1
    Hostname = Default
    ip address: 10.0.0.8
    expires in: 1 h, 31 min, 18 sec
    lease is being used.
Total size of table: 36, in use: 1 free: 97 %
=>
```

RELATED COMMANDS:

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

dhcp server flush

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

SYNTAX:

dhcp server flush

EXAMPLE:

```
=>dhcp server list
Leases:
Lease 2: 01:52:41:53:20:A0:1B:A7:EB:AD:3C:C0:01:01:00:00:00
    ip address: 10.0.7.79
    expires in: 11 sec
    lease is not being used.
Lease 1: 01:52:41:53:20:20:4D:0D:CB:03:40:C0:01:01:00:00:00
    ip address: 10.0.7.62
    Spoofed lease from 2: DHCP_SPOOF
    Assigned (temporary) private ip address.
    expires in: 1 min, 39 sec
    lease is not being used.
Lease 0: 01:00:A0:24:AE:66:E1
    Hostname = Default
    ip address: 10.0.0.8
    expires in: 1 h, 57 min, 9 sec
    lease is being used.
Lease 3: 01:23:55:67:89:AB
    Hostname = Tempo
    ip address: 10.0.0.1
    never expires!
    lease is not being used.
Total size of table: 36, in use: 7 free: 80 %
=>dhcp server flush
=>dhcp server list
No active leases
Total size of table: 36, in use: 0 free: 100 %
=>
```

RELATED COMMANDS:

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

dhcp server list

List current DHCP leases, indicated by their index number.

SYNTAX:

```
dhcp server list
```

EXAMPLE OUTPUT:

```
=>dhcp server list
Leases:
Lease 2: 01:52:41:53:20:50:6D:C0:40:02:32:C0:01:01:00:00:00
    ip address: 10.0.7.142
    expires in: 3 sec
    lease is not being used.
Lease 3: 01:52:41:53:20:A0:1B:A7:EB:AD:3C:C0:01:01:00:00:00
    ip address: 10.0.7.143
    expires in: 17 sec
    lease is not being used.
Lease 5: 01:52:41:53:20:F0:90:8F:09:E1:35:BE:01:01:00:00:00
    ip address: 10.0.7.144
    expires in: 55 sec
    lease is not being used.
Lease 6: 01:52:41:53:20:30:F4:89:5F:9B:44:C0:01:01:00:00:00
    ip address: 10.0.7.145
    expires in: 1 min, 6 sec
    lease is not being used.
Lease 1: 01:52:41:53:20:20:4D:CB:03:40:C0:01:01:00:00:00
    ip address: 10.0.7.62
    Spoofed lease from 2: DHCP_SPOOF
    Assigned (temporary) private ip address.
    expires in: 1 min, 57 sec
    lease is not being used.
Lease 0: 01:00:A0:24:AE:66:E1
    Hostname = Default
    ip address: 10.0.0.8
    expires in: 1 h, 17 min, 21 sec
    lease is being used.
Total size of table: 36, in use: 7 free: 80 %
=>
```

RELATED COMMANDS:

dhcp server add

Add a DHCP lease manually.

dhcp server delete

Delete a DHCP lease.

dhcp server flush

Delete complete DHCP server configuration and dynamic leases.

dhcp server load

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

SYNTAX:

dhcp server load	[defaults <yes no>]
-------------------------	----------------------------------

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

EXAMPLE:

```
=>dhcp server status
DHCP Server Status: Stopped
.....
=>dhcp server load defaults yes
=>dhcp server status
DHCP Server Status: Running
Current configuration:
    Address Range: 10.0.0.1 ... 10.255.255.254
    Netmask: 255.0.0.0
    Lease time: 7200 seconds
    Gateway (default router): 11.11.144.144
    DNS server: 10.0.0.1
    Domain name: lan
Policies:
    Verify first: no
    Trust client: yes
    Spoofing: no
    Start as client: yes
Spoofing parameters:
    Failure timeout (!DoD): 4 sec
    Failure lease time (!DoD): 60 sec
    Temp. lease time (DoD): 10 sec
Start-up client parameters:
    Timeout: 20 sec
Tracing: off
Memory usage:
    Leases: total: 36, in use: 7 free: 80 %
=>
```

RELATED COMMANDS:

dhcp server flush	Flush current DHCP server configuration and dynamic leases.
dhcp server load	Load saved or default DHCP server configuration and permanent leases.

dhcp server policy

Set **SpeedTouch™** DHCP server policy.

SYNTAX:

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

[verifyfirst]	Probe the network for conflicting IP addresses before giving a suggested IP address to the requesting DHCP client (yes) or not (no).	OPTIONAL
[trustclient]	Take the IP address suggested by a DHCP client into account (yes) or not (no).	OPTIONAL
[spoofing]	Allow a remote DHCP server to hand out IP addresses negotiated by PPP on WAN side (yes) or not (no). DHCP spoofing is used to relay local DHCP requests to an external PPP connection having a specific IP address negotiation mechanism. DHCP replies are in turn generated by the DHCP server based on the IP address information received by the PPP link.	OPTIONAL
[client]	Allow the Speed Touch™ DHCP server to present itself as DHCP client (AutoDHCP mode) at boot time and probe for another DHCP server on the network for some time before starting the DHCP server (yes) or immediately start the DHCP server (no).	OPTIONAL

EXAMPLE:

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

RELATED COMMANDS:

dhcp server status Show current DHCP server configuration.

dhcp server save

Save complete **Speed Touch™** DHCP server configuration and permanent DHCP leases.

SYNTAX:

```
dhcp server save
```

EXAMPLE:

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

RELATED COMMANDS:

dhcp server flush

Flush complete DHCP server configuration and dynamic leases

dhcp server load

Load saved or default DHCP server configuration and permanent leases.

dhcp server spoof

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

SYNTAX:

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

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

EXAMPLE:

```
=>dhcp server status
DHCP Server Status: Running
.....
Spoofing parameters:
    Failure timeout (!DoD): 4 sec
    Failure lease time (!DoD): 60 sec
    Temp. lease time (DoD): 10 sec
Start-up client parameters:
    Timeout: 20 sec
Tracing: off
Memory usage:
    Leases: total: 36, in use: 6 free: 83 %
=>dhcp server spoof failtime 8 errorlt 120 dodlt 20
=>dhcp server status
DHCP Server Status: Running
.....
Spoofing parameters:
    Failure timeout (!DoD): 8 sec
    Failure lease time (!DoD): 120 sec
    Temp. lease time (DoD): 20 sec
Start-up client parameters:
    Timeout: 20 sec
Tracing: off
Memory usage:
    Leases: total: 36, in use: 5 free: 86 %
=>
```

RELATED COMMANDS:

dhcp server policy
dhcp server status

Set DHCP server policy.
Show current DHCP server configuration.

dhcp server start

Start **Speed Touch**™ DHCP server.

SYNTAX:

```
dhcp server start
```

EXAMPLE:

```
=>dhcp server status
DHCP Server Status: Stopped
Current configuration:
    Address Range: 10.0.0.1 ... 10.255.255.254
.....
    Leases: total: 36, in use: 0 free: 100 %
=>dhcp server start
=>dhcp server status
DHCP Server Status: Searching for server...
Current configuration:
    Address Range: 10.0.0.1 ... 10.255.255.254
.....
    Leases: total: 36, in use: 0 free: 100 %
=>
=>dhcp server status
DHCP Server Status: Running
Current configuration:
    Address Range: 10.0.0.1 ... 10.255.255.254
.....
    Leases: total: 36, in use: 1 free: 97 %
=>
```

RELATED COMMANDS:

dhcp server status

Show current DHCP server configuration.

dhcp server stop

Stop DHCP server.

dhcp server stats

Show **Speed Touch**™ DHCP server statistics.

SYNTAX:

```
dhcp server stats
```

EXAMPLE OUTPUT:

```
=>dhcp server stats
DHCP server statistics:
Corrupted packet recv :      0
DISCOVER               :    2451
REQUEST                :      19
DECLINE                :       0
RELEASE                :       0
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
=>
```

DESCRIPTION:

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

<i>ACKs sent</i>	Indicates the number of ACKnowledgement replies were sent to successfully configured DHCP clients.
<i>NAKs sent</i>	Indicates the number of Not-AcKnowledge ment 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 Speed Touch™ DHCP server to verify whether the IP address is already in use on the LAN or not. (dhcp server policy verifyfirst=yes)
<i>Second DHCP server seen</i>	Indicates whether a concurrent DHCP server was found on the LAN or not.

RELATED COMMANDS:

dhcp server clrstats Clear DHCP server statistics.

dhcp server status

Show current DHCP server configuration.

SYNTAX:

<i>dhcp server status</i>

EXAMPLE OUTPUT IN A NETWORKED ENVIRONMENT (**SpeedTouch™** as DHCP client):

```
=>dhcp server status
DHCP Server Status: Client
Current configuration:
  Address Range: 10.0.0.1 ... 10.255.255.254
  Netmask: 255.0.0.0
  Lease time: 7200 seconds
  Gateway (default router): 10.0.0.3 (auto)
  DNS server: 10.0.0.3 (auto)
  Domain name: SpeedTouch.local
Policies:
  Verify first: no
  Trust client: yes
  Spoofing: no
  Start as client: yes
Spoofing parameters:
  Failure timeout (!DoD): 4 sec
  Failure lease time (!DoD): 60 sec
  Temp. lease time (DoD): 10 sec
Start-up client parameters:
  Timeout: 20 sec
Tracing: off
Memory usage:
Leases: total: 36, in use: 0 free: 100 %
=>
```

RELATED COMMANDS:

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

dhcp server stop

Stop **Speed Touch**™ DHCP server.

SYNTAX:

```
dhcp server stop
```

EXAMPLE:

```
=>dhcp server status
DHCP Server Status: Running
Current configuration:
    Address Range: 10.0.0.1 ... 10.255.255.254
.....
    Leases: total: 36, in use: 1 free: 97 %
=>dhcp server stop
=>dhcp server status
DHCP Server Status: Stopped
Current configuration:
    Address Range: 10.0.0.1 ... 10.255.255.254
.....
    Leases: total: 36, in use: 0 free: 100 %
=>
```

RELATED COMMANDS:

dhcp server start

Start DHCP server.

dhcp server status

Show current DHCP server configuration.

dhcp server troff

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

SYNTAX:

<i>dhcp server troff</i>

EXAMPLE:

```
=>dhcp server status
DHCP Server Status: Running
.....
Start-up client parameters:
    Timeout: 20 sec
Tracing: on
Memory usage:
    Leases: total: 36, in use: 0 free: 100 %
=>dhcp server troff
=>dhcp server status
DHCP Server Status: Running
.....
Start-up client parameters:
    Timeout: 20 sec
Tracing: off
Memory usage:
    Leases: total: 36, in use: 1 free: 97 %
=>
```

RELATED COMMANDS:

dhcp server status	Show current DHCP server configuration.
dhcp server tron	Enable verbose console logging.

dhcp server tron

Enable verbose console logging. Debug traces are generated.

SYNTAX:

<i>dhcp server tron</i>

EXAMPLE:

<pre>=>dhcp server status DHCP Server Status: Running Start-up client parameters: Timeout: 20 sec <u>Tracing: off</u> Memory usage: Leases: total: 36, in use: 1 free: 97 % =>dhcp server tron =>dhcp server status DHCP Server Status: Running Start-up client parameters: Timeout: 20 sec <u>Tracing: on</u> Memory usage: Leases: total: 36, in use: 1 free: 97 % =></pre>
--

RELATED COMMANDS:

dhcp server status

Show current DHCP server configuration

dhcp server troff

Disable verbose console logging.

7 DNS Commands

dns (to access the DNS level)

dns add

dns clear

dns clrstats

dns delete

dns domain

dns flush

dns fwdadd

dns fwddelete

dns fwdlist

dns fwdtable

dns list

dns load

dns nslookup

dns rem

dns save

dns start

dns stats

dns status

dns stop

dns toutfwd

dns troff

dns tron

dns add

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

SYNTAX:

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

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

EXAMPLE:

```
=>dns list
Domain: lan
Nr.      Hostname      IP Address
 0      TestHost      10.0.0.140
 1      Default        10.0.0.8
Total Table Size: 73 entries
Amount used: 2 (2%)
=>dns add hostname ftpserver addr 172.16.0.1
=>dns list
Domain: lan
Nr.      Hostname      IP Address
 0      TestHost      10.0.0.140
 1      Default        10.0.0.8
 2      ftpserver      172.16.0.1
Total Table Size: 73 entries
Amount used: 3 (4%)
=>
```

RELATED COMMANDS:

dns list	List current DNS entries.
dns delete	Delete a DNS entry (via its index number).
dns rem	Delete a DNS entry (via its hostname).

dns clear

Delete current DNS entries.

SYNTAX:

<i>dns clear</i>

EXAMPLE:

```
=>dns list
Domain: lan
Nr.      Hostname      IP Address
  0      TestHost      10.0.0.140
  1      Default       10.0.0.8
  2      ftpserver      172.16.0.1
Total Table Size: 73 entries
Amount used: 3 (4%)
=>dns clear
DNS database now empty.
=>dns list
Domain: 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:

<i>dns clrstats</i>

EXAMPLE:

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

RELATED COMMANDS:

dns stats	Show DNS server/forwarder statistics.
------------------	---------------------------------------

dns delete

Delete a DNS entry (via its index number).

SYNTAX:

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

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

EXAMPLE:

```
=>dns list
Domain: lan
Nr.      Hostname      IP Address
  0      TestHost      10.0.0.140
  1      Default       10.0.0.8
  2      ftpserver     172.16.0.1
Total Table Size: 73 entries
Amount used: 3 (4%)
=>dns delete index 0
Successfully removed.
=>dns list
Domain: lan
Nr.      Hostname      IP Address
  1      Default       10.0.0.8
  2      ftpserver     172.16.0.1
Total Table Size: 73 entries
Amount used: 2 (2%)
=>
```

RELATED COMMANDS:

dns add	Add a static DNS entry.
dns list	List current DNS entries.
dns rem	Delete a DNS entry (via its hostname).

dns domain

Set local DNS (sub)domain name.

SYNTAX:

<i>dns domain</i>	<i>domain <string></i>
--------------------------	-------------------------------------

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

EXAMPLE:

```
=>dns list
Domain: lan
Nr.      Hostname      IP Address
  0      TestHost      10.0.0.140
  1      Default        10.0.0.8
  2      peckelbs_pc     172.16.0.1
Total Table Size: 73 entries
Amount used: 3 (4%)
=>dns domain domain office.home.dom
=>dns list
Domain: office.home.dom
Nr.      Hostname      IP Address
  0      TestHost      10.0.0.140
  1      Default        10.0.0.8
  2      peckelbs_pc     172.16.0.1
Total Table Size: 73 entries
Amount used: 3 (4%)
=>
```

RELATED COMMANDS:

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

dns flush

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

SYNTAX:

<i>dns flush</i>

EXAMPLE:

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

RELATED COMMANDS:

dns load

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

dns save

Save current DNS server/forwarder configuration and static entries

dns fwdadd

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

SYNTAX:

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

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

EXAMPLE:

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

RELATED COMMANDS:

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

dns fwddelete

Delete a DNS forwarding entry.

SYNTAX:

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

<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
192.6.11.150	192.6.0.0	255.255.0.0	yes
10.0.0.138	10.0.0.0	255.0.0.0	yes
=>dns fwddelete src 10.0.0.0 mask 255.0.0.0 dns 10.0.0.138			
Dns forwarding server deleted.			
=>dns fwdlist			
DNS forwarding servers:			
DNS	SRC	MASK	Direct
192.6.11.150	192.6.0.0	255.255.0.0	yes
=>			

RELATED COMMANDS:

<i>dns fwdadd</i>	Add a DNS forwarding entry.
<i>dns fwdlist</i>	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	10.0.0.2	255.255.255.0	yes
10.0.0.138	10.0.0.3	255.255.255.0	yes
=>			

RELATED COMMANDS:

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

dns fwdtable

Show DNS forwarding table, i.e. list all 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: 1
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.dom
Nr.   Hostname      IP Address
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: 4 (5%)
=>
```

EXAMPLE INPUT/OUTPUT IN A NETWORKED ENVIRONMENT (**SpeedTouch™** as DNS server):

```
=>dns list
Domain: SpeedPro.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 rem	Delete a DNS entry (via its hostname).

dns load

Load saved or default **Speed Touch**™ DNS server/forwarder configuration and static DNS entries. Execute **dns flush** prior to **dns load**.

SYNTAX:

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

EXAMPLE:

```
=>dns list
Domain: office.home.dom
Nr.      Hostname      IP Address
  0      TestHost      10.0.0.140
  1      Default        10.0.0.8
  2      peckelbs_pc     172.16.0.1
Total Table Size: 73 entries
Amount used: 3 (4%)
=>dns save
=>dns flush
=>dns list
Domain: office.home.dom
Nr.      Hostname      IP Address
Total Table Size: 73 entries
Amount used: 0 (0%)
=>dns load defaults = yes
=>dns list
Domain: lan
Nr.      Hostname      IP Address
  0      SpeedTouch     *.*.*.*
Total Table Size: 73 entries
Amount used: 1 (1%)
=>dns flush
=>dns load defaults = no
=>dns list
Domain: lan
Nr.      Hostname      IP Address
  0      TestHost      10.0.0.140
  1      Default        10.0.0.8
  2      peckelbs_pc     172.16.0.1
Total Table Size: 73 entries
Amount used: 3 (4%)
=>
```

RELATED COMMANDS:

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

dns nslookup

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

SYNTAX:

<i>dns nslookup</i>	<i>lookup <string></i>
----------------------------	-------------------------------------

lookup The DNS hostname or IP address to query.

EXAMPLE:

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

RELATED COMMANDS:

dns list List current DNS entries.

dns save

Save current **Speed Touch**™ DNS server/forwarder configuration and static entries.

SYNTAX:

```
dns save
```

EXAMPLE:

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

RELATED COMMANDS:

dns flush

Flush complete DNS server/forwarder configuration and dynamic entries.

dns load

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

dns start

Start **Speed Touch**™ DNS server/forwarder.

SYNTAX:

<i>dns start</i>

EXAMPLE:

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

RELATED COMMANDS:

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

dns stats

Show **Speed Touch**™ DNS server/forwarder statistics.

SYNTAX:

```
dns stats
```

EXAMPLE INPUT/OUTPUT IN A NETWORKED ENVIRONMENT (**SpeedTouch**™ as DNS server):

```
=>dns list
Domain: SpeedPro.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.SpeedPro.local)
=>(CTRL + Q)
dnisd: Internet class type A request received from 10.10.1.1.
dnisd: Client.Speedpro.local found in local database.
dnisd: Client.Speedpro.local resolved into 10.0.0.3.
=>(Ping Server.SpeedPro.local)
dnisd: Internet class type A request received from 10.10.1.1.
dnisd: Server.SpeedPro.local found in local database.
dnisd: Server.SpeedPro.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 **Speed Touch**™ DNS server/forwarder configuration.

SYNTAX:

```
dns status
```

EXAMPLE OUTPUT:

```
=>dns status  
DNS server status: Started  
DNS table size : 73, in use: 3, free: 95 %  
DNS forwarding table size : 10, in use: 0, free: 100 %  
DNS forwarding dns servers table size : 25, in use: 0, free: 100 %  
No dns cache.  
Tracing: off  
=>
```

RELATED COMMANDS:

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

dns stop

Stop **Speed Touch**™ DNS server/forwarder.

SYNTAX:

```
dns stop
```

EXAMPLE:

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

RELATED COMMANDS:

dns status

Show DNS server/forwarder configuration.

dns start

Start DNS server/forwarder.

dns toutfwd

Set DNS forwarding timeout.

SYNTAX:

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

timeout	<p>A number between 0 and (seconds). REQUIRED</p> <p>Represents the query forwarding timeout.</p> <p>This parameter determines how long the Speed Touch™ DNS server should try to contact a (remote) DNS server before (temporarily) declaring the DNS requests unresolved.</p> <p>By default the timeout is 15 seconds.</p>
----------------	--

EXAMPLE:

```
=>dns fwdtable
Forwarding table:
Nr.      Ip Address      (port#): id(hex) (expiry)      dns server      tries
Timeout: 15 seconds
Table size: 10
Amount of table used: 0 (0%)
=>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
Timeout: 20 seconds
Table size: 10
Amount of table used: 0 (0%)
=>
```

RELATED COMMANDS:

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

dns troff

Disable verbose console messaging. No debug traces are generated.

SYNTAX:

```
dns troff
```

EXAMPLE:

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

RELATED COMMANDS:

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

dns tron

Enable verbose console messaging. Debug traces are generated.

SYNTAX:

```
dns tron
```

EXAMPLE:

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

RELATED COMMANDS:

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

8 Firewall Commands

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

firewall assign

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

SYNTAX:

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

EXAMPLE:

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

RELATED COMMANDS:

firewall chain create	Create a chain.
firewall chain list	Show a list of all current chains.
firewall chain load	Load saved or default chain configuration(s).
firewall chain save	Save current chain(s) configuration.

firewall chain create

Create a new chain.

SYNTAX:

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

chain

The name of the chain to create.

REQUIRED

EXAMPLE:

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

RELATED COMMANDS:

firewall assign

Assign a chain to an entry point.

firewall chain delete

Delete a chain.

firewall chain list

Show a list of all current chains.

firewall chain load

Load saved or default chain configuration(s).

firewall chain save

Save current chain(s) configuration.

firewall chain delete

Delete a chain.

SYNTAX:

<i>firewall chain delete chain = <string></i>
--

<i>chain <string></i>	The name of the chain to be deleted.	REQUIRED
-----------------------------	--------------------------------------	----------

EXAMPLE:

<pre>=>firewall chain list Telnet, Tempo, source, forward, sink =>firewall chain list Telnet, <u>Tempo</u>, source, forward, sink =>firewall chain delete chain Tempo =>firewall chain list Telnet, source, forward, sink =></pre>
--

RELATED COMMANDS:

RELATED COMMANDS:

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

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 save file TelConfig
=>firewall chain list file Telconfig
Telnet, source, forward, sink
=>
```

RELATED COMMANDS:

firewall assign	Assign a chain to an entry point.
firewall chain create	Create a chain.
firewall chain delete	Delete a chain.
firewall chain load	Load saved or default chain configuration(s).
firewall chain save	Save current chain(s) configuration.

firewall chain load

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

SYNTAX:

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

[file]	The name of the chain configuration to be loaded. Not specifying this parameter loads the default configuration	OPTIONAL
---------------	--	----------

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

EXAMPLE:

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

RELATED COMMANDS:

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

firewall chain save

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

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

SYNTAX:

firewall chain save [file = <string>]

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

EXAMPLE:

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

RELATED COMMANDS:

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

firewall flush

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

SYNTAX:

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

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

EXAMPLE:

```
=>firewall 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
=>
```

RELATED COMMANDS:

firewall assign	Assign a chain to an entry point.
firewall chain load	Load saved or default chain configuration(s).
firewall chain save	Save current chain(s) configuration.

firewall list

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

SYNTAX:

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

EXAMPLE INPUT/OUTPUT:

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

RELATED COMMANDS:

firewall assign

Assign a chain to an entry point.

firewall flush

Clear associations for all or a selected entry point(s).

firewall load

Load saved (or default) firewall configuration.

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

SYNTAX:

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

EXAMPLE:

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

RELATED COMMANDS:

firewall save

Save current firewall configuration.

firewall flush

Clear associations for all or a selected entry point(s).

firewall match

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

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

SYNTAX:

```
firewall match chain = <string>
[srcintf = <string>]
[src = <ip-address>]
[srcbridgeport = <number>]
[dstintf = <string>]
[dst = <ip-address>]
[tos = <number>]
[prot = <{tcp|udp|icmp|protocol}>]
[syn = <yes|no>]
[urg = <yes|no>]
[ack = <yes|no>]
[srcport = <{ftp|ftp-data|telnet|mail|smtp|dns|domain|tftp|port}>]
[dstport = <{ftp|ftp-data|telnet|mail|smtp|dns|domain|tftp|port}>]
[icmptype = <{echo-reply|destination-unreachable|source-quench|
redirect|echo-request|router-advertisement|
router-solicitation|time-exceeded|parameter-problems|
timestamp-request|timestamp-reply|
information-request|information-reply|
address-mask-request|address-mask-reply|
icmpnumber}>]
[icmpcode = <number{1-15}>]
```

<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>	The bridge port the virtual packet arrived on.	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 in the virtual packet.	OPTIONAL
<i>[prot]</i>	The protocol (name or number) in the virtual IP packet. Choose between: <ul style="list-style-type: none"> ▪ tcp ▪ udp ▪ icmp Or specify the protocol number.	OPTIONAL
<i>[syn]</i>	Set the TCP SYN flag (yes) or not (no).	OPTIONAL

<i>[urg]</i>	Set the TCP URG flag (yes) or not (no).	OPTIONAL
<i>[ack]</i>	Set the TCP ACK flag (yes) or not (no).	OPTIONAL
<i>[srcport]</i>	<p>The TCP/UDP port the virtual packet is coming from.</p> <p>Choose between:</p> <ul style="list-style-type: none">▪ ftp▪ ftp-data▪ telnet▪ mail▪ smtp▪ dns▪ domain▪ tftp <p>Or specify the port number.</p>	OPTIONAL
<i>[dstport]</i>	<p>The TCP/UDP port the virtual packet is going to.</p> <p>Choose between:</p> <ul style="list-style-type: none">▪ ftp▪ ftp-data▪ telnet▪ mail▪ smtp▪ dns▪ domain▪ tftp <p>Or specify the port number.</p>	OPTIONAL
<i>[icmptype]</i>	<p>The ICMP (Internet Control Message Protocol) type (name or number) of the virtual packet.</p> <p>Choose between:</p> <ul style="list-style-type: none">▪ 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 <p>Or specify the ICMP type number.</p>	OPTIONAL
<i>[icmpcode]</i>	<p>A number between 0 and 15.</p> <p>Represents the ICMP code of the virtual packet.</p>	OPTIONAL

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 rule clear

Clear statistics for a given rule.

SYNTAX:

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

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

EXAMPLE:

```
=>firewall rule stats
Chain Telnet, index 0, packets 0, bytes 0
Chain Telnet, index 1, packets 0, bytes 0
Chain Telnet, index 2, packets 0, bytes 0
Chain source, index 0, packets 203, bytes 11522
Chain source, index 1, packets 0, bytes 0
Chain source, index 2, packets 0, bytes 0
Chain forward, index 0, packets 0, bytes 0
Chain sink, index 0, packets 202, bytes 10159
Chain sink, index 1, packets 0, bytes 0
Chain sink, index 2, packets 0, bytes 0
=>firewall rule clear chain=source index=0
=>firewall rule stats
Chain Telnet, index 0, packets 0, bytes 0
Chain Telnet, index 1, packets 0, bytes 0
Chain Telnet, index 2, packets 0, bytes 0
Chain source, index 0, packets 11, bytes 559
Chain source, index 1, packets 0, bytes 0
Chain source, index 2, packets 0, bytes 0
Chain forward, index 0, packets 0, bytes 0
Chain sink, index 0, packets 409, bytes 21535
Chain sink, index 1, packets 0, bytes 0
Chain sink, index 2, packets 0, bytes 0
=>
```

RELATED COMMANDS:

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

firewall rule create

Create a rule.

SYNTAX:

```

firewall rule create  chain = <string>
                        [index = <number>]
                        [srcintf [!]= <string>]
                        [srcintfgrp [!]= <{wan|local|lan}>]
                        [srcbridgeport [!]= <number>]
                        [src [!]= <ip-address>]
                        [srcmsk = <ip-mask(dotted or cidr)>]
                        [dstintf [!]= <string>]
                        [dstintfgrp [!]= <{wan|local|lan}>]
                        [dst [!]= <ip-address>]
                        [dstmsk = <ip-mask(dotted or cidr)>]
                        [tos [!]= <number{1-255}>]
                        [prot [!]= <{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}>]
                        [icmp type [!]= <{echo-reply|destination-unreachable|source-quench|
                        redirect|echo-request|router-advertisement|
                        router-solicitation|time-exceeded|parameter-problems|
                        timestamp-request|timestamp-reply|
                        information-request|information-reply|
                        address-mask-request|address-mask-reply|
                        icmpnumber}>]
                        [icmpcode [!]= <number{0-15}>]
                        [icmpcodeend = <number{0-15}>]
                        [clink <string>]
                        action <{accept|deny|drop|count}>

```

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

<i>[src]</i>	The source IP address (range) the packet should [or should NOT] come from. (Supports cidr notation).	OPTIONAL
<i>[srcmsk]</i>	The source IP address mask defining the range (see src).	OPTIONAL
<i>[dstintf]</i>	The name of the interface the packet should [or should NOT] be going to. (NOT applicable if used in a chain assigned to the input hook)	OPTIONAL
<i>[dstintfgrp]</i>	The interface group the packet should [or should NOT] be going to. Choose between: <ul style="list-style-type: none">▪ wan▪ local▪ lan (NOT applicable if used in a chain assigned to the input hook)	OPTIONAL
<i>[dst]</i>	The destination IP address (range) the packet should [or should NOT] be going to. (supports cidr notation).	OPTIONAL
<i>[dstmsk]</i>	The destination IP address mask defining the range (see dst).	OPTIONAL
<i>[tos]</i>	A number between 0 and 255. Represents the Type Of Service specification expected [or NOT expected] in the IP packet.	OPTIONAL
<i>[prot]</i>	The protocol (name or number) in the IP packet expected [or NOT expected] in the IP packet. Choose between: <ul style="list-style-type: none">▪ tcp▪ udp▪ icmp Or specify the protocol number.	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>	The TCP/UDP port (or beginning of range) the packet should [or should NOT] be from. Choose between: <ul style="list-style-type: none">▪ ftp▪ ftp-data▪ telnet▪ mail▪ smtp▪ dns▪ domain▪ tftp Or specify the port number.	OPTIONAL
<i>[srcportend]</i>	The source TCP/UDP port range end (inclusive). (Only applicable for ranges)	OPTIONAL

<i>[dstport]</i>	<p>The TCP/UDP port (or beginning of range) the packet should [or should NOT] be going to.</p> <p>Choose between:</p> <ul style="list-style-type: none"> ▪ ftp ▪ ftp-data ▪ telnet ▪ mail ▪ smtp ▪ dns ▪ domain ▪ tftp <p>Or specify the port number.</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"> ▪ 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 <p>Or specify the ICMP type number.</p>	OPTIONAL
<i>[icmpcode]</i>	<p>A number between 0 and 15. Represents the expected [or NOT expected] ICMP code (or beginning of range) of the packet.</p>	OPTIONAL
<i>[icmpcodeend]</i>	<p>A number between 0 and 15. Represents the ICMP code range end. Only applicable for ranges.</p>	OPTIONAL
<i>[clink]</i>	<p>The name of the chain to be parsed when this rule applies. (action is ignored).</p>	OPTIONAL
<i>action</i>	<p>Action to be taken when this rule applies.</p> <p>Choose between:</p> <ul style="list-style-type: none"> ▪ accept : the packet may pass. ▪ deny : ICMP error destination unreachable. An error message is sent back to the sender. ▪ drop : packet disappears. It is silently dropped, that is, without sending an error message to the sender. ▪ count : update of statistics. Has no influence on the packet. 	REQUIRED

EXAMPLE:

```
=>firewall rule list chain Telnet
=>firewall rule create chain=telnet src=10.0.0.0/8 dst=200.200.200.1 srcintfgrp=lan
  prot=tcp srcport=1024 srcportend=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:

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

firewall rule delete

Delete a rule.

SYNTAX:

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

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

EXAMPLE:

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

RELATED COMMANDS:

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

firewall rule flush

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

SYNTAX:

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

[chain]

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
=>
```

REMARK:

No load command to (re)load rules in chains exists.

Instead, use **firewall chain load**, to load both chains and (associated) rules.

RELATED COMMANDS:

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

firewall rule list

Show a list of rules.

SYNTAX:

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

EXAMPLE INPUT AND OUTPUT:

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

RELATED COMMANDS:

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

firewall rule stats

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

SYNTAX:

firewall rule stats	[chain <string>] [index <number>]
----------------------------	--

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

EXAMPLE OUTPUT:

```
=>firewall rule list chain Test
:firewall rule create chain=Test index=0 srcintfgrp=lan src=200.200.0.1/32
dst=200.200.0.2/32 prot=udp srcport=0 srcportend=65535 dstport=telnet
action=deny
=>firewall rule clear
=>firewall rule stats
Chain sink, index 0, packets 43, bytes 1743
Chain sink, index 1, packets 0, bytes 0
Chain sink, index 2, packets 0, bytes 0
Chain sink, index 3, packets 0, bytes 0
Chain forward, index 0, packets 0, bytes 0
Chain source, index 0, packets 43, bytes 1977
Chain source, index 1, packets 0, bytes 0
Chain source, index 2, packets 0, bytes 0
Chain Test, index 0, packets 0, bytes 0
=>firewall rule stats
Chain sink, index 0, packets 104, bytes 6143
Chain sink, index 1, packets 0, bytes 0
Chain sink, index 2, packets 0, bytes 0
Chain sink, index 3, packets 0, bytes 0
Chain forward, index 0, packets 0, bytes 0
Chain source, index 0, packets 139, bytes 7348
Chain source, index 1, packets 0, bytes 0
Chain source, index 2, packets 0, bytes 0
Chain Test, index 0, packets 44, bytes 21032
=>
```

DESCRIPTION:

The statistics for the 'Test' chain are the result of sending udp packets to the **Speed Touch™**. 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 (**SpeedTouch™** as DHCP client):

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

RELATED COMMANDS:

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

firewall save

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

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

SYNTAX:

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

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

EXAMPLE:

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

RELATED COMMANDS:

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

firewall troff

Disable verbose console messaging.

SYNTAX:

<i>firewall troff</i>

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.

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 load
ip ping
ip rtadd
ip rtdelete
ip rtlist
ip save
ip sendto

ip apadd

Assign an IP address to an interface.

SYNTAX:

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

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

EXAMPLE:

```

=>ip aplist
1  eth0      Type:Ethernet HWaddr 00:80:9f:24:ab:cf BRHWaddr ff:ff:ff:ff:ff:ff
    inet addr:10.10.10.147      Bcast:10.10.10.255      Mask:255.0.0.0
    UP RUNNING pat MTU:1500 ReasmMAX:65535 Group:2
    IPRX bytes:19791886 unicastpkts:11341 brcastpkts:290555
    IPTX bytes:839550 unicastpkts:11477 brcastpkts:0 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:Ethernet HWaddr 00:80:9f:24:ab:cf BRHWaddr ff:ff:ff:ff:ff:ff
    inet addr:10.0.0.2      Bcast:10.0.0.255      Mask:255.255.255.0
    UP RUNNING pat MTU:1500 ReasmMAX:65535 Group:2
    IPRX bytes:19810763 unicastpkts:11515 brcastpkts:290669
    IPTX bytes:852562 unicastpkts:11661 brcastpkts:0 dropkts:0
    HWRX bytes:0 unicastpkts:0 brcastpkts:0
    HWTX bytes:0 unicastpkts:0 brcastpkts:0 dropkts:0
1  eth0      Type:Ethernet HWaddr 00:80:9f:24:ab:cf BRHWaddr ff:ff:ff:ff:ff:ff
    inet addr:10.10.10.147      Bcast:10.10.10.255      Mask:255.0.0.0
    UP RUNNING pat MTU:1500 ReasmMAX:65535 Group:2
    IPRX bytes: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 apdelete

Remove an IP address from an interface.

ip aplist

Show current IP addresses.

ip apdelete

Remove an IP address from an interface.

SYNTAX:

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

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

EXAMPLE:

```
=>ip aplist
1  eth0      Type:Ethernet HWaddr 00:80:9f:24:ab:cf BRHWaddr ff:ff:ff:ff:ff:ff
    inet addr:10.0.0.2      Bcast:10.0.0.255      Mask:255.255.255.0
    UP RUNNING pat MTU:1500 ReasmMAX:65535 Group:2
    IPRX bytes:20030209 unicastpkts:11536  brcastpkts:293337
    IPTX bytes:856124   unicastpkts:11685  brcastpkts:0      droppkts:0
    HWRX bytes:0        unicastpkts:0      brcastpkts:0
    HWTX bytes:0        unicastpkts:0      brcastpkts:0      droppkts:0
1  eth0      Type:Ethernet HWaddr 00:80:9f:24:ab:cf BRHWaddr ff:ff:ff:ff:ff:ff
    inet addr:10.10.10.147  Bcast:10.10.10.255  Mask:255.0.0.0
    UP RUNNING pat MTU:1500 ReasmMAX:65535 Group:2
    IPRX bytes:20040580 unicastpkts:11612  brcastpkts:293427
    IPTX bytes:862320   unicastpkts:11764  brcastpkts:0      droppkts:0
    HWRX bytes:0        unicastpkts:0      brcastpkts:0
    HWTX bytes:0        unicastpkts:0      brcastpkts:0      droppkts:0
=>ip apdelete addr 10.0.0.2
=>ip aplist
1  eth0      Type:Ethernet HWaddr 00:80:9f:24:ab:cf BRHWaddr ff:ff:ff:ff:ff:ff
    inet addr:10.10.10.147  Bcast:10.10.10.255  Mask:255.0.0.0
    UP RUNNING pat MTU:1500 ReasmMAX:65535 Group:2
    IPRX bytes:20040580 unicastpkts:11612  brcastpkts:293427
    IPTX bytes:862320   unicastpkts:11764  brcastpkts:0      droppkts:0
    HWRX bytes:0        unicastpkts:0      brcastpkts:0
    HWTX bytes:0        unicastpkts:0      brcastpkts:0      droppkts:0
=>
```

RELATED COMMANDS:

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

ip aplist

Show a list of all configured IP addresses.

SYNTAX:

```
ip aplist
```

EXAMPLE:

```
=>ip aplist
1  eth0      Type:Ethernet HWaddr 00:80:9f:24:ab:cf BRHWaddr ff:ff:ff:ff:ff:ff
    inet addr:10.10.10.147      Bcast:10.10.10.255      Mask:255.0.0.0
    UP RUNNING pat MTU:1500 ReasmMAX:65535 Group:2
    IPRX bytes:20030209 unicastpkts:11536 brcastpkts:293337
    IPTX bytes:856676 unicastpkts:11686 brcastpkts:0 dropkts:0
    HWRX bytes:0 unicastpkts:0 brcastpkts:0
    HWTX bytes:0 unicastpkts:0 brcastpkts:0 dropkts:0
5  cip0      Type:ATM
    inet addr:172.16.0.5      Bcast:172.16.0.255      Mask:255.255.255.0
    UP RUNNING MTU:9180 ReasmMAX:65535 Group:0
    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
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 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 **Speed Touch™** ARP cache.

SYNTAX:

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

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

EXAMPLE:

```
=>ip arplist
Intf      IP-address      HW-address      Type
eth0      10.0.0.1        00:01:42:5f:7d:81 DYNAMIC
eth0      10.0.0.8        00:a0:24:ae:66:e1 DYNAMIC
eth0      10.0.1.99       52:41:53:20:20:4d STATIC
eth0      10.0.1.100      52:41:53:20:f0:90 STATIC
=>ip arpadd intf eth0 ip 10.0.0.2 hwaddr 00:10:a4:d0:9a:db
=>ip arplist
Intf      IP-address      HW-address      Type
eth0      10.0.0.1        00:01:42:5f:7d:81 DYNAMIC
eth0      10.0.0.2        00:10:a4:d0:9a:db STATIC
eth0      10.0.0.8        00:a0:24:ae:66:e1 DYNAMIC
eth0      10.0.1.103      52:41:53:20:20:4d STATIC
eth0      10.0.0.166      08:00:20:91:c9:40 DYNAMIC
eth0      10.0.0.237      08:00:20:8e:8a:f2 DYNAMIC
=>ip arplist
intf      IP-address      HW-address      Type
eth0      10.0.0.1        00:01:42:5f:7d:81 DYNAMIC
eth0      10.0.0.2        00:10:a4:d0:9a:db STATIC
eth0      10.0.0.8        00:a0:24:ae:66:e1 DYNAMIC
eth0      10.0.1.111      52:41:53:20:20:4d STATIC
eth0      10.0.1.112      52:41:53:20:f0:90 STATIC
=>
```

RELATED COMMANDS:

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

ip arpdelete

Remove an entry from the **Speed Touch**™ ARP cache.

SYNTAX:

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

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

EXAMPLE:

=>ip arplist			
Intf	IP-address	HW-address	Type
eth0	10.0.0.1	00:01:42:5f:7d:81	DYNAMIC
<u>eth0</u>	<u>10.0.0.2</u>	<u>00:10:a4:d0:9a:db</u>	<u>STATIC</u>
eth0	10.0.0.8	00:a0:24:ae:66:e1	DYNAMIC
eth0	10.0.1.111	52:41:53:20:20:4d	STATIC
eth0	10.0.1.112	52:41:53:20:f0:90	STATIC
=>ip arpdelete intf eth0 ip 10.0.0.2 hwaddr 00:10:a4:d0:9a:db			
=>ip arplist			
intf	IP-address	HW-address	Type
eth0	10.0.0.1	00:01:42:5f:7d:81	DYNAMIC
eth0	10.0.0.8	00:a0:24:ae:66:e1	DYNAMIC
eth0	10.0.1.112	52:41:53:20:f0:90	STATIC
eth0	10.0.1.113	52:41:53:20:20:4d	STATIC
=>			

RELATED COMMANDS:

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

ip arplist

Show the **Speed Touch**™ ARP cache.

SYNTAX:

<i>ip arplist</i>

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	10.0.1.122	52:41:53:20:f0:90	STATIC
=>			

RELATED COMMANDS:

ip arpadd

Add a static entry to the ARP cache.

ip arpdelete

Delete an entry from the ARP cache.

ip config

Show/set global IP stack configuration options.

SYNTAX:

ip config	[forwarding = <{off on}>] [firewalling = <{off on}>] [redirects = <{off on}>] [sourcerouting = <{off on}>] [netbroadcasts = <{off on}>] [ttl = <number{0–255}>] [fraglimit = <number{1–1024}>] [defragmode = <{normal always nat}>] [addrcheck = <{off own static dynamic}>] [mssclamping = <{off on}>]	
[forwarding]	Disable (off) or enable (on) the IP routing functionality.	OPTIONAL
[firewalling]	Enable (on) or disable (off) IP firewalling (master switch). If applicable the CLI firewall level allows configuration of the Speed Touch™ firewall. For security reasons this parameter is enabled per default. It is strongly recommended never to disable the Speed Touch™ 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]	<p>Define which packets are reassembled under which circumstances. Choose between:</p> <ul style="list-style-type: none">▪ normal Packets to be forwarded will not be reassembled. Packets with local destination, i.e. destined for the Speed Touch™, are reassembled.▪ always Packets are always reassembled.▪ nat Same behaviour as normal except for packets to be forwarded through the NAT engine. Packets on which address translation is performed are reassembled as the NAT engine requires the entire packet.	OPTIONAL
[addrcheck]	<p>Set the level of IP address checks. Choose between:</p> <ul style="list-style-type: none">▪ off No address checking is performed. For advanced users only; in normal circumstances there should always be some kind of address checking.▪ own Minimum level of checking. Only the address configuration on the Speed Touch™ is checked.▪ static Checking of the address configuration of the Speed Touch™ and also of traffic: addresses of incoming packets; this checking is related to constants (e. g. an address may not be entirely composed of one's or zero's).▪ dynamic Besides the address configuration of the Speed Touch™ itself, and besides the checking of traffic on a constants level, additional checking is performed on the IP addresses that are determined by the configuration, more specifically by the network.	OPTIONAL
[mssclamping]	<p>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.</p>	OPTIONAL

EXAMPLE:

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

RELATED COMMANDS:

ip ifconfig

Configure interface parameters.

ip flush

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

Do not perform this command on **Speed Touch™** products that are not equipped with the serial interface (indicated by "Console") on the back panel.

SYNTAX:

```
ip flush
```

EXAMPLE:

```
=>ip aplist
1  eth0      Type:Ethernet HWaddr 00:80:9f:24:ab:cf BRHWaddr ff:ff:ff:ff:ff:ff
    inet addr:10.10.10.147 Bcast:10.10.10.255 Mask:255.0.0.0
    UP RUNNING pat MTU:1500 ReasmMAX:65535 Group:2
    IPRX bytes:26737989 unicastpkts:3689 bcastpkts:450753
    IPTX bytes:383568 unicastpkts:3968 bcastpkts:0 dropkts:0
    HWRX bytes:0 unicastpkts:0 bcastpkts:0
    HWTX bytes:0 unicastpkts:0 bcastpkts:0 dropkts:0
2  cip1      Type:ATM
    inet addr:172.16.0.5 Bcast:172.16.0.255 Mask:255.255.255.0
    UP RUNNING pat MTU:9180 ReasmMAX:65535 Group:0
    IPRX bytes:0 unicastpkts:0 bcastpkts:0
    IPTX bytes:0 unicastpkts:0 bcastpkts:0 dropkts:0
    HWRX bytes:0 unicastpkts:0 bcastpkts:0
    HWTX bytes:0 unicastpkts:0 bcastpkts:0 dropkts:0
=>ip flush
=>ip aplist
2  cip1      Type:ATM
    inet addr:172.16.0.5 Bcast:172.16.0.255 Mask:255.255.255.0
    UP RUNNING pat MTU:9180 ReasmMAX:65535 Group:0
    IPRX bytes:0 unicastpkts:0 bcastpkts:0
    IPTX bytes:0 unicastpkts:0 bcastpkts:0 dropkts:0
    HWRX bytes:0 unicastpkts:0 bcastpkts:0
    HWTX bytes:0 unicastpkts:0 bcastpkts:0 dropkts:0
=>
```

RELATED COMMANDS:

ip load

Load saved or default IP configuration.

ip save

Save current IP configuration.

ip ifconfig

Configure interface parameters.

SYNTAX:

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

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

EXAMPLE:

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

RELATED COMMANDS:

ip config Show/set global IP stack configuration options.

ip iflist

Show all current interfaces.

SYNTAX:

<i>ip iflist</i>

EXAMPLE OUTPUT:

=>ip iflist							
Interface	GRP	MTU	RX	TX	TX-DROP	STATUS	HWADDR
0 loop	1	1500	116	0	0	UP	
1 eth0	2	1500	21075049	1026735	0	UP	00:80:9f:24:ab:cf
2 NewMer	0	1500	0	0	0	UP	00:80:9f:24:ab:cf
5 cip0	0	9180	0	0	0	UP	
=>							

RELATED COMMANDS:

ip ifconfig

Configure interface parameters.

ip load

Load saved (or default) IP configuration.

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

SYNTAX:

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

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

EXAMPLE:

```
=>ip rtlist
  Destination      Source      Gateway      Intf      Mtrc
    10.0.0.0/24    10.0.0.0/24  10.0.0.140   eth0      0
    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   cip0      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   cip0      1
=>ip rtadd dst=10.10.0.0/24 src 10.10.0.0/24 intf eth0
=>ip rtlist
  Destination      Source      Gateway      Intf      Mtrc
    10.10.0.0/24    10.10.0.0/24  10.0.0.140   eth0      0
    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   cip0      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   cip0      1
=>ip save
=>ip flush
=>ip load
=>ip rtlist
  Destination      Source      Gateway      Intf      Mtrc
    10.10.0.0/24    10.10.0.0/24  10.0.0.140   eth0      0
    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   cip0      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   cip0      1
=>
```

RELATED COMMANDS:

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

ip ping

Send ICMP ECHO_REQUEST packets.

SYNTAX:

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

<i>addr</i>	The destination IP address.	REQUIRED
<i>[count]</i>	A number between 1 and 1000000. Represents the number of pings to send.	OPTIONAL
<i>[size]</i>	A number between 1 and 20000 (bytes). Represents the size of the ping packet(s).	OPTIONAL
<i>[interval]</i>	A number between 100 and 1000000 (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 **Speed Touch™** routing table.

SYNTAX:

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

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

EXAMPLE:

=>ip rtlist					
Destination	Source	Gateway	Intf	Mtrc	
10.0.0.0/24	10.0.0.0/24	10.0.0.140	eth0	0	
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	
10.10.0.0/24	10.0.0.2/24	10.0.0.140	eth0	0	
10.0.0.0/24	10.0.0.0/24	10.0.0.140	eth0	0	
127.0.0.1/32	0.0.0.0/0	127.0.0.1	loop	0	
=>					

RELATED COMMANDS:

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

ip rtdelete

Delete a route from the **Speed Touch™** routing table.

SYNTAX:

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

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

EXAMPLE:

=>ip rtlist					
Destination	Source	Gateway	Intf	Mtrc	
10.10.0.0/24	10.0.0.2/24	10.0.0.140	eth0	0	
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	cip0	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=10.0.0.2/24 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	
172.16.0.5/32	0.0.0.0/0	172.16.0.5	cip0	0	
127.0.0.1/32	0.0.0.0/0	127.0.0.1	loop	0	
=>					

RELATED COMMANDS:

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

ip rtlist

Show current **Speed Touch™** routing table.

SYNTAX:

```
ip rtlist
```

EXAMPLE OUTPUT:

```
=>ip rtlist
  Destination      Source           Gateway          Intf      Mtrc
  172.16.0.0/24    172.16.0.0/24    172.16.0.5       eth0      0
  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       cip0      0
  127.0.0.1/32     0.0.0.0/0        127.0.0.1        loop      0
=>
```

RELATED COMMANDS:

ip rtadd

Add a route to the routing table.

ip rtdelete

Remove a route from the routing table.

ip save

Save current IP configuration.

SYNTAX:

```
ip save
```

EXAMPLE:

```
=>ip rtlist
  Destination      Source      Gateway      Intf      Mtrc
    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   cip0      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
=>ip rtadd dst 172.16.0.2/24 src 10.0.0.2/24 intf eth0
=>ip rtlist
  Destination      Source      Gateway      Intf      Mtrc
    172.16.0.0/24   10.0.0.2/24  10.0.0.140   eth0      0
    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   cip0      0
    127.0.0.1/32    0.0.0.0/0   127.0.0.1    loop      0
=>ip save
=>ip flush
=>ip load
=>ip rtlist
  Destination      Source      Gateway      Intf      Mtrc
    172.16.0.0/24   10.0.0.2/24  10.0.0.140   eth0      0
    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   cip0      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
=>
```

RELATED COMMANDS:

ip flush

Flush complete IP configuration.

ip load

Load saved or default IP configuration.

ip sendto

Send UDP packets.

SYNTAX:

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

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

EXAMPLE:

<pre>=>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 =></pre>

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 load
mer save

mer flush

Flush complete MER configuration.

The flush command does not impact previously saved configurations.

SYNTAX:

<i>mer flush</i>

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

=>
```

RELATED COMMANDS:

mer load

Load saved or default MER configuration.

mer save

Save current MER configuration.

mer ifadd

Create a new MER interface.

SYNTAX:

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

[intf]	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
[dest]	The destination for the new MER interface. Typically, an phonebook entry.	OPTIONAL

EXAMPLE:

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

RELATED COMMANDS:

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

mer ifattach

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

SYNTAX:

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

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

EXAMPLE:

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

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

=>
```

RELATED COMMANDS:

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

mer ifconfig

Configure a MER interface.

SYNTAX:

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

<i>intf</i>	The name of the MER interface to configure.	REQUIRED
<i>[dest]</i>	The destination for this interface. Typically a phonebook entry. This parameter needs only to be specified in case of an interface created without specified destination.	OPTIONAL
<i>[qos]</i>	The name of a configured Quality Of Service book entry.	OPTIONAL
<i>[encaps]</i>	The type of encapsulation to be used for this bridge interface. Choose between: ▪ llc/snap ▪ vc mux	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
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 : vcmux  Fcs : off
              Connection State : connected
              RX bytes: 0      frames: 0
              TX bytes: 0      frames: 0      dropframes: 0
=>qosbook list
Name      Ref Type   TX peek  sust      burst    RX peek  sust      burst
              (Kbits) (Kbits) (bytes) (Kbits) (Kbits) (bytes)
default  9  ubr    Linerate 0          0          Linerate 0          0
voice    0  cbr    64        0          0          64        0          0
=>mer ifconfig intf MoreMer qos voice encaps llc/snap retry 15
=>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 : 15  QoS : voice  Encaps : llc/snap  Fcs : off
              Connection State : connected
              RX bytes: 0      frames: 0
              TX bytes: 0      frames: 0      dropframes: 0
=>

```

RELATED COMMANDS:

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

mer ifdelete

Delete a MER interface.

SYNTAX:

<i>mer ifdelete</i>	<i>intf = <ifname></i>	
<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 : vcmux   Fcs : off
              Connection State : connected
              RX bytes: 0          frames: 0
              TX bytes: 0          frames: 0          dropframes: 0
=>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:

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

mer ifdetach

Detach a MER interface.

SYNTAX:

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

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

EXAMPLE:

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

RELATED COMMANDS:

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

mer iflist

Show all or a specified MER interface(s).

SYNTAX:

mer iflist	[intf = <ifname>]	
[intf]	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 : not-connected
=>
```

RELATED COMMANDS:

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

mer load

Load saved (or default) MER configuration.

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

SYNTAX:

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

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

EXAMPLE:

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

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

=>
```

RELATED COMMANDS:

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

mer save

Save current MER configuration.

SYNTAX:

```
mer save
```

EXAMPLE:

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

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

=>
```

RELATED COMMANDS:

mer flush

Flush complete MER configuration.

mer load

Load saved or default MER configuration.

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 load
nat save
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 **Speed Touch™**, i.e. the inherent knowledge of the **Speed Touch™** on this matter.

SYNTAX:

<i>nat applist</i>

EXAMPLE OUTPUT:

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

RELATED COMMANDS:

nat bind

Create a new helper/port binding.

nat bindlist

List current NATP helper/port bindings.

nat unbind

Delete an existing helper/port binding.

nat bind

Create a new helper/port binding.

SYNTAX:

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

EXAMPLE INPUT:

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

RELATED COMMANDS:

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

nat bindlist

List current NAPT helper/port bindings.

SYNTAX:

<i>nat bindlist</i>

EXAMPLE OUTPUT:

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

RELATED COMMANDS:

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

nat create

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

SYNTAX:

nat create	protocol = <IP protocol name or number> inside_addr = <ip-address> [inside_port = <TCP/UDP service name or port number>] outside_addr = <ip-address> [outside_port = <TCP/UDP service name or port number>]	
<i>protocol</i>	The IP protocol name (or number) of the incoming stream.	REQUIRED
<i>inside_addr</i>	The IP address of the local host (intended to receive the incoming traffic) behind the Speed Touch ™'s 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 Speed Touch ™. Use '0' to create a template. Such template will then be valid for any of Speed Touch ™'s 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:Ethernet HWaddr 00:80:9f:24:ab:cf BRHWaddr ff:ff:ff:ff:ff:ff
    inet addr:10.0.0.147      Bcast:10.0.0.255      Mask:255.0.0.0
    UP RUNNING pat MTU:1500  ReasmMAX:65535 Group:2
    IPRX bytes:28935023 unicastpkts:15064    brcastpkts:415221
    IPTX bytes:1088733  unicastpkts:15313    brcastpkts:0      droppkts:0
    HWRX bytes:0        unicastpkts:0        brcastpkts:0      droppkts:0
    HWTX bytes:0        unicastpkts:0        brcastpkts:0      droppkts:0
5  cip0      Type:ATM
    inet addr:172.16.0.5      Bcast:172.16.0.255      Mask:255.255.255.0
    UP RUNNING 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      droppkts: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      droppkts: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 State Control
  1    6  10.0.0.138:80    172.16.0.5:1080    0.0.0.0:0        19    8    9
=>
```

RELATED COMMANDS:

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

nat defserver

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

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

SYNTAX:

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

EXAMPLE INPUT/OUTPUT:

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

nat delete

Delete a static NATP entry.

SYNTAX:

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

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

EXAMPLE:

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

RELATED COMMANDS:

nat create	Create a static NATP entry.
nat disable	Disable NATP on one of the Speed Touch™ IP addresses.
nat enable	Enable NATP on one of the Speed Touch™ IP addresses.
nat list	List NATP connection database.

nat disable

Disable NAPT on a **Speed Touch™** IP address.

SYNTAX:

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

addr One of **Speed Touch™**'s IP addresses one which NAPT is enabled. REQUIRED

EXAMPLE:

```
=>nat list
Indx Prot Inside-addr:Port Outside-addr:Port Foreign-addr:Port Flgs Expir State Control
1      6  10.0.0.138:80      172.16.0.5:1080    0.0.0.0:0          19      8      9
2     17  10.0.0.138:138      10.0.0.140:138    0.0.0.20:138       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
=>nat disable addr 172.16.0.5
=>nat list
Indx Prot Inside-addr:Port Outside-addr:Port Foreign-addr:Port Flgs Expir State Control
1     17  10.0.0.138:138      10.0.0.140:138    10.0.0.20:138       11      20     10
2     17  10.0.0.138:137      10.0.0.140:137    10.0.0.254:137     11      20     10
3     17  10.0.0.138:7938     10.0.0.140:7938   10.0.0.96:4756      11      20     10
4     17  10.0.0.138:513      10.0.0.140:513    10.0.0.109:513      11      20     10
=>
```

RELATED COMMANDS:

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

nat enable

Enable NAPT on a **Speed Touch™** IP address.

SYNTAX:

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

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

EXAMPLE:

```
=>ip aplist
1  eth0      Type:Ethernet HWaddr 00:80:9f:24:ab:cf BRHWaddr ff:ff:ff:ff:ff:ff
   inet addr:10.0.0.140      Mask:255.255.255.0
   UP RUNNING pat MTU:1500  ReasmMAX:65535 Group:2
   IPRX bytes:29264560 unicastpkts:15069  brcastpkts:419164
   IPTX bytes:1089173  unicastpkts:15316  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:172.16.0.255      Mask:255.255.255.0
   UP RUNNING 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
=>nat enable addr 172.16.0.5 type pat
=>ip aplist
1  eth0      Type:Ethernet HWaddr 00:80:9f:24:ab:cf BRHWaddr ff:ff:ff:ff:ff:ff
   inet addr:10.0.0.140      Bcast:10.0.0.255      Mask:255.255.255.0
   UP RUNNING pat MTU:1500  ReasmMAX:65535 Group:2
   IPRX bytes:29313523 unicastpkts:15074  brcastpkts:419681
   IPTX bytes:1090231  unicastpkts:15321  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: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
=>
```

RELATED COMMANDS:

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

nat flush

Flush complete NAPT configuration.

The flush command does not impact previously saved configurations.

SYNTAX:

nat flush

EXAMPLE:

```
=>nat list
Indx Prot Inside-addr:Port Outside-addr:Port Foreign-addr:Port Flgs Expir State Control
  1  17    10.0.0.138:135    10.0.0.140:135    10.0.0.155:1034  11    20    10
  2  17    10.0.0.138:137    10.0.0.140:137    10.0.0.208:137  11    20    10
  3  17    10.0.0.138:7938    10.0.0.140:7938    10.0.0.96:1282  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    10.0.0.96:1282  11    20    10
=>nat save
=>nat flush
=>nat list
=>nat load
=>nat list
Indx Prot Inside-addr:Port Outside-addr:Port Foreign-addr:Port Flgs Expir State Control
  1  17    10.0.0.138:135    10.0.0.140:135    10.0.0.155:1034  11    20    10
  2  17    10.0.0.138:137    10.0.0.140:137    10.0.0.208:137  11    20    10
  3  17    10.0.0.138:7938    10.0.0.140:7938    10.0.0.96:1282  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    10.0.0.96:1282  11    20    10
=>
```

RELATED COMMANDS:

nat load

Load saved or default NAPT configuration.

nat save

Save current NAPT configuration.

nat list

Show NAPT connection database.

SYNTAX:

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

<i>[addr]</i>	The Speed Touch™ 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 State Control
  1   6  10.0.0.138:80      172.16.0.5:1080    0.0.0.0:0          19    8    9
  2  17  10.0.0.138:135      10.0.0.140:135    10.0.0.155:1034    11    20   10
  3  17  10.0.0.138:138      10.0.0.140:138    10.0.0.54:138      11    20   10
  4  17  10.0.0.138:137      10.0.0.140:137    10.0.0.208:137     11    20   10
  5  17  10.0.0.138:7938     10.0.0.140:7938    10.0.0.96:1365     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     10.0.0.96:1365     11    20   10
=>
```

RELATED COMMANDS:

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

nat load

Load saved (or default) NAPT configuration.
Execute **nat flush** prior to **nat load**.

SYNTAX:

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

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

EXAMPLE:

=>nat list									
Indx	Prot	Inside-addr:Port	Outside-addr:Port	Foreign-addr:Port	Flgs	Expir	State	Control	
1	17	10.0.0.138:138	10.0.0.140:138	10.0.0.204:138	11	20	10		
2	17	10.0.0.138:137	10.0.0.140:137	10.0.0.208:137	11	20	10		
=>nat list									
Indx	Prot	Inside-addr:Port	Outside-addr:Port	Foreign-addr:Port	Flgs	Expir	State	Control	
1	17	10.0.0.138:138	10.0.0.140:138	10.0.0.204:138	11	20	10		
2	17	10.0.0.138:137	10.0.0.140:137	10.0.0.208:137	11	20	10		
=>nat load defaults									
=>nat list									
Indx	Prot	Inside-addr:Port	Outside-addr:Port	Foreign-addr:Port	Flgs	Expir	State	Control	
1	17	10.0.0.138:138	10.0.0.140:138	10.0.0.204:138	11	20	10		
2	17	10.0.0.138:137	10.0.0.140:137	10.0.0.208:137	11	20	10		
3	17	10.0.0.138:513	10.0.0.140:513	10.0.0.109:513	11	20	10		
=>									

RELATED COMMANDS:

nat flush	Flush complete NAPT configuration.
nat save	Save current NAPT configuration.

nat save

Save current NAPT configuration.

SYNTAX:

```
nat save
```

EXAMPLE:

```
=> nat create 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 State Control
  1   6  10.0.0.138:80      172.16.0.5:1080    0.0.0.0:0          19   8     9
  2  17  10.0.0.138:138      10.0.0.140:138    10.0.0.75:138      11  20    10
  3  17  10.0.0.138:137      10.0.0.140:137    10.0.0.75:137      11  20    10
  4  17  10.0.0.138:7938     10.0.0.140:7938    10.0.0.96:1698      11  20    10
=>nat save
=>nat flush
=>nat load
addr option required
addr option required
=>nat list
Indx Prot Inside-addr:Port Outside-addr:Port Foreign-addr:Port Flgs Expir State Control
  1   6  10.0.0.138:80      172.16.0.5:1080    0.0.0.0:0          19   8     9
  2  17  10.0.0.138:137      10.0.0.140:137    10.0.0.53:137      11  20    10
=>
```

RELATED COMMANDS:

nat flush

Flush complete NAPT configuration.

nat load

Load saved or default NAPT configuration.

nat unbind

Delete an existing helper/port binding.

SYNTAX:

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

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

EXAMPLE:

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

RELATED COMMANDS:

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

12 Phonebook Commands

phonebook (to access the Phonebook level)

phonebook add

phonebook autolist

phonebook delete

phonebook flush

phonebook list

phonebook load

phonebook save

phonebook add

Add a phonebook entry.

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

SYNTAX:

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

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

EXAMPLE:

```
=>phonebook list
Name      Type    Use    Address
Br1       bridge 1      8.35
Br2       bridge 1      8.36
Br3       bridge 1      8.37
Br4       bridge 1      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     1      8.80
CIPPVC2   cip     1      8.81
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
Br1       bridge 1      8.35
Br2       bridge 1      8.36
Br3       bridge 1      8.37
Br4       bridge 1      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     1      8.80
CIPPVC2   cip     1      8.81
CIPPVC3   cip     1      8.82
CIPPVC4   cip     1      8.83
Alcatel  ppp     0      8.68
=>
```

RELATED COMMANDS:**phonebook delete**

Remove a phonebook entry.

phonebook list

Show current phonebook.

phonebook autolist

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

SYNTAX:

<i>phonebook autolist</i>

EXAMPLE INPUT/OUTPUT:

<pre>=>phonebook autolist 8.35 =></pre>

RELATED COMMANDS:

phonebook list Show current phonebook.

phonebook delete

Remove an unused phonebook entry.

SYNTAX:

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

EXAMPLE:

=>phonebook list			
Name	Type	Use	Address
Br1	bridge	1	8.35
Br2	bridge	1	8.36
Br3	bridge	1	8.37
Br4	bridge	1	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
DHCP_Spoof01	ppp	0	8.68
=>phonebook delete name DHCP_Spoof01			
=>phonebook list			
Name	Type	Use	Address
Br1	bridge	1	8.35
Br2	bridge	1	8.36
Br3	bridge	1	8.37
Br4	bridge	1	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
=>			

RELATED COMMANDS:

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

phonebook flush

Flush complete phonebook.

The flush command does not impact previously saved configurations.

SYNTAX:

phonebook flush

EXAMPLE:

```
=>phonebook list
Name      Type    Use  Address
Br1       bridge 0     8.35
Br2       bridge 0     8.36
Br3       bridge 0     8.37
RELAY_PPP3 ppp     0     8.50
RELAY_PPP4 ppp     0     8.51
PPP1      ppp     0     8.64
PPP2      ppp     0     8.65
PPP3      ppp     0     8.66
DHCP_SPOOF ppp     0     8.67
=>phonebook save
=>phonebook flush
=>phonebook list
Name      Type    Use  Address
=>phonebook load
=>phonebook list
Name      Type    Use  Address
Br1       bridge 0     8.35
Br2       bridge 0     8.36
Br3       bridge 0     8.37
RELAY_PPP3 ppp     0     8.50
RELAY_PPP4 ppp     0     8.51
PPP1      ppp     0     8.64
PPP2      ppp     0     8.65
PPP3      ppp     0     8.66
DHCP_SPOOF ppp     0     8.67
=>
```

RELATED COMMANDS:

phonebook load

Load saved or default phonebook.

phonebook save

Save current phonebook.

phonebook list

Show current phonebook.

SYNTAX:

phonebook list	[opt = <{long}>]	
[opt]	Select output format. Only applicable for showing SVC entries. (depending on the Speed Touch ™ release).	OPTIONAL

EXAMPLE INPUT/OUTPUT:

=>phonebook list			
Name	Type	Use	Address
Br1	bridge	1	8.35
Br2	bridge	1	8.36
Br3	bridge	1	8.37
Br4	bridge	1	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
<u>=>phonebook add name DHCP Spoof01 addr 8.68 type ppp</u>			
=>phonebook list			
Name	Type	Use	Address
Br1	bridge	1	8.35
Br2	bridge	1	8.36
Br3	bridge	1	8.37
Br4	bridge	1	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
DHCP Spoof01	ppp	0	8.68
=>			

RELATED COMMANDS:

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

phonebook load

Load saved (or default) phonebook.

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

SYNTAX:

phonebook load [defaults <yes no>]

[defaults]

Load factory defaults (yes) or saved configuration (no).

OPTIONAL

Not specifying this parameter loads the saved configuration

EXAMPLE:

```
=>phonebook list
Name      Type    Use  Address
Br1       bridge 1    8.35
Br2       bridge 1    8.36
Br3       bridge 1    8.37
Br4       bridge 1    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
DHCP_SPOOF ppp    1    8.67
CIPPVC1   cip    1    8.80
CIPPVC2   cip    1    8.81
CIPPVC3   cip    1    8.82
CIPPVC4   cip    1    8.83
DHCP_Spoof01 ppp    0    8.68
=>phonebook save
=>phonebook flush
=>phonebook list
Name      Type    Use  Address
=>phonebook load defaults=no
=>phonebook list
Name      Type    Use  Address
Br1       bridge 0    8.35
Br2       bridge 0    8.36
Br3       bridge 0    8.37
Br4       bridge 0    8.38
RELAY_PPP1 ppp    0    8.48
RELAY_PPP2 ppp    0    8.49
RELAY_PPP3 ppp    0    8.50
RELAY_PPP4 ppp    0    8.51
DHCP_SPOOF ppp    0    8.67
CIPPVC1   cip    0    8.80
CIPPVC2   cip    0    8.81
CIPPVC3   cip    0    8.82
CIPPVC4   cip    0    8.83
DHCP_Spoof01 ppp    0    8.68
=>
```

RELATED COMMANDS:

phonebook flush

Flush complete phonebook.

phonebook save

Save current phonebook.

phonebook save

Save current phonebook.

SYNTAX:

```
phonebook save
```

EXAMPLE:

```
=>phonebook add name DHCP_Spoof01 addr 8.68 type ppp
=>phonebook list
Name      Type    Use  Address
Br1       bridge 1    8.35
Br2       bridge 1    8.36
Br3       bridge 1    8.37
Br4       bridge 1    8.38
PPP1      ppp     1    8.64
PPP2      ppp     1    8.65
PPP3      ppp     1    8.66
DHCP_SPOOF ppp     1    8.67
CIPPVC1   cip     1    8.80
CIPPVC2   cip     1    8.81
CIPPVC3   cip     1    8.82
CIPPVC4   cip     1    8.83
DHCP_Spoof01 ppp     0    8.68
=>phonebook save
=>phonebook flush
=>phonebook list
Name Type Use Address
=>phonebook load
=>phonebook list
Name      Type    Use  Address
Br1       bridge 0    8.35
Br2       bridge 0    8.36
Br3       bridge 0    8.37
Br4       bridge 0    8.38
PPP1      ppp     0    8.64
PPP2      ppp     0    8.65
PPP3      ppp     0    8.66
DHCP_SPOOF ppp     0    8.67
CIPPVC1   cip     0    8.80
CIPPVC2   cip     0    8.81
CIPPVC3   cip     0    8.82
CIPPVC4   cip     0    8.83
DHCP_Spoof01 ppp     0    8.68
=>
```

RELATED COMMANDS:

phonebook flush

Flush complete phonebook.

phonebook load

Load saved or default phonebook.

13 PPP Commands

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

ppp flush

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

SYNTAX:

ppp flush

EXAMPLE:

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

RELATED COMMANDS:

ppp load

Load saved or default PPP configuration.

ppp save

Save current PPP configuration.

ppp ifadd

Create a new PPP interface.

SYNTAX:

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

[<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: ▪ vcmux ▪ llc/snap	OPTIONAL
[<i>speed</i>]	A number between 4800 and 10000000 (bits per second). Represents the speed of the peer-to-peer connection. Use for backward compatibility. Use Quality Of Service instead.	OPTIONAL

EXAMPLE:

```
=>ppp iflist
PPP1: dest : PPP1
.....
IPCP: state = initial retransm = 0 term. reason =
=>=>ppp ifadd intf PPPLink dest PPP3 encaps vcmux
=>ppp iflist
PPP1: dest : PPP1
.....
IPCP: state = initial retransm = 0 term. reason =
PPPLink: dest : PPP3
  Retry : 10 QoS default encaps VC-MUX
  flags = echo magic accomp restart mru addr savepwd
  mru = 1500
  user name = password =
  admin state = down oper state = down link state = not-connected
  LCP : state = initial retransm = 10 term. reason =
  IPCP: state = initial retransm = 0 term. reason =
=>
```

RELATED COMMANDS:

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

ppp ifattach

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

SYNTAX:

ppp ifattach	intf = <ifname>
---------------------	------------------------------

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

EXAMPLE:

```
=>ppp iflist intf PPPLink
PPPLink: dest : PPP3
  Retry : 10  QoS default  encaps VC-MUX
  flags = echo magic accomp restart mru addr savepwd
  mru = 1500
  user name =      password =
  admin state = down  oper state = down  link state = not-connected
  LCP : state = initial  retransm = 10  term. reason =
  IPCP: state = initial  retransm = 0  term. reason =
=>ppp ifattach intf PPPLink
=>ppp iflist intf PPPLink
PPPLink: dest : PPP3
  Retry : 10  QoS default  encaps VC-MUX
  flags = echo magic accomp restart mru addr savepwd
  mru = 1500
  user name =      password =
  admin state = up   oper state = down  link state = connected
  LCP : state = reqsent  retransm = 7  term. reason =
  IPCP: state = initial  retransm = 0  term. reason =
=>
```

RELATED COMMANDS:

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

ppp ifconfig

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

SYNTAX:

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

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

[proto]	<p>The encapsulation method for the PPP frames, i.e. the applicable packet service for the connection.</p> <p>Select:</p> <ul style="list-style-type: none">▪ pppoa For a Routed PPPoA (PPP & IP Routing) connections.▪ pppe For a Routed PPPoE connection. <p>Per default the PPPoA protocol applies.</p>	OPTIONAL
[acname]	<p>The Access Concentrator name for a Routed PPPoE connection.</p> <p>This parameter is applicable only for Routed PPPoE PPP interfaces (proto=pppe).</p> <p>Execute the ppp ifscan command to see the names of available access concentrators, if any.</p>	OPTIONAL
[servicename]	<p>The Service Name for a Routed PPPoE connection.</p> <p>This parameter is applicable only for Routed PPPoE PPP interfaces (proto=pppe).</p> <p>Execute the ppp ifscan command to see the available service names, if any.</p>	OPTIONAL
[encaps]	<p>The type of encapsulation to be used for this PPP interface.</p> <p>Choose between:</p> <ul style="list-style-type: none">▪ vcmux (default) Standard encapsulation method for PPPoA (ppp) frames.▪ llc Standard encapsulation method for ETHoA (bridge) frames.	OPTIONAL
[pcomp]	<p>Try (on) or do not try (off) to negotiate PPP protocol compression (LCP PCOMP).</p> <p>Per default the negotiation is disabled (off).</p>	OPTIONAL
[accomp]	<p>Try (on), do never try (off) or negotiate (negotiate) to negotiate PPP address & control field compression (LCP ACCOMP).</p> <p>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.</p> <p>Therefore by default this parameter is enabled (on).</p> <p>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.</p>	OPTIONAL
[trace]	<p>Enable (on) or disable (off) verbose console logging.</p> <p>By default tracing is disabled (off).</p>	OPTIONAL
[pap]	<p>Force PAP based authentication (on) or use CHAP based authentication, if available (off).</p> <p>For security reasons PAP negotiation is disabled (off) per default.</p>	OPTIONAL
[restart]	<p>Automatically restart the connection when LCP link goes down (on) or do not restart automatically (off).</p> <p>By default restart is disabled (off).</p>	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 Speed Touch™ 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 PPP client (if it allows to) to accept this IP address as the Speed Touch™ PPP session IP address. If not specified, the Speed Touch™ 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 client (if it allows to) to accept this IP address as its PPP session IP address. If not specified, the Speed Touch™ will accept any IP address. Typically the remote IP address parameter is not specified.	OPTIONAL
<i>[savepwd]</i>	Save password (on), if supplied, or do not save the password (off). Per default the saving of the password is disabled.	OPTIONAL
<i>[demanddial]</i>	Enable (on) or disable (off) the dial-on-demand feature.	OPTIONAL
<i>[primdns]</i>	The IP address of the primary DNS server. In case a primary DNS server is specified the Speed Touch™ will negotiate this IP address with the remote side. If not specified, the Speed Touch™ 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 Speed Touch™ will negotiate this IP address with the remote side. If not specified, the Speed Touch™ will accept any IP address.	OPTIONAL
<i>[idle]</i>	A number between 1 and 1000000 (seconds). Represents after how many seconds an idle link goes down.	OPTIONAL

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

EXAMPLE:

```
=>ppp iflist intf PPPLink
PPPLink: dest : PPP3
  Retry : 10  QoS default  encaps VC-MUX
  flags = echo magic accomp restart mru addr savepwd
  mru = 1500
  user name = password =
  admin state = down  oper state = down  link state = not-connected
  LCP : state = initial  retransm = 10  term. reason =
  IPCP: state = initial  retransm = 0  term. reason =
=>ppp ifconfig intf PPPLink password mdp996 savepwd off
=>ppp iflist intf PPPLink
PPPLink: dest : PPP3
  Retry : 10  QoS default  encaps VC-MUX
  flags = echo magic accomp restart mru addr
  mru = 1500
  user name = password = *****
  admin state = down  oper state = down  link state = not-connected
  LCP : state = initial  retransm = 10  term. reason =
  IPCP: state = initial  retransm = 0  term. reason =
=>
```

RELATED COMMANDS:

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

ppp ifdelete

Delete a PPP interface.

SYNTAX:

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

intf The name of the PPP interface to delete.

EXAMPLE:

```
=>ppp iflist
PPP1: dest : PPP1
  Retry : 10  QoS default  encaps VC-MUX
  flags = echo magic accomp mru addr route savepwd
  trans addr = pat      mru = 1500
  route =                0.0.0.0/0 -                0.0.0.0/0 (metric 0)
  user name = guest      password = *****
  admin state = down      oper state = down      link state = not-connected
  LCP : state = initial  retransm = 10  term. reason =
  IPCP: state = initial  retransm = 0   term. reason =
PPPLink: dest : PPP3
  Retry : 10  QoS default  encaps VC-MUX
  flags = echo magic accomp restart mru addr savepwd
  mru = 1500
  user name =          password = *****
  admin state = up      oper state = down      link state = connected
  LCP : state = reqsent  retransm = 1   term. reason =
  IPCP: state = initial  retransm = 0   term. reason =
=>ppp ifdelete intf PPPLink
=>ppp iflist
PPP1: dest : PPP1
  Retry : 10  QoS default  encaps VC-MUX
  flags = echo magic accomp mru addr route savepwd
  trans addr = pat      mru = 1500
  route =                0.0.0.0/0 -                0.0.0.0/0 (metric 0)
  user name = guest      password = *****
  admin state = down      oper state = down      link state = not-connected
  LCP : state = initial  retransm = 10  term. reason =
  IPCP: state = initial  retransm = 0   term. reason =
=>
```

RELATED COMMANDS:

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

ppp ifdetach

Detach a PPP interface.

SYNTAX:

ppp ifdetach	intf = <ifname>
---------------------	------------------------------

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

EXAMPLE:

```
=>ppp iflist intf PPPLink
PPPLink: dest : PPP3
  Retry : 10  QoS default  encaps VC-MUX
  flags = echo magic accomp restart mru addr savepwd
  mru = 1500
  user name =      password =
  admin state = up      oper state = down      link state = connected
  LCP : state = reqsent  retransm = 4  term. reason =
  IPCP: state = initial  retransm = 0  term. reason =
=>ppp ifdetach intf PPPLink
=>ppp iflist intf PPPLink
PPPLink: dest : PPP3
  Retry : 10  QoS default  encaps VC-MUX
  flags = echo magic accomp restart mru addr savepwd
  mru = 1500
  user name =      password =
  admin state = down  oper state = down      link state = not-connected
  LCP : state = closed  retransm = 10  term. reason = User request
  IPCP: state = initial  retransm = 0  term. reason =
=>
```

RELATED COMMANDS:

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

ppp iflist

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

SYNTAX:

ppp iflist	[intf = <ifname>]	
<i>intf <ifname></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
  flags = echo magic accomp mru addr route savepwd
  trans addr = pat      mru = 1500
  route =      10.0.0.130/0 -      0.0.0.0/0 (metric 1)
  user name = alcaboy@Internetprovider  password = *****
  admin state = up    oper state = up    link state = connected
  LCP : state = opened  retransm = 0  term. reason =
  IPCP: state = opened  retransm = 10  term. reason =
DHCP_SPOOF: dest : DHCP_SPOOF
  Retry : 10  QoS default  encaps VC-MUX
  flags = echo magic accomp mru addr route savepwd
  mru = 1500
  route =      0.0.0.0/32 -      0.0.0.0/0 (metric 0)
  user name = guest  password = *****
  admin state = down  oper state = down  link state = not-connected
  LCP : state = initial  retransm = 10  term. reason =
  IPCP: state = initial  retransm = 0  term. reason =
=>ppp iflist intf DHCP_SPOOF
DHCP_SPOOF: dest : DHCP_SPOOF
  Retry : 10  QoS default  encaps VC-MUX
  flags = echo magic accomp mru addr route savepwd
  mru = 1500
  route =      0.0.0.0/32 -      0.0.0.0/0 (metric 0)
  user name = guest  password = *****
  admin state = down  oper state = down  link state = not-connected
  LCP : state = initial  retransm = 10  term. reason =
  IPCP: state = initial  retransm = 0  term. reason =
=>
```

RELATED COMMANDS:

ppp ifadd	Create a PPP interface.
ppp ifattach	Attach a PPP interface.
ppp ifconfig	Configure a PPP interface.
ppp ifdelete	Delete a PPP interface.
ppp ifdetach	Detach a PPP interface.

ppp ifscan

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

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

SYNTAX:

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

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

EXAMPLE:

```
=>ppp iflist
My Connection : dest : My Connection
  Retry : 10  QoS default  encaps LLC
  flags = echo magic accomp mru addr route PPPoE
  trans addr = pat      mru = 1500
  route =          0.0.0.0/0 -          0.0.0.0/0 (metric 1)
  user name =      password =
  admin state = down  oper state = down  link state = not-connected
  LCP : state = initial  retransm = 9  term. reason =
  IPCP: state = initial  retransm = 10  term. reason =
  acname : , service : .
=>ppp ifscan intf=My_Connection time=30
  Service Name          Access Concentrator
Done !
=>
```

RELATED COMMANDS:

ppp ifconfig Configure a PPP interface.

ppp load

Load saved (or default) PPP configuration.

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

SYNTAX:

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

EXAMPLE:

```
=>ppp iflist
PPP1: dest : PPP1
  Retry : 10  QoS default  encaps VC-MUX
  flags = echo magic accomp mru addr route savepwd
  trans addr = pat      mru = 1500
  route =      0.0.0.0/0 -      0.0.0.0/0 (metric 0)
  user name = guest  password = *****
  admin state = down  oper state = down  link state = not-connected
  LCP : state = initial  retransm = 10  term. reason =
  IPCP: state = initial  retransm = 0  term. reason =
=>ppp save
=>ppp flush
=>ppp iflist
=>ppp load defaults=no
=>ppp iflist
PPP1: dest : PPP1
  Retry : 10  QoS default  encaps VC-MUX
  flags = echo magic accomp mru addr route savepwd
  trans addr = pat      mru = 1500
  route =      0.0.0.0/0 -      0.0.0.0/0 (metric 0)
  user name = guest  password = *****
  admin state = down  oper state = down  link state = not-connected
  LCP : state = initial  retransm = 10  term. reason =
  IPCP: state = initial  retransm = 0  term. reason =
=>
```

RELATED COMMANDS:

ppp flush

Flush complete PPP configuration.

ppp save

Save current PPP configuration.

ppp rtadd

Automatically add a route configuration to the routing table in case the specified PPP interface link comes up.

This route configuration will determine which local hosts are allowed to use this link and/or which remote destinations should be or should not be reachable.

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

SYNTAX:

ppp rtadd	intf = <ifname> dst = <ip-address> [dstmsk = <ip-mask(dotted or cidr)>] [src = <ip-address>] [srcmsk = <ip-mask(dotted or cidr)>] [metric <number{0-100}>]	
<i>intf</i>	The name of the PPP interface.	REQUIRED
<i>dst</i>	The destination IP address specification for the route to be added when the link comes up.	REQUIRED
<i>[dstmsk]</i>	The destination IP mask. Depending on the destination netmask: <ul style="list-style-type: none"> Any remote destination is reachable, i.e. the PPP connection acts as default route (dstmsk=0) Only the remote (sub)net is reachable (dstmsk=1) The actual destination mask will be the default netmask applicable for destination IP address Only the single remote host is reachable (dstmsk=32) 	OPTIONAL
<i>[src]</i>	The source IP address specification for the route to be added when the link comes up.	OPTIONAL
<i>[srcmsk]</i>	The source IP mask. Depending on the source netmask: <ul style="list-style-type: none"> Everybody is allowed to use this PPP connection (dstmsk=0) Only members of the same subnet as the host which opened the PPP connection are allowed to use the PPP connection (dstmsk=1) The actual destination mask will be the netmask applicable for the IP address of the host which opened the PPP connection. Only the host which opened the PPP connection is allowed to use the PPP connection. (dstmsk=32) 	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 intf PPPLink
PPPLink: dest : PPP3
  Retry : 10  QoS default  encaps VC-MUX
  flags = echo magic accomp restart mru addr _____ savepwd
  mru = 1500
  user name =      password =
  admin state = down   oper state = down   link state = not-connected
  LCP : state = initial  retransm = 10  term. reason =
  IPCP: state = initial  retransm = 0   term. reason =
=>ppp rtadd intf PPPLink dst 172.16.0.5 dstmsk 24 src 10.0.0.2 srcmsk 24
=>ppp iflist intf PPPLink
PPPLink: dest : PPP3
  Retry : 10  QoS default  encaps VC-MUX
  flags = echo magic accomp restart mru addr route savepwd
  mru = 1500
  route =          10.0.0.2/24 -          172.16.0.5/24 (metric 1)
  user name =      password =
  admin state = down   oper state = down   link state = not-connected
  LCP : state = initial  retransm = 10  term. reason =
  IPCP: state = initial  retransm = 0   term. reason =
=>
```

RELATED COMMANDS:

ppp rtdelete

Delete the route specification for an upcoming PPP link.

ppp rtdelete

Delete the route specification for a PPP link.

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

SYNTAX:

ppp rtdelete	intf = <ifname>
---------------------	------------------------------

<i>intf</i>	The PPP interface name for which to delete the route settings.	REQUIRED
-------------	--	----------

EXAMPLE:

```
=>ppp iflist intf PPPLink
PPPLink: dest : PPP3
  Retry : 10  QoS default  encaps VC-MUX
  flags = echo magic accomp restart mru addr route savepwd
  mru = 1500
  route = 10.0.0.2/24 - 172.16.0.5/24 (metric 1)
  user name = password =
  admin state = down oper state = down link state = not-connected
  LCP : state = initial retransm = 10 term. reason =
  IPCP: state = initial retransm = 0 term. reason =
=>ppp rtdelete intf PPPLink
=>ppp iflist intf PPPLink
PPPLink: dest : PPP3
  Retry : 10  QoS default  encaps VC-MUX
  flags = echo magic accomp restart mru addr _____ savepwd
  mru = 1500
  user name = password =
  admin state = down oper state = down link state = not-connected
  LCP : state = initial retransm = 10 term. reason =
  IPCP: state = initial retransm = 0 term. reason =
=>
```

RELATED COMMANDS:

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

ppp save

Save current PPP configuration.

SYNTAX:

ppp save

EXAMPLE:

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

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

=>
```

RELATED COMMANDS:

ppp flush

Flush complete PPP configuration.

ppp load

Load saved or default PPP configuration.

14 PPTP Commands

pptp (to access the PPTP level)
pptp flush
pptp ifadd
pptp list
pptp load
pptp profadd
pptp profdelete
pptp proflist
pptp save

pptp flush

Flush complete PPTP configuration.

The flush command does not impact previously saved configurations.

SYNTAX:

<i>pptp flush</i>

EXAMPLE:

```
=>pptp ifadd dest PPP3 rate 6000 encaps vcmux ac always
=>pptp list
=>pptp proflist
Profile           QoS           Encaps   AC
default           default        vcmux    never
PPP3              PPP3          vcmux    always
=>pptp flush
=>pptp proflist
=>pptp load
=>pptp proflist
Profile           QoS           Encaps   AC
default           default        vcmux    never
=>
```

RELATED COMMANDS:

pptp load

Load saved or default PPTP configuration.

pptp save

Save current PPTP configuration.

pptp ifadd

Add a (PPTP profile for an existing) pptp interface.

If available it is recommended to use the **pptp profadd** command instead.

SYNTAX:

pptp ifadd	dest = <string> [rate = <number{10–10000}>] [encaps = <{vcmux nlpid}>] [ac <{never always keep}>]	
<i>dest</i>	The destination for the relayed PPTP tunnel. Typically a phonebook name.	REQUIRED
<i>[rate]</i>	A number between 10 and 10000 (bits per second). Indicates the (maximum) transmission speed on the WAN link.	OPTIONAL
<i>[encaps]</i>	The type of encapsulation to be used for the relayed PPPoA interface. Choose between: <ul style="list-style-type: none"> ▪ vcmux (default) ▪ nlpid 	OPTIONAL
<i>[ac]</i>	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 ifadd dest=PPTP rate=6000 encaps=vcmux
=>pptp proflist
Profile      QoS      Encaps  AC
default      default  vcmux   never
PPTP3        PPTP3    vcmux   never
=>
```

RELATED COMMANDS:

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

pptp list

Show current PPTP configuration.

SYNTAX:

<i>pptp list</i>

EXAMPLE INPUT/OUTPUT:

=>pptp list						
Dialstr	Destination	QoS	Encaps	AC	State	User
DIA	IALUP_PPP3	default	vcmux	never	CONNECTED	(10.0.0.2)
=>						

pptp load

Load saved (or default) PTP configuration.

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

SYNTAX:

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

EXAMPLE:

```
=>pptp proflist
Profile      QoS      Encaps      AC
default      default    vcmux       never
=>qosbook list
Name      Ref Type      TX peek    sust      burst      RX peek    sust      burst
           (Kbits) (Kbits) (bytes) (Kbits) (Kbits) (bytes)
default 24  ubr      Linerate 0          0          Linerate 0          0
voice  0  cbr      64        0          0          64        0          0
PPP3    0  ubr      6144      0          0          Linerate 0          0
=>pptp profadd name PPTPLink qos PPP3 encaps nlpid ac always
=>pptp proflist
Profile      QoS      Encaps      AC
default      default    vcmux       never
PPTPLink     PPP3      nlpid       always
=>pptp save
=>pptp flush
=>pptp load
=>pptp proflist
Profile      QoS      Encaps      AC
default      default    vcmux       never
PPTPLink     PPP3      nlpid       always
=>
```

RELATED COMMANDS:

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

pptp profadd

Define a new PPTP profile.

If available use this command instead of **pptp ifadd**.

SYNTAX:

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

EXAMPLE:

```
=>pptp proflist
Profile           QoS           Encaps   AC
default           default       vcmux    never

=>qosbook list
Name      Ref Type      TX peek  sust      burst      RX peek  sust      burst
              (Kbits) (Kbits)  (bytes)  (Kbits) (Kbits)  (bytes)
default  24  ubr      Linerate 0         0         Linerate 0         0
PPP3     0  ubr      6144     0         0         Linerate 0         0

=>pptp profadd name PPTPLink qos PPP3 encaps nlpid ac always
=>pptp proflist
Profile           QoS           Encaps   AC
default           default       vcmux    never
PPTPLink          PPP3         nlpid    always

=>
```

RELATED COMMANDS:

pptp ifadd	Add a (PPTP profile for an existing) pptp interface.
pptp profdelete	Delete a PPTP profile.
pptp proflist	Show current PPTP profiles.

pptp profdelete

Delete a PTP profile.

SYNTAX:

<i>pptp profdelete</i>	<i>name <string></i>	
<i>name</i>	The name for the PTP profile.	REQUIRED

EXAMPLE:

=>pptp proflist			
Profile	QoS	Encaps	AC
default	default	vcmux	never
PPTPLink	PPP3	nlpid	always
=>pptp profdelete name PPTPLink			
=>pptp proflist			
Profile	QoS	Encaps	AC
default	default	vcmux	never
=>			

RELATED COMMANDS:

pptp ifadd	Add a (PTP profile for an existing) pptp interface.
pptp profadd	Define a new PTP profile.
pptp proflist	Show current PTP profiles.

pptp proflist

Show all current PPTP profiles.

pptp proflist

EXAMPLE:

=>pptp proflist			
Profile	QoS	Encaps	AC
default	default	vcmux	never
PPTPLink	PPP3	nlpid	always
=>			

RELATED COMMANDS:

pptp ifadd	Add a (PPTP profile for an existing) pptp interface.
pptp profadd	Define a new PPTP profile.
pptp profdelete	Delete a PPTP profile.

pptp save

Save current PTP configuration.

SYNTAX:

pptp save

EXAMPLE:

```
=>pptp proflist
Profile      QoS      Encaps  AC
default     default  vcmux   never
=>pptp profadd name PPTPLink qos PPP3 encaps nlpid ac always
=>pptp proflist
Profile      QoS      Encaps  AC
default     default  vcmux   never
PPTPLink    PPP3     nlpid   always
=>pptp save
=>pptp flush
=>pptp load
=>pptp proflist
Profile      QoS      Encaps  AC
default     default  vcmux   never
PPTPLink    PPP3     nlpid   always
=>
```

RELATED COMMANDS:

pptp flush

Flush complete PTP configuration.

pptp load

Load saved or default PTP configuration.

15 QoSBook Commands

qosbook (to access the QoSbook level)

qosbook add

qosbook delete

qosbook flush

qosbook list

qosbook load

qosbook save

qosbook add

Add a Quality of Service book entry.

SYNTAX:

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

<i>name</i>	The name for the new QoS entry.	REQUIRED
<i>class</i>	The ATM service category. Choose between: <ul style="list-style-type: none"> ▪ ubr: unspecified bit rate ▪ cbr: constant bit rate ▪ vbr-nrt: variable bit rate - non real time 	REQUIRED
<i>[tx_peekrate]</i>	A number between 0 and 2147483 (Kilobits per second). Indicates the peek rate in the transmit (upstream) direction. Use tx_peekrate=0 to indicate Linerate.	OPTIONAL
<i>[tx_sustrate]</i>	A number between 0 and 2147483 (Kilobits per second). Indicates the sustainable rate in the transmit (upstream) direction. Use tx_susrate=0 to indicate Linerate. Only applicable in case class=vbr.	OPTIONAL
<i>[tx_maxburst]</i>	A number between 0 and 2147483600 (Kilobits per second). Indicates the maximum burst size in the transmit (upstream) direction. Use tx_maxburst=0 to indicate Linerate. Only applicable in case class=vbr.	OPTIONAL
<i>[rx_peekrate]</i>	A number between 0 and 2147483 (Kilobits per second). Indicates the peek rate in the receive (downstream) direction. Use rx_peekrate=0 to indicate Linerate. Only applicable in an SVC environment.	OPTIONAL
<i>[rx_sustrate]</i>	A number between 0 and 2147483 (Kilobits per second). Indicates the sustainable rate in the receive (downstream) direction. Use rx_susrate=0 to indicate Linerate. Only applicable in an SVC environment.	OPTIONAL
<i>[rx_maxburst]</i>	A number between 0 and 2147483600 (Kilobits per second). Indicates the maximum burst size in the receive (downstream) direction. Use rx_maxburst=0 to indicate Linerate. Only applicable in an SVC environment with class=vbr.	OPTIONAL

EXAMPLE:

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

RELATED COMMANDS:

qosbook delete

Remove a QoS book entry.

qosbook list

Show current QoS book.

qosbook delete

Remove a Quality of Service book entry.

SYNTAX:

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

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

EXAMPLE:

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

RELATED COMMANDS:

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

qosbook flush

Flush complete Quality of Service book. The flush command does not impact previously saved configurations.

SYNTAX:

```
qosbook flush
```

EXAMPLE:

```
=>qosbook list
Name      Ref Type      TX peek  sust      burst      RX peek  sust      burst
          (Kbits) (Kbits) (bytes) (Kbits) (Kbits) (bytes)
default 24  ubr      Linerate 0          0          Linerate 0          0
PPP3    1  ubr      6144     0          0          Linerate 0          0
=>qosbook save
=>qosbook add name voice class cbr tx peekrate 64 rx peekrate 64
=>qosbook list
Name      Ref Type      TX peek  sust      burst      RX peek  sust      burst
          (Kbits) (Kbits) (bytes) (Kbits) (Kbits) (bytes)
default 24  ubr      Linerate 0          0          Linerate 0          0
voice    0  cbr      64       0          0          64       0          0
PPP3    1  ubr      6144     0          0          Linerate 0          0
=>qosbook flush
=>qosbook list
Name      Ref Type      TX peek  sust      burst      RX peek  sust      burst
          (Kbits) (Kbits) (bytes) (Kbits) (Kbits) (bytes)
=>qosbook load
sustainable rate & burstsize ignored for UBR & CBR
sustainable rate & burstsize ignored for UBR & CBR
=>qosbook list
Name      Ref Type      TX peek  sust      burst      RX peek  sust      burst
          (Kbits) (Kbits) (bytes) (Kbits) (Kbits) (bytes)
default 0  ubr      Linerate 0          0          Linerate 0          0
PPP3    0  ubr      6144     0          0          Linerate 0          0
=>
```

RELATED COMMANDS:

qosbook load

Load saved or default QoS book.

qosbook save

Save current QoS book.

qosbook list

Show current Quality of Service book.

SYNTAX:

qosbook list

EXAMPLE OUTPUT:

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

RELATED COMMANDS:

qosbook add

Add a QoS book entry.

qosbook delete

Remove a QoS book entry.

qosbook load

Load saved (or default) Quality of Service book.

Execute **qosbook flush** prior to **qosbook load**.

SYNTAX:

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

EXAMPLE:

```
=>qosbook list
Name      Ref Type      TX peek  sust      burst      RX peek  sust      burst
          (Kbits) (Kbits) (bytes) (Kbits) (Kbits) (bytes)
default 24  ubr      Linerate 0        0        Linerate 0        0
PPP3    1  ubr      6144     0        0        Linerate 0        0
=>qosbook save
=>qosbook add name voice class cbr tx peekrate 64 rx peekrate 64
=>qosbook list
Name      Ref Type      TX peek  sust      burst      RX peek  sust      burst
          (Kbits) (Kbits) (bytes) (Kbits) (Kbits) (bytes)
default 24  ubr      Linerate 0        0        Linerate 0        0
voice    0  cbr      64       0        0        64       0        0
PPP3    1  ubr      6144     0        0        Linerate 0        0
=>qosbook save
=>qosbook flush
=>qosbook list
Name      Ref Type      TX peek  sust      burst      RX peek  sust      burst
          (Kbits) (Kbits) (bytes) (Kbits) (Kbits) (bytes)
=>qosbook load defaults=no
sustainable rate & burstsize ignored for UBR & CBR
sustainable rate & burstsize ignored for UBR & CBR
=>qosbook list
Name      Ref Type      TX peek  sust      burst      RX peek  sust      burst
          (Kbits) (Kbits) (bytes) (Kbits) (Kbits) (bytes)
default 0   ubr      Linerate 0        0        Linerate 0        0
voice    0  cbr      64       0        0        64       0        0
PPP3    0  ubr      6144     0        0        Linerate 0        0
=>
```

RELATED COMMANDS:

qosbook flush Flush complete QoS book.
qosbook save Save current QoS book.

qosbook save

Save current Quality of Service book.

SYNTAX:

```
qosbook save
```

EXAMPLE:

```
=>qosbook list
Name      Ref Type      TX peek  sust      burst      RX peek  sust      burst
          (Kbits) (Kbits) (bytes) (Kbits) (Kbits) (bytes)
default 0   ubr      Linerate 0         0         Linerate 0         0
PPP3    0   ubr      6144     0         0         Linerate 0         0
=>qosbook add name voice class cbr tx peekrate 64 rx peekrate 64
=>qosbook list
Name      Ref Type      TX peek  sust      burst      RX peek  sust      burst
          (Kbits) (Kbits) (bytes) (Kbits) (Kbits) (bytes)
default 0   ubr      Linerate 0         0         Linerate 0         0
PPP3    0   ubr      6144     0         0         Linerate 0         0
voice   0   cbr      64       0         0         64       0         0
=>qosbook save
=>qosbook flush
=>qosbook load
sustainable rate & burstsize ignored for UBR & CBR
sustainable rate & burstsize ignored for UBR & CBR
sustainable rate & burstsize ignored for UBR & CBR
=>qosbook list
Name      Ref Type      TX peek  sust      burst      RX peek  sust      burst
          (Kbits) (Kbits) (bytes) (Kbits) (Kbits) (bytes)
default 0   ubr      Linerate 0         0         Linerate 0         0
PPP3    0   ubr      6144     0         0         Linerate 0         0
voice   0   cbr      64       0         0         64       0         0
=>
```

RELATED COMMANDS:

qosbook flush

Flush complete QoS book.

qosbook load

Load saved (or default) QoS book.

16 SHDSL Commands

shdsl (to access the SHDSL level)

shdsl info

shdsl load

shdsl psd

shdsl save

shdsl info

Show SHDSL statistics and information about the **SpeedTouch™** status.

SYNTAX:

shdsl info

EXAMPLE:

<pre>=>shdsl info WAN interface : SHDSL Line Status : UP at 2304 kbit/s Total bytes since power on: Downstream : 9932253 Upstream : 244701 PSD mask : Symmetric PSD mask for Europe Shdsl uptime : 47:55:28 SNR margin : 7 dB Attenuation : 24 dB Transmit Power : 15 dBm =></pre>
--

shdsl load

Load saved (or default) SHDSL configuration.

SYNTAX:

shdsl load	[defaults <{yes no}>]	
[defaults]	Load factory defaults (yes) or saved configuration (no). Not specifying this parameter loads the saved configuration	OPTIONAL

EXAMPLE:

```
=>shdsl load default=yes
=>shdsl info
WAN interface      : SHDSL
Line Status        : UP at 2304 kbit/s
Total bytes since power on:
  Downstream      : 9932253          Upstream      : 244701
PSD mask           : Symmetric PSD mask for Europe
Shdsl uptime       : 47:55:28
SNR margin         : 7 dB
Attenuation        : 24 dB
Transmit Power     : 15 dBm
=>shdsl psd mask=asym_EU_2304
=>shdsl info
WAN interface      : SHDSL
Line Status        : DATA MODE
Total bytes since power on:
  Downstream      : 0                Upstream      : 0
PSD mask           : Asymmetric PSD mask for 2304kbps (European)
Shdsl uptime       : 0:00:00
=>shdsl save
=>shdsl load defaults=yes
=>shdsl info
WAN interface      : SHDSL
Line Status        : UP at 2304 kbit/s
Total bytes since power on:
  Downstream      : 895              Upstream      : 632
PSD mask           : Symmetric PSD mask for Europe
Shdsl uptime       : 00:00:15
SNR margin         : 7 dB
Attenuation        : 24 dB
Transmit Power     : 15 dBm
=>shdsl load defaults=no
=>shdsl info
WAN interface      : SHDSL
Line Status        : DATA MODE
Total bytes since power on:
  Downstream      : 0                Upstream      : 0
PSD mask           : Asymmetric PSD mask for 2304kbps (European)
Shdsl uptime       : 0:00:00
=>
```

RELATED COMMANDS:

shdsl save Save current SHDSL configuration.

shdsl psd

Configure the PSD mask to be used, then restart the SHDSL line.

SYNTAX:

shdsl psd	mask = <{sym_NA sym_EU asym_NA_768 asym_NA_1536 asym_EU_2048 asym_EU_2304}>
------------------	---

<i>mask</i>	<p>The selected PSD mask must comply to the regional demands and to the required payload rate. REQUIRED</p> <p>Choose between:</p> <ul style="list-style-type: none"> ▪ sym_NA Symmetric PSD mask for North America ▪ sym_EU Symmetric PSD mask for Europe ▪ asym_NA_768 Asymmetric PSD mask for 768kbps (North American) ▪ asym_NA_1536 Asymmetric PSD mask for 1536bps (North American) ▪ asym_EU_2048 Asymmetric PSD mask for 2048bps (European) ▪ asym_EU_2304 Asymmetric PSD mask for 2304bps (European)
-------------	---

EXAMPLE:

```
=>shdsl info
WAN interface      : SHDSL
Line Status        : UP at 2304 kbit/s
Total bytes since power on:
  Downstream : 9932253      Upstream   : 244701
PSD mask          : Symmetric PSD mask for Europe
Shdsl uptime      : 47:55:28
SNR margin         : 7 dB
Attenuation        : 24 dB
Transmit Power     : 15 dBm
=>shdsl psd mask=asym_EU_2304
=>shdsl info
WAN interface      : SHDSL
Line Status        : DATA MODE
Total bytes since power on:
  Downstream : 0           Upstream   : 0
PSD mask          : Asymmetric PSD mask for 2304kbps (European)
Shdsl uptime      : 0:00:00
=>
```

shdsl save

Save current SHDSL configuration.

SYNTAX:

```
shdsl save
```

EXAMPLE:

```
=>shdsl info
WAN interface      : SHDSL
Line Status        : UP at 2304 kbit/s
Total bytes since power on:
  Downstream      : 9932253      Upstream      : 244701
PSD mask           : Symmetric PSD mask for Europe
Shdsl uptime       : 47:55:28
SNR margin         : 7 dB
Attenuation        : 24 dB
Transmit Power     : 15 dBm
=>shdsl psd mask=asym_EU_2304
=>shdsl info
WAN interface      : SHDSL
Line Status        : DATA MODE
Total bytes since power on:
  Downstream      : 0            Upstream      : 0
PSD mask          : Asymmetric PSD mask for 2304kbps (European)
Shdsl uptime       : 0:00:00
=>shdsl save
=>shdsl flush
=>shdsl load defaults=yes
=>shdsl info
WAN interface      : SHDSL
Line Status        : UP at 2304 kbit/s
Total bytes since power on:
  Downstream      : 895          Upstream      : 632
PSD mask          : Symmetric PSD mask for Europe
Shdsl uptime       : 00:00:15
SNR margin         : 7 dB
Attenuation        : 24 dB
Transmit Power     : 15 dBm
=>shdsl flush
=>shdsl load defaults=no
=>shdsl info
WAN interface      : SHDSL
Line Status        : DATA MODE
Total bytes since power on:
  Downstream      : 0            Upstream      : 0
PSD mask          : Asymmetric PSD mask for 2304kbps (European)
Shdsl uptime       : 0:00:00
=>
```

RELATED COMMANDS:

shdsl load Load saved (or default) SHDSL configuration.

17 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:

<i>software cleanup</i>

EXAMPLE:

<pre>=>software cleanup =></pre>
--

RELATED COMMANDS:

software deletepassive
software setpassive

Delete the passive software.

Mark an uploaded file as passive software version.

software deletepassive

Delete passive software.

SYNTAX:

<i>software deletepassive</i>

EXAMPLE:

<pre>=>Software version => =>Active : Sascha3.426 Passive : Bene3.416 => =>Software deletepassive => =>Software version => =>Active : Sascha3.426 <u>Passive :</u> =></pre>
--

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 this **Speed Touch™** product can be marked as passive software.

SYNTAX:

software setpassive file = <string>

<i>file</i>	the filename (without directory path) of the software package.	REQUIRED
-------------	--	----------

EXAMPLE:

```
=>Software version
Active : KHDSAA3.264           Passive : KHDSAA3.235
=>Software deletepassive
=>Software version
Active : KHDSAA3.264           Passive :
.....
(FTP file transfer or upload via the Speed Touch™ pages of new software KHDSBA3.264)
.....
=>software setpassive file=KHDSBA3.264
=>Software version
Active : KHDSAA3.264           Passive : KHDSBA3.264
=>
```

RELATED COMMANDS:

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

software switch

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

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

SYNTAX:

software switch	[copyconfig={no yes}>]
------------------------	----------------------------------

copyconfig	Switch software versions and reboot to come online with the currently saved configuration (no) or switch software versions and reboot to come online with the new configuration (placed on passive subdirectory via an FTP session). If not specified or available the Speed Touch ™ will switch software versions and will reboot to come online with the currently saved configuration.	REQUIRED
-------------------	---	----------

EXAMPLE:

```
=>Software version
Active : KHDSAA3.264           Passive : KHDSBA3.264
=>software switch
.....
(after reboot and re-opening the (Telnet) session)
.....
=>Software version
Active : KHDSBA3.264           Passive : KHDSAA3.264
=>
```

RELATED COMMANDS:

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

software version

Show active and passive software versions.

SYNTAX:

software version

EXAMPLE:

<pre>=>software version Active : GXKLAA3.426 Passive : GXKLAA3.246 =></pre>

RELATED COMMANDS:

software switch

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

18 System Commands

system (to access the System level)

system clearpassword

system flush

system load

system reboot

system save

system setpassword

system clearpassword

Clear current **Speed Touch**™ system password.

To allow unauthorized access to the **Speed Touch**™ – which is not recommended – even after a reboot of the **Speed Touch**™, execute the **system save** command directly after executing this command.

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

SYNTAX:

<i>system clearpassword</i>

EXAMPLE:

<pre>=>system clearpassword =></pre>
--

RELATED COMMANDS:

system setpassword Set/change current system password.

system flush

Flush current **Speed Touch**™ system configuration.

The flush command does not impact previously saved configurations.

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

SYNTAX:

<i>system flush</i>

EXAMPLE:

<pre>=>system flush =></pre>

RELATED COMMANDS:

system load
system save

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

system load

Load saved (or default) system configuration.

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

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

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

SYNTAX:

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

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

EXAMPLE:

<pre>=>system load defaults =></pre>
--

RELATED COMMANDS:

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

system reboot

Reboot the **Speed Touch™**.

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

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

SYNTAX:

system reboot

EXAMPLE:

<pre>=>system reboot (lost session connectivity due to reboot)</pre>

RELATED COMMANDS:

software switch

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

system save

Save current system configuration.

This command saves the currently configured **Speed Touch**™ system password to persistent memory.

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

SYNTAX:

system save

EXAMPLE:

<pre>=>system save =></pre>

RELATED COMMANDS:

system load

Load saved or default system configuration.

system flush

Flush complete system configuration.

system setpassword

Set/change the current **Speed Touch™** system password.

Because rebooting implies a flush of all non-saved configurations it is highly recommended to save the current configuration via the **system save** command.

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

SYNTAX:

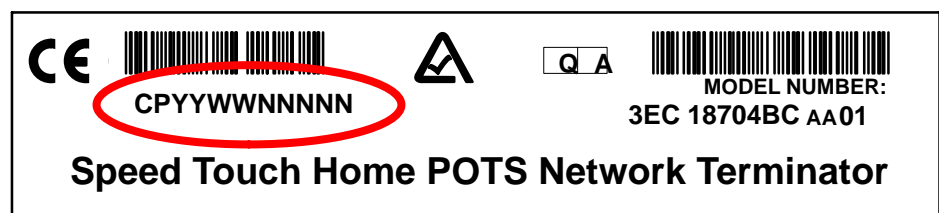
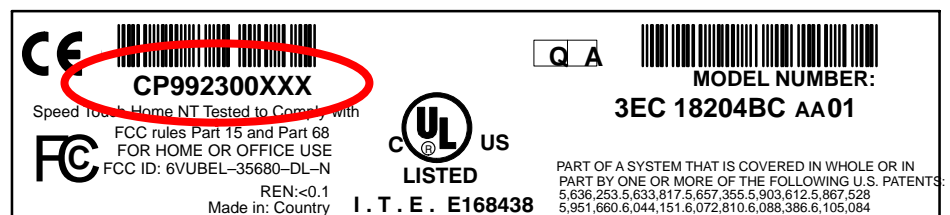
```
system setpassword password = {<string>|$_BOARD_SERIAL_NBR}
```

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

IMPORTANT NOTE:

Serial number

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



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 !!!

EXAMPLE 1:

```
=>system setpassword password=Sascha
=>
```

EXAMPLE 2:

```
=>system setpassword password=$_BOARD_SERIAL_NBR
=>system save
```

• • • • •

After reboot a telnet session is opened to the **Speed Touch™** with the following serial number: **013214578** (remind to strip of the string 'CP')

```
/home/doejohn{1}$ telnet 10.0.0.138
Trying 10.0.0.138...
Connected to 10.0.0.138.
Escape character is '^]'.
User :
SpeedTouch (00-90-D0-00-01-23-45)
Password : 013214578
```

[illegible]

RELATED COMMANDS:

system clearpassword	Clear current system password.
-----------------------------	--------------------------------

19 TD Commands

td (to access this level)
td call
td prompt

td call

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

SYNTAX:

<i>td call</i>	<i>cmd = <string></i> <i>[pwd = <string>]</i>
-----------------------	--

<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:

<i>td prompt</i>

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 **Speed Touch™** from unrestricted and unauthorized access.

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

20 Voice Commands

voice (to access the System level)
voice config
voice ifconfig
voice load
voice save

voice config

Show current voice configuration. Optionally set VoDSL destination.

SYNTAX:

voice config	proto dest = <atmaddress vpi*vci>
---------------------	--

proto	Indicates the voice gateway for the VoDSL service. This parameter depends on the regulatory domain the Speed Touch™ is used in and can/may not be changed.	
dest	The ATM address for VoDSL service. 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.	OPTIONAL

EXAMPLE:

```
=>voice config
proto = Alcatel VoDSL Gateway protocol dest = 0*40
=>voice config dest = 0*39
=>voice config
proto = Alcatel VoDSL Gateway protocol dest = 0*39
=>
```

RELATED COMMANDS:

voice save	Save current VoDSL configuration.
voice load	Load saved or default VoDSL configuration.

voice ifconfig

Show current VoDSL port configuration setting. Optionally set start condition.

SYNTAX:

voice ifconfig	intf = <number{0–3(7)}> [ground_start={off on}]	
intf	The VoDSL port to configure.	REQUIRED
[ground_start]	Configure, if available, the start condition of the particular VoDSL port. Choose between: <ul style="list-style-type: none">▪ off (Loopstart condition)▪ on (Groundstart condition)	OPTIONAL

EXAMPLE:

```
=>voice ifconfig intf=2
intf=2 pcm_loop=off tx_loop=off rx_loop=off ground_start=off
=>voice ifconfig intf=2 groundstart=on
=>voice ifconfig intf=2
intf=2 pcm_loop=off tx_loop=off rx_loop=off ground_start=on
=>
```

RELATED COMMANDS:

voice save	Save current VoDSL configuration.
voice load	Load saved or default VoDSL configuration.

voice load

Load saved (or default) VoDSL configuration.

SYNTAX:

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

[defaults]

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

OPTIONAL

EXAMPLE:

<pre>=>voice load defaults yes =></pre>

RELATED COMMANDS:

voice save

Save current VoDSL configuration.

voice save

Save current VoDSL configuration.

SYNTAX:

voice save

EXAMPLE:

<pre>=>voice save =></pre>

RELATED COMMANDS:

voice load Load saved or default VoDSL configuration.

21 Wireless Commands

wireless (to access the Wireless level)

wireless beacon_period

wireless channel

wireless counters

wireless deletemac

wireless disablecontrol

wireless enablecontrol

wireless filtconfig

wireless flush

wireless fragment_thres

wireless listacl

wireless long_retry_lim

wireless params

wireless random

wireless reset

wireless rts_threshold

wireless short_retry_lim

wireless ssid

wireless startWEP

wireless stopWEP

wireless status

wireless wepkey

wireless beacon_period

Set the value of the beacon period.

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

SYNTAX:

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

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

EXAMPLE:

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

RELATED COMMANDS:

wireless params	Show Speed Touch ™ wireless communication configuration.
wireless counters	Show Speed Touch ™ wireless communication counters.

wireless channel

Set WLAN radio channel.

SYNTAX:

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

EXAMPLE:

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

RELATED COMMANDS:

wireless status Show **Speed Touch™** WLAN configuration.

wireless counters

Show some wireless communication counters.

SYNTAX:

wireless counters

EXAMPLE:

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

RELATED COMMANDS:

wireless params

Show **Speed Touch**™ wireless communication configuration.

wireless deletemac

Delete a WLAN client in the access control list.
This command only applies in case association control is enabled.

SYNTAX:

wireless deletemac	deletemac = <string>
---------------------------	-----------------------------------

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

EXAMPLE:

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

RELATED COMMANDS:

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

wireless disablecontrol

Disable association control.

SYNTAX:

wireless disablecontrol

EXAMPLE:

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

RELATED COMMANDS:

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

wireless enablecontrol

Enable association control.

SYNTAX:

```
wireless enablecontrol
```

EXAMPLE:

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

RELATED COMMANDS:

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

wireless filtconfig

Show/set association control of a WLAN client.

This command only applies in case association control is enabled.

SYNTAX:

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

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

EXAMPLE:

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

RELATED COMMANDS:

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

wireless flush

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

SYNTAX:

```
wireless flush
```

EXAMPLE:

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

RELATED COMMANDS:

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

wireless fragment_thres

Set the value of the fragmentation threshold.

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

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

SYNTAX:

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

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

EXAMPLE:

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

RELATED COMMANDS:

wireless params	Show Speed Touch ™ wireless communication configuration.
wireless counters	Show Speed Touch ™ wireless communication counters.

wireless listacl

Show access control list.

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

SYNTAX:

wireless listacl

EXAMPLE:

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

RELATED COMMANDS:

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

wireless long_retry_lim

Set the value of the long retry limit.

SYNTAX:

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

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

EXAMPLE:

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

RELATED COMMANDS:

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

wireless params

Show **Speed Touch**™ wireless communication configuration.

SYNTAX:

```
wireless params
```

EXAMPLE:

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

RELATED COMMANDS:

wireless counters

Show **Speed Touch**™ wireless communication counters.

wireless random

Generate a random hexadecimal value for the WEP key.

SYNTAX:

wireless random

EXAMPLE:

<pre>=>wireless random The random key = 41:e6:27:d4:7d =></pre>

RELATED COMMANDS:

wireless startWEP

Start WEP encryption.

wireless stopWEP

Stop WEP encryption.

wireless wepkey

Set a 40-bits WEP key yourself.

wireless reset

Reset **Speed Touch™** WLAN configuration to defaults.

wireless reset

Reset **Speed Touch**™ wireless parameters to default values.

SYNTAX:

```
wireless reset
```

EXAMPLE:

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

RELATED COMMANDS:

wireless status

Show **Speed Touch**™ WLAN configuration.

wireless rts_threshold

Set the value of the RTS Threshold.

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

SYNTAX:

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

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

EXAMPLE:

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

RELATED COMMANDS:

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

wireless short_retry_lim

Set the value of the short retry limit.

SYNTAX:

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

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

EXAMPLE:

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

RELATED COMMANDS:

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

wireless ssid

Set **Speed Touch**™ WLAN Service Set Identifier.

SYNTAX:

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

ssid	A string between 1 and 32 characters.	REQUIRED
------	---------------------------------------	----------

EXAMPLE:

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

RELATED COMMANDS:

wireless status Show **Speed Touch**™ WLAN configuration.

wireless startWEP

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

SYNTAX:

```
wireless startWEP
```

EXAMPLE:

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

RELATED COMMANDS:

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

wireless stopWEP

Stop WEP encryption.

SYNTAX:

wireless stopWEP

EXAMPLE:

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

RELATED COMMANDS:

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

wireless status

Show **Speed Touch**™ WLAN configuration.

SYNTAX:

```
wireless status
```

EXAMPLE:

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

RELATED COMMANDS:

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

wireless wepkey

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

SYNTAX:

wireless wepkey	wepkey = <string>
------------------------	--------------------------------

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

REQUIRED

EXAMPLE:

<pre>=>wireless wepkey wepkey = 1:22:33:4:ab =></pre>

RELATED COMMANDS:

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

Alcatel Speed Touch™

CLI Command Index

Index

A

add

- atmf add, 22
- dhcp server add, 78
- dns add, 100
- phonebook add, 198
- qosbook add, 236

apadd, ip apadd, 148

apdelete, ip apdelete, 150

aplist, ip aplist, 151

applist, nat applist, 182

arpadd, ip arpadd, 152

arpdelete, ip arpdelete, 153

arplist, ip arplist, 154

assign, firewall assign, 122

autolist, phonebook autolist, 200

B

beacon_period, wireless beacon_period, 274

bind, nat bind, 183

bindlist, nat bindlist, 184

C

call, td call, 264

channel, wireless channel, 275

cleanup, software cleanup, 250

clear

- dhcp client clear, 63
- dns clear, 101
- firewall rule clear, 134

clearpassword, system clearpassword, 256

client, dhcp server client, 79

clrstats

- dhcp server clrstats, 80

- dns clrstats, 102

config

- bridge config, 30

- dhcp client config, 64

- dhcp server config, 81

- ip config, 155

- voice config, 268

counters, wireless counters, 276

create

- firewall chain create, 123

- firewall rule create, 135

- nat create, 185

D

defserver, nat defserver, 187

delete

- atmf delete, 23

- dhcp server delete, 83

- dns delete, 103

- firewall chain delete, 124

- firewall rule delete, 139

- nat delete, 188

- phonebook delete, 201

- qosbook delete, 238

deletemac, wireless deletemac, 277

deletemac, wireless deletemac, 277

disable, nat disable, 189

disablecontrol, wireless disablecontrol, 278

domain, dns domain, 104

E

enable, nat enable, 190

enablecontrol, wireless enablecontrol, 279

erase, config erase, 56

F

filtconfig, wireless filtconfig, 280

flush

- atmf flush, 24
- bridge flush, 31
- cip flush, 46
- config flush, 57
- dhcp client flush, 65
- dhcp server flush, 84
- dns flush, 105
- firewall flush, 128
- firewall rule flush, 140
- ip flush, 158
- mer flush, 170
- nat flush, 191
- phonebook flush, 202
- ppp flush, 208
- pptp flush, 226
- qosbook flush, 239
- system flush, 257
- wireless flush, 281

fragment_thres, wireless fragment_thres, 282

fwddadd, dns fwddadd, 106

fwdddelete, dns fwdddelete, 107

fwddlist, dns fwddlist, 108

fwddtable, dns fwddtable, 109

H

help, 11

I

ifadd

- bridge ifadd, 32
- cip ifadd, 47
- dhcp client ifadd, 66
- mer ifadd, 171
- ppp ifadd, 209
- pptp ifadd, 227

ifattach

- bridge ifattach, 33
- dhcp client ifattach, 67
- mer ifattach, 172
- ppp ifattach, 210

ifconfig

- bridge ifconfig, 34
- dhcp client ifconfig, 68
- ip ifconfig, 159
- mer ifconfig, 173
- ppp ifconfig, 211
- voice ifconfig, 269

ifdelete

- bridge ifdelete, 36
- cip ifdelete, 48
- dhcp client ifdelete, 69
- mer ifdelete, 175
- ppp ifdelete, 215

ifdetach

- bridge ifdetach, 37
- mer ifdetach, 176
- ppp ifdetach, 216

iflist

- bridge iflist, 38
- cip iflist, 49
- dhcp client iflist, 70
- ip iflist, 160
- mer iflist, 177
- ppp iflist, 217

ifrelease, dhcp client ifrelease, 71

ifrenew, dhcp client ifrenew, 73

ifscan, ppp ifscan, 218

info

- adsl info, 20
- shdsl info, 244

L

list

- atmf list, 25
- dhcp server list, 85
- dns list, 110
- firewall chain list, 125
- firewall list, 129
- firewall rule list, 141
- nat list, 192
- phonebook list, 203
- pptp list, 228
- qosbook list, 240

listacl, wireless listacl, 283

load

- atmf load, 26
- bridge load, 39
- cip load, 50
- config load, 58
- dhcp client load, 75
- dhcp server load, 86
- dns load, 111
- firewall chain load, 126
- firewall load, 130
- ip load, 161
- mer load, 178
- nat load, 193
- phonebook load, 204
- ppp load, 219
- pptp load, 229
- qosbook load, 241
- shdsl load, 245
- system load, 258
- voice load, 270

long_retry_lim, wireless long_retry_lim, 284

M

macadd, bridge macadd, 40

macdelete, bridge macdelete, 41

maclist, bridge maclist, 42

match, firewall match, 131

N

nslookup, dns nslookup, 112

P

params, wireless params, 285

ping, ip ping, 162

policy, dhcp server policy, 87

profadd, pptp profadd, 230

profdelete, pptp profdelete, 231

proflist, pptp proflist, 232

prompt, td prompt, 265

psd, shdsl psd, 246

pvcadd, cip pvcadd, 51

pvcdelete, cip pvcdelete, 52

pvclist, cip pvclist, 53

R

random, wireless random, 286

reboot, system reboot, 259

reset

- config reset, 59

- wireless reset, 287

rtadd

- ip rtadd, 163

- ppp rtadd, 220

rtdelete

- ip rtdelete, 164

- ppp rtdelete, 222

rtlist, ip rtlist, 165

rts_threshold, wireless rts_threshold, 288

S

save
 atmf save, 27
 bridge save, 43
 cip save, 54
 config save, 60
 dhcp client save, 76
 dhcp server save, 88
 dns save, 113
 firewall chain save, 127
 firewall save, 144
 ip save, 166
 mer save, 179
 nat save, 194
 phonebook save, 205
 ppp save, 223
 pptp save, 233
 qosbook save, 242
 shdsl save, 247
 system save, 260
 voice save, 271

sendto, ip sendto, 167

setpassive, software setpassive, 252

setpassword, system setpassword, 261

short_retry_lim, wireless short_retry_lim, 289

spoof, dhcp server spoof, 89

ssid, wireless ssid, 290

start
 dhcp server start, 91
 dns start, 114

startWEP, wireless startWEP, 291

stats
 dhcp client stats, 77
 dhcp server stats, 92
 dns stats, 115
 firewall rule stats, 142

status
 dhcp server status, 94
 dns status, 116
 wireless status, 293

stop
 dhcp server stop, 95
 dns stop, 117

stopWEP, wireless stopWEP, 292

switch, software switch, 253

T

toutfwd, dns toutfwd, 118

troff
 dhcp server troff, 96
 dns troff, 119

tron
 dhcp server tron, 97
 dns tron, 120

U

unbind, nat unbind, 195

V

version, software version, 254

W

wepkey, wireless wepkey, 294