KEENETIC

CITY

AC750 Dual Band Whole Home Wi-Fi Router
with 4-port Managed Switch

Command Reference Guide

Model	City (KN-1510)
OS Version	3.7
Edition	1.120 11.02.2022

help.keenetic.net

Preface

This guide contains Command-Line Interface (CLI) commands to maintain the City device. This guide provides a complete listing of all possible commands. The other chapters provide examples of how to implement the most common of these commands, general information on the interrelationships between the commands and the conceptual background of how to use them.

1 Readership

This guide is for the networking or computer technician responsible for configuring and maintaining the City on-site. It is also intended for the operator who manages the City. This manual cover high-level technical support procedures available to Root administrators and City technical support personnel.

2 Organization

This manual covers the following topics:

Introduction to the CLI	Describes how to use the City Command-Line Interface (CLI), its hierarchical structure, authorization levels and its help features.
Command Reference	Provides an alphabetical list of the available CLI commands that you can use to configure the City device.

3 Document Conventions

Command descriptions use the following conventions:

boldface font	Commands and keywords are in **boldface**. Must be typed exactly as shown. Bold font is used as a user input in examples.
italic font	Arguments for which you supply values are in *italics*.
[*optional*]	Elements in square brackets are optional.
‹*replaceable*›	Elements in angle brackets are replaceable.
(x | y | z)	Alternative keywords are grouped in round brackets and separated by vertical bars.
[x | y | z]	Optional alternative keywords are grouped in brackets and separated by vertical bars.

Each command description is broken down into the following sub-sections:

Description	Description of what the command does.
Synopsis	The general format of the command.
Prefix **no**	The possibility of using **no** prefix with command.
Change settings	The ability of command to change the settings.
Multiple input	The possibility of multiple input.
Group entry	Name of the group that owns the command. If there is no group, this section does not displayed.
Interface type	Type of interface, which can be managed by the command. The section does not displayed, if this context has no meaning for the command. Interfaces used in the system and the relationships between them are shown in the diagrams below.
Arguments	List of arguments if they exists, and explanations to them.
Example	An illustration of how the command looks when invoked. Because the interface is straightforward, some of the examples are obvious, but they are included for clarity.

Notes, cautionary statements, and safety warnings use these conventions.

Note: Means "reader take note". Notes contain helpful suggestions or references to materials not contained in this manual.

Warning: Means "reader be careful". You are capable of doing something that might result in equipment damage or loss of data.

Contents Overview

Table of Contents

1

Product Overview

1.1 Hardware Configuration

CPU MediaTek MT7628AN MIPS® 24KEc 580 MHz

RAM Winbond W9751G6KB-25 64MB DDR2

Flash Winbond W25Q128JVSQ 16MB SPI

Ethernet

Ports	Chipset	Notes
4	Integrated	

Label	Speed	Notes
0	100 Mbps	WAN port
1	100 Mbps	
2	100 Mbps	
3	100 Mbps	

Wi-Fi

Band	Chipset	Notes
2.4 GHz	MediaTek MT7603 (on-die)	802.11bgn 2x2
5 GHz	MediaTek MT7610EN (PCIe Gen 1)	802.11an+ac 1x1

2

Introduction to the CLI

This chapter describes how to use the City Command-Line Interface (CLI), its hierarchical structure, authorization levels and its help features.

The primary tool for managing the City router is the command line interface (*CLI*). System settings can be defined as a sequence of commands, which can be executed to bring the device to the specified condition.

City has three types of settings:

Current settings	*running config* is a set of commands describing the current status of the system. Current settings are stored in RAM and reflect every change of the system settings. However, the content of RAM is lost when the device is turned off. To restore the settings after reboot, they must be saved in non-volatile memory.
Startup configuration	*startup config* is a sequence of commands, which is stored in a specific partition of the non-volatile memory. It is used to initialize the system immediately after startup.
Default settings	*default config* contains factory default settings of City. RESET button is used to reset startup configuration to the factory default.

Files `startup-config` and `running-config` can be edited manually, without participation of the command line. It should be remembered that the lines with `!` in the beginning are ignored by the parser and the arguments which contain spaces must be enclosed in double quotes (for example, ssid "Free Wi-Fi"). Quotes themselves are ignored by the parser.

Responsibility for the accuracy of the changes rests with their author.

2.1 Enter commands in the CLI

Command line interpreter in City is designed for beginners as well as experts. All command names and options are clear and easy to remember.

Commands are divided into groups and arranged in a hierarchy. Thus, to do a setting, the operator needs to enter a sequence of nested command group names (node commands), and then enter the final command with parameters.

For example, IP-address of the FastEthernet0/Vlan2 network interface is set using the **address** command, which is located in the **interface** → **ip** group:

```
(config)>interface FastEthernet0/Vlan2 ip address 192.168.15.43/24
Network address saved.
```

2.1.1 Entering a group

Some of the node commands (containing a group of child commands) can be “entered” to allow direct executing of the child commands without typing the node name as prefix. In this case the prompt is changed to indicate the entered group.

The **exit** command or [Ctrl]+[D] key combination can be used to exit a group.

For example, after entering the interface group the command line prompt is changed to `(config-if)`:

```
(config)>interface FastEthernet0/Vlan2
(config-if)>ip address 192.168.15.43/24
Network address saved.
(config-if)>[Ctrl]+[D]
(config)>
```

2.2 Getting Help and auto-completion

To make the configuring process as comfortable as possible, the CLI provides auto-completion of commands and parameters, hinting the operator, which commands are available at the current level of nesting. Auto-completion works by pressing [Tab]. Example:

```
(config)>in[Tab]

 interface - network interface configuration

(config)> interface Fa[Tab]

 Usage template:
 interface {name}

 Variants:
 FastEthernet0
 FastEthernet0/Vlan1
 FastEthernet0/Vlan2

(config)> interface FastEthernet0[Tab]

 Usage template:
 interface {name}

 Variants:
 FastEthernet0/Vlan1
 FastEthernet0/Vlan2

(config)> interface FastEthernet0[Enter]
(config-if)> ip[Tab]

      address - set interface IP address
        alias - add interface IP alias
         dhcp - enable dhcp client
```

```
          mtu - set Maximum Transmit Unit size
          mru - set Maximum Receive Unit size
 access-group - bind access-control rules
          apn - set 3G access point name

(config-if)> ip ad[Tab]

     address - set interface IP address

(config-if)> ip address[Tab]

 Usage template:
 address {address} {mask}

(config-if)> ip address 192.168.15.43[Enter]
Configurator error[852002]: address: argument parse error.
(config-if)> ip address 192.168.15.43/24[Enter]
Network address saved.
(config-if)>
```

Hint for the current command can always be displayed by pressing [Tab]. Example:

```
(config)> interface FastEthernet0/Vlan2 [Tab]

   description - set interface description
         alias - add interface name alias
   mac-address - set interface MAC address
        dyndns - DynDns updates
 security-level - assign security level
 authentication - configure authentication
           ip - set interface IP parameters
         igmp - set interface IGMP parameters
           up - enable interface
         down - disable interface

(config)> interface FastEthernet0/Vlan2
```

2.3 Prefix no

Prefix **no** is used to negate a command.

For example, the command **interface** is responsible for creating a network interface with the given name. When used with this command, prefix **no** causes the opposite action — removing of the interface:

```
(config)> no interface PPPoE0
```

If the command is composite, **no** can be placed in front of any member. For example, **service dhcp** enables the *DHCP* service. It consists of two parts: **service** — the group name in the hierarchy of commands, and **dhcp** — the final command. Prefix **no** can be placed either at the beginning, or in the middle. The action is the same in both cases: stopping of the service.

```
(config)> no service dhcp
(config)> service no dhcp
```

2.4 Multiple input

Many commands have the property of *idempotence*, which means that multiple input of a command has the same effect as as the single input. For example, entering **service http** adds a single line "`service http`" to the current settings, and re-entering does not change anything.

However, some of the commands allow you to add not a single, but multiple records, if they are entered with different arguments. For example, static routing table entries **ip route** or filters **access-list** are added sequentially and appear in the settings as a list:

Example 2.1. Using a command with multiple input

```
(config)> ip route 1.1.1.0/24 PPTP0
Network::RoutingTable: Added static route: 1.1.1.0/24 via PPTP0.
(config)> ip route 1.1.2.0/24 PPTP0
Network::RoutingTable: Added static route: 1.1.2.0/24 via PPTP0.
(config)> ip route 1.1.3.0/24 PPTP1
Network::RoutingTable: Added static route: 1.1.3.0/24 via PPTP1.
(config)> show running-config
...
ip route 1.1.1.0 255.255.255.0 PPTP0
ip route 1.1.2.0 255.255.255.0 PPTP0
ip route 1.1.3.0 255.255.255.0 PPTP1
...
```

Records from such tables can be removed one by one, using prefix **no** and arguments to identify the record you want to remove:

```
(config)> no ip route 1.1.2.0/24
Network::RoutingTable: Deleted static route: 1.1.2.0/24 via PPTP0.
(config)> show running-config
...
ip route 1.1.1.0 255.255.255.0 PPTP0
ip route 1.1.3.0 255.255.255.0 PPTP1
...
```

2.5 Saving to startup settings

Current and startup settings are stored in the files running-config and startup-config, respectively. To save the current settings in the non-volatile memory, copy them as shown below:

```
(config)> copy running-config startup-config
Copied: running-config -> startup-config
```

2.6 Delayed restart

If City device is located away from the operator and is managed remotely, there is a risk to lose control over it because of a misoperation. In this case it will be difficult to reboot and return to the saved settings.

The **system reboot** command lets you set a delayed restart timer, perform “risky” settings, then turn off the timer and save the changes. If connection to the device is lost during configuration, the operator will be enough to wait for automatic reboot and connect to the device again.

3

Command Reference

3.1 Core commands

Core commands are used to manage files on your device.

3.1.1 copy

Description Copy the contents of one file to another. Used for the firmware updating, saving the current settings, resetting to factory, etc.

Prefix no No

Change settings No

Multiple input No

Synopsis

(config)> **copy** *‹source› ‹destination›*

Arguments

Argument	Value	Description
`source`	*`Filename`*	Full path to the file to be copied in <file system>:<path> format
`destination`	*`Filename`*	Full path to the directory for the new file.

Example Current settings can be saved as follows:

```
(config)> copy running-config startup-config
```

```
(config)> copy log MyPassport:/log.txt
```

File names in this example are aliases. Full names of the configuration files are `system:running-config` and `flash:startup-config`, respectively.

History

Version	Description
2.00	The **copy** command has been introduced.

3.1.2 erase

Description Delete a file from the City device.

Prefix no No

Change settings Yes

Multiple input Yes

Synopsis

(config)> **erase** *‹filename›*

Arguments

Argument	Value	Description
filename	*Filename*	Specifies the file to be removed.

Example

```
(config)> erase ext-opkg:/.dlna_files.db
FileSystem::Repository: "ext-opkg:/.dlna_files.db" erased.
```

History

Version	Description
2.00	The **erase** command has been introduced.

3.1.3 exit

Description Leave the command node.

Prefix no No

Change settings No

Multiple input No

Synopsis

(config)> **exit**

Example

```
(show)> exit
Core::Configurator: Done.
(config)>
```

History

Version	Description
2.00	The **exit** command has been introduced.

3.1.4 ls

Description Display list of files from the specified directory.

Prefix no No

Change settings No

Multiple input No

Synopsis

(config)> **ls** [*‹directory›*]

Arguments

Argument	Value	Description
directory	*String*	Path to the directory. Must contain the name of the file system and path to the folder directly in the following format <file system>:<path>. Examples of file systems — flash, temp, proc, usb. etc.

Example

```
(config)> ls FILES:

              rel: FILES:

            entry, type = D:
                 name: com

            entry, type = R:
                 name: IMAX.mkv
                 size: 1886912512

            entry, type = D:
                 name: speedfan

            entry, type = D:
                 name: portable

            entry, type = D:
                 name: video

            entry, type = D:
                 name: Новая папка
```

History

Version	Description
2.00	The **ls** command has been introduced.

3.1.5 mkdir

Description Create a new directory.

Prefix no No

Change settings No

Multiple input No

Synopsis

(config)> **mkdir** *‹directory›*

Arguments

Argument	Value	Description
directory	*String*	Path to the directory.

Example

```
(config)> mkdir SANDSK:/test
FileSystem::Repository: "SANDSK:/test" created.

(config)> mkdir SANDSK:/test/onetest
FileSystem::Repository: "SANDSK:/test/onetest" created.
```

History

Version	Description
2.12	The **mkdir** command has been introduced.

3.1.6 more

Description Display the contents of a text file line by line.

Prefix no No

Change settings No

Multiple input No

Synopsis

(config)> **more** ‹*filename*›

Arguments

Argument	Value	Description
filename	*Filename*	Full path to the file or alias.

Example

```
(config)> more temp:/resolv.conf
nameserver 127.0.0.1
options timeout:1 attempts:1 rotate
```

History

Version	Description
2.00	The **more** command has been introduced.

3.2 access-list

Description Access to a group of commands to configure the selected list of packet filtering rules. If the list is not found, the command tries to create it. Such a list can be assigned to a network interface using **interface ip access-group** command.

Command with **no** prefix removes the list of rules.

Prefix no Yes

Change settings Yes

Multiple input Yes

Group entry `(config-acl)`

Synopsis

(config)> **access-list** *‹name›*

(config)> **no access-list** *‹name›*

Arguments

Argument	Value	Description
name	*String*	Filtering rules list name (*Access Control List*, ACL).

Example

```
(config)> access-list test_acl
Network::Acl: "test_acl" access list created.
(config-acl)>
```

```
(config)> no access-list test_acl
Network::Acl: "test_acl" access list removed.
```

History

Version	Description
2.00	The **access-list** command has been introduced.

3.2.1 access-list deny

Description

Add a packet filtering deny rule into a specified *ACL*.

Command with **no** prefix removes the rule.

Prefix no

Yes

Change settings

Yes

Multiple input

Yes

Synopsis

(config-acl)> **deny** (**tcp** | **udp**) *‹source› ‹source-mask›*
[**port**((*‹src-port-operator› ‹source-port›*)|
(**range** *‹source-port› ‹source-end-port›*))]
‹destination› ‹destination-mask›
[**port**((*‹dst-port-operator› ‹destination-port›*)|
(**range** *‹destination-port› ‹destination-end-port›*))]

(config-acl)> **deny** (**icmp** | **esp** | **gre** | **ipip** | **ip**) *‹source› ‹source-mask›*
‹destination› ‹destination-mask›

(config-acl)> **no deny** (**tcp** | **udp**) *‹source› ‹source-mask›*
[**port**((*‹src-port-operator› ‹source-port›*)|
(**range** *‹source-port› ‹source-end-port›*))]
‹destination› ‹destination-mask›
[**port**((*‹dst-port-operator› ‹destination-port›*)|
(**range** *‹destination-port› ‹destination-end-port›*))]

(config-acl)> **no deny** (**icmp** | **esp** | **gre** | **ipip** | **ip**) *‹source› ‹source-mask›*

‹destination› ‹destination-mask›

Arguments

Argument	Value	Description
tcp	*Keyword*	*TCP* protocol.
udp	*Keyword*	*UDP* protocol.
icmp	*Keyword*	*ICMP* protocol.
esp	*Keyword*	*ESP* protocol.
gre	*Keyword*	*GRE* protocol.
ipip	*Keyword*	*IP in IP* protocol.
ip	*Keyword*	*IP* protocol (include *TCP*, *UDP*, *ICMP* and other).
source	*IP-address*	The source address in the header of IP-packet.
source-mask	*IP-mask*	Mask to be applied to the source address in the header of IP-packet before comparison with *source*. There are two ways to enter the mask: the canonical form (for example, 255.255.255.0) and the form of prefix bit length (for example, /24).
source-port	*Integer*	Source port in the *TCP* or *UDP* header.
source-end-port	*Integer*	The end of the source range of ports.
src-port-operator	lt	Operator “less” to compare the port with the specified *source-port*.
	eq	Operator “equal” to compare the port with the specified *source-port*.
	gt	Operator “greater” to compare the port with the specified *source-port*.
destination	*IP-address*	The destination address in the header of IP-packet.
destination-mask	*IP-mask*	Mask to be applied to the destination address in the header of IP-packet before comparison with *destination*. There are two ways to enter the mask: in the canonical form (for example, 255.255.255.0) and in the form of prefix with bit length (for example, /24).
destination-port	*Integer*	Destination port in the *TCP* or *UDP* header.
destination-end-port	*Integer*	The end of the destination range of ports.
dst-port-operator	lt	Operator “less” to compare the port with the specified *destination-port*.
	eq	Operator “equal” to compare the port with the specified *destination-port*.

Argument	Value	Description
	`gt`	Operator “greater” to compare the port with the specified *`destination-port`*.

Example

```
(config-acl)> deny tcp 0.0.0.0/24 port eq 80 0.0.0.0/24 port ►
range 18 88
Network::Acl: Rule accepted.
```

```
(config-acl)> deny icmp 192.168.0.0 255.255.255.0 192.168.1.1 ►
255.255.255.0
Network::Acl: Rule accepted.
```

```
(config-acl)> no deny tcp 0.0.0.0/24 port eq 80 0.0.0.0/24 port ►
range 18 88
Network::Acl: Rule deleted.
```

```
(config-acl)> no deny icmp 192.168.0.0 255.255.255.0 192.168.1.1 ►
255.255.255.0
Network::Acl: Rule deleted.
```

History

Version	Description
2.00	The **access-list deny** command has been introduced.
2.06	New value `ip` was added to the `protocol` argument.
2.08	New protocols `esp`, `gre` and `ipip` were added.
2.09.A.2.1	Port ranges were added.

3.2.2 access-list permit

Description

Add a packet filtering permit rule into a specified *ACL*.

Command with **no** prefix removes the rule.

Prefix no

Yes

Change settings

Yes

Multiple input

Yes

Synopsis

(config-acl)> **permit** (**tcp** | **udp**) ‹*source*› ‹*source-mask*›
[**port**((‹*src-port-operator*› ‹*source-port*›)|
(**range** ‹*source-port*› ‹*source-end-port*›))]
‹*destination*› ‹*destination-mask*›
[**port**((‹*dst-port-operator*› ‹*destination-port*›)|
(**range** ‹*destination-port*› ‹*destination-end-port*›))]

(config-acl)> **permit** (**icmp** | **esp** | **gre** | **ipip** | **ip**) ‹*source*› ‹*source-mask*›
‹*destination*› ‹*destination-mask*›

(config-acl)> **no permit** (**tcp** | **udp**) ‹*source*› ‹*source-mask*›
[**port**((‹*src-port-operator*› ‹*source-port*›) |
(**range** ‹*source-port*› ‹*source-end-port*›))]
‹*destination*› ‹*destination-mask*›
[**port**((‹*dst-port-operator*› ‹*destination-port*›) |
(**range** ‹*destination-port*› ‹*destination-end-port*›))]

(config-acl)> **no permit** (**icmp** | **esp** | **gre** | **ipip** | **ip**) ‹*source*› ‹*source-mask*›
‹*destination*› ‹*destination-mask*›

Arguments

Argument	Value	Description
`tcp`	*Keyword*	*TCP* protocol.
`udp`	*Keyword*	*UDP* protocol.
`icmp`	*Keyword*	*ICMP* protocol.
`esp`	*Keyword*	*ESP* protocol.
`gre`	*Keyword*	*GRE* protocol.
`ipip`	*Keyword*	*IP in IP* protocol.
`ip`	*Keyword*	*IP* protocol (include *TCP*, *UDP*, *ICMP* and other).
`source`	*IP-address*	The source address in the header of IP-packet.
`source-mask`	*IP-mask*	Mask to be applied to the source address in the header of IP-packet before comparison with *source*. There are two ways to enter the mask: the canonical form (for example, `255.255.255.0`) and the form of prefix bit length (for example, `/24`).
`source-port`	*Integer*	Source port in the *TCP* or *UDP* header.
`source-end-port`	*Integer*	The end of the source range of ports.
`src-port-operator`	`lt`	Operator “less” to compare the port with the specified *source-port*.
	`eq`	Operator “equal” to compare the port with the specified *source-port*.
	`gt`	Operator “greater” to compare the port with the specified *source-port*.
`destination`	*IP-address*	The destination address in the header of IP-packet.
`destination-mask`	*IP-mask*	Mask to be applied to the destination address in the header of IP-packet before comparison with *destination*. There are two ways to enter the mask: in the canonical form (for example, `255.255.255.0`) and in the form of prefix with bit length (for example, `/24`).

Argument	Value	Description
destination-port	*Integer*	Destination port in the *TCP* or *UDP* header.
destination-end-port	*Integer*	The end of the destination range of ports.
dst-port-operator	lt	Operator "less" to compare the port with the specified *destination-port*.
	eq	Operator "equal" to compare the port with the specified *destination-port*.
	gt	Operator "greater" to compare the port with the specified *destination-port*.

Example

```
(config-acl)> permit icmp 192.168.0.0 255.255.255.0 192.168.1.1 ►
255.255.255.0
Network::Acl: Rule accepted.
```

```
(config-acl)> permit tcp 0192.168.1.0/24 port eq 443 0.0.0.0/24 ►
port range 8080 9090
Network::Acl: Rule accepted.
```

```
(config-acl)> no permit icmp 192.168.0.0 255.255.255.0 ►
192.168.1.1 255.255.255.0
Network::Acl: Rule deleted.
```

```
(config-acl)> no permit tcp 0192.168.1.0/24 port eq 443 ►
0.0.0.0/24 port range 8080 9090
Network::Acl: Rule deleted.
```

History

Version	Description
2.00	The **access-list permit** command has been introduced.
2.06	New value ip was added to the protocol argument.
2.08	New protocols esp, gre and ipip were added.
2.09.A.2.1	Port ranges were added.

3.2.3 access-list rule

Description

Disable, set operation time by schedule, change the order or set description for the *ACL* rule.

Command with **no** prefix enables the rule, removes schedule and description for *ACL* rule.

Prefix no Yes

Change settings Yes

Multiple input Yes

Synopsis

(config-acl)> **rule** ‹*index*› (**disable** | **schedule** ‹*schedule*› | **order** ‹*new-index*› | **description** ‹*description*›)

(config-acl)> **no rule** ‹*index*› (**disable** | **schedule** | **description**)

Arguments

Argument	Value	Description
index	*Integer*	The ACL rule number.
disable	*Keyword*	Disable the ACL rule.
schedule	*Schedule name*	The name of the schedule that was created with **schedule** group of commands.
order	*Integer*	New position of the ACL rule in the list.
description	*String*	The ACL rule description.

Example

```
(config-acl)> rule 0 disable
Network::Acl: Rule disabled.
```

```
(config-acl)> rule 0 schedule acl_schedule
Network::Acl: Rule schedule set to "acl_schedule".
```

```
(config-acl)>rule 0 description myacl
Network::Acl: Rule description set to "myacl".
```

```
(config-acl)>  rule 0 order 1
Network::Acl: Rule 0 moved to position 1.
```

```
(config-acl)> no rule 0 disable
Network::Acl: Rule enabled.
```

```
(config-acl)> no rule 0 schedule
Network::Acl: Rule schedule removed.
```

```
(config-acl)>  no rule 0 description
Network::Acl: Rule description removed.
```

History

Version	Description
2.08	The **access-list rule** command has been introduced.

3.3 adguard-dns

Description Access to a group of commands to configure *AdGuard DNS* profiles.

Prefix no No

Change settings No

Multiple input No

Group entry `(adguard-dns)`

Synopsis

(config)> **adguard-dns**

Example

```
(config)> adguard-dns
Core::Configurator: Done.
(adguard-dns)>
```

History

Version	Description
2.12	The **adguard-dns** command has been introduced.

3.3.1 adguard-dns assign

Description Assign profile of protection to the host. By default `standard` profile is used for all hosts.

Command with **no** prefix resets setting to default `standard` profile.

Prefix no Yes

Change settings Yes

Multiple input Yes

Synopsis

(adguard-dns)> **assign** [*‹host›*] *‹type›*

(adguard-dns)> **no assign** [*‹host›*]

Arguments

Argument	Value	Description
`host`	*`MAC-address`*	Host to which type of protection is applied. If not specified, the protection is applied to all hosts.
`type`	`default`	No protection used.
	`base`	Blocking advertising,tracking and phishing.
	`standard`	Secure DNS resolving, no blocking.
	`family`	Blocking advertising,tracking, phishing, adult sites, providing secure search.

Example

```
(adguard-dns)> assign base
AdguardDns::Client: Default type set.
```

```
(adguard-dns)> assign 4C:0F:6E:4B:3C:BA default
AdguardDns::Client: "4C:0F:6E:4B:3C:BA" has been associated with ►
"default" profile.
```

```
(adguard-dns)> assign 4C:0F:6E:4B:3C:BA standard
AdguardDns::Client: "4C:0F:6E:4B:3C:BA" has been reassociated ►
with "standard" profile.

(adguard-dns)> assign 4C:0F:6E:4B:3C:BA family
AdguardDns::Client: "4C:0F:6E:4B:3C:BA" has been reassociated ►
with "family" profile.

(adguard-dns)> no assign a8:1e:84:85:f2:72
AdguardDns::Client: Host "a8:1e:84:85:f2:72" has been removed.

(adguard-dns)> no assign
AdguardDns::Client: Default type set.
```

History

Version	Description
2.12	The **adguard-dns assign** command has been introduced.

3.3.2 adguard-dns check-availability

Description Check availability of *AdGuard DNS* service.

Prefix no No

Change settings No

Multiple input No

Synopsis

(adguard-dns)> **check-availability**

Example

```
(adguard-dns)> check-availability
AdguardDns::Client: AdGuard DNS is available.
```

History

Version	Description
2.12	The **adguard-dns check-availability** command has been introduced.

3.3.3 adguard-dns enable

Description Enable *AdGuard DNS* service.

Command with **no** prefix disables the service.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(adguard-dns)> **enable**

(adguard-dns)> **no enable**

Example

```
(adguard-dns)> enable
AdguardDns::Client: AdGuard DNS enabled.
```

```
(adguard-dns)> no enable
AdguardDns::Client: AdGuard DNS disabled.
```

History

Version	Description
2.12	The **adguard-dns enable** command has been introduced.

3.4 cloud control2 security-level

Description

Set Cloud Control2 service security level for Keenetic mobile application. By default, `public` value is set.

Prefix no

No

Change settings

Yes

Multiple input

No

Synopsis

(config)> **cloud control2 security-level** (**public** | **private**)

Arguments

Argument	Value	Description
`public`	*Keyword*	Access to the Cloud Control2 is allowed for public, private and protected interfaces.
`private`	*Keyword*	Access to the Cloud Control2 is allowed for private interfaces only.

Example

```
(config)> cloud control2 security-level public
CloudControl2::Agent: Security level changed to public.
```

```
(config)> cloud control2 security-level private
CloudControl2::Agent: Security level changed to private.
```

History

Version	Description
3.05	The **cloud control2 security-level** command has been introduced.

3.5 cloudflare-dns

Description Access to a group of commands to configure *Cloudflare DNS* profiles.

Prefix no No

Change settings No

Multiple input No

Group entry `(cloudflare-dns)`

Synopsis

(config)> **`cloudflare-dns`**

Example

```
(config)> cloudflare-dns
Core::Configurator: Done.
(cloudflare-dns)>
```

History

Version	Description
3.05	The **cloudflare-dns** command has been introduced.

3.5.1 cloudflare-dns assign

Description Assign profile of protection to the host. By default `standard` profile is used for all hosts.

Command with **no** prefix resets setting to default.

Prefix no Yes

Change settings Yes

Multiple input Yes

Synopsis

(cloudflare-dns)> **`assign`** [*‹host›*] *‹type›*

(cloudflare-dns)> **`no assign`** [*‹host›*]

Arguments

Argument	Value	Description
`host`	*`MAC-address`*	Host to which type of protection is applied. If not specified, the protection is applied to all hosts.
`type`	`default`	No protection used.
	`standard`	Secure DNS resolving, no blocking.
	`malware`	Blocking malware.
	`family`	Blocking malware and adult sites.

Example

```
(cloudflare-dns)> assign default
CloudflareDns::Client: Default type set.

(cloudflare-dns)> assign c0:b8:83:c2:cb:11 default
CloudflareDns::Client: "c0:b8:83:c2:cb:11" has been reassociated ►
with "default" profile.

(cloudflare-dns)> assign c0:b8:83:c2:cb:11 standard
CloudflareDns::Client: "c0:b8:83:c2:cb:11" has been reassociated ►
with "standard" profile.

(cloudflare-dns)> assign c0:b8:83:c2:cb:11 malware
CloudflareDns::Client: "c0:b8:83:c2:cb:11" has been reassociated ►
with "malware" profile.

(cloudflare-dns)> assign c0:b8:83:c2:cb:11 family
CloudflareDns::Client: "c0:b8:83:c2:cb:11" has been reassociated ►
with "family" profile.

(cloudflare-dns)> no assign c0:b8:83:c2:cb:11
CloudflareDns::Client: Host "c0:b8:83:c2:cb:11" has been removed.

(cloudflare-dns)> no assign
CloudflareDns::Client: Default type set.
```

History

Version	Description
3.05	The **cloudflare-dns assign** command has been introduced.

3.5.2 cloudflare-dns check-availability

Description Check availability of *Cloudflare DNS* service.

Prefix no No

Change settings No

Multiple input No

Synopsis

(cloudflare-dns)> **check-availability**

Example

```
(cloudflare-dns)> check-availability
CloudflareDns::Client: Cloudflare DNS is available.
```

History

Version	Description
3.05	The **cloudflare-dns check-availability** command has been introduced.

3.5.3 cloudflare-dns enable

Description Enable *Cloudflare DNS* service.

Command with **no** prefix disables the service.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(cloudflare-dns)> **`enable`**

(cloudflare-dns)> **`no enable`**

Example

```
(cloudflare-dns)> enable
CloudflareDns::Client: Cloudflare DNS enabled.
```

```
(cloudflare-dns)> no enable
CloudflareDns::Client: Cloudflare DNS disabled.
```

History

Version	Description
3.05	The **cloudflare-dns** command has been introduced.

3.6 components

Description Access to a group of commands to manage firmware components.

Prefix no No

Change settings No

Multiple input No

Group entry `(config-comp)`

Synopsis

(config)> **`components`**

History

Version	Description
2.00	The **components** command has been introduced.

3.6.1 components auto-update channel

Description Set source of components for auto-update feature. By default, value `stable` is used.

Command with **no** prefix resets setting to default.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config-comp)> **auto-update channel** *‹channel›*

(config-comp)> **no auto-update channel**

Arguments

Argument	Value	Description
channel	stable	Components have been fully tested and recommended for installation. The web interface specifies this channel as Main.
	preview	Components contain the latest features and enhancements, but have not been fully tested yet. The web interface specifies this channel as Preview.
	draft	The components contain the latest features and are used for testing. The web interface specifies this channel as Dev.

Example

```
(config-comp)> auto-update channel preview
Components::Manager: Auto-update channel is "preview".
```

```
(config-comp)> no auto-update channel
Components::Manager: Reset an auto-update channel to default.
```

History

Version	Description
3.01	The **components auto-update channel** command has been introduced.

3.6.2 components auto-update disable

Description Components auto-update function. By default, automatic update is enabled.

Command with **no** prefix enables auto-update.

Prefix no Yes

Change settings Yes

Multiple input Yes

Synopsis

(config-comp)> **auto-update disable**

(config-comp)> **no auto-update disable**

Example

```
(config-comp)> auto-update disable
Components::Manager: Components auto-update disabled.
```

```
(config-comp)> no auto-update disable
Components::Manager: Components auto-update enabled.
```

History

Version	Description
2.09	The **components auto-update disable** command has been introduced.

3.6.3 components auto-update schedule

Description

Assign a schedule for the auto-update operation. Schedule must be created and customized with **schedule action** command before execution.

Command with **no** prefix unbinds the schedule.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config-comp)> **auto-update schedule** ‹*schedule*›

(config-comp)> **no auto-update schedule**

Arguments

Argument	Value	Description
schedule	*Schedule name*	The name of the schedule that was created with **schedule** group of commands.

Example

```
(config-comp)> auto-update schedule Update
Components::Manager: Set auto-update schedule "Update".
```

```
(config-comp)> no auto-update schedule
Components::Manager: Schedule disabled.
```

History

Version	Description
3.03	The **components auto-update schedule** command has been introduced.

3.6.4 components check-update

Description

Check the firmware updates for the candidate or member of Modular Wi-Fi System.

Prefix no

No

Change settings No

Multiple input No

Synopsis

(config-comp)> **check-update** [*force*]

Arguments

Argument	Value	Description
force	*Keyword*	Check for updates constantly.

Example

```
(config-comp)> check-update

release: 2.15.A.3.0-2
          sandbox: draft
        timestamp: Dec 17 18:58:55
            valid: no
```

```
(config-comp)> check-update force

release: 2.15.A.3.0-2
          sandbox: draft
        timestamp: Dec 17 18:58:55
            valid: no
```

History

Version	Description
2.14	The **components check-update** command has been introduced.

3.6.5 components commit

Description Apply the changes made by **components install** and **components remove** commands.

Prefix no No

Change settings Yes

Multiple input No

Synopsis

(config-comp)> **commit**

History

Version	Description
2.00	The **components commit** command has been introduced.

3.6.6 components install

Description Mark component to install. Final installation carried out with **components commit** command.

Prefix no No

Change settings Yes

Multiple input Yes

Synopsis

(config-comp)> **install** ‹*component*›

Arguments

Argument	Value	Description
component	*String*	Component name. List of available components for installation can be displayed with the **components list** command.

Example

```
(config-comp)> install ntfs
Components::Manager: Component "ntfs" is queued for installation.
```

History

Version	Description
2.00	The **components install** command has been introduced.

3.6.7 components list

Description Switch to the selected sandbox and mark for installation all the components that require changes to match the version in the sandbox. If you use no argument, the entire list of all components for current sandbox (installed and available) will be displayed. If there is no Internet connection, only the list of installed components will be displayed.

Prefix no No

Change settings No

Multiple input No

Synopsis

(config-comp)> **list** [*sandbox*]

Arguments

Argument	Value	Description
sandbox	*String*	Remote sandbox, such as stable or beta.

Example

```
(config-comp)> list

        firmware:
```

```
             version: 2.13.C.0.0-1

          sandbox: stable

            local:
              sandbox: beta

        component:
                 name: base

             priority: optional
                 size: 35233
              version: 2.13.C.0.0-1
                 hash: f65428af2a6fd636db779370deb58f40
            installed: 2.13.B.1.0-1

                preset: minimal
                preset: recommended
                queued: yes
        ...
```

History

Version	Description
2.00	The **components list** command has been introduced.
2.06.A.6	The *sandbox* parameter has been introduced. The command **components list** should be used in favour of **components sync**.

3.6.8 components preset

Description

Select a predefined set of components. Installation of preset is carried out with **components commit** command.

Before preset installation check the latest versions of components on the update server with **components list** command. Internet connection is required.

Prefix no

No

Change settings

Yes

Multiple input

No

Synopsis

(config-comp)> **preset** ‹*preset*›

Arguments

Number and names of presets can be changed, so check the list of available presets with help of **preset** [Tab] command.

Argument	Value	Description
`preset`	`minimal`	Minimal set of components will be marked.

Argument	Value	Description
	recommended	Recommended set of components will be marked for installation.

Example

```
(config-comp)> preset [Tab]

 Usage template:
           preset {preset}

   Choose:
         minimal
     recommended
```

```
(config-comp)> preset recommended
lib::libndmComponents error[268369922]: updates are available ►
for this system.
(config-comp)> commit
Components::Manager: Update task started.
```

History

Version	Description
2.00	The **components preset** command has been introduced.

3.6.9 components preview

Description

Show size of firmware as current set of components selected with **components install** command.

Prefix no

No

Change settings

Yes

Multiple input

No

Synopsis

(config-comp)> **preview**

Example

```
(config-comp)> preview

          preview:
                 size: 7733308
```

History

Version	Description
2.06	The **components preview** command has been introduced.

3.6.10 components remove

Description Mark component to remove. Final removal carried out with **components commit** command.

Prefix no No

Change settings Yes

Multiple input Yes

Synopsis

(config-comp)> **remove** ‹*component*›

Arguments

Argument	Value	Description
component	*String*	Component name. List of available components for removal can be displayed with the **components list** command.

Example

```
(config-comp)> remove ntfs
Components::Manager: Component "ntfs" is queued for removal.
```

History

Version	Description
2.00	The **components remove** command has been introduced.

3.6.11 components validity-period

Description Set a validity period of a local component list. After this time the command **components list** will be automalically executed to get actual list of components from update server.

Command with **no** prefix resets period to default. By default, value 1800 is used.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config-comp)> **validity-period** ‹*seconds*›

(config-comp)> **no validity-period**

Arguments

Argument	Value	Description
seconds	*Integer*	Validity period of a local component list in seconds. Can

Argument	Value	Description
		take values in the range from 0 to 604800 inclusively.

Example

```
(config-comp)> validity-period 500
Components::Manager: Validity period set to 500 seconds.
```

```
(config-comp)> no validity-period
Components::Manager: Validity period reset to 1800 seconds.
```

History

Version	Description
2.03	The **components validity-period** command has been introduced.

3.7 crypto engine

Description

Select the type of *ESP* packets processing with *IPsec*.

Command with **no** prefix disables the feature.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config)> **crypto engine** *‹type›*

(config)> **no crypto engine**

Arguments

Argument	Value	Description
type	software	Software mode.

Example

```
(config)> crypto engine software
IpSec::CryptoEngineManager: IPsec crypto engine set to "software".
```

```
(config)> no crypto engine
IpSec::CryptoEngineManager: IPsec crypto engine was disabled.
```

History

Version	Description
2.06	The **crypto engine** command has been introduced.

3.8 crypto ike key

Description

Add *IKE* key with remote side ID.

Command with **no** prefix removes specified key.

Prefix no

Yes

Change settings

Yes

Multiple input

Yes

Synopsis

(config)> **crypto ike key** ‹*name*› ‹*psk*› (‹*type*› ‹*id*› | **any**)

(config)> **no crypto ike key** ‹*name*›

Arguments

Argument	Value	Description
name	*String*	Name of the key. Latin letters, numbers, dots, hyphens and underscores are acceptable.
psk	*String*	Password for authentication. Password length can be from 6 to 96 characters.
type	address	ID type is IP-address.
	fqdn	ID type is full domain name.
	dn	ID type is domain name.
	email	ID type is e-mail address.
id	*String*	Value of the remote side ID.
any	*Keyword*	Allow the key usage for any remote side.

Example

```
(config)> crypto ike key VirtualIPServer ►
aDjsOC1gvWCsOiE4Ijhs+HRnNPiheGA478 any
IpSec::Manager: "VirtualIPServer": crypto ike key successfully ►
added.
```

```
(config)>  crypto ike key VirtualIPServer ►
aDjsOC1gvWCsOiE4Ijhs+HRnNPiheGA478R4M6d4+O54LLihe any
IpSec::Manager: "VirtualIPServer": crypto ike key successfully ►
updated.
```

```
(config)> no crypto ike key VirtualIPServer
IpSec::Manager: "VirtualIPServer": crypto ike key successfully ►
removed.
```

History

Version	Description
2.06	The **crypto ike key** command has been introduced.

3.9 crypto ike nat-keepalive

Description

Set the timeout between keepalive packets in case of NAT between the client and server *IPsec*. By default, `20` value is set.

Command with **no** prefix resets setting to default.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config)> **crypto ike nat-keepalive** ‹*nat-keepalive*›

(config)> **no crypto ike nat-keepalive**

Arguments

Argument	Value	Description
`nat-keepalive`	*Integer*	Timeout between keepalive packets in seconds. Can take values from `5` to `3600` inclusively.

Example

```
(config)> crypto ike nat-keepalive 90
IpSec::Manager: Set crypto ike nat-keepalive timeout to 90 s.
```

```
(config)> no crypto ike nat-keepalive
IpSec::Manager: Reset crypto ike nat-keepalive timeout to 20 s.
```

History

Version	Description
2.06	The **crypto ike nat-keepalive** command has been introduced.

3.10 crypto ike policy

Description

Access to a group of commands to configure selected *IKE* policy. If *IKE* policy is not found, the command tries to create it.

Command with **no** prefix removes *IKE* policy. At the same time references to this *IKE* policy are automatically deleted from all *IPsec* profiles.

Prefix no

Yes

Change settings

Yes

Multiple input

Yes

Group entry

`(config-ike-policy)`

Synopsis

(config)> **crypto ike policy** *‹name›*

(config)> **no crypto ike policy** *‹name›*

Arguments

Argument	Value	Description
name	*String*	*IKE* policy name. Latin letters, numbers, dots, hyphens and underscores are acceptable.

Example

```
(config)> crypto ike policy test
IpSec::Manager: "test": crypto ike policy successfully created.
```

```
(config)> no crypto ike policy test
IpSec::Manager: Crypto ike policy "test" removed.
```

History

Version	Description
2.06	The **crypto ike policy** command has been introduced.

3.10.1 crypto ike policy lifetime

Description

Set lifetime of *IPsec IKE* association. By default, the value `86400` is used.

Command with **no** prefix resets setting to default.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config-ike-policy)> **lifetime** *‹lifetime›*

(config-ike-policy)> **no lifetime**

Arguments

Argument	Value	Description
lifetime	*Integer*	Lifetime of *IPsec IKE* association in seconds. Can take values from 60 to 2147483647.

Example

```
(config-ike-policy)> lifetime 3600
IpSec::Manager: "test": crypto ike policy lifetime set to 3600 s.
```

```
(config-ike-policy)> no lifetime
IpSec::Manager: "test": crypto ike policy lifetime reset.
```

History

Version	Description
2.06	The **crypto ike policy lifetime** command has been introduced.

3.10.2 crypto ike policy mode

Description

Set *IKE* protocol version. By default, the value `ikev1` is used.

Command with **no** prefix resets setting to default.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config-ike-policy)> **mode** *‹mode›*

(config-ike-policy)> **no mode**

Arguments

Argument	Value	Description
mode	ikev1	Protocol version IKEv1.
	ikev2	Protocol version IKEv2.

Example

```
(config-ike-policy)> mode ikev2
IpSec::Manager: "test": crypto ike policy mode set to "ikev2".
```

```
(config-ike-policy)> no mode
IpSec::Manager: "test": crypto ike policy mode reset.
```

History

Version	Description
2.06	The **crypto ike policy mode** command has been introduced.

3.10.3 crypto ike policy negotiation-mode

Description

Set exchange mode for IKEv1 (see **crypto ike policy mode** command). By default, the value `main` is used.

Command with **no** prefix resets setting to default.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config-ike-policy)> **negotiation-mode** ‹*negotiation-mode*›

(config-ike-policy)> **no negotiation-mode**

Arguments

Argument	Value	Description
negotiation-mode	main	Main mode, protects the identity of the peers.
	aggressive	Aggressive mode, does not protect the identity of the peers.

Example

```
(config-ike-policy)> negotiation-mode aggressive
IpSec::Manager: "test": crypto ike policy negotiation-mode set ►
to "aggressive".
```

```
(config-ike-policy)> no negotiation-mode
IpSec::Manager: "test": crypto ike policy negotiation-mode reset.
```

History

Version	Description
2.06	The **crypto ike policy negotiation-mode** command has been introduced.

3.10.4 crypto ike policy proposal

Description

Add reference on existing *IKE* proposal to *IKE* policy. The order of adding has a value for data exchange on the *IKE* protocol.

Command with **no** prefix removes reference on *IKE* proposal.

Prefix no

Yes

Change settings

Yes

Multiple input

Yes

Synopsis

(config-ike-policy)> **proposal** ‹*proposal*›

(config-ike-policy)> **no proposal** ‹*proposal*›

Arguments

Argument	Value	Description
proposal	*String*	*IKE* proposal name. Latin letters, numbers, dots, hyphens and underscores are acceptable.

Example

```
(config-ike-policy)> proposal test
IpSec::Manager: "test": crypto ike proposal "test" successfully ►
added.
```

```
(config-ike-policy)> no proposal
IpSec::Manager: "test": crypto ike policy proposal "test" ►
successfully removed.
```

History

Version	Description
2.06	The **crypto ike policy proposal** command has been introduced.

3.11 crypto ike proposal

Description

Access to a group of commands to configure selected *IKE* proposal. If *IKE* proposal is not found, the command tries to create it.

A full list of encryption algorithms implemented in the system is provided in the Appendix.

Command with **no** prefix removes *IKE* proposal. At the same time references to this *IKE* proposal are automatically deleted from all *IKE* policy.

Prefix no

Yes

Change settings

Yes

Multiple input

Yes

Group entry

`(config-ike-proposal)`

Synopsis

(config)> **crypto ike proposal** *‹name›*

(config)> **no crypto ike proposal** *‹name›*

Arguments

Argument	Value	Description
name	*String*	*IKE* proposal name. Latin letters, numbers, dots, hyphens and underscores are acceptable.

Example

```
(config)> crypto ike proposal test
IpSec::Manager: "test": crypto ike proposal successfully created.
```

```
(config)> no crypto ike proposal test
IpSec::Manager: Crypto ike proposal "test" removed.
```

History

Version	Description
2.06	The **crypto ike proposal** command has been introduced.

3.11.1 crypto ike proposal aead

Description Enable *AEAD* cypher mode on *IKE* proposal.

Prefix no No

Change settings No

Multiple input No

Synopsis

(config-ike-proposal)> **aead**

Example

```
(config-ike-proposal)> aead
IpSec::Manager: "TEST": crypto ike proposal "TEST" enabled AEAD ►
mode.
```

History

Version	Description
3.05	The **crypto ike proposal aead** command has been introduced.

3.11.2 crypto ike proposal dh-group

Description Add the selected *DH* group to *IKE* proposal to work in the *PFS* mode. The order of adding has a value for data exchange on the *IKE* protocol.

Command with **no** prefix removes the selected group.

Prefix no Yes

Change settings Yes

Multiple input Yes

Synopsis

(config-ike-proposal)> **dh-group** *‹dh-group›*

(config-ike-proposal)> **no dh-group** *‹dh-group›*

Arguments

Argument	Value	Description
dh-group	1	*DH* group to work in the *PFS* mode.
	2	
	5	
	14	
	15	
	16	
	17	
	18	

Argument	Value	Description
	19	
	20	
	21	
	25	
	26	
	31	
	32	

Example

```
(config-ike-proposal)> dh-group 14
IpSec::Manager: "test": crypto ike proposal DH group "14" ►
successfully added.
```

```
(config-ike-proposal)> no dh-group 14
IpSec::Manager: "test": crypto ike proposal "test" group type ►
successfully removed.
```

History

Version	Description
2.06	The **crypto ike proposal dh-group** command has been introduced.

3.11.3 crypto ike proposal encryption

Description

Add the selected type of encryption to *IKE* proposal. The order of adding has a value for data exchange on the *IKE* protocol.

Command with **no** prefix removes the selected type of encryption.

Prefix no

Yes

Change settings

Yes

Multiple input

Yes

Synopsis

(config-ike-proposal)> **encryption** *‹encryption›*

(config-ike-proposal)> **no encryption** *‹encryption›*

Arguments

Argument	Value	Description
encryption	des	Type of *IKE* encryption.
	3des	
	aes-cbc-128	
	aes-cbc-192	
	aes-cbc-256	

Argument	Value	Description
	aes-ctr-128	
	aes-ctr-192	
	aes-ctr-256	

Example

```
(config-ike-proposal)> encryption des
IpSec::Manager: "test": crypto ike proposal encryption algorithm ►
"des" added.
```

```
(config-ike-proposal)> no encryption des
IpSec::Manager: "test": crypto ike proposal "test" encryption ►
type successfully removed.
```

History

Version	Description
2.06	The **crypto ike proposal encryption** command has been introduced.

3.11.4 crypto ike proposal integrity

Description

Add the selected value of *HMAC* signature algorithm to *IKE* proposal. The order of adding has a value for data exchange on the *IKE* protocol.

Command with **no** prefix removes the selected algorithm.

Prefix no

Yes

Change settings

Yes

Multiple input

Yes

Synopsis

(config-ike-proposal)> **integrity** ‹*integrity*›

(config-ike-proposal)> **no integrity** ‹*integrity*›

Arguments

Argument	Value	Description
integrity	md5	*HMAC* signature algorithm of *IKE* messages.
	sha1	
	sha256	
	sha384	
	sha512	

Example

```
(config-ike-proposal)> integrity sha256
IpSec::Manager: "test": crypto ike proposal integrity algorithm ►
"sha256" successfully added.
```

```
(config-ike-proposal)> no integrity sha256
IpSec::Manager: "test": crypto ike proposal "test" integrity ►
type successfully removed.
```

History

Version	Description
2.06	The **crypto ike proposal integrity** command has been introduced.

3.11.5 crypto ike proposal prf

Description

Add the selected *PRF* group to *IKE* proposal.

Command with **no** prefix removes the selected algorithm.

Prefix no

Yes

Change settings

Yes

Multiple input

Yes

Synopsis

(config-ike-proposal)> **prf** ‹*prf*›

(config-ike-proposal)> **no prf** ‹*prf*›

Arguments

Argument	Value	Description
prf	md5	*HMAC* signature algorithm of *IKE* messages.
	sha1	
	aes-xcbc	
	sha256	
	sha384	
	sha512	
	aes-cmac	

Example

```
(config-ike-proposal)> prf sha256
IpSec::Manager: "TEST": crypto ike proposal prf algorithm ►
"sha256" successfully added.
```

```
(config-ike-proposal)> no prf sha256
IpSec::Manager: "TEST": crypto ike proposal "TEST" prf type ►
successfully removed.
```

History

Version	Description
3.05	The **crypto ike proposal prf** command has been introduced.

3.12 crypto ipsec incompatible

Description Disable *IPsec* tunnels compatibility checking. By default, the setting is disabled.

Command with **no** prefix enables the checking back.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config)> **crypto ipsec incompatible**

(config)> **no crypto ipsec incompatible**

Example

```
(config)> crypto ipsec incompatible
IpSec::Manager: Compatibility checks is disabled.
```

```
(config)> no crypto ipsec incompatible
IpSec::Manager: Compatibility checks is enabled.
```

History

Version	Description
2.10	The **crypto ipsec incompatible** command has been introduced.

3.13 crypto ipsec mtu

Description Set *MTU* value to be transmitted to *IPsec*. By default, `auto` value is used.

Prefix no No

Change settings No

Multiple input No

Synopsis

(config)> **crypto ipsec mtu** (**auto** | *‹value›*)

Arguments

Argument	Value	Description
`auto`	*Keyword*	*MTU* will be assigned automatically.
`value`	*Integer*	*MTU* value. Can take values from 128 to 1500 inclusively.

Example

```
(config)> crypto ipsec mtu auto
IpSec::Manager: MTU is set to auto.
```

```
(config)> crypto ipsec mtu 1400
IpSec::Manager: Static MTU value is set to 1400.
```

History

Version	Description
2.08	The **crypto ipsec mtu** command has been introduced.

3.14 crypto ipsec profile

Description

Access to a group of commands to configure selected *IPsec* profile. If profile is not found, the command tries to create it.

Command with **no** prefix removes profile. At the same time references to this profile are automatically deleted from all *IPsec* crypto maps.

Prefix no

Yes

Change settings

Yes

Multiple input

Yes

Group entry

`(config-ipsec-profile)`

Synopsis

(config)> **crypto ipsec profile** *‹name›*

(config)> **no crypto ipsec profile** *‹name›*

Arguments

Argument	Value	Description
name	*String*	*IPsec* profile name. Latin letters, numbers, dots, hyphens and underscores are acceptable.

Example

```
(config)> crypto ipsec profile test
IpSec::Manager: "test": crypto ipsec profile successfully created.
```

```
(config)> no crypto ipsec profile test
IpSec::Manager: Crypto ipsec profile "test" removed.
```

History

Version	Description
2.06	The **crypto ipsec profile** command has been introduced.

3.14.1 crypto ipsec profile authentication-local

Description

Set authentication type for local host. By default, value `pre-share` is used.

Command with **no** prefix resets setting to default.

Prefix no

Yes

Change settings

Yes

Multiple input No

Synopsis

(config-ipsec-profile)> **authentication-local** *‹auth›*

(config-ipsec-profile)> **no authentication-local**

Arguments

Argument	Value	Description
auth	pre-share	A single available type of authorization for now.

Example

```
(config-ipsec-profile)> authentication-local pre-share
IpSec::Manager: "test": crypto ipsec profile authentication-local ►
type "pre-share" is set.
```

```
(config-ipsec-profile)> no authentication-local
IpSec::Manager: "test": crypto ipsec profile authentication-local ►
reset.
```

History

Version	Description
2.06	The **crypto ipsec profile authentication-local** command has been introduced.

3.14.2 crypto ipsec profile authentication-remote

Description Set authentication type for remote host. By default, value pre-share is used.

Command with **no** prefix resets setting to default.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config-ipsec-profile)> **authentication-remote** *‹auth›*

(config-ipsec-profile)> **no authentication-remote**

Arguments

Argument	Value	Description
auth	pre-share	A single available type of authorization for now.

Example

```
(config-ipsec-profile)> authentication-remote pre-share
IpSec::Manager: "test": crypto ipsec profile ►
authentication-remote type "pre-share" is set.
```

```
(config-ipsec-profile)> no authentication-remote
IpSec::Manager: "test": crypto ipsec profile ►
authentication-remote reset.
```

History

Version	Description
2.06	The **crypto ipsec profile authentication-remote** command has been introduced.

3.14.3 crypto ipsec profile dpd-clear

Description

Set method of action when detecting a dead *IKE* peer. By default, the setting is enabled, which means deleting peer information.

Command with **no** prefix set action to `restart`.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config-ipsec-profile)> **dpd-clear**

(config-ipsec-profile)> **no dpd-clear**

Example

```
(config-ipsec-profile)> dpd-clear
IpSec::Manager: "VPNL2TPServer": crypto ipsec profile DPD action ►
set to "clear".
```

```
(config-ipsec-profile)> no dpd-clear
IpSec::Manager: "VPNL2TPServer": crypto ipsec profile DPD action ►
set to "restart".
```

History

Version	Description
2.11	The **crypto ipsec profile dpd-clear** command has been introduced.

3.14.4 crypto ipsec profile dpd-interval

Description

Set parameters of method to detect a dead *IKE* peer. By default, `interval` is set to 30, `retry-count` is set to 3.

Command with **no** prefix resets settings to default.

Prefix no

Yes

Change settings

Yes

Multiple input No

Synopsis

(config-ipsec-profile)> **dpd-interval** ‹*interval*› [*retry-count*]

(config-ipsec-profile)> **no dpd-interval**

Arguments

Argument	Value	Description
`interval`	*Integer*	The interval of sending *DPD* packets in seconds. Can take values from 2 to 3600.
`retry-count`	*Integer*	Number of attempts to send *DPD* packets. Can take values from 3 to 60.

Example

```
(config-ipsec-profile)> dpd-interval 5 30
IpSec::Manager: "test": crypto ipsec profile dpd retry count is ►
set to 30.
```

```
(config-ipsec-profile)> no dpd-interval
IpSec::Manager: "test": crypto ipsec profile dpd retry count ►
reset.
```

History

Version	Description
2.06	The **crypto ipsec profile dpd-interval** command has been introduced.

3.14.5 crypto ipsec profile identity-local

Description Set a local identifier of *IPsec* profile.

Command with **no** prefix removes the local identifier.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config-ipsec-profile)> **identity-local** ‹*type*› ‹*id*›

(config-ipsec-profile)> **no identity-local**

Arguments

Argument	Value	Description
`type`	`address`	ID type is IP-address.
	`fqdn`	ID type is full domain name.
	`dn`	ID type is domain name.
	`email`	ID type is e-mail address.

Argument	Value	Description
id	*String*	Local ID value.

Example

```
(config-ipsec-profile)> identity-local address 10.10.10.5
IpSec::Manager: "test": crypto ipsec profile identity-local is ►
set to "10.10.10.5" with type "address".
```

```
(config-ipsec-profile)> no identity-local
IpSec::Manager: "test": crypto ipsec profile identity-local reset.
```

History

Version	Description
2.06	The **crypto ipsec profile identity-local** command has been introduced.

3.14.6 crypto ipsec profile match-identity-remote

Description

Set remote host identifier for *IPsec* profile.

Command with **no** prefix removes remote host ID.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config-ipsec-profile)> **match-identity-remote** (*‹type› ‹id›* | **any**)

(config-ipsec-profile)> **no match-identity-remote**

Arguments

Argument	Value	Description
type	address	ID type is IP-address.
	fqdn	ID type is full domain name.
	dn	ID type is domain name.
	email	ID type is e-mail address.
id	*String*	Remote host ID value.
any	*Keyword*	Allow usage of any remote host.

Example

```
(config-ipsec-profile)> match-identity-remote any
IpSec::Manager: "test": crypto ipsec profile ►
match-identity-remote is set to any.
```

```
(config-ipsec-profile)> no match-identity-remote
IpSec::Manager: "test": crypto ipsec profile ►
match-identity-remote reset.
```

History

Version	Description
2.06	The **crypto ipsec profile match-identity-remote** command has been introduced.

3.14.7 crypto ipsec profile mode

Description Set the mode of operation *IPsec*. By default, `tunnel` value is set.

Command with **no** prefix resets setting to default.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config-ipsec-profile)> **mode** *‹mode›*

(config-ipsec-profile)> **no mode**

Arguments

Argument	Value	Description
`mode`	`tunnel`	Tunnel mode, when the entire IP packet is encrypted and/or authenticated.
	`transport`	Transport mode, when only the payload of the IP packet is encrypted and/or authenticated.

Example

```
(config-ipsec-profile)> mode transport
IpSec::Manager: "test": crypto ipsec profile mode set to ►
"transport".
```

```
(config-ipsec-profile)> no mode
IpSec::Manager: "test": crypto ipsec profile mode reset.
```

History

Version	Description
2.06	The **crypto ipsec profile mode** command has been introduced.

3.14.8 crypto ipsec profile policy

Description Set the reference to existing *IKE* policy (see **crypto ike policy** command).

Command with **no** prefix removes the reference.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config-ipsec-profile)> **policy** *‹policy›*

(config-ipsec-profile)> **no policy**

Arguments

Argument	Value	Description
policy	*String*	*IKE* policy name. You can see the list of available policies with help of **policy** [Tab] command.

Example

```
(config-ipsec-profile)> policy [Tab]
 Usage template:
           policy {name: {A-Z, a-z, 0-9, ., _, -}}

   Choose:
 VirtualIPServer
   VPNL2TPServer
```

```
(config-ipsec-profile)> policy VirtualIPServer
IpSec::Manager: "TEST": crypto ipsec profile policy set to ►
"VirtualIPServer".
```

```
(config-ipsec-profile)> no policy
IpSec::Manager: "test": crypto ipsec profile policy reset.
```

History

Version	Description
2.06	The **crypto ipsec profile policy** command has been introduced.

3.14.9 crypto ipsec profile preshared-key

Description Set pre-shared key for *IPsec* profile.

Command with **no** prefix removes pre-shared key.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config-ipsec-profile)> **preshared-key** *‹preshare-key›*

(config-ipsec-profile)> **no preshared-key**

Arguments

Argument	Value	Description
preshare-key	*String*	Pre-shared key value.

Example

```
(config-ipsec-profile)> preshared-key testkey
IpSec::Manager: "test": crypto ipsec profile preshared key was ►
set.
```

```
(config-ipsec-profile)> no preshared-key
IpSec::Manager: "test": crypto ipsec profile preshared key reset.
```

History

Version	Description
2.06	The **crypto ipsec profile preshared-key** command has been introduced.

3.14.10 crypto ipsec profile xauth

Description

Enable additional authentication *XAuth* for IKEv1 mode. By default, function is disabled.

Command with **no** prefix disables additional authentication.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config-ipsec-profile)> **xauth** *‹type›*

(config-ipsec-profile)> **no xauth**

Arguments

Argument	Value	Description
type	client	Client mode.
	server	Server mode.

Example

```
(config-ipsec-profile)> xauth client
IpSec::Manager: "test": crypto ipsec profile xauth set to ►
"client".
```

```
(config-ipsec-profile)> no xauth
IpSec::Manager: "test": crypto ipsec profile xauth is disabled.
```

History

Version	Description
2.06	The **crypto ipsec profile xauth** command has been introduced.

3.14.11 crypto ipsec profile xauth-identity

Description

Set login for additional authentication *XAuth* in client mode.

Command with **no** prefix removes the login.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config-ipsec-profile)> **xauth-identity** ‹*identity*›

(config-ipsec-profile)> **no xauth-identity**

Arguments

Argument	Value	Description
identity	*String*	Login for *XAuth* client mode.

Example

```
(config-ipsec-profile)> xauth-identity ident
IpSec::Manager: "test": crypto ipsec profile xauth-identity is ►
set to "ident".
```

```
(config-ipsec-profile)> no xauth-identity
IpSec::Manager: "test": crypto ipsec profile xauth identity is ►
deleted.
```

History

Version	Description
2.06	The **crypto ipsec profile xauth-identity** command has been introduced.

3.14.12 crypto ipsec profile xauth-password

Description Set password for additional authentication *XAuth* in client mode.

Command with **no** prefix removes the password.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config-ipsec-profile)> **xauth-password** ‹*password*›

(config-ipsec-profile)> **no xauth-password**

Arguments

Argument	Value	Description
password	*String*	Password for *XAuth* client mode.

Example

```
(config-ipsec-profile)> xauth-password password
IpSec::Manager: "test": crypto ipsec profile xauth-password is ►
set.
```

```
(config-ipsec-profile)> no xauth-password
IpSec::Manager: "test": crypto ipsec profile xauth password is ►
deleted.
```

History

Version	Description
2.06	The **crypto ipsec profile xauth-password** command has been introduced.

3.15 crypto ipsec rekey delete-delay

Description

Set interval before removing the IKE SA after receiving the DELETE command from the remote side. By default, the 10 value is used.

Command with **no** prefix resets setting to default.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config)> **crypto ipsec rekey delete-delay** *‹delay›*

(config)> **no crypto ipsec rekey delete-delay**

Arguments

Argument	Value	Description
delay	*Integer*	Delay value in seconds. Can take value in the range from 1 till 60.

Example

```
(config)> crypto ipsec rekey delete-delay 1
IpSec::Manager: Rekey delete-delay value is set to 1.
```

```
(config)> no crypto ipsec rekey delete-delay
IpSec::Manager: Rekey delete-delay value is set to 10.
```

History

Version	Description
2.11	The **crypto ipsec rekey delete-delay** command has been introduced.

3.16 crypto ipsec rekey make-before

Description

Set the mode when new IKE SA creates before the breaking the old one. By default, the feature is disabled.

Command with **no** prefix disables the mode.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config)> **crypto ipsec rekey make-before**

(config)> **no crypto ipsec rekey make-before**

Example

```
(config)> crypto ipsec rekey make-before
IpSec::Manager: Enable make-before-brake scheme for IKEv2 rekey.
```

```
(config)> no crypto ipsec rekey make-before
IpSec::Manager: Disable make-before-brake scheme for IKEv2 rekey.
```

History

Version	Description
2.11	The **crypto ipsec rekey make-before** command has been introduced.

3.17 crypto ipsec transform-set

Description

Access to a group of commands to configure selected *IPsec ESP* transformation during Phase 2. If transformation is not found, the command tries to create it.

Command with **no** prefix removes transformation. At the same time references to this transformation are automatically deleted from all *IPsec* crypto maps.

Prefix no

Yes

Change settings

Yes

Multiple input

Yes

Group entry

`(config-ipsec-transform)`

Synopsis

(config)> **crypto ipsec transform-set** *‹name›*

(config)> **no crypto ipsec transform-set** *‹name›*

Arguments

Argument	Value	Description
name	*String*	*IPsec* transformation name. Latin letters, numbers, dots, hyphens and underscores are acceptable.

Example

```
(config)> crypto ipsec transform-set test
IpSec::Manager: "test": crypto ipsec transform-set successfully ►
created.
```

```
(config)> no crypto ipsec transform-set test
IpSec::Manager: Crypto ipsec transform-set "test" removed.
```

History

Version	Description
2.06	The **crypto ipsec transform-set** command has been introduced.

3.17.1 crypto ipsec transform-set aead

Description Enable *AEAD* cypher mode on *IPsec*.

Prefix no No

Change settings No

Multiple input No

Synopsis

(config-ipsec-transform)> **aead**

Example

```
(config-ipsec-transform)> dh-group 14
IpSec::Manager: "TEST": crypto ipsec transform-set "TEST" enabled ►
AEAD mode.
```

History

Version	Description
3.05	The **crypto ipsec transform-set aead** command has been introduced.

3.17.2 crypto ipsec transform-set cypher

Description Add the selected type of encryption to *IPsec* transformation. The order of adding has a value for data exchange on the *IKE* protocol.

Command with **no** prefix removes the selected type of encryption.

Prefix no Yes

Change settings Yes

Multiple input Yes

Synopsis

(config-ipsec-transform)> **cypher** *‹cypher›*

(config-ipsec-transform)> **no cypher** *‹cypher›*

Arguments

Argument	Value	Description
cypher	esp-des	Type of *IPsec ESP* encryption.
	esp-3des	
	esp-aes-128	
	esp-aes-192	
	esp-aes-256	

Example

```
(config-ipsec-transform)> cypher esp-3des
IpSec::Manager: "test": crypto ipsec transform-set cypher ►
"esp-3des" successfully added.
```

```
(config-ipsec-transform)> no cypher esp-3des
IpSec::Manager: "test": crypto ipsec transform-set "test" cypher ►
successfully removed.
```

History

Version	Description
2.06	The **crypto ipsec transform-set cypher** command has been introduced.

3.17.3 crypto ipsec transform-set dh-group

Description Add the selected *DH* group to *IPsec* transformation to work in the *PFS* mode. The order of adding has a value for data exchange on the *IKE* protocol.

Command with **no** prefix removes the selected group.

Prefix no Yes

Change settings Yes

Multiple input Yes

Synopsis

(config-ipsec-transform)> **dh-group** *‹dh-group›*

(config-ipsec-transform)> **no dh-group** *‹dh-group›*

Arguments

Argument	Value	Description
dh-group	1	*DH* group to work in the *PFS* mode.
	2	

Argument	Value	Description
	5	
	14	
	15	
	16	
	17	
	18	

Example

```
(config-ipsec-transform)> dh-group 14
IpSec::Manager: "test": crypto ipsec transform-set dh-group "14" ►
successfully added.
```

```
(config-ipsec-transform)> no dh-group 14
IpSec::Manager: "test": crypto ipsec transform-set "test" ►
dh-group successfully removed.
```

History

Version	Description
2.06	The **crypto ipsec transform-set dh-group** command has been introduced.

3.17.4 crypto ipsec transform-set hmac

Description

Add the selected value of *HMAC* signature algorithm to *IPsec* transformation. The order of adding has a value for data exchange on the *IKE* protocol.

Command with **no** prefix removes the selected algorithm.

Prefix no

Yes

Change settings

Yes

Multiple input

Yes

Synopsis

(config-ipsec-transform)> **hmac** *‹hmac›*

(config-ipsec-transform)> **no hmac** *‹hmac›*

Arguments

Argument	Value	Description
hmac	esp-md5-hmac	*HMAC* signature algorithm of *IPsec ESP* transformation.
	esp-sha1-hmac	
	esp-sha256-hmac	

Example

```
(config-ipsec-transform)> hmac esp-sha1-hmac
IpSec::Manager: "test": crypto ipsec transform-set hmac ►
"esp-sha1-hmac" successfully added.
```

```
(config-ipsec-transform)> no hmac esp-sha1-hmac
IpSec::Manager: "test": crypto ipsec transform-set "test" hmac ►
successfully removed.
```

History

Version	Description
2.06	The **crypto ipsec transform-set hmac** command has been introduced.

3.17.5 crypto ipsec transform-set lifetime

Description

Set lifetime of selected *IPsec* transformation. By default, the value `3600` is used.

Command with **no** prefix resets setting to default.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config-ipsec-transform)> **lifetime** ‹*lifetime*›

(config-ipsec-transform)> **no lifetime**

Arguments

Argument	Value	Description
`lifetime`	*Integer*	Lifetime of *IPsec* transformation in seconds. Can take values from 60 to 2147483647.

Example

```
(config-ipsec-transform)> lifetime 8640
IpSec::Manager: "test": crypto ipsec transform-set lifetime set ►
to 8640 s.
```

```
(config-ipsec-transform)> no lifetime
IpSec::Manager: "test": crypto ipsec transform-set lifetime reset.
```

History

Version	Description
2.06	The **crypto ipsec transform-set lifetime** command has been introduced.

3.18 crypto map

Description Access to a group of commands to configure selected *IPsec* crypto map. If crypto map is not found, the command tries to create it.

Command with **no** prefix removes crypto map.

Prefix no Yes

Change settings Yes

Multiple input Yes

Group entry `(config-crypto-map)`

Synopsis

(config)> **crypto map** *‹name›*

(config)> **no crypto map** *‹name›*

Arguments

Argument	Value	Description
name	*String*	*IPsec* crypto map name. Latin letters, numbers, dots, hyphens and underscores are acceptable.

Example

```
(config)> crypto map test
IpSec::Manager: "test": crypto map successfully created.
```

```
(config)> no crypto map test
IpSec::Manager: Crypto map profile "test" removed.
```

History

Version	Description
2.06	The **crypto map** command has been introduced.

3.18.1 crypto map connect

Description Enable automatic unconditional *IPsec* connection to the remote host. Setting has no meaning if basic remote host was set to any (see **crypto map set-peer** command). By default, setting is disabled and connection is established when attempting to transmit traffic through the *IPsec ESP* transformation.

Command with **no** prefix disables automatic unconditional connection.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config-crypto-map)> **connect**

(config-crypto-map)> **no connect**

Example

```
(config-crypto-map)> connect
IpSec::Manager: "test": crypto map autoconnect enabled.
```

```
(config-crypto-map)> no connect
IpSec::Manager: "test": crypto map autoconnect disabled.
```

History

Version	Description
2.06	The **crypto map connect** command has been introduced.

3.18.2 crypto map enable

Description

Enable selected *IPsec* crypto map. By default, setting is enabled.

Command with **no** prefix disables crypto map.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config-crypto-map)> **enable**

(config-crypto-map)> **no enable**

Example

```
(config-crypto-map)> enable
IpSec::Manager: "test": crypto map enabled.
```

```
(config-crypto-map)> no enable
IpSec::Manager: "test": crypto map disabled.
```

History

Version	Description
2.06	The **crypto map enable** command has been introduced.

3.18.3 crypto map fallback-check-interval

Description

Enable periodic checking of basic host availability and return to it in case of presence basic and backup remote hosts both. By default, setting is disabled.

Command with **no** prefix disables checking.

Prefix no

Yes

Change settings

Yes

Multiple input No

Synopsis

(config-crypto-map)> **fallback-check-interval** ‹*interval-value*›

(config-crypto-map)> **no fallback-check-interval**

Arguments

Argument	Value	Description
`interval-value`	*Integer*	Period of checking in seconds. Can take values from 60 to 86400.

Example

```
(config-crypto-map)> fallback-check-interval 120
IpSec::Manager: "test": crypto map fallback check interval is ►
set to 120.
```

```
(config-crypto-map)> no fallback-check-interval
IpSec::Manager: "test": crypto map fallback check interval is ►
cleared.
```

History

Version	Description
2.06	The **crypto map fallback-check-interval** command has been introduced.

3.18.4 crypto map force-encaps

Description Enforce the *ESP* packet wrapping mode in *UDP* to bypass the firewall and NAT.

Command with **no** prefix disables the mode.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config-crypto-map)> **force-encaps**

(config-crypto-map)> **no force-encaps**

Example

```
(config-crypto-map)> force-encaps
IpSec::Manager: "test": crypto map force ESP in UDP encapsulation ►
enabled.
```

```
(config-crypto-map)> no force-encaps
IpSec::Manager: "test": crypto map force ESP in UDP encapsulation ►
disabled.
```

History

Version	Description
2.08	The **crypto map force-encaps** command has been introduced.

3.18.5 crypto map l2tp-server dhcp route

Description

Assign a route which is transmitted in DHCP INFORM messages to the *L2TP*-server clients.

Command with **no** prefix cancels the specified route. If you use no arguments, the entire list of routes will be cleared.

Prefix no

Yes

Change settings

Yes

Multiple input

Yes

Synopsis

(config-crypto-map)> **l2tp-server dhcp route** *‹address› ‹mask›*

(config-crypto-map)> **no l2tp-server dhcp route** [*‹address› ‹mask›*]

Arguments

Argument	Value	Description
address	*IP-address*	Network client address.
mask	*IP-mask*	Network client mask. There are two ways to enter the mask: the canonical form (for example, 255.255.255.0) and the form of prefix bit length (for example, /24).

Example

```
(config-crypto-map)> l2tp-server dhcp route 192.168.2.0/24
IpSec::Manager: "VPNL2TPServer": crypto map L2TP/IPsec server ►
added DHCP INFORM route to 192.168.2.0/255.255.255.0.
```

```
(config-crypto-map)> l2tp-server no dhcp route
IpSec::Manager: "VPNL2TPServer": Cleared DHCP INFORM routes.
```

History

Version	Description
2.12	The **crypto map l2tp-server dhcp route** command has been introduced.

3.18.6 crypto map l2tp-server enable

Description

Enable *L2TP*-server on *IPsec* crypto map. By default, the setting is enabled.

Command with **no** prefix disables the setting.

Prefix no

Yes

Change settings Yes

Multiple input No

Synopsis

(config-crypto-map)> **l2tp-server enable**

(config-crypto-map)> **no l2tp-server enable**

Example

```
(config-crypto-map)> l2tp-server enable
IpSec::Manager: "VPNL2TPServer": crypto map L2TP/IPsec server ►
enabled.
```

```
(config-crypto-map)> no l2tp-server enable
IpSec::Manager: "VPNL2TPServer": crypto map L2TP/IPsec server ►
disabled.
```

History

Version	Description
2.11	The **crypto map l2tp-server enable** command has been introduced.

3.18.7 crypto map l2tp-server interface

Description Bind *L2TP*-server to the specified interface.

Command with **no** prefix unbinds the server.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config-crypto-map)> **l2tp-server interface** ‹*interface*›

(config-crypto-map)> **no l2tp-server interface**

Arguments

Argument	Value	Description
`interface`	*`Interface name`*	Full name or an alias of the interface. You can see the list of available interfaces with help of **l2tp-server interface** [Tab] command.

Example

```
(config-crypto-map)> l2tp-server interface [Tab]

 Usage template:
        interface {interface}

   Choose:
         GigabitEthernet1
```

```
                      ISP
 WifiMaster0/AccessPoint2
 WifiMaster1/AccessPoint1
 WifiMaster0/AccessPoint3
 WifiMaster0/AccessPoint0
              AccessPoint
 WifiMaster1/AccessPoint2
 WifiMaster0/AccessPoint1
               GuestWiFi
```

```
(config-crypto-map)> l2tp-server interface ISP
IpSec::Manager: "VPNL2TPServer": crypto map L2TP/IPsec server ►
is bound to ISP.
```

```
(config-crypto-map)> no l2tp-server interface ISP
IpSec::Manager: "VPNL2TPServer": crypto map L2TP/IPsec server ►
is unbound.
```

History

Version	Description
2.11	The **crypto map l2tp-server interface** command has been introduced.

3.18.8 crypto map l2tp-server ipv6cp

Description

Enable IPv6 support. DHCP IPv6 pools are created for each *L2TP*-server. By default, the setting is disabled.

Command with **no** prefix disables IPv6 support.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config-crypto-map)> **l2tp-server ipv6cp**

(config-crypto-map)> **no l2tp-server ipv6cp**

Example

```
(config-crypto-map)> l2tp-server ipv6cp
IpSec::Manager: "VPNL2TPServer": crypto map L2TP/IPsec server ►
IPv6CP is enabled.
```

```
(config-crypto-map)> no l2tp-server ipv6cp
IpSec::Manager: "VPNL2TPServer": crypto map L2TP/IPsec server ►
IPv6CP is disabled.
```

History

Version	Description
3.00	The **crypto map l2tp-server ipv6cp** command has been introduced.

3.18.9 crypto map l2tp-server lcp echo

Description Specify the testing rules of the *L2TP*-server connections with *LCP* echo tools.

Command with **no** prefix disables *LCP* echo.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config-crypto-map)> **l2tp-server lcp echo** ‹*interval*› ‹*count*›

(config-crypto-map)> **no l2tp-server lcp echo**

Arguments

Argument	Value	Description
`interval`	*Integer*	Interval between sending *LCP* echo, in seconds. If within the specified time interval there is no *LCP* echo request from the remote location, the same request will be sent there asking for response *LCP* reply.
`count`	*Integer*	The number of consecutive requests *LCP* echo sent, for which no response *LCP* reply was received. If count of *LCP* echo requests goes unanswered, the connection is terminated.

Example

```
(config-crypto-map)> l2tp-server lcp echo 5 3
IpSec::Manager: "VPNL2TPServer": crypto map L2TP/IPsec server ►
set LCP echo to "5" : "3".
```

```
(config-crypto-map)> no l2tp-server lcp echo
IpSec::Manager: "VPNL2TPServer": crypto map L2TP/IPsec server ►
LCP echo disabled.
```

History

Version	Description
2.11	The **crypto map l2tp-server lcp echo** command has been introduced.

3.18.10 crypto map l2tp-server mru

Description Set *MRU* value to be transmitted to *L2TP*-server. By default, `1200` value is used.

Command with **no** prefix resets value to default.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config-crypto-map)> **l2tp-server mru** *‹mru›*

(config-crypto-map)> **no l2tp-server mru**

Arguments

Argument	Value	Description
mru	*Integer*	*MRU* value. Can take values from 128 to 1500 inclusively.

Example

```
(config-crypto-map)> l2tp-server mru 1500
IpSec::Manager: "VPNL2TPServer": crypto map L2TP/IPsec server ►
set MRU to "1500".
```

```
(config-crypto-map)> no l2tp-server mru
IpSec::Manager: "VPNL2TPServer": crypto map L2TP/IPsec server ►
MRU reset to default.
```

History

Version	Description
2.11	The **crypto map l2tp-server mru** command has been introduced.

3.18.11 crypto map l2tp-server mtu

Description Set *MTU* value to be transmitted to *L2TP*-server. By default, 1400 value is used.

Command with **no** prefix resets value to default.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config-crypto-map)> **l2tp-server mtu** *‹mtu›*

(config-crypto-map)> **no l2tp-server mtu**

Arguments

Argument	Value	Description
mtu	*Integer*	*MTU* value. Can take values from 576 to 1500 inclusively.

Example

```
(config-crypto-map)> l2tp-server mtu 1400
IpSec::Manager: "VPNL2TPServer": crypto map L2TP/IPsec server ►
set MTU to "1400".
```

```
(config-crypto-map)> no l2tp-server mtu
IpSec::Manager: "VPNL2TPServer": crypto map L2TP/IPsec server ►
MTU reset to default.
```

History

Version	Description
2.11	The **crypto map l2tp-server mtu** command has been introduced.

3.18.12 crypto map l2tp-server multi-login

Description Allow connection to *L2TP*-server for multiple users from one account.

Command with **no** prefix disables the feature.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config-crypto-map)> **l2tp-server multi-login**

(config-crypto-map)> **no l2tp-server multi-login**

Example

```
(config-crypto-map)> l2tp-server multi-login
IpSec::Manager: "VPNL2TPServer": crypto map L2TP/IPsec server ►
multiple login is enabled.
```

```
(config-crypto-map)> no l2tp-server multi-login
IpSec::Manager: "VPNL2TPServer": crypto map L2TP/IPsec server ►
multiple login is disabled.
```

History

Version	Description
2.11	The **crypto map l2tp-server multi-login** command has been introduced.

3.18.13 crypto map l2tp-server nat

Description Enable translation of addresses for *L2TP*-server.

Command with **no** prefix disables the translation.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config-crypto-map)> **l2tp-server nat**

(config-crypto-map)> **no l2tp-server nat**

Example

```
(config-crypto-map)> l2tp-server nat
IpSec::Manager: "VPNL2TPServer": crypto map L2TP/IPsec server ►
SNAT is enabled.
```

```
(config-crypto-map)> no l2tp-server nat
IpSec::Manager: "VPNL2TPServer": crypto map L2TP/IPsec server ►
SNAT is disabled.
```

History

Version	Description
2.11	The **crypto map l2tp-server nat** command has been introduced.

3.18.14 crypto map l2tp-server range

Description

Assign a pool of addresses for the clients of *L2TP*-server. By default, size `100` is used.

Command with **no** prefix removes a pool.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config-crypto-map)> **l2tp-server range** *‹begin›* (*‹end›* | *‹size›*)

(config-crypto-map)> **no l2tp-server range**

Arguments

Argument	Value	Description
`begin`	*IP-address*	Start address of pool.
`end`	*IP-address*	End address of pool.
`size`	*Integer*	Pool size.

Example

```
(config-crypto-map)> l2tp-server range  172.16.2.33 172.16.2.38
IpSec::Manager: "VPNL2TPServer": crypto map L2TP/IPsec server ►
pool range set from "172.16.2.33" to "172.16.2.38".
```

```
(config-crypto-map)> l2tp-server range  172.16.2.33 100
IpSec::Manager: "VPNL2TPServer": crypto map L2TP/IPsec server ►
pool range set from "172.16.2.33" to "172.16.2.132".
```

```
(config-crypto-map)> no l2tp-server range
IpSec::Manager: "VPNL2TPServer": crypto map L2TP/IPsec server ►
pool range deleted.
```

History

Version	Description
2.11	The **crypto map l2tp-server range** command has been introduced.

3.18.15 crypto map l2tp-server static-ip

Description

Bind IP-address to the user. User account must have `ipsec-l2tp` tag.

Command with **no** prefix removes binding.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config-crypto-map)> **static-ip** *‹user› ‹address›*

(config-crypto-map)> **no static-ip** *‹user›*

Arguments

Argument	Value	Description
`user`	*String*	Username.
`address`	*IP-address*	IP-address to bind.

Example

```
(config-crypto-map)> l2tp-server static-ip admin 172.16.2.33
IpSec::Manager: "VPNL2TPServer": crypto map L2TP/IPsec server ►
static IP "172.16.2.33" assigned to user "admin".
```

```
(config-crypto-map)> no l2tp-server static-ip admin
IpSec::Manager: "VPNL2TPServer": crypto map L2TP/IPsec server ►
static IP removed for user "admin".
```

History

Version	Description
2.11	The **crypto map l2tp-server static-ip** command has been introduced.

3.18.16 crypto map match-address

Description

Set the reference to existing list of packet filtering rules (see **access-list** command). The first rule in the list will be used for *IPsec* Phase 2.

Command with **no** prefix removes the reference.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config-crypto-map)> **match-address** ‹*access-list*›

(config-crypto-map)> **no match-address**

Arguments

Argument	Value	Description
access-list	*String*	Packet filtering rules name. You can see available lists with help of **match-address** [Tab] command.

Example

```
(config-crypto-map)> match-address [Tab]

 Usage template:
    match-address {access-list}

   Choose:
 _WEBADMIN_GigabitEthernet0/Vlan4
                    _WEBADMIN_ISP
                   _WEBADMIN_Home
                _WEBADMIN_Bridge2
              _WEBADMIN_Wireguard2
```

```
(config-crypto-map)> match-address test
IpSec::Manager: "test": crypto map match-address set to "test".
```

```
(config-crypto-map)> no match-address
IpSec::Manager: "test": crypto map match-address reset.
```

History

Version	Description
2.06	The **crypto map match-address** command has been introduced.

3.18.17 crypto map nail-up

Description Enable automatic renegotiation of *IPsec ESP* transformations at their obsolescence. By default, setting is disabled.

Command with **no** prefix disables automatic renegotiation.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config-crypto-map)> **nail-up**

(config-crypto-map)> **no nail-up**

Example

```
(config-crypto-map)> nail-up
IpSec::Manager: "test": crypto map SA renegotiation enabled.
```

```
(config-crypto-map)> no nail-up
IpSec::Manager: "test": crypto map SA renegotiation disabled.
```

History

Version	Description
2.06	The **crypto map nail-up** command has been introduced.

3.18.18 crypto map priority

Description

Set priority for *IPsec* crypto map. By default, value `0` is used.

Command with **no** prefix resets value to default.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config-crypto-map)> **priority** ‹*priority*›

(config-crypto-map)> **no priority**

Arguments

Argument	Value	Description
`priority`	*Integer*	Priority value. Can take values in the range from 0 till 255 inclusively.

Example

```
(config-crypto-map)> priority 255
IpSec::Manager: "VPNL2TPServer": crypto map priority set to 255.
```

```
(config-crypto-map)> no priority
IpSec::Manager: "VPNL2TPServer": crypto map priority reset.
```

History

Version	Description
2.06	The **crypto map priority** command has been introduced.

3.18.19 crypto map reauth-passive

Description

Enable passive reauthentication of *IPsec* crypto map. By default, setting is disabled.

Command with **no** prefix disables passive reauthentication.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config-crypto-map)> **reauth-passive**

(config-crypto-map)> **no reauth-passive**

Example

```
(config-crypto-map)> reauth-passive
IpSec::Manager: "VPNL2TPServer": crypto map SA passive ►
reauthentication enabled.
```

```
(config-crypto-map)> no reauth-passive
IpSec::Manager: "VPNL2TPServer": crypto map SA passive ►
reauthentication disabled.
```

History

Version	Description
2.11	The **crypto map reauth-passive** command has been introduced.

3.18.20 crypto map set-peer

Description Set basic remote host for *IPsec* connection.

Command with **no** prefix removes the setting.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config-crypto-map)> **set-peer** *‹remote-ip›*

(config-crypto-map)> **no set-peer**

Arguments

Argument	Value	Description
`remote-ip`	*String*	IP-address or domain name of remote host.
	any	Accept any incoming connections.

Example

```
(config-crypto-map)> set-peer ipsec.test.com
IpSec::Manager: "test": crypto map primary remote peer is set ►
to "ipsec.test.com".
```

```
(config-crypto-map)> no set-peer
IpSec::Manager: "test": crypto map remote primary and fallback ►
peer reset.
```

History

Version	Description
2.06	The **crypto map set-peer** command has been introduced.

3.18.21 crypto map set-peer-fallback

Description

Set backup remote host for *IPsec* connection. This setting can be made after assignment of basic host (see **crypto map set-peer** command).

Command with **no** prefix removes the setting.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config-crypto-map)> **set-peer-fallback** ‹*remote-ip*›

(config-crypto-map)> **no set-peer-fallback**

Arguments

Argument	Value	Description
`remote-ip`	*String*	IP-address or domain name of remote host.

Example

```
(config-crypto-map)> set-peer-fallback test.com
IpSec::Manager: "test": crypto map fallback remote peer cannot ►
be set without primary peer.
```

```
(config-crypto-map)> no set-peer-fallback
IpSec::Manager: "test": crypto map fallback remote peer reset.
```

History

Version	Description
2.06	The **crypto map set-peer-fallback** command has been introduced.

3.18.22 crypto map set-profile

Description

Set the reference to existing *IPsec* profile (see **crypto ipsec profile** command).

Command with **no** prefix removes the reference.

Prefix no

Yes

Change settings

Yes

Multiple input No

Synopsis

(config-crypto-map)> **set-profile** *‹profile›*

(config-crypto-map)> **no set-profile**

Arguments

Argument	Value	Description
profile	*String*	*IPsec* profile name. You can see the list of available profiles with help of **set-profile** [Tab] command.

Example

```
(config-crypto-map)> set-profile [Tab]

 Usage template:
      set-profile {name: {A-Z, a-z, 0-9, ., _, -}}

   Choose:
            TEST
            MYMY
 VirtualIPServer
   VPNL2TPServer
```

```
(config-crypto-map)> set-profile test
IpSec::Manager: "test": crypto map ipsec profile is set to "test".
```

```
(config-crypto-map)> no set-profile
IpSec::Manager: "test": crypto map ipsec profile reset.
```

History

Version	Description
2.06	The **crypto map set-profile** command has been introduced.

3.18.23 crypto map set-tcpmss

Description Set the limit on the segment size of outgoing *TCP* sessions within *IPsec* tunnel. If the *MSS* value, which is transmitted in the header of SYN-packets, exceeds the specified limit, command changes it. Path MTU Discovery mode allows automatically identifiy *MSS* limit.

Command with **no** prefix removes all limits from *MSS*.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config-crypto-map)> **set-tcpmss** *‹mss-value›*

(config-crypto-map)> **no set-tcpmss**

Arguments

Argument	Value	Description
mss-value	*Integer*	*MSS* upper limit. Can take values from 576 to 1500.
	pmtu	Enable Path MTU Discovery mode.

Example

```
(config-crypto-map)> set-tcpmss 1280
IpSec::Manager: "test": crypto map tcpmss set to 1280.
```

```
(config-crypto-map)> no set-tcpmss
IpSec::Manager: "test": crypto map tcpmss reset.
```

History

Version	Description
2.06	The **crypto map set-tcpmss** command has been introduced.

3.18.24 crypto map set-transform

Description

Set the reference to existing *IPsec ESP* transformation (see **crypto ipsec transform-set** command).

Command with **no** prefix removes the reference.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config-crypto-map)> **set-transform** ‹*transform-set*›

(config-crypto-map)> **no set-transform**

Arguments

Argument	Value	Description
transform-set	*String*	*IPsec* transformation name. You can see the list of available transformations with help of **set-transform** [Tab] command.

Example

```
(config-crypto-map)> set-transform [Tab]
 Usage template:
    set-transform {name: {A-Z, a-z, 0-9, ., _, -}}

   Choose:
 VirtualIPServer
   VPNL2TPServer
```

```
(config-crypto-map)> set-transform test
IpSec::Manager: "test": crypto map ipsec transform-set is set ►
to "test".
```

```
(config-crypto-map)> no set-transform
IpSec::Manager: "test": crypto map ipsec transform-set reset.
```

History

Version	Description
2.06	The **crypto map set-transform** command has been introduced.

3.18.25 crypto map virtual-ip dhcp route

Description

Assign a route which is transmitted in DHCP INFORM messages to the Virtual IP server clients.

Command with **no** prefix deletes the specified route. If you use no arguments, the entire list of routes will be cleared.

Prefix no

Yes

Change settings

Yes

Multiple input

Yes

Synopsis

(config-crypto-map)> **virtual-ip dhcp route** ‹*address*› ‹*mask*›

(config-crypto-map)> **no virtual-ip dhcp route** [‹*address*› ‹*mask*›]

Arguments

Argument	Value	Description
address	*IP-address*	Network client address.
mask	*IP-mask*	Network client mask. There are two ways to enter the mask: the canonical form (for example, 255.255.255.0) and the form of prefix bit length (for example, /24).

Example

```
(config-crypto-map)> virtual-ip dhcp route 192.168.2.0/24
IpSec::ManagerVirtualIp: "VirtualIPServerIKE2": crypto map ►
Virtual IP server added DHCP INFORM route to ►
192.168.2.0/255.255.255.0.
```

```
(config-crypto-map)> no virtual-ip dhcp route 192.168.2.0/24
IpSec::ManagerVirtualIp: "VirtualIPServerIKE2": crypto map ►
Virtual IP server DHCP INFORM route to 192.168.2.0/255.255.255.0 ►
removed.
```

```
(config-crypto-map)> no virtual-ip dhcp route
IpSec::ManagerVirtualIp: "VirtualIPServerIKE2": crypto map ►
Virtual IP server DHCP INFORM routes cleared.
```

History

Version	Description
3.06	The **crypto map virtual-ip dhcp route** command has been introduced.

3.18.26 crypto map virtual-ip dns-server

Description

Set *DNS*-server issued to clients in Virtual IP server mode.

Command with **no** prefix deletes the address.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

```
(config-crypto-map)>  virtual-ip dns-server ‹address›
```

```
(config-crypto-map)> no virtual-ip dns-server
```

Arguments

Argument	Value	Description
address	*IP-address*	IP-address of *DNS*-server.

Example

```
(config-crypto-map)> virtual-ip dns-server 10.5.5.5
IpSec::Manager: "test": crypto map Virtual IP DNS server set to ►
"10.5.5.5".
```

```
(config-crypto-map)> no virtual-ip dns-server
IpSec::Manager: "test": crypto map Virtual IP DNS server deleted.
```

History

Version	Description
2.08	The **crypto map virtual-ip dns-server** command has been introduced.

3.18.27 crypto map virtual-ip enable

Description

Enable Virtual IP server mode, when clients receive addresses from a given range. The value of a remote subnet, specified in the corresponding access-list, will be ignored. By default, the setting is disabled.

Command with **no** prefix disables the setting.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config-crypto-map)> **virtual-ip enable**

(config-crypto-map)> **no virtual-ip enable**

Example

```
(config-crypto-map)> virtual-ip enable
IpSec::Manager: "test": crypto map Virtual IP mode enabled.
```

```
(config-crypto-map)> no virtual-ip enable
IpSec::Manager: "test": crypto map Virtual IP mode disabled.
```

History

Version	Description
2.08	The **crypto map virtual-ip enable** command has been introduced.

3.18.28 crypto map virtual-ip multi-login

Description

Allow connection to Virtual IP server for multiple users from one account.

Command with **no** prefix disables the feature.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config-crypto-map)> **virtual-ip multi-login**

(config-crypto-map)> **no virtual-ip multi-login**

Example

```
(config-crypto-map)> virtual-ip multi-login
IpSec::Manager: "VirtualIPServer": crypto map Virtual IP server ►
multiple login is enabled.
```

```
(config-crypto-map)> no  virtual-ip multi-login
IpSec::Manager: "VirtualIPServer": crypto map Virtual IP server ►
multiple login is disabled.
```

History

Version	Description
3.05	The **crypto map virtual-ip multi-login** command has been introduced.

3.18.29 crypto map virtual-ip nat

Description

Enable translation for remote network of Virtual IP extension server.

Command with **no** prefix removes the rule.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config-crypto-map)> **virtual-ip nat**

(config-crypto-map)> **no virtual-ip nat**

Example

```
(config-crypto-map)> virtual-ip nat
IpSec::Manager: "test": crypto map Virtual IP remote pool SNAT ►
is enabled.
```

```
(config-crypto-map)> no virtual-ip nat
IpSec::Manager: "test": crypto map Virtual IP remote pool SNAT ►
is disabled.
```

History

Version	Description
2.08	The **crypto map virtual-ip nat** command has been introduced.

3.18.30 crypto map virtual-ip range

Description Configure the range of addresses issued to clients in Virtual IP server mode.

Command with **no** prefix removes the range.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config-crypto-map)> **virtual-ip range** *‹begin›* (*‹end›* | *‹size›*)

(config-crypto-map)> **no virtual-ip range**

Arguments

Argument	Value	Description
begin	*IP-address*	The beginning of the address range.
end	*IP-address*	The end of the address range.
size	*Integer*	Address range size.

Example

```
(config-crypto-map)> virtual-ip range 10.5.0.0 20
IpSec::Manager: "test": crypto map Virtual IP pool range set ►
from "10.5.0.0" to "10.5.0.19" (CIDR 10.5.0.0/27).
```

```
(config-crypto-map)> no virtual-ip range
IpSec::Manager: "test": crypto map Virtual IP pool range deleted.
```

History

Version	Description
2.08	The **crypto map virtual-ip range** command has been introduced.

3.18.31 crypto map virtual-ip static-ip

Description Bind IP-address to the user. User account must have `ipsec-xauth` tag.

Command with **no** prefix removes binding.

Prefix no Yes

Change settings Yes

Multiple input Yes

Synopsis

(config-crypto-map)> **virtual-ip static-ip** *‹user› ‹address›*

(config-crypto-map)> **no virtual-ip static-ip** *‹user›*

Arguments

Argument	Value	Description
`user`	*String*	Username.
`address`	*IP-address*	IP-address to bind.

Example

```
(config-crypto-map)> virtual-ip static-ip admin 172.20.0.1
IpSec::ManagerVirtualIp: "VirtualIPServer": crypto map Virtual ►
IP server static address "172.20.0.1" assigned to user "admin".
```

```
(config-crypto-map)> no virtual-ip static-ip admin
IpSec::ManagerVirtualIp: "VirtualIPServer": crypto map Virtual ►
IP server static address removed for user "admin".
```

History

Version	Description
3.05	The **crypto map virtual-ip static-ip** command has been introduced.

3.19 dns-proxy

Description Access to a group of commands to manage DNS proxy service.

Prefix no No

Change settings No

Multiple input No

Group entry `(config-dnspx)`

Synopsis

(config)> **`dns-proxy`**

Example

```
(config)> dns-proxy
Core::Configurator: Done.
(config-dnspx)>
```

History

Version	Description
2.04	The **dns-proxy** command has been introduced.

3.19.1 dns-proxy https upstream

Description Add *DNS over HTTPS* server.

Command with **no** prefix removes the defined server from the list. If you use no argument, the entire list of servers will be cleared.

Prefix no Yes

Change settings Yes

Multiple input Yes

Synopsis

(config-dnspx)> **`https upstream`** *‹url›* [*‹format›*] [**`sni`** *‹hash›*] [**`on`** *‹interface›*]

(config-dnspx)> **`no https upstream`** [*‹url›*]

Arguments

Argument	Value	Description
`url`	*String*	Custom URL of DNS service.
`format`	`dnsm`	The format to represent DNS data.
	`json`	
`hash`	*String*	Hash TLS certificate.
`interface`	*Interface name*	Interface name to configure.

Example

```
(config-dnspx)>https upstream ►
https://cloudflare-dns.com/dns-query?ct=application/dns-json json
Dns::Secure::ManagerDoh: DNS-over-HTTPS name server ►
"https://cloudflare-dns.com/dns-query?ct=application/dns-json" ►
(json) added.
```

```
(config-dnspx)>https upstream https://dns.adguard.com/dns-query ►
dnsm
```

```
Dns::Secure::ManagerDoh: DNS-over-HTTPS name server ►
"https://dns.adguard.com/dns-query" (dnsm) added.
```

```
(config-dnspx)>https upstream https://dns.adguard.com/dns-query ►
dnsm on ISP
Dns::Secure::ManagerDoh: DNS-over-HTTPS name server ►
"https://dns.adguard.com/dns-query" (dnsm) added.
```

```
(config-dnspx)>no https upstream https://dns.adguard.com/dns-query
Dns::Secure::ManagerDoh: DNS-over-HTTPS name server ►
"https://dns.adguard.com/dns-query" deleted.
```

```
(config-dnspx)>no https upstream
Dns::Secure::ManagerDoh: DNS-over-HTTPS name servers cleared.
```

History

Version	Description
3.01	The **dns-proxy https upstream** command has been introduced.

3.19.2 dns-proxy intercept enable

Description Enable transit DNS requests interception. This feature is also enabled when the Internet filter is running. By default, the interception is disabled.

Command with **no** prefix disables the interception.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config-dnspx)> **intercept enable**

(config-dnspx)> **no intercept enable**

Example

```
(config-dnspx)> intercept enable
Dns::Filter::Interceptor: Enabled.
(config-dnspx)> no intercept enable
Dns::Filter::Interceptor: Disabled.
```

History

Version	Description
3.06	The **dns-proxy intercept enable** command has been introduced.

3.19.3 dns-proxy max-ttl

Description Set maximum TTL for DNS proxy cached entries.

Command with **no** prefix removes maximum TTL value.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config-dnspx)> **max-ttl** *‹max-ttl›*

(config-dnspx)> **no max-ttl**

Arguments

Argument	Value	Description
max-ttl	*Integer*	The maximum value of TTL. Can take values from 1 to 604800000 milliseconds (1 week).

Example

```
(config-dnspx)> max-ttl 10000
Dns::Proxy: Dns-proxy set max-ttl to 10000.
```

```
(config-dnspx)> no max-ttl
Dns::Proxy: Dns-proxy max-ttl cleared.
```

History

Version	Description
2.05	The **dns-proxy max-ttl** command has been introduced.

3.19.4 dns-proxy proceed

Description Set interval between concurrent requests, which is sent by DNS proxy to multiple DNS servers. By default, 500 value is used.

Command with **no** prefix resets proceed to default.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config-dnspx)> **proceed** *‹proceed›*

(config-dnspx)> **no proceed**

Arguments

Argument	Value	Description
proceed	*Integer*	The value of DNS proxy proceed in milliseconds. Can take values from 1 to 50000.

Example

```
(config-dnspx)> proceed 600
Dns::Proxy: Dns-proxy set 600 msec. proceed.
```

```
(config-dnspx)> no proceed
Dns::Proxy: Dns-proxy proceed timeout reset.
```

History

Version	Description
2.04	The **dns-proxy proceed** command has been introduced.

3.19.5 dns-proxy rebind-protect

Description

Enable protect against *DNS rebinding* attacks. By default, `auto` mode is used.

Command with **no** prefix disables protection.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config-dnspx)> **rebind-protect** (**auto** | **strict**)

(config-dnspx)> **no rebind-protect**

Arguments

Argument	Value	Description
`auto`	*Keyword*	Protect subnets for `private` interfaces.
`strict`	*Keyword*	Protect subnets from list IANA IPv4 Special-Purpose Address Registry[1].

Example

```
(config-dnspx)> rebind-protect auto
Dns::Manager: Enabled rebind protection.
(config-dnspx)> no rebind-protect
Dns::Manager: Disabled rebind protection.
```

History

Version	Description
3.04	The **dns-proxy rebind-protect** command has been introduced.

3.19.6 dns-proxy srr-reset

Description

Set DNS proxy send-response rating reset time. By default, value 600000 is used.

Command with **no** prefix resets time reset to default.

Prefix no

Yes

[1] https://www.iana.org/assignments/iana-ipv4-special-registry/iana-ipv4-special-registry.xhtml

Change settings Yes

Multiple input No

Synopsis

(config-dnspx)> **srr-reset** ‹*srr-reset*›

(config-dnspx)> **no srr-reset**

Arguments

Argument	Value	Description
srr-reset	*Integer*	The value of time reset in milliseconds. Can take values from 0 to 600000.

Example

```
(config-dnspx)> srr-reset 111
Dns::Manager: Set send-response rating reset time to 111 ms.
```

```
(config-dnspx)> no srr-reset
Dns::Manager: Reset send-response rating reset time to default.
```

History

Version	Description
2.12	The **dns-proxy srr-reset** command has been introduced.

3.19.7 dns-proxy tls upstream

Description Add *DNS over TLS* server.

Command with **no** prefix removes the defined server from the list. If you use no argument, the entire list of servers will be cleared.

Prefix no Yes

Change settings Yes

Multiple input Yes

Synopsis

(config-dnspx)> **tls upstream** ‹*address*› [‹*port*›] [**sni** ‹*fqdn*›] [**spki** ‹*hash*›] [**on** ‹*interface*›]

(config-dnspx)> **no tls upstream** [‹*address*›] [‹*port*›]

Arguments

Argument	Value	Description
address	*IP-address*	IP-address of the server.
port	*Integer*	The server port.
fqdn	*String*	Full domain name.
hash	*String*	Hash TLS certificate.
interface	*Interface name*	Interface name to configure.

Example

```
(config-dnspx)>tls upstream 1.1.1.1 853 sni cloudflare-dns.com
Dns::Secure::ManagerDot: DNS-over-TLS name server 1.1.1.1:853 ►
added.

(config-dnspx)>tls upstream 1.1.1.1 853 sni cloudflare-dns.com ►
on ISP
Dns::Secure::ManagerDot: DNS-over-TLS name server 1.1.1.1:853 ►
added.

(config-dnspx)>no tls upstream 1.1.1.1 853
Dns::Secure::ManagerDot: DNS-over-TLS name server 1.1.1.1:853 ►
deleted.

(config-dnspx)>no tls upstream
Dns::Secure::ManagerDot: DNS-over-TLS name servers cleared.
```

History

Version	Description
3.01	The **dns-proxy tls upstream** command has been introduced.

3.20 dpn accept

Description

Accept user agreement *DPN*. Until the license is accepted, the configurator does not accept any command except READ_ONLY.

Prefix no

No

Change settings

No

Multiple input

No

Synopsis

(config)> **dpn accept**

Example

```
(config)> dpn accept
Core::Legal: Accepted dpn version 20200330.
```

History

Version	Description
3.05	The **dpn accept** command has been introduced.

3.21 dyndns profile

Description

Access to a group of commands to configure DynDns profile. If the profile is not found, the command tries to create it. You can enter up to 32 profiles.

Command with **no** prefix removes DynDns profile.

Prefix no

Yes

Change settings Yes

Multiple input Yes

Group entry `(config-dyndns)`

Synopsis

(config)> **dyndns profile** *‹name›*

(config)> **no dyndns profile** *‹name›*

Arguments

Argument	Value	Description
name	*String*	The profile name. Maximum name length is 64 characters.

Example

```
(config)>  dyndns profile _WEBADMIN
Core::Configurator: Done.
(config-dyndns)>
```

History

Version	Description
2.00	The **dyndns profile** command has been introduced.

3.21.1 dyndns profile domain

Description Assign permanent domain name to the computer. You need to register this domain name on the site dyndns.com[2] or no-ip.com[3] before execution.

Command with **no** prefix removes the setting.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config-dyndns)> **domain** *‹domain›*

(config-dyndns)> **no domain**

Arguments

Argument	Value	Description
domain	*String*	The domain name. Maximum domain name length is 254 characters.

Example

```
(config-dyndns)> domain support.ddns.net
DynDns::Profile: "_WEBADMIN": domain saved..
```

[2] http:\\www.dyndns.com
[3] http:\\www.no-ip.com

```
(config-dyndns)> no domain
ynDns::Profile: "_WEBADMIN" domain cleared.
```

History

Version	Description
2.00	The **dyndns profile domain** command has been introduced.

3.21.2 dyndns profile password

Description Set password for access via DynDns.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config-dyndns)> **password** *‹password›*

(config-dyndns)> **no password**

Arguments

Argument	Value	Description
password	*String*	The password for authentication. Maximum password length is 64 characters.

Example

```
(config-dyndns)> password 123456789
DynDns::Profile: "_WEBADMIN": password saved.
```

```
(config-dyndns)> no password
DynDns::Profile: "_WEBADMIN" password cleared.
```

History

Version	Description
2.00	The **dyndns profile password** command has been introduced.

3.21.3 dyndns profile send-address

Description Enable the necessity of connection IP-address indication in DynDns request.

Command with **no** prefix removes the setting.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config-dyndns)> **send-address**

(config-dyndns)> **no send-address**

Example

```
(config-dyndns)> send-address
DynDns::Profile: Send address is enabled.
```

```
(config-dyndns)> no send-address
DynDns::Profile: Send address is disabled.
```

History

Version	Description
2.03	The **dyndns profile send-address** command has been introduced.

3.21.4 dyndns profile type

Description

Set DynDns type depending on the site where the domain name was registered.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config-dyndns)> **type** ‹*type*›

(config-dyndns)> **no type**

Arguments

Argument	Value	Description
`type`	`dyndns`	Used if the domain name was registered on the dyndns.com[4] site.
	`noip`	Used if the domain name was registered on the no-ip.com[5] site.
	`custom`	Used if the domain name was registered on the other site (defined with dyndns profile url command).

Example

```
(config-dyndns)> type noip
DynDns::Profile: "_WEBADMIN": type saved.
```

```
(config-dyndns)> no type
DynDns::Profile: "_WEBADMIN" type cleared.
```

[4] http://www.dyndns.com
[5] http://www.no-ip.com

History

Version	Description
2.00	The **dyndns profile type** command has been introduced.

3.21.5 dyndns profile update-interval

Description Set the address update interval for DynDns.

Command with **no** prefix cancels the ability to update.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config-dyndns)> **update-interval** *‹days›* **days [** *‹hours›* **hours] [** *‹minutes›* **minutes] [** *‹seconds›* **seconds]**

(config-dyndns)> **no update-interval**

Arguments

Argument	Value	Description
days	*Integer*	Interval time in days.
hours	*Integer*	Interval time in hours.
minutes	*Integer*	Interval time in minutes.
seconds	*Integer*	Interval time in seconds.

Example

```
(config-dyndns)> update-interval 5 days 5 hours 5 minutes 5 ► seconds
DynDns::Profile: Interval is set to 450305 seconds.
```

```
(config-dyndns)> update-interval 5 days
DynDns::Profile: Interval is set to 432000 seconds.
```

```
(config-dyndns)> no update-interval
DynDns::Profile: Periodic registration disabled.
```

History

Version	Description
2.03	The **dyndns profile update-interval** command has been introduced.

3.21.6 dyndns profile url

Description Set dynamic DNS service custom URL.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config-dyndns)> **url** *‹url›*

(config-dyndns)> **no url**

Arguments

Argument	Value	Description
url	*String*	Custom URL of DNS service.

Example

```
(config-dyndns)> url http://members.dyndns.org/nic/update
DynDns::Profile: "_WEBADMIN": URL saved.
```

```
(config-dyndns)> no url
DynDns::Profile: "_WEBADMIN" URL cleared.
```

History

Version	Description
2.05	The **dyndns profile url** command has been introduced.

3.21.7 dyndns profile username

Description Set username for access via DynDns.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config-dyndns)> **username** *‹username›*

(config-dyndns)> **no username**

Arguments

Argument	Value	Description
username	*String*	Username for authentication. Maximum name length is 64 characters.

Example

```
(config-dyndns)> username test@gmail.com
DynDns::Profile: "_WEBADMIN": username saved.
```

```
(config-dyndns)> no username
DynDns::Profile: "_WEBADMIN" username cleared.
```

History

Version	Description
2.00	The **dyndns profile username** command has been introduced.

3.22 easyconfig check

Description Access to a group of commands to configure Internet access check. To check Internet access, first requests to the default gateway are sent. If the answer is received, then the remote hosts specified in the settings are polled. The duration and frequency of requests are also specified in the settings. If all the checks have been passed, then the Internet access is provided.

Prefix no No

Change settings No

Multiple input No

Group entry `(ezconfig-check)`

Synopsis `(config)>` **easyconfig check**

Example

```
(config)> easyconfig check
(ezconfig-check)>
```

History

Version	Description
2.00	The **easyconfig check** command has been introduced.

3.22.1 easyconfig check exclude-gateway

Description Disable default gateway check. By default, the setting is enabled.

Command with **no** prefix enables the check back.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(ezconfig-check)> **exclude-gateway**

(ezconfig-check)> **no exclude-gateway**

Example

```
(ezconfig-check)> exclude-gateway
Network::InternetChecker: Gateway checking disabled.
```

```
(ezconfig-check)> no exclude-gateway
Network::InternetChecker: Gateway checking enabled.
```

History

Version	Description
2.05	The **easyconfig check exclude-gateway** command has been introduced.

3.22.2 easyconfig check host

Description

Specify the hostnames used to send requests for Internet access detection. By default, host address is `google.com`.

Command with **no** prefix resets hostnames to default.

Prefix no

Yes

Change settings

Yes

Multiple input

Yes

Synopsis

(ezconfig-check)> **host** *‹host›*

(ezconfig-check)> **no host** [*‹host›*]

Arguments

Argument	Value	Description
`host`	*`String`*	Remote host name.

Example

```
(ezconfig-check)> host google.com
Network::InternetChecker: "google.com" name added.
```

```
(ezconfig-check)> no host google.com
Network::InternetChecker: "google.com" name removed.
```

```
(ezconfig-check)> no host
Network::InternetChecker: Domain name set reset to default.
```

History

Version	Description
2.00	The **easyconfig check host** command has been introduced.

3.22.3 easyconfig check max-fails

Description

Specify the number of consecutive failed requests to the hostnames determined with **easyconfig check host** command. By default, value 3 is used.

Command with **no** prefix resets setting to default.

Prefix no

Yes

Change settings Yes

Multiple input No

Synopsis

(ezconfig-check)> **max-fails** *‹count›*

(ezconfig-check)> **no max-fails**

Arguments

Argument	Value	Description
count	*Integer*	Amount of failed requests. Can take values from 2 to 8 inclusively.

Example

```
(ezconfig-check)> max-fails 5
Network::InternetChecker: A new maximum fail count set to 5.
```

```
(ezconfig-check)> no max-fails
Network::InternetChecker: The maximum fail count reset to the ►
default value (3).
```

History

Version	Description
2.00	The **easyconfig check max-fails** command has been introduced.

3.22.4 easyconfig check period

Description Set a period of checking. By default, the value 15 is used.

Command with **no** prefix resets setting to default.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(ezconfig-check)> **period** *‹period›*

(ezconfig-check)> **no period**

Arguments

Argument	Value	Description
period	*Integer*	Check interval in seconds. Can take values in the range from 10 to 60 inclusively.

Example

```
(ezconfig-check)> period 20
Network::InternetChecker: A new check period set to 20 seconds.
```

```
(ezconfig-check)> no period
Network::InternetChecker: Check period reset to default (15 ►
seconds).
```

History

Version	Description
2.00	The **easyconfig check period** command has been introduced.

3.23 easyconfig disable

Description Disable initial setup wizard. By default, the setting is enabled.

Command with **no** prefix enables initial setup wizard.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config)> **easyconfig disable**

(config)> **no easyconfig disable**

Example

```
(config)> easyconfig disable
EasyConfig::Manager: Disabled.
```

```
(config)> no easyconfig disable
EasyConfig::Manager: Enabled.
```

History

Version	Description
3.01	The **easyconfig disable** command has been introduced.

3.24 eula accept

Description Accept user agreement *EULA*. Until the license is accepted, the configurator does not accept any command except READ_ONLY.

Prefix no No

Change settings No

Multiple input No

Synopsis

(config)> **eula accept**

Example

```
(config)> eula accept
Core::Eula: "20181001" license accepted.
```

History

Version	Description
2.15	The **eula accept** command has been introduced.

3.25 igmp-proxy

Description Access to a group of commands to configure *IGMP*.

Prefix no No

Change settings No

Multiple input No

Group entry `(igmp-proxy)`

Synopsis

(config)> **igmp-proxy**

Example

```
(config)> igmp-proxy
(igmp-proxy)>
```

History

Version	Description
2.06	The **igmp-proxy** command has been introduced.

3.25.1 igmp-proxy force

Description Force old version of *IGMP*. By default, the setting is disabled and the protocol version is selected in automatic mode.

Command with **no** prefix resets setting to default.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(igmp-proxy)> **force** ‹*protocol*›

(igmp-proxy)> **no force**

Arguments

Argument	Value	Description
`protocol`	`igmp-v1`	Apply filtering to incoming packets.
	`igmp-v2`	Apply filtering to outgoing packets.

Example

```
(igmp-proxy)> force igmp-v1
Igmp::Proxy: Forced protocol: igmp-v1.
```

```
(igmp-proxy)> no force
Igmp::Proxy: Enabled IGMP auto-detect.
```

History

Version	Description
2.08	The **igmp-proxy force** command has been introduced.

3.26 igmp-snooping disable

Description

Disable IGMP snooping. Command is available in Client, Repeater or AP modes only.

Command with **no** prefix enables IGMP snooping.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config)> **igmp-snooping disable**

Example

```
(config)> igmp-snooping disable
Igmp::Snooping: Disabled.
```

```
(config)> no igmp-snooping disable
Igmp::Snooping: Enabled.
```

History

Version	Description
2.12	The **igmp-snooping disable** command has been introduced.

3.27 interface

Description

Access to a group of commands to configure the selected interface. If the interface is not found, the command tries to create it.

The interface name specifies its class that inherits certain properties, see the diagrams in the Appendix. The commands work in relation to classes. The corresponding interface class is specified in the command description.

Command with **no** prefix deletes the interface.

Prefix no

Yes

Change settings

Yes

Multiple input Yes

Group entry `(config-if)`

Synopsis

(config)> **interface** *‹name›*

(config)> **no interface** *‹name›*

Arguments

Argument	Value	Description
name	*Interface name*	Full interface name or an alias. You can see the list of available interfaces with help of **interface** [Tab] command.

Example

```
(config)> interface [Tab]

Usage template:
        interface {name}

   Choose:
                      Pvc
                     Vlan
              CdcEthernet
                 UsbModem
           RealtekEthernet
              AsixEthernet
                   Davicom
                    UsbLte
                      Yota
                    Bridge
                     PPPoE
                      SSTP
                      PPTP
                      L2TP
                 Wireguard
                   OpenVPN
                      IPIP
           TunnelSixInFour
                       Gre
                      EoIP
           TunnelSixToFour
                    Chilli
```

History

Version	Description
2.00	The **interface** command has been introduced.

3.27.1 interface authentication chap

Description Enable *CHAP* authentication support.

Command with **no** prefix disables *CHAP*.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type Secure

Synopsis

(config-if)> **authentication chap**

(config-if)> **no authentication chap**

Example

```
(config-if)> authentication chap
Network::Interface::Supplicant: "PPTP0": added authentication: ►
CHAP.
```

```
(config-if)> no authentication chap
Network::Interface::Supplicant: "PPTP0": removed authentication: ►
CHAP.
```

History

Version	Description
2.00	The **interface authentication chap** command has been introduced.

3.27.2 interface authentication eap-md5

Description Enable EAP-MD5 authentication support.

Command with **no** prefix disables EAP-MD5.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type Secure

Synopsis

(config-if)> **authentication eap-md5**

(config-if)> **no authentication eap-md5**

Example

```
(config-if)> authentication eap-md5
Network::Interface::Ethernet: "GigabitEthernet1": configured ►
authentication: EAP-MD5.
```

```
(config-if)> no authentication eap-md5
Network::Interface::Supplicant: "GigabitEthernet1": removed ►
authentication: EAP-MD5.
```

History

Version	Description
2.00	The **interface authentication eap-md5** command has been introduced.

3.27.3 interface authentication eap-mschapv2

Description Enable EAP-MSCHAPv2 authentication support.

Command with **no** prefix disables EAP-MSCHAPv2, MS-CHAPv2.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type Secure

Synopsis

(config-if)> **authentication eap-mschapv2**

(config-if)> **no authentication eap-mschapv2**

Example

```
(config-if)> authentication eap-mschapv2
Network::Interface::Supplicant: "IKE0": authentication is ►
unchanged.
```

```
(config-if)> no authentication eap-mschapv2
Network::Interface::Supplicant: "IKE0": removed authentication: ►
EAP-MSCHAPv2, MS-CHAPv2.
```

History

Version	Description
3.05	The **interface authentication eap-mschapv2** command has been introduced.

3.27.4 interface authentication eap-ttls

Description Enable EAP-TTLS authentication support.

Command with **no** prefix disables EAP-TTLS.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type Secure

Synopsis

(config-if)> **authentication eap-ttls**

(config-if)> **no authentication eap-ttls**

Example

```
(config-if)> authentication eap-ttls
Network::Interface::Ethernet: "GigabitEthernet1": configured ►
authentication: EAP-TTLS.
```

```
(config-if)> no authentication eap-ttls
Network::Interface::Supplicant: "GigabitEthernet1": removed ►
authentication: EAP-TTLS.
```

History

Version	Description
2.00	The **interface authentication eap-ttls** command has been introduced.

3.27.5 interface authentication identity

Description

Specify user name for device authentication on the remote system. Equally often used on PPTP, PPPoE and L2TP connections.

Command with **no** prefix deletes the previously specified user name.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

Secure

Synopsis

(config-if)> **authentication identity** ‹*identity*›

(config-if)> **no authentication identity**

Arguments

Argument	Value	Description
identity	*String*	User name for authentication.

Example

```
(config-if)>  authentication identity mylogin
Network::Interface::Supplicant: "PPTP0": identity saved.
```

```
(config-if)> no authentication identity
Network::Interface::Supplicant: "PPTP0": identity cleared.
```

History

Version	Description
2.00	The **interface authentication identity** command has been introduced.

3.27.6 interface authentication mschap

Description Enable MS-CHAP authentication support.

Command with **no** prefix disables MS-CHAP.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type Secure

Synopsis

(config-if)> **authentication mschap**

(config-if)> **no authentication mschap**

Example

```
(config-if)> authentication mschap
Network::Interface::Supplicant: "PPTP0": added authentication: ►
MS-CHAP.
```

```
(config-if)> no authentication mschap
Network::Interface::Supplicant: "PPTP0": removed authentication: ►
MS-CHAP.
```

History

Version	Description
2.00	The **interface authentication mschap** command has been introduced.

3.27.7 interface authentication mschap-v2

Description Enable MS-CHAPv2 authentication support.

Command with **no** prefix disables MS-CHAPv2.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type Secure

Synopsis

(config-if)> **authentication mschap-v2**

(config-if)> **no authentication mschap-v2**

Example

```
(config-if)> authentication mschap-v2
Network::Interface::Supplicant: "PPTP0": authnentication is ►
unchanged.
```

```
(config-if)> no authentication mschap-v2
Network::Interface::Supplicant: "PPTP0": removed authentication: ►
MS-CHAPv2.
```

History

Version	Description
2.00	The **interface authentication mschap-v2** command has been introduced.

3.27.8 interface authentication pap

Description Enable *PAP* authentication support.

Command with **no** prefix disables *PAP*.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type Secure

Synopsis

(config-if)> **authentication pap**

(config-if)> **no authentication pap**

Example

```
(config-if)> authentication pap
Network::Interface::Supplicant: "PPTP0": added authentication: ►
PAP.
```

```
(config-if)> no authentication pap
Network::Interface::Supplicant: "PPTP0": removed authentication: ►
PAP.
```

History

Version	Description
2.00	The **interface authentication pap** command has been introduced.

3.27.9 interface authentication password

Description Specify password for device authentication on the remote system. Equally often used on PPTP, PPPoE and L2TP connections.

Command with **no** prefix deletes the password.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type Secure

Synopsis

(config-if)> **authentication password** *‹password›*

(config-if)> **no authentication password**

Arguments

Argument	Value	Description
password	*String*	Password for authentication.

Example

```
(config-if)> authentication password Aihoi2cha1
Network::Interface::Supplicant: "PPTP0": password saved.
```

```
(config-if)> no authentication password
Network::Interface::Supplicant: "PPTP0": password cleared.
```

History

Version	Description
2.00	The **interface authentication password** command has been introduced.

3.27.10 interface authentication peap

Description Enable *EAP-PEAP* authentication support.

Command with **no** prefix disables *EAP-PEAP*.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type Secure

Synopsis

(config-if)> **authentication peap**

(config-if)> **no authentication peap**

Example

```
(config-if)> authentication peap
Network::Interface::Ethernet: "WifiMaster1/AccessPoint0": ►
configured authentication: PEAP.
```

```
(config-if)> no authentication peap
Network::Interface::Supplicant: "WifiMaster1/AccessPoint0": ►
removed authentication: PEAP.
```

History

Version	Description
2.03	The **interface authentication peap** command has been introduced.

3.27.11 interface authentication shared

Description

Enable authentication with a *shared key*. This mode is used only in conjunction with *WEP* encryption. *Shared keys* are specified by **interface encryption key** command.

Command with **no** prefix turns authentication to open mode.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

WiFi

Synopsis

(config-if)> **authentication shared**

(config-if)> **no authentication shared**

Example

```
(config-if)> authentication shared
Network::Interface::Rtx::AccessPoint: "WifiMaster1/AccessPoint0": ►
shared authentication mode enabled.
```

```
(config-if)> no authentication shared
Network::Interface::Rtx::AccessPoint: "WifiMaster1/AccessPoint0": ►
shared authentication mode disabled.
```

History

Version	Description
2.00	The **interface authentication shared** command has been introduced.

3.27.12 interface authentication wpa-psk

Description

Specify the pre-agreed key for authentication via WPA-PSK protocol. It is possible to specify the key as a 256-bit hexadecimal number or as a string of ASCII-characters. In the second case, the string is used as a code phrase to generate the key (passphrase).

Command with **no** prefix removes setting.

Prefix no

Yes

Change settings

Yes

Multiple input No

Interface type WiFi

Synopsis

(config-if)> **authentication wpa-psk** *‹psk›*

(config-if)> **no authentication wpa-psk**

Arguments

Argument	Value	Description
psk	*String*	Pre-agreed key in the form of a 256-bit hexadecimal number, which consists of 64 hexadecimal digits, or in the form of ASCII string of 8 to 63 characters length.

Example

```
(config-if)> authentication wpa-psk Eethaich9z
Network::Interface::Wifi: "WifiMaster1/AccessPoint0": WPA PSK set.
```

```
(config-if)> no authentication wpa-psk
Network::Interface::Wifi: "WifiMaster1/AccessPoint0": WPA PSK ►
removed.
```

History

Version	Description
2.00	The **interface authentication wpa-psk** command has been introduced.

3.27.13 interface backhaul

Description Enable support of *VLAN* for wireless connection between routers Keenetic in the `trunk` mode. By default, setting is disabled.

Command with **no** prefix disables the setting.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type WiFiMaster

Synopsis

(config-if)> **backhaul**

(config-if)> **no backhaul**

Example

```
(config-if)> backhaul
Network::Interface::Rtx::AccessPoint: "WifiMaster0/AccessPoint1": ►
backhaul mode enabled.
```

```
(config-if)> no backhaul
Network::Interface::Rtx::AccessPoint: "WifiMaster0/AccessPoint1": ►
backhaul mode disabled.
```

History

Version	Description
3.02	The **interface backhaul** command has been introduced.

3.27.14 interface band-steering

Description

Enable *Band Steering* for AP 5 GHz. By default, the setting is enabled.

For correct *Band Steering* operation it is necessary to fulfill the following conditions:

- access points 2,4 GHz and 5 GHz are enabled both
- they have the same SSID's
- they have the same security settings (encryption type, key value, etc.)

Command with **no** prefix disables the *Band Steering*.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

WiFiMaster

Synopsis

(config-if)> **band-steering**

(config-if)> **no band-steering**

Example

```
(config-if)> band-steering
Network::Interface::Rtx::WifiMaster: "WifiMaster1": band steering ►
enabled.
```

```
(config-if)> no band-steering
Network::Interface::Rtx::WifiMaster: "WifiMaster1": band steering ►
disabled.
```

History

Version	Description
2.09	The **interface band-steering** command has been introduced.

3.27.15 interface band-steering preference

Description

Set the band to give a preference in *Band Steering* technology. By default, the value is not defined.

Command with **no** prefix resets setting to default.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

WiFiMaster

Synopsis

```
(config-if)> band-steering preference ‹band›
```

```
(config-if)> no band-steering preference
```

Arguments

Argument	Value	Description
band	2	2,4 GHz band.
	5	5 GHz band.

Example

```
(config-if)> band-steering preference 5
Network::Interface::Rtx::WifiMaster: "WifiMaster1": band steering ►
preference is 5 GHz.
```

```
(config-if)> no band-steering preference
Network::Interface::Rtx::WifiMaster: "WifiMaster1": band steering ►
preference disabled.
```

History

Version	Description
2.09	The **interface band-steering preference** command has been introduced.

3.27.16 interface ccp

Description

Enable *CCP* support during establishing connection.

Command with **no** prefix disables *CCP*.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

PPP

Synopsis

(config-if)> **ccp**

(config-if)> **no ccp**

Example

```
(config-if)> ccp
CCP enabled.
```

```
(config-if)> no ccp
CCP disabled.
```

History

Version	Description
2.00	The **interface ccp** command has been introduced.

3.27.17 interface channel

Description

Set the radio channel (broadcasting frequency band) for wireless interfaces. Wi-Fi interfaces take integers from 1 to 14 (frequency range from 2.412 GHz to 2.484 GHz) and from 36 to 165 (frequency range from 5.180 GHz to 5.825 GHz) as channel numbers. By default, `auto` value is used.

Command with **no** prefix resets to default.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

Radio

Synopsis

(config-if)> **channel** *‹channel›*

(config-if)> **no channel**

Arguments

Argument	Value	Description
`channel`	`number`	Number of radio channel.
	`auto`	Radio channel number is detected automatically.

Example

```
(config-if)> channel 8
Network::Interface::Rtx::WifiMaster: "WifiMaster0": channel set ►
to 8.
```

```
(config-if)> channel 36
Network::Interface::Rtx::WifiMaster: "WifiMaster1": channel set ►
to 36.
```

```
(config-if)> no channel
Network::Interface::Rtx::WifiMaster: "WifiMaster0": auto channel ►
mode set.
```

History

Version	Description
2.00	The **interface channel** command has been introduced.

3.27.18 interface channel auto-rescan

Description

Set a schedule for radio channel automatic scanning. By default, the setting is disabled.

Command with **no** prefix disables the setting.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

Radio

Synopsis

(config-if)> **channel auto-rescan** [‹*hh*›:‹*mm*›]**interval** ‹*interval*›

(config-if)> **no channel auto-rescan**

Arguments

Argument	Value	Description
interval	1	Rescan interval in hours.
	6	
	12	
	24	

Example

```
(config-if)> channel auto-rescan interval 1
Network::Interface::Rtx::WifiMaster: "WifiMaster0": scheduled ►
auto rescan, interval 1 hour.
```

```
(config-if)> no channel auto-rescan
Network::Interface::Rtx::WifiMaster: "WifiMaster0": auto rescan ►
disabled.
```

History

Version	Description
2.07	The **interface channel auto-rescan** command has been introduced.

3.27.19 interface channel width

Description Set the bandwidth for a specified channel. By default, `40-below` value is used.

Command with **no** prefix resets to default.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type Radio

Synopsis

(config-if)> **channel width** *‹width›*

(config-if)> **no channel width**

Arguments

Argument	Value	Description
`width`	`20`	Set bandwidth equal to 20 MHz.
	`40-above`	Expand the bandwidth up to 40 MHz using next channel.
	`40-below`	Expand the bandwidth up to 40 MHz using previous channel.

Example

```
(config-if)> channel width 20
Network::Interface::Rtx::WifiMaster: "WifiMaster0": channel ►
bandwidth setting applied.
```

```
(config-if)> no channel width
Network::Interface::Rtx::WifiMaster: "WifiMaster0": channel ►
bandwidth settings reset to default.
```

History

Version	Description
2.04	The **interface channel width** command has been introduced.

3.27.20 interface chilli coaport

Description Set *UDP* port to which disconnect requests from the *RADIUS* client are sent.

Command with **no** prefix removes the setting.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type Chilli

Synopsis

(config-if)> **chilli coaport** *‹coaport›*

(config-if)> **no chilli coaport**

Arguments

Argument	Value	Description
coaport	*Integer*	The *CoA* port number.

Example

```
(config-if)> chilli coaport 3940
Chilli::Interface: "Chilli0": coaport set to 3940.
```

```
(config-if)> no chilli coaport
Chilli::Interface: "Chilli0": coaport reset to default.
```

History

Version	Description
2.10	The **interface chilli coaport** command has been introduced.

3.27.21 interface chilli dhcpif

Description Assign Chilli interface to the system network interface.

Command with **no** prefix cancels the association.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type Chilli

Synopsis

(config-if)> **chilli dhcpif** *‹dhcpif›*

(config-if)> **no chilli dhcpif**

Arguments

Argument	Value	Description
dhcpif	*Interface name*	Full interface name or an alias.

Example

```
(config-if)> chilli dhcpif Bridge1
Chilli::Interface: "Chilli0": bound to Bridge1.
```

```
(config-if)> no chilli dhcpif
Chilli::Interface: "Chilli0": unbound.
```

History

Version	Description
2.10	The **interface chilli dhcpif** command has been introduced.

3.27.22 interface chilli dns

Description Set IP-address of the DNS-server.

Command with **no** prefix removes the setting.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type Chilli

Synopsis

(config-if)> **chilli dns** *‹dns1›* [*‹dns2›*]

(config-if)> **no chilli dns**

Arguments

Argument	Value	Description
dns1	*IP-address*	Address of primary DNS-server.
dns2	*IP-address*	Address of secondary DNS-server.

Example

```
(config-if)> chilli dns 8.8.8.8 1.1.1.1
Chilli::Interface: "Chilli0": DNS servers set to 8.8.8.8, 1.1.1.1.
```

```
(config-if)> no chilli dns
Chilli::Interface: "Chilli0": DNS servers reset to default.
```

History

Version	Description
2.10	The **interface chilli dns** command has been introduced.

3.27.23 interface chilli lease

Description Configure the lease time of the connected client IP-addresses. By default, the value 3600 is used.

Command with **no** prefix resets setting to default.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type Chilli

Synopsis

(config-if)> **chilli lease** *‹lease›*

(config-if)> **no chilli lease**

Arguments

Argument	Value	Description
lease	*Integer*	Lease time in seconds. The maximum value is 259200.

Example

```
(config-if)> chilli lease 1000
Chilli::Interface: "Chilli0": lease has been set 1000 seconds.
```

```
(config-if)> no chilli lease
Chilli::Interface: "Chilli0": lease has been reset to default ►
(3600 seconds).
```

History

Version	Description
2.11	The **interface chilli lease** command has been introduced.

3.27.24 interface chilli logout

Description Force the MAC-address of the specified client to be disabled.

Prefix no No

Change settings No

Multiple input No

Interface type Chilli

Synopsis

(config-if)> **chilli logout** (*‹mac›* | **all**)

Arguments

Argument	Value	Description
mac	*MAC-address*	MAC-address of the registered client.
all	Keyword	Disable all MAC-addresses.

Example

```
(config-if)> chilli logout 64:a2:22:51:b4:11
```

```
(config-if)> chilli logout all
Chilli::Interface: "Chilli0": service restarted.
```

History

Version	Description
2.10	The **interface chilli logout** command has been introduced.

3.27.25 interface chilli macauth

Description Enable user authentication option based on MAC-address detection only.

Command with **no** prefix disables the setting.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type Chilli

Synopsis

(config-if)> **chilli macauth**

(config-if)> **no chilli macauth**

Example

```
(config-if)>  chilli macauth
Chilli::Interface: "Chilli0": macauth set to "".
```

```
(config-if)> no chilli macauth
Chilli::Interface: "Chilli0": macauth cleared.
```

History

Version	Description
2.10	The **interface chilli macauth** command has been introduced.

3.27.26 interface chilli macpasswd

Description Set the password for MAC-address authentication.

Command with **no** prefix removes the setting.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type Chilli

Synopsis

(config-if)> **chilli macpasswd** *‹macpasswd›*

(config-if)> **no chilli macpasswd**

Arguments

Argument	Value	Description
macpasswd	*String*	The user password.

Example

```
(config-if)> chilli macpasswd 1234567890
Chilli::Interface: "Chilli0": macpasswd set to "1234567890".
```

```
(config-if)> no chilli macpasswd
Chilli::Interface: "Chilli0": macpasswd cleared.
```

History

Version	Description
2.11	The **interface chilli macpasswd** command has been introduced.

3.27.27 interface chilli nasip

Description

Set *RADIUS* option NAS IP Address. Allows you to configure and use an arbitrary IP-address.

Command with **no** prefix removes the setting.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

Chilli

Synopsis

(config-if)> **chilli nasip** (‹*address*› | **interface** ‹*wan*› | **auto**)

(config-if)> **no chilli nasip**

Arguments

Argument	Value	Description
address	*IP-address*	Specific IP-address of the server.
wan	*Interface name*	IP-address from the specified WAN interface.
auto	*Keyword*	IP-address from the current WAN interface.

Example

```
(config-if)> chilli nasip 95.213.215.187
Chilli::Interface: "Chilli0": NAS IP address set to ►
"95.213.215.187".
```

```
(config-if)> chilli nasip interface ISP
Chilli::Interface: "Chilli0": NAS IP interface set to ►
"GigabitEthernet1".
```

```
(config-if)> chilli nasip auto
Chilli::Interface: "Chilli0": NAS IP address set to auto.
```

```
(config-if)> no chilli nasip
Chilli::Interface: "Chilli0": NAS IP address cleared.
```

History

Version	Description
2.10	The **interface chilli nasip** command has been introduced.

3.27.28 interface chilli nasmac

Description

Set MAC-address for *RADIUS* Called-Station-ID attribute. By default, MAC-address of the guest network is used.

Command with **no** prefix resets setting to default.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

Chilli

Synopsis

(config-if)> **chilli nasmac** *‹mac›*

(config-if)> **no chilli nasmac**

Arguments

Argument	Value	Description
mac	*MAC-address*	New MAC-address for RADIUS Called-Station-ID.

Example

```
(config-if)> chilli nasmac  50:ff:20:00:1e:86
Chilli::Interface: "Chilli0": NAS MAC address set to ►
"50:ff:20:00:1e:86".
```

```
(config-if)> no chilli nasmac
Chilli::Interface: "Chilli0": NAS MAC address cleared.
```

History

Version	Description
2.11	The **interface chilli nasmac** command has been introduced.

3.27.29 interface chilli profile

Description

Assign Chilli profile to the Chilli interface.

Command with **no** prefix removes the setting.

Prefix no

Yes

Change settings Yes

Multiple input No

Interface type Chilli

Synopsis

(config-if)> **chilli profile** *‹profile›*

(config-if)> **no chilli profile**

Arguments

Argument	Value	Description
profile	*String*	*RADIUS* server profile name.

Example

```
(config-if)> chilli profile Wi-Fi_SYSTEM
Chilli::Interface: "Chilli0": assigned profile: Wi-Fi.
```

```
(config-if)> no chilli profile
Chilli::Interface: "Chilli0": profile cleared.
```

History

Version	Description
2.10	The **interface chilli profile** command has been introduced.

3.27.30 interface chilli radius

Description Add the *RADIUS* server addresses.

Command with **no** prefix removes the servers.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type Chilli

Synopsis

(config-if)> **chilli radius** *‹server1›* [*‹server2›*]

(config-if)> **no chilli radius**

Arguments

Argument	Value	Description
server1	*String*	Address of first *RADIUS* server.
server2	*String*	Address of second *RADIUS* server.

Example

```
(config-if)> chilli radius radius.example.net radius2.example.net
Chilli::Interface: "Chilli0": RADIUS servers set to ►
radius.example.net, radius2.example.net.
```

```
(config-if)> no chilli radius
Chilli::Interface: "Chilli0": RADIUS servers cleared.
```

History

Version	Description
2.10	The **interface chilli radius** command has been introduced.

3.27.31 interface chilli radiusacctport

Description

Set accounting UDP-port of *RADIUS* server. By default, value 1813 is used.

Command with **no** prefix resets port to default.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

Chilli

Synopsis

(config-if)> **chilli radiusacctport** ‹*radiusacctport*›

(config-if)> **no chilli radiusacctport**

Arguments

Argument	Value	Description
radiusacctport	*String*	The port number.

Example

```
(config-if)> chilli radiusacctport 1819
Chilli::Interface: "Chilli0": radiusacctport set to 1819.
```

```
(config-if)> no chilli radiusacctport
Chilli::Interface: "Chilli0": radiusacctport reset to default.
```

History

Version	Description
3.06	The **interface chilli radiusacctport** command has been introduced.

3.27.32 interface chilli radiusauthport

Description

Set authentication UDP-port of *RADIUS* server. By default, value 1812 is used.

Command with **no** prefix resets port to default.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type Chilli

Synopsis

(config-if)> **chilli radiusauthport** ‹*radiusauthport*›

(config-if)> **no chilli radiusauthport**

Arguments

Argument	Value	Description
radiusauthport	*String*	The port number.

Example

```
(config-if)> chilli radiusauthport 1820
Chilli::Interface: "Chilli0": radiusauthport set to 1820.
```

```
(config-if)> no chilli radiusauthport
Chilli::Interface: "Chilli0": radiusauthport reset to default.
```

History

Version	Description
3.06	The **interface chilli radiusauthport** command has been introduced.

3.27.33 interface chilli radiuslocationid

Description Set location identifier of *RADIUS* server. It should be in the format `isocc=, cc=, ac=, network=`.

Command with **no** prefix removes the setting.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type Chilli

Synopsis

(config-if)> **chilli radiuslocationid** ‹*radiuslocationid*›

(config-if)> **no chilli radiuslocationid**

Arguments

Argument	Value	Description
radiuslocationid	*String*	Location identifier value.

Example

```
(config-if)> chilli radiuslocationid ►
isocc=,cc=,ac=,network=WiFiSYSTEM,
Chilli::Interface: "Chilli0": radiuslocationid set to ►
"isocc=,cc=,ac=,network=WiFiSYSTEM,".
```

```
(config-if)> no chilli radiuslocationid
Chilli::Interface: "Chilli0": radiuslocationid cleared.
```

History

Version	Description
2.10	The **interface chilli radiuslocationid** command has been introduced.

3.27.34 interface chilli radiuslocationname

Description

Set location name of *RADIUS* server.

Command with **no** prefix removes the setting.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

Chilli

Synopsis

(config-if)> **chilli radiuslocationname** *‹radiuslocationname›*

(config-if)> **no chilli radiuslocationname**

Arguments

Argument	Value	Description
radiuslocationname	*String*	Location name.

Example

```
(config-if)> chilli radiuslocationname MyHotSpot
Chilli::Interface: "Chilli0": radiuslocationname set to ►
"MyHotSpot".
```

```
(config-if)> no chilli radiuslocationname
Chilli::Interface: "Chilli0": radiuslocationname cleared.
```

History

Version	Description
2.10	The **interface chilli radiuslocationname** command has been introduced.

3.27.35 interface chilli radiusnasid

Description

Set Network Access Server identifier.

Command with **no** prefix removes the setting.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type Chilli

Synopsis

(config-if)> **chilli radiusnasid** ‹*radiusnasid*›

(config-if)> **no chilli radiusnasid**

Arguments

Argument	Value	Description
radiusnasid	*String*	NAS identifier.

Example

```
(config-if)> chilli radiusnasid keeneticru_12
Chilli::Interface: "Chilli0": radiusnasid set to "keeneticru_12".
```

```
(config-if)> no chilli radiusnasid
Chilli::Interface: "Chilli0": radiusnasid cleared.
```

History

Version	Description
2.10	The **interface chilli radiusnasid** command has been introduced.

3.27.36 interface chilli radiussecret

Description Set shared secret for both *RADIUS* servers.

Command with **no** prefix removes the setting.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type Chilli

Synopsis

(config-if)> **chilli radiussecret** ‹*radiussecret*›

(config-if)> **no chilli radiussecret**

Arguments

Argument	Value	Description
radiussecret	*String*	A secret value.

Example

```
(config-if)> chilli radiussecret 12df34fd
Chilli::Interface: "Chilli0": radiussecret set to "12df34fd".
```

```
(config-if)>  no chilli radiussecret
Chilli::Interface: "Chilli0": radiussecret cleared.
```

History

Version	Description
2.10	The **interface chilli radiussecret** command has been introduced.

3.27.37 interface chilli uamallowed

Description

Specify the resource to which the client has access without first authenticating.

Command with **no** prefix removes the resource from the list. If you use no argument, the entire list of resources will be cleared.

Prefix no

Yes

Change settings

Yes

Multiple input

Yes

Interface type

Chilli

Synopsis

(config-if)> **chilli uamallowed** *‹uamallowed›*

(config-if)> **no chilli uamallowed** [*‹uamallowed›*]

Arguments

Argument	Value	Description
uamallowed	*String*	IP-addrress, URL or subnetwork.

Example

```
(config-if)> chilli uamallowed 188.166.114.0/24
Chilli::Interface: "Chilli0": "188.166.114.0/24" added to walled ►
garden.
```

```
(config-if)> chilli uamallowed www.example.link
Chilli::Interface: "Chilli0": "www.example.link" added to walled ►
garden.
```

```
(config-if)> no chilli uamallowed 188.166.114.0/24
Chilli::Interface: "Chilli0": "188.166.114.0/24" removed from ►
walled garden.
```

```
(config-if)> no chilli uamallowed www.example.link
Chilli::Interface: "Chilli0": "www.example.link" removed from ►
walled garden.
```

```
(config-if)> no chilli uamallowed
Chilli::Interface: "Chilli0": walled garden cleared.
```

History

Version	Description
2.10	The **interface chilli uamallowed** command has been introduced.

3.27.38 interface chilli uamdomain

Description

Specify the domain name to which the client has access without first authenticating.

Command with **no** prefix removes the domain name from the list. If you use no argument, the entire list of domain names will be cleared.

Prefix no

Yes

Change settings

Yes

Multiple input

Yes

Interface type

Chilli

Synopsis

(config-if)> **chilli uamdomain** *‹uamdomain›*

(config-if)> **no chilli uamdomain** [*‹uamdomain›*]

Arguments

Argument	Value	Description
uamdomain	*String*	Domain name of remote host.

Example

```
(config-if)> chilli uamdomain example.net
Chilli::Interface: "Chilli0": "example.net" added to walled ►
garden.
```

```
(config-if)> no chilli uamdomain example.net
Chilli::Interface: "Chilli0": "example.net" removed from walled ►
garden.
```

```
(config-if)> no chilli uamdomain
Chilli::Interface: "Chilli0": walled garden cleared.
```

History

Version	Description
2.10	The **interface chilli uamdomain** command has been introduced.

3.27.39 interface chilli uamhomepage

Description

Set URL of homepage to redirect unauthenticated users to.

Command with **no** prefix removes the setting.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type Chilli

Synopsis

(config-if)> **chilli uamhomepage** *‹uamhomepage›*

(config-if)> **no chilli uamhomepage**

Arguments

Argument	Value	Description
uamhomepage	*String*	Custom URL.

Example

```
(config-if)> chilli uamhomepage http://192.168.2.1/welcome.html
Chilli::Interface: "Chilli0": uamhomepage set to ►
"http://192.168.2.1/welcome.html".
```

```
(config-if)>  no chilli uamhomepage
Chilli::Interface: "Chilli0": uamhomepage cleared.
```

History

Version	Description
2.10	The **interface chilli uamhomepage** command has been introduced.

3.27.40 interface chilli uamport

Description Set *TCP* port to bind to for authenticating clients. By default, value 3990 is used.

Command with **no** prefix resets port to default.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type Chilli

Synopsis

(config-if)> **chilli uamport** *‹uamport›*

(config-if)> **no chilli uamport**

Arguments

Argument	Value	Description
uamport	*Integer*	The port number.

Example

```
(config-if)> chilli uamport 3922
Chilli::Interface: "Chilli0": uamport set to 3922.
```

```
(config-if)> no chilli uamport
Chilli::Interface: "Chilli0": uamport reset to default.
```

History

Version	Description
2.10	The **interface chilli uamport** command has been introduced.

3.27.41 interface chilli uamsecret

Description

Set shared secret between *UAM* server and Chilli. The *UAM* secret is used to hash the challenge before password computation.

Command with **no** prefix removes the setting.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

Chilli

Synopsis

(config-if)> **chilli uamsecret** *‹uamsecret›*

(config-if)> **no chilli uamsecret**

Arguments

Argument	Value	Description
uamsecret	*String*	A secret value.

Example

```
(config-if)> chilli uamsecret 12df34fd
Chilli::Interface: "Chilli0": uamsecret set to "12df34fd".
```

```
(config-if)> no chilli uamsecret
Chilli::Interface: "Chilli0": uamsecret set to "".
```

History

Version	Description
2.10	The **interface chilli uamsecret** command has been introduced.

3.27.42 interface chilli uamserver

Description

Set URL of web server to use for authenticating clients.

Command with **no** prefix removes the setting.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type Chilli

Synopsis

(config-if)> **chilli uamserver** *‹uamserver›*

(config-if)> **no chilli uamserver**

Arguments

Argument	Value	Description
uamserver	*String*	Custom URL of web server.

Example

```
(config-if)> chilli uamserver ►
https://auth.example.net/hotspotlogin
Chilli::Interface: "Chilli0": uamserver set to ►
"https://auth.example.net/hotspotlogin".
```

```
(config-if)> no chilli uamserver
Chilli::Interface: "Chilli0": uamserver cleared.
```

History

Version	Description
2.10	The **interface chilli uamserver** command has been introduced.

3.27.43 interface compatibility

Description Set the standard for wireless communications, with which a given wireless adapter (the interface) must be compatible. For Wi-Fi interfaces, the compatibility is set by string of Latin letters A, B, G, N, that denote extensions to the standard IEEE 802.11. For example, the presence 'N' in the compatibility line will imply that the given adapter will be able to deal with the 802.11n-compatible devices via radio channel. The set of admissible compatibility lines is defined by the hardware capabilities of a particular adapter and provisions of the relevant additions to the IEEE 802.11 standard.

By default, “BGN” value is used for 2.4 GHz, “AN” — for 5 GHz.

Prefix no No

Change settings Yes

Multiple input No

Interface type Radio

Synopsis

(config-if)> **compatibility** *‹annex›*

Arguments

Argument	Value	Description
annex	`B, G, N`	For 2,4 GHz.
	`A, N`	For 5 GHz.
	`A, N+AC`	Additional IEEE standard.

Example

```
(config-if)> compatibility N
Network::Interface::Rtx::WifiMaster: "WifiMaster0": PHY mode set.

(config-if)> compatibility N+AC
Network::Interface::Rtx::WifiMaster: "WifiMaster1": PHY mode set.
```

History

Version	Description
2.00	The **interface compatibility** command has been introduced.
2.06	New standard AC was added.

3.27.44 interface connect

Description

Start the process of connecting to a remote node.

Command with **no** prefix terminates the connection.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

PPP, IP

Synopsis

(config-if)> **connect** [**via** *‹via›*]

(config-if)> **no connect**

Arguments

Argument	Value	Description
via	*Interface name*	Interface through which remote node is accessed. For PPPoE this option is mandatory.

Example

```
(config-if)> connect via ISP

(config-if)> no connect
```

History

Version	Description
2.00	The **interface connect** command has been introduced.

3.27.45 interface country-code

Description Assign to the interface a literal country code, which influences the set of radio channels. By default, RU value is used.

Prefix no No

Change settings Yes

Multiple input No

Interface type Radio

Synopsis

(config-if)> **country-code** *‹code›*

Arguments

Argument	Value	Description
code	*String*	The country code.

Example

```
(config-if)> country-code RU
Network::Interface::Rtx::WifiMaster: "WifiMaster0": country code ►
set.
```

History

Version	Description
2.00	The **interface country-code** command has been introduced.

3.27.46 interface debug

Description Enable debug mode of *PPP* connection. Detailed info about connection progress is saved to the system log. By default, setting is disabled.

Command with **no** prefix disables the debug mode.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type PPP

Synopsis

(config-if)> **debug**

(config-if)> **no debug**

Example

```
(config-if)> debug
Network::Interface::Base: Debug enabled.
```

```
(config-if)> no debug
Network::Interface::Base: Debug disabled.
```

History

Version	Description
2.00	The **interface debug** command has been introduced.

3.27.47 interface description

Description

Assign arbitrary descrption to the specified network interface.

Command with **no** prefix deletes the description.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config-if)> **description** *‹description›*

(config-if)> **no description**

Arguments

Argument	Value	Description
description	*String*	Arbitrary description of the interface.

Example

```
(config-if)> description MYHOME
Network::Interface::Base: "Bridge0": description saved.
```

```
(config-if)> no description
Network::Interface::Base: "Bridge0": description saved.
```

History

Version	Description
2.00	The **interface description** command has been introduced.

3.27.48 interface down

Description

Disable the network interface and persist the state “down” to the settings.

Command with **no** prefix enables the network interface and deletes “down” from settings.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config-if)> **down**

(config-if)> **no down**

Example

```
(config-if)> down
Network::Interface::Base: "GigabitEthernet0/2": interface is down.
```

```
(config-if)> up
Network::Interface::Base: "GigabitEthernet0/2": interface is up.
```

History

Version	Description
2.00	The **interface down** command has been introduced.

3.27.49 interface duplex

Description

Set the duplex mode of the Ethernet port. By default, `auto` value is set.

Command with **no** prefix resets setting to default.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

Ethernet

Synopsis

(config-if)> **duplex** (**full** | **half** | **auto**)

(config-if)> **no duplex**

Arguments

Argument	Value	Description
mode	full	Full duplex protocol.
	half	Half duplex protocol.
	auto	Auto duplex protocol.

Example

```
(config-if)> duplex full
Network::Interface::Ethernet: "GigabitEthernet0/1": duplex set ►
to "full".
```

```
(config-if)> no duplex
Network::Interface::Ethernet: "GigabitEthernet0/1": duplex reset ►
to default.
```

History

Version	Description
2.06.B.1	The **interface duplex** command has been introduced.

3.27.50 interface dyndns profile

Description Assign the DynDns profile to the interface. Profile must be created and customized with dyndns profile commands before execution.

Command with **no** prefix unbinds the profile.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config-if)> **dyndns profile** ‹*profile*›

(config-if)> **no dyndns profile**

Arguments

Argument	Value	Description
`profile`	*String*	The name of DynDns profile.

Example

```
(config-if)> dyndns profile TEST
DynDns::Profile: Interface set.
```

```
(config-if)> no dyndns profile TEST
DynDns::Profile: Interface removed.
```

History

Version	Description
2.02	The **interface dyndns profile** command has been introduced.

3.27.51 interface dyndns update

Description Update IP-address for DynDns manually. By default command works in accordance with the policy of the DynDns service provider, that is not allows to update too often. Using the keyword `force` allows you to update excluding policy of the service provider.

Prefix no No

Change settings Yes

Multiple input No

Synopsis

(config-if)> **dyndns update** [**force**]

Arguments

Argument	Value	Description
`force`	*Keyword*	Not take into account the update rate recommended by service provider.

Example

```
(config-if)> dyndns update
```

History

Version	Description
2.00	The **interface dyndns update** command has been introduced.

3.27.52 interface encryption anonymous-dh

Description Enable Anonymous DH for SSTP-servers without a certificate.

Command with **no** prefix disables Anonymous DH.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type SSTP

Synopsis

(config-if)> **encryption anonymous-dh**

(config-if)> **no encryption anonymous-dh**

Example

```
(config-if)> encryption anonymous-dh
Network::Interface::Sstp: "SSTP0": anonymous DH TLS is enabled.

(config-if)> no encryption anonymous-dh
Network::Interface::Sstp: "SSTP0": anonymous DH TLS is disabled.
```

History

Version	Description
2.13	The **interface encryption anonymous-dh** command has been introduced.

3.27.53 interface encryption disable

Description Disable encryption on the wireless interface.

Prefix no No

Change settings Yes

Multiple input No

Interface type WiFi

Synopsis

(config-if)> **encryption disable**

Example

```
(config-if)> encryption disable
Network::Interface::Rtx::AccessPoint: "WifiMaster0/AccessPoint0": ►
wireless encryption disabled.
```

History

Version	Description
2.00	The **interface encryption disable** command has been introduced.

3.27.54 interface encryption enable

Description

Enable encryption on the wireless interface. By default, *WEP* encryption is used.

Command with **no** prefix disables wireless interface encryption.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

WiFi

Synopsis

(config-if)> **encryption enable**

(config-if)> **no encryption enable**

Example

```
(config-if)> encryption enable
Network::Interface::Rtx::AccessPoint: "WifiMaster0/AccessPoint0": ►
wireless encryption enabled.
```

```
(config-if)> no encryption enable
Network::Interface::Rtx::AccessPoint: "WifiMaster0/AccessPoint0": ►
wireless encryption disabled.
```

History

Version	Description
2.00	The **interface encryption enable** command has been introduced.

3.27.55 interface encryption key

Description

Specify the *WEP* encryption keys. Depending on the bit, the key can be standard 64-bit *WEP* uses a 40 bit key (also known as WEP-40), or 128-bit *WEP* uses a 26 hexadecimal characters (13 characters ASCII). Overall, there can be 1 to 4 encryption keys, with one of them default key must be assigned.

Command with **no** prefix removes key.

Prefix no Yes

Change settings Yes

Multiple input Yes

Interface type WiFi

Synopsis

(config-if)> **encryption key** ‹*id*› (‹*value*› [**default**] | **default**)

(config-if)> **no encryption key** ‹*id*›

Arguments

Argument	Value	Description
`id`	*Integer*	The key number. Overall, up to 4 keys could be specified.
`value`	*String*	The key value as a hexadecimal number, consisting of 10 or 26 digits.
`default`	*Keyword*	Indicates that this key will be used by default.

Example

```
(config-if)> encryption key 1 1231231234
Network::Interface::Wifi: "WifiMaster0/AccessPoint0": WEP key 1 ►
set.
```

```
(config-if)> no encryption key 1
Network::Interface::Wifi: "WifiMaster0/AccessPoint0": WEP key 1 ►
removed.
```

History

Version	Description
2.00	The **interface encryption key** command has been introduced.

3.27.56 interface encryption mppe

Description Enable *MPPE* encryption support.

Command with **no** prefix disables *MPPE* encryption.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type PPTP

Synopsis

(config-if)> **encryption mppe**

(config-if)> **no encryption mppe**

Example

```
(config-if)> encryption mppe
MPPE enabled.
```

```
(config-if)> no encryption mppe
MPPE disabled.
```

History

Version	Description
2.00	The **interface encryption mppe** command has been introduced.

3.27.57 interface encryption owe

Description

Enable *OWE* security algorithms on the wireless interface. By default, the setting is disabled.

Command with **no** prefix disables *OWE* support.

Prefix no

Yes

Changse settings

Yes

Multiple input

No

Interface type

WifiMaster

Synopsis

(config-if)> **encryption owe**

(config-if)> **no encryption owe**

Example

```
(config-if)> encryption owe
Network::Interface::Rtx::AccessPoint: "WifiMaster0/AccessPoint0": ►
OWE algorithms enabled.
```

```
(config-if)> no encryption owe
Network::Interface::Rtx::AccessPoint: "WifiMaster0/AccessPoint0": ►
OWE algorithms disabled.
```

History

Version	Description
3.00	The **interface encryption owe** command has been introduced.

3.27.58 interface encryption wpa

Description

Enable *WPA* security algorithms on the wireless interface. Wireless interface can support the joint use of *WPA* and *WPA2*, but supporting *WEP* automatically disables when any of the *WPA* is enabled.

Command with **no** prefix disables *WPA* support.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type WiFi

Synopsis

(config-if)> **encryption wpa**

(config-if)> **no encryption wpa**

Example

```
(config-if)> encryption wpa
WPA algorithms enabled.
```

History

Version	Description
2.00	The **interface encryption wpa** command has been introduced.

3.27.59 interface encryption wpa2

Description Enable *WPA2* (IEEE 802.11i, RSN) security algorithms on the wireless interface. Wireless interface can support the joint use of *WPA* and *WPA2*, but supporting *WEP* automatically disables when any of the *WPA* is enabled.

Command with **no** prefix disables *WPA2* support.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type WiFi

Synopsis

(config-if)> **encryption wpa2**

(config-if)> **no encryption wpa2**

Example

```
(config-if)> encryption wpa2
WPA2 algorithms enabled.
```

History

Version	Description
2.00	The **interface encryption wpa2** command has been introduced.

3.27.60 interface encryption wpa3

Description

Enable *WPA3* security algorithms on the wireless interface. Wireless interface can support the joint use of *WPA2* and *WPA3*. By default, the setting is disabled.

Command with **no** prefix disables *WPA3* support.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

WiFi

Synopsis

(config-if)> **encryption wpa3**

(config-if)> **no encryption wpa3**

Example

```
(config-if)> encryption wpa3
Network::Interface::Rtx::AccessPoint: "WifiMaster0/AccessPoint0": ►
WPA3 algorithms enabled.
```

```
(config-if)> no encryption wpa3
Network::Interface::Rtx::AccessPoint: "WifiMaster0/AccessPoint0": ►
WPA3 algorithms disabled.
```

History

Version	Description
3.00	The **interface encryption wpa3** command has been introduced.

3.27.61 interface encryption wpa3 suite-b

Description

Enable *WPA3* security algorithms to protect sensitive data Suite-B for *WPA Enterprise*. By default, the feature is disabled.

Prefix no

No

Change settings

Yes

Multiple input

No

Interface type

WiFi

Synopsis

(config-if)> **encryption wpa3 suite-b**

Example

```
(config-if)> encryption wpa3 suite-b
Network::Interface::Rtx::AccessPoint: "WifiMaster0/AccessPoint1": ►
WPA3 SuiteB enabled.
```

History

Version	Description
3.01	The **interface encryption wpa3 suite-b** command has been introduced.

3.27.62 interface flowcontrol

Description

Configure Ethernet flow control Tx/Rx. By default, the feature is enabled.

Command with **no** prefix disables the feature.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

Ethernet

Synopsis

(config-if)> **flowcontrol on**

(config-if)> **no flowcontrol** [**send**]

Arguments

Argument	Value	Description
send	*Keyword*	Flow control works asynchronously.

Example

```
(config-if)> flowcontrol on
Network::Interface::Ethernet: "GigabitEthernet0/0": flow control ►
enabled.
```

```
(config-if)> no flowcontrol send
Network::Interface::Ethernet: "GigabitEthernet0/0": flow control ►
send disabled.
```

History

Version	Description
2.08	The **interface flowcontrol** command has been introduced.

3.27.63 interface follow

Description

Copy settings from AP on WifiMaster0 (2.4 GHz) to the AP on WifiMaster with an index greater than zero (5 GHz or above).

The follower automatically copies all changes applied to the master access point.

If you change the follower settings, the link with the master access point is terminated.

Warning: The WifiMaster0 access points are always used as a source of settings. They never follow. They can only be followed.

Prefix no No

Change settings Yes

Multiple input No

Interface type AccessPoint

Synopsis

(config-if)> **follow** ‹*access-point*›

Arguments

Argument	Value	Description
`access-point`	*Interface name*	The name of an `AccessPoint` interface on the `WifiMaster0` 2.4 GHz. You can see the list of available interfaces with help of **follow** [Tab] command.

Example

```
(config-if)> follow WifiMaster0/AccessPoint0
Network::Interface::AccessPoint: "WifiMaster1/AccessPoint0": set ►
to follow WifiMaster0/AccessPoint0.
```

History

Version	Description
3.07	The **interface follow** command has been introduced.

3.27.64 interface ft enable

Description

Enable support of *FT* for Access Point (FT Over the Air, OTA) within the IEEE 802.11r standard. By default, the option is disabled.

For correct *FT* operation between 2,4 and 5 GHz APs it is necessary to fulfill the following conditions:

- access points 2,4 GHz and 5 GHz are enabled both
- they have the same SSID's
- they have the same security settings (encryption type — WPA2 or without password, password value, etc.)

Command with **no** prefix removes the setting.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type AccessPoint

Synopsis

(config-if)> **ft enable**

(config-if)> **no ft enable**

Example

```
(config-if)> ft enable
Network::Interface::Rtx::AccessPoint: "WifiMaster0/AccessPoint0": ►
fast transition enabled.
```

```
(config-if)> no ft enable
Network::Interface::Rtx::AccessPoint: "WifiMaster0/AccessPoint0": ►
fast transition disabled.
```

History

Version	Description
2.13	The **interface ft enable** command has been introduced.

3.27.65 interface ft mdid

Description

Set Mobility Domain ID for *FT*. By default, KN value is used.

Command with **no** prefix resets setting to default.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

AccessPoint

Synopsis

(config-if)> **ft mdid** *‹mdid›*

(config-if)> **no ft mdid**

Arguments

Argument	Value	Description
mdid	*String*	The value of Mobility Domain ID. Consists of 2 ASCII symbols.

Example

```
(config-if)> ft mdid 1F
Network::Interface::Rtx::AccessPoint: "WifiMaster0/AccessPoint0": ►
fast transition MDID set to "1F".
```

```
(config-if)> no ft mdid
Network::Interface::Rtx::AccessPoint: "WifiMaster0/AccessPoint0": ►
fast transition MDID reset to default.
```

History

Version	Description
2.13	The **interface ft mdid** command has been introduced.

3.27.66 interface ft otd

Description

Enable support of *FT* Over-the-DS (Distribution System) within the IEEE 802.11r standard. This type of *FT* is used for roaming in outdated subscriber devices, for example, in the iPhone 4s. By default, the setting is disabled.

Command with **no** prefix removes the setting.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

AccessPoint

Synopsis

(config-if)> **ft otd**

(config-if)> **no ft otd**

Example

```
(config-if)> ft otd
Network::Interface::Rtx::AccessPoint: "WifiMaster0/AccessPoint0": ►
fast transition OTD enabled.
```

```
(config-if)> no ft otd
Network::Interface::Rtx::AccessPoint: "WifiMaster0/AccessPoint0": ►
fast transition OTD disabled.
```

History

Version	Description
2.13	The **interface ft otd** command has been introduced.

3.27.67 interface hide-ssid

Description

Enable hidden *SSID* mode. When using this feature, Access Point will not be displayed in the list of available wireless networks. But if user informed of the existence of this network and know its *SSID*, than he can connect to it. The mode is disabled by default.

Command with **no** prefix disables the mode.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

Access Point

Synopsis

(config-if)> **hide-ssid**

(config-if)> **no hide-ssid**

Example

```
(config-if)> hide-ssid
Network::Interface::Rtx::AccessPoint: "WifiMaster0/AccessPoint0": ►
SSID broadcasting disabled.
```

```
(config-if)> no hide-ssid
Network::Interface::Rtx::AccessPoint: "WifiMaster0/AccessPoint0": ►
SSID broadcasting enabled.
```

History

Version	Description
2.00	The **interface hide-ssid** command has been introduced.

3.27.68 interface iapp auto

Description

Generate *IAPP* key in automatic mode. To assign the key manually, use **interface iapp key** command.

Prefix no

No

Change settings

Yes

Multiple input

No

Interface type

Bridge

Synopsis

(config-if)> **iapp auto**

Example

```
(config-if)> iapp auto
Network::Interface::Rtx::Iapp: Bridge0 autoconfigured.
```

History

Version	Description
3.03	The **interface iapp auto** command has been introduced.

3.27.69 interface iapp key

Description

Assign the *IAPP* Mobile Domain key for successful synchronization between Access Points where *FT* works (**interface ft enable** command). Access Points must belong to the same IP-subnet. By default, the key is not assigned.

Command with **no** prefix removes key value.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

Bridge

Synopsis

(config-if)> **iapp key** *‹key›*

(config-if)> **no iapp key**

Arguments

Argument	Value	Description
key	*String*	The value of *IAPP* key. Maximum key length is 64 characters.

Example

```
(config-if)> iapp key 11223344556677
Network::Interface::Rtx::Iapp: Bridge0 key applied.
```

```
(config-if)> no iapp key
Network::Interface::Rtx::Iapp: Bridge0 key cleared.
```

History

Version	Description
2.13	The **interface iapp key** command has been introduced.

3.27.70 interface idle-timeout

Description

Set the interval for the STA client to disconnect from the Access Point by inactivity timeout. By default, `600` value is used.

Command with **no** prefix disables the setting.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

WiFiMaster

Synopsis

(config-if)> **idle-timeout** *‹idle-timeout›*

(config-if)> **no idle-timeout**

Arguments

Argument	Value	Description
idle-timeout	*Integer*	Idle-timeout value in seconds. Can take values from 60 to 2147483646.

Example

```
(config-if)> idle-timeout 500
Network::Interface::Rtx::WifiMaster: "WifiMaster1": idle timeout ►
value is 500 sec.
```

```
(config-if)> no idle-timeout
Network::Interface::Rtx::WifiMaster: "WifiMaster1": idle timeout ►
disabled.
```

History

Version	Description
3.06	The **interface idle-timeout** command has been introduced.

3.27.71 interface igmp downstream

Description

Enable *IGMP* mode on the interface in the direction of the multicast recipients. **service igmp-proxy** must be enabled on the device. There can be several `downstream` interfaces.

Command with **no** prefix disables the mode.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

IP

Synopsis

(config-if)> **igmp downstream**

(config-if)> **no igmp downstream**

Example

```
(config-if)> igmp downstream
```

```
(config-if)> no igmp downstream
```

History

Version	Description
2.00	The **interface igmp downstream** command has been introduced.

3.27.72 interface igmp fork

Description

Enable the duplication of outgoing packets *IGMP*`upstream` to the specified interface. There can be only one `fork` interface.

Command with **no** prefix disables the mode.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

IP

Synopsis

(config-if)> **igmp fork**

(config-if)> **no igmp fork**

Example

```
(config-if)> igmp fork
```

```
(config-if)> no igmp fork
```

History

Version	Description
2.00	The **interface igmp fork** command has been introduced.

3.27.73 interface igmp upstream

Description

Enable *IGMP* mode on the interface in the direction of the multicast source. **service igmp-proxy** must be enabled on the device. Only one `upstream` interface is allowed.

Command with **no** prefix disables the mode.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

IP

Synopsis

(config-if)> **igmp upstream**

(config-if)> **no igmp upstream**

Example

```
(config-if)> igmp upstream
```

```
(config-if)> no igmp upstream
```

History

Version	Description
2.00	The **interface igmp upstream** command has been introduced.

3.27.74 interface include

Description

Specify Ethernet-interface name which will be added to the software bridge as a port.

Command with **no** prefix removes the interface from the bridge.

Prefix no

Yes

Change settings

Yes

Multiple input Yes

Interface type Bridge

Synopsis

(config-if)> **include** ‹*interface*›

(config-if)> **no include** ‹*interface*›

Arguments

Argument	Value	Description
interface	*Interface name*	Name or alias of the Ethernet-interface that should be plugged into the bridge.

Example

```
(config-if)> include ISP
Network::Interface::Bridge: "Bridge0": ISP included.
```

```
(config-if)> no include
Network::Interface::Bridge: "Bridge0": removed ISP.
```

History

Version	Description
2.00	The **interface include** command has been introduced.

3.27.75 interface inherit

Description Specify the name of the Ethernet-interface which will be added to the program bridge as a port. In contrast with the **include** command, **inherit** command transfers some settings of the interface being added to the bridge, such as IP-address, mask and IP-aliases. On removing either the bridge itself or the bridge interface, these settings, even if they have been changed will be copied back to the vacant interface.

The command allows one to add the device control interface to the bridge so that control is not lost.

Command with **no** prefix removes the interface from the bridge, returns the settings that have earlier been inherited by the bridge back to the interface, and resets these settings on the bridge.

Prefix no Yes

Change settings Yes

Multiple input Yes

Interface type Bridge

Synopsis

(config-if)> **inherit** ‹*interface*›

(config-if)> **no inherit** ‹*interface*›

Arguments

Argument	Value	Description
`interface`	*`Interface name`*	Name or alias of the Ethernet-interface that should be plugged into the bridge.

Example

```
(config-if)> inherit GigabitEthernet0/Vlan3
Network::Interface::Bridge: "Bridge1": GigabitEthernet0/Vlan3 ►
inherited in Bridge1.
```

```
(config-if)> no inherit
Network::Interface::Bridge: "Bridge1": inherit removed.
```

History

Version	Description
2.00	The **interface inherit** command has been introduced.

3.27.76 interface ip access-group

Description

Assign a named list of filtering rules (*ACL*, see **access-list**) to the interface. Parameter `in` or `out` indicates the traffic direction for which the *ACL* will be applied. Several ACLs can be assigned to a single interface.

Command with **no** prefix disables the *ACL* for the specified interface and traffic direction.

Prefix no

Yes

Change settings

Yes

Multiple input

Yes

Interface type

IP

Synopsis

(config-if)> **ip access-group** *‹acl›* *‹direction›*

(config-if)> **no ip access-group** [*‹acl›* [*‹direction›*]]

Arguments

Argument	Value	Description
`acl`	*`String`*	List of filtering rules as previously created using **access-list** command.
`direction`	`in`	Apply filtering to incoming packets.
	`out`	Apply filtering to outgoing packets.

Example

```
(config-if)> ip access-group BLOCK in
Network::Acl: Input "BLOCK" access list added to "CdcEthernet1".
```

```
(config-if)> ip access-group BLOCK out
Network::Acl: Output "BLOCK" access list added to "CdcEthernet1".
```

```
(config-if)> no ip access-group BLOCK in
Network::Acl: "BLOCK" access group deleted from "CdcEthernet1".
```

```
(config-if)> no ip access-group
Network::Acl: All access groups deleted from "CdcEthernet1".
```

History

Version	Description
2.00	The **interface ip access-group** command has been introduced.

3.27.77 interface ip address

Description

Change the IP-address and the mask of the network interface. If the address automatic configuration service is running on the interface, for instance, DHCP-client, (see **interface ip address dhcp**), then the manually set address can be overwritten.

Command with **no** prefix resets the address to `0.0.0.0`.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

IP

Synopsis

(config-if)> **ip address** *‹address› ‹mask›*

(config-if)> **no ip address**

Arguments

Argument	Value	Description
address	*IP-address*	The network interface address.
mask	*IP-mask*	The network interface mask. There are two ways to specify the mask: the canonical form (for example, `255.255.255.0`) and the prefix with bit length (for example, `/24`).

Example

The network address, defined by the IP-address and mask, can specified in either of the two ways: specify a mask in the canonical form, or set the prefix bit length.

```
(config)> ip address 192.168.9.1/24
Network::Interface::Ip: "Bridge3": IP address is 192.168.9.1/24.
```

```
(config)> no ip address
Network::Interface::Ip: "Bridge3": IP address cleared.
```

History

Version	Description
2.00	The **interface ip address** command has been introduced.

3.27.78 interface ip address dhcp

Description

Start the DHCP-client to automatically configure the network parameters: IP-address and mask of the interface, *DNS* servers and default gateway.

Command with **no** prefix stops the DHCP-client, removes the dynamically configured settings and restores the previous settings of IP-address and mask.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

Ethernet

Synopsis

(config-if)> **ip address dhcp [hostname** ‹*hostname*› **]**

(config-if)> **no ip address dhcp**

Arguments

Argument	Value	Description
hostname	*String*	Name of the host to be placed in the DHCP option 12 field. This name need not be the same as the host name entered in global configuration mode.

Example

```
(config-if)> ip address dhcp hostname QWERTY2
Dhcp::Client: Started DHCP client on ISP.
```

```
(config-if)> no ip address dhcp
Dhcp::Client: Stopped DHCP client on ISP.
```

History

Version	Description
2.00	The **interface ip address dhcp** command has been introduced.

3.27.79 interface ip adjust-ttl recv

Description

Modify the TTL for all inbound packets on the interface.

Command with **no** prefix cancels the setting.

Prefix no

Yes

Change settings Yes

Multiple input No

Interface type IP

Synopsis

(config-if)> **ip adjust-ttl recv** *‹recv›*

(config-if)> **no ip adjust-ttl recv**

Arguments

Argument	Value	Description
recv	*Integer*	The value of TTL changing. Can take values from 1 to 255 inclusively.

Example

```
(config-if)>  ip adjust-ttl recv 1
Network::Interface::Ip: "CdcEthernet0": incoming TTL set to 1.
```

```
(config-if)> no ip adjust-ttl recv
Network::Interface::Ip: "CdcEthernet0": incoming TTL settings ►
removed.
```

History

Version	Description
3.07	The **interface ip adjust-ttl recv** command has been introduced. Previous command name is **interface ip adjust-ttl**.

3.27.80 interface ip adjust-ttl send

Description Modify the TTL for all outbound packets on the interface.

Command with **no** prefix cancels the setting.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type IP

Synopsis

(config-if)> **ip adjust-ttl send** *‹send›*

(config-if)> **no ip adjust-ttl send**

Arguments

Argument	Value	Description
send	*Integer*	The value of TTL changing. Can take values from 1 to 255 inclusively.

Example

```
(config-if)> ip adjust-ttl send 65
Network::Interface::Ip: "CdcEthernet1": outgoing TTL set to 65.

(config-if)> no ip adjust-ttl send
Network::Interface::Ip: "CdcEthernet1": outgoing TTL settings ►
removed.
```

History

Version	Description
2.09	The **interface ip adjust-ttl send** command has been introduced.

3.27.81 interface ip alias

Description

Assign an additional IP-address and mask to the network interface (alias).

Command with **no** prefix resets the specified alias to `0.0.0.0`. If you use no arguments, the entire list of aliases will be removed.

Prefix no

Yes

Change settings

Yes

Multiple input

Yes

Interface type

IP, Ethernet

Synopsis

(config-if)> **ip alias** *‹address› ‹mask›*

(config-if)> **no ip alias** [*‹address› ‹mask›*]

Arguments

Argument	Value	Description
`address`	*IP-address*	Additional address of the network interface.
`mask`	*IP-mask*	Additional mask of the network interface. There are two ways to specify the mask: the canonical form (for example, `255.255.255.0`) and the prefix with bit length (for example, `/24`).

Example

```
(config-if)> ip alias 192.168.1.88/24
Network::Interface::Ip: "WifiMaster1/WifiStation0": alias 0 is ►
192.168.1.88/24.

(config-if)> no ip alias 192.168.1.88/24
Network::Interface::Ip: "WifiMaster1/WifiStation0": alias 0 reset ►
to 0.0.0.0/0.

(config-if)> no ip alias
Network::Interface::Ip: "WifiMaster1/WifiStation0": all aliases ►
removed.
```

History

Version	Description
2.00	The **interface ip alias** command has been introduced.

3.27.82 interface ip dhcp client broadcast

Description

Set broadcast bit in the DHCP Discover messages, that indicate to a server how the reply should be sent back to the client. By default, the setting is disabled.

Command with **no** prefix removes the setting.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

Ethernet

Synopsis

(config-if)> **ip dhcp client broadcast**

(config-if)> **no ip dhcp client broadcast**

Example

```
(config-if)> ip dhcp client broadcast
Dhcp::Client: ISP DHCP client request broadcast enabled.
```

```
(config-if)> no ip dhcp client broadcast
Dhcp::Client: ISP DHCP client request broadcast disabled.
```

History

Version	Description
2.15	The **interface ip dhcp client broadcast** command has been introduced.

3.27.83 interface ip dhcp client class-id

Description

Specify the device vendor name where *DHCP* client is running (dhcp option 60).

Command with **no** prefix removes the setting.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

Ethernet

Synopsis

(config-if)> **ip dhcp client class-id** *‹class›*

(config-if)> **no ip dhcp client class-id**

Arguments

Argument	Value	Description
`class`	*`String`*	Vendor class name, enclosed in double quotes.

Example

```
(config-if)> ip dhcp client class-id "City"
Dhcp::Client: ISP DHCP client vendor class is set to "City".
```

```
(config-if)> no ip dhcp client class-id
Dhcp::Client: ISP DHCP client vendor class is cleared.
```

History

Version	Description
2.02	The **interface ip dhcp client class-id** command has been introduced.

3.27.84 interface ip dhcp client debug

Description

Enable debug mode for DHCP-client. Detailed info about DHCP-client working is saved to the system log.

Command with **no** prefix disables the debug mode.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

Ethernet

Synopsis

(config-if)> **ip dhcp client debug**

(config-if)> **no ip dhcp client debug**

Example

```
(config-if)> ip dhcp client debug
Dhcp::Client: ISP DHCP client debug enabled.
```

```
(config-if)> no ip dhcp client debug
Dhcp::Client: ISP DHCP client debug disabled.
```

History

Version	Description
2.01	The **interface ip dhcp client debug** command has been introduced.

3.27.85 interface ip dhcp client displace

Description

Displace static address of *what* if it conflicts with an address from DHCP-client of main interface.

This command is executed automatically when you connect the USB Ethernet adapter. After that the configuration will be saved and device will be restarted.

Command with **no** prefix cancels the displacement for the specified interface.

Prefix no

Yes

Change settings

Yes

Multiple input

Yes

Interface type

Ethernet

Synopsis

(config-if)> **ip dhcp client displace** ‹*what*› [**check-session**]

(config-if)> **no ip dhcp client displace** ‹*what*› [**check-session**]

Arguments

Argument	Value	Description
what	*Interface name*	Name or alias of the interface whose static address will be displaced.
check-session	*Keyword*	With active SCGI sessions, it does not allow rebooting and changing the router's network address. By default, command is added to default-config.

Example

```
(config-if)> ip dhcp client displace Home
Dhcp::Client: ISP added "Home" displacement.
```

```
(config-if)> ip dhcp client displace Home check-session
Dhcp::Client: ISP added "Home" displacement.
```

```
(config-if)> no ip dhcp client displace Home
Dhcp::Client: ISP deleted "Home" displacement.
```

```
(config-if)> no ip dhcp client displace Home check-session
Dhcp::Client: ISP deleted "Home" displacement.
```

History

Version	Description
2.03	The **interface ip dhcp client displace** command has been introduced.
2.15	Argument check-session was added.

3.27.86 interface ip dhcp client dns-routes

Description Enable automatic addition of host routes to the DNS-server received from the DHCP-server. By default, the setting is enabled.

Command with **no** prefix disables the setting.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type Ethernet

Synopsis

(config-if)> **ip dhcp client dns-routes**

(config-if)> **no ip dhcp client dns-routes**

Example

```
(config-if)> ip dhcp client dns-routes
Dhcp::Client: ISP DHCP client DNS host routes are enabled.
```

```
(config-if)> no ip dhcp client dns-routes
Dhcp::Client: ISP DHCP client DNS host routes are disabled.
```

History

Version	Description
2.00	The **interface ip dhcp client dns-routes** command has been introduced.

3.27.87 interface ip dhcp client fallback

Description Set static IP-address in case of DHCP errors.

Command with **no** prefix cancels setting and sets `0.0.0.0.` address.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type Ethernet

Synopsis

(config-if)> **ip dhcp client fallback** *‹type›*

(config-if)> **no ip dhcp client fallback**

Arguments

Argument	Value	Description
`type`	*`String`*	The type of IP-address. Currently implemented only one type — `static`.

Example

```
(config-if)> ip dhcp client fallback static
Dhcp::Client: A DHCP address fallback is static.
```

```
(config-if)> no ip dhcp client fallback
Dhcp::Client: A DHCP address fallback set to zero for "ISP".
```

History

Version	Description
2.05	The **interface ip dhcp client fallback** command has been introduced.

3.27.88 interface ip dhcp client hostname

Description

Assign a host name which is sent in DHCP-request.

Command with **no** prefix resets the host name to default.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

Ethernet

Synopsis

(config-if)> **ip dhcp client hostname** ‹*hostname*›

(config-if)> **no ip dhcp client hostname**

Arguments

Argument	Value	Description
hostname	*String*	The host name to assign.

Example

```
(config-if)> ip dhcp client hostname MYHOME
Dhcp::Client: ISP DHCP client hostname is set to MYHOME.
```

```
(config-if)> no ip dhcp client hostname
Dhcp::Client: ISP DHCP client hostname is reset to default (HOME).
```

History

Version	Description
2.00	The **interface ip dhcp client hostname** command has been introduced.

3.27.89 interface ip dhcp client name-servers

Description

Use *DNS*-server addresses which are received via *DHCP*. By default, the function is enabled.

Command with **no** prefix denies using of *DNS*-server addresses which are received via *DHCP*.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type Ethernet

Synopsis

(config-if)> **ip dhcp client name-servers**

(config-if)> **no ip dhcp client name-servers**

Example

```
(config-if)> ip dhcp client name-servers
Dhcp::Client: ISP DHCP name servers are enabled.
```

```
(config-if)> no ip dhcp client name-servers
Dhcp::Client: ISP DHCP name servers are disabled.
```

History

Version	Description
2.00	The **interface ip dhcp client name-servers** command has been introduced.

3.27.90 interface ip dhcp client release

Description DHCP-client releases lease IP-address and goes into sleep mode. Another execution of this command takes DHCP-client to the mode of automatical obtaining of IP-address.

Prefix no No

Change settings Yes

Multiple input No

Interface type Ethernet

Synopsis

(config-if)> **ip dhcp client release**

Example

```
(config-if)> ip dhcp client release
Dhcp::Client: IP address released.
```

History

Version	Description
2.03	The **interface ip dhcp client release** command has been introduced.

3.27.91 interface ip dhcp client renew

Description DHCP-client releases lease IP-address and passes in a mode of obtaining a new one.

Prefix no No

Change settings Yes

Multiple input No

Interface type Ethernet

Synopsis

(config-if)> **ip dhcp client renew**

Example

```
(config-if)> ip dhcp client renew
Dhcp::Client: IP address renewed.
```

History

Version	Description
2.03	The **interface ip dhcp client renew** command has been introduced.

3.27.92 interface ip dhcp client routes

Description Enable receiving routes from the provider (dhcp options 33, 121, 242). By default it is enabled. In the configuration it is displayed only with **no** prefix.

Command with **no** prefix disables the setting.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type Ethernet

Synopsis

(config-if)> **ip dhcp client routes**

(config-if)> **no ip dhcp client routes**

Example

```
(config-if)> ip dhcp client routes
Dhcp::Client: ISP DHCP client static routes are enabled.
```

```
(config-if)> no ip dhcp client routes
Dhcp::Client: ISP DHCP client static routes are disabled.
```

History

Version	Description
2.05	The **interface ip dhcp client routes** command has been introduced.

3.27.93 interface ip flow

Description

Enable *NetFlow* sensor on the specified interface. By default, the setting is disabled.

Command with **no** prefix disables *NetFlow* sensor.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

IP

Synopsis

(config-if)> **ip flow** *‹direction›*

(config-if)> **no ip flow**

Arguments

Argument	Value	Description
direction	ingress	Collection of incoming traffic.
	egress	Collection of outgoing traffic.
	both	Collection of incoming and outgoing traffic both.

Example

```
(config-if)> ip flow ingress
Netflow::Manager: NetFlow collector is enabled on interface ►
"Home" in "ingress" direction.
```

```
(config-if)> ip flow egress
Netflow::Manager: NetFlow collector is enabled on interface ►
"Home" in "egress" direction.
```

```
(config-if)> ip flow both
Netflow::Manager: NetFlow collector is enabled on interface ►
"Home" in "both" direction.
```

History

Version	Description
2.11	The **interface ip flow** command has been introduced.

3.27.94 interface ip global

Description

Set property “global” with a parameter to the interface. This property is necessary to configure the default route, DynDNS-Client and NAT functioning. Can represent global-interfaces as leading to the global network (the Internet).

Property “global” affects the interface priority in setting the default route. The higher the priority the more desirable it is for the user to access the global network through the specified interface. Internet access backup (WAN backup) functionality is using priority “global”.

By default, setting is disabled.

Command with **no** prefix removes property.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

IP

Synopsis

(config-if)> **ip global** (‹*priority*› | **order** ‹*order*› | **auto**)

(config-if)> **no ip global**

Arguments

Argument	Value	Description
priority	*Integer*	Interface priority to configure the default route. Can take values from 1 to 65534.
order	*Integer*	Relative priority between interfaces. It can take values from 0 to 65534, but not more than the number of global interfaces.
auto	*Keyword*	Automatic priority calculation of the interface. The interface is located near the end of the list, but above order X.

Example

```
(config-if)> ip global 10
Network::Interface::IP: "L2TP0": global priority is 10.
```

```
(config-if)> ip global order 0
Network::Interface::IP: "L2TP0": order is 1.
```

```
(config-if)> ip global auto
Network::Interface::IP: Global priority recalculated.
```

```
(config-if)> no ip global
Network::Interface::IP: "L2TP0": global priority cleared.
```

History

Version	Description
2.00	The **interface ip global** command has been introduced.

2.09	The `order` and `auto` arguments were added.

3.27.95 interface ip mru

Description

Set the value of *MRU* to be transmitted to a remote node during establishing the *PPP* (*IPCP*) connection. By default, `1460` value is used.

Command with **no** prefix resets the *MRU* value to that which was before the first use of the command.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

PPP

Synopsis

```
(config-if)> ip mru ‹mru›
```

```
(config-if)> no ip mru
```

Arguments

Argument	Value	Description
`mru`	*Integer*	*MRU* value.

Example

```
(config-if)> ip mru 1492
Network::Interface::Ppp: "PPPoE0": MRU saved.
```

```
(config-if)> no ip mru
Network::Interface::Ppp: "PPPoE0": MRU reset to default.
```

History

Version	Description
2.00	The **interface ip mru** command has been introduced.

3.27.96 interface ip mtu

Description

Set the *MTU* value on the network interface. When establishing a connection via *PPP* (*IPCP*), packets with defined *MTU* size will be sent to the remote host, even if the host requested a lower *MTU* value.

Command with **no** prefix resets the *MTU* value to that which was before the first use of the command.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type IP

Synopsis

(config-if)> **ip mtu** *‹mtu›*

(config-if)> **no ip mtu**

Arguments

Argument	Value	Description
mtu	*Integer*	*MTU* value. Can take values from 64 to 65535 inclusively.

Example

```
(config-if)> ip mtu 1500
Network::Interface::Base: "GigabitEthernet1": static MTU is 1500.
```

```
(config-if)> no ip mtu
Network::Interface::Base: "GigabitEthernet1": static MTU reset ►
to default.
```

History

Version	Description
2.00	The **interface ip mtu** command has been introduced.

3.27.97 interface ip nat loopback

Description Enable reverse translation to send local requests to the local server from the Internet. By default, the setting is enabled for the Home segment interfaces (`private` and `protected` security levels).

Command with **no** prefix disables NAT loopback.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type IP

Synopsis

(config-if)> **ip nat loopback**

(config-if)> **no ip nat loopback**

Example

```
(config-if)> ip nat loopback
Network::StaticNat: NAT loopback is explicitly enabled on "Home".
```

```
(config-if)> no ip nat loopback
Network::StaticNat: NAT loopback is explicitly disabled on "Home".
```

History

Version	Description
2.11	The **ip nat loopback** command has been introduced.

3.27.98 interface ip remote

Description Set a remote peer static address.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type PPP

Synopsis

(config-if)> **ip remote** ‹*address*›

(config-if)> **no ip remote**

Arguments

Argument	Value	Description
address	*IP-address*	A remote peer address.

Example

```
(config-if)> ip remote 192.168.2.19
Network::Interface::Ppp: "L2TP0": remote address saved.
```

```
(config-if)> no ip remote
Network::Interface::Ppp: "L2TP0": remote address erased.
```

History

Version	Description
2.00	The **interface ip remote** command has been introduced.

3.27.99 interface ip tcp adjust-mss

Description Set the limit on the segment size of outgoing *TCP* sessions. If the *MSS* value, which is transmitted in the header of SYN-packets, exceeds the specified limit, command changes it. The command is applied to the interface and affects all outgoing *TCP* SYN packets.

Command with **no** prefix removes all limits from *MSS*.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type IP

Synopsis

(config-if)> **ip tcp adjust-mss** (**pmtu** | ‹*mss*›)

(config-if)> **no ip tcp adjust-mss**

Arguments

Argument	Value	Description
`pmtu`	*Keyword*	Set the upper limit of *MSS*, equal to the minimum *MTU* along the path to the remote peer.
`mss`	*Integer*	*MSS* upper limit.

Example

```
(config-if)> ip tcp adjust-mss pmtu
Network::Interface::Ip: "L2TP0": TCP-MSS adjustment enabled.
```

```
(config-if)> ip tcp adjust-mss 1300
Network::Interface::Ip: "L2TP0": TCP-MSS adjustment enabled.
```

```
(config-if)> no ip tcp adjust-mss
Network::Interface::Ip: "L2TP0": TCP-MSS adjustment disabled.
```

History

Version	Description
2.00	The **interface ip tcp adjust-mss** command has been introduced.

3.27.100 interface ipcp default-route

Description

Use the remote peer address as default gateway. By default, the setting is enabled.

Command with **no** prefix denies default gateway changing.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

PPP

Synopsis

(config-if)> **ipcp default-route**

(config-if)> **no ipcp default-route**

Example

```
(config-if)> ipcp default-route
Using peer as a default gateway.
```

History

Version	Description
2.00	The **interface ipcp default-route** command has been introduced.

3.27.101 interface ipcp dns-routes

Description Use routes which are received via *IPCP*. By default, the setting is enabled.

Command with **no** prefix removes the setting.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type PPP

Synopsis

(config-if)> **ipcp dns-routes**

(config-if)> **no ipcp dns-routes**

Example

```
(config-if)> ipcp dns-routes
DNS routes enabled
```

```
(config-if)> no ipcp dns-routes
DNS routes disabled
```

History

Version	Description
2.02	The **interface ipcp dns-routes** command has been introduced.

3.27.102 interface ipcp name-servers

Description Use *DNS* servers addresses which are received via *IPCP*. By default, the setting is enabled.

Command with **no** prefix removes the setting.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type PPP

Synopsis

(config-if)> **ipcp name-servers**

(config-if)> **no ipcp name-servers**

Example

```
(config-if)> ipcp name-servers
using remote name servers.
```

```
(config-if)> no ipcp name-servers
not using remote name servers.
```

History

Version	Description
2.00	The **interface ipcp name-servers** command has been introduced.

3.27.103 interface ipcp vj

Description

Enable compression of TCP/IP headers by Van Jacobson's method. By default, the setting is disabled.

Command with **no** prefix disables compression.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

PPP

Synopsis

(config-if)> **ipcp vj** [**cid**]

(config-if)> **no ipcp vj**

Arguments

Argument	Value	Description
`cid`	*Keyword*	Enable compression of Connection ID into headers.

Example

```
(config-if)> ipcp vj cid
VJ compression enabled.
```

```
(config-if)> no ipcp vj
VJ compression disabled.
```

History

Version	Description
2.03	The **interface ipcp vj** command has been introduced.

3.27.104 interface ipsec encryption-level

Description

Set encryption level for *IPSec* connection that is automatically associated with the tunnel. By default, the `normal` value is used.

A detailed description of each level is given in the Appendix.

Command with **no** prefix resets encryption level to default.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type Secure

Synopsis

(config-if)> **ipsec encryption-level** *‹level›*

(config-if)> **no ipsec encryption-level**

Arguments

Argument	Value	Description
level	weak	Weak level, DES and MD5 algorithms enabled.
	normal	Level is compatible with most systems, priority is given to AES128 and SHA1.
	normal-3des	Level is compatible with most systems, priority is given to 3DES and SHA1.
	strong	The strongest level, *PFS* is mandatory, priority is given to AES256 and SHA1.
	weak-pfs	The same as weak, but for the second phase *PFS* group 1 and 2 is enabled.
	normal-pfs	The same as normal, but for the second phase *PFS* group 2 and 5 is enabled.
	normal-3des-pfs	The same as normal-3des, but for the second phase *PFS* group 5 and 14 is enabled.
	high	A set of modern algorithms for external providers of VPN services.
	strong-aead	The strongest level, priority is given to AES256 and SHA1 with addition of *AEAD* algorithms.
	strong-aead-pfs	The strongest level, *PFS* is mandatory, priority is given to AES256 and SHA1 with addition of *AEAD* algorithms.

Example

```
(config-if)> ipsec encryption-level high
Network::Interface::Secure: "IKE0": security level is set to ►
"high".
```

```
(config-if)> no ipsec encryption-level
Network::Interface::Secure: "IKE0": security level was reset.
```

History

Version	Description
2.08	The **interface ipsec encryption-level** command has been introduced.
3.07	New levels of encryption has been added — high, strong-aead and strong-aead-pfs.

3.27.105 interface ipsec force-encaps

Description Enable support of *ESP* forced encapsulation in *UDP* for client tunnels. By default, the feature is disabled.

Command with **no** prefix cancels the setting.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type Secure

Synopsis

(config-if)> **ipsec force-encaps**

(config-if)> **no ipsec force-encaps**

Example

```
(config-if)> ipsec force-encaps
Network::Interface::Secure: Force ESP in UDP encapsulation ►
enabled.
```

```
(config-if)> no ipsec force-encaps
Network::Interface::Secure: Force ESP in UDP encapsulation ►
disabled.
```

History

Version	Description
2.12	The **interface ipsec force-encaps** command has been introduced.

3.27.106 interface ipsec ignore

Description Disable processing incoming *IKE* packets for *IPSec* service on the interface. By default the command is disabled.

Command with **no** prefix cancels the setting.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type Secure

Synopsis

(config-if)> **ipsec ignore**

(config-if)> **no ipsec ignore**

Example

```
(config-if)> ipsec ignore
IpSec::Manager: Interface "Gre0" added to IPsec ignore list.
```

```
(config-if)> no ipsec ignore
IpSec::Manager: Interface "Gre0" removed from IPsec ignore list.
```

History

Version	Description
2.10	The **interface ipsec ignore** command has been introduced.

3.27.107 interface ipsec ikev2

Description

Enable IKEv2 protocol for *IPSec* connection that is automatically associated with the tunnel. By default, IKEv1 is used.

Command with **no** prefix resets setting to default.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

Secure

Synopsis

(config-if)> **ipsec ikev2**

(config-if)> **no ipsec ikev2**

Example

```
(config-if)> ipsec ikev2
Network::Interface::Secure: IKEv2 is enabled.
```

```
(config-if)> no ipsec ikev2
Network::Interface::Secure: IKEv2 is disabled, enable IKEv1.
```

History

Version	Description
2.10	The **interface ipsec ikev2** command has been introduced.

3.27.108 interface ipsec nail-up

Description

Enable automatic changes of the secret keys for L2TP/IPsec, EoIP/IPsec, Gre/IPsec, IPIP/IPsec tunnels. By default, setting is enabled.

Command with **no** prefix disables the setting.

Prefix no

Yes

Change settings

Yes

Multiple input No

Interface type Secure

Synopsis

(config-if)> **ipsec nail-up**

(config-if)> **no ipsec nail-up**

Example

```
(config-if)> ipsec nail-up
Network::Interface::Secure: SA renegotiation enabled.
```

```
(config-if)> no ipsec nail-up
Network::Interface::Secure: SA renegotiation disabled.
```

History

Version	Description
2.12	The **interface ipsec nail-up** command has been introduced.

3.27.109 interface ipsec name-servers

Description Use *DNS*-server addresses which are received via IKEv1 or IKEv2 *IPSec*-server. By default, the function is enabled.

Command with **no** prefix denies using of *DNS*-server addresses which are received via IKEv1 and IKEv2 *IPSec*-server.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type Secure

Synopsis

(config-if)> **ipsec name-servers**

(config-if)> **no ipsec name-servers**

Example

```
(config-if)> ipsec name-servers
IpSec::Interface::Ike: "IKE0": automatic name servers via IKE ►
Configuration Payload are enabled.
```

```
(config-if)> no ipsec name-servers
IpSec::Interface::Ike: "IKE0": automatic name servers via IKE ►
Configuration Payload are disabled.
```

History

Version	Description
3.06	The **interface ipsec name-servers** command has been introduced.

3.27.110 interface ipsec preshared-key

Description

Set PSK key for *IPSec* connection that is automatically associated with the tunnel. Command also enables *IPSec* for this tunnel.

Command with **no** prefix resets the key.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

Secure

Synopsis

(config-if)> **ipsec preshared-key** *‹key›*

(config-if)> **no ipsec preshared-key**

Arguments

Argument	Value	Description
key	*String*	Secret PSK key value.

Example

```
(config-if)> ipsec preshared-key 12345678
Network::Interface::Secure: "Gre0": preshared key was set.
```

```
(config-if)> no ipsec preshared-key
Network::Interface::Secure: "Gre0": preshared key was reset.
```

History

Version	Description
2.08	The **interface ipsec preshared-key** command has been introduced.

3.27.111 interface ipsec proposal lifetime

Description

Set lifetime of *IPSec* transformation Phase1 on the interface. By default, the value `28800` is used.

Command with **no** prefix resets setting to default.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

Secure

Synopsis

(config-if)> **ipsec proposal lifetime** *‹lifetime›*

(config-if)> **no ipsec proposal lifetime**

Arguments

Argument	Value	Description
`lifetime`	*Integer*	Lifetime of *IPSec* transformation in seconds. Can take values from 60 to 2147483647.

Example

```
(config-if)> ipsec proposal lifetime 222222
Network::Interface::Secure: IPsec IKE proposal lifetime set to ►
222222 s.
```

```
(config-if)> no ipsec proposal lifetime
Network::Interface::Secure: IPsec IKE proposal lifetime reset ►
to 28800 s.
```

History

Version	Description
2.11	The **interface ipsec proposal lifetime** command has been introduced.

3.27.112 interface ipsec transform-set lifetime

Description

Set lifetime of *IPSec* transformation Phase2 on the interface. By default, the value 28800 is used.

Command with **no** prefix resets setting to default.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

Secure

Synopsis

(config-if)> **ipsec transform-set lifetime** ‹*lifetime*›

(config-if)> **no ipsec transform-set lifetime**

Arguments

Argument	Value	Description
`lifetime`	*Integer*	Lifetime of *IPSec* transformation in seconds. Can take values from 60 to 2147483647.

Example

```
(config-if)> ipsec transform-set lifetime 2222222
Network::Interface::Secure: IPsec ESP transform-set lifetime set ►
to 2222222 s.
```

```
(config-if)> no ipsec transform-set lifetime
Network::Interface::Secure: IPsec ESP transform-set lifetime ►
reset to 28800 s.
```

History

Version	Description
2.11	The **interface ipsec transform-set lifetime** command has been introduced.

3.27.113 interface ipv6 address

Description

Confgure an IPv6 address on the interface. If the argument is **auto**, address is autoconfigured. Passing a literal address as an argument will assign it statically.

Command with **no** prefix removes the setting.

Prefix no

Yes

Change settings

Yes

Multiple input

Yes

Synopsis

(config-if)> **ipv6 address** (*‹address›* | **auto**)

(config-if)> **no ipv6 address** [*‹address›* | **auto**]

Arguments

Argument	Value	Description
address	*IPv6-address*	Name server address.
auto	*Keyword*	Enable stateless autoconfiguration.

Example

```
(config-if)> ipv6 address 2001:db8::1
Static IPv6 address saved.
```

History

Version	Description
2.00	The **interface ipv6 address** command has been introduced.

3.27.114 interface ipv6 force-default

Description

Force the interface to be used as default IPv6 gateway. By default, the setting is disabled.

Command with **no** prefix removes the setting.

Prefix no

Yes

Change settings Yes

Multiple input No

Synopsis

(config-if)> **ipv6 force-default**

(config-if)> **no ipv6 force-default**

Example

```
(config-if)> ipv6 force-default
interface is forced to be the default IPv6 gateway
```

History

Version	Description
2.00	The **interface ipv6 force-default** command has been introduced.

3.27.115 interface ipv6 name-servers

Description Configure retrieval of *DNS* information. When **auto** is set, enables DHCPv6 name-server requests.

Command with **no** prefix removes the setting.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config-if)> **ipv6 name-servers** (**auto**)

(config-if)> **no ipv6 name-servers** [**auto**]

Arguments

Argument	Value	Description
auto	*Keyword*	Enable name-server autoconfiguration.

Example

```
(config-if)> ipv6 name-servers auto
Name servers provided by the interface network are accepted.
```

History

Version	Description
2.00	The **interface ipv6 name-servers** command has been introduced.

3.27.116 interface ipv6 prefix

Description Configure prefix delegation. When **auto** is set, prefix is requested via DHCPv6-PD.

Command with **no** prefix removes the setting.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config-if)> **ipv6 prefix** (*‹prefix›* | **auto**)

(config-if)> **no ipv6 prefix** [*‹prefix›* | **auto**]

Arguments

Argument	Value	Description
auto	*Keyword*	Enable prefix delegation.
prefix	*Prefix*	Manual input of prefix.

Example

```
(config-if)> ipv6 prefix 2001:db8:43:ab12::/64
Static IPv6 prefix added.
```

History

Version	Description
2.00	The **interface ipv6 prefix** command has been introduced.

3.27.117 interface ipv6cp

Description Enable *IPv6CP* support during establishing connection.

Command with **no** prefix disables *IPv6CP*.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type PPP

Synopsis

(config-if)> **ipv6cp**

(config-if)> **no ipv6cp**

Example

```
(config-if)> ipv6cp
IPv6CP enabled.
```

History

Version	Description
2.00	The **interface ipv6cp** command has been introduced.

3.27.118 interface lcp acfc

Description

Enable compression negotiation of the *Data Link Layer Address and Control fields*. By default, the feature is disabled.

Command with **no** prefix disables this option and all the remote peer requests for the *ACFC* negotiation will be rejected.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

PPP

Synopsis

(config-if)> **lcp acfc** [**cid**]

(config-if)> **no lcp acfc**

Arguments

Argument	Value	Description
cid	*Keyword*	Enable compression of Connection ID into headers.

Example

```
(config-if)> lcp acfc cid
ACFC compression enabled
```

```
(config-if)> no lcp acfc cid
ACFC compression disabled
```

History

Version	Description
2.03	The **interface lcp acfc** command has been introduced.

3.27.119 interface lcp echo

Description

Specify the testing rules of the *PPP* connection with *LCP* echo tools.

By default, `interval` is set to `30`, `count` is set to `3`.

Command with **no** prefix disables *LCP* echo.

Prefix no

Yes

Change settings

Yes

Multiple input No

Interface type PPP

Synopsis

(config-if)> **lcp echo** ‹*interval*› ‹*count*› [**adaptive**]

(config-if)> **no lcp echo**

Arguments

Argument	Value	Description
`interval`	*Integer*	Interval between sending *LCP* echo, in seconds. If within the specified time interval there is no *LCP* echo request from the remote location, the same request will be sent there asking for response *LCP* reply.
`count`	*Integer*	The number of consecutive requests *LCP* echo sent, for which no response *LCP* reply was received. If count of *LCP* echo requests goes unanswered, the connection is terminated.
`adaptive`	*Keyword*	Pppd will send LCP echo-request frames only if no traffic was received from the peer since the last echo-request was sent.

Example

```
(config-if)> lcp echo 20 2
Network::Interface::Ppp: "PPPoE0": LCP echo parameters updated.
```

```
(config-if)> no lcp echo
Network::Interface::Ppp: "PPPoE0": LCP echo disabled.
```

History

Version	Description
2.00	The **interface lcp echo** command has been introduced.
2.06	The `adaptive` keyword has been added.

3.27.120 interface lcp pfc

Description Enable compression negotiation of the *PPP Protocol field*. By default, the feature is disabled.

Command with **no** prefix disables this option and all the remote peer requests for the *PFC* negotiation will be rejected.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type PPP

Synopsis

(config-if)> **lcp pfc** [**cid**]

(config-if)> **no lcp pfc**

Arguments

Argument	Value	Description
cid	*Keyword*	Enable compression of Connection ID into headers.

Example

```
(config-if)> lcp pfc cid
PFC compression enabled
```

```
(config-if)> no lcp pfc cid
PFC compression disabled
```

History

Version	Description
2.03	The **interface lcp pfc** command has been introduced.

3.27.121 interface led wan

Description

Display the interface status by means of LED. SelectedWan control should be chosen with **system led** command. By default, function is disabled.

Command with **no** prefix disables the feature.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config-if)> **led wan**

(config-if)> **no led wan**

Example

```
(config-if)> led wan
Network::Interface::Led: Selected WAN GigabitEthernet1.
```

```
(config-if)> no led wan
Network::Interface::Led: Selected no WAN.
```

History

Version	Description
2.08	The **interface led wan** command has been introduced.

3.27.122 interface lldp disable

Description

Disable *LLDP* agent on interface. By default, the feature is enabled.

Command with **no** prefix enables *LLDP* agent.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config-if)> **lldp disable**

(config-if)> **no lldp disable**

Example

```
(config-if)> lldp disable
Network::DiscoveryManager: LLDP agent is disabled on interface ►
"ISP".
```

```
(config-if)> no lldp disable
Network::DiscoveryManager: LLDP agent is enabled on interface ►
"ISP".
```

History

Version	Description
2.11	The **interface lldp disable** command has been introduced.

3.27.123 interface mac access-list address

Description Add a MAC-address to the permit/deny filtering list of the interface. Type of access list is set with **interface mac access-list type** command.

Command with **no** prefix removes the specified MAC-address from the *ACL*.

Prefix no Yes

Change settings Yes

Multiple input Yes

Interface type Access Point

Synopsis

(config-if)> **mac access-list address** *‹address›*

(config-if)> **no mac access-list address** *‹address›*

Arguments

Argument	Value	Description
address	*MAC-address*	A MAC-address to be added to the *ACL*.

Example

```
(config-if)> mac access-list address 64:a2:f9:53:b2:12
Network::Interface::Ethernet: "WifiMaster0/AccessPoint1": added ►
64:a2:f9:53:b2:12 to the ACL.
```

```
(config-if)> no mac access-list address 64:a2:f9:53:b2:12
Network::Interface::Ethernet: "WifiMaster0/AccessPoint1": removed ►
64:a2:f9:53:b2:12 from the ACL.
```

```
(config-if)>  no mac access-list address
Network::Interface::Ethernet: "WifiMaster0/AccessPoint1": ACL ►
cleared.
```

History

Version	Description
2.00	The **interface mac access-list address** command has been introduced.

3.27.124 interface mac access-list type

Description Set the type for filtering list of the interface. Type is not defined by default (none value assigned).

Prefix no No

Change settings Yes

Multiple input No

Interface type Access Point

Synopsis

(config-if)> **mac access-list type** *‹type›*

Arguments

Argument	Value	Description
type	none	Type of filtering list is not defined.
	permit	Only approved MAC-addresses will be added to the list.
	deny	Only restricted MAC-addresses will be added to the list.

Example

```
(config-if)> mac access-list type permit
Network::Interface::Ethernet: "WifiMaster0/AccessPoint1": ACL ►
type changed to permit.
```

History

Version	Description
2.00	The **interface mac access-list type** command has been introduced.

3.27.125 interface mac address

Description

Set the MAC-address to the specified network interface. Address is specified in hexadecimal format `00:00:00:00:00:00`. The command allows one to assign arbitrary address, but warns the user if the new address “multicast” bit is set or “OUI enforced” bit is cleared.

Command with **no** prefix resets the original MAC-addresses on the interface.

Warning: Change MAC-address on Wi-Fi interface is prohibited.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

MAC

Synopsis

(config-if)> **mac address** *‹mac›*

(config-if)> **no mac address**

Arguments

Argument	Value	Description
mac	*MAC-address*	New MAC-address of the interface.

Example

```
(config-if)> mac address 3C:1F:6E:2A:1C:BA
```

```
(config-if)> no mac address
```

History

Version	Description
2.00	The **interface mac address** command has been introduced.

3.27.126 interface mac address factory

Description

Set the factory MAC-address to the interface.

Prefix no

No

Change settings

Yes

Multiple input

No

Interface type

MAC

Synopsis

(config-if)> **mac address factory** *‹name›*

Arguments

Argument	Value	Description
name	lan	"LAN" MAC-address will be assigned to the interface.
	wan	"WAN" MAC-address will be assigned to the interface.
	wlan5	"WLAN5" MAC-address will be assigned to the interface.

Example

```
(config-if)> mac address factory lan
Core::System::UConfig: done.
```

History

Version	Description
2.00	The **interface mac address factory** command has been introduced.

3.27.127 interface mac band

Description

Bind a registered host to a 2.4 GHz or 5 GHz frequency band.

Command with **no** prefix removes the binding. If you use no argument, the entire list of bindings will be cleared.

Prefix no

Yes

Change settings

Yes

Multiple input

Yes

Interface type

Bridge

Synopsis

(config-if)> **mac band** *‹mac›* *‹band›*

(config-if)> **no mac band** [*‹mac›*]

Arguments

Argument	Value	Description
mac	*MAC-address*	MAC-address of the registered client.
band	0	2,4 GHz band.
	1	5 GHz band.

Example

```
(config-if)> mac band c0:b8:83:c2:cb:11 0
Network::Interface::Rtx::MacBand: "Bridge0": bound ►
c0:b8:83:c2:cb:11 to 2.4 GHz.
```

```
(config-if)> mac band c0:b8:83:c2:cb:11 1
Network::Interface::Rtx::MacBand: "Bridge0": bound ►
c0:b8:83:c2:cb:11 to 5 GHz.
```

```
(config-if)> no mac band c0:b8:83:c2:cb:85
Network::Interface::Rtx::MacBand: "Bridge0": unbound ►
c0:b8:83:c2:cb:85 from 2.4 GHz.
```

```
(config-if)> no mac band
Network::Interface::Rtx::MacBand: Unbound all hosts.
```

History

Version	Description
3.05	The **interface mac band** command has been introduced.

3.27.128 interface mac bssid

Description Set the new MAC-address of access point 2,4 GHz or 5 GHz in WISP mode.

Command with **no** prefix returns the original MAC-address to the interface.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type WifiStation

Synopsis

(config-if)> **mac bssid** ‹*bssid*›

(config-if)> **no mac bssid**

Arguments

Argument	Value	Description
bssid	*MAC-address*	New MAC-address of the acess point interface.

Example

```
(config-if)> mac bssid 56:ff:20:00:1e:11
Network::Interface::WifiStation: BSSID set to 56:ff:20:00:1e:11.
```

```
(config-if)> no mac bssid
Network::Interface::WifiStation: BSSID cleared.
```

History

Version	Description
2.13	The **interface mac bssid** command has been introduced.

3.27.129 interface mac clone

Description Clone the MAC-address from the operator's PC to the interface.

Prefix no No

Change settings Yes

Multiple input No

Interface type MAC, IP

Synopsis

(config-if)> **mac clone**

Example

```
(config-if)> mac clone
```

History

Version	Description
2.00	The **interface mac clone** command has been introduced.

3.27.130 interface openvpn accept-routes

Description Enable receiving routes from a remote side via OpenVPN.

Command with **no** prefix disables the feature.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type OpenVPN

Synopsis

(config-if)> **openvpn accept-routes**

(config-if)> **no openvpn accept-routes**

Example

```
(config-if)> openvpn accept-routes
Network::Interface::OpenVpn: "OpenVPN0": enable automatic routes ►
accept via tunnel.
```

```
(config-if)> no openvpn accept-routes
Network::Interface::OpenVpn: "OpenVPN0": disable automatic routes ►
accept via tunnel.
```

History

Version	Description
2.10	The **interface openvpn accept-routes** command has been introduced.

3.27.131 interface openvpn connect

Description Set interface for OpenVPN connection. If you use no argument, connection is set via any interface.

Prefix no No

Change settings Yes

Multiple input No

Interface type OpenVPN

Synopsis

(config-if)> **openvpn connect** [**via** *‹via›*]

(config-if)> **openvpn connect**

Arguments

Argument	Value	Description
via	*Interface name*	Full interface name or an alias.

Example

```
(config-if)> openvpn connect via ISP
Network::Interface::OpenVpn: "OpenVPN0": set connection via ISP.
```

```
(config-if)> openvpn connect
Network::Interface::OpenVpn: "OpenVPN0": set connection via any ►
interface.
```

History

Version	Description
2.10	The **interface openvpn connect** command has been introduced.

3.27.132 interface openvpn name-servers

Description Use *DNS*-server addresses which are received via OpenVPN-server. By default, the function is enabled.

Command with **no** prefix denies using of *DNS*-server addresses which are received via OpenVPN-server.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type OpenVPN

Synopsis

(config-if)> **openvpn name-servers**

(config-if)> **no openvpn name-servers**

Example

```
(config-if)> openvpn name-servers
Network::Interface::OpenVpn: "OpenVPN0": automatic name servers ►
via tunnel are enabled.
```

```
(config-if)> no openvpn name-servers
Network::Interface::OpenVpn: "OpenVPN0": automatic name servers ►
via tunnel are disabled.
```

History

Version	Description
3.06	The **interface openvpn name-servers** command has been introduced.

3.27.133 interface peer

Description

Specify ID of the remote peer to which the *PPP* connection will be used. A more precise meaning of configurition depends on interface type. For example, for PPPoE the **interface peer** command specifies the name of access hub, for PPTP — remote host name or IP-address, and for SSTP — specifies a remote server with port 443 or another.

Command with **no** prefix cancels the setting.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

PPP

Synopsis

(config-if)> **peer** *‹peer›*

(config-if)> **no peer**

Arguments

Argument	Value	Description
peer	*String*	Remote connection point ID or remote server address `host.example.net:port`. By default, port number is 443.

Example

```
(config-if)> peer 111
```

```
(config-if)> peer host.example.net:5555
```

History

Version	Description
2.00	The **interface peer** command has been introduced.
2.12	Added the ability to change the port of a remote server.

3.27.134 interface peer-isolation

Description Enable the isolation of wireless clients in the Home segment. The setting applies on the Bridge interface and has an effect for all access points included in it. Also, it blocks traffic from wireless clients inside the L2 network.

Command with **no** prefix cancels the setting.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type Bridge

Synopsis

(config-if)> **peer-isolation**

(config-if)> **no peer-isolation**

Example

```
(config-if)> peer-isolation
Network::Interface::Ethernet: "Bridge0": peer isolation enabled.
```

```
(config-if)> no peer-isolation
Network::Interface::Ethernet: "Bridge0": peer isolation disabled.
```

History

Version	Description
2.10	The **interface peer-isolation** command has been introduced.

3.27.135 interface ping-check profile

Description Assign *Ping Check* profile to the interface.

Command with **no** prefix cancels the setting.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config-if)> **ping-check profile** ‹*profile*›

(config-if)> **no ping-check profile**

Arguments

Argument	Value	Description
`profile`	*String*	Profile name to assign.

Example

```
(config-if)> ping-check profile test
PingCheck::Client: Set ping-check profile for interface "ISP".
```

```
(config-if)> no ping-check profile
PingCheck::Client: Reset ping-check profile for interface "ISP".
```

History

Version	Description
2.04	The **interface ping-check profile** command has been introduced.

3.27.136 interface ping-check restart

Description

Enable interface restart if *Ping Check* is triggered (Internet is not available on `interface`). By default the function is disabled.

Command with **no** prefix disables the function.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config-if)> **ping-check restart** [‹*interface*›]

(config-if)> **no ping-check restart**

Arguments

Argument	Value	Description
`interface`	*Interface name*	Full name or alias of the interface to be restarted when the *Ping Check* on the binded interface is triggered. If this argument is not specified, the interface binded with *Ping Check* profile will be restarted.

Example

```
(config-if)> ping-check restart
PingCheck::Client: Enabled "PPPoE0" interface restart.
```

```
(config-if)>  ping-check restart ISP
PingCheck::Client: Enabled "ISP" interface restart for "PPPoE0".
```

```
(config-if)> no ping-check restart
PingCheck::Client: Remove restart settings for "PPPoE0".
```

History

Version	Description
3.04	The **interface ping-check restart** command has been introduced.

3.27.137 interface pmf

Description Enable *PMF* functionality.

Command with **no** prefix disables the feature.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type WiFi

Synopsis

(config-if)> **pmf**

(config-if)> **no pmf**

Example

```
(config-if)> pmf
Network::Interface::Rtx::WifiStation: "WifiMaster0/WifiStation0": ►
PMF enabled.
```

```
(config-if)> no pmf
Network::Interface::Rtx::WifiStation: "WifiMaster0/WifiStation0": ►
PMF disabled.
```

History

Version	Description
2.09	The **interface pmf** command has been introduced.

3.27.138 interface power

Description Set the transmitter power for the radio interface. Transmitter power is limited by the hardware capabilities and state laws applicable to radio broadcast. This command allows one to only reduce the power of the transmitter relative to its maximum power, such as to decrease potential interference with other devices in this range/band. By default, the setting value of the power is set to `100`.

Prefix no No

Change settings Yes

Multiple input No

Interface type Radio

Synopsis

(config-if)> **power** ‹*power*›

Arguments

Argument	Value	Description
power	*Integer*	The transmitter power as the percentage of the maximum power (from 1 to 100).

Example

```
(config-if)>  power 1
Network::Interface::Rtx::WifiMaster: "WifiMaster0": TX power ►
level set.
```

History

Version	Description
2.00	The **interface power** command has been introduced.

3.27.139 interface pppoe service

Description Specify PPPoE service. If service is not defined, then PPPoE-client will be connected to an arbitrary service.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type PPPoE

Synopsis

(config-if)> **pppoe service** *‹service›*

(config-if)> **no pppoe service**

Arguments

Argument	Value	Description
service	*String*	Name of PPPoE service.

Example

```
(config-if)> pppoe service TEST
Network::Interface::Pppoe: "PPPoE0": service set.
```

```
(config-if)> no pppoe service
Network::Interface::Pppoe: "PPPoE0": service removed.
```

History

Version	Description
2.05	The **interface pppoe service** command has been introduced.

3.27.140 interface pppoe session auto-cleanup

Description Enable sending a PADT packet for the unfinished PPPoE session. By default the option is enabled.

Command with **no** prefix disables sending a PADT packet.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type PPPoE

Synopsis

(config-if)> **pppoe session auto-cleanup**

(config-if)> **no pppoe session auto-cleanup**

Example

```
(config-if)> pppoe session auto-cleanup
Network::Interface::Ppp: "PPPoE0": enabled session auto cleanup.
```

```
(config-if)> no pppoe session auto-cleanup
Network::Interface::Ppp: "PPPoE0": disabled session auto cleanup.
```

History

Version	Description
3.03	The **interface pppoe session auto-cleanup** command has been introduced.

3.27.141 interface preamble-short

Description Use short *preamble*. By default, the setting is disabled.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type Radio

Synopsis

(config-if)> **preamble-short**

(config-if)> **no preamble-short**

Example

```
(config-if)> preamble-short
Network::Interface::Rtx::WifiMaster: "WifiMaster0": short ►
preamble enabled.
```

```
(config-if)> no preamble-short
Network::Interface::Rtx::WifiMaster: "WifiMaster0": short ►
preamble disabled.
```

History

Version	Description
2.00	The **interface preamble-short** command has been introduced.

3.27.142 interface reconnect-delay

Description Set the period of time between reconnection attempts. By default, value 3 is used.

Command with **no** prefix resets setting to default.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type PPP

Synopsis

(config-if)> **reconnect-delay** *‹sec›*

(config-if)> **no reconnect-delay**

Arguments

Argument	Value	Description
sec	*Integer*	Value of time in seconds. Can take values from 3 to 600.

Example

```
(config-if)> reconnect-delay 3
Network::Interface::Ppp: "PPTP1": reconnect delay set to 3 ►
seconds.
```

```
(config-if)> no reconnect-delay
Network::Interface::Ppp: "PPTP0": reconnect delay reset to ►
default.
```

History

Version	Description
2.11	The **interface reconnect-delay** command has been introduced.

3.27.143 interface rekey-interval

Description Set the period of time between automatic changes of the secret keys, which all devices on the network share. By default, 86400 value is used.

Command with **no** prefix disables keys changing.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type WiFi

Synopsis

(config-if)> **rekey-interval** *‹interval›*

(config-if)> **no rekey-interval**

Arguments

Argument	Value	Description
interval	*Integer*	Value of rekey interval in seconds.

Example

```
(config-if)> rekey-interval 3000
Network::Interface::Rtx::WifiMaster: "WifiMaster0": rekey ►
interval is 3000 sec.
```

```
(config-if)> no rekey-interval
Network::Interface::Rtx::WifiMaster: "WifiMaster0": rekey ►
interval disabled.
```

History

Version	Description
2.06	The **interface rekey-interval** command has been introduced.
2.15	Added default value of rekey interval 3600 sec.
3.04	Default value of rekey interval is changed to 86400 sec.

3.27.144 interface rename

Description Assign arbitrary name to the specified network interface. The interface can be referred to by the new name just like by ID.

Command with **no** prefix removes the setting.

Warning: Do not rename Home interface. This can cause unpredictable system errors.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config-if)> **rename** *‹rename›*

(config-if)> **no rename**

Arguments

Argument	Value	Description
rename	*String*	New interface name.

Example

```
(config-if)> rename PPPoE1
Network::Interface::Base: "PPPoE0": renamed to "PPPoE1".
```

```
(config-if)> no rename
Network::Interface::Base: "PPPoE0": name cleared.
```

History

Version	Description
2.08	The **interface rename** command has been introduced.

3.27.145 interface rf e2p set

Description Change the memory cell value of calibration data at *offset* by *value* for the specified interface.

Prefix no No

Change settings No

Multiple input No

Interface type Radio

Synopsis

(config-if) **rf e2p set** ‹*offset*› ‹*value*›

Arguments

Argument	Value	Description
offset	*Hexadecimal number*	Memory cell location. Can take values from 1E0 to 1FE.
value	*Hexadecimal number*	Value to be set. Can take values from 0 to FFFF.

Example

```
(config-if)> rf e2p set 1f6 0
Network::Interface::Rtx::WifiMaster: EEPROM [0x01F6]:0000 set.
```

History

Version	Description
2.04	The **interface rf e2p set** command has been introduced.

3.27.146 interface role

Description Set a role for the interface. Multiple roles can be assigned to one interface. Command is used for correct view of VLAN connections in the web interface.

Command with **no** prefix removes the role. If you use no arguments, the entire list of roles will be removed.

Prefix no Yes

Change settings No

Multiple input Yes

Synopsis

(config-if)> **role** ‹*role*› [**for** ‹*ifor*›]

(config-if)> **no role** [*role*]

Arguments

Argument	Value	Description
role	inet	Interface is used for Internet connection.
	iptv	Interface is used for IPTV service.
	voip	Interface is used for VoIP service.
	misc	Interface is used for IP Policy.
ifor	*Interface name*	Full interface name or an alias.

Example

```
(config-if)> role iptv for GigabitEthernet1
Network::Interface::Base: "GigabitEthernet1": assigned role ►
"iptv" for GigabitEthernet1.
```

```
(config-if)> no role iptv for GigabitEthernet1
Network::Interface::Base: "GigabitEthernet1": deleted role "iptv".
```

```
(config-if)> no role
Network::Interface::Base: "GigabitEthernet1": deleted all roles.
```

History

Version	Description
2.06	The **interface role** command has been introduced.
2.10	Argument misc was added.

3.27.147 interface rrm

Description Enable *RRM* for search of nearby APs according to IEEE 802.11k standard in order to provide this AP list to the subscriber device by request. By default, the option is disabled.

Command with **no** prefix removes the setting.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type AccessPoint

Synopsis

(config-if)> **rrm**

(config-if)> **no rrm**

Example

```
(config-if)> rrm
Network::Interface::Rtx::AccessPoint: "WifiMaster0/AccessPoint0": ►
RRM enabled.
```

```
(config-if)> no rrm
Network::Interface::Rtx::AccessPoint: "WifiMaster0/AccessPoint0": ►
RRM disabled.
```

History

Version	Description
2.13	The **interface rrm** command has been introduced.

3.27.148 interface schedule

Description Assign a schedule to the interface. Schedule must be created and customized with **schedule action** command before execution.

Command with **no** prefix unbinds the schedule.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config-if)> **schedule** ‹*schedule*›

(config-if)> **no schedule**

Arguments

Argument	Value	Description
schedule	*Schedule name*	The name of the schedule that was created with **schedule** group of commands.

Example

```
(config-if)> schedule WIFI
Network::Interface::Base: "WifiMaster0": schedule is "WiFi".
```

```
(config-if)> no schedule
Network::Interface::Base: "WifiMaster0": schedule cleared.
```

History

Version	Description
2.06	The **interface schedule** command has been introduced.

3.27.149 interface security-level

Description

Specify the interface security level. The security levels define the firewall logic:

- Allow establishing `private` → `public` connections.
- Prohibit establishing connections coming to the `public` interface, i. e. in the direction `public` → `private` and `public` → `public`.
- The device itself accepts network connections (allows control) only from `private` interfaces.
- Data transfer between `private` interfaces can be allowed or disallowed depending on the **isolate-private** global parameter.
- `protected` interfaces have no access to device and to other `private`/`protected` subnetworks, but they have access to `public` interfaces and to the internet. The device provides only DHCP and DNS services to the protected segments.
- Data transfer from `private` to `protected` interfaces is forbidden by default. To allow such connection use the **no isolate-private** command.

Note: By default, to all newly created interfaces `public` security level assigned.

Access lists **access-list** have higher priority than the security levels, so they can be used to set additional rules of packet filtering.

Prefix no No

Change settings Yes

Multiple input No

Interface type IP

Synopsis

(config-if)> **security-level** (**public** | **private** | **protected**)

Example

Despite the fact that there is no functionality to disable the firewall completely, it is possible to disable it for particular directions. Suppose that it is necessary to allow data transfer between the “home” network `Home` and global network `PPPoE0`. To accomplish that, to both interfaces must be assigned `private` security level and function **isolate-private** must be disabled.

```
(config)> interface Home security-level private
Network::Interface::IP: "Bridge0": security level set to ►
"private".

(config)> interface PPPoE0 security-level private
Network::Interface::IP: "PPPoE0": security level set to "private".

(config)> no isolate-private
Netfilter::Manager: Private networks not isolated.
```

Note: The firewall and the address translation — are the functions designed to solve fundamentally different problems. Enabling NAT between `Home` and `PPPoE0` interfaces in the configuration shown above, does not prohibit access to the network `Home` from the global network. Even as the address translation is enabled by command **`ip nat Home`**, the packets from `PPPoE0` will get to `Home` network.

History

Version	Description
2.00	The **interface security-level** command has been introduced.
2.06	The `protected` parameter was added.

3.27.150 interface speed

Description Configure the speed of the Ethernet interface. By default, `auto` value is set.

Command with **no** prefix resets setting to default.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type Ethernet

Synopsis

(config-if)> **speed** (**10** | **100** | **1000** | **auto**)

(config-if)> **no speed**

Arguments

Argument	Value	Description
`10`	*Keyword*	Connection speed in Mbit/s.
`100`		
`1000`		
`auto`	*Keyword*	Automatical speed configuration.

Example

```
(config-if)> speed 1000
Network::Interface::Ethernet: "GigabitEthernet1/0": speed set ►
to 1000.
```

```
(config-if)> no speed
Network::Interface::Ethernet: "GigabitEthernet1/0": speed reset ►
to default (auto-negotiation).
```

History

Version	Description
2.06.B.1	The **interface speed** command has been introduced.

3.27.151 interface speed nonegotiate

Description

Disable autonegotiation. By default, autonegotiation is enabled.

Command with **no** prefix enables autonegotiation.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

Ethernet

Synopsis

(config-if)> **speed nonegotiate**

(config-if)> **no speed nonegotiate**

Example

```
(config-if)> speed nonegotiate
Network::Interface::Ethernet: "GigabitEthernet1/0": ►
autonegotiation will be disabled for fixed speed.
```

```
(config-if)> no speed nonegotiate
Network::Interface::Ethernet: "GigabitEthernet1/0": ►
autonegotiation enabled..
```

History

Version	Description
2.08	The **interface speed nonegotiate** command has been introduced.

3.27.152 interface ssid

Description

Specify the wireless network name (SSID) for `WiFiStation` and `AccessPoint` interfaces. Depending on the interface type, the SSID value is processed differently.

- For AccessPoint, the SSID is a necessary setting, without which the connection will not be accepted.
- For the WiFiStation SSID determines which access point WiFiStation will connect to. Without a specified SSID, WiFiStation can connect to any available wireless network at its discretion.

Command with **no** prefix resets network name to default.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type WiFi

Synopsis

(config-if)> **ssid** *‹ssid›*

(config-if)> **no ssid**

Arguments

Argument	Value	Description
ssid	*String*	Wireless Network Name (SSID).

Example

```
(config-if)> ssid MYNETWORK
Network::Interface::Wireless: "WifiMaster0/AccessPoint0": SSID ►
saved.
```

```
(config-if)> no ssid
Network::Interface::Rtx::AccessPoint: "WifiMaster0/AccessPoint0": ►
SSID reset.
```

History

Version	Description
2.00	The **interface ssid** command has been introduced.

3.27.153 interface switchport access

Description Set the port *VLAN* ID for access mode. Allows to transfer frames of the specified *VLAN* to the port and remove *VLAN* marker from the transferred frames.

Command with **no** prefix removes the setting.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type Port

Synopsis

(config-if)> **switchport access vlan** *‹vid›*

(config-if)> **no switchport access vlan**

Arguments

Argument	Value	Description
vid	*Integer*	Access *VLAN* ID. Can take values from 1 to 4094 inclusively.

Example

```
(config-if)> switchport access vlan 1
Network::Interface::Switch: "FastEthernet0/0": set access VLAN ►
ID: 1.
```

History

Version	Description
2.06	The **interface switchport access** command has been introduced.

3.27.154 interface switchport friend

Description

Configure undirectional *VLAN* for multicast traffic in addition to access *VLAN*. Port can be a member of one access *VLAN*. This command enables forwarding of downstream traffic from a different *VLAN* (called "friend"). Friend packets are transmitted without a tag.

Command with **no** prefix removes the setting.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

Port

Synopsis

(config-if)> **switchport friend vlan** *‹vid›*

(config-if)> **no switchport friend vlan**

Arguments

Argument	Value	Description
vid	*Integer*	Friend *VLAN* ID. Can take values from 1 to 4094 inclusively.

Example

```
(config-if)> switchport friend vlan 2
Network::Interface::Switch: "FastEthernet0/0": set friend VLAN ►
ID: 2.
```

History

Version	Description
2.06	The **interface switchport friend** command has been introduced.

3.27.155 interface switchport mode

Description

Set `access` or `trunk` mode for *VLAN*. By default, `access` mode is set.

Command with **no** prefix resets setting to default.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

Port

Synopsis

(config-if)> **switchport mode** [(**access** [**q-in-q**]) | **trunk**]

(config-if)> **no switchport mode**

Arguments

Argument	Value	Description
`mode`	`access`	Enable the access mode to a *VLAN*, that is the mode when only the untagged frames pass through the port. The incoming frames get tagged with the PVID marker, which is set with **switchport access** command. The port is an output one only for *VLAN* with PVID ID. Once a frame is transferred to the port, the *VLAN* marker gets removed.
	`trunk`	Enable the *VLAN* trunk mode, that is the mode when frames belonging to several VLANs get transmitted through the port. In this case each frame gets tagged. The list of IDs of *VLAN* networks that include the port is set with **switchport trunk** command.
`q-in-q`	*Keyword*	Enable double tagging.

Example

```
(config-if)> switchport mode access
Network::Interface::Switch: "FastEthernet0/1": access mode ►
enabled.
```

History

Version	Description
2.06	The **interface switchport mode** command has been introduced.

3.27.156 interface switchport trunk

Description

Add a port to the *VLAN*. Allows receiving and transmitting of the given *VLAN* frames to the port, such that VLAN marker from the transmitted frames is not removed. In the `trunk` mode it is allowed to add a port to several VLANs.

Command with **no** prefix removes the port from the specified *VLAN*. If you use no argument, the port will be removed from all the VLANs.

Prefix no

Yes

Change settings

Yes

Multiple input

Yes

Interface type

Port

Synopsis

(config-if)> **switchport trunk vlan** *‹vid›*

(config-if)> **no switchport trunk vlan** [*vid*]

Arguments

Argument	Value	Description
vid	*Integer*	*VLAN* ID. Can take values from 1 to 4094 inclusively.

Example

```
(config-if)> switchport trunk vlan 100
Network::Interface::Switch: "FastEthernet0/1": set trunk VLAN ►
ID: 100.
```

History

Version	Description
2.06	The **interface switchport trunk** command has been introduced.

3.27.157 interface traffic-shape

Description

Set the limit of data rate on a specified interface in both directions. By default speed is not limited.

Command with **no** prefix removes the setting.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config-if)> **traffic-shape rate** *‹rate›* [**asymmetric** *‹upstream-rate›*] [**schedule** *‹schedule›*]

(config-if)> **no traffic-shape**

Arguments

Argument	Value	Description
rate	*Integer*	Value of data download rate in Kbps. Limit sould be in the range from 64 Kbps to 1 Gbps.
upstream-rate	*Integer*	Data upload rate in Kbps. Value can be in the range from 64 Kbps to 1 Gbps.
schedule	*Schedule name*	The name of the schedule that was created with **schedule** group of commands.

Example

```
(config-if)> traffic-shape rate 5000
TrafficControl::Manager: "Bridge0" interface rate limited to ►
5000 kbit/s.
```

```
(config-if)> traffic-shape rate 5000 asymmetric 500
TrafficControl::Manager: "Bridge0" interface rate limited to ►
5000/500 kbit/s.
```

```
(config-if)> no traffic-shape
TrafficControl::Manager: Rate limit removed for "Bridge0" ►
interface.
```

History

Version	Description
2.05	The **interface traffic-shape** command has been introduced.
3.04	The **upstream-rate** argument was added.

3.27.158 interface tunnel destination

Description

Set the remote end of tunnel. If it is used in conjunction with an automatic *IPSec* connection associated with the tunnel, remote host becomes the initiator of an *IPSec* connection.

Command with **no** prefix resets the setting.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

Tunnel

Synopsis

(config-if)> **tunnel destination** ‹*destination*›

(config-if)> **no tunnel destination**

Arguments

Argument	Value	Description
`destination`	*String*	IP address or domain name of the remote host.

Example

```
(config-if)> tunnel destination example.net
Network::Interface::Tunnel: "Gre0": destination set to ►
example.net.
```

```
(config-if)> no tunnel destination
Network::Interface::Tunnel: "Gre0": destination was reset.
```

History

Version	Description
2.08	The **interface tunnel destination** command has been introduced.

3.27.159 interface tunnel eoip id

Description

Set identifier of EoIP tunnel.

Command with **no** prefix resets the setting.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

Eoip

Synopsis

(config-if)> **tunnel eoip id** ‹*id*›

(config-if)> **no tunnel eoip id**

Arguments

Argument	Value	Description
`id`	*Integer*	Tunnel ID.

Example

```
(config-if)> tunnel eoip id 50
Network::Interface::Tunnel: "Gre0": eoip id interface set to auto.
```

```
(config-if)> no tunnel eoip id
Network::Interface::Tunnel: "Gre0": eoip id was reset.
```

History

Version	Description
2.08	The **interface tunnel eoip id** command has been introduced.

3.27.160 interface tunnel gre keepalive

Description

Enable support of Cisco-like keepalive for GRE tunnel. By default, `interval` is set to 5, `count` is set to 3.

Command with **no** prefix removes the setting.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

Tunnel

Synopsis

(config-if)> **tunnel gre keepalive** ‹*interval*› [*count*]

(config-if)> **no tunnel gre keepalive**

Arguments

Argument	Value	Description
`interval`	*Integer*	The interval of sending keepalive packets in seconds. Can take values from `0` to `60`. If `0` is set, then GRE keepalive replies is enabled only and the router will not react on the tunnel state change.
`count`	*Integer*	Number of attempts to send keepalive packets. Can take values from `1` to `20`.

Example

```
(config-if)> tunnel gre keepalive 10 7
Network::Interface::Gre: "Gre0": set GRE keepalive to 10 s (7 ►
retries).
```

```
(config-if)> no tunnel gre keepalive
Network::Interface::Gre: "Gre0": disable GRE keepalive.
```

```
(config-if)> tunnel gre keepalive 0
Network::Interface::Gre: "Gre0": enable only GRE keepalive ►
replies.
```

History

Version	Description
2.10	The **interface tunnel gre keepalive** command has been introduced.

3.27.161 interface tunnel source

Description

Set the local end of tunnel. If it is used in conjunction with an automatic *IPSec* connection associated with the tunnel, then the reception mode of IPsec IKE connections is activated to establish a secure tunnel.

Command with **no** prefix resets the setting.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type Tunnel

Synopsis

(config-if)> **tunnel source** (**auto** | *‹interface›* | *‹address›*)

(config-if)> **no tunnel source**

Arguments

Argument	Value	Description
auto	*Keyword*	Set the current working WAN interface.
interface	*Interface name*	Full interface name or an alias.
address	*IP-address*	Local IP-adress of the tunnel.

Example

```
(config-if)> tunnel source auto
Network::Interface::Tunnel: "Gre0": source interface set to auto.
```

```
(config-if)> no tunnel source
Network::Interface::Tunnel: "Gre0": source was reset.
```

History

Version	Description
2.08	The **interface tunnel source** command has been introduced.
2.09	The **auto** argument has been added.

3.27.162 interface tx-burst

Description Enable Wi-Fi packet aggregation (Tx Burst). By default, the setting is disabled.

Command with **no** prefix disables the setting.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config-if)> **tx-burst**

(config-if)> **no tx-burst**

Example

```
(config-if)> tx-burst
Network::Interface::Rtx::WifiMaster: Tx Burst enabled.
```

History

Version	Description
2.07	The **interface tx-burst** command has been introduced.

3.27.163 interface tx-queue length

Description

Set the size of the queue of outgoing packets on the interface. By default 1000 is set.

Command with **no** prefix resets to default.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config-if)> **tx-queue length** *‹length›*

(config-if)> **no tx-queue length**

Arguments

Argument	Value	Description
length	*Integer*	Queue length can take values from 0 to 65536.

Example

```
(config-if)> tx-queue length 255
Network::Interface::Base: "L2TP0": TX queue length is 255.
```

```
(config-if)> no tx-queue length
Network::Interface::Base: "L2TP0": TX queue length reset to ►
default.
```

History

Version	Description
3.06	The **interface tx-queue length** command has been introduced.

3.27.164 interface tx-queue scheduler cake

Description

Set the *CAKE* package scheduler for the interface. By default, the value cake is used for DSL and USB-modem interfaces, fq_codel — for all others.

Command with **no** prefix resets the scheduler to default.

Prefix no

Yes

Change settings

Yes

Multiple input No

Synopsis

(config-if)> **tx-queue scheduler cake**

(config-if)> **no tx-queue scheduler cake**

Example

```
(config-if)> tx-queue scheduler cake
Network::Interface::Base: "L2TP0": set TX queue scheduler to ►
"cake".
```

```
(config-if)> no tx-queue scheduler cake
Network::Interface::Base: "L2TP0": set default TX queue scheduler.
```

History

Version	Description
3.06	The **interface tx-queue scheduler cake** command has been introduced.

3.27.165 interface tx-queue scheduler fq_codel

Description Set the *FQ_CODEL* package scheduler for the interface. By default, the value cake is used for DSL and USB-modem interfaces, `fq_codel` — for all others.

Command with **no** prefix resets the scheduler to default.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config-if)> **tx-queue scheduler fq_codel**

(config-if)> **no tx-queue scheduler fq_codel**

Example

```
(config-if)> tx-queue scheduler fq_codel
Network::Interface::Base: "L2TP0": set TX queue scheduler to ►
"fq_codel".
```

```
(config-if)> no tx-queue scheduler fq_codel
Network::Interface::Base: "L2TP0": set default TX queue scheduler.
```

History

Version	Description
3.06	The **interface tx-queue scheduler fq_codel** command has been introduced.

3.27.166 interface up

Description Enable the network interface and persist the state “up” to the settings.

Command with **no** prefix disables the the network interface and deletes “up” from settings. Also **interface down** command can be used.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config-if)> **up**

(config-if)> **no up**

Example

```
(config-if)> up
Interface enabled.
```

History

Version	Description
2.00	The **interface up** command has been introduced.

3.27.167 interface wireguard listen-port

Description Specify *UDP* port number to which incoming connections are accepted. By default, port number is not defined.

Command with **no** prefix resets the port.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type Wireguard

Synopsis

(config-if)> **wireguard listen-port** *‹port›*

(config-if)> **no wireguard listen-port**

Arguments

Argument	Value	Description
port	*Integer*	Port number. Can take values from 1 to 65535 inclusively.

Example

```
(config-if)> wireguard listen-port 11633
Wireguard::Interface: "Wireguard4": set listen port to "11633".
```

```
(config-if)> no wireguard listen-port
Wireguard::Interface: "Wireguard4": reset listen port.
```

History

Version	Description
3.03	The **interface wireguard listen-port** command has been introduced.

3.27.168 interface wireguard peer

Description

Add the remote peer public key to configure the secure connection using the *WireGuard* protocol.

Command with **no** prefix removes specified key.

Prefix no

Yes

Change settings

Yes

Multiple input

Yes

Interface type

Wireguard

Group entry

`(config-wg-peer)`

Synopsis

(config-if)> **wireguard peer** *‹key›*

(config-if)> **no wireguard peer** *‹key›*

Arguments

Argument	Value	Description
key	*String*	Value of the key. Latin letters, numbers and equal signs are acceptable. The key length is 44 characters (Base64-encoded 32-byte string representation).

Example

```
(config-if)> wireguard peer ►
gbp1gW3pBQKssrAdah1hiib13Jl123ZM8dBIjjPmm0g=
(config-wg-peer)>
```

```
(config-if)> no wireguard peer ►
gbp1gW3pBQKssrAdah1hiib13Jl123ZM8dBIjjPmm0g=
Wireguard::Interface: "Wireguard4": removed peer ►
"gbp1gW3pBQKssrAdah1hiib13Jl123ZM8dBIjjPmmg0=".
```

History

Version	Description
3.03	The **interface wireguard peer** command has been introduced.

3.27.168.1 interface wireguard peer allow-ips

Description

Add the subnet of IP-addresses to which the transmission of packets inside the tunnel is allowed.

Note: You can add `0.0.0.0/0` subnet to allow transmission to any addresses.

Command with **no** prefix removes the subnet. If you use no argument, the entire list of subnets will be removed.

Prefix no

Yes

Change settings

Yes

Multiple input

Yes

Interface type

Wireguard

Synopsis

(config-wg-peer)> **allow-ips** *‹address› ‹mask›*

(config-wg-peer)> **no allow-ips** [*‹address› ‹mask›*]

Arguments

Argument	Value	Description
address	*IP-address*	Together with mask *mask* sets the subnet of IP-addresses to be translated.
mask	*IP-mask*	Mask of subnet. There are two ways to enter the mask: the canonical form (for example, `255.255.255.0`) and the form of prefix bit length (for example, `/24`).

Example

```
(config-wg-peer)> allow-ips 0.0.0.0/0
Wireguard::Interface: "Wireguard4": add allowed IPs ►
"0.0.0.0/0.0.0.0" from peer ►
"gbp1gW3pBQKssrAdah1hiib13Jl123ZM8dBIjjPmm2g=".
```

```
(config-wg-peer)> allow-ips 192.168.11.0 255.255.255.0
Wireguard::Interface: "Wireguard4": add allowed IPs ►
"192.168.11.0/255.255.255.0" from peer ►
"gbp1gW3pBQKssrAdah1hiib13Jl123ZM8dBIjjPmm2g=".
```

```
(config-wg-peer)> no allow-ips
Wireguard::Interface: "Wireguard4": clear allowed IPs of peer ►
"gbp1gW3pBQKssrAdah1hiib13Jl123ZM8dBIjjPmm2g=".
```

History

Version	Description
3.03	The **interface wireguard peer allow-ips** command has been introduced.

3.27.168.2 interface wireguard peer endpoint

Description Set the remote peer address to which the *WireGuard* connection will be established.

Command with **no** prefix removes the endpoint.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type Wireguard

Synopsis

(config-wg-peer)> **endpoint** ‹*address*› [:‹*port*›]

(config-wg-peer)> **no endpoint**

Arguments

Argument	Value	Description
address	*IP-address*	IP-address or domain name of the server.
port	*Integer*	The *UDP* server port.

Example

```
(config-wg-peer)> endpoint 10.0.1.10:11635
Wireguard::Interface: "Wireguard4": set peer ►
"gbp1gW3pBQKssrAdah1hiib13Jl123ZM8dBIjjPmm2g=" endpoint to ►
"10.0.1.10:11635".
```

```
(config-wg-peer)> no endpoint
Wireguard::Interface: "Wireguard4": reset endpoint for peer ►
"gbp1gW3pBQKssrAdah1hiib13Jl123ZM8dBIjjPmm2g=".
```

History

Version	Description
3.03	The **interface wireguard peer endpoint** command has been introduced.

3.27.168.3 interface wireguard peer keepalive-interval

Description Set the interval of keepalive packet sending for *WireGuard* connection monitoring. By default, the interval is not set.

Command with **no** prefix removes the setting.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type Wireguard

Synopsis

(config-wg-peer)> **keepalive-interval** *‹interval›*

(config-wg-peer)> **no keepalive-interval**

Arguments

Argument	Value	Description
`interval`	*Integer*	The interval of keepalive packet sending in seconds. Can take values from 3 to 3600 inclusively.

Example

```
(config-wg-peer)> keepalive-interval 3
Wireguard::Interface: "Wireguard4": set peer ►
"gbp1gW3pBQKssrAdah1hiib13Jl123ZM8dBIjjPmm2g=" keepalive interval ►
to "3".
```

```
(config-wg-peer)> no keepalive-interval
Wireguard::Interface: "Wireguard4": reset persistent keepalive ►
interval for peer "gbp1gW3pBQKssrAdah1hiib13Jl123ZM8dBIjjPmm2g=".
```

History

Version	Description
3.03	The **interface wireguard peer keepalive-interval** command has been introduced.

3.27.168.4 interface wireguard peer preshared-key

Description

Set preshared key for *WireGuard* connection to remote peer. The preshared key (PSK) is an optional security improvement as per the*WireGuard* protocol and should be a unique PSK per client for highest security. By default, PSK is not used.

Command with **no** prefix removes the setting.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

Wireguard

Synopsis

(config-wg-peer)> **preshared-key** *‹preshared-key›*

(config-wg-peer)> **no preshared-key**

Arguments

Argument	Value	Description
`preshared-key`	*String*	Secret PSK key value. Latin letters, numbers and equal signs are acceptable. The key length is 44 characters.

Example

```
(config-wg-peer)> preshared-key ►
WY2fkhJZuDCbYew7L8whBMzkReVf8KKzWJrmaR79F8z=
Wireguard::Interface: "Wireguard4": set preshared key for peer ►
"gbp1gW3pBQKssrAdah1hiib13Jl123ZM8dBIjjPmm2g=".
```

```
(config-wg-peer)> no preshared-key
Wireguard::Interface: "Wireguard4": reset preshared key for peer ►
"gbp1gW3pBQKssrAdah1hiib13Jl123ZM8dBIjjPmm2g=".
```

History

Version	Description
3.03	The **interface wireguard peer preshared-key** command has been introduced.

3.27.169 interface wireguard private-key

Description

Set or generate the private key to connect to the remote peers via *WireGuard* protocol. By default, private key is not configured.

Prefix no

No

Change settings

No

Multiple input

No

Interface type

Wireguard

Synopsis

(config-if)> **wireguard private-key** [‹*private-key*›]

Arguments

Argument	Value	Description
`private-key`	*String*	A new private key value. Latin letters, numbers and equal signs are acceptable. The key length is 44 characters.

Example

```
(config-if)> wireguard private-key
Wireguard::Interface: "Wireguard4": generated new private key.
```

```
(config-if)> wireguard private-key ►
UshaeghezaiJ7reo8iK6ear0eomujohkeen8jahX5uo=
Wireguard::Interface: "Wireguard4": set private key.
```

History

Version	Description
3.03	The **interface wireguard private-key** command has been introduced.

3.27.170 interface wmm

Description

Enable *WMM* on the interface.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type Access Point

Synopsis

(config-if)> **wmm**

(config-if)> **no wmm**

Example

```
(config-if)> wmm
WMM extensions enabled.
```

History

Version	Description
2.00	The **interface wmm** command has been introduced.

3.27.171 interface wpa-eap radius secret

Description Specify the shared secret for secure communication between a *RADIUS* server and a *RADIUS* client.

Command with **no** prefix deletes the shared secret.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type Bridge

Synopsis

(config-if)> **wpa-eap radius secret** *‹secret›*

(config-if)> **no wpa-eap radius secret**

Arguments

Argument	Value	Description
secret	*String*	The value of *RADIUS* shared secret. Maximum key length is 64 characters.

Example

```
(config-if)> wpa-eap radius secret ►
(+>R#G`}-JNxru'i8i|lK}wBN9E^XOXa{xFOG-N^%FaTnr|S(e(q$/lP2/tbX/#Q
Network::Interface::Rtx::WpaEap: Bridge0 RADIUS secret applied.
```

```
(config-if)> no wpa-eap radius secret
Network::Interface::Rtx::WpaEap: Bridge0 RADIUS secret cleared.
```

History

Version	Description
3.01	The **interface wpa-eap radius secret** command has been introduced.

3.27.172 interface wpa-eap radius server

Description

Specify *RADIUS* server address.

Command with **no** prefix deletes the address.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

Bridge

Synopsis

(config-if)> **wpa-eap radius server** ‹*address*› [: ‹*port*›]

(config-if)> **no wpa-eap radius server**

Arguments

Argument	Value	Description
address	*IP-address*	*RADIUS* server IP-address.
port	*Integer*	*RADIUS* server port.

Example

```
(config-if)> wpa-eap radius server 192.168.10.10
Network::Interface::Rtx::WpaEap: Bridge0 RADIUS server set to ►
192.168.10.10.
```

```
(config-if)> wpa-eap radius server 192.168.10.10:1111
Network::Interface::Rtx::WpaEap: Bridge0 RADIUS server set to ►
192.168.10.10:1111.
```

```
(config-if)> no wpa-eap radius server
Network::Interface::Rtx::WpaEap: Bridge0 RADIUS server cleared.
```

History

Version	Description
3.01	The **interface wpa-eap radius server** command has been introduced.

3.27.173 interface wps

Description

Enable *WPS* functionality.

Prefix no

Yes

Change settings Yes

Multiple input No

Interface type WiFi

Synopsis

(config-if)> **wps**

(config-if)> **no wps**

Example

```
(config-if)> wps
WPS functionality enabled.
```

History

Version	Description
2.00	The **interface wps** command has been introduced.

3.27.174 interface wps auto-self-pin

Description Enable *WPS* auto-self-pin mode. By default auto-self-pin mode is enabled.

Command with **no** prefix disables this mode.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type WiFi

Synopsis

(config-if)> **wps auto-self-pin**

(config-if)> **no wps auto-self-pin**

Example

```
(config-if)> wps auto-self-pin
Network::Interface::Rtx::Wps: an auto self PIN mode enabled.
```

History

Version	Description
2.04	The **interface wps auto-self-pin** command has been introduced.

3.27.175 interface wps button

Description Start WPS process using a software button. Process takes 2 minutes or until the first connection occured.

Prefix no No

Change settings No

Multiple input No

Interface type WiFi

Synopsis

(config-if)> **wps button** *‹direction›*

Arguments

Argument	Value	Description
direction	send	Send WiFi configuration.
	receive	Receive WiFi configuration from City.

Example

```
(config-if)> wps button send
Sending WiFi configuration process started (software button mode).
```

History

Version	Description
2.00	The **interface wps button** command has been introduced.

3.27.176 interface wps peer

Description Start WPS process using remote peer's PIN. Process takes 2 minutes or until the first connection occured. By default, WPS PIN is disabled.

Prefix no No

Change settings No

Multiple input No

Interface type WiFi

Synopsis

(config-if)> **wps peer** *‹direction› ‹pin›*

Arguments

Argument	Value	Description
direction	send	Send WiFi configuration.
	receive	Receive WiFi configuration from the remote peer.
pin	*String*	PIN code of the remote peer.

Example

```
(config-if)> wps peer send 53794141
Network::Interface::Rtx::Wps: "WifiMaster0/AccessPoint0": peer ►
PIN WPS session started.
```

History

Version	Description
2.04	The **interface wps peer** command has been introduced.

3.27.177 interface wps self-pin

Description Start WPS process using self PIN. Process takes 2 minutes or until the first connection occur.

Prefix no No

Change settings No

Multiple input No

Interface type WiFi

Synopsis

(config-if)> **wps self-pin** *‹direction›*

Arguments

Argument	Value	Description
direction	send	Send WiFi configuration.
	receive	Receive WiFi configuration from City.

Example

```
(config-if)> wps self-pin receive
Receiving WiFi configuration process started (self PIN mode).
```

History

Version	Description
2.00	The **interface wps self-pin** command has been introduced.

3.28 ip arp

Description Set static mapping between an IP-address and a MAC-address for hosts that do not support dynamic *ARP*.

Command with **no** prefix removes entry from ARP table. If you use no arguments, the whole list of ARP entrys will be removed.

Prefix no Yes

Change settings Yes

Multiple input Yes

Synopsis

(config)> **ip arp** *‹ip› ‹mac›*

(config)> **no ip arp** [*‹ip›*]

Arguments

Argument	Value	Description
ip	*IP-address*	IP-address in four-part dotted decimal format corresponding to the local data-link address.
mac	*MAC-address*	MAC-address as six groups of two hexadecimal digits separated by colons.

Example

```
(config)> ip arp 192.168.2.50 a1:2e:84:85:f4:21
Network::ArpTable: Static ARP entry saved.
```

```
(config)> no ip arp 192.168.2.50
Network::ArpTable: Static ARP entry deleted for 192.168.2.50.
```

```
(config)> no ip arp
Network::ArpTable: Static ARP table cleared.
```

History

Version	Description
2.00	The **ip arp** command has been introduced.

3.29 ip dhcp class

Description

Access to a group of commands to configure *DHCP* vendor class (option 60). If specified class name is not found, the command tries to create it.

Command with **no** prefix removes selected class.

Prefix no

Yes

Change settings

No

Multiple input

Yes

Group entry

`(config-dhcp-class)`

Synopsis

(config)> **ip dhcp class** *‹class›*

(config)> **no ip dhcp class** *‹class›*

Arguments

Argument	Value	Description
class	*String*	The vendor-class name.

Example

```
(config)> ip dhcp class STB-One
Dhcp::Server: Vendor class "STB-One" has been created.
```

History

Version	Description
2.00	The **ip dhcp class** command has been introduced.

3.29.1 ip dhcp class option

Description

Set an option 60 to match the vendor-class.

Command with **no** prefix removes selected option.

Prefix no

Yes

Change settings

Yes

Multiple input

Yes

Synopsis

(config-dhcp-class)> **option** ‹*number*› **hex** ‹*data*›

(config-dhcp-class)> **no option** ‹*number*›

Arguments

Argument	Value	Description
`number`	*Integer*	Option number. Now the only `60` value is used.
`data`	*String*	Value of an option.

Example

```
(config-dhcp-class)> option 60 hex FF
Dhcp::Server: Option 60 is set to FF.
```

History

Version	Description
2.00	The **ip dhcp class option** command has been introduced.

3.30 ip dhcp host

Description

Configure static linking of IP-address to MAC-address of the host. If the host with the specified name is not found, the command tries to create it. If the specified IP-address is not in range of any pool, the command will remain in the settings, but will not affect the *DHCP-server* functioning.

The command allows one to change the MAC-address, leaving the old value IP-address and vice versa — to change the IP-address, leaving the old MAC-address value intact.

Command with **no** prefix removes the host.

Prefix no

Yes

Change settings

Yes

Multiple input Yes

Synopsis

(config)> **ip dhcp host** ‹*host*› [*mac*] [*ip*]

(config)> **no ip dhcp host** ‹*host*›

Arguments

Argument	Value	Description
host	*String*	Arbitrary host name, used to identify a MAC-IP pair in the settings.
mac	*MAC-address*	MAC-address of the host for static linking of IP-address. If not specified, the value is taken from the previous configuration.
ip	*IP-address*	IP-address of the host. If not specified, the value is taken from the previous configuration.

Example

```
(config)> ip dhcp host HOST 192.168.1.44
new host "HOST" has been created.
```

History

Version	Description
2.00	The **ip dhcp host** command has been introduced.

3.31 ip dhcp pool

Description Access to a group of commands to configure DHCP-pool. If the pool is not found, the command tries to create it. For a pool one sets a list of DNS-servers (**dns-server** command), default gateway (**default-router** command) and the lease time (**lease** command), as well as a range of dynamic IP-addresses (**range** command).

Having configured the pool, it is necessary to enable the *DHCP* service using the **service dhcp** command.

You can enter up to 32 pools. Maximum pool name length is 32 characters.

Note: In the current version of the system no more than one pool per interface is supported. For *DHCP-server* to function correctly it is required that the range of IP-addresses set by **range** command belong to the network that is configured on one of the device's Ethernet-interfaces.

Command with **no** prefix removes the pool.

Prefix no Yes

Change settings Yes

Multiple input Yes

Group entry `(config-dhcp-pool)`

Synopsis

(config)> **ip dhcp pool** *‹name›*

(config)> **no ip dhcp pool** *‹name›*

Arguments

Argument	Value	Description
name	*String*	DHCP pool name.

Example

```
(config)> ip dhcp pool test_pool
pool "test_pool" has been created.
```

History

Version	Description
2.00	The **ip dhcp pool** command has been introduced.

3.31.1 ip dhcp pool bind

Description Bind the pool to specified interface.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type Ethernet

Synopsis

(config-dhcp-pool)> **bind** *‹interface›*

(config-dhcp-pool)> **no bind** *‹interface›*

Arguments

Argument	Value	Description
interface	*Interface name*	Full interface name or an alias.

Example

```
(config-dhcp-pool)> bind FastEthernet0/Vlan2
pool "test_pool" bound to interface FastEthernet0/Vlan2.
```

History

Version	Description
2.00	The **ip dhcp pool bind** command has been introduced.

3.31.2 ip dhcp pool bootfile

Description Set boot file path on TFTP server for DHCP client (option 67).

Command with **no** prefix removes the setting.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type Ethernet

Synopsis

(config-dhcp-pool)> **bootfile** *‹bootfile›*

(config-dhcp-pool)> **no bootfile**

Arguments

Argument	Value	Description
bootfile	*Filename*	The boot file path.

Example

```
(config-dhcp-pool)> bootfile test.cnf
Dhcp::Pool: "_WEBADMIN": set bootfile option to "test.cnf".
```

```
(config-dhcp-pool)> no bootfile
Dhcp::Pool: "_WEBADMIN": cleared bootfile option.
```

History

Version	Description
2.11	The **ip dhcp pool bootfile** command has been introduced.

3.31.3 ip dhcp pool class

Description Access to a group of commands to configure *DHCP* vendor class for selected pool. If specified class name is not found, the command tries to create it.

To work correctly class name should be the same as for **ip dhcp class** command.

Command with **no** prefix removes selected class.

Prefix no Yes

Change settings Yes

Multiple input Yes

Group entry `(config-dhcp-pool-class)`

Synopsis

(config-dhcp-pool)> **class** *‹class›*

(config-dhcp-pool)> **no class** *‹class›*

Arguments

Argument	Value	Description
`class`	*String*	The vendor-class name.

Example

```
(config-dhcp-pool)> class STB-One
Dhcp::Server: Vendor class "STB-One" has been created.
```

History

Version	Description
2.00	The **ip dhcp pool class** command has been introduced.

3.31.3.1 ip dhcp pool class option

Description

Set additional options for *DHCP* client in case of vendor-class matching.

Command with **no** prefix removes selected option.

Prefix no

Yes

Change settings

Yes

Multiple input

Yes

Synopsis

(config-dhcp-pool-class)> **option** *‹number› ‹type› ‹data›*

(config-dhcp-pool-class)> **no option** *‹number›*

Arguments

Argument	Value	Description
`number`	6	6 option, DNS server.
	42	42 option, NTP server.
	43	43 option, vendor specific information.
`type`	`ip`	Type of `data` is IP-address. This type is not used for 43 option.
	hex	Type of `data` is hexadecimal number.
`data`	*String*	Value of an option.

Example

```
(config-dhcp-pool-class)> option 6 ip 192.168.1.1
Dhcp::Server: Option 6 is set to 192.168.1.1.
```

History

Version	Description
2.00	The **ip dhcp pool class option** command has been introduced.

3.31.4 ip dhcp pool debug

Description Add debug messages to the system log. By default, the setting is disabled.

Command with **no** prefix disables debugging.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config-dhcp-pool)> **debug**

(config-dhcp-pool)> **no debug**

History

Version	Description
2.01	The **ip dhcp pool debug** command has been introduced.

3.31.5 ip dhcp pool default-router

Description Configure default gateway IP-address. If not specified, the address of the Ethernet-interface determined automatically for a given range **range** will be used.

Command with **no** prefix cancels the setting.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config-dhcp-pool)> **default-router** *‹address›*

(config-dhcp-pool)> **no default-router**

Arguments

Argument	Value	Description
address	*IP-address*	Default gateway address.

Example

```
(config-dhcp-pool)> default-router 192.168.1.88
pool "test_pool" router address has been saved.
```

History

Version	Description
2.00	The **ip dhcp pool default-router** command has been introduced.

3.31.6 ip dhcp pool dns-server

Description

Configure IP-addresses of the DNS servers (DHCP option 6). If not specified, the address of the Ethernet-interface determined automatically for a given range **range** will be used.

Command with **no** prefix cancels the setting.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config-dhcp-pool)> **dns-server** (*‹address1›* [*address2*] | **disable**)

(config-dhcp-pool)> **no dns-server**

Arguments

Argument	Value	Description
address1	*IP-address*	Address of primary DNS-server.
address2	*IP-address*	Address of secondary DNS-server.
disable	*Keyword*	Disable DHCP option 6.

Example

```
(config-dhcp-pool)> dns-server 192.168.1.88
pool "test_pool" name server list has been saved.
```

History

Version	Description
2.00	The **ip dhcp pool dns-server** command has been introduced.
2.11	`Disable` argument has been added.

3.31.7 ip dhcp pool domain

Description

Specify the domain name that client should use when resolving hostnames via DNS (option 15).

Command with **no** prefix cancels the setting.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config-dhcp-pool)> **domain** *‹domain›*

(config-dhcp-pool)> **no domain**

Arguments

Argument	Value	Description
domain	*String*	Local domain name.

Example

```
(config-dhcp-pool)> domain example.net
Dhcp::Pool: Domain option has been saved.
```

History

Version	Description
2.05	The **ip dhcp pool domain** command has been introduced.

3.31.8 ip dhcp pool enable

Description

Start to use the pool in the system.

Command with **no** prefix disables pool using.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config-dhcp-pool)> **enable**

(config-dhcp-pool)> **no enable**

Example

```
(config-dhcp-pool)> enable
Dhcp::Server: pool "111" is enabled.
```

History

Version	Description
2.03	The **ip dhcp pool enable** command has been introduced.

3.31.9 ip dhcp pool lease

Description

Set the lease time of DHCP pool IP-address. By default, 25200 value is used (7 hours).

Command with **no** prefix resets lease time to default.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config-dhcp-pool)> **lease** ‹*lease*›

(config-dhcp-pool)> **no lease**

Arguments

Argument	Value	Description
lease	*Integer*	Lease time in seconds. Can take values from 1 to 259200 seconds (3 days).

Example

```
(config-dhcp-pool)> lease 259200
Dhcp::Pool: "_WEBADMIN": set lease time: 259200 seconds.
```

```
(config-dhcp-pool)> no lease
Dhcp::Pool: "_WEBADMIN": lease time reset to default (25200 ►
seconds).
```

History

Version	Description
2.00	The **ip dhcp pool lease** command has been introduced.

3.31.10 ip dhcp pool next-server

Description

Set TFTP server address for DHCP client (option 66).

Command with **no** prefix removes the setting.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

Ethernet

Synopsis

(config-dhcp-pool)> **next-server** ‹*address*›

(config-dhcp-pool)> **no next-server**

Arguments

Argument	Value	Description
address	*IP-address*	TFTP server address.

Example

```
(config-dhcp-pool)> next-server 10.1.1.11
Dhcp::Pool: "_WEBADMIN": set next server address: 10.1.1.11.
```

```
(config-dhcp-pool)> no next-server
Dhcp::Pool: "_WEBADMIN": cleared next server address.
```

History

Version	Description
2.11	The **ip dhcp pool next-server** command has been introduced.

3.31.11 ip dhcp pool option

Description Set additional options for DHCP client.

Command with **no** prefix removes the setting.

Prefix no Yes

Change settings Yes

Multiple input Yes

Interface type Ethernet

Synopsis

(config-dhcp-pool)> **option** *‹number› ‹type› ‹data›*

(config-dhcp-pool)> **no option** *‹number›*

Arguments

Argument	Value	Description
number	4	4 option, Time server.
	6	6 option, DNS server.
	42	42 option, NTP server.
	44	44 option, NetBIOS server.
	26	26 option, MTU.
	121	121 option, Classless Static Routes.
	249	249 option, MS Routes.
type	ip	Type of data is IP-address. It is not applicable to 26 option.
	hex	Type of data is hexadecimal number.
	ascii	Type of data is ASCII number.
	mtu	Type of data is Maximum Transmit Unit size.
data	*String*	Value of an option.

Example

```
(config-dhcp-pool)> option 4 hex 00010203
(config-dhcp-pool)> option 4 ascii test
(config-dhcp-pool)> option 6 8.8.8.8,8.8.4.4,192.168.1.1
(config-dhcp-pool)> no option 6 8.8.8.8,8.8.4.4,192.168.1.1
```

History

Version	Description
2.09	The **ip dhcp pool option** command has been introduced.

3.31.12 ip dhcp pool range

Description

Configure the range of dynamic addresses issued to DHCP-clients of a subnet. The range is set by start and end IP-addresses or the start address and size. The network interface to which the settings are applied is chosen automatically. Address of the chosen interface is used as the default gateway and DNS-server, if other addresses are not specified using commands **ip dhcp pool default-router** and **ip dhcp pool dns-server**.

Command with **no** prefix removes the range.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config-dhcp-pool)> **range** ‹*begin*› (‹*end*› | ‹*size*›)

(config-dhcp-pool)> **no range**

Arguments

Argument	Value	Description
begin	*IP-address*	Pool's start address.
end	*IP-address*	Pool's end address.
size	*Integer*	Pool size.

Example

```
(config-dhcp-pool)> range 192.168.15.43 3
pool "_WEBADMIN" range has been saved.
```

History

Version	Description
2.00	The **ip dhcp pool range** command has been introduced.

3.31.13 ip dhcp pool update-dns

Description

Add static records into DNS-proxy when DHCP-address is assigned. The name of record is the hostname of the DHCP-request. By default, the feature is disabled.

Command with **no** prefix disables the feature.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config-dhcp-pool)> **update-dns**

(config-dhcp-pool)> **no update-dns**

Example

```
(config-dhcp-pool)> update-dns
Dhcp::Pool: DNS update has been enabled.
```

History

Version	Description
2.06	The **ip dhcp pool update-dns** command has been introduced.

3.31.14 ip dhcp pool wpad

Description

Configure DHCP option 252 — *WPAD* protocol. By default, the option is disabled.

Command with **no** prefix disables the setting.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config-dhcp-pool)> **wpad** *‹wpad›*

(config-dhcp-pool)> **no wpad**

Arguments

Argument	Value	Description
wpad	*String*	URL of proxy.

Example

```
(config-dhcp-pool)> wpad http://wpad/wpad.dat
Dhcp::Pool: WPAD option has been saved.
```

History

Version	Description
2.05	The **ip dhcp pool wpad** command has been introduced.

3.32 ip dhcp relay lan

Description

Specify which network interface the DHCP relay will use to handle client's requests. Several “lan” interfaces can be specified, to which end the command should be entered several times, enumerating all desired interfaces one by one.

Command with **no** prefix disables the DHCP relay on the specified interface. If you use no argument, the DHCP relay will be removed from all interfaces.

Prefix no Yes

Change settings Yes

Multiple input Yes

Synopsis

(config)> **ip dhcp relay lan** ‹*interface*›

(config)> **no ip dhcp relay lan** [*interface*]

Arguments

Argument	Value	Description
interface	*Interface name*	Full name or an alias of Ethernet interface, through which DHCP relay will accept requests from clients.

Example

```
(config)> ip dhcp relay lan Home
added LAN interface Home.
```

History

Version	Description
2.00	The **ip dhcp relay lan** command has been introduced.

3.33 ip dhcp relay server

Description Specify the IP-address of the *DHCP-server*, to which the relay will forward client requests from the LAN.

Command with **no** prefix removes the setting.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config)> **ip dhcp relay server** ‹*address*›

(config)> **no ip dhcp relay server** [*address*]

Arguments

Argument	Value	Description
address	*IP-address*	IP-address of the *DHCP-server*.

Example

```
(config)> ip dhcp relay server 192.168.1.11
using DHCP server 192.168.1.11.
```

History

Version	Description
2.00	The **ip dhcp relay server** command has been introduced.

3.34 ip dhcp relay wan

Description

Specify the network interface through which DHCP relay will interact with higher level *DHCP-server*. There can be only one interface of such type in the system. If exact address of the server is not specified (see **ip dhcp relay server**), the requests will be broadcasted. It is recommended to specify server address.

Command with **no** prefix removes the setting.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config)> **ip dhcp relay wan** ‹*interface*›

(config)> **no ip dhcp relay wan** [*interface*]

Arguments

Argument	Value	Description
`interface`	*Interface name*	Full name or an alias of Ethernet interface, on which requests from the DHCP-clients will be sent.

Example

```
(config)> ip dhcp relay wan FastEthernet0/Vlan2
using WAN interface FastEthernet0/Vlan2.
```

History

Version	Description
2.00	The **ip dhcp relay wan** command has been introduced.

3.35 ip esp alg enable

Description

Enable *IPSec Passthrough* mode for *IPsec ESP* tunnel. By default, the setting is disabled.

Command with **no** prefix disables the feature.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config)> **ip esp alg enable**

(config)> **no ip esp alg enable**

Example

```
(config)> ip esp alg enable
Esp::Alg: Enabled.
```

```
(config)> no ip esp alg enable
Esp::Alg: Disabled.
```

History

Version	Description
3.05	The **ip esp alg enable** command has been introduced.

3.36 ip flow-cache timeout active

Description

Set timeout of active sessions in cache. By default, the value `10` is used.

Command with **no** prefix resets the setting to default.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config)> **ip flow-cache timeout active** *‹timeout›*

(config)> **no ip flow-cache timeout active**

Arguments

Argument	Value	Description
`timeout`	*Integer*	The timeout value, in minutes. Can take value in the range from 1 to 30.

Example

```
(config)> ip flow-cache timeout active 1
Netflow::Manager: Active timeout set to "1" min.
```

```
(config)> no ip flow-cache timeout active
Netflow::Manager: Active timeout reset to "10" min.
```

History

Version	Description
2.11	The **ip flow-cache timeout active** command has been introduced.

3.37 ip flow-cache timeout inactive

Description Set timeout of inactive sessions in cache. By default, the value `20` is used.

Command with **no** prefix resets the setting to default.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config)> **ip flow-cache timeout inactive** ‹*timeout*›

(config)> **no ip flow-cache timeout inactive**

Arguments

Argument	Value	Description
`timeout`	*Integer*	The timeout value, in seconds. Can take value in the range from 1 to 600.

Example

```
(config)> ip flow-cache timeout inactive 1
Netflow::Manager: Inactive timeout set to "1" s.
```

```
(config)> no ip flow-cache timeout inactive
Netflow::Manager: Inactive timeout reset to "20" s.
```

History

Version	Description
2.11	The **ip flow-cache timeout inactive** command has been introduced.

3.38 ip flow-export destination

Description Set parameters of *NetFlow* collector.

Command with **no** prefix removes collector's parameters.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config)> **ip flow-export destination** ‹*address*› ‹*port*›

(config)> **no ip flow-export destination**

Arguments

Argument	Value	Description
address	*IP-address*	IP-address of the data collector.
port	*Integer*	Collector's UDP port number. Can take values 2055, 2056, 4432, 4739, 9025, 9026, 9995, 9996, 6343.

Example

```
(config)> ip flow-export destination 192.168.101.31 4739
Netflow::Manager: Export destination is set to ►
192.168.101.31:4739.
```

```
(config)> no ip flow-export destination
Netflow::Manager: Export destination is unset.
```

History

Version	Description
2.11	The **ip flow-export destination** command has been introduced.

3.39 ip flow-export version

Description

Set version of *NetFlow* collector. By default, 5 value is used.

Command with **no** prefix resets version to default.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config)> **ip flow-export version** ‹*version*›

(config)> **no ip flow-export version**

Arguments

Argument	Value	Description
version	*String*	Version of protocol.

Example

```
(config)> ip flow-export version 9
Netflow::Manager: Set export protocol version to 9.
```

```
(config)> no ip flow-export version
Netflow::Manager: Reset export version to 5.
```

History

Version	Description
3.05	The **ip flow-export version** command has been introduced.

3.40 ip host

Description Add a domain name and address as a DNS-record.

Prefix no Yes

Change settings Yes

Multiple input Yes

Synopsis

(config)> **ip host** *‹domain› ‹address›*

(config)> **no ip host** [*‹domain› ‹address›*]

Arguments

Argument	Value	Description
domain	*String*	A domain name of a host.
address	*IP-address*	An IP-address of a host.

Example

```
(config)> ip host keenetic.local 192.168.1.22
Dns::Manager: Added static record for "keenetic.local", address ►
192.168.1.22.
```

```
(config)> no ip host keenetic.local 192.168.1.22
Dns::Manager: Record "keenetic.local", address 192.168.1.22 ►
deleted.
```

History

Version	Description
2.00	The **ip host** command has been introduced.

3.41 ip hotspot

Description Enter the Hotspot configuration command group.

Prefix no No

Change settings No

Multiple input No

Interface type IP

Group entry (config-hotspot)

Synopsis

(config)> **ip hotspot**

Example

```
(config)> ip hotspot
(config-hotspot)>
```

History

Version	Description
2.06	The **ip hotspot** command has been introduced.

3.41.1 ip hotspot auto-scan interface

Description Enable subnetwork passive scanning on interface. By default is enabled.

Command with **no** prefix disables the setting.

Prefix no Yes

Change settings Yes

Multiple input Yes

Interface type IP

Synopsis

(config-hotspot)> **auto-scan interface** ‹*interface*›

(config-hotspot)> **no auto-scan interface** ‹*interface*›

Arguments

Argument	Value	Description
`interface`	*`Interface name`*	Full interface name or an alias.

Example

```
(config-hotspot)> auto-scan interface WifiMaster0/AccessPoint1
Hotspot::Discovery::Manager: Subnetwork scanning on interface ►
"WifiMaster0/AccessPoint1" is unchanged.
```

```
(config-hotspot)> auto-scan interface WifiMaster0/AccessPoint1
Hotspot::Discovery::Manager: Subnetwork scanning on interface ►
"WifiMaster0/AccessPoint1" is disabled.
```

History

Version	Description
2.08	The **ip hotspot auto-scan interface** command has been introduced.

3.41.2 ip hotspot auto-scan interval

Description Set interval for probes of online hosts.

Command with **no** prefix resets setting to default.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type IP

Synopsis

(config-hotspot)> **auto-scan interval** *‹interval›*

(config-hotspot)> **no auto-scan interval**

Arguments

Argument	Value	Description
interval	*Integer*	Auto-scan probe interval in seconds. By default, the value 30 is used.

Example

```
(config-hotspot)> auto-scan interval 10
Hotspot::Discovery::Manager: Auto-scan probe interval is set to ►
10 s.
```

```
(config-hotspot)> no auto-scan interval
Hotspot::Discovery::Manager: Auto-scan probe interval reset to ►
default.
```

History

Version	Description
2.08	The **ip hotspot auto-scan interval** command has been introduced.

3.41.3 ip hotspot auto-scan passive

Description Set passive autoscan rate in hosts per seconds.

Command with **no** prefix resets setting to default.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type IP

Synopsis

(config-hotspot)> **auto-scan passive** *‹rate›* **hps**

(config-hotspot)> **no auto-scan passive**

Arguments

Argument	Value	Description
rate	*Integer*	Passive autoscan rate. By default, the value 3 is used.

Example

```
(config-hotspot)> auto-scan passive 5 hps
Hotspot::Discovery::Manager: Auto-scan rate is set to 5 hps.
```

```
(config-hotspot)> no auto-scan passive
Hotspot::Discovery::Manager: Auto-scan rate reset to default.
```

History

Version	Description
2.08	The **ip hotspot auto-scan passive** command has been introduced.

3.41.4 ip hotspot auto-scan timeout

Description

Set offline timeout for hosts. After the specified time, the missing host is removed from the online host list.

Command with **no** prefix resets setting to default.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

IP

Synopsis

(config-hotspot)> **auto-scan timeout** *‹timeout›*

(config-hotspot)> **no auto-scan timeout**

Arguments

Argument	Value	Description
timeout	*Integer*	Offline timeout in seconds. By default, the value 35 is used.

Example

```
(config-hotspot)> auto-scan timeout 31
Hotspot::Discovery::Manager: Auto-scan host offline timeout is ►
set to 31 s.
```

```
(config-hotspot)> no auto-scan timeout
Hotspot::Discovery::Manager: Auto-scan host offline timeout reset ►
to default.
```

History

Version	Description
2.08	The **ip hotspot auto-scan timeout** command has been introduced.

3.41.5 ip hotspot default-policy

Description

Define the Hotspot policy for all interfaces or assign IP Policy. Policy applies to all hosts that have no explicitly configured access rule, **ip hotspot policy**.

Default policy: `permit`.

Command with **no** prefix resets policy to default.

Prefix no Yes

Change settings Yes

Multiple input Yes

Interface type IP

Synopsis

(config-hotspot)> **default-policy** (‹*access*› | ‹*policy*›)

(config-hotspot)> **no default-policy**

Arguments

Argument	Value	Description
`access`	`permit`	Permit access to the internet.
	`deny`	Deny access to the internet.
`policy`	*`Policy name`*	Name of IP Policy profile.

Example

```
(config-hotspot)> default-policy permit
FHotspot::Manager: Default policy "permit" applied.
```

```
(config-hotspot)> default-policy deny
Hotspot::Manager: Default policy "deny" applied.
```

```
(config-hotspot)> default-policy Policy0
Hotspot::Manager: Default policy "Policy0" applied.
```

```
(config-hotspot)> no default-policy
Hotspot::Manager: Default policy cleared.
```

History

Version	Description
2.09	The **ip hotspot default-policy** command has been introduced.
2.12	Argument `policy` was added.

3.41.6 ip hotspot host

Description Setup bypass or block rules for specific Hotspot clients. Host rules override interface based policy (see **ip hotspot policy** command).

Command with **no** prefix removes the setting.

Prefix no Yes

Change settings Yes

Multiple input Yes

Interface type IP

Synopsis

(config-hotspot)> **host** ‹*mac*› (‹*access*› | **schedule** ‹*schedule*› | **policy** ‹*policy*›)

(config-hotspot)> **no host** ‹*mac*› (‹*access*› | **schedule** | **policy**)

Arguments

Argument	Value	Description
mac	*MAC-address*	Host MAC address. Host must be registered via **known host** in advance.
access	permit	Permit access to the internet.
	deny	Deny access to the internet.
schedule	*Schedule name*	The name of the schedule that was created with **schedule** group of commands.
policy	*Policy name*	Name of IP Policy profile.

Example

```
(config)> known host MYTEST 54:e4:3a:8a:f3:a7
Hotspot::Manager: Policy "permit" applied to interface "Home".

(config-hotspot)> host 54:e4:3a:8a:f3:a7 permit
Hotspot::Manager: Rule "permit" applied to host ►
"54:e4:3a:8a:f3:a7".

(config-hotspot)> host 54:e4:3a:8a:f3:a7 deny
Hotspot::Manager: Rule "deny" applied to host "54:e4:3a:8a:f3:a7".

(config-hotspot)> host 54:e4:3a:8a:f3:a7 schedule MYSCHEDULE
Hotspot::Manager: Schedule "MYSCHEDULE" applied to host ►
"54:e4:3a:8a:f3:a7".

(config-hotspot)> no host 54:e4:3a:8a:f3:a7 schedule
Hotspot::Manager: Host "54:e4:3a:8a:f3:a7" schedule disabled.

(config-hotspot)> host 54:e4:3a:8a:f3:a7 policy Policy0
Hotspot::Manager: Policy "Policy0" applied to host ►
"54:e4:3a:8a:f3:a7".

(config-hotspot)> no host 54:e4:3a:8a:f3:a7 policy
Hotspot::Manager: Policy removed from host "54:e4:3a:8a:f3:a7".
```

History

Version	Description
2.06	The **ip hotspot host** command has been introduced.
2.12	Arguments permit, deny, schedule, policy were added.

3.41.7 ip hotspot host service-class

Description

Assign a specific class to all traffic bound to a registered host. The class is represented by an integer from 1 to 6. Registration of a host is performed in advance by the **known host** command.

Command with **no** prefix removes the class.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

IP

Synopsis

(config-hotspot)> **host** *‹mac›* **service-class** *‹service-class›*

(config-hotspot)> **no host** *‹mac›* **service-class**

Arguments

Argument	Value	Description
mac	*MAC-address*	Host MAC-address.
service-class	1	Minimum latency (VoIP).
	2	Real-time interactive (Games, Video conferencing).
	3	Broadcast services (YouTube, NetFlix).
	4	Low latency data (Database, SSH).
	5	High-throughput data (Web traffic).
	6	Low-priority data (File sharing, BitTorrent).

Example

```
(config-hotspot)> host 04:d4:c4:54:bc:11 service-class 3
Hotspot::Manager: Service class "3" applied to host ►
"04:d4:c4:54:bc:11".
```

```
(config-hotspot)> no host 04:d4:c4:54:bc:59 service-class
Hotspot::Manager: Service class removed from host ►
"04:d4:c4:54:bc:59".
```

History

Version	Description
3.05	The **ip hotspot host service-class** command has been introduced.

3.41.8 ip hotspot policy

Description

Define the Hotspot policy for a specific interface. Policy applies to all hosts that have no explicitly configured access rule, **ip hotspot host**.

Default policy: `permit`.

Command with **no** prefix resets policy to default.

Prefix no Yes

Change settings Yes

Multiple input Yes

Interface type IP

Synopsis

(config-hotspot)> **policy** *‹interface›* (*‹access›* | *‹policy›*)

(config-hotspot)> **no policy** *‹interface›*

Arguments

Argument	Value	Description
`interface`	*`Interface name`*	Ethernet interface full name or an alias.
`access`	`permit`	Permit access to the internet.
	`deny`	Deny access to the internet.
`policy`	*`Policy name`*	Name of IP Policy profile.

Example

```
(config-hotspot)> policy Home permit
Hotspot::Manager: Policy "permit" applied to interface "Home".
```

```
(config-hotspot)> policy Home deny
Hotspot::Manager: Policy "deny" applied to interface "Home".
```

```
(config-hotspot)> policy Home Policy0
Hotspot::Manager: Policy "Policy0" applied to interface "Home".
```

```
(config-hotspot)> no policy Home
Hotspot::Manager: Interface "Home" policy cleared.
```

History

Version	Description
2.06	The **ip hotspot policy** command has been introduced.
2.12	Argument `policy` was added.

3.41.9 ip hotspot wake

Description Send Wake-on-LAN packet to private and protected interfaces of the host.

Prefix no No

Change settings No

Multiple input No

Interface type IP

Synopsis

(config-hotspot)> **wake** *‹mac›*

Arguments

Argument	Value	Description
mac	*MAC-address*	Host MAC address.

Example

```
(config-hotspot)> wake a8:1e:84:11:f1:22
Hotspot::Manager: WoL sent to host: a8:1e:84:11:f1:22.
```

History

Version	Description
2.08	The **ip hotspot wake** command has been introduced.

3.42 ip http lockout-policy

Description Set HTTP bruteforce detection parameters for public interfaces. By default, feature is enabled.

Command with **no** prefix disables bruteforce detection.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type IP

Synopsis

(config)> **ip http lockout-policy** *‹threshold›* [*‹duration›* [*‹observation-window›*]]

(config)> **no ip http lockout-policy**

Arguments

Argument	Value	Description
threshold	*Integer*	The number of failed attempts to log in. By default, 5 value is used.
duration	*Integer*	An authorization ban duration for the specified IP in minutes. By default, 15 value is used.
observation-window	*Integer*	Duration of suspicious activity observation in minutes. By default, 3 value is used.

Example

```
(config)> ip http lockout-policy 10 30 2
Http::Manager: Bruteforce detection is enabled.
```

```
(config)>  no ip http lockout-policy
Http::Manager: Bruteforce detection is disabled.
```

History

Version	Description
2.08	The **ip http lockout-policy** command has been introduced.

3.43 ip http log access

Description Enable debug mode for web-server (nginx). By default, feature is disabled.

Command with **no** prefix disables the debug mode.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type IP

Synopsis

(config)> **ip http log access**

(config)> **no ip http log access**

Example

```
(config)> ip http log access
Http::Manager: Enabled access logging.
```

```
(config)> no ip http log access
Http::Manager: Disabled access logging.
```

History

Version	Description
3.00	The **ip http log access** command has been introduced.

3.44 ip http log auth

Description Enable logging of failed authorization attempts to the system. By default, feature is disabled.

Command with **no** prefix disables logging.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type IP

Synopsis

(config)> **ip http log auth**

(config)> **no ip http log auth**

Example

```
(config)> ip http log auth
Http::Manager: Auth logging enabled.
```

```
(config)> no ip http log auth
Http::Manager: Auth logging disabled.
```

History

Version	Description
2.08	The **ip http log auth** command has been introduced.

3.45 ip http log webdav

Description

Enable logging of failed connection attempts to the *WebDAV* server. By default, feature is disabled.

Command with **no** prefix disables logging.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

IP

Synopsis

(config)> **ip http log webdav**

(config)> **no ip http log webdav**

Example

```
(config)> ip http log webdav
WebDav::Server: Enabled request tracing.
```

```
(config)> no ip http log webdav
WebDav::Server: Disabled request tracing.
```

History

Version	Description
3.04	The **ip http log webdav** command has been introduced.

3.46 ip http port

Description

Assign HTTP port for Web interface of City. By default, `80` value is used.

Command with **no** prefix resets HTTP port to default.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type IP

Synopsis

(config)> **ip http port** *‹port›*

(config)> **no ip http port**

Arguments

Argument	Value	Description
port	*Integer*	New HTTP port.

Example

```
(config)> ip http port 8080
Http::Manager: Port changed to 8080.
```

```
(config)> no ip http port
Http::Manager: Port reset to 80.
```

History

Version	Description
2.08	The **ip http port** command has been introduced.

3.47 ip http proxy

Description Access to a group of commands to configure HTTP proxy. If the proxy is not found, the command tries to create it.

Command with **no** prefix removes the proxy.

Prefix no Yes

Change settings Yes

Multiple input Yes

Interface type IP

Group entry (config-http-proxy)

Synopsis

(config)> **ip http proxy** *‹name›*

(config)> **no ip http proxy** *‹name›*

Arguments

Argument	Value	Description
name	*String*	HTTP proxy name.

Example

```
(config)> ip http proxy TEST
Http::Manager: Proxy "TEST" successfully created.
```

History

Version	Description
2.08	The **ip http proxy** command has been introduced.

3.47.1 ip http proxy auth

Description

Enable authorization for HTTP proxy. By default, the setting is disabled.

Command with **no** prefix disables HTTP proxy authorization.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

IP

Synopsis

(config-http-proxy)> **auth**

(config-http-proxy)> **no auth**

Example

```
(config-http-proxy)> auth
Http::Manager: Proxy password auth is enabled.
```

```
(config-http-proxy)> no auth
Http::Manager: Proxy password auth is disabled.
```

History

Version	Description
2.10	The **ip http proxy auth** command has been introduced.

3.47.2 ip http proxy domain

Description

Set domain name that specifies the *FQDN* of the virtual host.

Command with **no** prefix removes the setting.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

IP

Synopsis

(config-http-proxy)> **domain static** *‹domain›*

(config-http-proxy)> **no domain**

Arguments

Argument	Value	Description
domain	*String*	A domain name.

Example

```
(config-http-proxy)> domain static example.net
Http::Manager: Configured base domain for proxy: test.
```

```
(config-http-proxy)> no domain
Http::Manager: Removed ndns domain for proxy: test.
```

History

Version	Description
2.08	The **ip http proxy domain** command has been introduced.

3.47.3 ip http proxy domain ndns

Description

Set HTTP proxy domain through NDNS. If enabled, setting ip http proxy domain is deleted.

Command with **no** prefix removes the setting.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

IP

Synopsis

(config-http-proxy)> **domain ndns**

(config-http-proxy)> **no domain ndns**

Example

```
(config-http-proxy)> domain ndns
Http::Manager: Configured ndns domain for proxy: test.
```

```
(config-http-proxy)> no domain
Http::Manager: Removed ndns domain for proxy: test.
```

History

Version	Description
2.08	The **ip http proxy domain ndns** command has been introduced.

3.47.4 ip http proxy force-host

Description Enable the Host header rewriting for the upstream.

Command with **no** prefix disables the setting.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type IP

Synopsis

(config-http-proxy)> **force-host** ‹*force-host*›

(config-http-proxy)> **no force-host**

Arguments

Argument	Value	Description
force-host	*String*	IP-address or domain name.

Example

```
(config-http-proxy)> force-host 192.168.8.1
Http::Proxy: "modem": enabled Host header enforcing to ►
"192.168.8.1".
```

```
(config-http-proxy)> force-host modem.keenetic.pro
Http::Proxy: "modem": enabled Host header enforcing to ►
"modem.keenetic.pro".
```

```
(config-http-proxy)> no force-host
Http::Proxy: "modem": disabled Host header enforcing.
```

History

Version	Description
3.06	The **ip http proxy force-host** command has been introduced.

3.47.5 ip http proxy preserve-host

Description Set option to save the original header for the host when passing through a proxy.

Command with **no** prefix disable option.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type IP

Synopsis

(config-http-proxy)> **preserve-host**

(config-http-proxy)> **no preserve-host**

Example

```
(config-http-proxy)> preserve-host
Http::Manager: Proxy HTTP Host header preservation is enabled.
```

```
(config-http-proxy)> no preserve-host
Http::Manager: Proxy HTTP Host header preservation is disabled.
```

History

Version	Description
2.13	The **ip http proxy preserve-host** command has been introduced.

3.47.6 ip http proxy security-level

Description

Set the security level for HTTP proxy service. By default, `private` value is set.

Command with **no** prefix resets setting to default.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

IP

Synopsis

(config-http-proxy)> **security-level** (**public** | **private**)

(config-http-proxy)> **no security-level**

Arguments

Argument	Value	Description
`public`	*Keyword*	Access to the HTTP proxy is allowed for public, private and protected interfaces.
`private`	*Keyword*	Access to the HTTP proxy is allowed for private interfaces only.

Example

```
(config-http-proxy)> security-level public
Http::Proxy: "test1": set public security level.
```

```
(config-http-proxy)> no security-level
Http::Proxy: "test1": unset public security level.
```

History

Version	Description
3.05	The **ip http proxy security-level** command has been introduced.

3.47.7 ip http proxy upstream

Description Set HTTP or HTTPS server address for request redirecting.

Command with **no** prefix removes the setting.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type IP

Synopsis

(config-http-proxy)> **upstream** (**http** | **https**) (‹*mac*› | ‹*ip*› | ‹*fqdn*›) [‹*port*›]

(config-http-proxy)> **no upstream**

Arguments

Argument	Value	Description
`http`	*Keyword*	HTTP server.
`https`	*Keyword*	HTTPS server.
`mac`	*MAC-address*	MAC-address of server.
`ip`	*IP-address*	IP-address of server.
`fqdn`	*FQDN*	Full domain name of server.
`port`	*Integer*	The port number.

Example

```
(config-http-proxy)> upstream http 192.168.1.1 8080
Http::Manager: Proxy "TEST" upstream was set.
```

```
(config-http-proxy)> upstream https google.com 443
Http::Proxy: "modem": set https upstream google.com, port 443.
```

```
(config-http-proxy)> no upstream
Http::Manager: Remove upstream info for proxy "test".
```

History

Version	Description
2.08	The **ip http proxy upstream** command has been introduced.
3.05	`https` keyword was added.

3.47.8 ip http proxy x-real-ip

Description Enable X-Real-IP and X-Forwarded-For header support for HTTP proxy.

Command with **no** prefix disables headers.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type IP

Synopsis

(config-http-proxy)> **x-real-ip**

(config-http-proxy)> **no x-real-ip**

Example

```
(config-http-proxy)> x-real-ip
Http::Proxy: "test1": enabled X-Real-IP and X-Forwarded-For ►
headers.
```

```
(config-http-proxy)> no x-real-ip
Http::Proxy: "test1": disabled X-Real-IP and X-Forwarded-For ►
headers.
```

History

Version	Description
3.05	The **ip http proxy x-real-ip** command has been introduced.

3.48 ip http security-level

Description Set the security level for remote access to the Keenetic web interface. By default, `private` value is set.

Prefix no No

Change settings Yes

Multiple input No

Interface type IP

Synopsis

(config)> **ip http security-level** (**public** [**ssl**] | **private** | **protected**)

Arguments

Argument	Value	Description
`public`	*Keyword*	Access to the web interface is allowed for public, private and protected interfaces via HTTP and HTTPS.
`private`	*Keyword*	Access to the web interface is allowed for private interfaces.
`protected`	*Keyword*	Access to the web interface is allowed for private and protected interfaces.

Argument	Value	Description
ssl	*Keyword*	Access to the web interface is allowed for public interfaces via HTTPS only.

Example

```
(config)> ip http security-level protected
Http::Manager: Security level changed to protected.
```

```
(config)> ip http security-level public ssl
Http::Manager: Security level set to public SSL.
```

History

Version	Description
2.08	The **ip http security-level** command has been introduced.
3.00	Parameter **ssl** was added.

3.49 ip http ssl acme get

Description

Generate and sign SSL certificate for the specified domain name (by default, KeenDNS). Access from the Internet should be granted.

Prefix no

No

Change settings

No

Multiple input

No

Synopsis

(config)> **ip http ssl acme get** [*‹domain›*]

Arguments

Argument	Value	Description
domain	*String*	KeenDNS domain name.

Example

```
(config)> ip http ssl acme get mytest.keenetic.pro
Acme::Client: Obtaining certificate for domain ►
"mytest.keenetic.pro" is started.
```

History

Version	Description
2.11	The **ip http ssl acme get** command has been introduced.

3.50 ip http ssl acme revoke

Description

Revoke and remove SSL certificate for the specified domain name (KeenDNS, by default).

Prefix no No

Change settings No

Multiple input No

Synopsis

(config)> **ip http ssl acme revoke** *‹domain›*

Arguments

Argument	Value	Description
domain	*String*	KeenDNS domain name.

Example

```
(config)> ip http ssl acme revoke mytest.keenetic.pro
Acme::Client: Revoking certificate for domain ►
"mytest.keenetic.pro" is started.
```

History

Version	Description
2.11	The **ip http ssl acme revoke** command has been introduced.

3.51 ip http ssl acme list

Description Show a list of free Let`s Encrypt certificates in the system.

Prefix no No

Change settings No

Multiple input No

Synopsis

(config)> **ip http ssl acme list**

Example

```
(config)> ip http ssl acme list
     certificate:
                domain: cc6b5a71a7644903b51a5454.keenetic.io
    should-be-renewed: no
           is-expired: no
           issue-time: 2018-06-20T09:16:30.000Z
      expiration-time: 2018-09-17T09:16:30.000Z

     certificate:
                domain: mytest.keenetic.pro
    should-be-renewed: no
           is-expired: no
           issue-time: 2018-06-28T16:36:56.000Z
      expiration-time: 2018-09-25T16:36:56.000Z
```

History

Version	Description
2.11	The **ip http ssl acme list** command has been introduced.

3.52 ip http ssl enable

Description

Enable HTTP SSL server. By default, the server is disabled.

Command with **no** prefix disables SSL server.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

IP

Synopsis

(config)> **ip http ssl enable**

(config)> **no ip http ssl enable**

Example

```
(config)> ip http ssl enable
Http::Manager: Enabled SSL service.
```

```
(config)> no ip http ssl enable
Http::Manager: Disabled SSL service.
```

History

Version	Description
2.07	The **ip http ssl enable** command has been introduced.

3.53 ip http ssl redirect

Description

Enable automatic redirection on domains with SSL certificate. By default, the redirection is enabled.

Command with **no** prefix disables redirection.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

IP

Synopsis

(config)> **ip http ssl redirect**

(config)> **no ip http ssl redirect**

Example

```
(config)> ip http ssl redirect
Http::Manager: Redirect to SSL is enabled.
```

```
(config)> no ip http ssl redirect
Http::Manager: Redirect to SSL is disabled.
```

History

Version	Description
2.11	The **ip http ssl redirect** command has been introduced.

3.54 ip http x-frame-options

Description

Set X-Frame-Options header value for web-server (nginx) in Home network segment.

Command with **no** prefix disables the feature.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

IP

Synopsis

(config)> **ip http x-frame-options** *‹x-frame-options›*

(config)> **no ip http x-frame-options** *‹x-frame-options›*

Arguments

Argument	Value	Description
`x-frame-options`	*String*	The X-Frame-Option value.

Example

```
(config)> ip http x-frame-options DENY
Http::Manager: Set X-Frame-Options to "DENY".
```

```
(config)> no ip http x-frame-options DENY
Http::Manager: Disabled X-Frame-Options header.
```

History

Version	Description
3.05	The **ip http x-frame-options** command has been introduced.

3.55 ip name-server

Description

Configure DNS server IP-addresses. Addresses saved in this fashion are called static as opposite to dynamic — as registered by *PPP* or *DHCP* services.

Active, that addressed being used are the ones that have been registered most recently as compared to the others. Usually, the system uses the addresses which were obtained by several recent successfully connected *PPP* or *DHCP* services. If none of the services registers *DNS* addresses, static settings will be active. However, if after registering dynamic addresses the static settings are changed by the user, they become active until the new dynamic addresses are registered.

ip name-server command can be entered multiple times if several DNS-server addresses need to be setup. Moreover, each entered address can be associated with one or more domain names for working with specific areas, such as local names in the corporate network.

Command with **no** prefix removes the specified DNS server address from the static and the active lists if the command is furnished with arguments. If you use no arguments, the entire list of static addresses will be removed.

Prefix no Yes

Change settings Yes

Multiple input Yes

Interface type IP

Synopsis

(config)> **ip name-server** *‹address›* [**:** *‹port›*] [*‹domain›* [**on** *‹interface›*]]

(config)> **no ip name-server** [*‹address›* [**:** *‹port›*]] [*‹domain›* [**on** *‹interface›*]]

Arguments

Argument	Value	Description
address	*IP-address*	Name server address.
port	*Integer*	Name server port.
domain	*String*	Domain for which the server will be used. In resolving names the DNS-proxy first selects the address of the server with name best matching the requested domain. If the domain is not specified, the server will be used for all requests. Use "" as default domain. The maximum number of domains per one DNS entry is 16.
interface	*Interface name*	Interface name to configure.

Example

```
(config)> ip name-server 8.8.8.8 "" on ISP
Dns::InterfaceSpecific: Name server 8.8.8.8 added, domain ►
(default), interface ISP.
```

```
(config)>  no ip name-server
Dns::Manager: Static name server list cleared.
```

History

Version	Description
2.00	The **ip name-server** command has been introduced.
2.14	Argument `port` was added.

3.56 ip nat

Description

Enable translation of "local" addresses of network *network* or network behind the interface *interface*. For example, command `ip nat Home` means that all packets from the network `Home`, passing through the router will undergo IP spoofing.

Prefix no

Yes

Change settings

Yes

Multiple input

Yes

Interface type

IP

Synopsis

(config)> **ip nat** (‹*interface*› | ‹*address*› ‹*mask*›)

(config)> **no ip nat** (‹*interface*› | ‹*address*› ‹*mask*›)

Arguments

Argument	Value	Description
`interface`	*Interface name*	Source interface name (full name or an alias).
`address`	*IP-address*	Together with mask *mask* sets the range of source IP-addresses to be translated.
`mask`	*IP-mask*	Mask of a translation range. There are two ways to enter the mask: the canonical form (for example, `255.255.255.0`) and the form of prefix bit length (for example, `/24`).

Example

```
(config)> ip nat Home
Network::Nat: A NAT rule added.
```

```
(config)> no ip nat Home
Network::Nat: A NAT rule removed.
```

History

Version	Description
2.00	The **ip nat** command has been introduced.

3.57 ip nat full-cone

Description Enable mode *Full Cone NAT*. By default, the mode is disabled.

Command with **no** prefix disables the mode.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type IP

Synopsis

(config)> **ip nat full-cone**

(config)> **no ip nat full-cone**

Example

```
(config)> ip nat full-cone
Network::Nat: Full cone mode enabled.
```

```
(config)> no ip nat full-cone
Network::Nat: Full cone mode disabled.
```

History

Version	Description
3.01	The **ip nat full-cone** command has been introduced.

3.58 ip nat restricted-cone

Description Enable mode *Restricted NAT*. By default, the mode is disabled.

Command with **no** prefix disables the mode.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type IP

Synopsis

(config)> **ip nat restricted-cone**

(config)> **no ip nat restricted-cone**

Example

```
(config)> ip nat restricted-cone
Network::Nat: Restricted cone mode enabled.
```

```
(config)> no ip nat restricted-cone
Network::Nat: Restricted cone mode disabled.
```

History

Version	Description
3.01	The **ip nat restricted-cone** command has been introduced.

3.59 ip nat sstp

Description Enable translation for *SSTP* clients.

Command with **no** prefix removes the rule.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type IP

Synopsis

(config)> **ip nat sstp**

(config)> **no ip nat sstp**

Example

```
(config)> ip nat sstp
SstpServer::Nat: SSTP VPN NAT enabled.
```

```
(config)> no ip nat sstp
SstpServer::Nat: SSTP VPN NAT disabled.
```

History

Version	Description
2.12	The **ip nat sstp** command has been introduced.

3.60 ip nat vpn

Description Enable translation for VPN clients.

Command with **no** prefix removes the rule.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type IP

Synopsis

(config)> **ip nat vpn**

(config)> **no ip nat vpn**

Example

```
(config)> ip nat vpn
VpnServer::Nat: PPTP VPN NAT enabled.
```

```
(config)> no ip nat vpn
VpnServer::Nat: PPTP VPN NAT disabled.
```

History

Version	Description
2.04	The **ip nat vpn** command has been introduced.

3.61 ip policy

Description

Access to a group of commands to configure IP Policy — a default route selection rules for hosts and home network segments. If the IP Policy profile is not found, the command tries to create it. You can enter up to 16 profiles.

Command with **no** prefix removes the defined IP Policy profile from the list.

Prefix no

Yes

Change settings

Yes

Multiple input

Yes

Group entry

`(config-policy)`

Synopsis

(config)> **ip policy** *‹name›*

(config)> **no ip policy** *‹name›*

Arguments

Argument	Value	Description
name	*Policy name*	IP Policy name. Latin letters, numbers, hyphens and underscores are acceptable. Not more than 32 characters.

Example

```
(config)> ip policy Policy0
Network::PolicyTable: Created policy "Policy0".
```

```
(config)> no ip policy Policy0
Network::PolicyTable: Removed policy "Policy0".
```

History

Version	Description
2.12	The **ip policy** command has been introduced.

3.61.1 ip policy description

Description

Assign an arbitrary description to the specified IP Policy profile.

Command with **no** prefix removes description.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type IP

Synopsis

(config-policy)> **description** ‹*description*›

(config-policy)> **no description**

Arguments

Argument	Value	Description
description	*String*	An arbitrary description of the IP Policy. Latin letters, numbers, hyphens and underscores are acceptable. Not more than 256 characters.

Example

```
(config-policy)> description PolicyOne
Network::PolicyTable: "Policy0": updated description.
```

```
(config-policy)> no description
Network::PolicyTable: "Policy0": updated description.
```

History

Version	Description
2.12	The **ip policy description** command has been introduced.

3.61.2 ip policy multipath

Description Enable the function of simultaneous use of WAN connections in the balancing mode.

Command with **no** prefix disables the function.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type IP

Synopsis

(config-policy)> **multipath**

(config-policy)> **no multipath**

Example

```
(config-policy)> multipath
Network::PolicyTable: "Policy0": enable multipath.
```

```
(config-policy)> no multipath
Network::PolicyTable: "Policy0": disable multipath.
```

History

Version	Description
2.14	The **ip policy multipath** command has been introduced.

3.61.3 ip policy permit

Description

Permit IP Policy for the global interface. If single IP Policy is permitted for multiple interfaces, you can specify a priority for each of them.

Command with **no** prefix denies the IP Policy for specified interface. If you use no arguments, IP Policy will be denied for the entire list of interfaces.

Prefix no

Yes

Change settings

Yes

Multiple input

Yes

Interface type

IP

Synopsis

(config-policy)> **permit global** *‹interface›* [**order** *‹order›*]

(config-policy)> **no permit** [**global** *‹interface›*]

Arguments

Argument	Value	Description
`interface`	*`Interface name`*	Full interface name or an alias.
`order`	*`Integer`*	The priority of global interface to which the IP Policy is permitted. Can take values from 1 to 65534, but not more than the number of global interfaces.

Example

```
(config-policy)> permit global L2TP0 order 0
Network::PolicyTable: "Policy0": set permission to use L2TP0.
```

```
(config-policy)> no permit global L2TP0
Network::PolicyTable: "Policy0": set no permission to use L2TP0.
```

History

Version	Description
2.12	The **ip policy permit** command has been introduced.

3.61.4 ip policy permit auto

Description

Permit new connections for the IP Policy automatically. By default, the feature is disabled.

Command with **no** prefix removes auto permission.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type IP

Synopsis

(config-policy)> **permit auto**

(config-policy)> **no permit auto**

Example

```
(config-policy)> permit auto
Network::PolicyTable: "Policy0": set auto permission.
```

```
(config-policy)> no permit auto
Network::PolicyTable: "Policy0": set auto permission.
```

History

Version	Description
2.12	The **ip policy permit auto** command has been introduced.

3.61.5 ip policy rate-limit input

Description Add the input rate-limiting parameters to global interfaces of the IP Policy.

Command with **no** prefix removes the setting.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type IP

Synopsis

(config-policy)> **rate-limit** ‹*interface*› **input** (‹*rate*› | **auto**)

(config-policy)> **rate-limit** ‹*interface*› **no input**

Arguments

Argument	Value	Description
`interface`	*Interface name*	The name of a global IP interface to rate-limit its traffic for a group of policy assignees.
`rate`	*Integer*	The ingress rate limit in kbps. Can take values in the range from 64 to 1000000.
`auto`	*Keyword*	Auto-ingress mode.

Example

```
(config-policy)> rate-limit WifiMaster1/WifiStation0 input auto
Network::PolicyTable: "Policy0": set input rate limit to "auto".

(config-policy)> rate-limit WifiMaster1/WifiStation0 input 100000
Network::PolicyTable: "Policy0": set input rate limit to "100000" ►
kbps.

(config-policy)> rate-limit WifiMaster1/WifiStation0 no input
Network::PolicyTable: "Policy0": reset input rate limit.
```

History

Version	Description
3.05	The **ip policy rate-limit input** command has been introduced.

3.61.6 ip policy rate-limit output

Description

Add output rate-limiting parameters to global interfaces of the IP Policy.

Command with **no** prefix removes the setting.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

IP

Synopsis

(config-policy)> **rate-limit** *‹interface›* **output** *‹rate›*

(config-policy)> **rate-limit** *‹interface›* **no output**

Arguments

Argument	Value	Description
`interface`	*`Interface name`*	The name of a global IP interface to rate-limit its traffic for a group of policy assignees.
`rate`	*`Integer`*	The ingress rate limit in kbps. Can take values in the range from 64 to 1000000.

Example

```
(config-policy)> rate-limit WifiMaster1/WifiStation0 output 1000
Network::PolicyTable: "Policy0": set output rate limit to "1000" ►
kbps.

(config-policy)> rate-limit WifiMaster1/WifiStation0 no output
Network::PolicyTable: "Policy0": reset ouput rate limit.
```

History

Version	Description
3.05	The **ip policy rate-limit output** command has been introduced.

3.62 ip route

Description

Add a static route to the routing table to describe a rule of IP-packets transmission through a particular gateway or network interface.

As the destination network, one can specify `default` keyword. In this case, a default route will be created.

Command with **no** prefix removes the route with the specified parameters.

Prefix no

Yes

Change settings

Yes

Multiple input

Yes

Interface type

IP

Synopsis

(config)> **ip route** (*‹network› ‹mask›* | *‹host›* | **default**) (*‹gateway›* [*interface*] | *‹interface›*) [**auto**] [*metric*]

(config)> **no ip route** (*‹network› ‹mask›* | *‹host›* | **default**) [*‹gateway›* | *‹interface›*] [*metric*]

Arguments

Argument	Value	Description
`network`	*IP-address*	IP-address of the destination network.
`mask`	*IP-mask*	Mask of the destination network. There are two ways to enter the mask: in the canonical form (for example, `255.255.255.0`) and in the form of prefix bit length (for example, `/24`).
`host`	*IP-address*	IP-address of the destination node.
`default`	*Keyword*	Helps specify default routes.
`interface`	*Interface name*	Interface full name or an alias. Specified as the direction of the packet transferring, if the interface has a point-to-point channel connected that requires no additional addressing within the channel. If priority **interface ip global** is set on the interface, the route is added to the system table only if there is no other higher priority route with the same address.

Argument	Value	Description
gateway	*IP-address*	IP-address of the router in a directly connected network. Can be specified along with the interface name, if it is required to specify **interface ip global** priority. If no interface is specified, the systemd determines it automatically based on the current IP settings.
auto	*Keyword*	Allows you to apply the route when specified gateway becomes available.
metric	*Integer*	Route metrics. Ignored in the current implementation.

Example

```
(config)> ip route default Home
Network::RoutingTable: Added static route: 0.0.0.0/0 via Home.
```

```
(config)> ip route default Home
```

History

Version	Description
2.00	The **ip route** command has been introduced.

3.63 ip search-domain

Description

Assign search domain to resolve hostnames that are not fully qualified.

Command with **no** prefix removes the setting.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config)> **ip search-domain** *‹domain›*

(config)> **no ip search-domain**

Arguments

Argument	Value	Description
domain	*String*	The domain name to assign.

Example

```
(config)> ip search-domain my.example
```

```
(config)> no ip search-domain my.example
```

History

Version	Description
2.00	The **ip search-domain** command has been introduced.

3.64 ip sip alg direct-media

Description

Replace IP address in `Owner` field of SDP. This feature is used to not configure port forwarding separately for VoIP traffic. By default, the setting is disabled.

Command with **no** prefix disables the feature.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config)> **ip sip alg direct-media**

(config)> **no ip sip alg direct-media**

Example

```
(config)> ip sip alg direct-media
Sip::Alg: Direct media enabled.
```

```
(config)> no ip sip alg direct-media
Sip::Alg: Direct media disabled.
```

History

Version	Description
2.11	The **ip sip alg direct-media** command has been introduced.

3.65 ip sip alg port

Description

Specify a port number for SIP messages other than the default port. By default, port number is `5060`.

Command with **no** prefix resets port to default.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config)> **ip sip alg port** *‹port›*

(config)> **no ip sip alg port**

Arguments

Argument	Value	Description
`port`	*Integer*	The port number.

Example

```
(config)> ip sip alg port 7090
Sip::Alg: Port set to 7090.
```

```
(config)> no ip sip alg port
Sip::Alg: Port reset to default.
```

History

Version	Description
2.12	The **ip sip alg port** command has been introduced.

3.66 ip ssh

Description Access to a group of commands to manage SSH-server.

Prefix no No

Change settings No

Multiple input No

Interface type IP

Group entry `(config-ssh)`

Synopsis

(config)> **ip ssh**

Example

```
(config)> ip ssh
(config-ssh)>
```

History

Version	Description
2.12	The **ip ssh** command has been introduced.

3.66.1 ip ssh cipher

Description Set a symmetric key cipher for SSH session.

Command with **no** prefix removes the specified cipher.

Prefix no Yes

Change settings Yes

Multiple input Yes

Interface type IP

Synopsis

(config-ssh)> **cipher** *‹cipher›*

(config-ssh)> **no cipher** *‹cipher›*

Arguments

Argument	Value	Description
cipher	chacha20-poly1305@openssh.com	An encryption algorithm ChaCha20-Poly1305.
	aes128-ctr	An encryption algorithm AES128-CTR.
	aes256-ctr	An encryption algorithm AES1256-CTR.
	aes128-gcm@openssh.com	An encryption algorithm AES128-GCM.
	aes256-gcm@openssh.com	An encryption algorithm AES256-GCM.

Example

```
(config-ssh)> cipher chacha20-poly1305@openssh.com
Ssh::Manager: Added cipher "chacha20-poly1305@openssh.com".
```

```
(config-ssh)> no cipher chacha20-poly1305@openssh.com
Ssh::Manager: Use default ciphers.
```

History

Version	Description
3.04	The **ip ssh cipher** command has been introduced.

Version	Description
3.05	New encryption algorithms aes128-gcm@openssh.com, aes256-gcm@openssh.com were added.

3.66.2 ip ssh keygen

Description Regeneration of a given type key.

Prefix no No

Change settings Yes

Multiple input No

Interface type IP

Synopsis

(config-ssh)> **keygen** *‹keygen›*

Arguments

Argument	Value	Description
keygen	default	Automatic generation of a new open key RSA2048 + ECDSA-NISTP521.

Argument	Value	Description
	`rsa-1024`	Automatic generation of a new open RSA-key with a length of 1024 bits.
	`rsa-2048`	Automatic generation of a new open RSA-key with a length of 2048 bits.
	`rsa-4096`	Automatic generation of a new open RSA-key with a length of 4096 bits.
	`ecdsa-nistp256`	Automatic generation of a new open ECDSA-key with a length of 256 bits.
	`ecdsa-nistp384`	Automatic generation of a new open ECDSA-key with a length of 384 bits.
	`ecdsa-nistp521`	Automatic generation of a new open ECDSA-key with a length of 521 bits.
	`ed25519`	Automatic generation of a new open ED25519 key with a length of 256 bits.

Example

```
(config-ssh)> keygen default
Ssh::Manager: Key generation is in progress...
```

History

Version	Description
2.12	The **ip ssh keygen** command has been introduced.

3.66.3 ip ssh lockout-policy

Description

Set SSH bruteforce detection parameters for public interfaces. By default, feature is enabled.

Command with **no** prefix disables bruteforce detection.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

IP

Synopsis

(config)> **ip ssh lockout-policy** ‹*threshold*› [‹*duration*› [‹*observation-window*›]]

(config)> **no ip ssh lockout-policy**

Arguments

Argument	Value	Description
`threshold`	*Integer*	The number of failed attempts to log in. By default, 5 value is used.

Argument	Value	Description
duration	*Integer*	An authorization ban duration for the specified IP in minutes. By default, 15 value is used.
observation-window	*Integer*	Duration of suspicious activity observation in minutes. By default, 3 value is used.

Example

```
(config-ssh)> lockout-policy 10 30 2
Ssh::Manager: Bruteforce detection is reconfigured.
```

```
(config-ssh)> no lockout-policy
Ssh::Manager: Bruteforce detection is disabled.
```

History

Version	Description
2.12	The **ip ssh lockout-policy** command has been introduced.

3.66.4 ip ssh port

Description

Specify port number for SSH connection. By default, 22 port number is used.

Command with **no** prefix resets port number to default.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

IP

Synopsis

(config-ssh)> **port** *‹number›*

(config-ssh)> **no port**

Arguments

Argument	Value	Description
number	*Integer*	Port number. Can take values from 1 to 65535 inclusively.

Example

```
(config-ssh)> port 2626
Ssh::Manager: Port changed to 2626.
```

```
(config-ssh)> no port
Ssh::Manager: Port reset to 22.
```

History

Version	Description
2.12	The **ip ssh port** command has been introduced.

3.66.5 ip ssh security-level

Description Set SSH security level. By default, `private` value is set.

Prefix no No

Change settings Yes

Multiple input No

Interface type IP

Synopsis

(config-ssh)> **security-level** (**public** | **private** | **protected**)

Arguments

Argument	Value	Description
`public`	*Keyword*	Access to the SSH-server is allowed for public, private and protected interfaces.
`private`	*Keyword*	Access to the SSH-server is allowed for private interfaces.
`protected`	*Keyword*	Access to the SSH-server is allowed for private and protected interfaces.

Example

```
(config-ssh)> security-level protected
Ssh::Manager: Security level changed to protected.
```

History

Version	Description
2.12	The **ip ssh security-level** command has been introduced.

3.66.6 ip ssh session timeout

Description Set the lifetime of inactive session for SSH connection. By default, `300` value is used, i.e. the function of activity tracking within a session is disabled.

Command with **no** prefix resets timeout to default.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type IP

Synopsis

(config-ssh)> **session timeout** ‹*timeout*›

(config-ssh)> **no session timeout**

Arguments

Argument	Value	Description
`timeout`	*Integer*	The lifetime of inactive session. Can take values from 5 to 2^{32}-1 seconds inclusively.

Example

```
(config-ssh)> session timeout 123456
Ssh::Manager: A session timeout value set to 123456 seconds.
```

```
(config-ssh)> no session timeout
Ssh::Manager: A session timeout reset.
```

History

Version	Description
3.03	The **ip ssh session timeout** command has been introduced.

3.67 ip static

Description

Define translation rule for global and local IP-addresses. If *interface* or *network* corresponds to the interface with secutity level `public`, then the destination address translation (DNAT) will occure. If *to-address* corresponds to the interface with secutity level `public`, then source address translation (SNAT) will occure. TCP/UDP port number is always treated as the destination port.

If *network* corresponds to a single address and this address is equal to *to-address*, then this rule will prohibit the translation of the specified address, which could have been done based on the specified rules **ip nat**.

ip static rules have higher priority than the **ip nat** rules.

Prefix no Yes

Change settings Yes

Multiple input Yes

Interface type IP

Synopsis

(config)> **ip static** [‹*protocol*›] (‹*interface*› | (‹*address*› ‹*mask*›))
(‹*port*› **through** ‹*end-port*› (‹*to-address*› | ‹*to-host*›) |
[*port*] (‹*to-address*› | ‹*to-host*›) [*to-port*] |
‹*to-address*› | ‹*to-host*› | ‹*to-interface*›)

(config)> **no ip static** [‹*protocol*›] (‹*interface*› | (‹*address*› ‹*mask*›))

(*‹port›* **through** *‹end-port›* (*‹to-address›* | *‹to-host›*) |
[*port*] (*‹to-address›* | *‹to-host›*) [*to-port*] |
‹to-address› | *‹to-host›* | *‹to-interface›*)

Arguments

Argument	Value	Description
protocol	tcp	*TCP* protocol.
	udp	*UDP* protocol.
interface	*Interface name*	Input interface name (full name or alias).
comment	*String*	User's notes with symbol ! before them.
address	*IP-address*	Along with mask *mask* sets the range of destination IP-addresses that are to be translated.
mask	*IP-mask*	Translation range mask. There are two ways to enter the mask: the canonical form (for example, 255.255.255.0) and the form of prefix bit length (for example, /24).
port	*Integer*	TCP/UDP port number for which a translation request comes. If not specified, all incoming requests will be translated.
end-port	*Integer*	The end of the range of ports.
to-address	*IP-address*	The destination address after translation.
to-host	*MAC-address*	The destination MAC-address after translation. Only MAC-address from known hosts are accepted. If the known host is deleted, then the associated rules will be deleted too.
to-port	*Integer*	TCP/UDP port number after translation. If not specified, the destination port remains the same.

Example

Let there be a router between the “local” network 172.16.1.0/24 (secutity level private) and “global” network 10.0.0.0/16 (secutity level public). It is required that all requests coming to the “global” interface of this router on port 80 to be broadcast to the “local” server with the address 172.16.1.33. The sequence of commands to implement the required schema might look like this:

```
(config)> interface Home ip address 192.168.1.1/24
Network::Interface::Ip: "Bridge0": IP address is 192.168.1.1/24.
```

```
(config)> ip static tcp ISP 80 172.16.1.33 80
Network::StaticNat: Static NAT rule has been added.
```

```
(config)> ip static tcp ISP 21 00:0e:c6:a1:22:11 !test
Network::StaticNat: Static NAT rule is already there.
```

```
(config)> ip static disable
Network::StaticNat: Static NAT disable unchanged.
```

```
(config)> no ip static
Network::StaticNat: Static NAT rules have been removed.
```

History

Version	Description
2.00	The **ip static** command has been introduced.
2.06	The `to-host` argument has been added.

3.68 ip static rule

Description

Disable IP-address translation rule or set rule operation time by schedule.

Command with **no** prefix enables the rule or removes the rule schedule.

Prefix no

Yes

Change settings

Yes

Multiple input

Yes

Interface type

IP

Synopsis

(config)> **ip static rule** ‹*index*› (**disable** | **schedule** ‹*schedule*›)

(config)> **no ip static rule** ‹*index*› (**disable** | **schedule**)

Arguments

Argument	Value	Description
`index`	*Integer*	The translation rule number.
`disable`	*Keyword*	Disable the translation rule.
`schedule`	*Schedule name*	The name of the schedule that was created with **schedule** group of commands.

Example

```
(config)> ip static rule 0 schedule test_schedule
Network::StaticNat: Static NAT rule schedule applied.
```

```
(config)> ip static rule 0 disable
Network::StaticNat: Static NAT rule disabled.
```

```
(config)> no ip static rule 0 disable
Network::StaticNat: Static NAT rule enabled.
```

```
(config)> no ip static rule 0 schedule
Network::StaticNat: Static NAT rule schedule removed.
```

History

Version	Description
2.08	The **ip static rule** command has been introduced.

3.69 ip telnet

Description Access to a group of commands to manage Telnet-server.

Prefix no No

Change settings No

Multiple input No

Interface type IP

Group entry `(config-telnet)`

Synopsis

(config)> **ip telnet**

Example

```
(config)> ip telnet
(config-telnet)>
```

History

Version	Description
2.08	The **ip telnet** command has been introduced.

3.69.1 ip telnet lockout-policy

Description Set Telnet bruteforce detection parameters for public interfaces. By default, feature is enabled.

Command with **no** prefix disables bruteforce detection.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type IP

Synopsis

(config)> **ip telnet lockout-policy** *‹threshold›* [*‹duration›* [*‹observation-window›*]]

(config)> **no ip telnet lockout-policy**

Arguments

Argument	Value	Description
`threshold`	*Integer*	The number of failed attempts to log in. By default, 5 value is used.
`duration`	*Integer*	An authorization ban duration for the specified IP in minutes. By default, 15 value is used.
`observation-window`	*Integer*	Duration of suspicious activity observation in minutes. By default, 3 value is used.

Example

```
(config)> ip telnet lockout-policy 10 30 2
Telnet::Manager: Bruteforce detection is reconfigured.
```

History

Version	Description
2.08	The **ip telnet lockout-policy** command has been introduced.

3.69.2 ip telnet port

Description

Specify port number for telnet connection. By default, 23 port number is used.

Command with **no** prefix resets port number to default.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

IP

Synopsis

(config-telnet)> **port** *‹number›*

(config-telnet)> **no port**

Arguments

Argument	Value	Description
`number`	*Integer*	Port number. Can take values from 1 to 65535 inclusively.

Example

```
(config-telnet)> port 2525
Telnet::Server: Port unchanged.
```

```
(config-telnet)> no port
Telnet::Server: Port unchanged.
```

History

Version	Description
2.08	The **ip telnet port** command has been introduced.

3.69.3 ip telnet security-level

Description Set Telnet security level. By default, `private` value is set.

Prefix no No

Change settings Yes

Multiple input No

Interface type IP

Synopsis

(config-telnet)> **security-level** (**public** | **private** | **protected**)

Arguments

Argument	Value	Description
`public`	*Keyword*	Access to the Telnet-server is allowed for public, private and protected interfaces.
`private`	*Keyword*	Access to the Telnet-server is allowed for private interfaces.
`protected`	*Keyword*	Access to the Telnet-server is allowed for private and protected interfaces.

Example

```
(config-telnet)> security-level protected
Telnet::Manager: Security level changed to protected.
```

History

Version	Description
2.08	The **ip telnet security-level** command has been introduced.

3.69.4 ip telnet session max-count

Description Set the maximal number of simultaneous sessions for telnet connection. By default, 4 is used.

Command with **no** prefix resets count to default.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type IP

Synopsis

(config-telnet)> **session max-count** *‹count›*

(config-telnet)> **no session max-count**

Arguments

Argument	Value	Description
`count`	*Integer*	The maximal number of simultaneous sessions. Can take values from 1 to 4 inclusively.

Example

```
(config-telnet)> session max-count 4
Telnet::Server: The maximum session count set to 4.
```

```
(config-telnet)> no session max-count
Telnet::Server: The maximum session count reset to 4.
```

History

Version	Description
2.08	The **ip telnet session max-count** command has been introduced.

3.69.5 ip telnet session timeout

Description Set the lifetime of inactive session for telnet connection. By default, `300` value is used which means that the function of activity tracking within a session is disabled.

Command with **no** prefix resets timeout to default.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type IP

Synopsis

(config-telnet)> **session timeout** *‹timeout›*

(config-telnet)> **no session timeout**

Arguments

Argument	Value	Description
`timeout`	*Integer*	The lifetime of inactive session. Can take values from 5 to 2^{32} - 1 seconds inclusively.

Example

```
(config-telnet)> session timeout 600
Telnet::Server: A session timeout value set to 600 seconds.
```

```
(config-telnet)> no session timeout
Telnet::Server: A session timeout reset.
```

History

Version	Description
2.08	The **ip telnet session timeout** command has been introduced.

3.70 ip traffic-shape host

Description

Set the limit of data rate on a specified known host in both directions. By default speed is not limited.

Command with **no** prefix removes the setting for specified host. If you use no arguments, the entire list of rate limits for all hosts will be removed.

Prefix no

Yes

Change settings

Yes

Multiple input

Yes

Interface type

IP

Synopsis

(config)> **ip traffic-shape host** *‹mac›* **rate** *‹rate›* [**asymmetric** *‹upstream-rate›*] [**schedule** *‹schedule›*]

(config)> **no ip traffic-shape host** [*‹mac›*]

Arguments

Argument	Value	Description
mac	*MAC-address*	MAC-address of the known host.
rate	*Integer*	Value of data download rate in Kbps. Limit sould be in the range from 64 Kbps to 1 Gbps.
upstream-rate	*Integer*	Data upload rate in Kbps. Value can be in the range from 64 Kbps to 1 Gbps.
schedule	*Schedule name*	The name of the schedule that was created with **schedule** group of commands.

Example

```
(config)> ip traffic-shape host a8:1e:82:81:f1:21 rate 80
TrafficControl::Manager: "a8:1e:82:81:f1:21" host rate limited ►
to DL 80 / UL 80 Kbits/sec.
```

```
(config)> ip traffic-shape host a8:1e:82:81:f1:21 rate 80 ►
asymmetric 64
TrafficControl::Manager: "a8:1e:82:81:f1:21" host rate limited ►
to DL 80 / UL 64 Kbits/sec..
```

```
(config)> ip traffic-shape host a8:1e:82:81:f1:21 rate 80 ►
asymmetric 64 schedule Update
TrafficControl::Manager: "a8:1e:82:81:f1:21" host rate limited ►
to DL 80 / UL 64 Kbits/sec (controlled by schedule Update).
```

```
(config)> no ip traffic-shape host a8:1e:82:81:f1:21
TrafficControl::Manager: Rate limit removed for host ►
"a8:1e:82:81:f1:21".
```

```
(config)> no ip traffic-shape host a8:1e:82:81:f1:21
TrafficControl::Manager: Rate limit removed for host ►
"a8:1e:82:81:f1:21".
```

```
(config)> no ip traffic-shape host
TrafficControl::Manager: Rate limits for all hosts removed.
```

History

Version	Description
2.05	The **ip traffic-shape host** command has been introduced.
2.08	The **schedule** argument was added.

Version	Description
3.04	The **upstream-rate** argument was added.

3.71 ip traffic-shape unknown-host

Description

Set the data rate limitation for unregistered devices in both directions. By default, speed is unlimited.

Command with **no** prefix removes the setting.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

IP

Synopsis

(config)> **ip traffic-shape unknown-host rate** *‹rate›* [**asymmetric** *‹upstream-rate›*]

(config)> **no ip traffic-shape unknown-host rate**

Arguments

Argument	Value	Description
rate	*Integer*	The data download rate in Kbps. Value sould be in the range from 64 Kbps to 1 Gbps.

Argument	Value	Description
upstream-rate	*Integer*	Data upload rate in Kbps. Value can be in the range from 64 Kbps to 1 Gbps.

Example

```
(config)> ip traffic-shape unknown-host rate 80
TrafficControl::Manager: Rate limit for unknown hosts set to 80 ►
Kbits/sec.
```

```
(config)> ip traffic-shape unknown-host rate 80 asymmetric 64
TrafficControl::Manager: Rate limit for unknown hosts set to ►
80/64 Kbits/sec.
```

```
(config)> no ip traffic-shape unknown-host rate
TrafficControl::Manager: Rate limit for unknown hosts removed.
```

History

Version	Description
2.09	The **ip traffic-shape unknown-host** command has been introduced.
3.04	The **upstream-rate** argument was added.

3.72 ipv6 firewall

Description

Enable IPv6 firewall. By default, the setting is enabled.

Command with **no** prefix removes the setting.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config)> **ipv6 firewall**

(config)> **no ipv6 firewall**

Example

```
(config)> ipv6 firewall
```

```
(config)> no ipv6 firewall
```

History

Version	Description
2.06	The **ipv6 firewall** command has been introduced.

3.73 ipv6 local-prefix

Description

Configure a local (ULA) prefix. Argument can be a literal prefix or **default**, which generates a persistent unique prefix automatically.

Command with **no** prefix disables the setting.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config)> **ipv6 local-prefix** (**default** | *‹prefix›*)

(config)> **no ipv6 local-prefix** [**default** | *‹prefix›*]

Arguments

Argument	Value	Description
`default`	*Keyword*	Generate persistent unique prefix.
`prefix`	*Prefix*	Local ULA prefix. Must be a valid prefix in the block `fd00::/8` with a prefix length no longer than 48.

Example

```
(config)> ipv6 local-prefix default
Ip6::Prefixes: Default ULA prefix enabled.
```

```
(config)> ipv6 local-prefix fd01:db8:43::/48
Ip6::Prefixes: Added static prefix: fd01:db8:43::/48.
```

```
(config)> no ipv6 local-prefix default
Ip6::Prefixes: Default ULA prefix disabled.
```

```
(config)> no ipv6 local-prefix fd01:db8:43::/48
Ip6::Prefixes: Deleted static prefix: fd01:db8:43::/48.
```

History

Version	Description
2.00	The **ipv6 local-prefix** command has been introduced.

3.74 ipv6 name-server

Description

Configure DNS server IPv6-addresses. Addresses saved in this fashion are called static as opposite to dynamic — as registered by *PPP* or *DHCP* services.

ipv6 name-server command can be entered multiple times if several DNS-server addresses need to be setup.

Command with **no** prefix removes the specified DNS server address from the static and the active lists if the command is furnished with arguments, or clears the list of static addresses if the command has no arguments.

Prefix no Yes

Change settings Yes

Multiple input Yes

Synopsis

(config)> **ipv6 name-server** *‹address›* [*‹domain›*]

(config)> **no ipv6 name-server** [*‹address›* [*‹domain›*]]

Arguments

Argument	Value	Description
address	*IPv6-address*	Name server address.
domain	*String*	Domain for which the server will be used. In resolving names the DNS-proxy first selects the address of the server with name best matching the requested domain. If the domain is not specified, the server will be used for all requests. Use "" as default domain.

Example

```
(config)> ipv6 name-server 2001:4860:4860::8888
Dns::Manager: Name server 2001:4860:4860::8888 added, domain ►
(default).
```

```
(config)> ipv6 name-server 2001:4860:4860::8888 google.com
Dns::Manager: Name server 2001:4860:4860::8888 added, domain ►
google.com.
```

```
(config)> no ipv6 name-server
Dns::Manager: Static name server list cleared.
```

History

Version	Description
2.00	The **ipv6 name-server** command has been introduced.

3.75 ipv6 pass

Description Enable Pass Through mode on the router for IPv6-packets. By default, the feature is disabled.

Command with **no** prefix disables the function.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config)> **ipv6 pass through** *‹wan-iface›* *‹lan-iface›*

(config)> **no ipv6 pass**

Arguments

Argument	Value	Description
wan-iface	*Interface name*	Full WAN-interface name or an alias.
lan-iface	*Interface name*	Full LAN-interface name or an alias.

Example

```
(config)> ipv6 pass through ISP Home
Ip6::Pass: Configured pass from "GigabitEthernet1" to "Bridge0".

(config)> no ipv6 pass
Ip6::Pass: Disabled.
```

History

Version	Description
2.06	The **ipv6 pass** command has been introduced.

3.76 ipv6 route

Description Add a static route to the routing table to describe a rule of IPv6-packets transmission through a particular gateway or network interface.

As the destination network keyword `default` can be specified. In this case, a default route will be created.

Command with **no** prefix removes the route with the specified parameters.

Prefix no Yes

Change settings Yes

Multiple input Yes

Synopsis

(config)> **ipv6 route** (*‹prefix›* | **default**) (*‹interface›* [*‹gateway›*] | *‹gateway›*)

(config)> **no ipv6 route** (*‹prefix›* | **default**) (*‹interface›* [*‹gateway›*] | *‹gateway›*)

Arguments

Argument	Value	Description
prefix	*Prefix*	IPv6 prefix.
default	*Keyword*	Default prefix.

Argument	Value	Description
interface	*Interface name*	Full interface name or an alias.
gateway	*IP-address*	IP-address of the router in a directly connected network.

Example

```
(config)> ipv6 route 2002:c100:aeb5::/48 ISP
route added
```

```
(config)> no ipv6 route 2002:c100:aeb5::/48 ISP
route erased
```

```
(config)> ipv6 route 2002:c100:aeb5:100::/56 2002:c100:aeb5::33
route added
```

```
(config)> no ipv6 route 2002:c100:aeb5:100::/56 2002:c100:aeb5::33
route erased
```

History

Version	Description
2.00	The **ipv6 route** command has been introduced.
2.11	gateway argument has been added.

3.77 ipv6 static

Description

Define the rule to allow incoming connection to a specified port of a registered home network device.

ipv6 firewall should be enabled.

Command with **no** prefix removes the rule.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config)> **ipv6 static** ‹*protocol*› [‹*interface*›] ‹*mac*› ‹*port*› [**through** ‹*end-port*›]

(config)> **no ipv6 static** [‹*protocol*› [‹*interface*›] ‹*mac*› ‹*port*› [**through** ‹*end-port*›]]

Arguments

Argument	Value	Description
protocol	tcp	*TCP* protocol.
	udp	*UDP* protocol.

Argument	Value	Description
interface	*Interface name*	Input interface name (full name or an alias).
mac	*MAC-address*	MAC-address of host.
port	*Integer*	TCP/UDP port number for which incoming request comes.
end-port	*Integer*	The end of the range of ports.

Example

```
(config)> ipv6 static tcp ISP 64:a2:f9:51:b4:8a 80 through 80
Ip6::Firewall: Rule updated.

(config)> no ipv6 static tcp ISP 64:a2:f9:51:b4:8a 80 through 80
Ip6::Firewall: Static rule removed.
```

History

Version	Description
2.12	The **ipv6 static** command has been introduced.

3.78 ipv6 subnet

Description Access to a group of commands to configure a LAN IPv6 segment. If the segment is not found, the command tries to create it.

Prefix no Yes

Change settings Yes

Multiple input Yes

Group entry (config-subnet)

Synopsis

(config)> **ipv6 subnet** *‹name›*

(config)> **no ipv6 subnet** [*‹name›*]

Arguments

Argument	Value	Description
name	*String*	Subnet name or an alias.

Example

```
(config)> ipv6 subnet Default
(config-subnet)>
```

History

Version	Description
2.00	The **ipv6 subnet** command has been introduced.

3.78.1 ipv6 subnet bind

Description Bind the subnet to an interface.

Command with **no** prefix cancels binding.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config-subnet)> **bind** *‹bind›*

(config-subnet)> **no bind**

Arguments

Argument	Value	Description
bind	*Interface name*	Full interface name or an alias.

Example

```
(config-subnet)> bind WifiMaster0/AccessPoint1
Ip6::Subnets: Interface "WifiMaster0/AccessPoint1" bound to ►
subnet "Default".
```

```
(config-subnet)> no bind
Ip6::Subnets: Interface unbound from subnet "Default".
```

History

Version	Description
2.00	The **ipv6 subnet bind** command has been introduced.

3.78.2 ipv6 subnet mode

Description Select the address configuration mode for hosts in the subnet. Exclusive options are **dhcp** and **slaac**. The former will enable a local DHCPv6 server for the purposes of address assignment, and the latter will enable SLAAC (Stateless Address Autoconfiguration).

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config-subnet)> **mode** *‹mode›*

(config-subnet)> **no mode**

Arguments

Argument	Value	Description
mode	slaac	Enable SLAAC (stateless autoconfiguration).

Argument	Value	Description
	`dhcp`	Enable DHCPv6 server (stateful autoconfiguration).

Example

```
(config-subnet)> mode dhcp
Ip6::Subnets: Subnet "Default" enabled as DHCP.
```

```
(config-subnet)> no mode
Ip6::Subnets: Subnet "Default" disabled.
```

History

Version	Description
2.00	The **ipv6 subnet mode** command has been introduced.

3.78.3 ipv6 subnet number

Description Assign the subnet ID, which will determine the advertised prefix for the segment. Must be unique across subnets.

Prefix no No

Change settings Yes

Multiple input No

Synopsis

(config-subnet)> **number** ‹*number*›

Arguments

Argument	Value	Description
`number`	*Integer*	Unique subnet ID.

Example

```
(config-subnet)> number 2
Ip6::Subnets: Number 2 assigned to subnet "Default".
```

History

Version	Description
2.00	The **ipv6 subnet number** command has been introduced.

3.79 isolate-private

Description Prohibit data transfer between any interfaces with security level `private`. Enabled by default.

Command with **no** prefix cancels the command, allowing data transfer between `private` interfaces.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config)> **isolate-private**

(config)> **no isolate-private**

Example

```
(config)> isolate-private
Netfilter::Manager: Private networks isolated.
```

```
(config)> no isolate-private
Netfilter::Manager: Private networks not isolated.
```

History

Version	Description
2.00	The **isolate-private** command has been introduced.

3.80 kabinet

Description Access to a group of commands to configure КАБiNET authenticator parameters.

Command with **no** prefix resets all parameters to default.

Prefix no Yes

Change settings Yes

Multiple input No

Group entry (kabinet)

Synopsis

(config)> **kabinet**

(config)> **no kabinet**

Example

```
(config)> kabinet
(kabinet)>
```

```
(config)> no kabinet
Kabinet::Authenticator: A configuration reset.
```

History

Version	Description
2.02	The **kabinet** command has been introduced.

3.80.1 kabinet access-level

Description Set an access level for KABiNET authenticator. By default, access level `internet` is used.

Command with **no** prefix resets level to default.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(kabinet)> **access-level** *‹level›*

(kabinet)> **no access-level**

Arguments

Argument	Value	Description
`level`	`lan`	Access level value.
	`internet`	

Example

```
(kabinet)> access-level lan
Kabinet::Authenticator: An access level set to "lan".
```

```
(kabinet)> access-level internet
Kabinet::Authenticator: An access level set to "internet".
```

```
(kabinet)> no access-level
Kabinet::Authenticator: An access level reset to "internet".
```

History

Version	Description
2.02	The **kabinet access-level** command has been introduced.

3.80.2 kabinet interface

Description Bind KABiNET authenticator to the specified interface.

Command with **no** prefix unbinds interface.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(kabinet)> **interface** *‹interface›*

(kabinet)> **no interface**

Arguments

Argument	Value	Description
`interface`	*`Interface name`*	Full interface name or an alias. You can see the list of available interfaces with help of **interface** [Tab] command.

Example

```
(kabinet)> interface [Tab]

 Usage template:
        interface {interface}

   Choose:
         GigabitEthernet1
                      ISP
 WifiMaster0/AccessPoint2
 WifiMaster1/AccessPoint1
 WifiMaster0/AccessPoint3
 WifiMaster0/AccessPoint0
              AccessPoint
```

```
(kabinet)> interface ISP
Kabinet::Authenticator: Bound to GigabitEthernet1.
```

```
(kabinet)> no interface
Kabinet::Authenticator: Interface binding cleared.
```

History

Version	Description
2.02	The **kabinet interface** command has been introduced.

3.80.3 kabinet password

Description

Set a password for КАБiNET authenticator. By default, password is not set.

Command with **no** prefix cleares the password.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(kabinet)> **password** ‹*password*›

(kabinet)> **no password**

Arguments

Argument	Value	Description
`password`	*`String`*	The password for authentication.

Example

```
(kabinet)> password 123456789
Kabinet::Authenticator: A password set.
```

```
(kabinet)> no password
Kabinet::Authenticator: A password cleared.
```

History

Version	Description
2.02	The **kabinet password** command has been introduced.

3.80.4 kabinet port

Description

Set the server port for KABiNET authenticator. By default, values 8314 or 8899 are used.

Command with **no** prefix resets port to default.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(kabinet)> **port** *‹port›*

(kabinet)> **no port**

Arguments

Argument	Value	Description
port	*Integer*	The port number.

Example

```
(kabinet)> port 12345
Kabinet::Authenticator: A server port set.
```

```
(kabinet)> no port
Kabinet::Authenticator: A server port reset.
```

History

Version	Description
2.14	The **kabinet port** command has been introduced.

3.80.5 kabinet protocol-version

Description

Set version of KABiNET authenticator protocol. By default, protocol version 2 is used.

Command with **no** prefix resets protocol to default.

Prefix no

Yes

Change settings Yes

Multiple input No

Synopsis

(kabinet)> **protocol-version** *‹version›*

(kabinet)> **no protocol-version**

Arguments

Argument	Value	Description
version	*String*	Version of protocol.

Example

```
(kabinet)> protocol-version 1
Kabinet::Authenticator: A protocol version set to "1".
```

```
(kabinet)> no protocol-version
Kabinet::Authenticator: A protocol version reset to "2".
```

History

Version	Description
2.02	The **kabinet protocol-version** command has been introduced.

3.80.6 kabinet server

Description Set an IP-address of КАБiNET authentication server. By default, IP `10.0.0.1` is used.

Command with **no** prefix resets the address.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(kabinet)> **server** *‹address›*

(kabinet)> **no server**

Arguments

Argument	Value	Description
address	*IP-address*	Authentication server address.

Example

```
(kabinet)> server 77.222.111.1
Kabinet::Authenticator: A server address set.
```

```
(kabinet)> no server
Kabinet::Authenticator: A server address reset.
```

History

Version	Description
2.02	The **kabinet server** command has been introduced.

3.81 known host

Description Set known host.

Prefix no Yes

Change settings Yes

Multiple input Yes

Synopsis

(config)> **known host** *‹name› ‹mac›*

(config)> **no known host** [*mac*]

Arguments

Argument	Value	Description
name	*String*	Arbitrary host name.
mac	*MAC-address*	MAC-address.

Example

```
(config)> known host MY 00:0e:c6:a2:22:a1
Core::KnownHosts: New host "MY" has been created.
```

```
(config)> no known host 00:0e:c6:a2:22:a1
Core::KnownHosts: Host 00:0e:c6:a1:26:a8 has been removed.
```

History

Version	Description
2.00	The **known host** command has been introduced.

3.82 mws acquire

Description Attach new device to *MWS*.

Command with **no** prefix stops the acquisition.

Prefix no Yes

Change settings No

Multiple input No

Synopsis

(config)> **mws acquire** *‹candidate›* [**eula-accept**] [**dpn-accept**] [**no-update**]

(config)> **no mws acquire** *‹candidate›*

Arguments

Argument	Value	Description
candidate	*String*	Device ID — MAC-address or CID.
eula-accept	*Keyword*	Send **eula accept** command.
dpn-accept	*Keyword*	Send Device Privacy Notice accept.
no-update	*Keyword*	Acquisition without firmware update confirmation.

Example

```
(config)> mws acquire ab1409a2-0f87-11e8-8f23-3d5f5921b253 ►
eula-accept
Mws::Controller: Candidate "ab1409a2-0f87-11e8-8f23-3d5f5921b253" ►
acquire started.
```

```
(config)> mws acquire 7207838e-af7d-11e6-8029-25463bd03811 ►
eula-accept dpn-accept no-update
Mws::Controller: Candidate "7207838e-af7d-11e6-8029-25463bd03811" ►
acquire started.
```

```
(config)> no mws acquire 60:31:97:3f:36:00
Mws::Controller: Candidate "60:31:97:3f:36:00" acquire stopped.
```

History

Version	Description
2.15	The **mws acquire** command has been introduced.

3.83 mws backhaul shutdown

Description

Disable hidden wireless backhaul access points for *MWS* service. By default, the setting is enabled.

Command with **no** prefix enables hidden wireless backhaul access points.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config)> **mws backhaul shutdown**

(config)> **no mws backhaul shutdown**

Example

```
(config)> mws backhaul shutdown
Mws::Controller: Backhaul disabled.
```

```
(config)> no mws backhaul shutdown
Mws::Controller: Backhaul enabled.
```

History

Version	Description
3.04	The **mws backhaul shutdown** command has been introduced.

3.84 mws log stp

Description

Enable STP logging for the interface. Allows you to track sent and received BPDU packets.

Command with **no** prefix disables logging for specified interface. If you use no argument, the entire list of STP logging will be removed.

Prefix no

Yes

Change settings

No

Multiple input

Yes

Synopsis

(config)> **mws log stp** ‹*interface*›

(config)> **no mws log stp** [‹*interface*›]

Arguments

Argument	Value	Description
interface	*Interface name*	Full interface name or an alias. You can see the list of available interfaces with help of **interface** [Tab] command.

Example

```
(config)> mws log stp Bridge0
Network::Interface::Rtx::WifiController: Enabled STP logging for ►
"Bridge0".
```

```
(config)> no mws log stp Bridge0
Network::Interface::Rtx::WifiController: Disabled STP logging ►
for "Bridge0".
```

```
(config)> no mws log stp
Network::Interface::Rtx::WifiController: Disabled all STP logging.
```

History

Version	Description
3.06	The **mws log stp** command has been introduced.

3.85 mws member

Description

Command with **no** prefix removes *MWS* member. If you use no argument, the entire list of members will be cleared.

Prefix no

Yes

Change settings No

Multiple input No

Synopsis

(config)> **no mws member** [*member*]

Arguments

Argument	Value	Description
member	*String*	Device ID — MAC-address or CID.

Example

```
(config)> mws no member 2937a388-0d00-11e7-8029-7119319f930e
Mws::MemberList: Member 2937a388-0d00-11e7-8029-7119319f930e ►
pending factory reset.
```

History

Version	Description
2.15	The **mws member** command has been introduced.

3.86 mws member check-update

Description Initiate an update check for *MWS* member.

Prefix no No

Change settings No

Multiple input No

Synopsis

(config)> **mws member** ‹*member*› **check-update**

Arguments

Argument	Value	Description
member	*String*	Device ID — MAC-address or CID.

Example

```
(config)> mws member ab1409a2-0f87-11e8-8f23-3d5f5921b253 ►
check-update
Mws::MemberList: Member "50:ff:20:08:7a:6a" ►
(ab1409a2-0f87-11e8-8f23-3d5f5921b253) checking for an update.
```

History

Version	Description
2.15	The **mws member check-update** command has been introduced.

3.87 mws member debug

Description Enable *MWS* member debug. By default, setting is disabled.

Command with **no** prefix disables the feature.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config)> **mws member** *‹member›* **debug**

(config)> **no mws member** *‹member›* **debug**

Arguments

Argument	Value	Description
member	*String*	Device ID — MAC-address or CID.

Example

```
(config)> mws member 60:31:97:3c:11:12 debug
Mws::MemberList: Member "60:31:97:3c:11:12" ►
(7207838e-af7d-11e6-8011-25463bd03812) RCI debug enabled.
```

```
(config)> no mws member 60:31:97:3c:11:12 debug
Mws::MemberList: Member "60:31:97:3c:11:12" ►
(7207838e-af7d-11e6-8011-25463bd03812) RCI debug disabled.
```

History

Version	Description
3.05	The **mws member debug** command has been introduced.

3.88 mws member dpn-accept

Description Accept *DPN* for *MWS* member.

Prefix no No

Change settings No

Multiple input No

Synopsis

(config)> **mws member** *‹member›* **dpn-accept**

Arguments

Argument	Value	Description
member	*String*	Device ID — MAC-address or CID.

Example

```
(config)> mws member 7207838e-af7d-11e6-8029-25463bd03828 ►
dpn-accept
Mws::Controller: Candidate "ab1409a2-0f87-11e8-8f23-3d5f5921b253" ►
acquire started.
```

History

Version	Description
3.05	The **mws member dpn-accept** command has been introduced.

3.89 mws revisit

Description Re-read status of potential *MWS* member.

Prefix no Yes

Change settings No

Multiple input No

Synopsis

(config)> **mws revisit** *‹candidate›*

(config)> **no mws revisit** *‹candidate›*

Arguments

Argument	Value	Description
candidate	*String*	Device ID — MAC-address or CID.

Example

```
(config)> mws revisit 50:ff:20:08:71:62
Mws::Controller: Candidate "50:ff:20:08:71:62" revisit started.
```

```
(config)> mws no revisit 50:ff:20:08:71:62
Mws::Controller: Candidate "50:ff:20:08:71:62" revisit stopped.
```

History

Version	Description
2.15	The **mws revisit** command has been introduced.

3.90 mws zone

Description Limit the connection area of the client device within the specified *MWS* members.

Command with **no** prefix removes the specified setting. If you use no arguments, the entire list of restrictions will be removed.

Prefix no Yes

Change settings No

Multiple input Yes

Synopsis

(config)> **mws zone** *‹mac› ‹cid›*

(config)> **no mws zone** [*‹mac› ‹cid›*]

Arguments

Argument	Value	Description
mac	*MAC-address*	MAC-address of client device. It must be listed as a known host.
cid	*CID*	Identifier of *MWS* member.

Example

```
(config)> mws zone 11:22:33:ec:58:e2 ►
12298f60-d886-11e7-9396-176971eeb8d6
Mws::Controller: Added zone 11:22:33:ec:58:e2 ►
12298f60-d886-11e7-9396-176971eeb8d6.
```

```
(config)> no mws zone 11:22:33:ec:58:e2 ►
12298f60-d886-11e7-9396-176971eeb8d6
Mws::Controller: Deleted zone 11:22:33:ec:58:e2 ►
12298f60-d886-11e7-9396-176971eeb8d6.
```

```
(config)> no mws zone
Mws::Controller: Cleared all zones.
```

History

Version	Description
3.06	The **mws zone** command has been introduced.

3.91 ndns

Description Access to a group of commands to manage KeenDNS service.

Prefix no No

Change settings No

Multiple input No

Group entry (ndns)

Synopsis

(config)> **ndns**

Example

```
(config)> ndns
Core::Configurator: Done.
```

History

Version	Description
2.07	The **ndns** command has been introduced.

3.91.1 ndns book-name

Description

Reserve Public DNS device hostname allocation.

For hostname transmission to another Keenetic device `transfer-code` parameter is used.

To transfer hostname it is necessary:

1. Execute command with `transfer-code` on the transmitting side.
2. Execute the same command with the same parameters on the receiving side.

Lifetime of `transfer-code` is 1 week.

Prefix no

No

Change settings

Yes

Multiple input

No

Synopsis

(ndns)> **book-name** *‹name›* *‹domain›* [*‹access›* [**ipv6** *‹access6›*] | *‹transfer-code›*]

Arguments

Argument	Value	Description
`name`	*`String`*	The hostname for allocation.
`domain`	*`String`*	Second-level domain.
`access`	`auto`	Automatic access type.
	`cloud`	Hostname is registered on the cloud server IP-address, HTTP traffic is tunneled to the City.
	`direct`	Hostname is registered on the City WAN-address.
`access6`	`cloud`	Enable cloud mode for IPv6 address.
`transfer-code`	*`Hexadecimal number`*	Code for domain transmission to another Keenetic device. The length is 32 symbols.

Example

```
(ndns)> book-name myhome23 keenetic.pro

             done, layout = view, title = NDSS::ndns/bookName ►
(Public DNS Hostname Booking), sub-title = The name booking was ►
successful.:
               client, geo = RU, ip = 193.0.174.200, format = ►
clean, date = 2019-05-23T09:46:54.536Z, standalone = false:

               fields:
                    field, name = name, title = Public Name:
                    field, name = domain, title = Domain Name:
```

```
                  field, name = updated, title = Updated, type ►
= date, variant = date:
                    field, name = address, title = IP Address:
                    field, name = access, title = Access Mode ►
IP4, default = unknown:
                  field, name = address6, title = IPv6 Address:
                   field, name = access6, title = Access Mode ►
IPv6, default = unknown:
                    field, name = transfer, title = Transfer:

                name: myhome23
              domain: keenetic.pro
                acme: LE
             updated: 2019-05-23T09:46:51.013Z
             address: 193.0.174.200
              access: direct
             access6: none
            transfer: false

              suffix, layout = message, code = 200, message = ►
The name booking was successful.:
                  detail, layout = list:
                     columns:
                          column, id = type, title = Type:

                          column, id = peer, title = Peer:

                         column, id = detail, title = Detail:

                          column, id = elapsed, title = Time, ►
variant = period, scale = 1:

                        item, elapsed = 18, origin = ►
[TaskUdpSingle "ndss111h2.ndm9.xyz" [MsgNdssMessage ►
["ndns/bookPrepare","014635737374513","myhome23","keenetic.pro",undefined]] ►
/ started], type = reply-final,
peer = ndss111h2.ndm9.xyz, detail = [MsgCack]:

                        item, elapsed = 19, origin = ►
[TaskBookName, ►
{"name":"myhome23","domain":"keenetic.pro","license":"014635737374513"}], ►
type = prepare-reply, peer = ndss111h2.ndm9.xyz, detail = success
reply: [MsgCack], quorumLeft=3:

                        item, elapsed = 27, origin = ►
[TaskUdpSingle "ndss112o1.ndm9.xyz" [MsgNdssMessage ►
["ndns/bookPrepare","014635737374513","myhome23","keenetic.pro",undefined]] ►
/ started], type = reply-final,
peer = ndss112o1.ndm9.xyz, detail = [MsgCack]:

                        item, elapsed = 27, origin = ►
[TaskBookName, ►
{"name":"myhome23","domain":"keenetic.pro","license":"014635737374513"}], ►
type = prepare-reply, peer = ndss112o1.ndm9.xyz, detail = success
```

```
reply: [MsgCack], quorumLeft=2:

                          item, elapsed = 67, origin = ►
[TaskUdpSingle "ndss111r3.ndm9.xyz" [MsgNdssMessage ►
["ndns/bookPrepare","014635737374513","myhome23","keenetic.pro",undefined]] ►
/ started], type = reply-final,
peer = ndss111r3.ndm9.xyz, detail = [MsgCack]:

                          item, elapsed = 68, origin = ►
[TaskBookName, ►
{"name":"myhome23","domain":"keenetic.pro","license":"014635737374513"}], ►
type = prepare-reply, peer = ndss111r3.ndm9.xyz, detail = success
reply: [MsgCack], quorumLeft=1:

                          item, elapsed = 70, origin = ►
[TaskUdpSingle "ndss112r3.ndm9.xyz" [MsgNdssMessage ►
["ndns/bookPrepare","014635737374513","myhome23","keenetic.pro",undefined]] ►
/ started], type = reply-final,
peer = ndss112r3.ndm9.xyz, detail = [MsgCack]:

                          item, elapsed = 79, origin = ►
[TaskBookName, ►
{"name":"myhome23","domain":"keenetic.pro","license":"014635737374513"}], ►
type = done, peer = local, detail = finalize: the name allocation
commited.:

                          item, elapsed = 91, origin = ►
[TaskBookName, ►
{"name":"myhome23","domain":"keenetic.pro","license":"014635737374513"}], ►
type = complete, peer = finalizer, detail = address updated:
193.0.174.200:

                          item, elapsed = 91, origin = ►
[TaskBookName, ►
{"name":"myhome23","domain":"keenetic.pro","license":"014635737374513"}], ►
type = finalize, peer = local, detail = post-process triggers
executed.:

                          item, elapsed = 91, origin = ►
[TaskBookName, ►
{"name":"myhome23","domain":"keenetic.pro","license":"014635737374513"}], ►
type = prepare-reply, peer = ndss112r3.ndm9.xyz, detail = success
reply: [MsgCack]:

                          item, elapsed = 97, origin = ►
[TaskUdpSingle "ndss112o1.ndm9.xyz" [MsgNdssMessage ►
["ndns/bookFinalize","014635737374513","myhome23","keenetic.pro","193.0.174.200","::2",undefined,"2019-05-
23T09:46:51.013Z"]] / started], type = reply-final, peer = ►
ndss112o1.ndm9.xyz, detail = [MsgCack]:

                          item, elapsed = 106, origin = ►
[TaskUdpSingle "ndss111h2.ndm9.xyz" [MsgNdssMessage ►
["ndns/bookFinalize","014635737374513","myhome23","keenetic.pro","193.0.174.200","::2",undefined,"2019-05-
23T09:46:51.013Z"]] / started], type = reply-final, peer = ►
```

```
ndss111h2.ndm9.xyz, detail = [MsgCack]:

                          item, elapsed = 153, origin = ►
[TaskUdpSingle "ndss112r3.ndm9.xyz" [MsgNdssMessage ►
["ndns/bookFinalize","014635737374513","myhome23","keenetic.pro","193.0.174.200","::2",undefined,"2019-05-
23T09:46:51.013Z"]] / started], type = reply-final, peer = ►
ndss112r3.ndm9.xyz, detail = [MsgCack]:

                          item, elapsed = 153, origin = ►
[TaskUdpSingle "ndss111r3.ndm9.xyz" [MsgNdssMessage ►
["ndns/bookFinalize","014635737374513","myhome23","keenetic.pro","193.0.174.200","::2",undefined,"2019-05-
23T09:46:51.013Z"]] / started], type = reply-final, peer = ►
ndss111r3.ndm9.xyz, detail = [MsgCack]:

                          item, elapsed = 3465, origin = ►
[TaskUdpSingle "ndss112h2.ndm9.xyz" [MsgNdssMessage ►
["ndns/bookFinalize","014635737374513","myhome23","keenetic.pro","193.0.174.200","::2",undefined,"2019-05-
23T09:46:51.013Z"]] / started], type = reply-final, peer = ►
ndss112h2.ndm9.xyz, detail = [MsgCack]:

                          item, elapsed = 3520, origin = ►
[TaskUdpSingle "ndss112h2.ndm9.xyz" [MsgNdssMessage ►
["ndns/bookPrepare","014635737374513","myhome23","keenetic.pro",undefined]] ►
/ started], type = reply-final,
peer = ndss112h2.ndm9.xyz, detail = [MsgCack]:

                          item, elapsed = 3521, origin = ►
[TaskBookName, ►
{"name":"myhome23","domain":"keenetic.pro","license":"014635737374513"}], ►
type = prepare-reply, peer = ndss112h2.ndm9.xyz, detail = success
reply: [MsgCack]:

                          item, elapsed = 3521, origin = ►
[TaskBookName, ►
{"name":"myhome23","domain":"keenetic.pro","license":"014635737374513"}], ►
type = complete, peer = *, detail = All done.:

Ndns::Client: Booked "myhome23.keenetic.pro".
```

```
(ndns)> book-name nnttnn keenetic.pro ►
121d567f901a345b289c121b567c903c

             done, layout = view, title = NDSS::ndns/bookName ►
(Public DNS Hostname Booking), sub-title =
The name booking was successful.: client, geo = RU, ip = ►
193.0.174.137, format =
clean, date = 2018-12-13T09:04:41.939Z, standalone = false:

               fields:
                    field, name = name, title = Public Name:
                    field, name = domain, title = Domain Name:
                  field, name = updated, title = Updated, type ►
= date, variant = date:
                    field, name = address, title = IP Address:
```

```
                    field, name = access, title = Access Mode ►
IP4, default = unknown:
                  field, name = address6, title = IPv6 Address:
                   field, name = access6, title = Access Mode ►
IPv6, default = unknown:
                    field, name = transfer, title = Transfer:

                 name: nnttnn
               domain: keenetic.pro
                 acme: LE
              updated: 2018-12-13T08:47:11.014Z
              address: 0.0.0.0
               access: cloud
              access6: none
             transfer: true

               suffix, layout = message, code = 200, message = ►
The name booking was successful.:
                   detail, layout = list:
                      columns:
                           column, id = o, title = Operation:

                           column, id = d, title = Detail:

                         column, id = t, title = Time, variant ►
= period, scale = 1:

                          item, hl = false, o = start, d = ►
[TaskBookName, {"name":"nnttnn","domain":
                          ►
"keenetic.pro","license":"730102642155400"}], t = 0:

                         item, hl = false, o = lock-local, d = ►
the name is locked (for current transaction), t = 1:

                          item, hl = false, o = cluster, d = ►
quorumRemaining: 2, quorumPossible: 4, quorumTotal: 4, t = 1:

                         item, hl = false, o = lock-reply, d = ►
Success: prepare, [NDSS
(key=Binary('PuR10V/kVezuoVCE'), alt=Binary('0gJ/Wh1606jlAm1M'), ►
dst="/192.168.21.14:17047")], [MsgCack], quorumLeft=2, t = 10:

                         item, hl = false, o = lock-reply, d = ►
Success: prepare, [NDSS
(key=Binary('EbxdTB4ne4ef/+p/'), alt=Binary('1c+3/pP6zaUjuE5w'), ►
dst="/88.198.177.100:17047")], [MsgCack], quorumLeft=1, t = 57:

                         item, hl = false, o = lock-reply, d = ►
Quorum reached, finalizing, t = 57:

                          item, hl = false, o = finalize, d = ►
local changes commited., t = 65:
```

```
                          item, hl = false, o = refreshed, d = ►
address updated: 0.0.0.0, t = 77:

                          item, hl = false, o = finalize, d = ►
post-process triggers executed., t = 77:

                         item, hl = false, o = lock-reply, d = ►
Success: prepare, [NDSS
(key=Binary('+sSJ5Oow6hn05f6n'), alt=Binary('7FsVtTpEppYeP7aj'),
dst="/46.105.148.85:17047")], [MsgCack], quorumLeft=0, t = 78:

                         item, hl = false, o = lock-reply, d = ►
Success: prepare, [NDSS
(key=Binary('KveTxYekUYk2BwXz'), alt=Binary('s1OR6mJvMmfQSeOs'),
dst="/88.198.177.100:16047")], [MsgCack], quorumLeft=0, t = 78:

                         item, hl = false, o = lock-reply, d = ►
Done, all replies collected., t = 79:

                         item, hl = false, o = commit-reply, d ►
= Success: finalize, [NDSS
(key=Binary('PuR10V/kVezuoVCE'), alt=Binary('0gJ/Wh1606jlAm1M'),
dst="/192.168.21.14:17047")], [MsgCack], t = 84:

                         item, hl = false, o = commit-reply, d ►
= Success: finalize, [NDSS
(key=Binary('EbxdTB4ne4ef/+p/'), alt=Binary('1c+3/pP6zaUjuE5w'), ►
dst="/88.198.177.100:17047")], [MsgCack], t = 126:

                         item, hl = false, o = commit-reply, d ►
= Success: finalize, [NDSS
(key=Binary('+sSJ5Oow6hn05f6n'), alt=Binary('7FsVtTpEppYeP7aj'), ►
dst="/46.105.148.85:17047")], [MsgCack], t = 133:

                         item, hl = false, o = commit-reply, d ►
= Success: finalize, [NDSS
key=Binary('KveTxYekUYk2BwXz'), alt=Binary('s1OR6mJvMmfQSeOs'), ►
dst="/88.198.177.100:16047")], [MsgCack], t = 145:

                         item, hl = false, o = commit-reply, d ►
= Commit stage complete., t = 146:

                        item, hl = false, o = complete, d = All ►
done., t = 146:

Ndns::Client: Booked "nnttnn.keenetic.pro".
```

```
(ndns)> book-name myhome23 keenetic.pro cloud ipv6 cloud

             done, layout = view, title = NDSS::ndns/bookName ►
(Public DNS Hostname Booking), sub-title = The name booking was ►
successful.:
               client, geo = RU, ip = 193.0.174.200, format = ►
clean, date = 2019-05-23T09:12:29.145Z, standalone = false:
```

```
               fields:
                    field, name = name, title = Public Name:
                    field, name = domain, title = Domain Name:
                  field, name = updated, title = Updated, type ►
= date, variant = date:
                    field, name = address, title = IP Address:
                    field, name = access, title = Access Mode ►
IP4, default = unknown:
                  field, name = address6, title = IPv6 Address:
                   field, name = access6, title = Access Mode ►
IPv6, default = unknown:
                    field, name = transfer, title = Transfer:

                 name: myhome23
               domain: keenetic.pro
                 acme: LE
              updated: 2019-05-23T09:12:16.197Z
              address: 0.0.0.0
               access: cloud
             address6: ::
              access6: cloud
             transfer: false

              suffix, layout = message, code = 200, message = ►
The name booking was successful.:
                  detail, layout = list:
                     columns:
                          column, id = type, title = Type:

                          column, id = peer, title = Peer:

                         column, id = detail, title = Detail:

                          column, id = elapsed, title = Time, ►
variant = period, scale = 1:

                        item, elapsed = 11, origin = ►
[TaskUdpSingle "ndss112h2.ndm9.xyz" [MsgNdssMessage ►
["ndns/bookPrepare","014635737374513","myhome23","keenetic.pro",undefined]] ►
/ started], type = reply-final,
peer = ndss112h2.ndm9.xyz, detail = [MsgCack]:

                        item, elapsed = 11, origin = ►
[TaskBookName, ►
{"name":"myhome23","domain":"keenetic.pro","license":"014635737374513"}], ►
type = prepare-reply, peer = ndss112h2.ndm9.xyz, detail = success
reply: [MsgCack], quorumLeft=3:

                        item, elapsed = 17, origin = ►
[TaskUdpSingle "ndss112o1.ndm9.xyz" [MsgNdssMessage ►
["ndns/bookPrepare","014635737374513","myhome23","keenetic.pro",undefined]] ►
/ started], type = reply-final,
peer = ndss112o1.ndm9.xyz, detail = [MsgCack]:
```

```
                          item, elapsed = 18, origin = ►
[TaskBookName, ►
{"name":"myhome23","domain":"keenetic.pro","license":"014635737374513"}], ►
type = prepare-reply, peer = ndss112o1.ndm9.xyz, detail = success
reply: [MsgCack], quorumLeft=2:

                          item, elapsed = 18, origin = ►
[TaskUdpSingle "ndss111o1.ndm9.xyz" [MsgNdssMessage ►
["ndns/bookPrepare","014635737374513","myhome23","keenetic.pro",undefined]] ►
/ started], type = reply-final,
peer = ndss111o1.ndm9.xyz, detail = [MsgCack]:

                          item, elapsed = 19, origin = ►
[TaskBookName, ►
{"name":"myhome23","domain":"keenetic.pro","license":"014635737374513"}], ►
type = prepare-reply, peer = ndss111o1.ndm9.xyz, detail = success
reply: [MsgCack], quorumLeft=1:

                          item, elapsed = 25, origin = ►
[TaskBookName, ►
{"name":"myhome23","domain":"keenetic.pro","license":"014635737374513"}], ►
type = done, peer = local, detail = finalize: the name allocation
commited.:

                          item, elapsed = 40, origin = ►
[TaskBookName, ►
{"name":"myhome23","domain":"keenetic.pro","license":"014635737374513"}], ►
type = complete, peer = finalizer, detail = address updated: ►
0.0.0.0:

                          item, elapsed = 40, origin = ►
[TaskBookName, ►
{"name":"myhome23","domain":"keenetic.pro","license":"014635737374513"}], ►
type = finalize, peer = local, detail = post-process triggers
executed.:

                          item, elapsed = 49, origin = ►
[TaskUdpSingle "ndss112o1.ndm9.xyz" [MsgNdssMessage ►
["ndns/bookFinalize","014635737374513","myhome23","keenetic.pro","0.0.0.0","::",undefined,"2019-05-
23T09:12:28.977Z"]] / started], type = reply-final, peer = ►
ndss112o1.ndm9.xyz, detail = [MsgCack]:

                          item, elapsed = 49, origin = ►
[TaskUdpSingle "ndss111o1.ndm9.xyz" [MsgNdssMessage ►
["ndns/bookFinalize","014635737374513","myhome23","keenetic.pro","0.0.0.0","::",undefined,"2019-05-
23T09:12:28.977Z"]] / started], type = reply-final, peer = ►
ndss111o1.ndm9.xyz, detail = [MsgCack]:

                          item, elapsed = 50, origin = ►
[TaskUdpSingle "ndss111r3.ndm9.xyz" [MsgNdssMessage ►
["ndns/bookPrepare","014635737374513","myhome23","keenetic.pro",undefined]] ►
/ started], type = reply-final,
peer = ndss111r3.ndm9.xyz, detail = [MsgCack]:
```

```
                          item, elapsed = 50, origin = ►
[TaskBookName, ►
{"name":"myhome23","domain":"keenetic.pro","license":"014635737374513"}], ►
type = prepare-reply, peer = ndss111r3.ndm9.xyz, detail = success
reply: [MsgCack]:

                          item, elapsed = 50, origin = ►
[TaskUdpSingle "ndss112r3.ndm9.xyz" [MsgNdssMessage ►
["ndns/bookPrepare","014635737374513","myhome23","keenetic.pro",undefined]] ►
/ started], type = reply-final,
peer = ndss112r3.ndm9.xyz, detail = [MsgCack]:

                          item, elapsed = 51, origin = ►
[TaskBookName, ►
{"name":"myhome23","domain":"keenetic.pro","license":"014635737374513"}], ►
type = prepare-reply, peer = ndss112r3.ndm9.xyz, detail = success
reply: [MsgCack]:

                          item, elapsed = 80, origin = ►
[TaskUdpSingle "ndss112r3.ndm9.xyz" [MsgNdssMessage ►
["ndns/bookFinalize","014635737374513","myhome23","keenetic.pro","0.0.0.0","::",undefined,"2019-05-
23T09:12:28.977Z"]] / started], type = reply-final, peer = ►
ndss112r3.ndm9.xyz, detail = [MsgCack]:

                          item, elapsed = 122, origin = ►
[TaskUdpSingle "ndss112h2.ndm9.xyz" [MsgNdssMessage ►
["ndns/bookFinalize","014635737374513","myhome23","keenetic.pro","0.0.0.0","::",undefined,"2019-05-
23T09:12:28.977Z"]] / started], type = reply-final, peer = ►
ndss112h2.ndm9.xyz, detail = [MsgCack]:

                          item, elapsed = 165, origin = ►
[TaskUdpSingle "ndss111r3.ndm9.xyz" [MsgNdssMessage ►
["ndns/bookFinalize","014635737374513","myhome23","keenetic.pro","0.0.0.0","::",undefined,"2019-05-
23T09:12:28.977Z"]] / started], type = reply-final, peer = ►
ndss111r3.ndm9.xyz, detail = [MsgCack]:

                          item, elapsed = 166, origin = ►
[TaskBookName, ►
{"name":"myhome23","domain":"keenetic.pro","license":"014635737374513"}], ►
type = complete, peer = *, detail = All done.:

Ndns::Client: Booked "myhome23.keenetic.pro".
```

History

Version	Description
2.07	The **ndns book-name** command has been introduced.
2.14	Parameter `ipv6` was added.

3.91.2 ndns check-name

Description Check the availability of hostname for allocation.

Prefix no No

Change settings No

Multiple input No

Synopsis

(ndns)> **check-name** *‹name›*

Arguments

Argument	Value	Description
name	*String*	The hostname for allocation.

Example

```
(ndns)> check-name testname

             list:
                 item:
                   domain: keenetic.link
                     name: testname
                available: yes
                     acme: yes

                 item:
                   domain: keenetic.name
                     name: testname
                available: yes
                     acme: yes

                 item:
                   domain: keenetic.pro
                     name: testname
                available: no
                     acme: yes

Ndns::Client: Check completed.
```

History

Version	Description
2.07	The **ndns check-name** command has been introduced.

3.91.3 ndns drop-name

Description Drop Public DNS device hostname allocation.

Prefix no No

Change settings Yes

Multiple input No

Synopsis

(ndns)> **drop-name** *‹name› ‹domain›*

Arguments

Argument	Value	Description
name	*String*	The hostname for dropping.
domain	*String*	Second-level domain.

Example

```
(ndns)> drop-name testname mykeenetic.net

             done, title = NDSS::ndns/dropName (Delete DNS ►
Hostname Booking), code = 200,
icon = tick, hl = true, layout = message:
               client, geo = RU, ip = 81.200.27.56, format = ►
clean, date = 2016-09-
22T10:52:35.685Z, standalone = false:
               reason: The name is un-booked.

               detail, layout = list:
                  columns:
                       column, id = o, title = Operation:
                       column, id = d, title = Detail:
                      column, id = t, title = Time, variant = ►
period, scale = 1:

                        item, hl = false, o = start, d = ►
[TaskDropName, {"name":"testname",
"domain":"mykeenetic.net","license":"243992935221479"}], t = 0:
                     item, hl = false, o = lock-local, d = the ►
name is locked (for current
transaction), t = 1:
                        item, hl = false, o = cluster, d = ►
quorumRemaining: 2, quorumPossible: 4,
quorumTotal: 4, t = 1:
                        item, hl = false, o = lock-reply, d = ►
Success: prepare, [NDSS
(key=Binary('vNEqUcIAWtrIaC5O'), alt=Binary('L2hVqanJmGJrzvKh'),
dst="/148.251.63.154:17047")], [MsgCack], quorumLeft=2, t = 55:
                        item, hl = false, o = lock-reply, d = ►
Success: prepare, [NDSS
(key=Binary('yp/ghaehxe5EtXyc'), alt=Binary('t+JluEWuGguJ+28h'),
dst="/46.105.148.81:17047")], [MsgCack], quorumLeft=1, t = 72:
                     item, hl = false, o = lock-reply, d = Quorum ►
reached, finalizing, t = 73:
                       item, hl = false, o = finalize, d = local ►
changes commited., t = 79:
                     item, hl = false, o = refreshed, d = address ►
cleared, t = 85:
                        item, hl = false, o = finalize, d = ►
post-process triggers executed., t = 85:
                        item, hl = false, o = commit-reply, d = ►
```

```
Success: finalize, [NDSS
(key=Binary('vNEqUcIAWtrIaC5O'), alt=Binary('L2hVqanJmGJrzvKh'),
dst="/148.251.63.154:17047")], [MsgCack], t = 134:
                    item, hl = false, o = commit-reply, d = ►
Success: finalize, [NDSS
(key=Binary('yp/ghaehxe5EtXyc'), alt=Binary('t+JluEWuGguJ+28h'),
dst="/46.105.148.81:17047")], [MsgCack], t = 161:
                    item, hl = false, o = lock-reply, d = ►
Success: prepare, [NDSS
(key=Binary('SyptNue2bys/mxiO'), alt=Binary('yPrQwfa/4yn676wk'),
dst="/148.251.129.152:17047")], [MsgCack], quorumLeft=0, t = 231:
                    item, hl = false, o = commit-reply, d = ►
Success: finalize, [NDSS
(key=Binary('SyptNue2bys/mxiO'), alt=Binary('yPrQwfa/4yn676wk'),
dst="/148.251.129.152:17047")], [MsgCack], t = 235:
                    item, hl = false, o = commit-reply, d = ►
Success: finalize, [NDSS
(key=Binary('pLNIsTXD+OP4D9Fc'), alt=Binary('kGImY2U/LublZ/Zr'),
dst="/91.218.112.118:17047")], [MsgCack], t = 3608:
                    item, hl = false, o = commit-reply, d = ►
Commit stage complete., t = 3608:
                    item, hl = false, o = complete, d = All ►
done., t = 3608:

Ndns::Client: Dropped "testname.mykeenetic.net".
```

History

Version	Description
2.07	The **ndns drop-name** command has been introduced.

3.91.4 ndns get-booked

Description Get actual info from the server about current booked Public DNS hostname.

Prefix no No

Change settings No

Multiple input No

Synopsis

(ndns)> **get-booked**

Example

```
(ndns)> get-booked

             done, layout = view, title = ►
NDSS::ndns/updateBooking (Update Name Booking
Address and Expiration):
               client, geo = RU, ip = 41.189.34.56, format = ►
xml, date = 2017-09-
14T08:30:19.266Z, standalone = false:
                    menu, src = ►
```

```
/index?__auth=force&__role=context-
menu&ref=%2fndns%2fupdateBooking:

               fields:
                    field, name = name, title = Public Name:

                    field, name = domain, title = Domain Name:

                    field, name = address, title = IP Address:

                  field, name = updated, title = Updated, type ►
= date, variant = date:

                   field, name = access, title = Access Mode, ►
default = unknown:

                    field, name = transfer, title = Transfer:

                 name: testname
               domain: mykeenetic.com
              address: 41.189.34.56
              updated: 2017-09-11T11:27:32.167Z
               access: direct
             transfer: false


Ndns::Client: Get-booked completed.
```

History

Version	Description
2.08	The **ndns get-booked** command has been introduced.

3.91.5 ndns get-update

Description Update Public DNS device hostname allocation on the server.

Prefix no No

Change settings No

Multiple input No

Synopsis

(ndns)> **get-update** [‹*access*› [**ipv6** ‹*access6*›]]

Arguments

Argument	Value	Description
access	auto	Automatic access type.
	cloud	Hostname is registered on the cloud server IP-address, HTTP traffic is tunneled to the City.

Argument	Value	Description
	direct	Hostname is registered on the City WAN-address. This command allows to enable support for the *Static NAT (NAT 1-1)* on the server side in the KeenDNS account parameters.
access6	cloud	Enable cloud mode for IPv6 address.

Example

```
(ndns)> get-update auto

             done, layout = view, title = ►
NDSS::ndns/updateBooking (Update Name Booking
Address and Expiration):
               client, geo = RU, ip = 81.200.27.56, format = ►
xml, date = 2016-09-
22T12:07:32.746Z, standalone = false:
                     menu, src = ►
/index?__auth=force&__role=context-
menu&ref=%2fndns%2fupdateBooking:

               fields:
                    field, name = name, title = Public Name:
                    field, name = domain, title = Domain Name:
                    field, name = address, title = IP Address:
                  field, name = updated, title = Updated, type ►
= date, variant = date:
                   field, name = access, title = Access Mode, ►
default = unknown:
                    field, name = transfer, title = Transfer:

                 name: testname
               domain: mykeenetic.net
              address: 81.200.27.56
              updated: 2016-09-22T12:07:32.744Z
               access: direct
             transfer: false

Ndns::Client: Get-update completed.
```

```
(ndns)> get-update cloud ipv6 cloud

             done, layout = view, title = ►
NDSS::ndns/updateBooking (Update Name Booking Address and ►
Expiration):
               client, geo = RU, ip = 193.0.174.168, format = ►
xml, date = 2019-05-21T15:26:45.552Z, standalone = false:
                     menu, src = ►
/index?__auth=force&__role=context-menu&ref=%2fndns%2fupdateBooking:

               fields:
                    field, name = name, title = Public Name:
                    field, name = domain, title = Domain Name:
```

```
                  field, name = updated, title = Updated, type ►
= date, variant = date:
                    field, name = address, title = IP Address:
                    field, name = access, title = Access Mode ►
(ip4), default = unknown:
                  field, name = address6, title = IPv6 Address:
                   field, name = access6, title = Access Mode ►
(ipv6), default = unknown:
                    field, name = transfer, title = Transfer:

                name: mytest
              domain: keenetic.pro
                acme: LE
             address: 0.0.0.0
              access: cloud
            address6: ::
             access6: cloud
             updated: 2019-05-21T15:26:45.547Z
            transfer: false

Ndns::Client: Get-update completed.
```

```
(ndns)> get-update direct

             done, layout = view, title = ►
NDSS::ndns/updateBooking (Update Name Booking Address and ►
Expiration):
               client, geo = RU, ip = 193.0.174.159, format = ►
xml, date = 2019-11-13T16:53:30.782Z, standalone = false:
                    menu, src = ►
/index?__auth=force&__role=context-menu&ref=%2fndns%2fupdateBooking:

               fields:
                    field, name = name, title = Public Name:
                    field, name = domain, title = Domain Name:
                  field, name = updated, title = Updated, type ►
= date, variant = date:
                    field, name = address, title = IP Address:
                    field, name = access, title = Access Mode ►
(ip4), default = unknown:
                  field, name = address6, title = IPv6 Address:
                   field, name = access6, title = Access Mode ►
(ipv6), default = unknown:
                    field, name = transfer, title = Transfer:

                name: myworknow
              domain: keenetic.link
                acme: LE
             address: 193.0.174.159
              access: direct
             access6: none
             updated: 2019-11-13T16:50:34.298Z
            transfer: false
```

History

Version	Description
2.07	The **ndns get-update** command has been introduced.
2.14	Parameter `ipv6` was added.

3.92 ntce

Description Access to a group of commands to configure the *NTCE* service.

Prefix no No

Change settings No

Multiple input No

Group entry `(config-ntce)`

Synopsis

(config)> **`ntce`**

Example

```
(config)> ntce
(config-ntce)>
```

History

Version	Description
3.07	The **ntce** command has been introduced.

3.92.1 ntce debug

Description Enable debug for the *NTCE* service. By default, setting is disabled.

Command with **no** prefix disables the feature.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config-ntce)> **`debug`**

(config-ntce)> **`no debug`**

Example

```
(config-ntce)> debug
Ntce::Manager: Enabled debug.
```

```
(config-ntce)> no debug
Ntce::Manager: Disabled debug.
```

History

Version	Description
3.07	The **ntce debug** command has been introduced.

3.92.2 ntce qos enable

Description

Enable IntelliQoS, which ensures inbound, and outbound bandwidth for prioritized applications and tasks via pre-defined category groups presets. By default the service is disabled.

Command with **no** prefix disables the feature.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config-ntce)> **qos enable**

(config-ntce)> **no qos enable**

Example

```
(config-ntce)> qos enable
Ntce::Manager: Enabled QoS.
```

```
(config-ntce)> no qos enable
Ntce::Manager: Disabled QoS.
```

History

Version	Description
3.07	The **ntce qos enable** command has been introduced.

3.92.3 ntce qos priority

Description

Set priorities for traffic categories.

Command with **no** prefix removes the setting.

Prefix no

Yes

Change settings

Yes

Multiple input

Yes

Synopsis

(config-ntce)> **qos priority** *‹category› ‹priority›*

(config-ntce)> **no qos priority**

Arguments

Argument	Value	Description
`category`	`calling`	① Minimum latency.

Argument	Value	Description
	`gaming`	② Real time interactive.
	`streaming`	③ Broadcast services.
	`work`	④ Low latency.
	`surfing`	⑤ High-throughput data.
	`filetransfering`	⑥ Low priority data.
`priority`	*`Integer`*	Priority value. Can take values from 1 to 6.

Example

```
(config-ntce)> qos priority calling 1
Ntce::Manager: Set priority "1" to "calling".
```

```
(config-ntce)> no qos priority
Ntce::Manager: Reset QoS priority list.
```

History

Version	Description
3.07	The **ntce qos priority** command has been introduced.

3.93 ntp

Description

Access to configure *NTP*-client.

Command with **no** prefix resets *NTP*-client configuration to default.

Prefix no

Yes

Change settings

No

Multiple input

No

Synopsis

(config)> **no ntp**

Example

```
(config)> no ntp
Ntp::Client: Configuration reset.
```

History

Version	Description
2.00	The **ntp** command has been introduced.

3.93.1 ntp server

Description

Add a new *NTP*-server to the list. You can enter up to 8 *NTP*-servers.

Command with **no** prefix deletes *NTP*-server from the list. If you use no argument, the entire list of *NTP*-servers will be removed.

Prefix no Yes

Change settings Yes

Multiple input Yes

Synopsis

(config)> **ntp server** ‹*server*›

(config)> **no ntp server** [‹*server*›]

Arguments

Argument	Value	Description
server	*String*	Host of *NTP*-server.

Example

```
(config)> ntp server pool.ntp.org
Ntp::Client: Server "pool.ntp.org" has been added.
```

```
(config)> no ntp server
Ntp::Client: All NTP servers removed.
```

History

Version	Description
2.00	The **ntp server** command has been introduced.

3.93.2 ntp sync-period

Description Set a period for time synchronization. By default, 1 week is used.

Command with **no** prefix resets time synchronization to default.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config)> **ntp sync-period** ‹*period*›

(config)> **no ntp sync-period**

Arguments

Argument	Value	Description
period	*Integer*	Time synchronization, in minutes. Can take values from 60 minutes to 1 month.

Example

```
(config)> ntp sync-period 60
Ntp::Client: A synchronization period set to 60 minutes.
```

```
(config)> no ntp sync-period
Ntp::Client: Synchronization period value reset.
```

History

Version	Description
2.00	The **ntp sync-period** command has been introduced.

3.94 ntp server

Description

Add a new *NTP*-server to the list. You can enter up to 8 *NTP*-servers.

Command with **no** prefix deletes *NTP*-server from the list. If you use no argument, the entire list of *NTP*-servers will be removed.

Prefix no

Yes

Change settings

Yes

Multiple input

Yes

Synopsis

(config)> **ntp server** *‹server›*

(config)> **no ntp server** [*‹server›*]

Arguments

Argument	Value	Description
server	*String*	Host of *NTP*-server.

Example

```
(config)> ntp server pool.ntp.org
Ntp::Client: Server "pool.ntp.org" has been added.
```

```
(config)> no ntp server
Ntp::Client: All NTP servers removed.
```

History

Version	Description
2.00	The **ntp server** command has been introduced.

3.95 ntp sync-period

Description

Set a period for time synchronization. By default, 1 week is used.

Command with **no** prefix resets time synchronization to default.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config)> **ntp sync-period** *‹period›*

(config)> **no ntp sync-period**

Arguments

Argument	Value	Description
period	*Integer*	Time synchronization, in minutes. Can take values from 60 minutes to 1 month.

Example

```
(config)> ntp sync-period 60
Ntp::Client: A synchronization period set to 60 minutes.
```

```
(config)> no ntp sync-period
Ntp::Client: Synchronization period value reset.
```

History

Version	Description
2.00	The **ntp sync-period** command has been introduced.

3.96 ping-check profile

Description

Access to a group of commands to configure *Ping Check* profile. If the profile is not found, the command tries to create it.

Command with **no** prefix removes *Ping Check* profile.

Prefix no

Yes

Change settings

Yes

Multiple input

Yes

Group entry

(config-pchk)

Synopsis

(config)> **ping-check profile** *‹name›*

(config)> **no ping-check profile** *‹name›*

Arguments

Argument	Value	Description
name	*String*	*Ping Check* profile name. You can see the list of available profiles with help of **ping-check profile** [Tab] command.

Example

```
(config)> ping-check profile [Tab]

 Usage template:
          profile {name}

   Choose:
             TEST
             MYMY
```

```
(config)> ping-check profile new_prof
PingCheck::Client: Profile "new_prof" has been created.
(config-pchk)>
```

```
(config)> no ping-check profile new_prof
PingCheck::Client: Profile "new_prof" has been deleted.
```

History

Version	Description
2.04	The **ping-check profile** command has been introduced.

3.96.1 ping-check profile host

Description

Assign hostname for testing. By default, hostname is assigned according to country code.

Command with **no** prefix removes the hostname.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config-pchk)> **host** *‹host›*

(config-pchk)> **no host** [*‹host›*]

Arguments

Argument	Value	Description
host	*Hostname*	Name or address of remote host.

Example

```
(config-pchk)> host 8.8.8.8
PingCheck::Profile: "test": add host "8.8.8.8" for testing.
```

```
(config-pchk)> host google.com
PingCheck::Profile: "test": add host "google.com" for testing.
```

```
(config-pchk)> no host
PingCheck::Profile: "test": hosts cleared.
```

History

Version	Description
2.04	The **ping-check profile host** command has been introduced.

3.96.2 ping-check profile max-fails

Description Specify the number of consecutive failed requests to a remote host by obtaining of which the Internet at the interface considered absent. By default, value 5 is used.

Command with **no** prefix resets to default.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config-pchk)> **max-fails** *‹count›*

(config-pchk)> **no max-fails**

Arguments

Argument	Value	Description
count	*Integer*	Amount of failed requests. Can take values from 1 to 10 inclusively.

Example

```
(config-pchk)> max-fails 7
PingCheck::Profile: "test": uses 7 fail count for disabling ►
interface.
```

```
(config-pchk)> no max-fails
PingCheck::Profile: "test": fail count is reset to 5.
```

History

Version	Description
2.04	The **ping-check profile max-fails** command has been introduced.

3.96.3 ping-check profile min-success

Description Specify the number of consecutive success requests to a remote host by obtaining of which the Internet at the interface considered present. By default, value 5 is used.

Command with **no** prefix resets to default.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config-pchk)> **min-success** *‹count›*

(config-pchk)> **no min-success**

Arguments

Argument	Value	Description
count	*Integer*	Amount of success requests. Can take values from 1 to 10 inclusively.

Example

```
(config-pchk)> min-success 3
PingCheck::Profile: "test": uses 3 success count for enabling ►
interface.
```

```
(config-pchk)> no min-success
PingCheck::Profile: "test": success count is reset to 5.
```

History

Version	Description
2.04	The **ping-check profile min-success** command has been introduced.

3.96.4 ping-check profile mode

Description Set *Ping Check* mode. By default, icmp value is used.

Prefix no No

Change settings Yes

Multiple input No

Synopsis

(config-pchk)> **mode** ‹*mode*›

Arguments

Argument	Value	Description
mode	icmp	The availability testing of remote host will be done by ICMP-echo request (ping) sending.
	connect	The availability testing of remote host will be done by TCP-connection establishing to specified port.

Example

```
(config-pchk)> mode connect
PingCheck::Profile: "TEST": uses connect mode.
```

History

Version	Description
2.04	The **ping-check profile mode** command has been introduced.

3.96.5 ping-check profile port

Description

Specify port for connection to the remote host. Setting has a meaning for `connect` mode of *Ping Check* (see **ping-check profile mode** command).

Command with **no** prefix removes the setting.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config-pchk)> **port** *‹port›*

(config-pchk)> **no port**

Arguments

Argument	Value	Description
`port`	*Integer*	Port number. Can take values from 1 to 65534 inclusively.

Example

```
(config-pchk)> port 80
PingCheck::Profile: "test": uses port 80 for testing.
```

```
(config-pchk)> no port
PingCheck::Profile: "test": port is cleared.
```

History

Version	Description
2.04	The **ping-check profile port** command has been introduced.

3.96.6 ping-check profile power-cycle

Description

Enable power-cycle for USB network interface. Enabled by default.

Command with **no** prefix disables the feature.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config-pchk)> **power-cycle**

(config-pchk)> **no power-cycle**

Example

```
(config-pchk)> power-cycle
PingCheck::Profile: "test": enabled USB power cycle.
```

```
(config-pchk)> power-cycle
PingCheck::Profile: "test": disabled USB power cycle.
```

History

Version	Description
2.04	The **ping-check profile power-cycle** command has been introduced.

3.96.7 ping-check profile timeout

Description

Set the maximum response time of the remote host for a single request in seconds. By default, 2 value is used.

Command with **no** prefix resets setting to default.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config-pchk)> **timeout** ‹*timeout*›

(config-pchk)> **no timeout**

Arguments

Argument	Value	Description
`timeout`	*Integer*	Response time in seconds. Can take values from 1 to 10 inclusively.

Example

```
(config-pchk)> timeout 4
PingCheck::Profile: "test": timeout is changed to 4 seconds.
```

```
(config-pchk)> no timeout
PingCheck::Profile: "test": timeout is reset to 2.
```

History

Version	Description
2.04	The **ping-check profile timeout** command has been introduced.

3.96.8 ping-check profile update-interval

Description

Set periodicity of *Ping Check* performing.

Prefix no

No

Change settings

Yes

Multiple input

No

Synopsis

(config-pchk)> **update-interval** *‹seconds›*

Arguments

Argument	Value	Description
seconds	*Integer*	Refresh period in seconds. Can take values from 3 to 3600 inclusively.

Example

```
(config-pchk)> update-interval 60
PingCheck::Profile: "test": update interval is changed to 60 ►
seconds.
```

History

Version	Description
2.04	The **ping-check profile update-interval** command has been introduced.

3.97 ppe

Description

Enable Packet Processing Engine. By default, the setting is turned on.

Command with **no** prefix disables specified accelerator.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config)> **ppe** *‹engine›*

(config)> **no ppe** [*‹engine›*]

Arguments

Argument	Value	Description
engine	software	Software accelerator.

Example

```
(config)> ppe software
Network::Interface::Rtx::Ppe: Software PPE enabled.
```

```
(config)> no ppe
Network::Interface::Rtx::Ppe: All PPE disabled.
```

History

Version	Description
2.00	The **ppe** command has been introduced.
2.05	Argument engine was implemented.

3.98 pppoe pass

Description Enable PPPoE Pass Through function. You can enter up to 10 network nodes.

Command with **no** prefix disables the function.

Prefix no Yes

Change settings Yes

Multiple input No

Interface type Ethernet

Synopsis

(config)> **pppoe pass through** *‹wan-iface›* *‹lan-iface›*

(config)> **no pppoe pass through**

Arguments

Argument	Value	Description
`wan-iface`	*`Interface name`*	The starting interface — full WAN-interface name or an alias.
`lan-iface`	*`Interface name`*	The finishing interface — full LAN-interface name or an alias.

Example

```
(config)> pppoe pass through Home ISP
Pppoe::Pass: Configured pass from "Bridge0" to "GigabitEthernet1".
```

```
(config)> no pppoe pass
Pppoe::Pass: Disabled.
```

History

Version	Description
2.00	The **pppoe pass** command has been introduced.

3.99 schedule

Description Access to a group of commands to configure the schedule. If the schedule is not found, the command tries to create it.

Command with **no** prefix deletes the schedule.

Prefix no Yes

Change settings Yes

Multiple input Yes

Group entry `(config-sched)`

Synopsis

(config)> **schedule** *‹name›*

(config)> **no schedule** *‹name›*

Arguments

Argument	Value	Description
name	*String*	A schedule name.

History

Version	Description
2.06	The **schedule** command has been introduced.

3.99.1 schedule action

Description Specify the actions to be performed according to the selected schedule.

Command with **no** prefix cancels the action.

Prefix no Yes

Change settings Yes

Multiple input Yes

Synopsis

(config-sched)> **action** *‹action› ‹min› ‹hour› ‹dow›*

(config-sched)> **no action** [*‹action› ‹min› ‹hour› ‹dow›*]

Arguments

Argument	Value	Description
action	start	Action of the beginning.
	stop	Action of the end.
min	*Integer*	The minutes.
hour	*Integer*	The hours.
dow	*Integer*	Days of the week, separated by commas. 0 and 7 mean Sunday. * means daily.

Example

```
(config-sched)> action start 0 9 1,2,3,4,5
Core::Schedule::Manager: Updated schedule "WIFI".
```

History

Version	Description
2.06	The **schedule action** command has been introduced.

3.99.2 schedule description

Description Set description for the selected schedule.

Command with **no** prefix deletes the description.

Prefix no Yes

Change settings No

Multiple input No

Synopsis

(config-sched)> **description** *‹description›*

(config-sched)> **no description**

Arguments

Argument	Value	Description
description	*String*	Text of the description.

Example

```
(config-sched)> description "Schedule for on/off Access Point"
Core::Schedule::Manager: Updated description of schedule "WIFI".
```

History

Version	Description
2.06	The **schedule description** command has been introduced.

3.99.3 schedule led

Description Set LED indication for the scheduled events. SelectedSchedule control shold be chosen with **system led** command.

Command with **no** prefix removes LED indication.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config-sched)> **led** *‹action›*

(config-sched)> **no led**

Arguments

Argument	Value	Description
action	start	LED shows the beginning of the scheduled event.
	stop	LED shows the end of the scheduled event.

Example

```
(config-sched)> led start
Core::Schedule::Led: Selected schedule "111".
```

History

Version	Description
2.08	The **schedule led** command has been introduced.

3.100 service dhcp

Description

Enable *DHCP-server*. If there is not enough settings to start the service (see **ip dhcp pool**), the service will not respond to the network. As soon as there are enough settings, the service will be enabled automatically.

Command with **no** prefix stops the service.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config)> **service dhcp**

(config)> **no service dhcp**

Example

```
(config)> service dhcp
service enabled.
```

History

Version	Description
2.00	The **service dhcp** command has been introduced.

3.101 service dhcp-relay

Description

Enable DHCP-relay. If there are not enough settings to start the service (see **ip dhcp relay lan**, **ip dhcp relay server**, **ip dhcp relay wan**), it will not respond within the network. As soon as there are enough settings, the service will be enabled automatically.

Command with **no** prefix stops the service.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config)> **service dhcp-relay**

(config)> **no service dhcp-relay**

Example

```
(config)> service dhcp-relay
service enabled.
```

History

Version	Description
2.00	The **service dhcp-relay** command has been introduced.

3.102 service dns-proxy

Description Enable DNS-proxy. To configure the parameters of the service, use Section 3.19 on page 104 group of commands.

Prefix no No

Change settings Yes

Multiple input No

Synopsis

(config)> **service dns-proxy**

Example

```
(config)> service dns-proxy
Dns::Manager: DNS proxy enabled.
```

History

Version	Description
2.00	The **service dns-proxy** command has been introduced.

3.103 service http

Description Enable HTTP-server that provides the user with Web-interface to configure City.

Command with **no** prefix stops the service.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config)> **service http**

(config)> **no service http**

Example

```
(config)> service http
HTTP server enabled.
```

History

Version	Description
2.00	The **service http** command has been introduced.

3.104 service igmp-proxy

Description

Enable IGMP-proxy. For the service functioning it is necessary to have one `upstream` interface and at least one `downstream` interface. If there are not enough settings to run the service, the service will not function. As soon as there are enough settings, the service will start automatically.

Command with **no** prefix stops the service.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config)> **service igmp-proxy**

(config)> **no service igmp-proxy**

Example

```
(config)> service igmp-proxy
IGMP proxy enabled.
```

History

Version	Description
2.00	The **service igmp-proxy** command has been introduced.

3.105 service internet-checker

Description

Enable the Internet-checker to monitor the state of Internet connection on the device. By default, service is enabled.

Command with **no** prefix stops the service.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config)> **service internet-checker**

(config)> **no service internet-checker**

Example

```
(config)> service internet-checker
Network::InternetChecker: Hosts check enabled.
```

```
(config)> no service internet-checker
Network::InternetChecker: Hosts check disabled.
```

History

Version	Description
2.13	The **service internet-checker** command has been introduced.

3.106 service ipsec

Description Enable *IPsec* service. By default, service is disabled.

Command with **no** prefix stops the service.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config)> **service ipsec**

(config)> **no service ipsec**

Example

```
(config)>service ipsec
IpSec::Manager: Service enabled.
```

History

Version	Description
2.06	The **service ipsec** command has been introduced.

3.107 service kabinet

Description Enable KABiNET authenticator service. By default it is disabled.

Command with **no** prefix stops the service.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config)> **service kabinet**

(config)> **no service kabinet**

Example

```
(config)> service kabinet
Kabinet::Authenticator: Authenticator enabled.
```

```
(config)> service kabinet
Kabinet::Authenticator: Authenticator disabled.
```

History

Version	Description
2.02	The **service kabinet** command has been introduced.

3.108 service mdns

Description Enable *mDNS* service. By default, service is enabled.

Command with **no** prefix stops the service.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config)> **service mdns**

(config)> **no service mdns**

Example

```
(config)>service mdns
```

```
(config)>no service mdns
```

History

Version	Description
2.15	The **service mdns** command has been introduced.

3.109 service mws

Description Enable *MWS* service. By default, service is disabled.

Command with **no** prefix stops the service.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config)> **service mws**

(config)> **no service mws**

Example

```
(config)> service mws
Mws::Controller: Enabled.
```

```
(config)> no service mws
Mws::Controller: Disabled.
```

History

Version	Description
2.15	The **service mws** command has been introduced.

3.110 service ntce

Description Enable *NTCE* service. By default it is disabled.

Command with **no** prefix stops the service.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config)> **service ntce**

(config)> **no service ntce**

Example

```
(config)> service ntce
Ntce::Manager: Enabled.
```

History

Version	Description
2.09	The **service ntce** command has been introduced. Previous command name is **service dpi**.

3.111 service ntp-client

Description Enable *NTP*-client.

Command with **no** prefix stops the service.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config)> **service ntp-client**

(config)> **no service ntp-client**

Example

```
(config)> service ntp-client
NTP client enabled.
```

History

Version	Description
2.00	The **service ntp-client** command has been introduced.

3.112 service snmp

Description

Enable *SNMP* service. By default, the service is disabled.

Command with **no** prefix stops the service.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config)> **service snmp**

(config)> **no service snmp**

Example

```
(config)> service snmp
Snmp::Manager: SNMP service was enabled.
(config)> no service snmp
Snmp::Manager: SNMP service was disabled.
```

History

Version	Description
2.08	The **service snmp** command has been introduced.

3.113 service ssh

Description

Enable the SSH server that provides the user with command line interface to configure the device.

Command with **no** prefix stops the service.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config)> **service ssh**

(config)> **no service ssh**

Example

```
(config)> service ssh
Ssh::Manager: SSH server enabled.
```

```
(config)> no service ssh
Ssh::Manager: SSH server disabled.
```

History

Version	Description
2.12	The **service ssh** command has been introduced.

3.114 service sstp-server

Description Enable *SSTP*-server.

Command with **no** prefix stops the service.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config)> **service sstp-server**

(config)> **no service sstp-server**

Example

```
(config)> service sstp-server
SstpServer::Manager: Service enabled.
```

```
(config)> no service sstp-server
SstpServer::Manager: Service disabled.
```

History

Version	Description
2.12	The **service sstp-server** command has been introduced.

3.115 service telnet

Description Enable the telnet server that provides the user with command line interface to configure the device.

Command with **no** prefix stops the service.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config)> **service telnet**

(config)> **no service telnet**

Example

```
(config)> service tel
Telnet server enabled.
```

History

Version	Description
2.00	The **service telnet** command has been introduced.

3.116 service udpxy

Description

Enable *udpxy* service.

Command with **no** prefix stops the service.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config)> **service udpxy**

(config)> **no service udpxy**

Example

```
(config)> service udpxy
Udpxy::Manager: a service enabled.
```

History

Version	Description
2.03	The **service udpxy** command has been introduced.

3.117 service upnp

Description

Enable *UPnP* service.

Command with **no** prefix stops the service.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config)> **service upnp**

(config)> **no service upnp**

History

Version	Description
2.00	The **service upnp** command has been introduced.

3.118 service vpn-server

Description Enable VPN-server.

Command with **no** prefix stops the service.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config)> **service vpn-server**

(config)> **no service vpn-server**

Example

```
(config)> service vpn-server
VpnServer::Manager: Service enabled.
```

```
(config)> no service vpn-server
VpnServer::Manager: Service disabled.
```

History

Version	Description
2.04	The **service vpn-server** command has been introduced.

3.119 show

Description Access to a group of commands to display various diagnostic information about system. All commands of this group do not change system settings.

Prefix no No

Change settings No

Multiple input No

Group entry (show)

Synopsis

(config)> **show**

History

Version	Description
2.00	The **show** command has been introduced.

3.119.1 show acme

Description Show *ACME* client status.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **acme**

Example

```
(show)> acme
    acme:
             real-time: yes
           ndns-domain: mytest.keenetic.pro
     ndns-domain-acme: yes
    ndns-domain-error: no
       default-domain: cc6b5a71a7644903b51a5454.keenetic.io
      account-pending: no
      account-running: no
          get-pending: no
          get-running: no
       revoke-pending: no
       revoke-running: no
    reissue-queue-size: 0
     revoke-queue-size: 0
               retries: 0
         checker-timer: 82499
           apply-timer: 0
          acme-account: 36902346
```

History

Version	Description
2.11	The **show acme** command has been introduced.

3.119.2 show adguard-dns availability

Description Check and show *AdGuard DNS* availability.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **adguard-dns availability**

Example

```
(show)> adguard-dns availability

       available: yes
            port: 53
```

History

Version	Description
2.12	The **show adguard-dns availability** command has been introduced.

3.119.3 show adguard-dns profiles

Description Show *AdGuard DNS* profiles.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **adguard-dns profiles**

Example

```
(show)> adguard-dns profiles

         profiles:
              profile: default

              profile: standard

              profile: family
```

History

Version	Description
2.11	The **show adguard-dns profiles** command has been introduced.

3.119.4 show associations

Description Show list of wireless stations associated with an access point. If you use no argument, the entire list of wireless stations will be displayed.

Prefix no No

Change settings No

Multiple input No

Interface type Access Point

Synopsis

(show)> **associations** [*‹name›*]

Arguments

Argument	Value	Description
name	*String*	An access point name. You can see the list of available access points with help of **associations** [Tab] command.

Example

```
(show)> associations [Tab]

 Usage template:
     associations [{name}]

   Choose:
 WifiMaster0/AccessPoint2
 WifiMaster1/AccessPoint1
 WifiMaster0/AccessPoint3
 WifiMaster0/AccessPoint0
              AccessPoint
 WifiMaster1/AccessPoint2
 WifiMaster0/AccessPoint1
               GuestWiFi
 WifiMaster1/AccessPoint3
 WifiMaster1/AccessPoint0
           AccessPoint_5G
```

```
(show)> associations WifiMaster0/AccessPoint0

          station:
                  mac: ec:1f:72:d3:6d:3f
                   ap: WifiMaster0/AccessPoint0
        authenticated: 1
                txrate: 130
                uptime: 3804
               txbytes: 2058837
               rxbytes: 25023483
                    ht: 20
                  mode: 11n
                    gi: 800
                  rssi: -26
                   mcs: 15

          station:
                  mac: 20:aa:4b:5c:09:0e
                   ap: WifiMaster0/AccessPoint0
        authenticated: 1
                txrate: 270
                uptime: 19662
               txbytes: 19450396
               rxbytes: 70800065
                    ht: 40
                  mode: 11n
                    gi: 800
```

```
                rssi: -41
                 mcs: 15
```

History

Version	Description
2.00	The **show associations** command has been introduced.

3.119.5 show button

Description Show information about specified system button. If you use no argument, the entire list of all buttons on the device will be displayed. Available buttons depend on hardware configuration.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **button** [*‹name›*]

Arguments

Argument	Value	Description
name	*String*	The button name.

Example

```
(show)> button FN1

          buttons:
               button, name = FN1:
                is_switch: no
                  position: 2
           position_count: 2
                   clicks: 0
                  elapsed: 0
               hold_delay: 3000
```

History

Version	Description
2.00	The **show button** command has been introduced.

3.119.6 show button bindings

Description Show a list of actions associated with device buttons.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **button bindings**

Example

```
(show)> button bindings

bindings:

          binding, index = 0:
               button: RESET
               action: click
       active_handler: Reboot
      default_handler: Reboot
            protected: yes

          binding, index = 1:
               button: RESET
               action: hold
       active_handler: FactoryReset
      default_handler: FactoryReset
            protected: yes

          binding, index = 2:
               button: WLAN
               action: click
       active_handler: WpsStartMainAp
      default_handler: WpsStartMainAp
            protected: no

          binding, index = 3:
               button: WLAN
               action: double-click
       active_handler: WpsStartMainAp5
      default_handler: WpsStartMainAp5
            protected: no

          binding, index = 4:
               button: WLAN
               action: hold
       active_handler: WifiToggle
      default_handler: WifiToggle
            protected: no

          binding, index = 5:
               button: FN1
               action: click
       active_handler: UnmountUsb1
      default_handler: UnmountUsb1
            protected: no

          binding, index = 6:
               button: FN1
               action: double-click
       active_handler:
      default_handler:
```

```
            protected: no

          binding, index = 7:
               button: FN1
               action: hold
       active_handler:
      default_handler:
            protected: no

          binding, index = 8:
               button: FN2
               action: click
       active_handler: UnmountUsb2
      default_handler: UnmountUsb2
            protected: no

          binding, index = 9:
               button: FN2
               action: double-click
       active_handler:
      default_handler:
            protected: no

          binding, index = 10:
               button: FN2
               action: hold
       active_handler:
      default_handler:
            protected: no
```

History

Version	Description
2.03	The **show button bindings** command has been introduced.

3.119.7 show button handlers

Description Show a list of available button handlers in the system.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **button handlers**

Example

```
(show)> button handlers

          handlers:
               handler, name = LedToggle:
```

```
         short_description: toggle system LED states
                 protected: no
            switch_related: no

              handler, name = FactoryReset:
         short_description: reset a configuration to factory ►
defaults
                 protected: yes
            switch_related: no

              handler, name = UnmountUsb1:
         short_description: unmount USB 1 port storages
                 protected: no
            switch_related: no

              handler, name = UnmountUsb2:
         short_description: unmount USB 2 port storages
                 protected: no
            switch_related: no

              handler, name = Reboot:
         short_description: reboot the system
                 protected: yes
            switch_related: no

              handler, name = DlnaDirectoryRescan:
        short_description: rescan DLNA directory for newer media ►
files
                 protected: no
            switch_related: no

              handler, name = DlnaDirectoryFullRescan:
         short_description: remove a DLNA database and rescan a ►
DLNA directory
                 protected: no
            switch_related: no

              handler, name = DectHandsetRegistrationToggle:
         short_description: toggle a DECT handset registration
                 protected: no
            switch_related: no

              handler, name = DectHandsetPagingToggle:
         short_description: toggle a DECT handset paging
                 protected: no
            switch_related: no

              handler, name = OpkgRunScript:
         short_description: run Opkg script
                 protected: no
            switch_related: no

              handler, name = TorrentAltSpeedToggle:
         short_description: toggle a Torrent alternative speed ►
```

```
mode
                protected: no
           switch_related: no

                handler, name = TorrentClientStateToggle:
         short_description: toggle a Torrent client state
                protected: no
           switch_related: no

                handler, name = WifiToggle:
         short_description: on/off all Wi-Fi interfaces
                protected: no
           switch_related: no

                handler, name = WpsStartMainAp:
        short_description: start WPS (2.4 GHz main access point)
                protected: no
           switch_related: no

                handler, name = WpsStartMainAp5:
         short_description: start WPS (5 GHz main access point)
                protected: no
           switch_related: no

                handler, name = WifiGuestApToggle:
          short_description: toggle a guest access point state ►
(2.4 GHz)
                protected: no
           switch_related: no

                handler, name = WpsStartStation:
         short_description: start WPS (2.4 GHz Wi-Fi station)
                protected: no
           switch_related: no

                handler, name = WpsStartStation5:
         short_description: start WPS (5 GHz Wi-Fi station)
                protected: no
           switch_related: no
```

History

Version	Description
2.03	The **show button handlers** command has been introduced.

3.119.8 show chilli profiles

Description Show the list of available *RADIUS*-server profiles.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **chilli profiles**

Example

```
(show)> chilli profiles

         profile:
                 name: Iron Wi-Fi
                  url: https://www.ironwifi.com/
          description: Hosted RADIUS and Captive Portal

               preset:
                uamserver: ►
https://europe-west3.ironwifi.com/api/pages/uam/

                   radius:
                      server1: 35.198.88.176

         radiuslocationid:

                      dns:
                         dns1: 8.8.8.8
                         dns2: 8.8.4.4

               custom: uamsecret

               custom: radiussecret

               custom: radiusnasid
```

History

Version	Description
2.10	The **show chilli profiles** command has been introduced.

3.119.9 show clock date

Description Show the current system date.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **clock date**

Example

```
(show)> clock date

          weekday: 4
              day: 18
            month: 1
```

```
            year: 2018
            hour: 8
             min: 46
             sec: 2
            msec: 660
             dst: inactive

              tz:
            locality: GMT
           stdoffset: 0
           dstoffset: 0
             usesdst: no
                rule: GMT0
              custom: no
```

History

Version	Description
2.00	The **show clock date** command has been introduced.

3.119.10 show clock timezone-list

Description Show the list of available timezones.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **clock timezone-list**

Example

```
(show)> clock timezone-list

       timezones:
                  tz:
                locality: Adak
               stdoffset: -36000
               dstoffset: -32400
                  tz:
                locality: Aden
               stdoffset: 10800
               dstoffset: -1
                  tz:
                locality: Almaty
               stdoffset: 21600
               dstoffset: -1
                  tz:
                locality: Amsterdam
               stdoffset: 3600
               dstoffset: 7200
                  tz:
```

```
                locality: Anadyr
               stdoffset: 43200
               dstoffset: -1
    ...
    ...
    ...
```

History

Version	Description
2.00	The **show clock timezone-list** command has been introduced.

3.119.11 show cloudflare-dns availability

Description Check and show *Cloudflare DNS* availability.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **cloudflare-dns availability**

Example

```
(show)> cloudflare-dns availability

        available: yes
   doh-supported: yes
   doh-available: yes
   dot-supported: yes
   dot-available: yes
    blocked-name: ►
31bd8460-89fd-e2de-8865-63ffb93d1c9e.is-cf.cloudflareresolve.com
  ipv6-supported: no
    ipv6-enabled: no
```

History

Version	Description
3.05	The **show cloudflare-dns availability** command has been introduced.

3.119.12 show cloudflare-dns profiles

Description Show *Cloudflare DNS* profiles.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **cloudflare-dns profiles**

Example

```
(show)> cloudflare-dns profiles

         profiles:
              profile: default

              profile: standard

              profile: malware

              profile: family
```

History

Version	Description
3.05	The **show cloudflare-dns profiles** command has been introduced.

3.119.13 show configurator status

Description Show information about system configurator.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **configurator status**

Example

```
(show)> configurator status

touch: Thu, 18 Oct 2018 14:37:25 GMT

           header, name = Model: Keenetic Giga

           header, name = Version: 2.06.1

           header, name = Agent: http/rci

          header, name = Last change: Thu, 18 Oct 2018 14:37:25 ►
GMT

          serving:
                 name: Session /var/run/ndm.core.socket
                 time: 0.000397

              request, host = 192.168.1.42, name = admin:
                    parse: show configurator status
```

History

Version	Description
2.06	The **show configurator status** command has been introduced.

3.119.14 show credits

Description

Show the license information about specified installed package in KeeneticOS. If you use no argument, the entire list of all installed packages on the device will be displayed.

Prefix no

No

Change settings

No

Multiple input

No

Synopsis

(show)> **credits** [*‹package›*]

Arguments

Argument	Value	Description
package	*String*	Package name.

Example

```
(show)> credits

          package:
                 name: accel-ppp
                title: High performance accel-ppp VPN server
             homepage: https://accel-ppp.org/

          package:
                 name: accel-ppp-l2tp
                title: L2TP plugin for accel-ppp
             homepage: https://accel-ppp.org/

          package:
                 name: accel-ppp-pptp
                title: PPTP plugin for accel-ppp
             homepage: https://accel-ppp.org/

          package:
                 name: accel-ppp-sstp
                title: SSTP plugin for accel-ppp
             homepage: https://accel-ppp.org/

          package:
                 name: avahi-daemon
                title: An mDNS/DNS-SD implementation (daemon)
             homepage: http://www.avahi.org/

          package:
                 name: coova-chilli
```

```
                title: Wireless LAN HotSpot controller (Coova ►
Chilli Version)
             homepage: http://www.coova.org/CoovaChilli

          package:
                 name: crconf
                title: Netlink-based CryptoAPI userspace ►
management utility
             homepage:

          package:
                 name: dhcpv6
                title: DHCPv6 client + server
             homepage: http://wide-dhcpv6.sourceforge.net/

          package:
                 name: dropbear
                title: Small SSH2 client/server
             homepage: http://matt.ucc.asn.au/dropbear/

          package:
                 name: iperf3-ssl
                title: Internet Protocol bandwidth measuring ►
tool with iperf_auth support
             homepage: https://github.com/esnet/iperf

          package:
                 name: kernel
                title: Linux kernel
             homepage: http://www.kernel.org/

          package:
                 name: kmod-ipt-account
                title: ACCOUNT netfilter module
             homepage:

          package:
                 name: kmod-ipt-chaos
                title: CHAOS netfilter module
             homepage:

          package:
                 name: kmod-ipt-compat-xtables
                title: API compatibilty layer netfilter module
             homepage:

          package:
                 name: kmod-ipt-condition
                title: Condition netfilter module
             homepage:

          package:
                 name: kmod-ipt-delude
                title: DELUDE netfilter module
```

```
            homepage:

         package:
                name: kmod-ipt-dhcpmac
               title: DHCPMAC netfilter module
            homepage:

         package:
                name: kmod-ipt-dnetmap
               title: DNETMAP netfilter module
            homepage:

         package:
                name: kmod-ipt-fuzzy
               title: fuzzy netfilter module
            homepage:

         package:
                name: kmod-ipt-geoip
               title: geoip netfilter module
            homepage:

         package:
                name: kmod-ipt-iface
               title: iface netfilter module
            homepage:

         package:
                name: kmod-ipt-ipmark
               title: IPMARK netfilter module
            homepage:

         package:
                name: kmod-ipt-ipp2p
               title: IPP2P netfilter module
            homepage:

         package:
                name: kmod-ipt-ipv4options
               title: ipv4options netfilter module
            homepage:

         package:
                name: kmod-ipt-length2
               title: length2 netfilter module
            homepage:

         package:
                name: kmod-ipt-logmark
               title: LOGMARK netfilter module
            homepage:

         package:
                name: kmod-ipt-lscan
```

```
               title: lscan netfilter module
            homepage:

          package:
                 name: kmod-ipt-netflow
                title: Netflow netfilter module for Linux kernel
            homepage: http://ipt-netflow.sourceforge.net/

          package:
                 name: kmod-ipt-psd
                title: psd netfilter module
            homepage:

          package:
                 name: kmod-ipt-quota2
                title: quota2 netfilter module
            homepage:

          package:
                 name: kmod-ipt-sysrq
                title: SYSRQ netfilter module
            homepage:

          package:
                 name: kmod-ipt-tarpit
                title: TARPIT netfilter module
            homepage:

          package:
                 name: kmod-nf-nathelper-rtsp
                title: RTSP Conntrack and NAT helpers
            homepage: https://github.com/maru-sama/rtsp-linux

          package:
                 name: kmod-wireguard
                title: WireGuard kernel module
            homepage:

          package:
                 name: libattr
               title: Extended attributes (xattr) manipulation ►
library
            homepage: http://savannah.nongnu.org/projects/attr

          package:
                 name: libav
                title: This package contains Libav library
            homepage: https://libav.org/

          package:
                 name: libavahi
                title: An mDNS/DNS-SD implementation (No D-Bus)
            homepage: http://www.avahi.org/
```

```
          package:
                 name: libcurl
                title: A client-side URL transfer library
             homepage: http://curl.haxx.se/

          package:
                 name: libdaemon
                title: A lightweight C library that eases the ►
writing of UNIX daemons
             homepage: ►
http://0pointer.de/lennart/projects/libdaemon/

          package:
                 name: libdb47
                title: Berkeley DB library (4.7)
            homepage: http://www.sleepycat.com/products/db.shtml

          package:
                 name: libevent
                title: Event notification library
             homepage: http://www.monkey.org/~provos/libevent/

          package:
                 name: libexif
                title: Library for JPEG files with EXIF tags
             homepage: https://libexif.github.io

          package:
                 name: libexpat
                title: An XML parsing library
             homepage: https://libexpat.github.io/

          package:
                 name: libgcrypt
                title: GNU crypto library
             homepage: ►
http://directory.fsf.org/security/libgcrypt.html

          package:
                 name: libgpg-error
                title: GnuPG error handling helper library
             homepage: ►
http://www.gnupg.org/related_software/libgpg-error/

          package:
                 name: libid3tag
                title: An ID3 tag manipulation library
             homepage: https://www.underbit.com/products/mad/

          package:
                 name: libjpeg
                title: The Independent JPEG Group's JPEG runtime ►
library
             homepage: http://www.ijg.org/
```

```
          package:
                 name: liblzo
                title: A real-time data compression library
             homepage: http://www.oberhumer.com/opensource/lzo/

          package:
                 name: libnghttp2
                title: Library implementing the framing layer ►
of HTTP/2
             homepage: https://nghttp2.org/

          package:
                 name: libopenssl
                title: Open source SSL toolkit (libraries ►
(libcrypto.so, libssl.so))
             homepage: http://www.openssl.org/

          package:
                 name: libpcap
                title: Low-level packet capture library
             homepage: http://www.tcpdump.org/

          package:
                 name: libtommath
                title: A free number theoretic multiple-precision ►
integer library
             homepage: https://www.libtom.net/

          package:
                 name: libusb
                title: A library for accessing Linux USB devices
             homepage: http://libusb.info/

          package:
                 name: mini_snmpd
                title: Lightweight SNMP daemon
            homepage: http://troglobit.github.io/mini-snmpd.html

          package:
                 name: minidlna
                title: UPnP A/V & DLNA Media Server
             homepage: http://minidlna.sourceforge.net/

          package:
                 name: miniupnpd
                title: Lightweight UPnP daemon
             homepage: http://miniupnp.tuxfamily.org/

          package:
                 name: netatalk
                title: netatalk
             homepage: http://netatalk.sourceforge.net
```

```
          package:
                 name: nginx
                title: Nginx web server
             homepage: http://nginx.org/

          package:
                 name: nginx-stream-module
                title: Nginx stream module
             homepage:

          package:
                 name: openvpn
                title: Open source VPN solution using OpenSSL
             homepage: http://openvpn.net

          package:
                 name: pjproject
                title: PJSIP
             homepage: http://www.pjsip.org/

          package:
                 name: pureftpd
                title: FTP server
             homepage: http://www.pureftpd.org

          package:
                 name: radvd
                title: Router advertisement daemon
             homepage: http://www.litech.org/radvd/

          package:
                 name: sstp-client
                title: SSTP client for Linux
             homepage: http://sstp-client.sourceforge.net/

          package:
                 name: strongswan
               title: Strongswan IKEv1/IKEv2 ISAKMP and IPSec ►
suite
             homepage: https://www.strongswan.org/

          package:
                 name: transmission-daemon
                title: A free, lightweight BitTorrent client
             homepage: http://www.transmissionbt.com

          package:
                 name: tspc
                title: TSP client
             homepage: http://www.broker.ipv6.ac.uk

          package:
                 name: tzdata
                title: Timezone data files
```

```
             homepage: https://www.iana.org/time-zones

          package:
                 name: udpxy
                title: Convert UDP IPTV streams into HTTP stream
             homepage: http://sourceforge.net/projects/udpxy

          package:
                 name: zlib
                title: Library implementing the deflate ►
compression method
             homepage: http://www.zlib.net/
```

```
(show)> credits nginx

          copying: /*
                   * Copyright (C) 2002-2019 Igor Sysoev
                   * Copyright (C) 2011-2019 Nginx, Inc.
                   * All rights reserved.
                   *
                  * Redistribution and use in source and binary ►
forms, with or without
                   * modification, are permitted provided that ►
the following conditions
                   * are met:
                   * 1. Redistributions of source code must ►
retain the above copyright
                  *    notice, this list of conditions and the ►
following disclaimer.
                   * 2. Redistributions in binary form must ►
reproduce the above copyright
                  *    notice, this list of conditions and the ►
following disclaimer in the
                   *    documentation and/or other materials ►
provided with the distribution.
                   *
                  * THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND ►
CONTRIBUTORS ``AS IS'' AND
                   * ANY EXPRESS OR IMPLIED WARRANTIES, ►
INCLUDING, BUT NOT LIMITED TO, THE
                  * IMPLIED WARRANTIES OF MERCHANTABILITY AND ►
FITNESS FOR A PARTICULAR PURPOSE
                   * ARE DISCLAIMED.  IN NO EVENT SHALL THE ►
AUTHOR OR CONTRIBUTORS BE LIABLE
                   * FOR ANY DIRECT, INDIRECT, INCIDENTAL, ►
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
                   * DAMAGES (INCLUDING, BUT NOT LIMITED TO, ►
PROCUREMENT OF SUBSTITUTE GOODS
                  * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; ►
OR BUSINESS INTERRUPTION)
                   * HOWEVER CAUSED AND ON ANY THEORY OF ►
LIABILITY, WHETHER IN CONTRACT, STRICT
                   * LIABILITY, OR TORT (INCLUDING NEGLIGENCE ►
OR OTHERWISE) ARISING IN ANY WAY
```

```
                    * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ►
ADVISED OF THE POSSIBILITY OF
                    * SUCH DAMAGE.
                    */
```

History

Version	Description
3.01	The **show credits** command has been introduced.

3.119.15 show crypto ike key

Description

Show info about selected *IKE* key. If you use no argument, the entire list of *IKE* keys will be displayed.

Prefix no

No

Change settings

No

Multiple input

No

Synopsis

(show)> **crypto ike key** [*name*]

Arguments

Argument	Value	Description
name	*String*	Name of selected *IKE* key.

Example

```
(show)> crypto ike key

            IpSec:
              ike_key, name = test:
                     type: address
                       id: 10.10.10.10

              ike_key, name = test2:
                     type: any
                       id: ►
```

History

Version	Description
2.06	The **show crypto ike key** command has been introduced.

3.119.16 show crypto map

Description

Show info about selected *IPsec* crypto map. If you use no argument, the entire list of *IPsec* crypto maps will be displayed.

Prefix no

No

Change settings

No

Multiple input No

Synopsis

(show)> **crypto map** [*map-name*]

Arguments

Argument	Value	Description
map-name	*String*	Name of selected crypto map.

Example

```
(show)> crypto map test

            IpSec:
           crypto_map, name = test:
                   config:
                            remote_peer: ipsec.example.com
             crypto_ipsec_profile_name: prof1
                                  mode: tunnel

                         local_network:
                              net: 172.16.200.0
                             mask: 24
                         protocol: IPv4

                        remote_network:
                              net: 172.16.201.0
                             mask: 24
                         protocol: IPv4

                   status:
                 primary_peer: true

                      phase1:
                            name: test
                       unique_id: 572
                       ike_state: ESTABLISHED
                  establish_time: 1451301596
                      rekey_time: 0
                     reauth_time: 1451304277
                      local_addr: 10.10.10.15
                     remote_addr: 10.10.10.20
                     ike_version: 2
                        local_spi: 00a6ebfc9d90f1c2
                       remote_spi: 3cd201ef496df75c
                       local_init: yes
                       ike_cypher: aes-cbc-256
                         ike_hmac: sha1
                    ike_dh_group: 2

                phase2_sa_list:
                          phase2_sa, index = 0:
                              unique_id: 304
                             request_id: 185
                               sa_state: INSTALLED
```

```
                                    mode: TUNNEL
                                protocol: ESP
                           encapsulation: yes
                               local_spi: ca59bfcf
                              remote_spi: cde23d83
                            ipsec_cypher: esp-aes-256
                              ipsec_hmac: esp-sha1-hmac
                          ipsec_dh_group:
                                in_bytes: 7152
                              in_packets: 115
                                 in_time: 1451302507
                               out_bytes: 6008
                             out_packets: 98
                                out_time: 1451302507
                              rekey_time: 1451305159
                                local_ts: 172.16.200.0/24
                               remote_ts: 172.16.201.0/24

                           state: PHASE2_ESTABLISHED
```

History

Version	Description
2.06	The **show crypto map** command has been introduced.

3.119.17 show defaults

Description Show the general default wireless and system parameters.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **defaults**

Example

```
(show)> defaults

       servicetag: 014635737374***
     servicehost: ndss.keenetic.ndmsystems.com
     servicepass: ***************
        wlanssid: Keenetic-0000
         wlankey: xFxTH***
         wlanwps: 75534***
         country: RU
         ndmhwid: KN-1010
         ctrlsum: 4712e0849ccea477ccdd18e2fedb***
          serial: S1749WF***
       signature: valid
       integrity: ok
          locked: yes
```

History

Version	Description
2.00	The **show defaults** command has been introduced.

3.119.18 show dns-proxy

Description Show a list of current *DNS over TLS* and *DNS over HTTPS* servers.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **dns-proxy**

Example

```
(show)> dns-proxy

     proxy-status:
              proxy-name: System

            proxy-config:

rpc_port = 54321
rpc_ttl = 10000
rpc_wait = 10000
timeout = 7000
proceed = 500
stat_file = /var/ndnproxymain.stat
stat_time = 10000
dns_server = 127.0.0.1:40500 .
dns_server = 127.0.0.1:40501 .
dns_server = 127.0.0.1:40508 .
dns_server = 127.0.0.1:40509 .
static_a = my.keenetic.net 78.47.125.180
static_a = cc6b5a71a7644903b51a5454.keenetic.io 78.47.125.180
static_a = myhome23.keenetic.pro 78.47.125.180
set-profile-ip 127.0.0.1 0
set-profile-ip ::1 0
dns_tcp_port = 53
dns_udp_port = 53

              proxy-stat:

# ndnproxy statistics file

Total incoming requests: 809
Proxy requests sent:     659
Cache hits ratio:        0.192 (155)
Memory usage:            44.41K

DNS Servers
```

```
                      Ip   Port  R.Sent  A.Rcvd  NX.Rcvd  ►
Med.Resp  Avg.Resp  Rank
               127.0.0.1  40500       2       2        0      ►
40ms      40ms    10
               127.0.0.1  40501     652     651        0      ►
17ms      17ms    10
              127.0.0.1  40508       2       0        0       ►
0ms       0ms     4
               127.0.0.1  40509       3       1        0     ►
326ms     326ms     3

             proxy-safe:


              proxy-tls:
              server-tls:
                     address: 1.1.1.1
                        port: 853
                         sni: cloudflare-dns.com
                        spki:
                   interface:

              server-tls:
                     address: 8.8.8.8
                        port: 853
                         sni: dns.google.com
                        spki:
                   interface:

       proxy-tls-filters:

             proxy-https:
             server-https:
                         uri: https://dns.adguard.com/dns-query
                      format: dnsm
                        spki:
                   interface:

             server-https:
                          uri: ►
https://cloudflare-dns.com/dns-query?ct=application/dns-json
                      format: json
                        spki:
                   interface:

     proxy-https-filters:
```

History

Version	Description
3.01	The **show dns-proxy** command has been introduced.

3.119.19 show dpn document

Description Show *DPN* agreement text.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **dpn document** [‹*version*›] [‹*language*›]

Arguments

Argument	Value	Description
version	*String*	Version of *DPN*. If not specified, the latest version is shown.
language	*String*	The language of *DPN*. If not specified, the English version is shown.

Example

```
(show)> dpn document
20200330

DEVICE PRIVACY NOTICE

Last update 2020-30-03

This End User License Agreement (this “Agreement”) constitutes ►
a valid and
binding agreement between Keenetic Limited, including all ►
affiliates and
subsidiaries (“Keenetic”, “us”, “our” or “we”) and You (as ►
defined below)
of the Software (as defined below), including the Software ►
installed onto
any one of our Keenetic products (the “Product”) and/or the ►
Software
legally obtained from or provided by an App Platform (as defined ►
below)
authorised by Keenetic. Keenetic and You shall be collectively ►
referred to
as the “Parties”, and individually as a “Party”.
```

```
(show)> dpn document 20200330 es
20200330

CONTRATO DE LICENCIA DEL USUARIO FINAL

Última actualización 30/03/2020

El presente contrato de licencia del usuario final (el presente ►
Contrato“)
constituye un acuerdo válido y vinculante celebrado entre Keenetic
```

```
Limited, incluidas todas las filiales y sucursales (”Keenetic“,
”nosotros“, ”nuestro/a“ o ”nos“) y Usted (tal y como se define a
continuación) del Software (tal y como se define a continuación), ►
incluido
el Software instalado en cualquiera de nuestros productos de ►
Keenetic (el
”Producto“) y/o el Software obtenido o proporcionado legalmente ►
por la
Plataforma de la aplicación (tal y como se define a continuación)
autorizado por Keenetic. Se referirá a Keenetic y Usted, en ►
conjunto, como
las ”Partes“ y, de forma individual, como una ”Parte“.
```

History

Version	Description
3.05	The **show dpn document** command has been introduced.

3.119.20 show dpn list

Description Show the list of *DPN* available in the system.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **dpn list**

Example

```
(show)> dpn list
              dpn:
              version: 20200330

             document:
                     lang: de

                   format: txt

                   format: md

             document:
                     lang: en

                   format: txt

                   format: md

             document:
                     lang: es

                   format: txt
```

```
      format: md

document:
        lang: fr

      format: txt

      format: md

document:
        lang: it

      format: txt

      format: md

document:
        lang: pl

      format: txt

      format: md

document:
        lang: pt

      format: txt

      format: md

document:
        lang: ru

      format: txt

      format: md

document:
        lang: sv

      format: txt

      format: md

document:
        lang: tr

      format: txt

      format: md

document:
        lang: uk
```

```
                format: txt

                format: md
```

History

Version	Description
3.05	The **show dpn list** command has been introduced.

3.119.21 show dot1x

Description

Show 802.1x client status on the interface. To manage 802.1x client status on the interface authentication must be configured with **interface authentication** group of commands.

Prefix no

No

Change settings

No

Interface type

Ethernet

Multiple input

No

Synopsis

(show)> **dot1x** [*interface*]

Arguments

Argument	Value	Description
interface	*Interface name*	An Ethernet interface name. You can see the list of available Ethernet interfaces with help of **dot1x** [Tab] command.

Example

```
(show)> dot1x [Tab]

 Usage template:
           dot1x [{name}]

   Choose:
         GigabitEthernet1
                      ISP
 WifiMaster0/AccessPoint2
 WifiMaster1/AccessPoint1
 WifiMaster0/AccessPoint3
 WifiMaster0/AccessPoint0
             AccessPoint
```

```
(show)> dot1x ISP

           dot1x:
                  id: FastEthernet0/Vlan2
               state: CONNECTING
```

History

Version	Description
2.02	The **show dot1x** command has been introduced.

3.119.22 show drivers

Description Show the list of loaded kernel drivers.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **drivers**

Example

```
(show)> drivers

          module:
                name: rt2860v2_sta
                size: 546736
                used: 0
                subs: -
          module:
                name: rt2860v2_ap
                size: 554192
                used: 2
                subs: -
          module:
                name: rndis_host
                size: 5024
                used: 0
                subs: -
          module:
                name: dwc_otg
                size: 68416
                used: 0
                subs: -
          module:
                name: lm
                size: 1344
                used: 1
                subs: dwc_otg,[permanent]
    ...
    ...
    ...
```

History

Version	Description
2.00	The **show drivers** command has been introduced.

3.119.23 show dyndns updaters

Description Show the list of available DynDNS providers.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **dyndns updaters**

Example

```
(show)> dyndns updaters

          updater:
                 type: dyndns
                  url: https://account.dyn.com/dns/dyndns
                  api: http://members.dyndns.org/nic/update

          updater:
                 type: noip
                  url: https://www.noip.com/
                  api: http://dynupdate.no-ip.com/nic/update
```

History

Version	Description
2.12	The **show dyndns updaters** command has been introduced.

3.119.24 show easyconfig status

Description Show EasyConfig status and settings.

Prefix no No

Change settings No

Multiple input No

Synopsis (show)> **easyconfigstatus**

Example

```
(show)> easyconfig status

       easyconfig:
                checked: Tue Aug  6 11:50:21 2019
                enabled: yes
               reliable: yes
    gateway-accessible: yes
        dns-accessible: yes
       host-accessible: yes
              internet: yes
```

```
               gateway:
               interface: GigabitEthernet1
                 address: 193.0.175.2
                failures: 0
              accessible: yes
                excluded: no

                 hosts:
                    host:
                        name: google.com
                    failures: 0
                    resolved: no
                  accessible: no
```

History

Version	Description
2.00	The **show easyconfig status** command has been introduced.

3.119.25 show eula document

Description Show *EULA* agreement text.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **eula document** [*‹version›*] [*‹language›*]

Arguments

Argument	Value	Description
version	*String*	Version of *EULA*. If not specified, the latest version is shown.
language	*String*	The language of *EULA*. If not specified, the English version is shown.

Example

```
(show)> eula document 20181001
20181001

KEENETIC LIMITED
End User License Agreement

This End User License Agreement (this “Agreement”) constitutes ►
a valid and binding agreement between Keenetic Limited, including ►
all affiliates and subsidiaries (“Keenetic”, “us”, “our” or “we”) ►
and You (as
defined below) of the Software (as defined below), including the ►
Software installed onto any one of our Keenetic products (the ►
“Product”) and/or the Software legally obtained from or provided ►
```

```
by an App Platform
(as defined below) authorised by Keenetic. Keenetic and You shall ►
be collectively referred to as the “Parties”, and individually ►
as a “Party”.
```

```
(show)> eula document 20181001 ru
20181001

KEENETIC LIMITED
Лицензионное соглашение с конечным пользователем

Настоящее Лицензионное соглашение с конечным пользователем ►
(настоящее «Соглашение») представляет собой действительное и ►
обязательное соглашение между Keenetic Limited, включая все ►
связанные с ней компании и все
её подразделения («Keenetic», «нам», «наш» или «мы»), и Вами ►
(как определено ниже) о Программном обеспечении (как определено ►
ниже), включая Программное обеспечение, устанавливаемое на любом ►
из продуктов
производства Keenetic («Продукт») и/или Программное обеспечение, ►
полученное на законных основаниях или предоставленное Магазином ►
Приложений (как определено ниже), авторизованной Keenetic. ►
Keenetic и Вы вместе
упоминаетесь как «Стороны», а по отдельности — «Сторона».
```

History

Version	Description
2.15	The **show eula document** command has been introduced.

3.119.26 show eula list

Description Show the list of *EULA* available in the system.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **eula list**

Example

```
(show)> eula list
              eula:
              version: 20181001

             document:
                     lang: en

                  format: md

                  format: txt
```

```
            document:
                    lang: ru

                  format: md

                  format: txt

            document:
                    lang: tr

                  format: md

                  format: txt

            document:
                    lang: uk

                  format: md

                  format: txt
```

History

Version	Description
2.15	The **show eula list** command has been introduced.

3.119.27 show interface

Description Show information of specified interface. If you use no argument, the entire list of all network interfaces will be displayed.

Prefix no No

Change settings No

Multiple input No

Interface type IP

Synopsis

(show)> **interface** *‹name›*

Arguments

Argument	Value	Description
name	*Interface name*	Full name or an alias of the interface to display.

Example

Example 3.1. Review the status of switch ports

The command **show interface** displays different information depending on the interface type. In particular, for `FastEthernet0/Vlan1` switch it shows

current state of physical ports, speed and duplex, on top of general information.

```
(config)> show interface FastEthernet0/Vlan1

              id: GigabitEthernet0
           index: 0
            type: GigabitEthernet
     description:
  interface-name: GigabitEthernet0
            link: up
       connected: yes
           state: up
             mtu: 1500
        tx-queue: 2000

            port, name = 1:
                  id: GigabitEthernet0/0
               index: 0
      interface-name: 1
                type: Port
                link: up
               speed: 1000
              duplex: full
    auto-negotiation: on
        flow-control: on
                 eee: off
         last-change: 4578.185413
        last-overflow: 0
              public: no

            port, name = 2:
                  id: GigabitEthernet0/1
               index: 1
      interface-name: 2
                type: Port
                link: down
          last-change: 4590.205656
         last-overflow: 0
              public: no

            port, name = 3:
                  id: GigabitEthernet0/2
               index: 2
      interface-name: 3
                type: Port
                link: up

                role, for = GigabitEthernet0/Vlan2: inet

               speed: 100
              duplex: full
    auto-negotiation: on
        flow-control: off
```

```
                eee: off
         last-change: 4570.078144
       last-overflow: 0
              public: yes

            port, name = 4:
                  id: GigabitEthernet0/3
               index: 3
      interface-name: 4
                type: Port
                link: down
         last-change: 4590.202571
       last-overflow: 0
              public: no
```

History

Version	Description
2.00	The **show interface** command has been introduced.

3.119.28 show interface bridge

Description Display interface bridge status.

Prefix no No

Change settings No

Multiple input No

Interface type Bridge

Synopsis

(show)> **interface** ‹*name*› **bridge**

Arguments

Argument	Value	Description
name	*Interface name*	Full name or an alias of the interface to display.

Output

Element	Value
members	Root node.
interface	Interface name.
link	Link state of interface.
inherited	Attribute of inheritance.

Example

```
(show)> interface Bridge1 bridge

          members:
            interface, link = no, inherited = yes:
```

```
                        WifiMaster0/AccessPoint2
            interface, link = yes: UsbLte0
```

History

Version	Description
2.03	The **show interface bridge** command has been introduced.

3.119.29 show interface channels

Description Show information about the specified wireless interface channels.

Prefix no No

Change settings No

Multiple input No

Interface type Radio

Synopsis

(show)> **interface** ‹*name*› **channels**

Arguments

Argument	Value	Description
name	*Interface name*	Full name or an alias of the interface to display.

Output

Element	Value
channels	Root node.
channel, index	Record number in the list.
number	Channel number.
ext-40-above	Ability to expand channel above.
ext-40-below	Ability to expand channel below.
vhc-80	Ability to expand channel up to 80 MHz.

Example

```
(show)> interface WifiMaster0 channels

         channels:
              channel, index = 0:
                   number: 1
              ext-40-above: yes
              ext-40-below: no
                   vht-80: yes

              channel, index = 1:
                   number: 2
              ext-40-above: yes
```

```
              ext-40-below: yes
                    vht-80: yes

               channel, index = 2:
                    number: 3
              ext-40-above: yes
              ext-40-below: yes
                    vht-80: yes

               channel, index = 3:
                    number: 4
              ext-40-above: yes
              ext-40-below: yes
                    vht-80: yes

               channel, index = 4:
                    number: 5
              ext-40-above: yes
              ext-40-below: yes
                    vht-80: yes

               channel, index = 5:
                    number: 6
              ext-40-above: yes
              ext-40-below: yes
                    vht-80: yes

               channel, index = 6:
                    number: 7
              ext-40-above: yes
              ext-40-below: yes
                    vht-80: yes

               channel, index = 7:
                    number: 8
              ext-40-above: yes
              ext-40-below: yes
                    vht-80: yes
...
...
...
```

History

Version	Description
2.03	The **show interface channels** command has been introduced.

3.119.30 show interface chilli

Description Show information about statistics of connected clients to the *RADIUS* hotspot.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **interface** ‹*name*› **chilli**

Arguments

Argument	Value	Description
name	*Interface name*	Full name or an alias of the interface.

Example

```
(show)> interface Chilli0 chilli

            host:
          session-id: 4bf7c55f00000006
               user: 44w3c1
                 ip: 10.1.30.3
                mac: 55:a3:f9:51:b4:11
          start-time: 3884
            end-time: 0
           idle-time: 9
     idle-time-limit: 0
            tx-bytes: 695682
       tx-bytes-limit: 0
            rx-bytes: 1627453
       rx-bytes-limit: 0
            tx-speed: 0
       tx-speed-limit: 0
            rx-speed: 0
       rx-speed-limit: 0
```

History

Version	Description
2.10	The **show interface chilli** command has been introduced.

3.119.31 show interface country-codes

Description Show the list of available country codes on a radio interface.

Prefix no No

Change settings No

Multiple input No

Interface type Radio

Synopsis

(show)> **interface** ‹*name*› **country-codes**

Arguments

Argument	Value	Description
name	*Interface name*	Full name or an alias of the interface to display.

Output

Element	Value
country-codes	Root node.
code	Country code.
country	Country name.

Example

```
(show)> interface WifiMaster0 country-codes

    country-codes:
         country-code:
                     code: AL
                  country: Albania

         country-code:
                     code: DZ
                  country: Algeria

         country-code:
                     code: AR
                  country: Argentina

         country-code:
                     code: AM
                  country: Armenia

         country-code:
                     code: AU
                  country: Australia
...
...
...
```

History

Version	Description
2.03	The **show interface country-codes** command has been introduced.

3.119.32 show interface mac

Description Show the table of MAC-addresses of the switch.

Prefix no No

Change settings No

Multiple input No

Interface type Switch

Synopsis

(show)> **interface** *‹name›* **mac**

Arguments

Argument	Value	Description
name	*Interface name*	Full name or an alias of the interface to display.

Example

```
(show)> interface FastEthernet0 mac
============================================================
Port  MAC                        Aging
============================================================
0     b0:b2:dc:70:c4:28          6
0     f0:1b:21:6d:9a:c5          4
0     00:0c:43:76:20:77          6
0     b4:18:d1:6e:b5:6a          3
0     40:4a:03:78:01:af          2
0     84:8e:0c:3f:79:05          5
0     ec:43:f6:73:0a:99          6
0     ec:43:f6:04:2b:05          6
0     b2:b2:dc:5f:09:b3          1
0     ec:43:f6:72:4e:51          6
0     00:30:48:93:91:a7          6
0     f0:c1:f1:95:c3:fb          5
0     b8:ca:3a:8a:c7:43          6
0     ec:43:f6:da:78:79          5
0     10:7b:ef:59:7b:61          2
0     ec:43:f6:ff:f8:8b          6
0     58:8b:f3:65:8c:91          5
0     ec:43:f6:cf:0e:ef          2
0     00:ee:bd:a1:18:51          6
0     ec:43:f6:72:4e:69          6
0     90:e2:ba:07:9a:81          6
0     00:00:5e:00:01:01          6
0     00:08:9b:dc:8d:17          4
0     50:e5:49:58:2b:5a          6
0     90:e2:ba:07:99:55          6
0     ec:43:f6:04:36:8d          6
0     ec:43:f6:05:44:49          6
0     de:06:21:02:b3:e2          6
0     40:4a:03:60:80:05          6
0     00:0c:29:d5:84:c0          6
0     00:08:9b:dc:92:55          6
0     00:08:9b:dc:92:56          6
0     00:1b:0c:7f:b6:41          6
0     10:2a:b3:a6:86:18          5
0     10:7b:ef:df:83:a7          1
0     01:00:5e:00:00:fb          0
........................
```

History

Version	Description
2.00	The **show interface mac** command has been introduced.

3.119.33 show interface rf e2p

Description Show the current contents of all calibration data cells.

Prefix no No

Change settings No

Multiple input No

Interface type Radio

Synopsis

(show)> **interface** ‹*name*› **rf e2p**

Arguments

Argument	Value	Description
name	*Interface name*	Full name or an alias of the interface to display.

Example

```
(show)> interface WifiMaster0 rf e2p

[0x0000]:5392  [0x0002]:0103  [0x0004]:43EC  [0x0006]:04F6
[0x0008]:042B  [0x000A]:5392  [0x000C]:1814  [0x000E]:8001
[0x0010]:0000  [0x0012]:5392  [0x0014]:1814  [0x0016]:0000
[0x0018]:0001  [0x001A]:FF6A  [0x001C]:0213  [0x001E]:FFFF
[0x0020]:FFFF  [0x0022]:FFC1  [0x0024]:9201  [0x0026]:FFFF
[0x0028]:43EC  [0x002A]:04F6  [0x002C]:052B  [0x002E]:FFFF
[0x0030]:758E  [0x0032]:4301  [0x0034]:FF22  [0x0036]:0025
[0x0038]:FFFF  [0x003A]:012D  [0x003C]:FFFF  [0x003E]:FAD9
[0x0040]:88CC  [0x0042]:FFFF  [0x0044]:FF0A  [0x0046]:0000
[0x0048]:0000  [0x004A]:0000  [0x004C]:0000  [0x004E]:FFFF
[0x0050]:FFFF  [0x0052]:1111  [0x0054]:1111  [0x0056]:1111
[0x0058]:1011  [0x005A]:1010  [0x005C]:1010  [0x005E]:1010
[0x0060]:1111  [0x0062]:1211  [0x0064]:1212  [0x0066]:1312
[0x0068]:1313  [0x006A]:1413  [0x006C]:1414  [0x006E]:2264
[0x0070]:00F1  [0x0072]:1133  [0x0074]:0000  [0x0076]:FC62
[0x01E8]:FFFF  [0x01EA]:FFFF  [0x01EC]:FFFF  [0x01EE]:FFFF
[0x01F0]:FFFF  [0x01F2]:FFFF  [0x01F4]:FFFF  [0x01F6]:FFFF
[0x01F8]:FFFF  [0x01FA]:FFFF  [0x01FC]:FFFF  [0x01FE]:FFFF
.................................
```

History

Version	Description
2.04	The **show interface rf e2p** command has been introduced.

3.119.34 show interface rrd

Description Show network interface loading on the principle of Round Robin Database.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **interface** ‹*name*› **rrd** ‹*attribute*› [‹*detail*›]

Arguments

Argument	Value	Description
name	*Interface name*	Full name or an alias of the interface.
attribute	rxspeed	Value of data rate type.
	txspeed	
detail	0	Level of detail is 1 second.
	1	Level of detail is 2 seconds.
	2	Level of detail is 3 seconds.
	3	Level of detail is 5 seconds.
	4	Level of detail is 15 seconds.
	5	Level of detail is 30 seconds.
	6	Level of detail is 1 minute.
	7	Level of detail is 2 minutes.
	8	Level of detail is 3 minutes.
	9	Level of detail is 5 minutes.
	10	Level of detail is 15 minutes.
	11	Level of detail is 30 minutes.

Example

```
(show)> interface GigabitEthernet1 rrd rxspeed

              data:
                    t: 90083.990183
                    v: 200880

             data:
                    t: 90082.990128
                    v: 152392

             data:
                    t: 90081.990193
                    v: 110976

             data:
```

```
                    t: 90080.990142
                    v: 48000

             data:
                    t: 90079.990178
                    v: 38366
```

```
(show)> interface GigabitEthernet1 rrd txspeed

             data:
                    t: 87771.249486
                    v: 148202

             data:
                    t: 87768.248974
                    v: 10694

             data:
                    t: 87765.248977
                    v: 19070

             data:
                    t: 87762.249105
                    v: 48909

             data:
                    t: 87759.249105
                    v: 149277
```

```
(show)> interface GigabitEthernet1 rrd rxspeed 1

 data:
                    t: 90176.990054
                    v: 164766

             data:
                    t: 90174.990061
                    v: 121828

             data:
                    t: 90172.990052
                    v: 95430

             data:
                    t: 90170.990085
                    v: 57559

             data:
                    t: 90168.990119
                    v: 97759
```

History

Version	Description
2.10	The **show interface rrd** command has been introduced.

3.119.35 show interface stat

Description Show interface statistics.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **interface** ‹*name*› **stat**

Arguments

Argument	Value	Description
name	*Interface name*	Full name or an alias of the interface.

Example

```
(show)> interface WifiMaster0/AccessPoint0 stat

        rxpackets: 137033
          rxbytes: 23915722
         rxerrors: 0
        rxdropped: 0
        txpackets: 847802
          txbytes: 1192583473
         txerrors: 0
        txdropped: 0
        timestamp: 11754.721178
```

History

Version	Description
2.00	The **show interface stat** command has been introduced.

3.119.36 show interface wps pin

Description Show the access point WPS PIN.

Prefix no No

Change settings No

Multiple input No

Interface type WiFi

Synopsis

(show)> **interface** *‹name›* **wps pin**

Arguments

Argument	Value	Description
name	*Interface name*	Full name or an alias of the interface.

Output

Element	Value
pin	Pin number.

Example

```
(show)> interface WifiMaster0/AccessPoint0 wps pin

                pin: 60180360
```

History

Version	Description
2.00	The **show interface wps pin** command has been introduced.

3.119.37 show interface wps status

Description Show the access point WPS status.

Prefix no No

Change settings No

Multiple input No

Interface type WiFi

Synopsis

(show)> **interface** *‹name›* **wps status**

Arguments

Argument	Value	Description
name	*Interface name*	Full name or an alias of the interface.

Output

Element	Value
wps	Root node.
configured	WPS is configured for Access Point.
auto-self-pin	Auto-self-pin mode state.
status	disabled enabled

Element	Value
	active
direction	send receive
mode	pbc self-pin peer
left	Time to session closure in seconds.

Example

```
(show)> interface WifiMaster0/AccessPoint0 wps status

             wps:
          configured: yes
       auto-self-pin: yes
              status: active
           direction: send
                mode: self-pin
                left: infinite
```

History

Version	Description
2.00	The **show interface wps status** command has been introduced.

3.119.38 show internet status

Description Check for an Internet connection on the device. The "Internet" LED (the globe) lights up as a result of connecting to popular internet sites.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **internet status**

Example

```
(show)> internet status

         checked: Tue Apr 24 17:14:37 2018
        reliable: yes
gateway-accessible: yes
   dns-accessible: yes
  host-accessible: yes
        internet: yes
```

```
 gateway:
           interface: GigabitEthernet1
             address: 192.168.1.1
            failures: 0
           accessible: yes
             excluded: no

    hosts:
                host:
                    name: example.net
                failures: 0
                resolved: yes
               accessible: yes

                host:
                    name: google.com
                failures: 0
                resolved: no
               accessible: no
```

History

Version	Description
2.11	The **show internet status** command has been introduced.

3.119.39 show ip arp

Description Display the contents of the *ARP* cache.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **ip arp**

Example

```
(show)> ip arp
================================================
IP                MAC                Interface
================================================
192.168.75.209    9c:b7:0d:91:e7:31  Home
82.135.72.150     00:0e:0c:09:db:60  ISP
192.168.75.106    88:53:2e:5e:07:1d  Home
192.168.75.201    7c:61:93:eb:6c:77  Home
192.168.75.203    00:19:d2:48:d6:dc  Home
10.10.30.34       a0:88:b4:40:9c:98  GuestWiFi
192.168.75.203    7c:61:93:ee:88:67  Home
192.168.75.211    00:26:c7:4a:e0:16  Home
82.138.72.163     34:51:c9:c6:53:cf  ISP
192.168.75.200    60:d8:19:cb:1b:36  Home
```

```
192.168.75.204    4c:0f:6e:4b:3c:ba   Home
82.138.72.129     00:30:48:89:b5:9f   ISP
```

History

Version	Description
2.00	The **show ip arp** command has been introduced.

3.119.40 show ip dhcp bindings

Description

Show *DHCP-server* status. If you use no argument, the entire list of issued IPs for all pools will be displayed.

Prefix no

No

Change settings

No

Multiple input

No

Synopsis

(show)> **ip dhcp bindings** [*‹pool›*]

Arguments

Argument	Value	Description
pool	*String*	The pool name.

Example

```
(show)> ip dhcp bindings _WEBADMIN

            lease:
                   ip: 192.168.15.211
                  mac: 00:26:c7:4a:e0:16
              expires: 289
             hostname: lenovo
            lease:
                   ip: 192.168.15.208
                  mac: 00:19:d2:48:d6:dc
              expires: 258
             hostname: evo
    ...
    ...
```

History

Version	Description
2.00	The **show ip dhcp bindings** command has been introduced.

3.119.41 show ip dhcp pool

Description

Show information about specified pool. If you use no argument, the information about all system pools will be displayed.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **ip dhcp pool** [*‹pool›*]

Arguments

Argument	Value	Description
pool	*String*	The pool name.

Example

```
(show)> ip dhcp pool 123

             pool, name = 123:
            interface, binding = auto:
              network: 0.0.0.0/0
                begin: 0.0.0.0
                  end: 0.0.0.0
               router, default = yes: 0.0.0.0
                lease, default = yes: 25200
                state: down
                debug: no
```

History

Version	Description
2.03	The **show ip dhcp pool** command has been introduced.

3.119.42 show ip hotspot

Description Show hotspot hosts.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **ip hotspot**

Example

```
(show)> ip hotspot

             host:
                  mac: 24:92:0e:92:e5:44
                  via: 24:92:0e:92:e5:44
                   ip: 192.168.1.41
             hostname: android-41d997d510af8ff9
                 name:

            interface:
                       id: Bridge0
```

```
                    name: Home
            description: Home network (Wired and wireless hosts)

               expires: 207328
            registered: no
                access: permit
              schedule:
                active: yes
               rxbytes: 0
               txbytes: 0
                uptime: 4911
                  link: up
                  ssid: Bewilderbeast
                    ap: WifiMaster0/AccessPoint0
       authenticated: yes
               txrate: 65
                    ht: 20
                  mode: 11n
                    gi: 800
                  rssi: -24
                   mcs: 7

             host:
                  mac: 20:aa:4b:5c:09:0e
                  via: 20:aa:4b:5c:09:0e
                   ip: 192.168.1.51
             hostname: Julia-PC
                 name:

            interface:
                      id: Bridge0
                    name: Home
            description: Home network (Wired and wireless hosts)

               expires: 212967
            registered: no
                access: permit
              schedule:
                active: yes
               rxbytes: 0
               txbytes: 0
                uptime: 884
                  link: up
                  ssid: Bewilderbeast
                    ap: WifiMaster0/AccessPoint0
       authenticated: yes
               txrate: 130
                    ht: 20
                  mode: 11n
                    gi: 800
                  rssi: -37
                   mcs: 15
```

History

Version	Description
2.09	The **show ip hotspot** command has been introduced.

3.119.43 show ip hotspot rrd

Description Show registered host traffic information of Round Robin Database.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **ip hotspot** *‹mac›* **rrd** *‹attribute›* [*‹detail›*]

Arguments

Argument	Value	Description
mac	*MAC-address*	MAC-address of registered host.
attribute	rxspeed	Data rate type.
	txspeed	
	rxbytes	
	txbytes	
detail	0	Level of detail is 1 second.
	1	Level of detail is 2 seconds.
	2	Level of detail is 3 seconds.
	3	Level of detail is 5 seconds.
	4	Level of detail is 15 seconds.
	5	Level of detail is 30 seconds.
	6	Level of detail is 1 minute.
	7	Level of detail is 2 minutes.
	8	Level of detail is 3 minutes.
	9	Level of detail is 5 minutes.
	10	Level of detail is 15 minutes.

Argument	Value	Description
	11	Level of detail is 30 minutes.

Example

```
(show)> ip hotspot a8:1e:84:85:f2:11 rrd rxspeed

            data:
                   t: 2180.491855
                   v: 16298

            data:
                   t: 2177.492050
                   v: 9026

            data:
                   t: 2174.491916
                   v: 11450

            data:
                   t: 2171.491843
                   v: 626
```

```
(show)> ip hotspot a8:1e:84:85:f2:11 rrd txspeed

            data:
                   t: 2228.491841
                   v: 952

            data:
                   t: 2225.491920
                   v: 8813

            data:
                   t: 2222.492053
                   v: 28746

            data:
                   t: 2219.491845
                   v: 22474
```

```
(show)> ip hotspot a8:1e:84:85:f2:11 rrd rxbytes

            data:
                   t: 2279.491860
                   v: 4197

            data:
                   t: 2276.492050
                   v: 362

            data:
                   t: 2273.492040
```

```
                    v: 14337

             data:
                    t: 2270.491862
                    v: 3281
```

```
(show)> ip hotspot a8:1e:84:85:f2:11 rrd txbytes

             data:
                    t: 2360.491865
                    v: 3342

             data:
                    t: 2357.491853
                    v: 142

             data:
                    t: 2354.491949
                    v: 3333

             data:
                    t: 2351.491847
                    v: 3390
```

History

Version	Description
2.14	The **show ip hotspot rrd** command has been introduced.

3.119.44 show ip hotspot summary

Description Show the information about traffic usage for several registered hosts according to Round Robin Database. Sorting is in descending order.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **ip hotspot summary** ‹*attribute*› [**detail** ‹*detail*›] [**count** ‹*count*›]

Arguments

Argument	Value	Description
`attribute`	`rxspeed`	Value of data rate type.
	`txspeed`	
	`rxbytes`	
	`txbytes`	
`detail`	`0`	Level of detail is 3 seconds.

Argument	Value	Description
	1	Level of detail is 60 seconds.
	2	Level of detail is 180 seconds.
	3	Level of detail is 1440 seconds.
`count`	*`Integer`*	The number of hosts. If not specified, the entire list of hosts is displayed.

Example

```
(show)> ip hotspot summary rxspeed

               t: 255

            host:
              active: yes
                name: toshiba
             rxspeed: 143964

            host:
              active: yes
                name: lnx
             rxspeed: 24749

            host:
              active: yes
                name: oneplus6
             rxspeed: 2558
```

```
(show)> ip hotspot summary rxspeed detail 0

               t: 0

            host:
              active: yes
                name: toshiba
             rxspeed: 186519

            host:
              active: yes
                name: oneplus6
             rxspeed: 94298

            host:
              active: yes
                name: lnx
             rxspeed: 8237
```

```
(show)> ip hotspot summary rxspeed count 3

               t: 255

            host:
              active: yes
```

```
                name: toshiba
             rxspeed: 390322

            host:
              active: yes
                name: lnx
             rxspeed: 53518

            host:
              active: yes
                name: oneplus6
             rxspeed: 5284
```

History

Version	Description
2.14	The **show ip hotspot summary** command has been introduced.

3.119.45 show ip http proxy

Description Show HTTP proxy status.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **ip http proxy**

Example

```
(show)> ip http proxy

           proxy:
                name: modem
              domain: myhomemodem.keenetic.link
            upstream: http://192.168.8.1:80
               allow: public
                ndns: yes
```

History

Version	Description
2.09	The **show ip http proxy** command has been introduced.

3.119.46 show ip name-server

Description Show a list of current addresses of DNS-servers in order of decreasing priority.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **ip name-server**

Example

```
(show)> ip name-server

           server:
              address: 9.9.9.9
                 port:
               domain:
               global: 0

           server:
              address: 1.0.0.1
                 port:
               domain: keenetic.net
               global: 0

          server:
              address: 1.1.1.1
                 port:
               domain:
               global: 64509
```

History

Version	Description
2.00	The **show ip name-server** command has been introduced.

3.119.47 show ip nat

Description Show network address translation table.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **ip nat** [**tcp**]

Arguments

Argument	Value	Description
tcp	*Keyword*	Only the records with *TCP* type will be displayed.

Example

```
(show)> ip nat
================================================================================
Type | In  | Source         Port  Destination    Port    Packets
     | Out |
```

```
=================================================================
udp         10.1.30.34      6482  111.221.77.159  40005   1
            111.221.77.159  40005 82.138.7.164    6482    1
-----------------------------------------------------------------
udp         220.27.130.179  6896  82.138.7.164    28197   1
            192.168.15.204  28197 220.27.130.179  6896    1
-----------------------------------------------------------------
tcp         10.1.30.33      57474 78.141.179.15   12350   12
            78.141.179.15   12350 82.138.7.164    57474   11
-----------------------------------------------------------------
udp         10.1.30.34      6482  84.201.228.162  44423   11
            84.201.228.162  44423 82.138.7.164    6482    16
-----------------------------------------------------------------
tcp         10.1.30.34      46655 96.55.147.21    443     2
            96.55.147.21    443   82.138.7.164    46655   0
-----------------------------------------------------------------
udp         10.1.30.34      6482  213.199.179.158 40006   1
            213.199.179.158 40006 82.138.7.164    6482    1
-----------------------------------------------------------------
```

History

Version	Description
2.00	The **show ip nat** command has been introduced.

3.119.48 show ip neighbour

Description Show the list of discovered hosts on the network at the OSI model network level.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **ip neighbour** [**alive**]

Arguments

Argument	Value	Description
alive	*Keyword*	Show active hosts.

Example

```
(show)> ip neighbour

        neighbour:
                   id: 1
                  via: b8:88:e1:2b:30:af
                  mac: b8:88:e1:2b:30:af
       address-family: ipv4
              address: 192.168.22.16
            interface: Bridge0
           first-seen: 251387
```

```
           last-seen: 0
           leasetime: 7372
             expired: no
            wireless: no

       neighbour:
                  id: 4
                 via: b8:88:e2:4b:30:af
                 mac: b8:88:e2:4b:30:af
      address-family: ipv6

           addresses:
                 address:
                     address: fe80::a022:a505:fae6:c891
                      status: active
                   last-seen: 3

           interface: Bridge0
          first-seen: 251371
           last-seen: 251371
           leasetime: 0
             expired: no
            wireless: no
```

History

Version	Description
2.10	The **show ip neighbour** command has been introduced.

3.119.49 show ip policy

Description Show the IP Policy profile status.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **ip policy** [*‹policy›*]

Arguments

Argument	Value	Description
policy	*Policy name*	Name of IP Policy profile.

Example

```
(show)> ip policy
policy, name = Policy0, description = VPN-OpenVPN:
                 mark: fffffd00
                table: 42

                route:
           destination: 10.1.30.0/24
```

```
                  gateway: 0.0.0.0
                interface: Guest
                   metric: 0
                    proto: boot
                 floating: no

                route:
             destination: 172.16.3.33/32
                  gateway: 0.0.0.0
                interface: L2TPVPN
                   metric: 0
                    proto: boot
                 floating: no

                route:
             destination: 192.168.1.0/24
                  gateway: 0.0.0.0
                interface: Home
                   metric: 0
                    proto: boot
                 floating: no

          policy, name = Policy3, description = Home:
                mark: fffffd03
               table: 45

                route:
             destination: 10.1.30.0/24
                  gateway: 0.0.0.0
                interface: Guest
                   metric: 0
                    proto: boot
                 floating: no

                route:
             destination: 172.16.3.33/32
                  gateway: 0.0.0.0
                interface: L2TPVPN
                   metric: 0
                    proto: boot
                 floating: no

                route:
             destination: 192.168.1.0/24
                  gateway: 0.0.0.0
                interface: Home
                   metric: 0
                    proto: boot
                 floating: no
```

```
(show)> ip policy Policy0
policy, name = Policy0:
                mark: fffffd00
               table: 42
```

```
  route:
destination: 0.0.0.0/0
    gateway: 193.0.174.1
  interface: ISP
     metric: 0
      proto: boot
   floating: no

  route:
destination: 10.1.30.0/24
    gateway: 0.0.0.0
  interface: Guest
     metric: 0
      proto: boot
   floating: no

  route:
destination: 185.230.127.84/32
    gateway: 193.0.174.1
  interface: ISP
     metric: 0
      proto: boot
   floating: no

  route:
destination: 192.168.1.0/24
    gateway: 0.0.0.0
  interface: Home
     metric: 0
      proto: boot
   floating: no

  route:
destination: 193.0.174.0/24
    gateway: 0.0.0.0
  interface: ISP
     metric: 0
      proto: boot
   floating: no

  route:
destination: 193.0.175.0/25
    gateway: 193.0.174.10
  interface: ISP
     metric: 0
      proto: boot
   floating: no

  route:
destination: 193.0.175.22/32
    gateway: 193.0.174.1
  interface: ISP
     metric: 0
```

```
                    proto: boot
                 floating: no
```

History

Version	Description
2.12	The **show ip policy** command has been introduced.

3.119.50 show ip route

Description Show the current routing table.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **ip route** [**sort** *‹criteria› ‹direction›*]

Arguments

Argument	Value	Description
direction	ascending	Routing table records are sorted in ascending order.
	descending	Routing table records are sorted in descending order.
criteria	interface	Sorting criteria is the interface name.
	gateway	Sorting criteria is the gateway address.
	destination	Sorting criteria is the destination address.

Example

```
(show)> ip route sort destination ascending
============================================================
Destination          Gateway           Interface     Metric
============================================================
0.0.0.0/0            82.138.7.129      ISP           0
10.1.30.0/24         0.0.0.0           GuestWiFi     0
82.138.7.27/32       0.0.0.0           PPTP0         0
82.138.7.32/32       0.0.0.0           PPTP0         0
82.138.7.128/26      0.0.0.0           ISP           0
82.138.7.132/32      82.138.7.129      ISP           0
82.138.7.141/32      82.138.7.129      ISP           0
89.179.183.128/26    82.138.7.138      ISP           0
192.168.15.0/24      0.0.0.0           Home          0
```

History

Version	Description
2.00	The **show ip route** command has been introduced.

3.119.51 show ipsec

Description Show info about *IPsec*/*IKE* strongSwan service status.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **ipsec**

Example

```
(show)> ipsec

  ipsec_statusall:

Status of IKE charon daemon (strongSwan 5.3.4, Linux 2.6.36, ►
mips):
  uptime: 6 days, since Dec 22 10:23:36 2015
  worker threads: 11 of 16 idle, 5/0/0/0 working, job queue: ►
0/0/0/0, scheduled: 10
  loaded plugins: charon aes des sha1 sha2 md5 random nonce ►
openssl xcbc cmac hmac attr kernel-netlink socket-default stroke ►
updown eap-mschapv2 eap-dynamic xauth-generic xauth-eap ►
error-notify systime-fix
Listening IP addresses:
  192.168.1.1
  10.10.10.15
Connections:
     test:  %any...ipsec.example.org  IKEv2, dpddelay=10s
     test:   local:  [ipsec.example.org] uses pre-shared key ►
authentication
     test:   remote: [ipsec.example.com] uses pre-shared key ►
authentication
    test:   child:  172.16.200.0/24 === 172.16.201.0/24 TUNNEL, ►
dpdaction=restart
Security Associations (1 up, 0 connecting):
     test[572]: ESTABLISHED 24 minutes ago, ►
10.10.10.15[ipsec.example.org]...10.10.10.20[ipsec.example.com]
     test[572]: IKEv2 SPIs: 00a6ebfc9d90f1c2_i* ►
3cd201ef496df75c_r, pre-shared key reauthentication in 20 minutes
     test[572]: IKE proposal: ►
AES_CBC=256/HMAC_SHA1_96/PRF_HMAC_SHA1/MODP_1024/#
    test{304}:  INSTALLED, TUNNEL, reqid 185, ESP in UDP SPIs: ►
ca59bfcf_i cde23d83_o
     test{304}:  AES_CBC_256/HMAC_SHA1_96, 10055 bytes_i (164 ►
pkts, 0s ago), 10786 bytes_o (139 pkts, 0s ago), rekeying in 34 ►
minutes
     test{304}:   172.16.200.0/24 === 172.16.201.0/24
```

History

Version	Description
2.06	The **show ipsec** command has been introduced.

3.119.52 show ipv6 addresses

Description Show a list of current IPv6-addresses.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **ipv6 addresses**

Example

```
(show)> ipv6 addresses

         address:
             address: 2001:db8::1
           interface: ISP
      valid-lifetime: infinite
         address:
             address: 2001:db8::ce5d:4eff:fe4f:aab2
           interface: Home
      valid-lifetime: infinite
         address:
             address: fd3c:4268:1559:0:ce5d:4eff:fe4f:aab2
           interface: Home
      valid-lifetime: infinite
         address:
             address: fd01:db8:43:0:ce5d:4eff:fe4f:aab2
           interface: Home
      valid-lifetime: infinite
```

History

Version	Description
2.00	The **show ipv6 addresses** command has been introduced.

3.119.53 show ipv6 prefixes

Description Show a list of current IPv6-prefixes.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **ipv6 prefixes**

Example

```
(show)> ipv6 prefixes

          prefix:
                prefix: 2001:db8::/64
             interface: ISP
         valid-lifetime: infinite
     preferred-lifetime: infinite
          prefix:
                prefix: fd3c:4268:1559::/48
             interface:
         valid-lifetime: infinite
     preferred-lifetime: infinite
          prefix:
                prefix: fd01:db8:43::/48
             interface:
         valid-lifetime: infinite
     preferred-lifetime: infinite
```

History

Version	Description
2.00	The **show ipv6 prefixes** command has been introduced.

3.119.54 show ipv6 routes

Description Show a list of current IPv6-routes.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **ipv6 routes**

Example

```
(show)> ipv6 routes

           route_:
          destination: 2001:db8::/64
              gateway: ::
            interface: Home
           route_:
          destination: fd3c:4268:1559::/64
              gateway: ::
            interface: Home
           route_:
          destination: fd01:db8:43::/64
              gateway: ::
            interface: Home
```

History

Version	Description
2.00	The **show ipv6 routes** command has been introduced.

3.119.55 show kabinet status

Description Check for the status and configuration of KABiNET authenticator.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **kabinet status**

Example

```
(show)> kabinet status

         kabinet:
             enabled: yes
                 wan: yes
               state: STOPPED
              server: 10.0.0.1
        access-level: internet
    protocol-version: 2
```

History

Version	Description
2.02	The **show kabinet status** command has been introduced.

3.119.56 show last-change

Description Show when and who made the latest changes in the settings.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **last-change**

Example

```
(show)> last-change

             date: Thu, 12 Jul 2012 10:01:47 GMT

            agent: cli
```

History

Version	Description
2.00	The **show last-change** command has been introduced.

3.119.57 show led

Description

Show information about specified LED in the system. If you use no argument, the entire list of all LEDs on the device will be displayed. Available LEDs depend on hardware configuration.

Prefix no

No

Change settings

No

Multiple input

No

Synopsis

(show)> **led** [*‹name›*]

Arguments

Argument	Value	Description
name	SYS	The LED name. The number of available indicators depends on the selected device.
	FN	
	FW_UPD	
	ACT_ACK	
	WAN	
	DSL	
	WLAN	
	WLAN5	
	WPS_1	
	WPS_2	
	WPS_3	
	WPS_4	
	WPS5_1	
	WPS5_2	
	WPS5_3	
	WPS5_4	
	USB_1	
	USB_2	
	LTE	

Example

```
(show)> led FN_1

              leds:
```

```
                led, index = 0:
                    name: FN_1
      user_configurable: yes
                virtual: no
```

History

Version	Description
2.05	The **show led** command has been introduced.

3.119.58 show led bindings

Description Show the control associated with the specified LED. If you use no argument, the entire list of all LEDs with theirs controls will be displayed.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **led** [*‹name›*]**bindings**

Arguments

Argument	Value	Description
name	SYS	The LED name. Set of available indicators depends on the selected device.
	FN	
	FW_UPD	
	ACT_ACK	
	WAN	
	DSL	
	WLAN	
	WLAN5	
	WPS_1	
	WPS_2	
	WPS_3	
	WPS_4	
	WPS5_1	
	WPS5_2	
	WPS5_3	
	WPS5_4	
	USB_1	
	USB_2	
	LTE	

Example

```
(show)> led bindings

        bindings:

         binding, index = 0:
                  led: SYS
    user_configurable: no
       active_control: SystemState
      default_control: SystemState

         binding, index = 1:
                  led: FN_1
    user_configurable: yes
       active_control: Usb1PortDeviceAttached
      default_control: Usb1PortDeviceAttached

         binding, index = 2:
                  led: FN_2
    user_configurable: yes
       active_control: Usb2PortDeviceAttached
      default_control: Usb2PortDeviceAttached

         binding, index = 3:
                  led: ACT_ACK
    user_configurable: no
       active_control: ButtonActivityAcknowledgement
      default_control: ButtonActivityAcknowledgement

         binding, index = 4:
                  led: FW_UPD
    user_configurable: no
       active_control:
      default_control:

         binding, index = 5:
                  led: WAN
    user_configurable: no
       active_control: WanConnected
      default_control: WanConnected

         binding, index = 6:
                  led: WLAN
    user_configurable: no
       active_control: WlanActivity
      default_control: WlanActivity

         binding, index = 7:
                  led: WPS_1
    user_configurable: no
       active_control: WlanWps1Activity
      default_control: WlanWps1Activity

         binding, index = 8:
                  led: WPS_2
```

```
    user_configurable: no
       active_control: WlanWps2Activity
      default_control: WlanWps2Activity

         binding, index = 9:
                  led: WPS_3
    user_configurable: no
       active_control: WlanWps3Activity
      default_control: WlanWps3Activity

         binding, index = 10:
                  led: WPS_4
    user_configurable: no
       active_control: WlanWps4Activity
      default_control: WlanWps4Activity

         binding, index = 11:
                  led: WPS_STA
    user_configurable: no
       active_control: WstaWpsActivity
      default_control: WstaWpsActivity

         binding, index = 12:
                  led: WLAN5
    user_configurable: no
       active_control: Wlan5Activity
      default_control: Wlan5Activity

         binding, index = 13:
                  led: WPS5_1
    user_configurable: no
       active_control: Wlan5Wps1Activity
      default_control: Wlan5Wps1Activity

         binding, index = 14:
                  led: WPS5_2
    user_configurable: no
       active_control: Wlan5Wps2Activity
      default_control: Wlan5Wps2Activity

         binding, index = 15:
                  led: WPS5_3
    user_configurable: no
       active_control: Wlan5Wps3Activity
      default_control: Wlan5Wps3Activity

         binding, index = 16:
                  led: WPS5_4
    user_configurable: no
       active_control: Wlan5Wps4Activity
      default_control: Wlan5Wps4Activity

         binding, index = 17:
                  led: WPS5_STA
```

```
    user_configurable: no
       active_control: Wsta5WpsActivity
      default_control: Wsta5WpsActivity
```

History

Version	Description
2.08	The **show led bindings** command has been introduced.

3.119.59 show led controls

Description Show a list of LED controls in the system. Available controls depend on hardware configuration.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **led controls**

Example

```
(show)> led controls

         controls:
              control, index = 0:
                       name: SystemState
         short_description: System state
                     owner: ndm
         user_configurable: no

              control, index = 1:
                       name: ButtonActivityAcknowledgement
         short_description: Button activity acknowledgement
                     owner: ndm
         user_configurable: no

              control, index = 2:
                       name: SelectedSchedule
         short_description: Selected schedule is active
                     owner: ndm
         user_configurable: yes

              control, index = 3:
                       name: SelectedWan
        short_description: Selected WAN interface has default ►
route
                     owner: ndm
         user_configurable: yes

              control, index = 4:
                       name: BackupWan
```

```
        short_description: Backup WAN interface has default route
                    owner: ndm
         user_configurable: yes

              control, index = 5:
                      name: WanConnected
         short_description: WAN interface connected
                    owner: ndm
         user_configurable: no

              control, index = 6:
                      name: Usb1PortDeviceAttached
         short_description: USB port 1 known device attached
                    owner: ndm
         user_configurable: yes

              control, index = 7:
                      name: Usb2PortDeviceAttached
         short_description: USB port 2 known device attached
                    owner: ndm
         user_configurable: yes

              control, index = 8:
                      name: UpdatesAvailable
         short_description: Firmware updates available
                    owner: ndm
         user_configurable: yes

              control, index = 9:
                      name: OpkgLedControl
         short_description: OPKG LED control
                    owner: ndm
         user_configurable: yes

              control, index = 10:
                      name: Wlan5Activity
         short_description: WLAN 5GHz interface activity
                    owner: mt7615_ap
         user_configurable: no

              control, index = 11:
                      name: Wlan5Wps1Activity
         short_description: WLAN 5GHz SSID 1 WPS activity
                    owner: mt7615_ap
         user_configurable: no

              control, index = 12:
                      name: Wlan5Wps2Activity
         short_description: WLAN 5GHz SSID 2 WPS activity
                    owner: mt7615_ap
         user_configurable: no

              control, index = 13:
                      name: Wlan5Wps3Activity
```

```
        short_description: WLAN 5GHz SSID 3 WPS activity
                    owner: mt7615_ap
        user_configurable: no

             control, index = 14:
                     name: Wlan5Wps4Activity
        short_description: WLAN 5GHz SSID 4 WPS activity
                    owner: mt7615_ap
        user_configurable: no

             control, index = 15:
                     name: WlanActivity
        short_description: WLAN 2.4GHz interface activity
                    owner: mt7615_ap
        user_configurable: no

             control, index = 16:
                     name: WlanWps1Activity
        short_description: WLAN 2.4GHz SSID 1 WPS activity
                    owner: mt7615_ap
        user_configurable: no

             control, index = 17:
                     name: WlanWps2Activity
        short_description: WLAN 2.4GHz SSID 2 WPS activity
                    owner: mt7615_ap
        user_configurable: no

             control, index = 18:
                     name: WlanWps3Activity
        short_description: WLAN 2.4GHz SSID 3 WPS activity
                    owner: mt7615_ap
        user_configurable: no

             control, index = 19:
                     name: WlanWps4Activity
        short_description: WLAN 2.4GHz SSID 4 WPS activity
                    owner: mt7615_ap
        user_configurable: no

             control, index = 20:
                     name: Wsta5WpsActivity
        short_description: Station 5GHz WPS activity
                    owner: mt7615_ap
        user_configurable: no

             control, index = 21:
                     name: WstaWpsActivity
        short_description: Station 2.4GHz WPS activity
                    owner: mt7615_ap
        user_configurable: no
```

History

Version	Description
2.08	The **show led controls** command has been introduced.

3.119.60 show log

Description

Show system log contents (records that are present in a circular buffer). The command executes in the background, that is, until forced to stop by the user pressing [Ctrl]+[C].

Prefix no

No

Change settings

No

Multiple input

No

Synopsis

(show)> **log** [‹*max-lines*›] [**once**]

Arguments

Argument	Value	Description
max-lines	*Integer*	Limit for returned log items.
once	*Keyword*	Show current log and exit to the CLI.

Example

```
(show)> log
═══════════════════════════════════════════════════════════════════════
  Time                      Message
═══════════════════════════════════════════════════════════════════════
I [Jul 12 12:08:39] radvd[228]: attempting to reread config file
I [Jul 12 12:08:39] radvd[228]: resuming normal operation
I [Jul 12 12:08:40] wmond: WifiMaster0/AccessPoint0: ►
STA(d8:b3:77:36:05:c1)
                    occurred MIC different in key handshaking.
I [Jul 12 12:08:40] radvd[228]: attempting to reread config file
I [Jul 12 12:08:40] radvd[228]: resuming normal operation
I [Jul 12 12:08:41] wmond: WifiMaster0/AccessPoint0: ►
STA(d8:b3:77:36:05:c1)
                    occurred MIC different in key handshaking.
I [Jul 12 12:08:41] radvd[228]: attempting to reread config file
I [Jul 12 12:08:41] radvd[228]: resuming normal operation
I [Jul 12 12:08:44] wmond: WifiMaster0/AccessPoint0: ►
STA(d8:b3:77:36:05:c1)
                    pairwise key handshaking timeout.
I [Jul 12 12:08:44] wmond: WifiMaster0/AccessPoint0: ►
STA(d8:b3:77:36:05:c1) had
                    deauthenticated.
```

History

Version	Description
2.00	The **show log** command has been introduced.

3.119.61 show mws associations

Description Show the list of Access Points on the repeater(s) associated with *MWS* controller.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **mws associations**

Example

```
(show)> mws associations

          station:
                  mac: 51:ef:22:11:17:1a
                   ap: WifiMaster1/Backhaul0
       authenticated: yes
              txrate: 585
              rxrate: 270
              uptime: 31
             txbytes: 33569
             rxbytes: 74324
                  ht: 80
                mode: 11ac
                  gi: 800
                rssi: -27
                 mcs: 7
                txss: 2
                 ebf: yes
                  mu: yes
```

History

Version	Description
3.01	The **show mws associations** command has been introduced.

3.119.62 show mws candidate

Description Show the list of candidates or the description of specified candidate by the given identifier.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **mws candidate** [*‹candidate›*]

Arguments

Argument	Value	Description
candidate	*String*	Device ID — MAC-address or CID.

Example

```
(show)> mws candidate 50:ff:20:08:71:61

         candidate:
                  mac: 50:ff:20:08:71:61
                  cid:
                 mode:
                model:
                state: DISCONNECTED
```

```
(show)> mws candidate 50:ff:20:08:71:61

         candidate:
                  mac: 50:ff:20:08:71:61
                  cid: ab1409a2-0f87-11e8-8f23-3d5f5921b253
                 mode: ap
                model: Extra (KN-1710)
                state: COMPATIBLE
                   fw: 2.15.A.4.0-1
         fw-available: 2.15.A.4.0-1
              license: 273720056272398
```

History

Version	Description
2.15	The **show mws candidate** command has been introduced.

3.119.63 show mws log

Description

Show log of connections and transitions from one Access Point to another within *MWS*. The command executes in the background, that is, until forced to stop by the user pressing [Ctrl]+[C].

Prefix no

No

Change settings

No

Multiple input

No

Synopsis

(show)> **mws log** [*‹max-lines›*] [**once**]

Arguments

Argument	Value	Description
max-lines	*Integer*	Limit of entries in the response.
once	*Keyword*	Show recent entries in the log.

Example

```
(show)> mws log 1
═══════════════════════════════════════════════════════════════════
 Time             Message
═══════════════════════════════════════════════════════════════════
 [Jan 17 15:04:58] : 64:a2:f9:51:b1:82: associated -> ►
50:ff:20:00:11:82 (5 GHz)
```

```
(show)> mws log once

═══════════════════════════════════════════════════════════════════
 Time             Message
═══════════════════════════════════════════════════════════════════
 [Jan 17 14:46:37] : 64:a2:f9:51:b1:82: associated -> ►
50:ff:20:00:11:82 (5 GHz)
 [Jan 17 15:04:50] : 64:a2:f9:51:b1:82: 50:ff:20:00:11:82 (5 ►
GHz) -> disassociated
 [Jan 17 15:04:58] : 64:a2:f9:51:b1:82: associated -> ►
50:ff:20:00:11:82 (5 GHz)
```

History

Version	Description
2.15	The **show mws log** command has been introduced.

3.119.64 show mws member

Description Show the list of members or the description of specified member by the given identifier.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **mws member** [*‹member›*]

Arguments

Argument	Value	Description
member	*String*	Device ID — MAC-address or CID.

Example

```
(show)> mws member ab1409a2-0f87-11e8-8f23-3d5f5921b253

            member:
                  cid: ab1409a2-0f87-11e8-8f23-3d5f5921b253
                model: Extra (KN-1710)
                  mac: 50:ff:20:08:7a:6a
                   ip: 192.168.1.43
                 mode: ap
                   fw: 2.15.A.4.0-1
        fw-available: 2.15.A.4.0-1
           dual-band: yes
```

```
            system:
               cpuload: 3
                memory: 32680/131072
                uptime: 2696

               rci:
                errors: 0
```

History

Version	Description
2.15	The **show mws member** command has been introduced.

3.119.65 show ndns

Description Show KeenDNS parameters from the latest request to the server (see **ndns get-booked** and **ndns get-update** commands).

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **ndns**

Example

```
(show)> ndns

            name: testname
          booked: testname
          domain: mykeenetic.com
         address: 41.189.34.56
         updated: yes
          access: direct

             ttp:
              direct: yes
           interface: GigabitEthernet1
             address: 41.189.34.56
```

History

Version	Description
2.07	The **show ndns** command has been introduced.

3.119.66 show netfilter

Description Show information about the firewall working. Need to provide remote technical support.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **netfilter**

History

Version	Description
2.00	The **show netfilter** command has been introduced.

3.119.67 show ntce applications

Description Show the list of applications supported by the *NTCE* service.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **ntce applications**

Example

```
(show)> ntce applications

     application:
               id-num: 1
                short: facebook
                 long: Facebook
            group-id: 2065
          group-long: Social
         groupset-id: 4
    groupset-short-id: surfing
     groupset-long-id: Web surfing

     application:
               id-num: 2
                short: magicjack
                 long: magicJack
            group-id: 2054
          group-long: Voice over IP
         groupset-id: 0
    groupset-short-id: calling
     groupset-long-id: Calling and conferencing

     application:
               id-num: 3
                short: itunes
                 long: iTunes
            group-id: 2056
          group-long: Streaming
         groupset-id: 2
    groupset-short-id: streaming
```

```
     groupset-long-id: Video & Audio streaming

     application:
               id-num: 4
                short: myspace
                 long: MySpace
            group-id: 2065
          group-long: Social
         groupset-id: 4
    groupset-short-id: surfing
     groupset-long-id: Web surfing

     application:
               id-num: 5
                short: facetime
                 long: FaceTime
            group-id: 2054
          group-long: Voice over IP
         groupset-id: 0
    groupset-short-id: calling
     groupset-long-id: Calling and conferencing

     application:
               id-num: 6
                short: truphone
                 long: Truphone
            group-id: 2054
          group-long: Voice over IP
         groupset-id: 0
    groupset-short-id: calling
     groupset-long-id: Calling and conferencing

     application:
               id-num: 7
                short: twitter
                 long: Twitter
            group-id: 2065
          group-long: Social
         groupset-id: 4
    groupset-short-id: surfing
     groupset-long-id: Web surfing

     application:
               id-num: 8
                short: xbox
                 long: XBOX gaming console
            group-id: 2050
          group-long: Gaming
         groupset-id: 1
    groupset-short-id: gaming
     groupset-long-id: Gaming

     application:
               id-num: 9
```

```
                short: realmedia
                 long: RealMedia
            group-id: 2088
          group-long: Removed
         groupset-id: 5
    groupset-short-id: other
     groupset-long-id: Other

     application:
              id-num: 10
               short: google-mail
                long: Google Mail
           group-id: 2059
         group-long: Mail
        groupset-id: 3
    groupset-short-id: work
     groupset-long-id: Work & Learn from home
```

History

Version	Description
3.07	The **show ntce applications** command has been introduced.

3.119.68 show ntce attributes

Description Show the list of attributes supported by the *NTCE* service.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **ntce attributes**

Example

```
(show)> ntce attributes

        attribute:
               id-num: 1
                short: encrypted
                long: Indicates that the current connection is ►
encrypted traffic.

        attribute:
               id-num: 2
                short: audio
                long: Indicates that the current connection is ►
an audio or voice signal.

        attribute:
               id-num: 3
```

```
               short: out
               long: Indicates that the current connection is ►
a landline call, e.g. a call to a home phone.

        attribute:
               id-num: 4
               short: video
               long: Indicates that the current connection is ►
a video signal.

        attribute:
               id-num: 5
               short: file-transfer
               long: Indicates that the current connection is ►
a file transfer.

        attribute:
               id-num: 6
               short: web
               long: Indicates that the current connection is ►
a surf the Internet session.

        attribute:
               id-num: 7
               short: chat
               long: Indicates that the current connection is ►
a chat session.

        attribute:
               id-num: 8
               short: mail
               long: Indicates that the current connection is ►
mail traffic.

        attribute:
               id-num: 9
               short: stream
               long: Indicates that the current connection is ►
a continues unidirectional stream of audio and / or video.

        attribute:
               id-num: 10
               short: android
               long: Indicates that the client side uses the ►
operating system Android.

        attribute:
               id-num: 11
               short: ios
               long: Indicates that the client side uses the ►
operating system iOS.

        attribute:
               id-num: 12
```

```
                short: windows-mobile
                 long: Indicates that the client side uses the ►
operating system Windows Mobile.

        attribute:
               id-num: 13
                short: blackberry
                 long: Indicates that the client side uses the ►
operating system Blackberry.

        attribute:
               id-num: 14
                short: picture
                 long: Indicates that the current connection ►
transfers pictures.

        attribute:
               id-num: 15
                short: ddl
               long: Indicates that the current connection is ►
a Direct Download Hoster.

        attribute:
               id-num: 16
                short: google
               long: Indicates that the current connection is ►
a Google service.

        attribute:
               id-num: 17
                short: outlook_web_access
                 long: Indicates that the current connection ►
uses the Microsoft Exchange Outlook Web Access as authentication ►
mechanism.

        attribute:
               id-num: 18
                short: amazon-cloud
               long: Indicates that the current connection is ►
a service of Amazon Cloud.

        attribute:
               id-num: 19
                short: apache
                 long: Indicates that the server side is an ►
Apache server.

        attribute:
               id-num: 20
                short: mysql-server
               long: Indicates that the server side is a MySQL ►
database server.

        attribute:
```

```
               id-num: 21
                short: mariadb-server
                 long: Indicates that the server side is a ►
MariaDB database server.

        attribute:
               id-num: 22
                short: ntlm
                 long: Current connection uses NTLM as ►
authentication mechanism.

        attribute:
               id-num: 23
                short: microsoft-windows
                 long: Indicates that the client side is the ►
operating system Microsoft Windows.

        attribute:
               id-num: 24
                short: chrome
                 long: Indicates that the client side is the ►
operating system Chrome.

        attribute:
               id-num: 25
                short: akamai-cloud
                long: Indicates that the current connection is ►
a service of Akamai Cloud.

        attribute:
               id-num: 26
                short: dox
                long: Indicates that the current connection is ►
DoT (DNS over TLS) or DoH (DNS over HTTPS).

        attribute:
               id-num: 27
                short: rcs
                long: Indicates that the current connection is ►
RCS (Rich Communication Services).
```

History

Version	Description
3.07	The **show ntce attributes** command has been introduced.

3.119.69 show ntce groups

Description Show the list of groups supported by the *NTCE* service.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **ntce groups**

Example

```
(show)> ntce groups

            group:
                id-num: 2048
                  long: Generic
           groupset-id: 5
     groupset-short-id: other
      groupset-long-id: Other

            group:
                id-num: 2049
                  long: Peer to Peer
           groupset-id: 6
     groupset-short-id: filetransfering
      groupset-long-id: File transfering

            group:
                id-num: 2050
                  long: Gaming
           groupset-id: 1
     groupset-short-id: gaming
      groupset-long-id: Gaming

            group:
                id-num: 2051
                  long: Tunnel
           groupset-id: 3
     groupset-short-id: work
      groupset-long-id: Work & Learn from home

            group:
                id-num: 2052
                  long: Business
           groupset-id: 3
     groupset-short-id: work
      groupset-long-id: Work & Learn from home

            group:
                id-num: 2053
                  long: E-Commerce
           groupset-id: 3
     groupset-short-id: work
      groupset-long-id: Work & Learn from home

            group:
                id-num: 2054
                  long: Voice over IP
```

```
          groupset-id: 0
    groupset-short-id: calling
     groupset-long-id: Calling and conferencing

           group:
               id-num: 2055
                 long: Messaging
          groupset-id: 0
    groupset-short-id: calling
     groupset-long-id: Calling and conferencing

           group:
               id-num: 2056
                 long: Streaming
          groupset-id: 2
    groupset-short-id: streaming
     groupset-long-id: Video & Audio streaming

           group:
               id-num: 2057
                 long: Mobile
          groupset-id: 0
    groupset-short-id: calling
     groupset-long-id: Calling and conferencing

           group:
               id-num: 2058
                 long: Remote Control
          groupset-id: 3
    groupset-short-id: work
     groupset-long-id: Work & Learn from home

           group:
               id-num: 2059
                 long: Mail
          groupset-id: 3
    groupset-short-id: work
     groupset-long-id: Work & Learn from home

           group:
               id-num: 2060
                 long: Network Management
          groupset-id: 5
    groupset-short-id: other
     groupset-long-id: Other

           group:
               id-num: 2061
                 long: Database
          groupset-id: 3
    groupset-short-id: work
     groupset-long-id: Work & Learn from home

           group:
```

```
                id-num: 2062
                  long: Filetransfer
          groupset-id: 6
    groupset-short-id: filetransfering
     groupset-long-id: File transfering

           group:
                id-num: 2063
                  long: Web
          groupset-id: 4
    groupset-short-id: surfing
     groupset-long-id: Web surfing

           group:
                id-num: 2064
                  long: Conference
          groupset-id: 0
    groupset-short-id: calling
     groupset-long-id: Calling and conferencing

           group:
                id-num: 2065
                  long: Social
          groupset-id: 4
    groupset-short-id: surfing
     groupset-long-id: Web surfing

           group:
                id-num: 2066
                  long: Sharehosting
          groupset-id: 6
    groupset-short-id: filetransfering
     groupset-long-id: File transfering

           group:
                id-num: 2067
                  long: Deprecated
          groupset-id: 5
    groupset-short-id: other
     groupset-long-id: Other

           group:
                id-num: 2068
                  long: Industrial
          groupset-id: 5
    groupset-short-id: other
     groupset-long-id: Other

           group:
                id-num: 2069
                  long: Encrypted
          groupset-id: 5
    groupset-short-id: other
     groupset-long-id: Other
```

```
        group:
            id-num: 2070
              long: Advertisement and Analytic Services
      groupset-id: 5
groupset-short-id: other
 groupset-long-id: Other

        group:
            id-num: 2071
              long: News
      groupset-id: 4
groupset-short-id: surfing
 groupset-long-id: Web surfing

        group:
            id-num: 2072
              long: Health and Fitness
      groupset-id: 5
groupset-short-id: other
 groupset-long-id: Other

        group:
            id-num: 2073
              long: Cloud and CDN Services
      groupset-id: 5
groupset-short-id: other
 groupset-long-id: Other

        group:
            id-num: 2074
              long: Navigation
      groupset-id: 4
groupset-short-id: surfing
 groupset-long-id: Web surfing

        group:
            id-num: 2075
              long: Finance
      groupset-id: 5
groupset-short-id: other
 groupset-long-id: Other

        group:
            id-num: 2076
              long: Travel and Transportation
      groupset-id: 5
groupset-short-id: other
 groupset-long-id: Other

        group:
            id-num: 2077
              long: Pornography
      groupset-id: 5
```

```
    groupset-short-id: other
     groupset-long-id: Other

           group:
               id-num: 2078
                 long: Books and Magazines
          groupset-id: 5
    groupset-short-id: other
     groupset-long-id: Other

           group:
               id-num: 2079
                 long: Audio Entertainment
          groupset-id: 2
    groupset-short-id: streaming
     groupset-long-id: Video & Audio streaming

           group:
               id-num: 2080
                 long: Education
          groupset-id: 5
    groupset-short-id: other
     groupset-long-id: Other

           group:
               id-num: 2081
                 long: M2M and IoT
          groupset-id: 3
    groupset-short-id: work
     groupset-long-id: Work & Learn from home

           group:
               id-num: 2082
                 long: Device Security
          groupset-id: 4
    groupset-short-id: surfing
     groupset-long-id: Web surfing

           group:
               id-num: 2083
                 long: Multimedia Service Providers
          groupset-id: 2
    groupset-short-id: streaming
     groupset-long-id: Video & Audio streaming

           group:
               id-num: 2084
                 long: Organizers
          groupset-id: 3
    groupset-short-id: work
     groupset-long-id: Work & Learn from home

           group:
               id-num: 2085
```

```
                long: Enterprise Services
           groupset-id: 4
     groupset-short-id: surfing
      groupset-long-id: Web surfing

            group:
                id-num: 2086
                  long: App-Stores and OS Updates
           groupset-id: 6
     groupset-short-id: filetransfering
      groupset-long-id: File transfering

            group:
                id-num: 2087
                  long: Browsers
           groupset-id: 4
     groupset-short-id: surfing
      groupset-long-id: Web surfing

            group:
                id-num: 2088
                  long: Removed
           groupset-id: 5
     groupset-short-id: other
      groupset-long-id: Other

            group:
                id-num: 2089
                  long: Moved
           groupset-id: 5
     groupset-short-id: other
      groupset-long-id: Other
```

History

Version	Description
3.07	The **show ntce groups** command has been introduced.

3.119.70 show ntce groupsets

Description Show the list of groupsets supported by the *NTCE* service.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **ntce groupsets**

Example

```
(show)> ntce groupsets
```

```
        groupset:
              id-num: 0
               short: calling
                long: Calling and conferencing

        groupset:
              id-num: 1
               short: gaming
                long: Gaming

        groupset:
              id-num: 2
               short: streaming
                long: Video & Audio streaming

        groupset:
              id-num: 3
               short: work
                long: Work & Learn from home

        groupset:
              id-num: 4
               short: surfing
                long: Web surfing

        groupset:
              id-num: 5
               short: other
                long: Other

        groupset:
              id-num: 6
               short: filetransfering
                long: File transfering
```

History

Version	Description
3.07	The **show ntce groupsets** command has been introduced.

3.119.71 show ntce hosts

Description Show application statistics, which *NTCE* service has detected for hosts.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **ntce hosts**

Example

```
(show)> ntce hosts

            host:
                 mac: 04:d4:c4:54:31:12

          application:
                         id-num: 7
                          short: twitter
                           long: Twitter
                       group-id: 2065
                     group-long: Social
                    groupset-id: 4
                groupset-short-id: surfing
                 groupset-long-id: Web surfing
              groupset-service-class: 2
                        rxbytes: 62274
                        txbytes: 6020

          application:
                         id-num: 43
                          short: instagram
                           long: Instagram
                       group-id: 2065
                     group-long: Social
                    groupset-id: 4
                groupset-short-id: surfing
                 groupset-long-id: Web surfing
              groupset-service-class: 2
                        rxbytes: 57606
                        txbytes: 11148

          application:
                         id-num: 428
                          short: spotify
                           long: Spotify
                       group-id: 2079
                     group-long: Audio Entertainment
                    groupset-id: 2
                groupset-short-id: streaming
                 groupset-long-id: Video & Audio streaming
              groupset-service-class: 2
                        rxbytes: 155317
                        txbytes: 80526

          application:
                         id-num: 438
                          short: whatsapp
                           long: WhatsApp
                       group-id: 2055
                     group-long: Messaging
                    groupset-id: 0
                groupset-short-id: calling
                 groupset-long-id: Calling and conferencing
              groupset-service-class: 2
```

```
                        rxbytes: 826
                        txbytes: 706

         application:
                         id-num: 461
                          short: google-cloud
                           long: Google Cloud
                       group-id: 2073
                     group-long: Cloud and CDN Services
                    groupset-id: 5
                groupset-short-id: other
                 groupset-long-id: Other
               groupset-service-class: 2
                        rxbytes: 313
                        txbytes: 352

         application:
                         id-num: 498
                          short: telegram
                           long: Telegram
                       group-id: 2055
                     group-long: Messaging
                    groupset-id: 0
                groupset-short-id: calling
                 groupset-long-id: Calling and conferencing
               groupset-service-class: 2
                        rxbytes: 109895
                        txbytes: 15561

         application:
                         id-num: 559
                          short: google-play
                           long: Google Play
                       group-id: 2086
                     group-long: App-Stores and OS Updates
                    groupset-id: 6
                groupset-short-id: filetransfering
                 groupset-long-id: File transfering
               groupset-service-class: 2
                        rxbytes: 16736
                        txbytes: 28451

         application:
                         id-num: 590
                          short: yandex
                           long: Yandex
                       group-id: 2085
                     group-long: Enterprise Services
                    groupset-id: 4
                groupset-short-id: surfing
                 groupset-long-id: Web surfing
               groupset-service-class: 2
                        rxbytes: 606
                        txbytes: 200
```

```
 application:
                id-num: 611
                 short: zendesk
                  long: ZenDesk
             group-id: 2052
           group-long: Business
          groupset-id: 3
      groupset-short-id: work
       groupset-long-id: Work & Learn from home
  groupset-service-class: 2
              rxbytes: 101697
              txbytes: 187527

 application:
                id-num: 621
                 short: slack
                  long: Slack
             group-id: 2064
           group-long: Conference
          groupset-id: 0
      groupset-short-id: calling
       groupset-long-id: Calling and conferencing
  groupset-service-class: 2
              rxbytes: 30568
              txbytes: 3650

 application:
                id-num: 632
                 short: google-services
                  long: Google Shared Services
             group-id: 2085
           group-long: Enterprise Services
          groupset-id: 4
      groupset-short-id: surfing
       groupset-long-id: Web surfing
  groupset-service-class: 2
              rxbytes: 614512
              txbytes: 202174

 application:
                id-num: 664
                 short: microsoft-services
                  long: Microsoft Services
             group-id: 2085
           group-long: Enterprise Services
          groupset-id: 4
      groupset-short-id: surfing
       groupset-long-id: Web surfing
  groupset-service-class: 2
              rxbytes: 20243
              txbytes: 10699

 application:
```

```
                        id-num: 700
                         short: fastly
                          long: Fastly
                      group-id: 2073
                    group-long: Cloud and CDN Services
                   groupset-id: 5
               groupset-short-id: other
                groupset-long-id: Other
        groupset-service-class: 2
                     rxbytes: 14859
                     txbytes: 3147

         application:
                      id-num: 703
                       short: cloudflare
                        long: Cloudflare
                    group-id: 2073
                  group-long: Cloud and CDN Services
                 groupset-id: 5
             groupset-short-id: other
              groupset-long-id: Other
        groupset-service-class: 2
                     rxbytes: 2172
                     txbytes: 3593

         application:
                      id-num: 719
                       short: google-apis
                        long: Google APIs
                    group-id: 2052
                  group-long: Business
                 groupset-id: 3
             groupset-short-id: work
              groupset-long-id: Work & Learn from home
        groupset-service-class: 2
                     rxbytes: 11837
                     txbytes: 7602

         application:
                      id-num: 933
                       short: bamtech-media
                        long: BAMTech Media
                    group-id: 2083
                  group-long: Multimedia Service Providers
                 groupset-id: 2
             groupset-short-id: streaming
              groupset-long-id: Video & Audio streaming
        groupset-service-class: 2
                     rxbytes: 4734
                     txbytes: 6006

         application:
                      id-num: 1136
                       short: cloud-mail-ru
```

```
                        long: Cloud-Mail-Ru
                    group-id: 2062
                  group-long: Filetransfer
                 groupset-id: 6
           groupset-short-id: filetransfering
            groupset-long-id: File transfering
      groupset-service-class: 2
                    rxbytes: 61161
                    txbytes: 86671

       application:
                     id-num: 1281
                      short: kaspersky-services
                       long: Kaspersky Services
                    group-id: 2082
                  group-long: Device Security
                 groupset-id: 4
           groupset-short-id: surfing
            groupset-long-id: Web surfing
      groupset-service-class: 2
                    rxbytes: 40
                    txbytes: 70

             os-id: 3
           os-long: Windows

              host:
                   mac: 04:d4:c4:54:31:12
                   via: 04:d4:c4:54:31:12
                    ip: 192.168.11.19
              hostname: MyHost
                  name: MyHost

             interface:
                        id: Bridge0
                      name: Home
               description: Home network

                  dhcp:
                    static: yes

            registered: yes
                access: permit
              schedule:
                active: yes
               rxbytes: 0
               txbytes: 0
                uptime: 9083
            first-seen: 9097
             last-seen: 1
                  link: up
      auto-negotiation: yes
                 speed: 1000
                duplex: yes
```

```
                    port: 2

            traffic-shape:
                          rx: 0
                          tx: 0
                        mode: mac
                    schedule:
```

History

Version	Description
3.07	The **show ntce hosts** command has been introduced.

3.119.72 show ntce oses

Description Show the list of OSes supported by the *NTCE* service.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **ntce oses**

Example

```
(show)> ntce oses

                os:
                id-num: 1
                  long: Not detected

                os:
                id-num: 2
                  long: Other

                os:
                id-num: 3
                  long: Windows

                os:
                id-num: 4
                  long: Linux

                os:
                id-num: 5
                  long: OS X

                os:
                id-num: 6
                  long: iOS

                os:
```

```
              id-num: 7
                long: Symbian

              os:
              id-num: 8
                long: Android

              os:
              id-num: 9
                long: Blackberry

              os:
              id-num: 10
                long: WindowsMobile

              os:
              id-num: 11
                long: WindowsPhone

              os:
              id-num: 12
                long: Chrome

              os:
              id-num: 13
                long: Darwin
```

History

Version	Description
3.07	The **show ntce oses** command has been introduced.

3.119.73 show ntce status

Description Show *NTCE* service info.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **ntce status**

Example

```
(show)> ntce status

       conntrack:
                  hosts: 2
              applications: 16
     applications-flows: 63
    applications-events: 0
                 groups: 12
```

```
           groups-flows: 64
          groups-events: 0

                 memory:
         applications-flows: 1512
        applications-events: 0
               applications: 512
               groups-flows: 1536
              groups-events: 0
                     groups: 384
                      hosts: 72
                      total: 4016

           event:
               count: 0

              memory:
                   total: 0

        database:
               hosts: 1
        applications: 54
              groups: 30
          attributes: 6

              memory:
            applications: 2372976
                 groups: 1318320
              attributes: 263664
                   total: 3954960
```

History

Version	Description
3.07	The **show ntce status** command has been introduced.

3.119.74 show ntp status

Description

Show *NTP* system settings.

NTP state general info

❶ The time elapsed since the last synchronization in seconds.
❷ The indicator of the last synchronization.
❸ The indicator of the initial synchronization.
❹ Time is taken from NDSS server.
❺ Time is set by the user manually.

Prefix no

No

Change settings

No

Multiple input

No

Synopsis

(show)> **ntp status**

Example

```
(show)> ntp status

          status:
             elapsed: 435146 ❶
              server: 1.pool.ntp.org
            accurate: yes ❷
        synchronized: yes ❸
            ndsstime: no ❹
            usertime: no ❺
```

History

Version	Description
2.00	The **show ntp status** command has been introduced.

3.119.75 show nvox call-history

Description Show list of calls registered since the router is switched on.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **nvox call-history**

Example

```
(show)> nvox call-history

   call_history:
           revision: 13

               call:
                     type: missed
                     index: 0
               start_time: Thu Sep 14 12:13:23 2017
                     line: SIPLab1
                       hs: KX-TPA60
       other_party_number: 3254
         other_party_name:
                 duration:
            release_code:
          release_reason: rejected

               call:
                     type: accepted
                     index: 1
               start_time: Thu Sep 14 12:13:32 2017
                     line: SIPLab1
                       hs: Gigaset A540CAT
```

```
       other_party_number: 3254
         other_party_name:
                 duration: 3
             release_code:
           release_reason:

                 call:
                       type: internal
                       index: 2
                 start_time: Thu Sep 14 12:13:51 2017
                       line: intercom
                         hs: Gigaset A540CAT
       other_party_number: hs1
         other_party_name: KX-TGA250
                 duration: 3
             release_code:
           release_reason:

                 call:
                       type: internal
                       index: 3
                 start_time: Thu Sep 14 12:14:07 2017
                       line: intercom
                         hs: Gigaset A540CAT
       other_party_number: hs2
         other_party_name: KX-TPA60
                 duration: 2
             release_code:
           release_reason:

                 call:
                       type: internal
                       index: 4
                 start_time: Thu Sep 14 12:14:24 2017
                       line: intercom
                         hs: Gigaset A540CAT
       other_party_number: hs*
         other_party_name:
                 duration: 0
             release_code:
           release_reason:

                 call:
                       type: internal
                       index: 5
                 start_time: Thu Sep 14 12:14:42 2017
                       line: intercom
                         hs: Gigaset A540CAT
       other_party_number: hs2
         other_party_name: KX-TPA60
                 duration: 0
             release_code:
           release_reason:
```

```
               call:
                     type: outgoing
                     index: 6
               start_time: Thu Sep 14 12:15:44 2017
                     line: Data Group
                       hs: Gigaset A540CAT
       other_party_number: 0443647362
         other_party_name:
                 duration: 0
             release_code:
           release_reason:

               call:
                     type: missed
                     index: 7
               start_time: Thu Sep 14 12:15:44 2017
                     line: Data Group
                       hs:
       other_party_number: 3647362
         other_party_name:
                 duration:
             release_code:
           release_reason:

               call:
                     type: forwarded
                     index: 8
               start_time: Thu Sep 14 12:17:30 2017
                     line: Data Group
                       hs:
       other_party_number: 3647362
         other_party_name:
                 duration:
             release_code: 61773
           release_reason: 0687852828

               call:
                     type: outgoing
                     index: 9
               start_time: Thu Sep 14 12:17:30 2017
                     line: Data Group
                       hs: Panasonic KX-TPA60
       other_party_number: 0443647362
         other_party_name:
                 duration: 0
             release_code: 480
           release_reason: Temporarily Not Available
```

History

Version	Description
2.06	The **show dect call-history** command has been introduced.
3.05	The command renamed to **show nvox call-history**.

3.119.76 show ping-check

Description Show *Ping Check* profile status. If you use no arguments, the command displays information about all profiles.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **ping-check** [*‹profile_name›*]

Arguments

Argument	Value	Description
profile_name	*String*	Profile name.

Example

```
(show)> ping-check

        pingcheck:
              profile: TEST
                 host: 8.8.8.8
                 port: 80
            max-fails: 7
              timeout: 1
                 mode: connect

            interface: ISP
               fail count: 0
                   status: pass

        pingcheck:
              profile: TEST1
                 mode: icmp

        pingcheck:
              profile: TEST2
                 mode: icmp
```

History

Version	Description
2.04	The **show ping-check** command has been introduced.

3.119.77 show ppe

Description Show Packet Processing Engine status.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **ppe**

Example

```
(show)> ppe

hw_nat:

Total Entry Count = 2
IPv4_NAPT=1122 : 13.33.96.244:443->10.77.140.59:56457 => ►
13.33.96.244:443->192.168.232.44:56457
IPv4_NAPT=5454 : 173.194.220.97:443->10.77.140.59:56553 => ►
173.194.220.97:443->192.168.232.44:56553
done
```

History

Version	Description
2.03	The **show ppe** command has been introduced.

3.119.78 show processes

Description Show statistics of CPU usage by services and processess.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **processes**

Example

```
(show)> processes

          process, id = NETBIOS browser:
                 name: nqnd

                  arg: -i

                  arg: 50ff20001e87

                state: S (sleeping)
                  pid: 629
                 ppid: 192
              vm-size: 3188 kB
              vm-data: 1548 kB
               vm-stk: 136 kB
               vm-exe: 4 kB
               vm-lib: 1448 kB
              vm-swap: 0 kB
              threads: 1
                  fds: 15
```

```
          statistics:
                interval: 30

                     cpu:
                         now: 17319.483753
                         min: 0
                         max: 0
                         avg: 0
                         cur: 0

             service:
              configured: yes
                   alive: yes
                 started: yes
                   state: STARTED

        process, id = Dns::Proxy::Policy0:
                name: ndnproxy

                 arg: -c

                 arg: /var/ndnproxy_Policy0.conf

                 arg: -p

                 arg: /var/ndnproxy_Policy0.pid

               state: S (sleeping)
                 pid: 630
                ppid: 192
             vm-size: 1676 kB
             vm-data: 504 kB
              vm-stk: 136 kB
              vm-exe: 108 kB
              vm-lib: 896 kB
             vm-swap: 0 kB
             threads: 1
                 fds: 10

          statistics:
                interval: 30

                     cpu:
                         now: 17319.483764
                         min: 0
                         max: 0
                         avg: 0
                         cur: 0

             service:
              configured: yes
                   alive: yes
```

```
                  started: yes
                    state: STARTED
```

History

Version	Description
2.09	The **show processes** command has been introduced.

3.119.79 show running-config

Description Show current settings, that is file `system:running-config` conteins, just like command **more** does.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **running-config**

Example

```
(show)> running-config
! $$$ Model: Keenetic Start
! $$$ Version: 2.06.1
! $$$ Agent: http/rci
! $$$ Last change: Fri, 12 Jan 2017 07:23:56 GMT
system
    set net.ipv4.ip_forward 1
    set net.ipv4.netfilter.ip_conntrack_max 4096
   set net.ipv4.netfilter.ip_conntrack_tcp_timeout_established ►
1200
    set net.ipv4.netfilter.ip_conntrack_udp_timeout 60
    set net.ipv4.tcp_fin_timeout 30
    set net.ipv4.tcp_keepalive_time 120
    set net.ipv6.conf.all.forwarding 1
    hostname Keenetic
    domainname WORKGROUP
!
ntp server 0.pool.ntp.org
ntp server 1.pool.ntp.org
ntp server 2.pool.ntp.org
ntp server 3.pool.ntp.org
access-list _WEBADMIN_GuestWiFi
    deny tcp 0.0.0.0 0.0.0.0 10.1.30.1 255.255.255.255
!
access-list _WEBADMIN_ISP
    permit tcp 0.0.0.0 0.0.0.0 192.168.15.200 255.255.255.255 ►
port eq 3389
    permit icmp 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0
!
isolate-private
dyndns profile _ABCD
```

```
!
dyndns profile _WEBADMIN
    type dyndns
!
interface FastEthernet0
    up
!
interface FastEthernet0/0
    switchport mode access
    switchport access vlan 1
!
interface FastEthernet0/1
    switchport mode access
    switchport access vlan 1
!
interface Bridge0
    name Home
    description "Home network"
    inherit FastEthernet0/Vlan1
    include AccessPoint
    security-level private
    ip address 192.168.15.43 255.255.255.0
    up
!
interface WiMax0
    description Yota
    security-level public
    ip address auto
    ip global 400
    up
!
interface PPTP0
    description "Office VPN"
    peer crypton.example.net
    lcp echo 30 3
    ipcp default-route
    ipcp name-servers
    ccp
    security-level public
    authentication identity "00441"
    authentication password 123456
    authentication mschap
    authentication mschap-v2
    encryption mppe
    ip tcp adjust-mss pmtu
    connect via ISP
    up
!
ip route 82.138.7.141 ISP auto
ip route 82.138.7.132 ISP auto
ip route 82.138.7.27 PPTP0 auto
ip dhcp pool _WEBADMIN
    range 192.168.15.200 192.168.15.219
    bind Home
```

```
!
ip dhcp pool _WEBADMIN_GUEST_AP
    range 10.1.30.33 10.1.30.52
    bind GuestWiFi
!
ip dhcp host A 00:01:02:03:04:05 1.1.1.1
ip dhcp host B 00:01:02:03:04:06 1.1.1.2
ip nat Home
ip nat GuestWiFi
ipv6 subnet Default
    bind Home
    number 0
    mode slaac
!
ipv6 local-prefix default
no ppe
upnp lan Home
torrent
    rpc-port 8090
    peer-port 51413
!
user admin
    password md5 2320924ba6e5c1fec3957e587a21535b
    tag cli
    tag cifs
    tag http
    tag ftp
!
user test
    password md5 baadfb946f5d516379cfd75e31e409d9
    tag readonly
!
service dhcp
service dns-proxy
service ftp
service cifs
service http
service telnet
service ntp-client
service upnp
cifs
    share 9430B54530B52EDC 9430B54530B52EDC:
    automount
    permissive
!
!
!
```

History

Version	Description
2.00	The **show running-config** command has been introduced.

3.119.80 show schedule

Description Show parameters of defined schedule. If you use no argument, the entire list of system schedules will be displayed.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **schedule** [*‹name›*]

Arguments

Argument	Value	Description
name	*String*	A schedule name.

Example

```
(show)> schedule 123

         schedule, name = 123:
               action, type = start, left = 561514, next = yes:
                      dow: Tue
                     time: 01:29

               action, type = stop, left = 564274:
                      dow: Tue
                     time: 02:15
```

History

Version	Description
2.06	The **show schedule** command has been introduced.

3.119.81 show self-test

Description Show summary information about system activity. Need to provide remote technical support.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **self-test**

History

Version	Description
2.00	The **show self-test** command has been introduced.

3.119.82 show site-survey

Description Show available wireless networks.

Prefix no No

Change settings No

Multiple input No

Interface type Radio

Synopsis

(show)> **site-survey** *‹name›*

Arguments

Argument	Value	Description
name	*Interface name*	Full name or an alias of the interface. You can see the list of available interfaces with help of **site-survey** [Tab] command.

Example

```
(show)> site-survey [Tab]

 Usage template:
      site-survey {name}

   Choose:
     WifiMaster1
     WifiMaster0
```

```
(show)> site-survey WifiStation0
=================================================================
ESSID                     MAC                  Ch   Rate       Q
=================================================================
Gena                     00:23:f8:5b:d3:f5   11   300Mbit/s  100
Keenetic-2034            00:23:f8:5b:d3:f4   11   300Mbit/s  100
Sonar                     40:4a:03:b4:5d:18   4    54Mbit/s   34
```

History

Version	Description
2.00	The **show site-survey** command has been introduced.

3.119.83 show ssh fingerprint

Description Show current SSH server keys.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **ssh fingerprint**

Example

```
(show)> ssh fingerprint

         rsa: MD5:d0:b0:d4:f7:da:7b:c0:e0:d0:c8:8f:ea:85:3c:09:00

          rsa: SHA1:Nhxg8KNeE62E8zAZJngImcrJkmA

         rsa: SHA256:lM7MyrIaq4qFGT/dyF/t8TbJk5tCzreeGuhO3zaydu4

          ecdsa: ►
MD5:a6:db:b4:fb:3c:b9:ae:31:ca:6d:ca:ed:62:73:a5:7e

          ecdsa: SHA1:ndWg/dx/dP/P8rMkJcVC3XB8nFo

          ecdsa: ►
SHA256:Wp1K9d8MsquQBtlBeBlpVlyKdCN1Vay3BtBWbj0xs+o
```

History

Version	Description
2.12	The **show ssh fingerprint** command has been introduced.

3.119.84 show sstp-server

Description Show current connections to the *SSTP*-server.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **sstp-server**

Example

```
(show)> sstp-server

          enabled: yes
        ndns-name: mymy.keenetic.link
has-ndns-certificate: yes

           tunnel:
        clientaddress: 172.16.3.33
             username: mymy
               uptime: 29

            statistic:
                    rxpackets: 121
         rx-multicast-packets: 0
         rx-broadcast-packets: 0
                     rxbytes: 14715
```

```
                    rxerrors: 0
                   rxdropped: 0
                   txpackets: 78
          tx-multicast-packets: 0
          tx-broadcast-packets: 0
                     txbytes: 48265
                    txerrors: 0
                   txdropped: 0
                   timestamp: 104530.202229
                last-overflow: 0.000000
```

History

Version	Description
2.12	The **show sstp-server** command has been introduced.

3.119.85 show system

Description

Show the general state of the system.

System state general info

❶ CPU load, percentage.
❷ Occupied and available memory info, kilobytes.
❸ Swap file usage info, kilobytes.
❹ System uptime from the start, seconds.

Prefix no

No

Change settings

No

Multiple input

No

Synopsis

(show)> **system**

Example

```
(config)> show system

         hostname: Undefined
       domainname: WORKGROUP
          cpuload: 0 ❶
           memory: 13984/28976 ❷
             swap: 0/0 ❸
           uptime: 153787 ❹
```

History

Version	Description
2.00	The **show system** command has been introduced.

3.119.86 show system cpustat

Description Show device CPU usage.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **system cpustat**

Example

```
(show)> system cpustat

        interval: 36

             busy:
                  cur: 1
                  min: 0
                  max: 11
                  avg: 2

             user:
                  cur: 0
                  min: 0
                  max: 10
                  avg: 1

             nice:
                  cur: 0
                  min: 0
                  max: 0
                  avg: 0

           system:
                  cur: 0
                  min: 0
                  max: 2
                  avg: 0

           iowait:
                  cur: 0
                  min: 0
                  max: 0
                  avg: 0

              irq:
                  cur: 0
                  min: 0
                  max: 0
                  avg: 0

             sirq:
```

```
                cur: 0
                min: 0
                max: 0
                avg: 0
```

History

Version	Description
2.09	The **show system cpustat** command has been introduced.

3.119.87 show tags

Description Show available authentication tags.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **tags**

Example

```
(show)> tags

             tag: cli
             tag: readonly
             tag: http-proxy
             tag: http
             tag: printers
             tag: cifs
             tag: ftp
             tag: ipsec-xauth
             tag: ipsec-l2tp
             tag: opt
             tag: sstp
             tag: torrent
             tag: vpn
```

History

Version	Description
2.00	The **show tags** command has been introduced.

3.119.88 show threads

Description Show the list of active threads in NDM.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **threads**

Example

```
(show)> threads

               thread:
                       name: Cloud agent service
                        tid: 518
        lock_list_complete: yes
                     locks:

                 statistics:
                     interval: 30

                          cpu:
                              now: 17771.481435
                              min: 0
                              max: 0
                              avg: 0
                              cur: 0

               thread:
                       name: FTP brute force detection
                        tid: 519
        lock_list_complete: yes
                     locks:

                 statistics:
                     interval: 30

                          cpu:
                              now: 17771.481440
                              min: 0
                              max: 0
                              avg: 0
                              cur: 0
```

History

Version	Description
2.09	The **show threads** command has been introduced.

3.119.89 show torrent status

Description Show BitTorrent client status.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **torrent status**

Example

```
(show)> torrent status

           state: running
        rpc-port: 8090
```

History

Version	Description
2.03	The **show torrent status** command has been introduced.

3.119.90 show upnp redirect

Description

Show *UPnP* port translation rules. If you use no arguments, the entire list of translation rules will be displayed.

Prefix no

No

Change settings

No

Multiple input

No

Interface type

IP

Synopsis

(show)> **upnp redirect** [(*‹protocol› ‹interface› ‹port›*) | *‹index›*]

Arguments

Argument	Value	Description
protocol	tcp	Rules with *TCP* protocol will be displayed.
	udp	Rules with *UDP* protocol will be displayed.
interface	*Interface name*	Rules with specified interface name will be displayed.
port	*Integer*	Rules with specified port will be displayed.
index	*Integer*	Rule with specified number in the list will be displayed.

Example

```
(show)> upnp redirect udp ISP 11175

          entry:
              index: 1
          interface: ISP
           protocol: udp
               port: 11175
         to-address: 192.168.15.206
            to-port: 11175
        description: Skype UDP at 192.168.12.286:11175 (2024)
            packets: 0
              bytes: 0
```

History

Version	Description
2.00	The **show upnp redirect** command has been introduced.

3.119.91 show version

Description Show firmware version.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **version**

Example

```
(show)> version

          release: 2.10.C.1.0-0
             arch: mips

              ndm:
                exact: 0-d32118a
                cdate: 11 Dec 2017

              bsp:
                exact: 0-cbe0525
                cdate: 11 Dec 2017

              ndw:
              version: 4.2.3.92
             features: ►
wifi_button,flexible_menu,emulate_firmware_progress
           components: ►
ddns,dot1x,interface-extras,miniupnpd,nathelper-ftp,
                       ►
nathelper-pptp,nathelper-sip,ppe,trafficcontrol,
                       ►
cloudcontrol,base,components,corewireless,dhcpd,l2tp,
                       ►
igmp,easyconfig,pingcheck,ppp,pptp,pppoe,ydns

     manufacturer: Keenetic Ltd.
           vendor: Keenetic
           series: KN
            model: Start (KN-1110)
       hw_version: 10118000
            hw_id: KN-1110
           device: Start
            class: Internet Center
```

History

Version	Description
2.00	The **show version** command has been introduced.

3.119.92 show vpn-server

Description Show current connections to the VPN-server.

Prefix no No

Change settings No

Multiple input No

Synopsis

(show)> **vpn-server**

Example

```
(show)> vpn-server

          tunnel:
      clientaddress: 172.16.1.33
           username: test
             uptime: 3

         statistic:
                 rxpackets: 51
       rx-multicast-packets: 0
       rx-broadcast-packets: 0
                   rxbytes: 5440
                  rxerrors: 0
                 rxdropped: 0
                 txpackets: 46
       tx-multicast-packets: 0
       tx-broadcast-packets: 0
                   txbytes: 9229
                  txerrors: 0
                 txdropped: 0
                 timestamp: 146237.254244
              last-overflow: 0.000000
```

History

Version	Description
2.04	The **show vpn-server** command has been introduced.

3.120 snmp community

Description Set new name for *SNMP* community. By default, common name `public` is used.

Command with **no** prefix resets setting to default.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config)> **snmp community** *‹community›*

(config)> **no snmp community**

Arguments

Argument	Value	Description
community	*String*	New community name.

Example

```
(config)> snmp community Co_test
Snmp::Manager: SNMP community set to "Co_test".
(config)> no snmp community
Snmp::Manager: SNMP community reset to "public".
```

History

Version	Description
2.08	The **snmp community** command has been introduced.

3.121 snmp contact

Description Assign the contact name of *SNMP* agent. By default, the name is not defined.

Command with **no** prefix resets setting.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config)> **snmp contact** *‹contact›*

(config)> **no snmp contact**

Arguments

Argument	Value	Description
contact	*String*	*SNMP* contact info.

Example

```
(config)> snmp contact Cont_test
Snmp::Manager: SNMP contact info set to "Cont_test".
(config)> no snmp contact
Snmp::Manager: SNMP community info reset.
```

History

Version	Description
2.08	The **snmp contact** command has been introduced.

3.122 snmp location

Description Assign the location of *SNMP* agent. By default, the location is not defined.

Command with **no** prefix resets setting.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(config)> **snmp location** ‹*location*›

(config)> **no snmp location**

Arguments

Argument	Value	Description
`location`	*String*	*SNMP* device location.

Example

```
(config)> snmp location Odintsovo
Snmp::Manager: SNMP device location set to "Odintsovo".
(config)> no snmp location
Snmp::Manager: SNMP device location reset.
```

History

Version	Description
2.08	The **snmp location** command has been introduced.

3.123 sstp-server

Description Access to a group of commands to configure *SSTP*-server parameters.

Prefix no No

Change settings No

Multiple input No

Group entry `(sstp-server)`

Synopsis

(config)> **sstp-server**

History

Version	Description
2.12	The **sstp-server** command has been introduced.

3.123.1 sstp-server dhcp route

Description

Assign a route which is transmitted in DHCP INFORM messages to the *SSTP*-server clients.

Command with **no** prefix cancels the specified route. If you use no arguments, the entire list of routes will be cleared.

Prefix no

Yes

Change settings

Yes

Multiple input

Yes

Synopsis

(sstp-server)> **dhcp route** ‹*address*› ‹*mask*›

(sstp-server)> **no dhcp route** [‹*address*› ‹*mask*›]

Arguments

Argument	Value	Description
address	*IP-address*	Network client address.
mask	*IP-mask*	Network client mask. There are two ways to enter the mask: the canonical form (for example, 255.255.255.0) and the form of prefix bit length (for example, /24).

Example

```
(sstp-server)> dhcp route 192.168.2.0/24
SstpServer::Manager: Added DHCP INFORM route to ►
192.168.2.0/255.255.255.0.
```

```
(sstp-server)> no dhcp route
SstpServer::Manager: Cleared DHCP INFORM routes.
```

History

Version	Description
2.12	The **sstp-server dhcp route** command has been introduced.

3.123.2 sstp-server interface

Description

Bind *SSTP*-server to the specified interface.

Command with **no** prefix unbinds the interface.

Prefix no

Yes

Change settings Yes

Multiple input No

Synopsis

(sstp-server)> **interface** ‹*interface*›

(sstp-server)> **no interface**

Arguments

Argument	Value	Description
interface	*Interface name*	Full interface name or an alias. You can see the list of available interfaces with help of **interface** [Tab] command.

Example

```
(sstp-server)> interface [Tab]

 Usage template:
        interface {interface}

   Choose:
         GigabitEthernet1
                      ISP
 WifiMaster0/AccessPoint2
 WifiMaster1/AccessPoint1
 WifiMaster0/AccessPoint3
 WifiMaster0/AccessPoint0
              AccessPoint
 WifiMaster1/AccessPoint2
 WifiMaster0/AccessPoint1
                GuestWiFi
```

```
(sstp-server)> interface Bridge0
SstpServer::Manager: Bound to Bridge0.
```

History

Version	Description
2.12	The **sstp-server interface** command has been introduced.

3.123.3 sstp-server ipv6cp

Description Enable IPv6 support. DHCP IPv6 pools are created for each *SSTP*-server. By default, the setting is disabled.

Command with **no** prefix disables IPv6 support.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(sstp-server)> **ipv6cp**

(sstp-server)> **no ipv6cp**

Example

```
(sstp-server)> ipv6cp
SstpServer::Manager: IPv6 control protocol enabled.
```

```
(sstp-server)> no ipv6cp
SstpServer::Manager: IPv6 control protocol disabled.
```

History

Version	Description
3.00	The **sstp-server ipv6cp** command has been introduced.

3.123.4 sstp-server lcp echo

Description

Specify the testing rules of the SSTP-connections with *LCP* echo tools.

Command with **no** prefix disables *LCP* echo.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(sstp-server)> **lcp echo** ‹*interval*› ‹*count*› [**adaptive**]

(sstp-server)> **no lcp echo**

Arguments

Argument	Value	Description
`interval`	*Integer*	Interval between sending *LCP* echo, in seconds. If within the specified time interval there is no *LCP* echo request from the remote location, the same request will be sent there asking for response *LCP* reply.
`count`	*Integer*	The number of consecutive requests *LCP* echo sent, for which no response *LCP* reply was received. If count of *LCP* echo requests goes unanswered, the connection is terminated.
`adaptive`	*Keyword*	Pppd will send LCP echo-request frames only if no traffic was received from the peer since the last echo-request was sent.

Example

```
(sstp-server)> lcp echo 5 3
SstpServer::Manager: LCP echo parameters updated.
```

History

Version	Description
2.12	The **sstp-server lcp echo** command has been introduced.

3.123.5 sstp-server lcp force-pap

Description

Enforce the *PAP* authentication only for *SSTP*-server.

Command with **no** prefix disables *PAP* authentication.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(sstp-server)> **lcp force-pap**

(sstp-server)> **no lcp force-pap**

Example

```
(sstp-server)> lcp force-pap
SstpServer::Manager: Forced PAP-only authentication.
```

```
(sstp-server)> no lcp force-pap
SstpServer::Manager: Disabled forcing PAP-only authentication.
```

History

Version	Description
3.05	The **sstp-server lcp force-pap** command has been introduced.

3.123.6 sstp-server mru

Description

Set *MRU* value to be transmitted to *SSTP*-server. By default, `1350` value is used.

Command with **no** prefix resets value to default.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(sstp-server)> **mru** *‹value›*

(sstp-server)> **no mru**

Arguments

Argument	Value	Description
`value`	*Integer*	*MRU* value. Can take values from 128 to 1500 inclusively.

Example

```
(sstp-server)> mru 200
SstpServer::Manager: MRU set to 200.
```

History

Version	Description
2.12	The **sstp-server mru** command has been introduced.

3.123.7 sstp-server mtu

Description

Set *MTU* value to be transmitted to *SSTP*-server. By default, `1350` value is used.

Command with **no** prefix resets value to default.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(sstp-server)> **mtu** ‹*value*›

(sstp-server)> **no mtu**

Arguments

Argument	Value	Description
`value`	*Integer*	*MTU* value. Can take values from 128 to 1500 inclusively.

Example

```
(sstp-server)> mtu 200
SstpServer::Manager: MTU set to 200.
```

History

Version	Description
2.12	The **sstp-server mtu** command has been introduced.

3.123.8 sstp-server multi-login

Description

Allow connection to *SSTP*-server for multiple users from one account.

Command with **no** prefix disables this feature.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(sstp-server)> **multi-login**

(sstp-server)> **no multi-login**

Example

```
(sstp-server)> multi-login
SstpServer::Manager: Enabled multiple login.
```

History

Version	Description
2.12	The **sstp-server multi-login** command has been introduced.

3.123.9 sstp-server pool-range

Description

Assign a pool of addresses for the clients that connect to the *SSTP*-server.

Command with **no** prefix removes a pool.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(sstp-server)> **pool-range** ‹*begin*› [‹*size*›]

(sstp-server)> **no pool-range**

Arguments

Argument	Value	Description
`begin`	*IP-address*	Start address of pool.
`size`	*Integer*	Pool size. If not defined, size `10` is used.

Example

```
(sstp-server)> pool-range 192.168.1.22 7
SstpServer::Manager: Configured pool range 192.168.1.22 to ►
192.168.1.28.
```

History

Version	Description
2.12	The **sstp-server pool-range** command has been introduced.

3.123.10 sstp-server static-ip

Description

Bind IP-address to the user. User account must have `sstp` tag.

Command with **no** prefix removes binding.

Prefix no

Yes

Change settings Yes

Multiple input Yes

Synopsis

(sstp-server)> **static-ip** *‹name› ‹address›*

(sstp-server)> **no static-ip** *‹name›*

Arguments

Argument	Value	Description
name	*String*	Username.
address	*IP-address*	IP-address to bind.

Example

```
(sstp-server)> static-ip admin 192.168.1.22
SstpServer::Manager: Static IP 192.168.1.22 assigned to user ►
"admin".
```

History

Version	Description
2.12	The **sstp-server static-ip** command has been introduced.

3.124 system

Description Access to a group of commands to configure global parameters.

Prefix no No

Change settings No

Multiple input No

Group entry (system)

Synopsis

(config)> **system**

History

Version	Description
2.00	The **system** command has been introduced.

3.124.1 system button

Description Configure device buttons to handle specific actions. Available handlers depend on hardware configuration and installed modules.

Command with **no** prefix remove setting.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(system)> **button** *‹button›* **on** *‹action›* **do** *‹handler›*

(system)> **no button** *‹button›*

Arguments

Argument	Value	Description
button	RESET	RESET button.
	WLAN	Wireless LAN button.
action	click	Single click.
	double-click	Double click.
	hold	Push and hold for 3 seconds. RESET button hold is 10 seconds.
handler	FactoryReset	Reset system to factory defaults.
	Reboot	System reboot.
	WifiToggle	Switch Wi-Fi on/off.
	WifiGuestApToggle	Switch Guest Wi-Fi on/off.
	WpsStartMainAp	Start WPS (2.4GHz only).
	WpsStartMainAp5	Start WPS (5GHz only).
	WpsStartAllMainAp	Start WPS (all frequency bands).

Example

```
(system)> button WLAN on double-click do WifiGuestApToggle
Peripheral::Manager: "WLAN/double-click" handler set.
```

History

Version	Description
2.03	The **system button** command has been introduced.

3.124.2 system clock date

Description Adjust system date and time.

Prefix no No

Change settings Yes

Multiple input No

Synopsis

(system)> **clock date** *‹date-and-time›*

Arguments

Argument	Value	Description
`date-and-time`	*String*	Current date and time in `DD MM YYYY HH:MM:SS` format.

Example

```
(system)> clock date 18 07 2012 09:52:33
System date and time has been changed.
```

History

Version	Description
2.00	The **system clock date** command has been introduced.

3.124.3 system clock timezone

Description Set the system timezone.

Command with **no** prefix resets timezone to default (GMT).

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(system)> **clock timezone** *‹locality›*

(system)> **no clock timezone** *‹locality›*

Arguments

Argument	Value	Description
`locality`	*String*	Name of the city, indicating the time zone.

Example

```
(system)> clock timezone Dublin
the system timezone is set to "Dublin".
```

History

Version	Description
2.00	The **system clock timezone** command has been introduced.

3.124.4 system configuration factory-reset

Description Reset configuration to the factory settings for all modes.

Prefix no No

Change settings Yes

Multiple input No

Synopsis

(system)> **configuration factory-reset**

Example

```
(system)> configuration factory-reset
Core::Configuration: the system configuration reset to factory ►
defaults.
```

History

Version	Description
2.00	The **system configuration factory-reset** command has been introduced.

3.124.5 system configuration save

Description Save the system configuration asynchronously.

Prefix no No

Change settings Yes

Multiple input No

Synopsis

(system)> **configuration save**

Example

```
(system)> configuration save
Saving configuration.
```

History

Version	Description
2.05.B.1	The **system configuration save** command has been introduced.

3.124.6 system debug

Description Enable system debug. By default, setting is disabled.

Command with **no** prefix disables the feature.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(system)> **debug**

(system)> **no debug**

Example

```
(system)> debug
Core::Debug: System debug enabled.
```

History

Version	Description
2.03	The **system debug** command has been introduced.

3.124.7 system description

Description

Set the system description as an arbitrary string. By default, description City (KN-1510) is used.

Command with **no** prefix resets description to default.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(system)> **description** ‹*description*›

(system)> **no description**

Arguments

Argument	Value	Description
description	*String*	System description no longer than 256 bytes.

Example

```
(system)> description DEVICE
Core::System::Info: Description saved.
```

```
(config)> show version
  ...
     manufacturer: Keenetic Ltd.
           vendor: Keenetic
           series: KN
            model: Ultra (KN-1810)
       hw_version: 10188000
            hw_id: KN-1810
           device: Ultra
            class: Internet Center
           region: RU
      description: DEVICE
```

```
(config)> show running-config
  ...
    set vm.swappiness 60
    set vm.overcommit_memory 0
    set vm.vfs_cache_pressure 1000
    set dev.usb.force_usb2 0
    domainname WORKGROUP
    hostname Keenetic_Ultra
    description DEVICE
  ...
```

```
(system)> no description
Core::System::Info: Description reset to default.
```

```
(config)> show version
  ...
     manufacturer: Keenetic Ltd.
           vendor: Keenetic
           series: KN
            model: Ultra (KN-1810)
       hw_version: 10188000
            hw_id: KN-1810
           device: Ultra
            class: Internet Center
           region: RU
      description: Keenetic Ultra (KN-1810)
```

History

Version	Description
2.15	The **system description** command has been introduced.

3.124.8 system domainname

Description

Assign domain name for the system.

Command with **no** prefix removes domain name.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(system)> **domainname** ‹*domain*›

(system)> **no domainname**

Arguments

Argument	Value	Description
domain	*String*	The domain name to assign.

Example

```
(system)> domainname keenetic
Domainname saved.
```

History

Version	Description
2.00	The **system domainname** command has been introduced.

3.124.9 system hostname

Description

Set the host name. Host name used to identify a node in the network. It is required to enable some of the built-in services, such as CIFS.

Command with **no** prefix sets the default value, which depends on the model name.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(system)> **hostname** ‹*hostname*›

(system)> **no hostname**

Arguments

Argument	Value	Description
hostname	*String*	Name of the host.

Example

```
(system)> hostname KN1010
Core::System::Hostname: The host name set.
```

```
(system)> no hostname
Core::System::Hostname: The host name reset.
```

History

Version	Description
2.00	The **system hostname** command has been introduced.

3.124.10 system led

Description

Configure general purpose LEDs. By default, LED FN shows the updates for your device are available.

Command with **no** prefix resets the setting to default.

Prefix no

Yes

Change settings

Yes

Multiple input

Yes

Synopsis

(system)> **led** ‹*led*› **indicate** ‹*control*›

(system)> **no led** [‹*led*› [**indicate**]]

Arguments

Argument	Value	Description
led	FN	LED name.
control	UpdatesAvailable	LED notifies you the updates for your device are available.
	BackupWan	LED shows that backup connection is active at the moment.
	SelectedWan	LED shows status of the interface defined with **interface led wan** command.
	SelectedSchedule	LED shows status of scheduled event assigned with **schedule led** command.
indicate	*Keyword*	Turn off the indicator completely.

Example

```
(system)> led FN indicate SelectedWan
Peripheral::Manager: "SelectedWan" control bound to "FN" LED.
```

```
(system)> no led FN indicate
Peripheral::Manager: "FN" LED control binding removed.
```

History

Version	Description
2.08	The **system led** command has been introduced.

3.124.11 system led power schedule

Description

Assign a schedule for the LEDs on the device. Schedule must be created and customized with **schedule action** command before execution.

Command with **no** prefix unbinds the schedule.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(system)> **led power schedule** ‹*schedule*›

(system)> **no led power schedule**

Arguments

Argument	Value	Description
schedule	*Schedule name*	The name of the schedule that was created with **schedule** group of commands.

Example

```
(system)> led power schedule schedule1
Core::Peripheral::Manager: Set LED power schedule "schedule1".
```

```
(system)> no led power schedule
Core::Peripheral::Manager: Clear LED power schedule.
```

History

Version	Description
3.06	The **system led power schedule** command has been introduced.

3.124.12 system led power shutdown

Description Shutdown the LEDs on the device.

Command with **no** prefix turns LEDs on.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(system)> **led power shutdown** *‹mode›*

(system)> **no led power shutdown**

Arguments

Argument	Value	Description
mode	all	Shutdown all the LEDs.
	front	Shutdown the LEDs on the front panel.
	back	Shutdown the LEDs on the back panel.

Example

```
(system)> led power shutdown all
Core::Peripheral::Manager: Set LED shutdown mode to "all".
```

```
(system)> no led power shutdown
Core::Peripheral::Manager: Set LED shutdown mode to "none".
```

History

Version	Description
3.06	The **system led power shutdown** command has been introduced. Previous command name is **system led shutdown**.

3.124.13 system log clear

Description Clear the system log.

Prefix no No

Change settings No

Multiple input No

Synopsis

(system)> **log clear**

Example

```
(system)> log clear
Syslog: the system log has been cleared.
```

History

Version	Description
2.00	The **system log clear** command has been introduced.

3.124.14 system log reduction

Description Enable repeated message reduction. By default, the setting is enabled.

Command with **no** prefix disables the feature.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(system)> **log reduction**

(system)> **no log reduction**

Example

```
(system)> log reduction
```

```
(system)> no log reduction
```

History

Version	Description
2.04	The **system log reduction** command has been introduced.

3.124.15 system log server

Description Add remote log server.

Prefix no Yes

Change settings Yes

Multiple input Yes

Synopsis

(system)> **log server** ‹*address*› [**:** ‹*port*›]

(system)> **no log server** [‹*address*› [**:** ‹*port*›]]

Arguments

Argument	Value	Description
address	*IP-address*	Remote log server address.
port	*Integer*	Remote log server port.

Example

```
(system)> log server 192.168.1.1:8080
Syslog: server 192.168.1.1:8080 added.
```

History

Version	Description
2.00	The **system log server** command has been introduced.

3.124.16 system log suppress

Description

Add message suppression rule.

Command with **no** prefix removes the rule.

Prefix no

Yes

Change settings

Yes

Multiple input

Yes

Synopsis

(system)> **log suppress** ‹*ident*›

(system)> **no log suppress** [‹*ident*›]

Arguments

Argument	Value	Description
ident	*String*	Process ID which messages need to suppress.

Example

```
(system)> log suppress kernel
Core::Syslog: Added suppression "kernel".
```

```
(system)> no log suppress kernel
Core::Syslog: Deleted suppression "kernel".
```

```
(system)> log suppress transmissiond
Core::Syslog: Added suppression "transmissiond".
```

```
(system)> no log suppress transmissiond
Core::Syslog: Deleted suppression "transmissiond".
```

History

Version	Description
2.04	The **system log suppress** command has been introduced.

3.124.17 system mode

Description Select system operating mode for City.

Prefix no No

Change settings Yes

Multiple input No

Synopsis

(system)> **mode** *‹mode›*

Arguments

Argument	Value	Description
mode	router	Main mode.
	client	Network adapter mode to connect Ethernet devices to Wi-Fi network.
	repeater	Repeater mode to extend Wi-Fi network using a wireless connection.
	ap	Access point mode to extend Wi-Fi network using a wired Ethernet connection.

Example

```
(system)> mode repeater
Core::Mode: The system switched to "repeater" mode, reboot the ►
device to apply the settings.
```

History

Version	Description
2.05	The **system mode** command has been introduced.

3.124.18 system ndss dump-report disable

Description Disable product improvement program. By default, setting is enabled.

Command with **no** prefix enables the program.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(system)> **ndss dump-report disable**

(system)> **no ndss dump-report disable**

Example

```
(system)> ndss dump-report disable
Core::Ndss: Dump-reporting disabled.
```

```
(system)> no ndss dump-report disable
Core::Ndss: Dump-reporting enabled.
```

History

Version	Description
3.05	The **system ndss dump-report disable** command has been introduced. Previous command name is **system dump-report disable**.

3.124.19 system reboot

Description

Reboot the system. If the parameter is set, reboot is executed after a timeout, in seconds. If the timer is already set, using of the command replaces the old value of the timer to the new one.

Using a scheduled reboot is convenient in the case when the device is under remote control, and the user doesn't understand the effect of the commands he/she is trying. The user can turn on a scheduled reboot for fear of losing control over the device. After reboot the system will return to its original state and become available.

Command with **no** prefix cancels reboot or removes the reboot on schedule.

Prefix no

Yes

Change settings

No

Multiple input

No

Synopsis

(system)> **reboot** [‹*interval*› | **schedule** ‹*schedule*›]

(system)> **no reboot** [**schedule**]

Arguments

Argument	Value	Description
`interval`	*Integer*	Timeout for reboot, in seconds. If not specified, the reboot will be executed immediately.
`schedule`	*Schedule name*	The name of the schedule that was created with **schedule** group of commands.

Example

```
(system)> reboot 20
Core::System::RebootManager: Rebooting in 20 seconds.
```

```
(system)> no reboot
Core::System::RebootManager: Reboot cancelled.
```

```
(system)> reboot schedule rebootroute
Core::System::RebootManager: Set reboot schedule "rebootroute".
```

```
(system)> no reboot schedule
Core::System::RebootManager: Schedule disabled.
```

History

Version	Description
2.00	The **system reboot** command has been introduced.
2.12	The **schedule** argument has been added.

3.124.20 system set

Description

Set the value of the specified system parameter and save it in the current settings.

Command with **no** prefix returns the default value to the specified parameter (before the first change).

Prefix no

Yes

Change settings

Yes

Multiple input

Yes

Synopsis

(system)> **set** *‹name› ‹value›*

(system)> **no set** *‹name›*

Arguments

Argument	Value	Description
name	*String*	Identifier of the system parameter.
value	*String*	New value of the system parameter.

Example

```
(config)> system
(system)> set net.ipv4.ip_forward 1
(system)> set net.ipv4.tcp_fin_timeout 30
(system)> set net.ipv4.tcp_keepalive_time 120
(system)> set ►
net.ipv4.netfilter.ip_conntrack_tcp_timeout_established 1200
(system)> set net.ipv4.netfilter.ip_conntrack_udp_timeout 60
(system)> set net.ipv4.netfilter.ip_conntrack_max 4096
(system)> exit
(config)> show running-config
system
set net.ipv4.ip_forward 1
    set net.ipv4.tcp_fin_timeout 30
    set net.ipv4.tcp_keepalive_time 120
```

```
   set net.ipv4.netfilter.ip_conntrack_tcp_timeout_established ►
1200
    set net.ipv4.netfilter.ip_conntrack_udp_timeout 60
    set net.ipv4.netfilter.ip_conntrack_max 4096
!
...
(config)>
```

History

Version	Description
2.00	The **system set** command has been introduced.

3.124.21 system trace lock threshold

Description

Set a trace lock threshold for the system threads. If the threshold value is exceeded, information about this thread (for example, SCGI session) is saved in the system log. By default, setting is disabled.

Command with **no** prefix disables the trace lock threshold feature.

Prefix no

Yes

Change settings

No

Multiple input

No

Synopsis

(system)> **system trace lock threshold** *‹threshold›*

(system)> **no system trace lock threshold**

Arguments

Argument	Value	Description
`threshold`	*String*	Threshold value in milliseconds. Can take values from 100 to 100000000 inclusively. The threshold value is not saved into `startup-config`.

Example

```
(system)> system trace lock threshold 100
Lockable: Set threshold to 100 ms.
```

```
(system)> no trace lock threshold
Lockable: Reset threshold.
```

History

Version	Description
3.03	The **system trace lock threshold** command has been introduced.

3.125 tools

Description Access to a group of commands to test the environment.

Prefix no No

Change settings No

Multiple input No

Group entry `(tools)`

Synopsis

```
(config)>  tools
```

History

Version	Description
2.00	The **tools** command has been introduced.

3.125.1 tools arping

Description Command action is analogous to **tools ping** command, but operates at the link layer of the OSI model using the *ARP* protocol.

Prefix no No

Change settings No

Multiple input No

Synopsis

```
(tools)>  arping ‹address› source-interface ‹source-interface› [ count ‹count› ] [ wait-time ‹wait-time› ]
```

Arguments

Argument	Value	Description
address	*IP-address*	IP-address of the respondent.
source-interface	*Interface name*	Name of source-interface.
count	*Integer*	Quantity of requests. If not specified, the command will run until interrupted by the user.
wait-time	*Integer*	The maximum response time, in milliseconds.

Example

```
(tools)> arping 192.168.15.51 source-interface Home count 4 ► wait-time 3000
Starting the ARP ping to "192.168.15.51"...
ARPING 192.168.15.51 from 192.168.15.1 br0.
Unicast reply from 192.168.15.51 [9c:b7:0d:ce:51:6a] 1.884 ms.
Unicast reply from 192.168.15.51 [9c:b7:0d:ce:51:6a] 1.831 ms.
```

```
Sent 4 probes, received 2 responses.
Process terminated.
```

History

Version	Description
2.00	The **tools arping** command has been introduced.

3.125.2 tools ping

Description

Send Echo-Request requests of ICMP protocol to specified network node and register received Echo-Reply responses. The time between sending request and receiving the response Round Trip Time (RTT) allows you to define double ended delays on the route and frequency of packet losses, that is, indirectly determine loading on the channels of data transmission and intermediate devices.

Total absence of ICMP-replies can also mean that the remote node (or any of the intermediate routers) blocks ICMP Echo-Reply or ignores ICMP Echo-Request.

Prefix no

No

Change settings

No

Multiple input

No

Synopsis

(tools)> **ping** ‹*host*› [**count** ‹*count*›] [**size** ‹*packetsize*›]

Arguments

Argument	Value	Description
host	*String*	Domain name or host IP-address.
count	*Integer*	Quantity of ICMP Echo requests. If not specified, the command will run until interrupted by the user.
packetsize	*Integer*	Size of the ICMP Echo-Request data field in bytes. By default — 56, which together with the 8-byte header specifies the size of the ICMP-pack — 64 bytes.

Example

```
(tools)> ping 192.168.1.33 count 3 size 100
Sending ICMP ECHO request to 192.168.1.33
PING 192.168.1.33 (192.168.1.33) 72 (100) bytes of data.
100 bytes from 192.168.1.33: icmp_req=1, ttl=128, time=2.35 ms.
100 bytes from 192.168.1.33: icmp_req=2, ttl=128, time=1.07 ms.
100 bytes from 192.168.1.33: icmp_req=3, ttl=128, time=1.06 ms.
--- 192.168.1.33 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss,
0 duplicate(s), time 2002.65 ms.
Round-trip min/avg/max = 1.06/1.49/2.35 ms.
Process terminated.
```

History

Version	Description
2.00	The **tools ping** command has been introduced.

3.125.3 tools ping6

Description

Send Echo-Request requests of ICMPv6 protocol to specified network node and register received Echo-Reply responses. The time between sending request and receiving the response Round Trip Time (RTT) allows you to define double ended delays on the route and frequency of packet losses, that is, indirectly determine loading on the channels of data transmission and intermediate devices.

Total absence of ICMP-replies can also mean that the remote node (or any of the intermediate routers) blocks ICMP Echo-Reply or ignores ICMP Echo-Request.

Prefix no

No

Change settings

No

Multiple input

No

Synopsis

(tools)> **ping6** *‹host›* [**count** *‹count›*] [**size** *‹packetsize›*]

Arguments

Argument	Value	Description
host	*String*	Domain name or host IPv6-address.
count	*Integer*	Quantity of ICMPv6 Echo requests. If not specified, the command will run until interrupted by the user.
packetsize	*Integer*	Size of the ICMPv6 Echo-Request data field in bytes. By default — 56, which together with the 8-byte header specifies the size of the ICMPv6-pack — 64 bytes.

Example

```
(tools)> ping6 fd4b:f12b:5d59:0:1108:4407:b772:20cd count 3 size ►
100
Sending ICMPv6 ECHO request to ►
fd4b:f12b:5d59:0:1108:4407:b772:20cd
PING fd4b:f12b:5d59:0:1108:4407:b772:20cd ►
(fd4b:f12b:5d59:0:1108:4407:b772:20cd) 52 (60) bytes of data.
60 bytes from fd4b:f12b:5d59:0:1108:4407:b772:20cd ►
(fd4b:f12b:5d59:0:1108:4407:b772:20cd): icmp_req=1, ttl=64, ►
time=7.18 ms.
60 bytes from fd4b:f12b:5d59:0:1108:4407:b772:20cd ►
(fd4b:f12b:5d59:0:1108:4407:b772:20cd): icmp_req=2, ttl=64, ►
time=8.42 ms.
60 bytes from fd4b:f12b:5d59:0:1108:4407:b772:20cd ►
(fd4b:f12b:5d59:0:1108:4407:b772:20cd): icmp_req=3, ttl=64, ►
time=1.51 ms.
```

```
--- fd4b:f12b:5d59:0:1108:4407:b772:20cd ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss,
0 duplicate(s), time 2002.61 ms.
Round-trip min/avg/max = 1.51/5.70/8.42 ms.
Process terminated.
```

History

Version	Description
2.00	The **tools ping6** command has been introduced.

3.125.4 tools traceroute

Description Show the route to a network host.

Prefix no No

Change settings No

Multiple input No

Synopsis

(tools)> **traceroute** *‹host›* [**count** *‹count›*] [**interval** *‹interval›*] [**wait-time** *‹wait-time›*] [**packet-size** *‹packet-size›*] [**max-ttl** *‹max-ttl›*] [**port** *‹port›*] [**source-address** *‹source-address›*] [**source-interface** *‹source-interface›*] [**type** *‹type›*] [**tos** *‹tos›*]

Arguments

Argument	Value	Description
`host`	*String*	Name of the target host.
`count`	*Integer*	Number of probe packets per hop. Default value — 3. Value must be in the range [1;10].
`interval`	*Integer*	Time in seconds between sending packets. Default value — 0. Value must be in the range [0;15].
`wait-time`	*Integer*	Time to wait for a response to a probe (in seconds). Default value — 1. Value must be in the range [1;15].
`packet-size`	*Integer*	Size of packet according to the protocol `type`. For `tcp` type default packet size is 52. Range of values [52]. For `udp` and `icmp` types default packet size is 60. Range of values [28;65535].
`max-ttl`	*Integer*	Maximum number of hops (max time-to-live value) traceroute will probe. Default value — 30. Value must be in the range [1;255].
`port`	*Integer*	Destination port.

Argument	Value	Description
		For `tcp` type default port is 80. For `udp` type default port is 33434. For `icmp` type default port is 1.
`source-address`	*`String`*	Address of the outgoing interface.
`source-interface`	*`String`*	Interface to be used as the source interface in outgoing probe packets.
`type`	`tcp`	*TCP* protocol.
	`udp`	*UDP* protocol. Used by default.
	`icmp`	*ICMP* protocol.
`tos`	*`Integer`*	Type Of Service. Default value — 0. Value must be in the range [0;255]

Example

```
(tools)> traceroute google.com count 5 interval 5
starting traceroute to google.com...
traceroute to google.com (64.233.161.113), 30 hops maximum, 60 ►
byte packets.
 1  192.168.233.1 (192.168.233.1)  2.742 ms  2.406 ms  2.460 ms ►
 2.191 ms  2.957 ms
 2  10.77.140.1 (10.77.140.1)  3.301 ms  3.847 ms  3.839 ms
process terminated
```

History

Version	Description
2.00	The **tools traceroute** command has been introduced.

3.126 udpxy

Description Access to a group of commands to configure *udpxy* parameters.

Prefix no No

Change settings No

Multiple input No

Group entry `(udpxy)`

Synopsis

```
(config)>  udpxy
```

History

Version	Description
2.03	The **udpxy** command has been introduced.

3.126.1 udpxy buffer-size

Description Set *udpxy* buffer size. By default, 2048 value is used.

Command with **no** prefix resets buffer size to default.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(udpxy)> **buffer-size** ‹*size*›

(udpxy)> **no buffer-size**

Arguments

Argument	Value	Description
size	*Integer*	Buffer size in bytes. Can take values from 1 to 1048576.

Example

```
(udpxy)> buffer-size 500
Udpxy::Manager: a buffer size set to 500 bytes.
```

History

Version	Description
2.04	The **udpxy buffer-size** command has been introduced.

3.126.2 udpxy buffer-timeout

Description Set *udpxy* timeout to hold data in the buffer. By default, 1 value is used.

Command with **no** prefix resets timeout to default.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(udpxy)> **buffer-timeout** ‹*timeout*›

(udpxy)> **no buffer-timeout**

Arguments

Argument	Value	Description
timeout	*Integer*	Timeout value in seconds. Can take values from -1 to 60. -1 — unlimited timeout.

Example

```
(udpxy)> buffer-timeout 10
Udpxy::Manager: a hold data timeout set to 10 sec.
```

History

Version	Description
2.04	The **udpxy buffer-timeout** command has been introduced.

3.126.3 udpxy interface

Description

Bind *udpxy* to the specified interface. By default, current default gateway is used.

Command with **no** prefix resets setting to default.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(udpxy)> **interface** ‹*interface*›

(udpxy)> **no interface**

Arguments

Argument	Value	Description
interface	*Interface name*	Full interface name or an alias. You can see the list of available interfaces with help of **interface** [Tab] command.

Example

```
(udpxy)> interface [Tab]

 Usage template:
        interface {interface}

   Choose:
         GigabitEthernet1
                      ISP
 WifiMaster0/AccessPoint2
 WifiMaster1/AccessPoint1
 WifiMaster0/AccessPoint3
 WifiMaster0/AccessPoint0
              AccessPoint
```

```
(udpxy)> interface ISP
Udpxy::Manager: bound to FastEthernet0/Vlan2.
```

History

Version	Description
2.02	The **udpxy interface** command has been introduced.

3.126.4 udpxy port

Description Specify port for HTTP requests. By default, `4022` value is used.

Command with **no** prefix resets setting to default.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(udpxy)> **port** *‹port›*

(udpxy)> **no port**

Arguments

Argument	Value	Description
`port`	*Integer*	Port number. Can take values from 0 to 65535.

Example

```
(udpxy)> port 2323
Udpxy::Manager: a port set to 2323.
```

History

Version	Description
2.03	The **udpxy port** command has been introduced.

3.126.5 udpxy renew-interval

Description Set renew interval of subscription to the multicast channel. By default, `0` value is used, ie the subscription is not renewed.

Command with **no** prefix resets setting to default.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(udpxy)> **renew-interval** *‹renew-interval›*

(udpxy)> **no renew-interval**

Arguments

Argument	Value	Description
`renew-interval`	*Integer*	Renew interval of subscription in seconds. Can take values from 0 to 3600.

Example

```
(udpxy)> renew-interval 120
Udpxy::Manager: a renew subscription interval value set to 120 ►
sec.
```

History

Version	Description
2.03	The **udpxy renew-interval** command has been introduced.

3.126.6 udpxy timeout

Description Set connection timeout. By default, 5 value is used.

Command with **no** prefix resets setting to default.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(udpxy)> **timeout** *‹timeout›*

(udpxy)> **no timeout**

Arguments

Argument	Value	Description
timeout	*Integer*	Timeout in seconds. Can take values from 5 to 60.

Example

```
(udpxy)> timeout 10
Udpxy::Manager: a stream timeout set to 10 sec.
```

History

Version	Description
2.03	The **udpxy timeout** command has been introduced.

3.127 upnp forward

Description Add *UPnP* forwarding rule.

Command with **no** prefix removes rule from the list.

Prefix no Yes

Change settings Yes

Multiple input Yes

Interface type IP

Synopsis

(config)> **upnp forward** ‹*protocol*› [*interface*] ‹*address*› ‹*port*›

(config)> **no upnp forward** [‹*index*› | (‹*protocol*› ‹*address*› ‹*port*›)]

Arguments

Argument	Value	Description
protocol	tcp	Rule for *TCP* protocol will be added/deleted.
	udp	Rule for *UDP* protocol will be added/deleted.
interface	*Interface name*	Rule for specified interface name will be added.
address	*IP-address*	Rule for specified IP-address will be added/deleted.
port	*Integer*	Rule for specified port will be added/deleted.
index	*Integer*	Rule with specified number in the list will be removed.

History

Version	Description
2.00	The **upnp forward** command has been introduced.

3.128 upnp lan

Description

Set LAN interface where the *UPnP* service is running. The service works for one network segment only.

Command with **no** prefix removes setting.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Interface type

IP

Synopsis

(config)> **upnp lan** ‹*interface*›

(config)> **no upnp lan**

Arguments

Argument	Value	Description
interface	*Interface name*	Full interface name or an alias. You can see the list of available interfaces with help of **interface** [Tab] command.

Example

```
(config)> upnp lan [Tab]

 Usage template:
             lan {interface}

   Choose:
         GigabitEthernet1
                      ISP
 WifiMaster0/AccessPoint2
 WifiMaster1/AccessPoint1
 WifiMaster0/AccessPoint3
 WifiMaster0/AccessPoint0
             AccessPoint
 WifiMaster1/AccessPoint2
 WifiMaster0/AccessPoint1
               GuestWiFi
```

```
(config)> upnp lan PPTP0
using LAN interface: PPTP0.
```

History

Version	Description
2.00	The **upnp lan** command has been introduced.

3.129 upnp redirect

Description

Add *UPnP* port translation rule.

Command with **no** prefix removes rule from the list. If you use no arguments, the entire list of rules will be removed.

Prefix no

Yes

Change settings

Yes

Multiple input

Yes

Interface type

IP

Synopsis

(config)> **upnp redirect** ‹*protocol*› ‹*interface*› ‹*port*› ‹*to-address*› [*to-port*]

(config)> **no upnp redirect** [**and forward** | [‹*index*› | (‹*protocol*› ‹*port*›)]]

Arguments

Argument	Value	Description
protocol	tcp	Rule for *TCP* protocol will be added/deleted.
	udp	Rule for *UDP* protocol will be added/deleted.

Argument	Value	Description
`interface`	*`Interface name`*	Rule for specified interface name will be added.
`port`	*`Integer`*	Rule for specified port will be added/deleted.
`to-address`	*`IP-address`*	Rule for specified destination address will be added.
`to-port`	*`Integer`*	Rule for specified destination port will be added.
`and forward`	*`Keyword`*	Lists of forwarding and redirecting rules will be cleared.
`index`	*`Integer`*	Rule with specified number in the list will be removed.

History

Version	Description
2.00	The **upnp redirect** command has been introduced.

3.130 user

Description

Access to a group of commands to configure user account parameters. If specified user is not found, the command tries to create it.

Note: Account with reserved name `admin` can not be removed. In addition, the `admin` user can not lose the access right to command line.

Command with **no** prefix removes user account.

Prefix no

Yes

Change settings

Yes

Multiple input

Yes

Group entry

`(config-user)`

Synopsis

(config)> **`user`** *‹name›*

(config)> **`no user`** *‹name›*

Arguments

Argument	Value	Description
`name`	*`String`*	Tne user name.

History

Version	Description
2.00	The **user** command has been introduced.

3.130.1 user password

Description

Set the user password. The password is stored as MD5-hash, computed from the "*user*:*realm*:*password*" string. *realm* is the device model name from `startup-config.txt` file.

The command takes open string or hash-function value as argument. Saved password is used for user authentication.

Command with **no** prefix removes the password so that the user can access to the device unauthenticated.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(config-user)> **password** (**md5** *‹hash›* | *‹password›*)

(config-user)> **no password**

Arguments

Argument	Value	Description
hash	*String*	MD5-hash value.
password	*String*	Value of the password in open form, from which the hash value is calculated automatically.

Example

```
(config-user)> password 1111
Core::Authenticator: Password set has been changed for user ►
"test".
```

History

Version	Description
2.00	The **user password** command has been introduced.

3.130.2 user tag

Description

Assign a special tag to the user account, which presence is checked at the time of user authorization as well as performing any action in the system. Set of permitted tag values depends on the system functionality. The full list is shown in the table below.

Several different tags can be assigned to one account by entering the command several times. Each tag can be viewed as granting or revoking certain permissions.

Command with **no** prefix removes the specified tag.

Note: `admin` account cannot be tagged `readonly` or untagged `cli` or `ssh`.

Prefix no Yes

Change settings Yes

Multiple input Yes

Synopsis

(config-user)> **tag** *‹tag›*

(config-user)> **no tag** [*‹tag›*]

Arguments

Argument	Value	Description
`tag`	`cli`	Access to the command line (TELNET and SSH).
	`readonly`	Restrict commands that change the settings.
	`http-proxy`	Access to the HTTP proxy.
	`http`	Access to the Web-interface.
	`afp`	Access to USB drives via Apple File Protocol.
	`printers`	Access to USB printers via SMB/CIFS.
	`cifs`	Connection to the Windows files and printers service.
	`vpn-dlna`	Access to the *DLNA* for PPTP, L2TP/IPSec, SSTP tunnels.
	`ftp`	Connection to an integrated FTP-server.
	`ipsec-xauth`	Connection to an integrated IPsec/XAuth-server.
	`ipsec-l2tp`	Connection to an integrated L2TP/IPSec-server.
	`opt`	Access to services managed by OptWare.
	`sftp`	Access to SFTP file server.
	`sstp`	Connection to an integrated SSTP-server.
	`torrent`	Access to the BitTorrent client GUI.
	`vpn`	Connection to an integrated PPTP-server.
	`webdav`	Access to WebDAV file server.

Example

```
(config-user)> tag cli
Core::Authenticator: User "admin" tagged with "cli".
```

```
(config-user)> tag readonly
Core::Authenticator: User "my" tagged with "readonly".
```

```
(config-user)> tag http-proxy
Core::Authenticator: User "admin" tagged with "http-proxy".

(config-user)> tag http
Core::Authenticator: User "admin" tagged with "http".

(config-user)> tag afp
Core::Authenticator: User "test" tagged with "afp".

(config-user)> tag printers
Core::Authenticator: User "admin" tagged with "printers".

(config-user)> tag cifs
Core::Authenticator: User "admin" tagged with "cifs".

(config-user)> tag vpn-dlna
Core::Authenticator: User "enpa" tagged with "vpn-dlna".

(config-user)> tag ftp
Core::Authenticator: User "admin" tagged with "ftp".

(config-user)> tag ipsec-xauth
Core::Authenticator: User "admin" tagged with "ipsec-xauth".

(config-user)> tag ipsec-l2tp
Core::Authenticator: User "admin" tagged with "ipsec-l2tp".

(config-user)> tag opt
Core::Authenticator: User "admin" tagged with "opt".

(config-user)> tag sftp
Core::Authenticator: User "test" tagged with "sftp".

(config-user)> tag sstp
Core::Authenticator: User "admin" tagged with "sstp".

(config-user)> tag torrent
Core::Authenticator: User "admin" tagged with "torrent".

(config-user)> tag vpn
Core::Authenticator: User "admin" tagged with "vpn".

(config-user)> tag webdav
Core::Authenticator: User "test" tagged with "webdav".

(config-user)> no tag readonly
Core::Authenticator: User "admin": "readonly" tag deleted.
```

History

Version	Description
2.00	The **user tag** command has been introduced.
2.04	The **vpn** tag has been added.
2.06	The **opt**, **ipsec-xauth** tags have been added.
2.10	The **http-proxy** tag has been added.

2.11	The **ipsec-l2tp** tag has been added.
2.12	The **sstp** tag has been added.
3.04	The **vpn-dlna sftp** and **webdav** tags have been added.

3.131 vpn-server

Description Access to a group of commands to configure VPN-server parameters.

Prefix no No

Change settings No

Multiple input No

Group entry `(vpn-server)`

Synopsis

(config)> **`vpn-server`**

History

Version	Description
2.04	The **vpn-server** command has been introduced.

3.131.1 vpn-server dhcp route

Description Assign a route which is transmitted in DHCP INFORM messages to the VPN-server clients.

Command with **no** prefix cancels the specified route. If you use no arguments, the entire list of routes will be cleared.

Prefix no Yes

Change settings Yes

Multiple input Yes

Synopsis

(vpn-server)> **`dhcp route`** *‹address› ‹mask›*

(vpn-server)> **`no dhcp route`** [*‹address› ‹mask›*]

Arguments

Argument	Value	Description
`address`	*IP-address*	Network client address.
`mask`	*IP-mask*	Network client mask. There are two ways to enter the mask: the canonical form (for example, 255.255.255.0) and the form of prefix bit length (for example, /24).

Example

```
(vpn-server)> dhcp route 192.168.2.0/24
VpnServer::Manager: Added DHCP INFORM route to ►
192.168.2.0/255.255.255.0.
```

```
(vpn-server)> no dhcp route
VpnServer::Manager: Cleared DHCP INFORM routes.
```

History

Version	Description
2.12	The **vpn-server dhcp route** command has been introduced.

3.131.2 vpn-server interface

Description

Bind VPN-server to the specified interface.

Command with **no** prefix unbinds the interface.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(vpn-server)> **interface** ‹*interface*›

(vpn-server)> **no interface**

Arguments

Argument	Value	Description
interface	*Interface name*	Full interface name or an alias. You can see the list of available interfaces with help of **interface** [Tab] command.

Example

```
(vpn-server)> interface [Tab]

 Usage template:
        interface {interface}

   Choose:
         GigabitEthernet1
                      ISP
 WifiMaster0/AccessPoint2
 WifiMaster1/AccessPoint1
 WifiMaster0/AccessPoint3
 WifiMaster0/AccessPoint0
              AccessPoint
```

```
(vpn-server)> interface FastEthernet0/Vlan1
VpnServer::Manager: Bound to FastEthernet0/Vlan1
```

```
(vpn-server)> no interface
VpnServer::Manager: Reset interface binding.
```

History

Version	Description
2.04	The **vpn-server interface** command has been introduced.

3.131.3 vpn-server ipv6cp

Description

Enable IPv6 support. DHCP IPv6 pools are created for each VPN-server. By default, the setting is disabled.

Command with **no** prefix disables IPv6 support.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(vpn-server)> **ipv6cp**

(vpn-server)> **no ipv6cp**

Example

```
(vpn-server)> ipv6cp
VpnServer::Manager: IPv6 control protocol enabled.
```

```
(vpn-server)> no ipv6cp
VpnServer::Manager: IPv6 control protocol disabled.
```

History

Version	Description
3.00	The **vpn-server ipv6cp** command has been introduced.

3.131.4 vpn-server lcp echo

Description

Specify the testing rules of the PPTP connections with *LCP* echo tools.

Command with **no** prefix disables *LCP* echo.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(vpn-server)> **lcp echo** ‹*interval*› ‹*count*› [**adaptive**]

(vpn-server)> **no lcp echo**

Arguments

Argument	Value	Description
`interval`	*Integer*	Interval between sending *LCP* echo, in seconds. If within the specified time interval there is no *LCP* echo request from the remote location, the same request will be sent there asking for response *LCP* reply.
`count`	*Integer*	The number of consecutive requests *LCP* echo sent, for which no response *LCP* reply was received. If count of *LCP* echo requests goes unanswered, the connection is terminated.
`adaptive`	*Keyword*	Pppd will send LCP echo-request frames only if no traffic was received from the peer since the last echo-request was sent.

Example

```
(vpn-server)> lcp echo 5 3
LCP echo parameters updated.
```

History

Version	Description
2.06	The **vpn-server lcp echo** command has been introduced.

3.131.5 vpn-server lockout-policy

Description

Set VPN-server bruteforce detection parameters. By default, feature is enabled.

Command with **no** prefix disables bruteforce detection.

Prefix no

Yes

Change settings

Yes

Multiple input

No

Synopsis

(vpn-server)> **vpn-server lockout-policy** ‹*threshold*› [‹*duration*› [‹*observation-window*›]]

(vpn-server)> **no vpn-server lockout-policy**

Arguments

Argument	Value	Description
`threshold`	*Integer*	The number of failed attempts to log in. By default, 5 value is used.
`duration`	*Integer*	An authorization ban duration for the specified IP in minutes. By default, 15 value is used.
`observation-window`	*Integer*	Duration of suspicious activity observation in minutes. By default, 3 value is used.

Example

```
(vpn-server)> lockout-policy 10 30 2
VpnServer::Manager: Bruteforce detection is reconfigured.
```

```
(vpn-server)> no lockout-policy
VpnServer::Manager: Bruteforce detection is disabled.
```

History

Version	Description
3.01	The **vpn-server lockout-policy** command has been introduced.

3.131.6 vpn-server mppe

Description

Set mode for *MPPE* encryption. 40-bit key is used by default.

Command with **no** prefix disables selected mode.

Prefix no

Yes

Change settings

Yes

Multiple input

Yes

Synopsis

(vpn-server)> **mppe** *‹mode›*

(vpn-server)> **no mppe** *‹mode›*

Arguments

Argument	Value	Description
mode	40	Length of the encryption key is 40 bits.
	128	Length of the encryption key is 128 bits.

Example

```
(vpn-server)> mppe 40
VpnServer::Manager: Set encryption 40.
```

History

Version	Description
2.05	The **vpn-server mppe** command has been introduced.

3.131.7 vpn-server mppe-optional

Description

Enable *MPPE* encryption.

Command with **no** prefix disables encryption.

Prefix no

Yes

Change settings

Yes

Multiple input No

Synopsis

(vpn-server)> **mppe-optional**

(vpn-server)> **no mppe-optional**

Example

```
(vpn-server)> mppe-optional
VpnServer::Manager: Unencrypted connections enabled.
```

History

Version	Description
2.04	The **vpn-server mppe-optional** command has been introduced.

3.131.8 vpn-server mru

Description Set *MRU* value to be transmitted to PPTP-server. By default, `1350` value is used.

Command with **no** prefix resets value to default.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(vpn-server)> **mru** *‹value›*

(vpn-server)> **no mru**

Arguments

Argument	Value	Description
`value`	*Integer*	*MRU* value. Can take values from 128 to 1500 inclusively.

Example

```
(vpn-server)> mru 200
VpnServer::Manager: mru set to 200.
```

History

Version	Description
2.04	The **vpn-server mru** command has been introduced.

3.131.9 vpn-server mtu

Description Set *MTU* value to be transmitted to PPTP-server. By default, `1350` value is used.

Command with **no** prefix resets value to default.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(vpn-server)> **mtu** *‹value›*

(vpn-server)> **no mtu**

Arguments

Argument	Value	Description
value	*Integer*	*MTU* value. Can take values from 128 to 1500 inclusively.

Example

```
(vpn-server)> mtu 200
VpnServer::Manager: mtu set to 200.
```

History

Version	Description
2.04	The **vpn-server mtu** command has been introduced.

3.131.10 vpn-server multi-login

Description Allow connection to VPN-server for multiple users from one account.

Command with **no** prefix disables this feature.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(vpn-server)> **multi-login**

(vpn-server)> **no multi-login**

Example

```
(vpn-server)> multi-login
VpnServer::Manager: multi login enabled.
```

History

Version	Description
2.04	The **vpn-server multi-login** command has been introduced.

3.131.11 vpn-server pool-range

Description Assign a pool of addresses for the clients that connect to the VPN-server.

Command with **no** prefix removes a pool.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(vpn-server)> **pool-range** *‹begin›* [*‹size›*]

(vpn-server)> **no pool-range**

Arguments

Argument	Value	Description
begin	*IP-address*	Start address of pool.
size	*Integer*	Pool size. Can take values in the range from 1 to 64 inclusively. If the size is not specified, it is determined automatically depending on the device.

Example

```
(vpn-server)> pool-range 172.168.1.22 20
VpnServer::Manager: Configured pool range 172.168.1.22 to ►
172.168.1.41.
```

```
(vpn-server)> no pool-range
VpnServer::Manager: Reset pool range.
```

History

Version	Description
2.04	The **vpn-server pool-range** command has been introduced.

3.131.12 vpn-server static-ip

Description Bind IP-address to the user. User account must have vpn tag.

Command with **no** prefix removes binding.

Prefix no Yes

Change settings Yes

Multiple input Yes

Synopsis

(vpn-server)> **static-ip** *‹name› ‹address›*

(vpn-server)> **no static-ip** *‹name›*

Arguments

Argument	Value	Description
name	*String*	Username.

Argument	Value	Description
address	*IP-address*	IP-address to bind.

Example

```
(vpn-server)> static-ip test 172.16.1.35
VpnServer::Manager: Static IP 172.16.1.35 assigned to user "test".
```

```
(vpn-server)> static-ip test
VpnServer::Manager: Static IP address removed for user "test".
```

History

Version	Description
2.04	The **vpn-server static-ip** command has been introduced.

3.132 yandexdns

Description Access to a groupe of commands to configure *Yandex.DNS* profiles.

Prefix no No

Change settings No

Multiple input No

Group entry (yandexdns)

Synopsis

(config)> **yandexdns**

History

Version	Description
2.01	The **yandexdns** command has been introduced.

3.132.1 yandexdns assign

Description Assign types to the hosts. By default safe type is used for all hosts. default type can be assigned to a single host.

Command with **no** prefix resets setting to default.

Prefix no Yes

Change settings Yes

Multiple input Yes

Synopsis

(yandexdns)> **assign** [‹*host*›] ‹*type*›

(yandexdns)> **no assign** [‹*host*›]

Arguments

Argument	Value	Description
`host`	*`MAC-address`*	Host to which type of filtering is applied. If not specified, the type is applied to all hosts.
`type`	`default`	No filtering used.
	`safe`	Protection against malicious and phishing websites.
	`family`	Access denied to malicious and phishing websites, as well as to resources for adults.

History

Version	Description
2.01	The **yandexdns assign** command has been introduced.

3.132.2 yandexdns check-availability

Description Check availability of *Yandex.DNS* service.

Prefix no No

Change settings No

Multiple input No

Synopsis

(yandexdns)> **`check-availability`**

Example

```
(yandexdns)> check-availability
available
```

History

Version	Description
2.04	The **yandexdns check-availability** command has been introduced.

3.132.3 yandexdns enable

Description Enable *Yandex.DNS* service.

Command with **no** prefix disables the service.

Prefix no Yes

Change settings Yes

Multiple input No

Synopsis

(yandexdns)> **`enable`**

(yandexdns)> **no enable**

Example

```
(yandexdns)>  enable
YandexDns::Client: Yandex DNS is enabled.
```

History

Version	Description
2.01	The **yandexdns enable** command has been introduced.

Glossary

Address and Control Field Compression	*LCP* configuration option that provides a method to negotiate the compression of the Data Link Layer Address and Control fields.
Address Resolution Protocol	is a protocol for mapping an Internet Protocol address (IP address) to a physical machine address that is recognized in the local network. For example, in IP Version 4, the most common level of IP in use today, an address is 32 bits long. In an Ethernet local area network, however, addresses for attached devices are 48 bits long. (The physical machine address is also known as a Media Access Control or MAC address.) A table, usually called the ARP cache, is used to maintain a correlation between each MAC address and its corresponding IP address. ARP provides the protocol rules for making this correlation and providing address conversion in both directions.
AdGuard DNS	service of AdGuard company to protect home network. Provides three protection modes: • default mode: no bloked sites • standard mode: blocking advertising, tracking and phishing; • family mode: blocking advertising, tracking, phishing and adult sites, providing secure search in the browser.
Authenticated Encryption with Associated Data	this form of encryption which simultaneously assure the confidentiality and authenticity of data. AEAD is a variant of AE that allows a recipient to check the integrity of both the encrypted and unencrypted information in a message.
Automatic Certificate Management Environment	is a communications protocol for automating interactions between certificate authorities and their users' web servers, allowing the automated deployment of public key infrastructure at very low cost. It was designed by the Internet Security Research Group (ISRG) for their Let's Encrypt service.
Band Steering	is a feature that encourages dual-band capable wireless clients to connect to the less crowded 5GHz network, and leave the 2.4GHz network available for those clients who support 2.4GHz only; thus, Wi-Fi performance can be improved for all clients.
Challenge-Handshake Authentication Protocol	widely used algorithm for authentication, which provides the transfer of indirect information about user password. CHAP provides better security than *Password Authentication Protocol*.

Change of Authorization	is a provides a mechanism for changing RADIUS authentication and authorization session attributes. Allows you to set up an active client session.
Cloudflare DNS	is a service of Cloudflare company to protect home network. Provides three protection modes: • default mode: no bloked sites; • standard mode: secure dns resolving, no blocking; • malware mode: blocking malware; • family mode: blocking malware and adult sites.
Command Line Interface	is a user interface to a computer's operating system or an application in which the user responds to a visual prompt by typing in a command on a specified line, receives a response back from the system, and then enters another command, and so forth.
Common Applications Kept Enhanced	is a shaping-capable queue discipline which uses both AQM and FQ. It combines COBALT, which is an AQM algorithm combining Codel and BLUE, a shaper which operates in deficit mode, and a variant of DRR++ for flow isolation. 8-way set-associative hashing is used to virtually eliminate hash collisions. Priority queuing is available through a simplified diffserv implementation. CAKE uses a deficit-mode shaper, which does not exhibit the initial burst typical of token-bucket shapers. It will automatically burst precisely as much as required to maintain the configured throughput.
Compression Control Protocol	is used for establishing and configuring data compression algorithms over *PPP*.
Dead Peer Detection	is a method that network devices use to verify the current existence and availability of other peer devices.
Device Privacy Notice	is a Keenetic device privacy notice on data processing.
DHCP	is a network protocol that is used to configure network devices so that they can communicate on an IP network. A DHCP client uses the DHCP protocol to acquire configuration information, such as an IP address, a default route, and one or more DNS server addresses from a DHCP server. The DHCP client then uses this information to configure its host. Once the configuration process is complete, the host is able to communicate on the Internet.
DHCP-server	manages a pool of IP addresses and information about client configuration parameters such as default gateway, domain name, the name servers, other servers such as time servers, and so forth. On receiving a valid request, the server assigns the computer an IP address, a lease (length of time the allocation is valid), and other IP configuration parameters, such as the subnet mask and the default gateway. Depending on implementation, the DHCP server may have three methods of allocating IP-addresses:

- *dynamic allocation*: A network administrator assigns a range of IP addresses to DHCP, and each client computer on the LAN is configured to request an IP address from the DHCP server during network initialization. The request-and-grant process uses a lease concept with a controllable time period, allowing the DHCP server to reclaim (and then reallocate) IP addresses that are not renewed.
- *automatic allocation*: The DHCP server permanently a free IP address to a requesting client from the range defined by the administrator. This is like dynamic allocation, but the DHCP server keeps a table of past IP address assignments, so that it can preferentially assign to a client the same IP address that the client previously had.
- *static allocation*: The DHCP server allocates an IP address based on a table with MAC address/IP address pairs, which are manually filled in (perhaps by a network administrator). Only requesting clients with a MAC address listed in this table will be allocated an IP address. This feature (which is not supported by all DHCP servers) is variously called Static DHCP Assignment (by DD-WRT),fixed-address (by the dhcpd documentation), Address Reservation (by Netgear), DHCP reservation or Static DHCP (by Cisco/Linksys), and IP reservation or MAC/IP binding (by various other router manufacturers).

Diffie-Hellman
: is that part of the *IKE* protocol used for exchanging the material from which the symmetrical keys are built. The Diffie-Hellman algorithm builds an encryption key known as a "shared secret" from the private key of one party and the public key of the other. Since the *IPsec* symmetrical keys are derived from this DH key shared between the peers, at no point are symmetric keys actually exchanged.

DLNA
: standard that allows compatible devices to transfer media content (images, music, videos) over the home network and display it in real time. This technology is to connect home computers, mobile phones, notebooks and home electronics in a single digital network. DLNA-certified devices can be configured and combined in a home network automatically.

Domain Name System
: is a hierarchical distributed naming system for computers, services, or any resource connected to the Internet or a private network. It associates various information with domain names assigned to each of the participating entities. A Domain Name Service resolves queries for these names into IP addresses for the purpose of locating computer services and devices worldwide. By providing a worldwide, distributed keyword-based redirection service, the Domain Name System is an essential component of the functionality of the Internet.

DNS over HTTPS
: is a domain name system, computer distributed system for obtaining information about domains using secure data transfer between internet nodes resolution via the HTTPS protocol. The method is to increase user privacy and security by preventing eavesdropping and manipulation of DNS data by man-in-the-middle attacks. The standard is described in RFC 8484[1].

[1] https://tools.ietf.org/html/rfc8484

DNS over TLS	is a domain name system, computer distributed system for obtaining information about domains using secure data transfer between internet nodes. The standard is described in RFC 7858[2] and RFC 8310[3].
DNS rebinding	is a method of manipulating resolution of domain names. In this attack, a malicious web page causes visitors to run a client-side script that attacks machines elsewhere on the network. This attack can be used to breach a private network by causing the victim's web browser to access computers at private IP addresses and return the results to the attacker.
Encapsulating Security Payload	is a member of the *IPsec* protocol suite. In IPsec it provides origin authenticity, integrity, and confidentiality protection of packets.
End-user license agreement	is a legal contract between a software application author or publisher and the user of that application.
Fast Transition	is a new concept of roaming where the initial handshake with the new AP is done even before the client roams to the target AP.
Fair Queuing Controlled Delay	is queuing discipline that combines Fair Queuing with the CoDel AQM scheme. FQ_Codel uses a stochastic model to classify incoming packets into different flows and is used to provide a fair share of the bandwidth to all the flows using the queue. Each such flow is managed by the CoDel queuing discipline.
Fully Qualified Domain Name	is a domain name that specifies its exact location in the tree hierarchy of the *Domain Name System*. It specifies all domain levels, including the top-level domain and the root zone. A fully qualified domain name is distinguished by its lack of ambiguity: it can be interpreted only in one way.
Full Cone NAT	also Static NAT, one to one NAT, port forwarding is the only type of NAT where the port is permanently open and allows inbound connections from any external host. A full cone NAT maps a public IP address and port to a LAN IP and port. Any external host can send data to the LAN IP through the mapped NAT IP and port. If it tries to send data through a different port it will fail. Static NAT is required when a network device on a private network must be accessible from the Internet.
Generic Routing Encapsulation	is a tunneling protocol developed by Cisco Systems that can encapsulate a wide variety of network layer protocols inside virtual point-to-point links over an Internet Protocol network.
Hash Message Authentication Code	is a specific construction for calculating a message authentication code (MAC) involving a cryptographic hash function in combination with a secret cryptographic key. As with any MAC, it may be used to simultaneously verify both the data integrity and the authentication of a message. Any cryptographic hash function, such as MD5 or SHA-1, may be used in the calculation of an HMAC; the resulting MAC algorithm is termed HMAC-MD5 or HMAC-SHA1 accordingly. The cryptographic

[2] https://tools.ietf.org/html/rfc7858
[3] https://tools.ietf.org/html/rfc8310

	strength of the HMAC depends upon the cryptographic strength of the underlying hash function, the size of its hash output, and on the size and quality of the key.
Idempotence	is the property of certain operations in computer science, that they can be applied multiple times without changing the result beyond the initial application.
Inter-Access Point Protocol	is a standard IEEE 802.11F protocol exchange of service information for data transfer between access points. The protocol is responsible for combining the wireless network, secure data exchange between the current access point and the new access point in the specified period.
Internet Control Message Protocol	is a message control and error-reporting protocol between a host server and a gateway to the Internet. ICMP uses Internet Protocol (IP) datagrams, but the messages are processed by the IP software and are not directly apparent to the application user.
Internet Group Management Protocol	is an Internet protocol that provides a way for an Internet computer to report its multicast group membership to adjacent routers. Multicasting allows one computer on the Internet to send content to multiple other computers. Multicasting can be used for streaming media to an audience that has "tuned in" by setting up a multicast group membership.
Internet Key Exchange	is a standard protocol IPsec, used to ensure the safety of interaction in virtual private networks. IKE purpose is to establish a secure authenticated communication channel by using the *Diffie–Hellman* key exchange algorithm to generate a shared secret key to encrypt further *IPsec* communications.
Internet Protocol	is the principal communications protocol in the Internet. The first major version of IP, Internet Protocol Version 4 (IPv4), is the dominant protocol of the Internet. Its successor is Internet Protocol Version 6 (IPv6).
Internet Protocol Control Protocol	is a network control protocol for establishing and configuring Internet Protocol over a *Point-to-Point Protocol* (PPP) link. IPCP uses the same packet exchange mechanism as the Link Control Protocol. IPCP packets may not be exchanged until PPP has reached the Network-Layer Protocol phase, and any IPCP packets received before this phase is reached should be silently discarded.
Internet Protocol Security	commonly called IPsec, is a protocol suite for secure *Internet Protocol* (IP) communications by authenticating and encrypting each IP packet of a communication session. IPsec includes protocols for establishing mutual authentication between agents at the beginning of the session and negotiation of cryptographic keys to be used during the session. IPsec can be used in protecting data flows between a pair of hosts (host-to-host), between a pair of security gateways (network-to-network), or between a security gateway and a host (network-to-host). Internet Protocol security (IPsec) uses cryptographic security services to protect communications over Internet Protocol (IP) networks. IPsec supports network-level peer authentication, data origin

	authentication, data integrity, data confidentiality (encryption), and replay protection.
IPsec Passthrough	is technology that allows VPN-traffic to pass through NAT.
IPsec Security Association	is fundamental to IPsec. An SA is a relationship between two or more entities that describes how the entities will use security services to communicate securely. Each IPsec connection can provide encryption, integrity, authenticity, or all three. When the security service is determined, the two IPsec peers must determine exactly which algorithms to use (for example, DES or 3DES for encryption, MD5 or SHA for integrity). After deciding on the algorithms, the two devices must share session keys. The Security Association is the method that IPsec uses to track all the particulars concerning a given IPSec communication session.
IP in IP	is an IP tunneling protocol that encapsulates one IP packet in another IP packet.
IPv6CP	is responsible for configuring, enabling, and disabling the IPv6 protocol modules on both ends of the *Point-to-Point* (PPP) link. IPv6CP uses the same packet exchange mechanism as the *Link Control Protocol*. IPv6CP packets may not be exchanged until PPP has reached the Network-Layer Protocol phase. IPv6CP packets received before this phase is reached should be silently discarded.
Layer 2 Tunneling Protocol	is a tunneling protocol used to support virtual private networks (VPNs) or as part of the delivery of services by ISPs. It does not provide any encryption or confidentiality by itself. Rather, it relies on an encryption protocol that it passes within the tunnel to provide privacy.
Link Control Protocol	establishes, configures, and tests data-link Internet connections in the *Point-to-Point Protocol* (PPP). Before establishing communications over a point-to-point link, each end of the PPP link must send out LCP packets. The LCP packet either accepts or rejects the identity of its linked peer, agrees up on packet size limits, and looks for common misconfiguration errors. LCP packets are divided into three classes: • Link configuration packets used to establish and configure a link • Link termination packets used to terminate a link • Link maintenance packets used to manage and debug a link
Link Layer Discovery Protocol	is a vendor-neutral link layer protocol in the Internet Protocol Suite used by network devices for advertising their identity, capabilities, and neighbors on an IEEE 802 local area network, principally wired Ethernet. Information gathered with LLDP is stored in the device as a management information database (MIB) and can be queried with the Simple Network Management Protocol (SNMP).

Maximum Receive Unit	is the maximum size (in bytes) of the frame, which can be received at the data link layer of communication protocol.
Maximum Segment Size	is a parameter of the options field of the *TCP* header that specifies the largest amount of data, specified in bytes, that a computer or communications device can receive in a single TCP segment. It does not count the TCP header or the IP header.
Maximum Transmission Unit	is the largest size packet or frame, specified in octets (eight-bit bytes), that can be sent in a packet- or frame-based network such as the Internet. The Internet's Transmission Control Protocol (TCP) uses the MTU to determine the maximum size of each packet in any transmission. Most computer operating systems provide a default MTU value that is suitable for most users. In general, Internet users should follow the advice of their Internet service provider (ISP) about whether to change the default value and what to change it to.
Microsoft Point-to-Point Encryption	encrypts data in *Point-to-Point Protocol* based dial-up connections or Point-to-Point Tunneling Protocol (PPTP) connections. 128-bit key (strong), 56-bit key, and 40-bit key (standard) MPPE encryption schemes are supported. MPPE provides data security for the PPTP connection that is between the VPN client and the VPN server.
Modular Wi-Fi System	a system that allows several Keenetic devices to be combined into a single Internet space distributed over an area. One of the devices is defined as the controller, the others as the members.
Multicast DNS	is a way of using familiar DNS programming interfaces, packet formats and operating semantics, in a small network where no conventional DNS server has been installed. The mDNS protocol uses IP multicast UDP packets, and is implemented by the Apple Bonjour and open source Avahi software packages.
Network Access Control List	rules that are applied to IP interfaces that are available on a router, each with a list of hosts or networks that are permitted or denied to use the service. Access control lists can be configured to control both inbound and outbound traffic.
Network Flow	network protocol for network traffic accounting, uses UDP or SCTP protocols to send traffic data to the collector. Collector is an application that runs on a server and collects statistics received from sensors. A sensor is a device that collects traffic statistics and sends it to a collector. The sensor can be a Cisco third-level router or switch.
Network Time Protocol	is a protocol that is used to synchronize computer clock times in a network of computers. Developed by David Mills at the University of Delaware, NTP is now an Internet standard. In common with similar protocols, NTP uses Coordinated Universal Time (UTC) to synchronize computer clock times to a millisecond, and sometimes to a fraction of a millisecond.
Network Traffic Classification Engine	also DPI, Deep Deep Packet Inspection is a technology for accumulating statistics and inspecting network packets based on their contents. Deep Packet Inspection analyzes not

	only packet headers, but also the full content of traffic at OSI layers 2 and above. Deep Packet Inspection can determine which network application has generated or received data, collecting detailed connection statistics for each device and application individually. With quality of service Deep Packet Inspection controls the transmission speed of individual packets by raising or lowering it. The Traffic Classification Engine component operates completely independently and does not make any calls to external services.
Opportunistic Wireless Encryption	is an extension of the IEEE 802.11 standard, similar encryption method Simultaneous Authentication of Equals (SAE). This encryption method provides users with better protection when connected to an open Wi-Fi network.
Password Authentication Protocol	is an authentication protocol that uses a password. PAP is used by *Point-to-Point Protocol* to validate users before allowing them access to the remote network. PAP transmits unencrypted ASCII passwords over the network and is therefore considered insecure.
Protected Extensible Authentication Protocol	is a protocol that encapsulates the Extensible Authentication Protocol (EAP) within an encrypted and authenticated Transport Layer Security (TLS) tunnel. The purpose was to correct deficiencies in EAP; EAP assumed a protected communication channel, such as that provided by physical security, so facilities for protection of the EAP conversation were not provided.
Perfect Forward Secrecy	is a property of secure communication protocols: a secure communication protocol is said to have forward secrecy if compromise of long-term keys does not compromise past session keys. PFS protects past sessions against future compromises of secret keys or passwords.
Ping Check	performs ICMP and TCP based tests to verify if the internet connection is working fine. Test results may be used to switch between primary and backup connections.
Point-to-Point Protocol	is a protocol used to establish a direct connection between two nodes. It can provide connection authentication, transmission encryption, and compression. PPP is used over many types of physical networks including serial cable, phone line, cellular telephone, specialized radio links, and fiber optic links. After the link has been established, additional network (layer 3) configuration may take place. Most commonly, the *Internet Protocol Control Protocol* (IPCP) is used.
Preamble	it is the first part of the Physical Layer Convergence Protocol/Procedure (PLCP) Protocol Data Unit (PDU). A header is the remaining part of the data packets and has more information identifying the modulation scheme, transmission rate, and length of time to transmit the whole data frame. The Preamble type in IEEE 802.11 based wireless communication defines the length of the CRC (Cyclic Redundancy Check) block for

	communication between the Access Point and roaming wireless adapters. Long preamble: • PLCP with long preamble is transmitted at 1 Mbps regardless of transmit rate of data frames • Total long preamble transfer time is a constant at 192 usec • Compatible with legacy IEEE 802.11 systems running at 1 and 2 Mbps Short preamble: • Preamble is transmitted at 1 Mbps and header at 2 Mbps • Total short preamble transfer time is a constant at 96 usec • Not compatible with legacy IEEE 802.11 systems operating at 1 and 2 Mbps
Protected Management Frames	IEEE 802.11w is the Protected Management Frames standard for the IEEE 802.11 family of standards. This functionality is necessary to improve security by ensuring data confidentiality in control frames.
Protocol Field Compression	is a method to negotiate the compression of the *PPP* Protocol field. By default, all implementations MUST transmit packets with two octet PPP Protocol fields.
Pseudo-Random Function	is similar to an integrity algorithm, but instead of being used to authenticate messages, it is only used to provide randomness for purposes such as keying material. PRFs are primarily used with an authenticated encryption algorithm type such as AES-GCM.
Radio Resource Management	is the system level management of co-channel interference, radio resources, and other radio transmission characteristics in wireless communication systems. RRM includes control parameters such as transmit power, user allocation, beamforming, data rates, handover criteria, modulation scheme, coding scheme errors.
Remote Authentication in Dial-In User Service	is a protocol to implement authentication, authorization, and resource collection. It is used for charging the used resources by a specific user. Used to authenticate users on open Wi-Fi wireless networks.
Restricted NAT	also Dynamic NAT works in the same way as a *Full Cone NAT* but applies additional restrictions based on an IP address. The internal client must first have sent packets to IP address (X) before it can receive packets from X. In terms of restrictions the only requirement is that packets come in on the mapped port and from an IP address that the internal client has sent packets to.
Secure Socket Tunneling Protocol	is a type of VPN tunnel that utilizes an SSL 3.0 channel to send PPP or L2TP traffic. SSL allows for transmission and data encryption, as well

	as traffic integrity checking. Due to this, SSTP can pass through most firewalls and proxy servers by using the SSL channel over TCP port 443.
Service Set Identifier	is a sequence of characters that uniquely names a wireless local area network (WLAN). An SSID is sometimes referred to as a "network name". This name allows stations to connect to the desired network when multiple independent networks operate in the same physical area.
Shared key	is a mode by which a computer can gain access to a wireless network that uses the Wired Equivalent Privacy protocol. With Shared Key, a computer equipped with a wireless modem can fully access any WEP network and exchange encrypted or unencrypted data.
Simple Network Management Protocol	is an Internet-standard protocol for collecting and organizing information about managed devices on IP networks and for modifying that information to change device behavior. Devices that typically support SNMP include routers, switches, servers, workstations, printers, modem racks and more.
Transmission Control Protocol	is a core protocol of the *Internet Protocol* suite. TCP provides reliable, ordered, and error-checked delivery of a stream of octets between applications running on hosts communicating over an IP network.
Universal Access Method	is a method that allows a subscriber to access a wireless Wi-Fi network. The Internet browser will open a login page where the user should fill in his credentials before he can access. UAM uses the RADIUS client and the RADIUS server for authorization.
User Datagram Protocol	is a core protocol of the *Internet Protocol* suite. UDP uses a simple connectionless transmission model with a minimum of protocol mechanism. It has no handshaking dialogues, and thus exposes the user's program to any unreliability of the underlying network protocol. There is no guarantee of delivery, ordering, or duplicate protection. Time-sensitive applications often use UDP because dropping packets is preferable to waiting for delayed packets, which may not be an option in a real-time system.
udpxy	is a UDP-to-HTTP multicast traffic relay daemon: it forwards UDP traffic from a given multicast subscription to the requesting HTTP client.
Universal Plug and Play	is a standard that uses Internet and Web protocols to enable devices such as PCs, peripherals, intelligent appliances, and wireless devices to be plugged into a network and automatically know about each other. With UPnP, when a user plugs a device into the network, the device will configure itself, acquire a TCP/IP address, and use a discovery protocol based on the HTTP to announce its presence on the network to other devices.
Virtual LAN	is a local area network with a definition that maps workstations on some other basis than geographic location (for example, by department, type of user, or primary application). The virtual LAN controller can change or add workstations and manage loadbalancing and bandwidth allocation more easily than with a physical picture of the LAN.

Web Distributed Authoring and Versioning	is a extension of the Hypertext Transfer Protocol (HTTP) that allows clients to perform remote Web content authoring operations. Supports web server authentication and SSL encryption for HTTPS using the default TCP port 443.
Web Proxy Auto-Discovery Protocol	is a method used by clients to locate the URL of a configuration file using DHCP and/or DNS discovery methods. Once detection and download of the configuration file is complete, it can be executed to determine the proxy for a specified URL.
WireGuard	is a free and open-source software application and virtual private network (VPN) protocol to create secure point-to-point connections in routed configurations. WireGuard protocol uses modern cryptography options Curve25519 for key exchange, ChaCha20 for encryption, and Poly1305 for data authentication, SipHash for hashtable keys, and BLAKE2s for hashing. Supports layer 3 for both protocols IPv4 and IPv6.
Wi-Fi Multimedia	previously known as Wireless Multimedia Extensions (WME), is a subset of the 802.11e wireless LAN (WLAN) specification that enhances quality of service (QoS) on a network by prioritizing data packets according to four access categories (AC). Ranging from highest priority to lowest, these categories are: voice (AC_VO), video (AC_VI), best effort (AC_BE), and background (AC_BK). WMM also features a Power Save certification that helps small devices on a network conserve battery life. Power Save allows small devices, such as phones and PDAs, to transmit data while in a low-power "dozing" status. The certification gives software developers and hardware manufacturers a way to fine-tune battery use in the ever-increasing number of small devices that have Wi-Fi capabilities.
Wi-Fi Protected Access	Wi-Fi Protected Access II (WPA2), and Wi-Fi Protected Access 3 (WPA3) are three security protocols and security certification programs developed by the Wi-Fi Alliance to secure wireless computer networks. The Alliance defined these in response to serious weaknesses researchers had found in the previous system, WEP. WPA advantages are enhanced data security and tightened access control for wireless networks. Important characteristic is the compatibility between multiple wireless devices at the hardware level as well as at software level. WPA3 uses 128-bit encryption in WPA3-Personal mode (192-bit in WPA3-Enterprise). The WPA3 standard also replaces the Pre-Shared Key exchange with Simultaneous Authentication of Equals as defined in IEEE 802.11-2016 resulting in a more secure initial key exchange in personal mode. WPA Enterprise is a protocol-based authentication mode IEEE 802.1X using an external authentication server RADIUS and local client Supplicant.
Wi-Fi Protected Setup	provides an industry-wide mechanism to set up and configure networks for home and small office (SOHO) environments. Wi-Fi Protected Setup enables typical users who possess little understanding of traditional Wi-Fi configuration and security settings to easily configure new wireless networks, to add new devices and to enable security.

Wired Equivalent Privacy	is a security algorithm for IEEE 802.11 wireless networks. WEP, recognizable by the key of 10 or 26 hexadecimal digits, is widely in use and is often the first security choice presented to users by router configuration tools. In 2004, with the ratification of the full 802.11i standard (i.e. *WPA2*), the IEEE declared that both WEP-40 and WEP-104 have been deprecated.
Extended Authentication	or XAUTH, provides an additional level of authentication by allowing the *IPsec* gateway to request extended authentication from remote users, thus forcing remote users to respond with their credentials before being allowed access to the VPN.
Yandex.DNS	service of Yandex company to protect home network. Provides three filtering modes: • no filtering: resources are not blocked • safe mode: stops malicious and phishing websites • family mode: stops malicious and phishing websites, as well as resources for adults

A

Interface Hierarchy

Figure A.1. Core interfaces

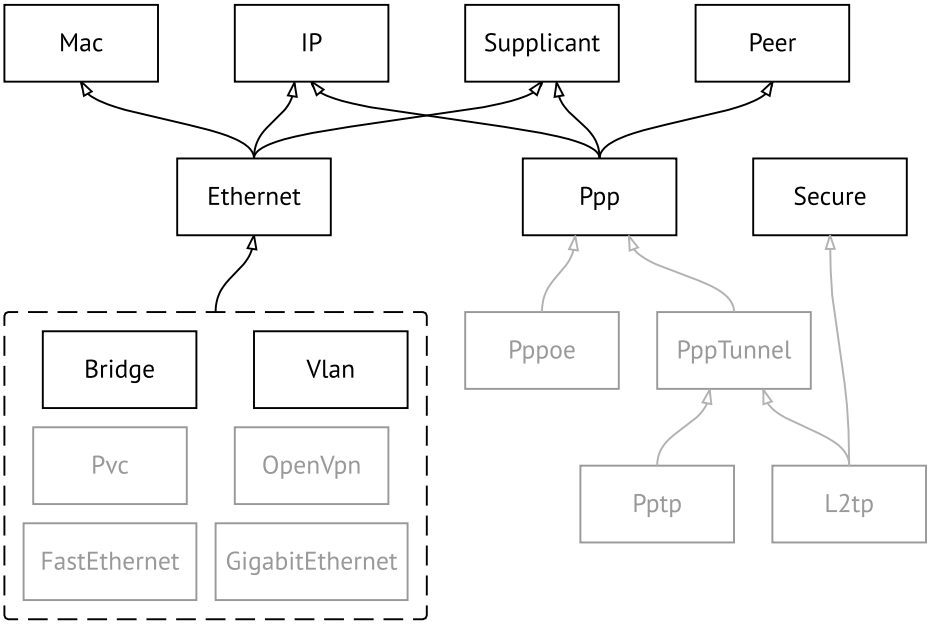


Figure A.2. Tunnel interfaces

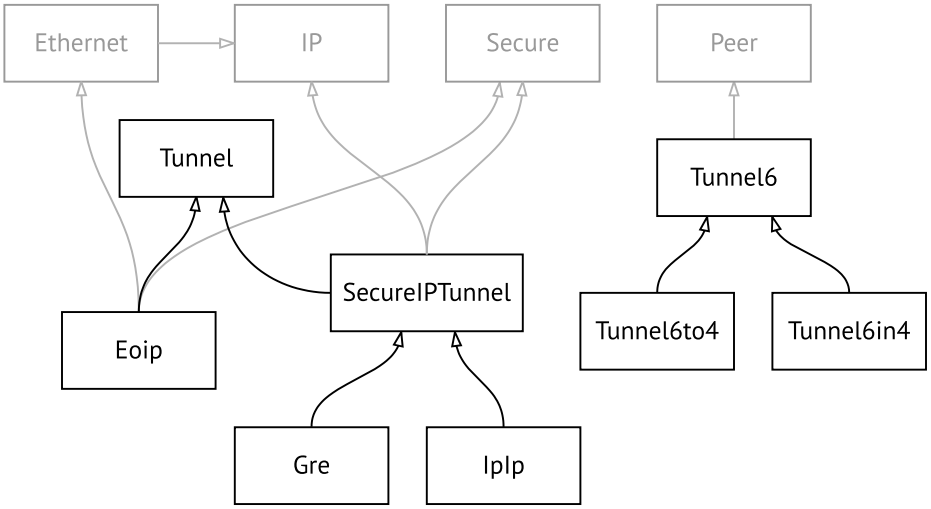


Figure A.3. USB interfaces

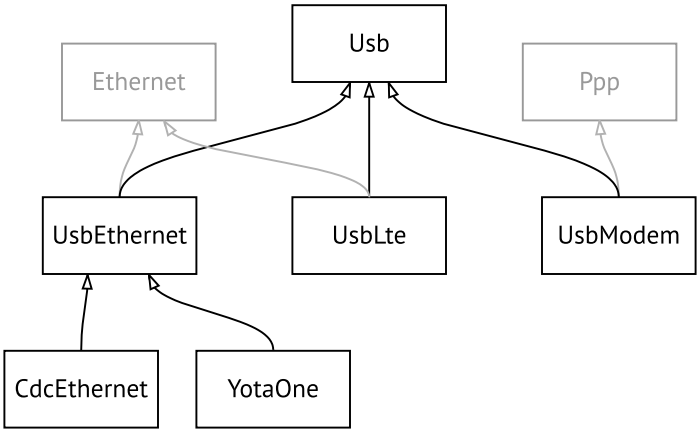
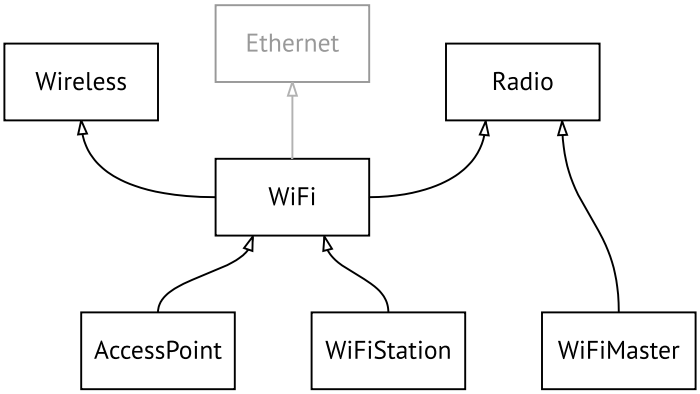


Figure A.4. Wi-Fi interfaces



B

HTTP API

B.1 REST Core Interface

City HTTP API lets you develop a custom application, that will access City settings using simple HTTP methods, such as GET and POST.

The base URL for all operations is `/rci`, that simply stands for REST Core Interface. It replaces the XML Core Interface, which is now deprecated but continues to be functional.

B.1.1 Resource Location

RCI is based on the City command tree. Device settings are mapped to RCI resources in such a way that every "`a b c`" command corresponds to the `/rci/a/b/c` URL.

As a result, hereby Command Reference gives you a complete picture of all RCI resources and their parameters. The words "command" and "resource" are used interchangeably in this manual.

Parameters are listed in the Arguments table of each command. They can be passed as part of the request using HTTP query: `/rci/a/b/c?`*`parameter`*`=`*`value`*. Unless otherwise specified for a certain command, query parameters are optional. Multiple parameters should be separated by ampersand (&) characters.

Parameters can also be passed in the POST request body, as described in Section B.1.3 on page 554.

B.1.2 Methods

Method semantics depend on the type of resource. There are three types of resources in RCI:

- Settings
- Actions
- Background processes

B.1.2.1 Settings

Settings are device configuration elements. You can view, modify, or delete settings using standard HTTP methods.

GET Retrieve settings.

POST Create or modify settings.

DELETE Delete settings (reset to default).

B.1.2.2 Actions

Actions are commands that do not modify settings. Actions run instantly as opposed to background processes, see also Section B.1.2.3 on page 554

GET Mapped to POST for `/rci/show`. Not applicable to other actions.

POST Execute a command and return its output.

DELETE Not applicable.

B.1.2.3 Background processes

Background processes are instances that can be created and polled for updates. Such processes are bound to a particular session, and cannot be accessed from anywhere else.

GET Retrieve updates from existing process. Returns 404 if there is no such process.

POST Create a background process.

DELETE Terminate a background process.

B.1.3 Data Format

HTTP POST requests must be submitted in a free-form JSON,[1] that is interpreted as a batch of parameters and nested settings, depending on the data type. Conversely, HTTP GET returns JSON data that was previously POSTed to the specified resource.

The primary data type is Object. This is unordered collection of key-value pairs, enclosed in curly brackets `{}`. Each key must be unique within an object.

Objects can be put one into another, or be combined in arrays as detailed in Section B.1.3.2 on page 555 and Section B.1.3.3 on page 555

B.1.3.1 Parameters

String, boolean and number values of an object are interpreted as parameters of the resource being addressed.

```
{
  "parameter": value
}
```

Example B.1. Set hotspot policy

Set policy “permit” for the Home network. Refer to Section 3.41.8 on page 270 to see how “interface” and “access” parameters are mentioned in the Arguments table.

[1] In compliance with RFC 7159.

```
POST /rci/ip/hotspot/policy HTTP/1.1
Host: 192.168.1.1
Content-length: 48
Content-type: application/json

{
  "interface":"Home",
  "access":"permit"
}
```

B.1.3.2 Nested resources

Object and array values of a parent object are interpreted as nested resources.

```
{
 "command": {
  "parameter": value
 }
}
```

In particular, empty object denotes a command with no parameters.

```
{
 "command": {}
}
```

Using this rule, you can address multiple resources at a time. RCI engine will process your request from top to bottom, recursing over the JSON structure. Parameters of a parent resource apply to all nested resources within the nearest surrounding scope.

Example B.2. Create and enable a PPP interface

Call “interface” to create a new PPPoE connection, as described in Section 3.27 on page 121, and enable it with “interface up”. The “name” parameter applies to both “interface” and “up”.

```
POST /rci HTTP/1.1
Host: 192.168.1.1
Content-length: 39
Content-type: application/json

{"interface":{"name":"PPPoE1","up":{}}}
```

B.1.3.3 Arrays

Arrays can be used to operate on a specific resource multiple times. The important thing is that arrays preserve the order of their elements, in contrast to object members.

```
{
 "command": [
  {"parameter1": value1},
  {"parameter2": value2} ]
}
```

B.1.3.4 Response structure

The structure of POST output strictly corresponds to input. RCI reproduces input arrays and nested objects, and replaces input parameters with output data. This approach lets you locate any part of the response using a resource name.

Example B.3. Show version and interface Home

Run two different “show” commands in a certain order.

```
POST /rci/show HTTP/1.1
Host: 192.168.1.1
Content-length: 46
Content-type: application/json

[{"version":{}},{"interface":{"name":"Home"}}]
```

Response is an array of two elements, in accordance with the request.

```
[
 {
  "version": {
   "release": "2.12.A.1.0-1",
   "arch": "mips",
   "ndm": {
    "exact": "0-cbf8590",
    "cdate": "15 Jan 2018"
   },
   "bsp": {
    "exact": "0-06ee10b",
    "cdate": "15 Jan 2018"
   },
   "ndw": {
    "version": "0.2.1",
    "features": "wifi_button,single_usb_port,dual_image",
    "components": "base,cloudcontrol,..."
   },
   "manufacturer": "Keenetic Ltd.",
   "vendor": "Keenetic",
   "series": "KN",
   "model": "4G (KN-1210)",
   "hw_version": "10128000",
   "hw_id": "KN-1210",
   "device": "4G",
   "class": "Internet Center"
  }
 },
 {
  "interface": {
   "id": "Bridge0",
   "index": 0,
   "type": "Bridge",
   "description": "Home network",
   "interface-name": "Home",
```

```
    "link": "up",
    "connected": "yes",
    "state": "up",
    "mtu": 1500,
    "tx-queue": 1000,
    "address": "192.168.1.1",
    "mask": "255.255.255.0",
    "uptime": 2621,
    "global": false,
    "security-level": "private",
    "mac": "50:ff:20:00:00:08",
    "auth-type": "none"
  }
 }
]
```

B.2 XML Core Interface

Warning: XML Core Interface is deprecated and is maintained for backward compatibility.

City provides an HTTP XML API. The API is implemented as `/ci` resource that accepts POST XML requests and returns XML after the user agent has been authenticated.

If City is reset to factory defaults, authentication is not required.

Example B.4. XML API call

Execute the "**show interface**" command for the WAN interface named `ISP`. This interface exists by default in City.

```
POST /ci HTTP/1.1
Host: 192.168.1.1
Connection: keep-alive
Content-Length: 177
Origin: http://192.168.1.1
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64)
Content-Type: application/xml
Referer: http://192.168.1.1/

<packet ref="/">
    <request id="1" ref="former.ifaces[load]">
        <command name="show interface">
            <name>ISP</name>
        </command>
    </request>
</packet>
```

The device responds with the current status of `ISP`:

```
HTTP/1.0 200 OK
Server: Ag [47]
Set-Cookie: _authorized=*; path=/
```

```
Content-type: text/xml
Content-Length: 760

<packet>
    <response id="1">
        <interface name="ISP">
            <mac>ec:43:f6:d3:22:d9</mac>
            <id>FastEthernet0/Vlan2</id>
            <index>2</index>
            <type>VLAN</type>
            <description>Broadband connection</description>
            <link>down</link>
            <connected>no</connected>
            <state>up</state>
            <mtu>1500</mtu>
            <tx-queue>1000</tx-queue>
            <global>yes</global>
            <defaultgw>no</defaultgw>
            <priority>700</priority>
            <security-level>public</security-level>
            <auth-type>none</auth-type>
        </interface>
        <message code="268370345" ident="Network::Interface::Base"
source="">done</message>
    </response>
</packet>
```

The <request> element is always sent from the user agent to the device. The device always responds with a <response>. The id attribute can be used to establish one-to-one correspondence between them.

Figure B.1. Request Element

```
<request id="identifier">
    <!-- request content -->
</request>
```

Figure B.2. Response Element

```
<response id="identifier">
    <!-- response content -->
</response>
```

There are two basic types of XML requests:

Command Request	Execute a specific command on the device. Available commands are described in Chapter 3 on page 33
Configuration Request	Get parameters that have been configured by a specific command.

B.2.1 Command Request

Command request can be used to execute a specific command on the device.

Figure B.3. Command Request

```
<request id="identifier">
    <command name="command">
        <no/>
        <argument>value</argument>
        ...
    </command>
</request>
```

command Space separated name of the command. Available commands are listed in Chapter 3 on page 33.

argument Name of the argument. Available arguments for each command are listed in Chapter 3 on page 33. Some commands do not require any arguments.

value Value of the argument.

no Optional element that is used to negate the action of the command. It has the same effect as the prefix no, see Section 2.3 on page 29.

B.2.2 Configuration Request

Configuration request can be used to get configured parameters. Web interface uses this kind of request to fill out the HTML forms.

Figure B.4. Configuration Request

```
<request id="identifier">
    <config name="command"/>
</request>
```

B.2.3 Request Packet

Multiple requests can be arranged in packets to optimize the performance.

Figure B.5. Request Packet

```
<packet>
    <request id="1">
        <!-- request content -->
    </request>
    <request id="2">
        <!-- request content -->
    </request>
    ...
</packet>
```

Response elements are returned as a packet. Response identifiers are used to bind response elements to requests. If there is no response, an empty <response/> element is returned.

Figure B.6. Response Packet

```
<packet>
    <response id="1">
        <!-- response content -->
    </response>
    <response id="2"/>
        <!-- no response for id=2 -->
    ...
</packet>
```

C

SNMP MIB

Management Information Bases (MIBs) are read-only.

The following MIBs are supported:

C.1 SNMPv2-MIB

OID: 1.3.6.1.2.1.1

The following data elements are supported:

- SNMPv2-MIB::sysDescr
- SNMPv2-MIB::sysUpTime
- SNMPv2-MIB::sysContact
- SNMPv2-MIB::sysName
- SNMPv2-MIB::sysLocation
- SNMPv2-MIB::sysServices

C.2 IF-MIB

OID: 1.3.6.1.2.1.2 and 1.3.6.1.2.1.31

The following data elements are supported:

Basical

OID: 1.3.6.1.2.1.2

- IF-MIB::ifNumber
- IF-MIB::ifIndex
- IF-MIB::ifDescr
- IF-MIB::ifType
- IF-MIB::ifMtu
- IF-MIB::ifSpeed
- IF-MIB::ifPhysAddress
- IF-MIB::ifAdminStatus

- IF-MIB::ifOperStatus
- IF-MIB::ifLastChange
- IF-MIB::ifInOctets
- IF-MIB::ifInUcastPkts
- IF-MIB::ifInDiscards
- IF-MIB::ifInErrors
- IF-MIB::ifOutOctets
- IF-MIB::ifOutUcastPkts
- IF-MIB::ifOutDiscards
- IF-MIB::ifOutErrors

Advanced

OID 1.3.6.1.2.1.31

- IF-MIB::ifName
- IF-MIB::ifInMulticastPkts
- IF-MIB::ifInBroadcastPkts
- IF-MIB::ifOutMulticastPkts
- IF-MIB::ifOutBroadcastPkts
- IF-MIB::ifHCInOctets
- IF-MIB::ifHCInUcastPkts
- IF-MIB::ifHCInMulticastPkts
- IF-MIB::ifHCInBroadcastPkts
- IF-MIB::ifHCOutOctets
- IF-MIB::ifHCOutUcastPkts
- IF-MIB::ifHCOutMulticastPkts
- IF-MIB::ifHCOutBroadcastPkts
- IF-MIB::ifLinkUpDownTrapEnable
- IF-MIB::ifHighSpeed
- IF-MIB::ifPromiscuousMode
- IF-MIB::ifConnectorPresent
- IF-MIB::ifAlias

- IF-MIB::ifCounterDiscontinuityTime

Main chipset	Switch	Device	Description
MT7621/RT63368	MT7530	Keenetic Giga III Keenetic Ultra II Keenetic LTE	64-bit per port octet counters. 32-bit per port packet counters. Separate per port broadcast, multicast and unicast packet counters.
	RTL8370M		
MT7620	RTL8367B	Keenetic Viva Keeentic Extra	
	Integrated	Keenetic 4G III Keenetic Lite II Keenetic Lite III Keenetic Omni Keenetic Omni II	32-bit per port octet counters & 16-bit per port packet counters. Last counter overflow event time set in IF-MIB::ifCounterDiscontinuityTime.
MT7628	Integrated	Keenetic Start II Keenetic Lite III rev.B Keenetic 4G III rev.B Keenetic Air Keenetic Extra II	16-bit per port packet counters only. Last counter overflow event time set in IF-MIB::ifCounterDiscontinuityTime.

C.3 IP-MIB

OID: 1.3.6.1.2.1.49

The following data elements are supported:

- TCP-MIB::tcpRtoAlgorithm
- TCP-MIB::tcpRtoMin
- TCP-MIB::tcpRtoMax
- TCP-MIB::tcpMaxConn
- TCP-MIB::tcpActiveOpens
- TCP-MIB::tcpPassiveOpens
- TCP-MIB::tcpAttemptFails

- TCP-MIB::tcpEstabResets
- TCP-MIB::tcpCurrEstab
- TCP-MIB::tcpInSegs
- TCP-MIB::tcpOutSegs
- TCP-MIB::tcpRetransSegs
- TCP-MIB::tcpInErrs
- TCP-MIB::tcpOutRsts

C.4 UDP-MIB

OID: 1.3.6.1.2.1.50

The following data elements are supported:

- UDP-MIB::udpInDatagrams
- UDP-MIB::udpNoPorts
- UDP-MIB::udpInErrors
- UDP-MIB::udpOutDatagrams
- UDP-MIB::udpHCInDatagrams
- UDP-MIB::udpHCOutDatagrams

C.5 HOST-RESOURCES-MIB

OID: 1.3.6.1.2.1.25

The following data elements are supported:

- HOST-RESOURCES-MIB::hrSystemUptime

C.6 UCD-SNMP-MIB

OID 1.3.6.1.4.1.2021

The following data elements are supported:

RAM info

- UCD-SNMP-MIB::memTotalReal
- UCD-SNMP-MIB::memAvailReal
- UCD-SNMP-MIB::memShared
- UCD-SNMP-MIB::memBuffer

- UCD-SNMP-MIB::memCached

USB-storage info

- UCD-SNMP-MIB::dskIndex
- UCD-SNMP-MIB::dskPath
- UCD-SNMP-MIB::dskTotal
- UCD-SNMP-MIB::dskAvail
- UCD-SNMP-MIB::dskUsed
- UCD-SNMP-MIB::dskPercent
- UCD-SNMP-MIB::dskPercentNode

System load info

- UCD-SNMP-MIB::laIndex
- UCD-SNMP-MIB::laNames
- UCD-SNMP-MIB::laLoad
- UCD-SNMP-MIB::laConfig
- UCD-SNMP-MIB::laLoadInt
- UCD-SNMP-MIB::ssCpuRawUser
- UCD-SNMP-MIB::ssCpuRawNice
- UCD-SNMP-MIB::ssCpuRawSystem
- UCD-SNMP-MIB::ssCpuRawIdle
- UCD-SNMP-MIB::ssRawInterrupts
- UCD-SNMP-MIB::ssRawContexts

D

IPsec Encryption Levels

The encryption level defines a set of *IKE* and *IPsec SA* algorithms.

Below a complete list of algorithms is displayed for each level in order of decreasing priority, as well as a set of commands **crypto ike proposal** to setup this profile manually.

In the list of algorithms is indicated:

- encryption with key length
- hash function for *HMAC* forming
- *PFS* mode (NO if disabled)

D.1 weak

Protocol	Encryption	Proposal
IKEv1	AES-128-CBC/SHA1/MODP1024	encryption aes-128-cbc
	AES-128-CBC/SHA1/MODP768	encryption 3des
	AES-128-CBC/MD5/MODP1024	encryption des
	AES-128-CBC/MD5/MODP768	integrity sha1
	3DES-CBC/SHA1/MODP1024	integrity md5
	3DES-CBC/SHA1/MODP768	dh-group 2
	3DES-CBC/MD5/MODP1024	dh-group 1
	3DES-CBC/MD5/MODP768	
	DES-CBC/SHA1/MODP1024	
	DES-CBC/SHA1/MODP768	
	DES-CBC/MD5/MODP1024	
	DES-CBC/MD5/MODP768	
IKEv2	AES-128-CBC/SHA1/MODP1024	encryption aes-128-cbc
	AES-128-CBC/SHA1/MODP768	encryption 3des
	AES-128-CBC/MD5/MODP1024	encryption des
	AES-128-CBC/MD5/MODP768	integrity sha1

Protocol	Encryption	Proposal
	3DES-CBC/SHA1/MODP1024	integrity md5
	3DES-CBC/SHA1/MODP768	dh-group 2
	3DES-CBC/MD5/MODP1024	dh-group 1
	3DES-CBC/MD5/MODP768	
	DES-CBC/SHA1/MODP1024	
	DES-CBC/SHA1/MODP768	
	DES-CBC/MD5/MODP1024	
	DES-CBC/MD5/MODP768	
IPsec SA	DES/MD5	cypher esp-des
	AES-128-CBC/SHA1	cypher esp-3des
	3DES-CBC/SHA1	cypher esp-aes-128
	DES/SHA1	hmac esp-md5-hmac
	AES-128-CBC/MD5	hmac esp-sha1-hmac
	3DES-CBC/MD5	

D.2 weak-pfs

Protocol	Encryption	Proposal
IKEv1	AES-128-CBC/SHA1/MODP1024	encryption aes-128-cbc
	AES-128-CBC/SHA1/MODP768	encryption 3des
	AES-128-CBC/MD5/MODP1024	encryption des
	AES-128-CBC/MD5/MODP768	integrity sha1
	3DES-CBC/SHA1/MODP1024	integrity md5
	3DES-CBC/SHA1/MODP768	dh-group 2
	3DES-CBC/MD5/MODP1024	dh-group 1
	3DES-CBC/MD5/MODP768	
	DES-CBC/SHA1/MODP1024	
	DES-CBC/SHA1/MODP768	
	DES-CBC/MD5/MODP1024	
	DES-CBC/MD5/MODP768	

Protocol	Encryption	Proposal
IKEv2	AES-128-CBC/SHA1/MODP1024	encryption aes-128-cbc
	AES-128-CBC/SHA1/MODP768	encryption 3des
	AES-128-CBC/MD5/MODP1024	encryption des
	AES-128-CBC/MD5/MODP768	integrity sha1
	3DES-CBC/SHA1/MODP1024	integrity md5
	3DES-CBC/SHA1/MODP768	dh-group 2
	3DES-CBC/MD5/MODP1024	dh-group 1
	3DES-CBC/MD5/MODP768	
	DES-CBC/SHA1/MODP1024	
	DES-CBC/SHA1/MODP768	
	DES-CBC/MD5/MODP1024	
	DES-CBC/MD5/MODP768	
IPsec SA	DES/MD5/MODP1024	cypher esp-des
	AES-128-CBC/SHA1	cypher esp-3des
	3DES-CBC/SHA1	cypher esp-aes-128
	DES/SHA1	hmac esp-md5-hmac
	AES-128-CBC/MD5	hmac esp-sha1-hmac
	3DES-CBC/MD5	dh-group 2
	AES-128-CBC/SHA1/MODP1024	dh-group 1
	3DES-CBC/SHA1/MODP1024	
	DES-CBC/SHA1/MODP1024	
	AES-128-CBC/SHA1/MODP768	
	3DES-CBC/SHA1/MODP768	
	DES-CBC/SHA1/MODP768	
	AES-128-CBC/MD5/MODP1024	
	3DES-CBC/MD5/MODP1024	
	AES-128-CBC/MD5/MODP768	
	3DES-CBC/MD5/MODP768	
	DES-CBC/MD5/MODP768	

D.3 normal

Protocol	Encryption	Proposal
IKEv1	AES-256-CBC/SHA1/MODP1536	encryption aes-256-cbc
	AES-256-CBC/SHA1/ECP384	encryption aes-128-cbc
	AES-256-CBC/SHA1/MODP2048	encryption 3des
	AES-256-CBC/SHA1/MODP1024	integrity sha1
	AES-128-CBC/SHA1/MODP1536	integrity sha256
	AES-128-CBC/SHA1/ECP256	dh-group 5
	AES-128-CBC/SHA1/MODP1024	dh-group 20
	3DES-CBC/SHA1/MODP2048	dh-group 14
	3DES-CBC/SHA1/MODP1536	dh-group 2
	3DES-CBC/SHA1/MODP1024	dh-group 26
	AES-256-CBC/SHA256/MODP1024	
	AES-128-CBC/SHA256/MODP1024	
	3DES-CBC/SHA256/MODP1024	
IKEv2	AES-256-CBC/SHA256/MODP1024	encryption aes-256-cbc
	AES-128-CBC/SHA256/MODP1024	encryption aes-128-cbc
	3DES-CBC/SHA256/MODP1024	encryption 3des
	AES-256-CBC/SHA1/MODP1024	integrity sha256
	AES-256-CBC/SHA1/ECP384	integrity sha1
	AES-256-CBC/SHA1/MODP2048	dh-group 2
	AES-128-CBC/SHA1/MODP1024	dh-group 20
	AES-128-CBC/SHA1/ECP256	dh-group 14
	AES-256-CBC/SHA256/MODP2048	dh-group 5
	3DES-CBC/SHA1/MODP2048	dh-group 26
	3DES-CBC/SHA1/MODP1536	
	3DES-CBC/SHA1/MODP1024	
IPsec SA	AES-128-CBC/SHA1	cypher esp-aes-128
	AES-256-CBC/SHA1	cypher esp-aes-256

Protocol	Encryption	Proposal
	3DES-CBC/SHA1	cypher esp-3des
	AES-128-CBC/SHA256	hmac esp-sha1-hmac
	AES-256-CBC/SHA256	hmac esp-sha256-hmac
	3DES-CBC/SHA256	

D.4 normal-pfs

Protocol	Encryption	Proposal
IKEv1	AES-256-CBC/SHA1/MODP1536	encryption aes-256-cbc
	AES-256-CBC/SHA1/ECP384	encryption aes-128-cbc
	AES-256-CBC/SHA1/MODP2048	encryption 3des
	AES-256-CBC/SHA1/MODP1024	integrity sha1
	AES-128-CBC/SHA1/MODP1536	integrity sha256
	AES-128-CBC/SHA1/ECP256	dh-group 5
	AES-128-CBC/SHA1/MODP1024	dh-group 20
	3DES-CBC/SHA1/MODP2048	dh-group 14
	3DES-CBC/SHA1/MODP1536	dh-group 2
	3DES-CBC/SHA1/MODP1024	dh-group 26
	AES-256-CBC/SHA256/MODP1024	
	AES-128-CBC/SHA256/MODP1024	
	3DES-CBC/SHA256/MODP1024	
IKEv2	AES-256-CBC/SHA256/MODP1024	encryption aes-256-cbc
	AES-128-CBC/SHA256/MODP1024	encryption aes-128-cbc
	3DES-CBC/SHA256/MODP1024	encryption 3des
	AES-256-CBC/SHA1/MODP1024	integrity sha256
	AES-256-CBC/SHA1/ECP384	integrity sha1
	AES-256-CBC/SHA1/MODP2048	dh-group 2
	AES-128-CBC/SHA1/MODP1024	dh-group 20
	AES-128-CBC/SHA1/ECP256	dh-group 14
	AES-256-CBC/SHA256/MODP2048	dh-group 5

Protocol	Encryption	Proposal
	3DES-CBC/SHA1/MODP2048	dh-group 26
	3DES-CBC/SHA1/MODP1536	
	3DES-CBC/SHA1/MODP1024	
IPsec SA	AES-128-CBC/SHA1/MODP1024	esp-aes-128
	AES-128-CBC/SHA1	cypher esp-aes-256
	AES-256-CBC/SHA1	cypher esp-3des
	3DES-CBC/SHA1	hmac esp-sha1-hmac
	AES-256-CBC/SHA1/MODP1536	hmac esp-sha256-hmac
	AES-128-CBC/SHA1/MODP1536	dh-group 2
	3DES-CBC/SHA1/MODP1536	dh-group 14
	AES-256-CBC/SHA1/MODP1024	
	3DES-CBC/SHA1/MODP1024	

D.5 normal-3des

Protocol	Encryption	Proposal
IKEv1	AES-256-CBC/SHA1/MODP1536	encryption aes-256-cbc
	AES-256-CBC/SHA1/ECP384	encryption aes-128-cbc
	AES-256-CBC/SHA1/MODP2048	encryption 3des
	AES-256-CBC/SHA1/MODP1024	integrity sha1
	AES-128-CBC/SHA1/MODP1536	integrity sha256
	AES-128-CBC/SHA1/ECP256	dh-group 5
	AES-128-CBC/SHA1/MODP1024	dh-group 20
	3DES-CBC/SHA1/MODP2048	dh-group 14
	3DES-CBC/SHA1/MODP1536	dh-group 2
	3DES-CBC/SHA1/MODP1024	dh-group 26
	AES-256-CBC/SHA256/MODP1024	
	AES-128-CBC/SHA256/MODP1024	
	3DES-CBC/SHA256/MODP1024	
IKEv2	AES-256-CBC/SHA256/MODP1024	encryption aes-256-cbc

Protocol	Encryption	Proposal
	AES-128-CBC/SHA256/MODP1024	encryption aes-128-cbc
	3DES-CBC/SHA256/MODP1024	encryption 3des
	AES-256-CBC/SHA1/MODP1024	integrity sha256
	AES-256-CBC/SHA1/ECP384	integrity sha1
	AES-256-CBC/SHA1/MODP2048	dh-group 2
	AES-128-CBC/SHA1/MODP1024	dh-group 20
	AES-128-CBC/SHA1/ECP256	dh-group 14
	AES-256-CBC/SHA256/MODP2048	dh-group 5
	3DES-CBC/SHA1/MODP2048	dh-group 26
	3DES-CBC/SHA1/MODP1536	
	3DES-CBC/SHA1/MODP1024	
IPsec SA	3DES-CBC/SHA1	cypher esp-3des
	AES-256-CBC/SHA1	cypher esp-aes-256
	AES-128-CBC/SHA1	cypher esp-aes-128
	3DES-CBC/SHA256	hmac esp-sha1-hmac
	AES-256-CBC/SHA256	hmac esp-sha256-hmac
	AES-128-CBC/SHA256	

D.6 normal-3des-pfs

Protocol	Encryption	Proposal
IKEv1	AES-256-CBC/SHA1/MODP1536	encryption aes-256-cbc
	AES-256-CBC/SHA1/ECP384	encryption aes-128-cbc
	AES-256-CBC/SHA1/MODP2048	encryption 3des
	AES-256-CBC/SHA1/MODP1024	integrity sha1
	AES-128-CBC/SHA1/MODP1536	integrity sha256
	AES-128-CBC/SHA1/ECP256	dh-group 5
	AES-128-CBC/SHA1/MODP1024	dh-group 20
	3DES-CBC/SHA1/MODP2048	dh-group 14
	3DES-CBC/SHA1/MODP1536	dh-group 2

Protocol	Encryption	Proposal
	3DES-CBC/SHA1/MODP1024	dh-group 26
	AES-256-CBC/SHA256/MODP1024	
	AES-128-CBC/SHA256/MODP1024	
	3DES-CBC/SHA256/MODP1024	
IKEv2	AES-256-CBC/SHA256/MODP1024	encryption aes-256-cbc
	AES-128-CBC/SHA256/MODP1024	encryption aes-128-cbc
	3DES-CBC/SHA256/MODP1024	encryption 3des
	AES-256-CBC/SHA1/MODP1024	integrity sha256
	AES-256-CBC/SHA1/ECP384	integrity sha1
	AES-256-CBC/SHA1/MODP2048	dh-group 2
	AES-128-CBC/SHA1/MODP1024	dh-group 20
	AES-128-CBC/SHA1/ECP256	dh-group 14
	AES-256-CBC/SHA256/MODP2048	dh-group 5
	3DES-CBC/SHA1/MODP2048	dh-group 26
	3DES-CBC/SHA1/MODP1536	
	3DES-CBC/SHA1/MODP1024	
IPsec SA	3DES-CBC/SHA1/MODP1024	cypher esp-3des
	3DES-CBC/SHA1	cypher esp-aes-256
	AES-256-CBC/SHA1	cypher esp-aes-128
	AES-128-CBC/SHA1	hmac esp-sha1-hmac
	AES-256-CBC/SHA1/MODP1536	hmac esp-sha256-hmac
	AES-128-CBC/SHA1/MODP1536	dh-group 2
	3DES-CBC/SHA1/MODP1536	dh-group 14
	AES-256-CBC/SHA1/MODP1024	
	AES-128-CBC/SHA1/MODP1024	

D.7 high

Protocol	Encryption	Proposal
IKEv1	AES-256-CBC/SHA256/MODP1024	encryption aes-256-cbc

Protocol	Encryption	Proposal
	AES-256-CBC/SHA256/ECP384	encryption aes-128-cbc
	AES-256-CBC/SHA256/MODP1536	integrity sha256
	AES-256-CBC/SHA1/MODP2048	integrity sha1
	AES-256-CBC/SHA1/ECP384	dh-group 2
	AES-256-CBC/SHA1/MODP1536	dh-group 20
	AES-128-CBC/SHA1/MODP2048	dh-group 5
	AES-128-CBC/SHA1/ECP256	dh-group 14
	AES-128-CBC/SHA1/MODP1536	dh-group 26
IKEv2	AES-256-CBC/SHA256/MODP1024	encryption aes-256-cbc
	AES-256-CBC/SHA256/ECP384	encryption aes-128-cbc
	AES-256-CBC/SHA256/MODP1536	integrity sha256
	AES-256-CBC/SHA1/MODP2048	integrity sha1
	AES-256-CBC/SHA1/ECP384	dh-group 2
	AES-256-CBC/SHA1/MODP1536	dh-group 20
	AES-128-CBC/SHA1/MODP2048	dh-group 5
	AES-128-CBC/SHA1/ECP256	dh-group 14
	AES-128-CBC/SHA1/MODP1536	dh-group 26
IPsec SA	AES-256-CBC/SHA256	cypher esp-aes-256
	AES-128-CBC/SHA256	cypher esp-aes-128
		hmac esp-hmac-sha256

D.8 strong

Protocol	Encryption	Proposal
IKEv1	AES-256-CBC/SHA1/MODP2048	encryption aes-256-cbc
	AES-256-CBC/SHA1/ECP384	encryption aes-128-cbc
	AES-256-CBC/SHA1/MODP1536	integrity sha1
	AES-128-CBC/SHA1/MODP2048	dh-group 14
	AES-128-CBC/SHA1/ECP256	dh-group 20
	AES-128-CBC/SHA1/MODP1536	dh-group 5

Protocol	Encryption	Proposal
		dh-group 26
IKEv2	AES-256-CBC/SHA1/MODP2048 AES-256-CBC/SHA1/ECP384 AES-256-CBC/SHA1/MODP1536 AES-128-CBC/SHA1/MODP2048 AES-128-CBC/SHA1/ECP256 AES-128-CBC/SHA1/MODP1536	encryption aes-256-cbc encryption aes-128-cbc integrity sha1 dh-group 14 dh-group 20 dh-group 5 dh-group 26
IPsec SA	AES-256-CBC/SHA1/MODP1536 AES-256-CBC/SHA1/MODP2048 AES-128-CBC/SHA1/MODP2048 AES-128-CBC/SHA1/MODP1536	cypher esp-aes-256 cypher esp-aes-128 hmac esp-sha1-hmac dh-group 5 dh-group 14

D.9 strong-aead

Protocol	Encryption	Proposal
IKEv1	AES-256-GCM-16/PRF-SHA384/ECP384	aead encryption aes-256-gcm-16 prf sha384 dh-group 20
IKEv2	AES-256-GCM-16/PRF-SHA384/ECP384	aead encryption aes-256-gcm-16 prf sha384 dh-group 20
IPsec SA	AES-256-GCM-16 CHACHA20POLY1305	aead cypher aes-256-gcm-16

D.10 strong-aead-pfs

Protocol	Encryption	Proposal
IKEv1	AES-256-GCM-16/PRF-SHA384/ECP384	aead

Protocol	Encryption	Proposal
		encryption aes-256-gcm-16 prf sha384 dh-group 20
IKEv2	AES-256-GCM-16/PRF-SHA384/ECP384	aead encryption aes-256-gcm-16 prf sha384 dh-group 20
IPsec SA	AES-256-GCM-16/ECP384 CHACHA20POLY1305-ECP384	aead cypher aes-256-gcm-16 dh-group 20