Console

Overview

The console is used for accessing the MikroTik Router's configuration and management features using text terminals, either remotely using a serial port, telnet, SSH, or console screen within Winbox, or directly using a monitor and keyboard. The console is also used for writing scripts. This manual describes the general console operation principles. Please consult the Scripting Manual on some advanced console commands and on how to write scripts.

Hierarchy

The console allows the configuration of the router's settings using text commands. Since there are a lot of available commands, they are split into groups organized in a way of hierarchical menu levels. The name of a menu level reflects the configuration information accessible in the relevant section, eg. **/ip hotspot**.

Example

For example, you can issue the **/ip route print** command:

```
[admin@MikroTik] > ip route print
Flags: X - disabled, A - active, D - dynamic,
C - connect, S - static, r - rip, b - bgp, o - ospf, m - mme,
B - blackhole, U - unreachable, P - prohibit
 # DST-ADDRESS PREF-SRC G GATEWAY DIS INTE...
0 A S 0.0.0.0/0 r 10.0.3.1 1 bridge1
1 ADC 1.0.1.0/24 1.0.1.1 0 bridge1
2 ADC 1.0.2.0/24 1.0.2.1 0 ether3
3 ADC 10.0.3.0/24 10.0.3.144 0 bridge1
4 ADC 10.10.10.0/24 10.10.10.1 0 wlan1
[admin@MikroTik] >
```

Instead of typing /**ip route** path before each command, the path can be typed only once to move into this particular branch of the menu hierarchy. Thus, the example above could also be executed like this:

```
[admin@MikroTik] > ip route
[admin@MikroTik] ip route> print
Flags: X - disabled, A - active, D - dynamic,
C - connect, S - static, r - rip, b - bgp, o - ospf, m - mme,
 B - blackhole, U - unreachable, P - prohibit #
DST-ADDRESS PREF-SRC G GATEWAY DIS INTE...
0 A S 0.0.0.0/0 r 10.0.3.1 1 bridge1
1 ADC 1.0.1.0/24 1.0.1.1 0 bridge1
2 ADC 1.0.2.0/24 1.0.2.1 0 ether3
3 ADC 10.0.3.0/24 10.0.3.144 0 bridge1
4 ADC 10.10.10.0/24 10.10.10.1 0 wlan1 [
admin@MikroTik] ip route>
```

Notice that the prompt changes to reflect where you are located in the menu hierarchy at the moment. To move to the top level again, type " **/** "

```
[admin@MikroTik] > ip route
[admin@MikroTik] ip route> /
[admin@MikroTik] >
```

To move up one command level, type " **..** "

```
[admin@MikroTik] ip route> ..
[admin@MikroTik] ip>
```

You can also use **/** and **..** to execute commands from other menu levels without changing the current level:

```
[admin@MikroTik] ip route> /ping 10.0.0.1
10.0.0.1 ping timeout
2 packets transmitted, 0 packets received, 100% packet loss
[admin@MikroTik] ip firewall nat> .. service-port print
Flags: X - disabled, I - invalid
# NAME PORTS
0 ftp 21
1 tftp 69
2 irc 6667
3 h323
4 sip
5 pptp
[admin@MikroTik] ip firewall nat>
```

Item Names and Numbers

Many of the command levels operate with arrays of items: interfaces, routes, users, etc. Such arrays are displayed in similarly looking lists. All items in the list have an item number followed by flags and parameter values.

To change the properties of an item, you have to use the **set** command and specify the name or number of the item.

Item Names

Some lists have items with specific names assigned to each of them. Examples are **interface** or **user** levels. There you can use item names instead of item numbers.

You do not have to use the **print** command before accessing items by their names, which, as opposed to numbers, are not assigned by the console internally, but are properties of the items. Thus, they would not change on their own. However, there are all kinds of obscure situations possible when several users are changing the router's configuration at the same time. Generally, item names are more "stable" than the numbers, and also more informative, so you should prefer them to numbers when writing console scripts.

Item Numbers

Item numbers are assigned by the print command and are not constant - two successive print commands may order items differently. But the results of the last print commands are memorized and, thus, once assigned, item numbers can be used even after **add**, **remove,** and **move** operations. Item numbers are assigned on a per-session basis, they will remain the same until you quit the console or until the next print command is executed. Also, numbers are assigned separately for every item list, so for example, the **ip address print** will not change the numbering of the interface list.

It is possible to use item numbers without running the **print** command. Numbers will be assigned just as if the **print** command was executed.

You can specify multiple items as targets for some commands. Almost everywhere, where you can write the number of item, you can also write a list of numbers.

```
[admin@MikroTik] > interface print
Flags: X - disabled, D - dynamic, R - running
# NAME TYPE MTU
0 R ether1 ether 1500
1 R ether2 ether 1500
2 R ether3 ether 1500
3 R ether4 ether 1500
[admin@MikroTik] > interface set 0,1,2 mtu=1460
[admin@MikroTik] > interface print
 Flags: X - disabled, D - dynamic, R - running
# NAME TYPE MTU
0 R ether1 ether 1460
1 R ether2 ether 1460
2 R ether3 ether 1460
3 R ether4 ether 1500
[admin@MikroTik] >
```

Warning: Do not use Item numbers in scripts, it is not a reliable way to edit items in the **scheduler. scripts**, etc. Instead, use the "find" command. More info here also look at scripting examples.

Quick Typing

Two features in the console help entering commands much quicker and easier - the **[Tab]** key completions, and abbreviations of command names. Completions work similarly to the bash shell in UNIX. If you press the **[Tab]** key after a part of a word, the console tries to find the command within the current context that begins with this word. If there is only one match, it is automatically appended, followed by a space:

*/inte***[Tab]_** becomes **/interface _**

If there is more than one match, but they all have a common beginning, which is longer than that of what you have typed, then the word is completed to this common part, and no space is appended:

*/interface set e***[Tab]_** becomes **/interface set ether_**

If you've typed just the common part, pressing the tab key once has no effect. However, pressing it for the second time shows all possible completions in compact form:

```
[admin@MikroTik] > interface set e[Tab]_
[admin@MikroTik] > interface set ether[Tab]_
[admin@MikroTik] > interface set ether[Tab]_
ether1 ether5
[admin@MikroTik] > interface set ether_
```

The **[Tab]** key can be used almost in any context where the console might have a clue about possible values - command names, argument names, arguments that have only several possible values (like names of items in some lists or name of protocol in firewall and NAT rules). You cannot complete numbers, IP addresses, and similar values.

Another way to press fewer keys while typing is to abbreviate command and argument names. You can type only the beginning of the command name, and, if it is not ambiguous, the console will accept it as a full name. So typing:

```
[admin@MikroTik] > pin 10.1 c 3 si 100
```

equals to:

```
[admin@MikroTik] > ping 10.0.0.1 count 3 size 100
```

It is possible to complete not only the beginning, but also any distinctive substring of a name: if there is no exact match, the console starts looking for words that have string being completed as first letters of a multiple-word name, or that simply contain letters of this string in the same order. If a single such word is found, it is completed at the cursor position. For example:

```
[admin@MikroTik] > interface x[TAB][TAB]_
dot1x  vxlan  export
[admin@MikroTik] > interface mt[TAB]_
[admin@MikroTik] > interface monitor-traffic _
```

General Commands

Some commands are common to nearly all menu levels, namely: **print**, **set**, **remove**, **add**, **find**, **get**, **export**, **enable**, **disable**, **comment**, and **move**. These commands have similar behavior throughout different menu levels.

- **add** - this command usually has all the same arguments as **set**, except the item number argument. It adds a new item with the values you have specified, usually at the end of the item list, in places where the order of items is relevant. There are some required properties that you have to supply, such as the interface for a new address, while other properties are set to defaults unless you explicitly specify them.
    - Common Parameters
        - *copy-from* - Copies an existing item. It takes the default values of a new item's properties from another item. If you do not want to make an exact copy, you can specify new values for some properties. When copying items that have names, you will usually have to give a new name to a copy
        - *place-before* - places a new item before an existing item with a specified position. Thus, you do not need to use the move command after adding an item to the list.
        - *disabled* - controls disabled/enabled state of the newly added item(-s)
        - *comment* - holds the description of a newly created item
    - Return Values
        - add command returns the internal number of item it has added
- **edit** - this command is associated with the **set** command. It can be used to edit values of properties that contain a large amount of text, such as scripts, but it works with all editable properties. Depending on the capabilities of the terminal, either a fullscreen editor or a single-line editor is launched to edit the value of the specified property. The edit field for console scripts is limited to 30 thousand characters.
- **find** - The find command has the same arguments as set, plus the flag arguments like *disabled* or *active* that take values *yes* or *no* depending on the value of the respective flag. To see all flags and their names, look at the top of the **print** command's output. The **find** command returns internal numbers of all items that have the same values of arguments as specified.
- **move** - changes the order of items in the list.
    - Parameters
        - first argument specifies the item(-s) being moved.
        - the second argument specifies the item before which to place all items being moved (they are placed at the end of the list if the second argument is omitted).
- **print** - shows all information that's accessible from a particular command level. Thus, **/system clock print** shows the system date and time, **/ip route print** shows all routes, etc. If there's a list of items in the current level and they are not read-only, i.e. you can change/remove them (an example of a read-only item list is /system history, which shows a history of executed actions), then print command also assigns numbers that are used by all commands that operate with items in this list.
    - Common Parameters
        - *from* - show only specified items, in the same order in which they are given.
        - *where* - show only items that match specified criteria. The syntax of *where* the property is similar to the **find** command.
        - *brief* - forces the print command to use tabular output form
        - *detail* - forces the print command to use property=value output form
        - *count-only* - shows the number of items
        - *file* - prints the contents of the specific submenu into a file on the router.
        - *interval* - updates the output from the *print* command for every interval of seconds.
        - *oid* - prints the OID value for properties that are accessible from SNMP
        - *without-paging* - prints the output without stopping after each screenful.
- **remove** - removes specified item(-s) from a list.
- **set** - allows you to change values of general parameters or item parameters. The set command has arguments with names corresponding to values you can change. Use "F1" or double [Tab] to see a list of all arguments. If there is a list of items in this command level, then "set" has one action argument that accepts the number of item (or list of numbers) you wish to set up. This command does not return anything.
- **reset** - reset parameters to default values

ⓘ You can combine commands, here are two variants of the same command that will place a new firewall filter entry, by looking up the comment:

/ip firewall/filter/add chain=forward place-before=[find where comment=CommentX]
/ip/firewall/filter/add chain=forward place-before="CommentX"

Modes

The console line editor works either in multiline mode or in single-line mode. In multiline mode line editor displays the complete input line, even if it is longer than a single terminal line. It also uses a full-screen editor for editing large text values, such as scripts. In single-line mode, only one terminal line is used for line editing, and long lines are shown truncated around the cursor. A full-screen editor is not used in this mode.

The choice of modes depends on detected terminal capabilities.

List of keys

```
F1                Give the list of available commands
command F1        Give help on the command and list of arguments

[Tab]             Complete the command/word. If the input is ambiguous,
                  a second [Tab] gives possible options

F3 or Ctrl-R      Search command history
F4 or Ctrl-X      Toggle safe mode
F5 or Ctrl-L      Repaint the screen
F7                Toggle hotlock mode
Ctrl-\            Split line
Home or Ctrl-A    Go to the beginning of the line
End or Ctrl-E     Go to the end of the line
Ctrl-C            Interrupt current action
Ctrl-D            Terminate session (on empty prompt)
Ctrl-K            Delete to the end of the line
Ctrl-U            Delete to the beginning of the line

/                 Move up to base level
..                Move up one level
/command          Use command at the base level
```

up, **down** and **split** keys leave the cursor at the end of the line.

Built-in Help

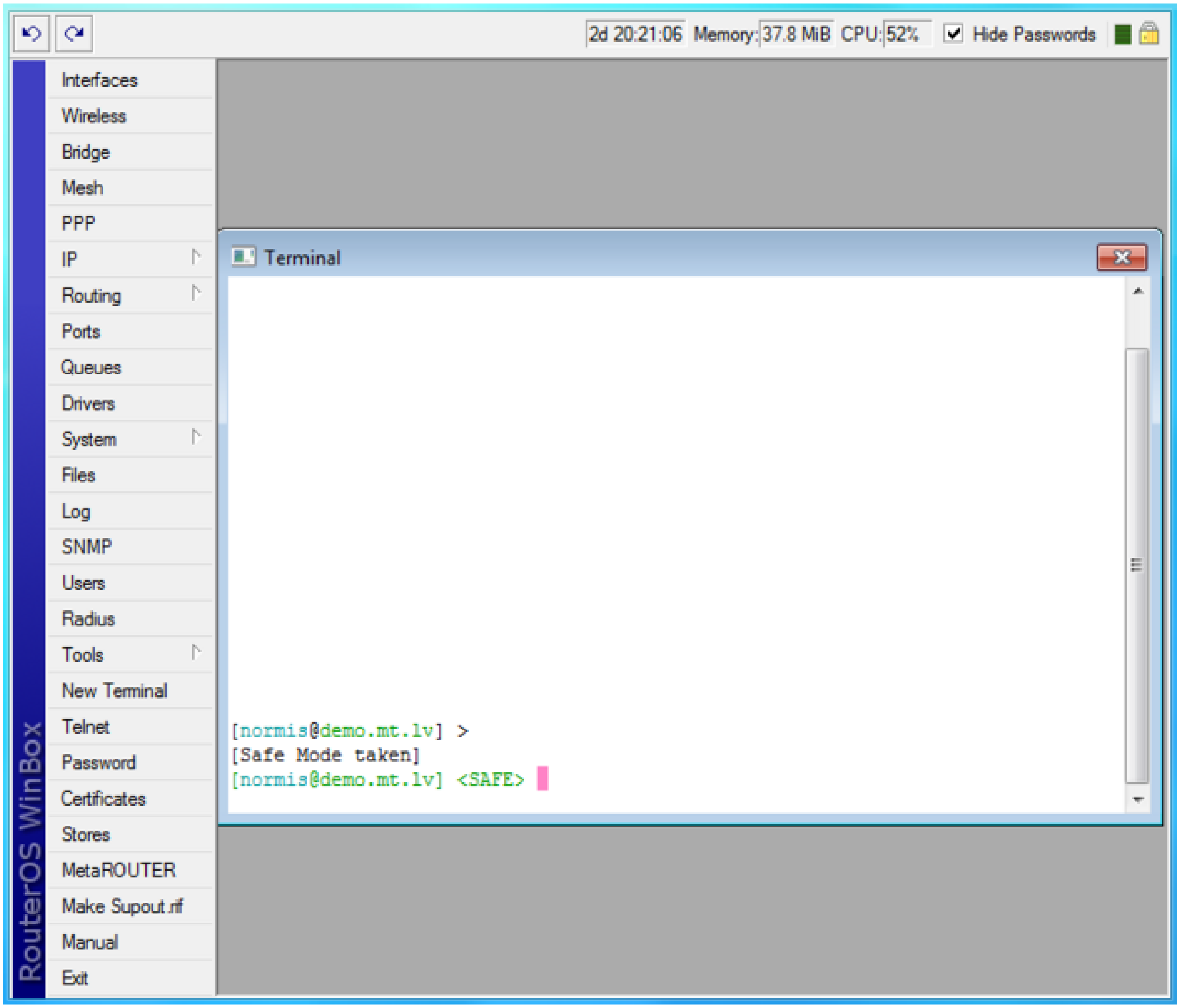
The console has built-in help. Press **F1** for general console usage Help. The general rule is that Help shows what you can type in a position where the **F1** was pressed (similarly to pressing **[Tab]** key twice, but in verbose form and with explanations).

Safe Mode

It is sometimes possible to change router configuration in a way that will make the router inaccessible (except from the local console). Usually, this is done by accident, but there is no way to undo the last change when the connection to the router is already cut. Safe mode can be used to minimize such risk.

The **"Safe Mode"** button in the Winbox GUI allows you to enter Safe Mode, while in the CLI, you can access it by either using the keyboard shortcut **F4** or pressing **[CTRL]+[X]**. To exit without saving the made changes in CLI, hit **[CTRL]+[D].**

```
[admin@MikroTik] ip route>[CTRL]+[X]
[Safe Mode taken]
[admin@MikroTik] ip route<SAFE>
```



```
[normis@demo.mt.lv] >
[Safe Mode taken]
[normis@demo.mt.lv] <SAFE>
```

Message **Safe Mode taken** is displayed and prompt changes to reflect that session is now in safe mode. All configuration changes that are made (also from other login sessions), while the router is in safe mode, are automatically undone if the safe mode session terminates abnormally. You can see all such changes that will be automatically undone and tagged with an **F** flag in the system history:

```
[admin@MikroTik] /ip/route>
[Safe Mode taken]
[admin@MikroTik] /ip/route<SAFE> add
[admin@MikroTik] /ip/route<SAFE> /system/history/print
Flags: U, F - FLOATING-UNDO
Columns: ACTION, BY, POLICY
  ACTION                 BY     POLICY
F route 0.0.0.0/0 added  admin  write
```

Now, if the telnet connection (or WinBox terminal) is cut, then after a while (TCP timeout is **9** minutes) all changes that were made while in safe mode will be undone. Exiting session by **[Ctrl]+[D]** also undoes all safe mode changes, while **/quit** does not.

If another user tries to enter safe mode, he's given the following message:

```
[admin@MikroTik] >
Hijacking Safe Mode from someone - unroll/release/don't take it [u/r/d]:
```

- [u] - undoes all safe mode changes, and puts the current session in safe mode.

- [r] - keeps all current safe mode changes, and puts the current session in a safe mode. The previous owner of safe mode is notified about this:

```
[admin@MikroTik] ip firewall rule input
[Safe mode released by another user]
```

- [d] - leaves everything as-is.

If too many changes are made while in safe mode, and there's no room in history to hold them all (currently history keeps up to 100 most recent actions), then the session is automatically put out of the safe mode, and no changes are automatically undone. Thus, it is best to change the configuration in small steps, while in safe mode. Pressing **[Ctrl]+[X]** twice is an easy way to empty the safe mode action list.

> ⚠ As "Safe Mode" operates within the user's session and stores configuration changes, it will be ignored for commands requiring a reboot, such as resetting configuration or restoring from a backup.

HotLock Mode

When HotLock mode is enabled commands will be auto-completed.

To enter/exit HotLock mode press **F7**.

```
[admin@MikroTik] /ip/address> [F7]
[admin@MikroTik] /ip/address>>
```

Double>>is an indication that HotLock mode is enabled. For example, if you type "`/in et`", it will be auto-completed to

```
[admin@MikroTik] /ip/address>> /interface/ethernet/
```