CAPsMAN

CAPsMAN AAA

Settings to configure CAPsMAN AAA functionality are found in the **/caps-man aaa** menu:

Property	Description
mac-format (*string*; Default: **XX:XX:XX:XX:XX:XX**)	Controls how the MAC address of the client is encoded by Access Point in the User-Name attribute of the MAC authentication and MAC accounting RADIUS requests.
mac-mode (*as-username | as-username-and-password*; Default: **as username**)	By default Access Point uses an empty password, when sending Access-Request during MAC authentication. When this property is set to as-username-and-password, Access Point will use the same value for the User-Password attribute as for the User-Name attribute.
mac-caching (*disabled | time-interval*; Default: **disabled**)	If this value is set to a time interval, the Access Point will cache RADIUS MAC authentication responses for a specified time, and will not contact the RADIUS server if matching cache entry already exists. The value disabled will disable the cache, Access Point will always contact the RADIUS server.
interim-update (*disabled | time-interval*; Default: **disabled**)	When RADIUS accounting is used, Access Point periodically sends accounting information updates to the RADIUS server. This property specifies the default update interval that can be overridden by the RADIUS server using the Acct-Interim-Interval attribute.
called-format (*mac | mac:ssid | ssid*; Default: **mac:ssid**)	Format of how the "called-id" identifier will be passed to RADIUS. When configuring radius server clients, you can specify "called-id" in order to separate multiple entires.

Example

Assuming that rest of the settings are already configured and only the "Security" part has been left.

Radius authentication with one server

1. Create CAPsMAN security configuration

2. Configure Radius server client

3. Assign the configuration to your master profile (or directly to CAP itself)

```
/caps-man security add authentication-types=wpa2-eap eap-methods=passthrough encryption=aes-ccm group-
encryption=aes-ccm name=radius
/radius add address=x.x.x.x secret=SecretUserPass service=wireless
/caps-man configuration set security=radius
```

Radius authentication with different radius servers for each SSID

1. Create CAPsMAN security configuration

2. Configure AAA settings

3. Configure Radius server clients

4. Assign the configuration to your master profile (or directly to CAP itself)

```
/caps-man security add authentication-types=wpa2-eap eap-methods=passthrough encryption=aes-ccm group-
encryption=aes-ccm name=radius
/caps-man aaa set called-format=ssid
/radius add address=x.x.x.x secret=SecretUserPass service=wireless called-id=SSID1
/radius add address=y.y.y.y secret=SecretUserPass service=wireless called-id=SSID2
/caps-man configuration set security=radius
```

Now everyone connecting to CAP's with ssid=**SSID1** will have their radius authentication requests sent to **x.x.x.x** and everyone connecting to CAP's with ssid=**SSID2** will have their radius authentication requests sent to **y.y.y.y**

CAPsMAN Access-list

Access list on CAPsMAN is an ordered list of rules that is used to allow/deny clients to connect to any CAP under CAPsMAN control. When a client attempts to connect to a CAP that is controlled by CAPsMAN, CAP forwards that request to CAPsMAN. As a part of the registration process, CAPsMAN consults an access list to determine if a client should be allowed to connect. The default behavior of the access list is to allow a connection.

Access list rules are processed one by one until a matching rule is found. Then the action in the matching rule is executed. If action specifies that the client should be accepted, the client is accepted, potentially overriding its default connection parameters with ones specified in access-list rule.

An access list is configured in the **/caps-man access-list** menu. There are the following parameters for access-list rules:

- client matching parameters:
  - address - MAC address of the client
  - mask - MAC address mask to apply when comparing client address
  - interface - optional interface to compare with an interface to which client actually connects to
  - time - a time of day and days when rule matches
  - signal-range - range in which client signal must fit for a rule to match
  - allow-signal-out-of-range - an option that permits the client's signal to be out of the range always or for some time interval
- action parameter - specifies an action to take when client matches:
  - accept - accept client
  - reject - reject client
  - query-radius - query RADIUS server if a particular client is allowed to connect
- connection parameters:
  - ap-tx-limit - tx speed limit in direction to client
  - client-tx-limit - tx speed limit in direction to AP (applies to RouterOS clients only)
  - client-to-client-forwarding - specifies whether to allow forwarding data received from this client to other clients connected to the same interface
  - private-passphrase - PSK passphrase to use for this client if some PSK authentication algorithm is used
  - radius-accounting - specifies if RADIUS traffic accounting should be used if RADIUS authentication gets done for this client
  - vlan-mode - VLAN tagging mode specifies if traffic coming from a client should get tagged (and untagged when going to a client).
  - vlan-id - VLAN ID to use if doing VLAN tagging.

CAPsMAN channel

Channel group settings allow for the configuration of lists of radio channel related settings, such as radio band, frequency, Tx Power extension channel, and width.

Channel group settings are configured in the Channels profile menu **/caps-man channels**

Property	Description
band (*2ghz-b* | *2ghz-b/g* | *2ghz-b/g/n* | *2ghz-onlyg* | *2ghz-onlyn* | *5ghz-a* | *5ghz-a/n* | *5ghz-onlyn*; Default:)	Define operational radio frequency band and mode taken from hardware capability of wireless card
comment (*string*; Default:)	Short description of the Channel Group profile
extension-channel (*Ce* | *Ceee* | *eC* | *eCee* | *eeCe* | *eeeC* | *disabled*; Default:)	Extension channel configuration. (E.g. Ce = extension channel is above Control channel, eC = extension channel is below Control channel)
frequency (*integer [0..4294967295]*; Default:)	Channel frequency value in MHz on which AP will operate.
name (*string*; Default:)	A descriptive name for the Channel Group Profile
tx-power (*integer [-30..40]*; Default:)	TX Power for CAP interface (for the whole interface not for individual chains) in dBm. It is not possible to set higher than allowed by country regulations or interface. By default max allowed by country or interface is used.
width (; Default:)	Sets Channel Width in MHz. (E.g. 20, 40)

save-selected (; Default: **yes**)	Saves selected channel for the CAP Radio - will select this channel after the CAP reconnects to CAPsMAN and use it till the channel Re-optimize is done for this CAP.

CAPsMAN configuration

Configuration profiles permit pre-defined 'top-level' master settings to be applied to CAP radios being provisioned.

Configuration Profiles are configured in **/caps-man configuration** menu:

Property	Description
channel (*list*; Default:)	User defined list taken from Channel names (**/caps-man channels**)
channel.band (*2ghz-b | 2ghz-b/g | 2ghz-b/g/n | 2ghz-onlyg | 2ghz-onlyn | 5ghz-a | 5ghz-a/n | 5ghz-onlyn | 5ghz-a/n/ac | 5ghz-only-ac*; Default:)	Defines set of used channels.
channel.control-channel-width (*40mhz-turbo | 20mhz | 10mhz | 5mhz*; Default:)	Defines set of used channel widths.
channel.extension-channel (*Ce | Ceee | eC | eCee | eeCe | eeeC | xx | xxxx | disabled*; Default:)	Extension channel configuration. (E.g. Ce = extension channel is above Control channel, eC = extension channel is below Control channel)
channel.frequency (*integer [0..4294967295]*; Default:)	Channel frequency value in MHz on which AP will operate. If left blank, CAPsMAN will automatically determine the best frequency that is least occupied.
channel.reselect-interval (*time [00:00:00]; [00:00:00..00:00:00]*; Default:)	The interval after which the least occupied frequency is chosen, can be defined as a random interval, ex. as "30m..60m". Works only if **channel.frequency** is left blank.
channel.save-selected (*yes | no*; Default: **no**)	If channel frequency is chosen automatically and **channel.reselect-interval** is used, then saves the last picked frequency.
channel.secondary-frequency (*integer [0..4294967295]*; Default: **auto**)	Specifies the second frequency that will be used for 80+80MHz configuration. Set it to **Disabled** in order to disable 80+80MHz capability.
channel.skip-dfs-channels (*yes | no*; Default: **no**)	If **channel.frequency** is left blank, the selection will skip DFS channels
channel.tx-power (*integer [-30..40]*; Default:)	TX Power for CAP interface (for the whole interface not for individual chains) in dBm. It is not possible to set higher than allowed by country regulations or interface. By default max allowed by country or interface is used.
channel.width (; Default:)	Sets Channel Width in MHz.
comment (*string*; Default:)	Short description of the Configuration profile
country (*name of the country | no_country_set*; Default: **no_country_set**)	Limits available bands, frequencies and maximum transmit power for each frequency. Also specifies default value of **scan-list**. Value *no_country_set* is an FCC compliant set of channels.
datapath (*list*; Default:)	User defined list taken from Datapath names (**/caps-man datapath**)
datapath.bridge (*list*; Default:)	Bridge to which particular interface should be automatically added as port. Required only when local-forwarding is not used.
datapath.bridge-cost (*integer [1..200000000]*; Default:)	bridge port cost to use when adding as bridge port
datapath.bridge-horizon (*integer [0..4294967295]*; Default:)	bridge horizon to use when adding as bridge port

datapath.client-to-client-forwarding (*yes* / *no*; Default: **no**)	controls if client-to-client forwarding between wireless clients connected to interface should be allowed, in local forwarding mode this function is performed by CAP, otherwise it is performed by CAPsMAN
datapath.interface-list (; Default:)	
datapath.l2mtu (; Default:)	set Layer2 MTU size
datapath.local-forwarding (*yes* / *no*; Default: **no**)	Controls forwarding mode. If disabled, all L2 and L3 data will be forwarded to CAPsMAN, and further forwarding decisions will be made only then. **Note**, if disabled, make sure that each CAP interface MAC Address that participates in the same broadcast domain is unique (including local MAC's, like Bridge-MAC).
datapath.mtu (; Default:)	set MTU size
datapath.openflow-switch (; Default:)	OpenFlow switch port (when enabled) to add interface to
datapath.vlan-id (*integer [1..4095]*; Default:)	VLAN ID to assign to interface if vlan-mode enables use of VLAN tagging
datapath.vlan-mode (*use-service-tag* / *use-tag*; Default:)	Enables and specifies the type of VLAN tag to be assigned to the interface (causes all received data to get tagged with VLAN tag and allows the interface to only send out data tagged with given tag)
disconnect-timeout (; Default:)	
distance (; Default:)	
frame-lifetime (; Default:)	
guard-interval (*any* / *long*; Default: **any**)	Whether to allow the use of short guard interval (refer to 802.11n MCS specification to see how this may affect throughput). "any" will use either short or long, depending on data rate, "long" will use long only.
hide-ssid (*yes* / *no*; Default:)	• *yes* - AP does not include SSID in the beacon frames and does not reply to probe requests that have broadcast SSID. • *no* - AP includes SSID in the beacon frames and replies to probe requests that have broadcast SSID. This property has effect only in AP mode. Setting it to *yes* can remove this network from the list of wireless networks that are shown by some client software. Changing this setting does not improve the security of the wireless network, because SSID is included in other frames sent by the AP.
hw-protection-mode (; Default:)	
hw-retries (; Default:)	
installation (*any* / *indoor* / *outdoor*; Default: **any**)	
keepalive-frames (*enabled* / *disabled*; Default: **enabled**)	
load-balancing-group (*string*; Default:)	Tags the interface to the load balancing group. For a client to connect to interface in this group, the interface should have the same number of already connected clients as all other interfaces in the group or smaller. Useful in setups where ranges of CAPs mostly overlap.
max-sta-count (*integer [1..2007]*; Default:)	Maximum number of associated clients.
mode (; Default: **ap**)	Set operational mode. Only ap currently supported.

multicast-helper (*default* | *disabled* | *full*; Default: **default**)	When set to full multicast packets will be sent with unicast destination MAC address, resolving multicast problem on a wireless link. This option should be enabled only on the access point, clients should be configured in **station-bridge** mode. Available starting from v5.15. • disabled - disables the helper and sends multicast packets with multicast destination MAC addresses • full - all multicast packet mac address are changed to unicast mac addresses prior sending them out • default - default choice that currently is set to *disabled*. Value can be changed in future releases.
name (*string*; Default:)	Descriptive name for the Configuration Profile
rates (; Default:)	User defined list taken from Rates names (**/caps-man rates**)
rates.basic (*1Mbps* | *2Mbps* | *5.5Mbps* | *6Mbps* | *11Mbps* | *11Mbps* | *12Mbps* | *18Mbps* | *24Mbps* | *36Mbps* | *48Mbps* | *54Mbps*; Default:)	
rates.supported (*1Mbps* | *2Mbps* | *5.5Mbps* | *6Mbps* | *11Mbps* | *11Mbps* | *12Mbps* | *18Mbps* | *24Mbps* | *36Mbps* | *48Mbps* | *54Mbps*; Default:)	
rates.ht-basic-mcs (*list of (mcs-0* | *mcs-1* | *mcs-2* | *mcs-3* | *mcs-4* | *mcs-5* | *mcs-6* | *mcs-7* | *mcs-8* | *mcs-9* | *mcs-10* | *mcs-11* | *mcs-12* | *mcs-13* | *mcs-14* | *mcs-15* | *mcs-16* | *mcs-17* | *mcs-18* | *mcs-19* | *mcs-20* | *mcs-21* | *mcs-22* | *mcs-23)*; Default: **mcs-0; mcs-1; mcs-2; mcs-3; mcs-4; mcs-5; mcs-6; mcs-7**)	Modulation and Coding Schemes that every connecting client must support. Refer to 802.11n for MCS specification.
rates.ht-supported-mcs (*list of (mcs-0* | *mcs-1* | *mcs-2* | *mcs-3* | *mcs-4* | *mcs-5* | *mcs-6* | *mcs-7* | *mcs-8* | *mcs-9* | *mcs-10* | *mcs-11* | *mcs-12* | *mcs-13* | *mcs-14* | *mcs-15* | *mcs-16* | *mcs-17* | *mcs-18* | *mcs-19* | *mcs-20* | *mcs-21* | *mcs-22* | *mcs-23)*; Default: **mcs-0; mcs-1; mcs-2; mcs-3; mcs-4; mcs-5; mcs-6; mcs-7; mcs-8; mcs-9; mcs-10; mcs-11; mcs-12; mcs-13; mcs-14; mcs-15; mcs-16; mcs-17; mcs-18; mcs-19; mcs-20; mcs-21; mcs-22; mcs-23**)	Modulation and Coding Schemes that this device advertises as supported. Refer to 802.11n for MCS specification.
rates.vht-basic-mcs (*none* | *MCS 0-7* | *MCS 0-8* | *MCS 0-9*; Default: **none**)	Modulation and Coding Schemes that every connecting client must support. Refer to 802.11ac for MCS specification. You can set MCS interval for each of Spatial Stream • *none* - will not use selected Spatial Stream • *MCS 0-7* - client must support MCS-0 to MCS-7 • *MCS 0-8* - client must support MCS-0 to MCS-8 • *MCS 0-9* - client must support MCS-0 to MCS-9
rates.vht-supported-mcs (*none* | *MCS 0-7* | *MCS 0-8* | *MCS 0-9*; Default: **none**)	Modulation and Coding Schemes that this device advertises as supported. Refer to 802.11ac for MCS specification. You can set MCS interval for each of Spatial Stream • *none* - will not use selected Spatial Stream • *MCS 0-7* - devices will advertise as supported MCS-0 to MCS-7 • *MCS 0-8* - devices will advertise as supported MCS-0 to MCS-8 • *MCS 0-9* - devices will advertise as supported MCS-0 to MCS-9
rx-chains (*list of integer [0..3]*; Default: **0**)	Which antennas to use for receive.
security (*string*; Default: **none**)	Name of security configuration from **/caps-man security**
security.authentication-types (*list of string*; Default: **none**)	Specify the type of Authentication from **wpa-psk**, **wpa2-psk**, **wpa-eap** or **wpa2-eap**
security.disable-pmkid (; Default:)	

security.eap-methods (*eap-tls* | *passthrough*; Default: **none**)	• eap-tls - Use built-in EAP TLS authentication. • passthrough - Access point will relay authentication process to the RADIUS server.
security.eap-radius-accounting (; Default:)	specifies if RADIUS traffic accounting should be used if RADIUS authentication gets done for this client
security.encryption (*aes-ccm* | *tkip*; Default:)	Set type of unicast encryption algorithm used
security.group-encryption (*aes-ccm* | *tkip*; Default: **aes-ccm**)	Access Point advertises one of these ciphers, multiple values can be selected. Access Point uses it to encrypt all broadcast and multicast frames. Client attempts connection only to Access Points that use one of the specified group ciphers. • tkip - Temporal Key Integrity Protocol - encryption protocol, compatible with legacy WEP equipment, but enhanced to correct some of the WEP flaws. • aes-ccm - more secure WPA encryption protocol, based on the reliable AES (Advanced Encryption Standard). Networks free of WEP legacy should use only this cipher.
security.group-key-update (*time: 30s..1h*; Default: **5m**)	Controls how often Access Point updates the group key. This key is used to encrypt all broadcast and multicast frames. property only has effect for Access Points.
security.passphrase (*string*; Default:)	WPA or WPA2 pre-shared key
security.tls-certificate (*none* | *name*; Default:)	Access Point always needs a certificate when **security.tls-mode** is set to value other than **no-certificates**.
security.tls-mode (*verify-certificate* | *dont-verify-certificate* | *no-certificates* | *verify-certificate-with-crl*; Default:)	This property has effect only when **security.eap-methods** contains *eap-tls*. • verify-certificate - Require remote device to have valid certificate. Check that it is signed by known certificate authority. No additional identity verification is done. Certificate may include information about time period during which it is valid. If router has incorrect time and date, it may reject valid certificate because router's clock is outside that period. See also the Certificates configuration. • dont-verify-certificate - Do not check certificate of the remote device. Access Point will not require client to provide certificate. • no-certificates - Do not use certificates. TLS session is established using 2048 bit anonymous Diffie-Hellman key exchange. • verify-certificate-with-crl - Same as verify-certificate but also checks if the certificate is valid by checking the Certificate Revocation List.
ssid (*string (0..32 chars)*; Default:)	SSID (service set identifier) is a name broadcast in the beacons that identifies wireless network.
tx-chains (*list of integer [0..3]*; Default: **0**)	Which antennas to use for transmit.

CAPsMAN datapath

Datapath settings control data forwarding related aspects. On CAPsMAN datapath settings are configured in the datapath profile menu **/caps-man datapath** or directly in a configuration profile or interface menu as settings with **datapath.** prefix.

There are 2 major forwarding modes:

- local forwarding mode, where CAP is locally forwarding data to and from wireless interface

- manager forwarding mode, where CAP sends to CAPsMAN all data received over wireless and only sends out the wireless data received from CAPsMAN. In this mode, even client-to-client forwarding is controlled and performed by CAPsMAN.

Forwarding mode is configured on a per-interface basis - so if one CAP provides 2 radio interfaces, one can be configured to operate in local forwarding mode and the other in manager forwarding mode. The same applies to Virtual-AP interfaces - each can have different forwarding mode from master interface or other Virtual-AP interfaces.

Most of the datapath settings are used only when in manager forwarding mode, because in local forwarding mode CAPsMAN does not have control over data forwarding.

There are the following datapath settings:

- bridge -- bridge interface to add interface to, as a bridge port, when enabled
- bridge-cost -- bridge port cost to use when adding as bridge port
- bridge-horizon -- bridge horizon to use when adding as bridge port
- client-to-client-forwarding -- controls if client-to-client forwarding between wireless clients connected to interface should be allowed, in local forwarding mode this function is performed by CAP, otherwise it is performed by CAPsMAN.
- local-forwarding -- controls forwarding mode
- openflow-switch -- OpenFlow switch to add interface to, as port when enabled
- vlan-id -- VLAN ID to assign to interface if vlan-mode enables use of VLAN tagging
- vlan-mode -- VLAN tagging mode specifies if VLAN tag should be assigned to interface (causes all received data to get tagged with VLAN tag and allows interface to only send out data tagged with given tag)

CAPsMAN interface

CAPsMAN interfaces are managed in **/caps-man interface** menu:

```
[admin@CM] > /caps-man interface print
Flags: M - master, D - dynamic, B - bound, X - disabled, I - inactive, R - running
# NAME RADIO-MAC MASTER-INTERFACE
0 M BR cap2 00:0C:42:1B:4E:F5 none
1 B cap3 00:00:00:00:00:00 cap2
```

CAPsMAN manager

Property	Description
enabled (*yes* | *no*; Default: **no**)	Disable or enable CAPsMAN functionality
certificate (*auto* | *certificate name* | *none*; Default: **none**)	Device certificate
ca-certificate (*auto* | *certificate name* | *none*; Default: **none**)	Device CA certificate
require-peer-certificate (*yes* | *no*; Default: **no**)	Require all connecting CAPs to have a valid certificate
package-path (*string* |; Default:)	Folder location for the RouterOS packages. For example, use "/upgrade" to specify the upgrade folder from the files section. If empty string is set, CAPsMAN can use built-in RouterOS packages, note that in this case only CAPs with the same architecture as CAPsMAN will be upgraded.
upgrade-policy (*none* | *require-same-version* | *suggest-same-upgrade*; Default: **none**)	Upgrade policy options • none - do not perform upgrade • require-same-version - CAPsMAN suggest to upgrade the CAP RouterOS version and if it fails it will not provision the CAP. (Manual provision is still possible) • suggest-same-version - CAPsMAN suggests to upgrade the CAP RouterOS version and if it fails it will still be provisioned

CAPsMAN provisioning

CAPsMAN distinguishes between CAPs based on a common-name identifier. The identifier is generated based on the following rules:

- if CAP provided a certificate, the identifier is set to the Common Name field in the certificate
- otherwise, an identifier is based on Base-MAC provided by CAP in the form: '[XX:XX:XX:XX:XX:XX]'.

When the DTLS connection with CAP is successfully established (which means that CAP identifier is known and valid), CAPsMAN makes sure there is no stale connection with CAP using the same identifier. Currently connected CAPs are listed in **/caps-man remote-cap** menu:

```
[admin@CM] /caps-man> remote-cap print
# ADDRESS IDENT STATE RADIOS 0 00:0C:42:00:C0:32/27044 MT-000C4200C032 Run 1
```

CAPsMAN distinguishes between actual wireless interfaces (radios) based on their built-in MAC address (radio-mac). This implies that it is impossible to manage two radios with the same MAC address on one CAPsMAN. Radios currently managed by CAPsMAN (provided by connected CAPs) are listed in **/caps-man radio** menu:

```
[admin@CM] /caps-man> radio print
Flags: L - local, P - provisioned
# RADIO-MAC INTERFACE REMOTE-AP-IDENT
0 P 00:03:7F:48:CC:07 cap1 MT-000C4200C032
```

When CAP connects, CAPsMAN at first tries to bind each CAP radio to CAPsMAN master interface based on radio-mac. If an appropriate interface is found, radio gets set up using master interface configuration and configuration of slave interfaces that refer to a particular master interface. At this moment interfaces (both master and slaves) are considered bound to radio and radio is considered provisioned.

If no matching master interface for radio is found, CAPsMAN executes 'provisioning rules'. Provisioning rules is an ordered list of rules that contain settings that specify which radio to match and settings that specify what action to take if a radio matches.

Provisioning rules for matching radios are configured in **/caps-man provisioning** menu:

Property	Description
action (*create-disabled* | *create-enabled* | *create-dynamic-enabled* | *none*; Default: **none**)	Action to take if rule matches are specified by the following settings: • **create-disabled** - create disabled static interfaces for radio. I.e., the interfaces will be bound to the radio, but the radio will not be operational until the interface is manually enabled; • **create-enabled** - create enabled static interfaces. I.e., the interfaces will be bound to the radio and the radio will be operational; • **create-dynamic-enabled** - create enabled dynamic interfaces. I.e., the interfaces will be bound to the radio, and the radio will be operational; • **none** - do nothing, leaves radio in the non-provisioned state;
comment (*string*; Default:)	Short description of the Provisioning rule
common-name-regexp (*string*; Default:)	Regular expression to match radios by common name. Each CAP's common name identifier can be found under "/caps-man radio" as value "REMOTE-CAP-NAME"
hw-supported-modes (*a|a-turbo|ac|an|b|g|g-turbo|gn*; Default:)	Match radios by supported wireless modes
identity-regexp (*string*; Default:)	Regular expression to match radios by router identity
ip-address-ranges (*IpAddressRange[, IpAddressRanges] max 100x*; Default: **""**)	Match CAPs with IPs within configured address range.
master-configuration (*string*; Default:)	If **action** specifies to create interfaces, then a new master interface with its configuration set to this configuration profile will be created

name-format (*cap* | *identity* | *prefix* | *prefix-identity*; Default: **cap**)	specify the syntax of the CAP interface name creation • cap - default name • identity - CAP boards system identity name • prefix - name from the name-prefix value • prefix-identity - name from the name-prefix value and the CAP boards system identity name
name-prefix (*string*; Default:)	name prefix which can be used in the name-format for creating the CAP interface names
radio-mac (*MAC address*; Default: **00:00:00:00:00:00**)	MAC address of radio to be matched, empty MAC (00:00:00:00:00:00) means match all MAC addresses
slave-configurations (*string*; Default:)	If **action** specifies to create interfaces, then a new slave interface for each configuration profile in this list is created.

⚠ If no rule matches radio, then implicit default rule with action **create-enabled** and no configurations set is executed.

To get the active provisioning matchers:

```
[admin@CM] /caps-man provisioning> print
Flags: X - disabled
0 radio-mac=00:00:00:00:00:00 action=create-enabled master-configuration=main-cfg
slave-configurations=virtual-ap-cfg name-prefix=""
```

For the user's convenience there are commands that allow the re-execution of the provisioning process for some radio or all radios provided by some AP:

```
[admin@CM] > caps-man radio provision 0
```

and

```
[admin@CM] > caps-man remote-cap provision 0
```

CAPsMAN radio

see /caps-man provisioning

CAPsMAN rates

see /caps-man configuration

CAPsMAN registration-table

Registration table contains a list of clients that are connected to radios controlled by CAPsMAN and is available in **/caps-man registration-table** menu:

```
[admin@CM] /caps-man> registration-table print
# INTERFACE MAC-ADDRESS UPTIME RX-SIGNAL
0 cap1 00:03:7F:48:CC:0B 1h38m9s210ms -36
```

CAPsMAN remote-cap

see /caps-man provisioning

CAPsMAN security

Example

Assuming that rest of the settings are already configured and only the "Security" part has been left.

Radius authentication with one server

1. Create CAPsMAN security configuration

2. Configure Radius server client

3. Assign the configuration to your master profile (or directly to CAP itself)

```
/caps-man security add authentication-types=wpa2-eap eap-methods=passthrough encryption=aes-ccm group-
encryption=aes-ccm name=radius
/radius add address=x.x.x.x secret=SecretUserPass service=wireless
/caps-man configuration set security=radius
```

Radius authentication with different radius servers for each SSID

1. Create CAPsMAN security configuration

2. Configure AAA settings

3. Configure Radius server clients

4. Assign the configuration to your master profile (or directly to CAP itself)

```
/caps-man security add authentication-types=wpa2-eap eap-methods=passthrough encryption=aes-ccm group-
encryption=aes-ccm name=radius
/caps-man aaa set called-format=ssid
/radius add address=x.x.x.x secret=SecretUserPass service=wireless called-id=SSID1
/radius add address=y.y.y.y secret=SecretUserPass service=wireless called-id=SSID2
/caps-man configuration set security=radius
```

Now everyone connecting to CAP's with ssid=**SSID1** will have their radius authentication requests sent to **x.x.x.x** and everyone connecting to CAP's with ssid=**SSID2** will have their radius authentication requests sent to **y.y.y.y**