OpenSync 2.2 Northbound API

Date: August 10, 2020
Document ID: EUB-020-033-101

Table of Contents

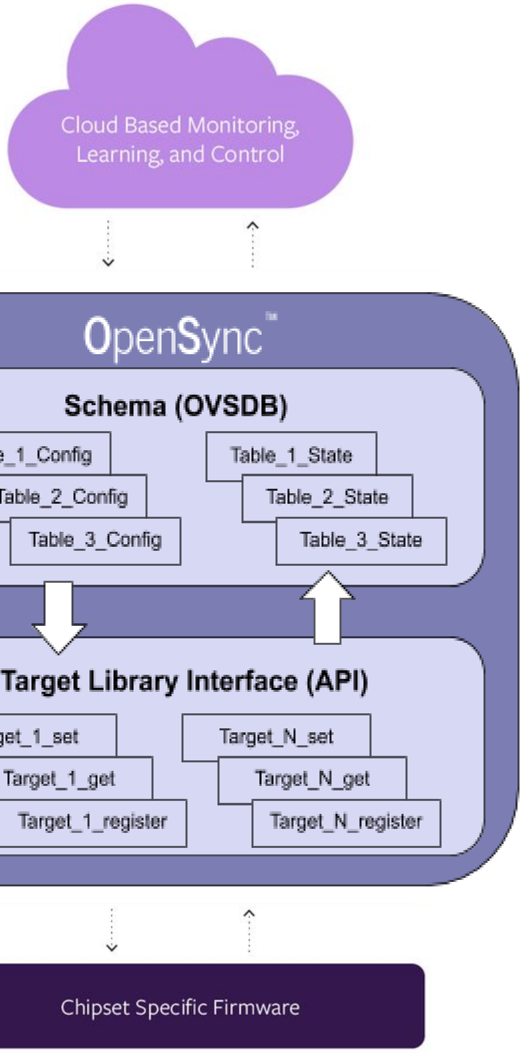
References

[1] http://docs.openvswitch.org/en/latest/ref/ovsdb.7/

[2] EUB-020-013-001 OpenSync Overview.pdf

Introduction

OpenSync^{TM} is a cloud-agnostic open-source software for the delivery, curation, and management of wireless services for the home. It follows the SDN concept where the OVSDB serves as interprocess communication between services in the cloud and *OpenSync* managers on the device, as illustrated in the image below.



Tables in the Cloud schema are roughly organized into two groups - “config” and “state” tables. Usually, the cloud writes instructions and configuration into the config tables and the device must report its status in the state tables.

List of Tables

Name	Description
Alarms	Used for reporting custom-defined alarms to the cloud
AutoAttach	*Default OVSDB table - LLDP and SPB settings* *(Not used)*
AW_Bluetooth_Config	Contains Bluetooth configuration
AW_Bluetooth_State	Reflects the status of currently applied Bluetooth configuration *(Not used)*
AW_Debug	Setting log levels to registered modules
AWLAN_Node	Stores device entity information and configuration, and cloud connection configuration
AW_LM_Config	Used for logpull functionality
AW_LM_State	*(Deprecated)*
Band_Steering_Clients	Per client band steering configuration and states
Band_Steering_Config	Configuration of band steering and client steering parameters
BeaconReport	*(Not used)*
Bridge	*Default OVSDB table - Configuration for a bridge within the Open_vSwitch table*
Client_Freeze_Config	List of clients for device freeze when OpenFlow is not used
Client_Nickname_Config	List of client nicknames
Connection_Manager_Uplink	Provides info and status of the uplink connection
Controller	*Default OVSDB table - An OpenFlow controller*
DHCP_leased_IP	Contains client information obtained through DHCP fingerprinting
DHCP_reserved_IP	Contains DHCP IP reservation for specific clients MAC addresses
DHCP_Option	*Used for storing DHCPv6 and DHCPv4 options*
DHCPv4_Client	*(Not used)*
DHCPv4_Lease	*(Not used)*
DHCPv4_Server	*(Not used)*
DHCPv6_Client	Used for DHCPv6 client configuration and status reporting
DHCPv6_Lease	Used for reporting DHCPv6 leases. Referenced by DHCPv6_Server

DHCPv6_Server	Used for DHCPv6 server configuration and status reporting
FCM_Collector_Config	Used for loading and configuring collector plugins within FCM
FCM_Filter	Used for configuring sampling and collector filters for statistics in FCM
FCM_Report_Config	Used for configuring statistics reports from FCM to the cloud
Flow_Service_Manager_Config	Used for loading and configuring FSM plugins
Flow_Sample_Collector_Set	*Default OVSDB table - used only for IPFIX flow-based sampling* *(Not used)*
Flow_Table	*Default OVSDB table - configuration for a particular OpenFlow table* *(Not used)*
FSM_Policy	Used for configuring policies applied by FSM plugins
Interface	*Default OVSDB table - An interface within a Port*
IPFIX	*Default OVSDB table - configuration for sending packets to IPFIX collectors* *(Not used)*
IP_Interface	IP Interface configuration - used only for IPv6
IP_Port_Forward	User-defined port forwarding rules
IPv4_Address	*(Not used)*
IPv4_Neighbors	Used for storing the ARP mappings exchange
IPv6_Address	Used for storing various IPv6 addresses - referenced by other tables
IPv6_Neighbors	Used for IPv6 neighbor reporting
IPv6_Prefix	Used for storing various IPv6 prefixes - referenced by other tables
IPv6_RouteAdv	Used for IPv6 Router Advertisement configuration
IGMP_Config	Configures IGMP parameters in the system (available in *OpenSync 2.0* and later)
Manager	Contains information required to establish and keep a connection to the Cloud
MLD_Config	Configure MLD parameters in the system (available in *OpenSync 2.0* and later)
Mirror	*Default OVSDB table - port mirror within a Bridge* *(Not used)*
NetFlow	*Default OVSDB table - NetFlow target configuration* *(Not used)*
Node_Services	Contains services (managers) that are available on the device (available in *OpenSync 2.0* and later)
Node_State	Vendor-specific state table (used for third-party integration)
Openflow_Config	Contains configuration for all OVS flows that need to be applied

Openflow_State	Reflects the status of all currently applied flows
Openflow_Tag	Used for packet flow rules expansion
Openflow_Tag_Group	Combines a list of tags defined in the Openflow_Tag table to create a new group tag
Open_vSwitch	Describes the switch as a whole
OVS_MAC_Learning	Used to report the bridge MAC learning table to the cloud
Port	*Default OVSDB table - port within a Bridge*
QoS	Default OVSDB table - Quality of Service configuration for each Port that refers to it *(Not used)*
Captive_Portal	This table configures Captive Portal authentication parameters on the devices.
Reboot_Status	Device boot/reboot reason, status reporting, and crash statistics
Queue	*Default OVSDB table - Configuration for a port output queue, used in configuring QoS features. May be referenced by the queues column in the QoS table.* *(Not used)*
Service_Announcement	Enables mDNS service announcing (available in *OpenSync 2.0* and later)
sFlow	*Default OVSDB table - a set of sFlow® targets.* *(Not used)*
SSL	Information about certificates, used for SSL/TLS connections
Wifi_Associated_Clients	List of all connected clients and their MAC addresses
Wifi_Credential_Config	Credential details
Wifi_Ethernet_State	*(Not used)*
Wifi_Inet_Config	Contains system-wide network configuration.
Wifi_Inet_State	Reflects the status of currently applied network configuration
Wifi_Master_State	Reflects current states of network interfaces/bridges (up/down)
Wifi_Radio_Config	Contains system-wide wireless physical (radio) configuration
Wifi_Radio_State	Reflects the status of currently applied wireless physical (radio) configuration
Wifi_Route_State	Contains ARP table information
Wifi_Speedtest_Config	Speedtest configuration table. Only used if speedtest is supported. *(Not part of OpenSync)*
Wifi_Speedtest_Status	Reflects the status of speedtest results on the device. Only used if speedtest is supported. *(Not part of OpenSync)*

Wifi_Stats_Config	Configuration for stats collecting
Wifi_Test_Config	*(Not used)*
Wifi_Test_State	*(Not used)*
Wifi_VIF_Config	Contains system-wide wireless interface (AP, STA) configuration
Wifi_VIF_Neighbors	Contains neighboring AP information for steering and other features
Wifi_Channels	Enables configuration of maximum Tx power per radio from the Cloud
Wifi_VIF_State	Reflects the status of currently applied wireless interface (AP, STA) configuration

Alarms

The *Alarms* table is used for reporting custom-defined alarms to the cloud.

Name	Type	Description
code	string	Alarm code
timestamp	integer	Timestamp of the alarm
source	string	Source of the alarm
add_info	string	Additional information

AW_Bluetooth_Config

AW_Bluetooth_Config is a configurational table where we can set payload in the BLE broadcast frame. CM2 uses this table to notify the onboarding state.

Name	Type	Description
mode	enum (*on*, *off*)	Enable/disable sending BLE broadcast frames
interval_millis	integer	Interval time in msec, how often BLE broadcast is sent 0 - default value
connectable	boolean	Enables bidirectional communication without changing the target API.
txpower	integer	Tx power 0 - default value.
command	enum (*on_boarding*, *diagnostic*, *locate*)	In *OpenSync 2.0* only *on_boarding* is used
payload	string	Payload of broadcast frame

Example:

```
# ovsh s AW_Bluetooth_Config
-------------------------------------
_uuid           | 0c63~06b2         |
_version        | d2a9~62c7         |
command         | on_boarding       |
connectable     | false             |
interval_millis | 0                 |
mode            | off               |
payload         | 75:00:00:00:00:00 |
txpower         | 0                 |
------------------------------------
```

AWLAN_Node

AWLAN_Node is a configurational and informational table that stores all *device_entity* information. The table also stores information about the Cloud connection specifics. Once the *redirector_address* is specified, the Cloud configures connection information through *manager_address* (See Connection Manager - CM [2] for details), and specifies the MQTT configuration (See Queue Manager - QM [2] for details).

Name	Type	Description
id	string	Device identification number - **serial_number** used by the Cloud to differentiate device management (could also be MAC address)
device_mode	enum (*cloud*, *monitor*, *battery*, *custom*)	Different devices operate in different modes. This requires special handling in the Cloud: ● **cloud** - full control including reading statistics, configuring channels, and managing backhaul POD topology changes - Adaptive Wifi ● **monitor** - (or read-only mode) is a mode where only certain statistics are collected and/or channels are managed - depends on customer requirements ● **battery** - devices running on battery need to save power, therefore, the cloud turns off most of the functionality ● **custom** - custom mode, individual features and responsibilities are governed via profile in the Cloud
factory_reset	boolean	Indicates the status of the device after factory reset has been issued
firmware_url	string	The descriptor in URL format indicating a new SW release download location *(Only applicable when Upgrade Manager is used)*
firmware_pass	string	Password for firmware-encrypted images
firmware_version	string	*OpenSync* firmware package version
platform_version	string	Joined platform firmware version containing *OpenSync*
serial_number	string	Device serial number as seen on the BAR code
model	string	Device-friendly name used in the network
revision	string	Device HW revision
sku_number	string	Device SKU number/revision
version_matrix	key/value map	Detailed firmware version description including ● *OpenSync* version ● Date of generation ● Firmware package version

		● Build number ● Source code repository version details ● other details
upgrade_dl_timer		Specifies upgrade behavior *(Only applicable when Upgrade Manager is used)*
upgrade_status		
upgrade_timer		
redirector_addr	string	The cloud redirector address used to differentiate between different deployments (production or development). The syntax is : **<protocol>:<cloud>:<port>**
manager_addr		
mqtt_headers	key/value map	
mqtt_settings	key/value map	parameters: ● broker: hostname ● port: port number ● compress: none, zlib ● topics: topic name ● QoS: 0, 1, 2
mqtt_topics	key/value map	
led_config	key/value map	Specifies LED behavior *(Only applicable when the LED Manager is used)*
min_backoff	integer	CM reconnect backoff timer configuration
max_backoff	integer	
vendor_name	string	Device vendor name. Up to 64 characters can be used.
vendor_part_number	string	Device model as defined by the vendor. Up to 64 characters can be used.
vendor_manufacturer	string	Name of the device manufacturer. Up to 64 characters can be used.
vendor_factory	string	Factory at which the device has been produced. Up to 64 characters can be used.
vendor_mfg_date	string	Manufacturing date. Up to 16 characters can be used.
redirector_addr	string	Redirector url. The ovsdb-server supports IPv4 and IPv6 connections.

Example:

```
# ovsh s AWLAN_Node
-------------------------------------------------------------------------------------------------
_uuid               | c765~772b                                                                   |
_version            | 2756~66c7                                                                   |
device_mode         | ["set",[]]                                                                  |
factory_reset       | ["set",[]]                                                                  |
firmware_pass       |                                                                             |
firmware_url        |                                                                             |
firmware_version    | 3.2.0-17-g4db38f7-dev-debug                                                 |
id                  | 4C71100117                                                                  |
led_config          | ["map",[["state","connecting"]]]                                            |
manager_addr        | ssl:52.36.197.250:443                                                       |
max_backoff         | 60                                                                          |
min_backoff         | 30                                                                          |
model               | PP203X                                                                      |
mqtt_headers        | ["map",[]]                                                                  |
mqtt_settings       | ["map",[]]                                                                  |
mqtt_topics         | ["map",[]]                                                                  |
platform_version    | 1                                                                           |
redirector_addr     | ssl:wildfire.plume.tech:443                                                 |
revision            | DVT.1                                                                       |
serial_number       | 4C71100117                                                                  |
sku_number          | 945100800                                                                   |
upgrade_dl_timer    | 0                                                                           |
upgrade_status      | 0                                                                           |
upgrade_timer       | 0                                                                           |
vendor_factory      | Shanghai-1                                                                  |
vendor_manufacturer | CIG                                                                         |
vendor_mfg_date     | 2018/9                                                                      |
vendor_name         | Plume                                                                       |
vendor_part_number  | PP203X                                                                      |
version_matrix      | ["map",[["3rdparty/cognitive","0.0/=30/g41a5b1a"],["3rdparty/plume",        |
                    : "0.0/=140/g6647e24"],["3rdparty/plume-private","0.0/=27/g6d9fd8c"],         :
                    : ["3rdparty/symantec","0.0/=99/g314494e"],["3rdparty/walleye",               :
                    : "0.0/=61/g0b7a2fa"],["3rdparty/webroot","0.0/=28/g6a88e49"],["DATE",        :
                    : "Tue Aug  4 20:02:36 UTC 2020"],["FIRMWARE","3.2.0-17-g4db38f7-dev-debug"], :
                    : ["FW_BUILD","17"],["FW_COMMIT","g4db38f7"],["FW_PROFILE","dev-debug"],      :
                    : ["FW_VERSION","3.2.0"],["HOST",                                             :
                    : "jenkins@jenkins-jenkins-slave-xenial-04.inf.us-west-2.aws.plume"],          :
                    : ["OPENSYNC","2.2.0.0"],["core","2.2.0.0/=3843/g39ebfc6"],["device",         :
                    : "0.0/=5534/g4db38f7"],["platform/qca","0.0/=482/g22dd2a8"],["sdk/qsdk53",   :
                    : "0.0/=1014/g45c7c14"],["vendor/plume","3.3.0/=2996/gd847b39"]]]             :

-------------------------------------------------------------------------------------------------
```

Open_vSwitch

When *Open_vSwitch* is used, this table contains the reference to the networking elements (Details Network Manager - NM [2]).

Name	Type	Description
bridges	uuid	Reference to the table: Bridge
manager_options	uuid	Reference to the table: Manager
ssl	uuid	Reference to the table: SSL
other_config	string	Additional configuration
external_ids	string	*Not used by CM2*
next_cfg	integer	*Not used by CM2*
cur_cfg	integer	Version of the current configuration
statistics	string	*Not used by CM2*
ovs_version	string	Version of OVS
db_version	string	Version of database
dpdk_initialized	boolean	
dpdk_version	string	
system_type	string	*Not used by CM2*
system_version	string	
datapath_types	string	*Not used by CM2*
iface_types	string	Supported type of interfaces

Example:

```
# ovsh s Open_vSwitch
-----------------------------------------------------------------
_uuid            | f5eb~58cc                                                                       |
_version         | 84c9~97d5                                                                       |
bridges          | [ca0e~2356,ca66~5670]                                                           |
cur_cfg          | 4                                                                               |
datapath_types   | ["set",["netdev","system"]]                                                     |
db_version       | ["set",[]]                                                                      |
dpdk_initialized | false                                                                           |
dpdk_version     | ["set",[]]                                                                      |
external_ids     | ["map",[]]                                                                      |
iface_types      | ["set",["geneve","gre","internal","lisp","patch","stt","system","tap","vxlan"]] |
manager_options  | 569e~bbda                                                                       |
next_cfg         | 4                                                                               |
other_config     | ["map",[["stats-update-interval","3600000"]]]                                   |
ovs_version      | ["set",[]]                                                                      |
ssl              | fc24~5e95                                                                        |
statistics       | ["map",[]]                                                                       |
system_type      | ["set",[]]                                                                       |
system_version   | ["set",[]]                                                                       |

-----------------------------------------------------------------
```

SSL

SSL table contains Information with the path of currently used certificates.

Name	Type	Description
private_key	string	Path to private key
certificate	string	Path to certificate
ca_cert	string	Path to CA certificate
bootstrap_ca_cert	boolean	Enable/disable bootstrap certificate
external_ids	sting	*Not used by CM2*

Example:

```
# ovsh s SSL
-----------------------------------------------
_uuid             | 8e9a~8be8                 |
_version          | c61b~d046                 |
bootstrap_ca_cert | false                     |
ca_cert           | /var/certs/ca.pem         |
certificate       | /var/certs/client.pem     |
external_ids      | ["map",[]]                |
private_key       | /var/certs/client_dec.key |
-----------------------------------------------
```

Manager

The table contains information required to establish and keep the cloud connection.The Manager table status field is a good source during debugging the Cloud connection.

Name	Type	Description
target	string	Direct SSL address to the Cloud
max_backoff	integer	Reconnect backoff timer configuration *Not used by CM2*
inactivity_probe	integer	Inactive timer, default value: 30000 ms
connection_mode	enum (*in-band*, *out-of-band*)	*Not used by CM2*
other_config	string	*Not used by CM2*
external_ids	sting	*Not used by CM2*
is_connected	boolean	Connection state
status	string	Connection status

Example:

```
# ovsh s Manager
--------------------------------------------------------------------------------
_uuid            | 0d1e~0c85                                                  |
_version         | ba1a~efba                                                  |
connection_mode  | ["set",[]]                                                 |
external_ids     | ["map",[]]                                                 |
inactivity_probe | 30000                                                      |
is_connected     | true                                                       |
max_backoff      | ["set",[]]                                                 |
other_config     | ["map",[]]                                                 |
status           | ["map",[["sec_since_connect","16001"],["state","ACTIVE"]]] |
target           | ssl:IP:443                                                 |
--------------------------------------------------------------------------------
```

Connection_Manager_Uplink

The *Connection_Manager_Uplink* table provides info and status of the uplink connection.

Name	Type	Description
bridge	string	Manages WAN bridges for the main active link.
has_L2	boolean	physical link state
has_L3	boolean	uplink available
if_name	string	interface name
if_type	enum (*bridge*, *eth*, *vif*, *gre*, *pppoe*, *softwds*, *tap*, *vlan*)	interface type
is_used	boolean	link used as a main link
loop	boolean	used for ethernet link, notify potential loop if link will be used in br-home bridge
ntp_state	boolean	system clock valid (NTP success)
priority	integer	link priority, link with the highest priority should be used as a main link
unreachable_cloud_counter	integer	stats
unreachable_internet_counter	integer	stats
unreachable_link_counter	integer	stats
unreachable_router_counter	integer	stats

Example:

```
# ovsh s Connection_Manager_Uplink
-------------------------------------------
_uuid                        | 8478~2cb0  |
_version                     | b513~574a  |
bridge                       | br-wan     |
has_L2                       | true       |
has_L3                       | true       |
if_name                      | eth0       |
if_type                      | eth        |
is_used                      | true       |
loop                         | ["set",[]] |
ntp_state                    | true       |
priority                     | 2          |
unreachable_cloud_counter    | 0          |
unreachable_internet_counter | 0          |
unreachable_link_counter     | 0          |
unreachable_router_counter   | 0          |
-------------------------------------------
```

Wifi_Radio_Config

Wifi_Radio_Config table that stores the system-wide wireless radio (physical) configuration.

Name	Type	Description
if_name	string	Interface name
freq_band	enum (*2.4G*, *5G*, *5GL*, *5GU*)	Frequency band of the specified interface. If two separate 5 GHz radios are available on the device, these radios should be specified as "lower": **5GL** or "upper": **5GU**.
enabled	boolean	The desired interface state. "True" if the interface should be configured.
dfs_demo	boolean	Controls the demo mode for DFS.
hw_type	string	The descriptor of the hardware type for a specific wifi radio. For example this is useful e.g. for differentiating the configurations for different hardware chips.
hw_config	string	Configuration type, complementing the "hw_type" parameter
country	string	Two-letter country code descriptor for setting the wireless regulatory domain. Examples: "AT", "AU", "BE", etc.
channel	integer	Channel number of the specified radio. The available channels list depends on the frequency band, regulatory domain, and channel width.
channel_sync	integer	*Deprecated*
channel_mode	enum (*auto*, *manual*, *cloud*)	Channel mode; specifies the method using which the channel is set: ● **auto**: device sets channel config automatically ● **manual:** manually override channel config ● **cloud:** the Cloud pushes channel configurations ● **ACS:** auto channel selection activated
hw_mode	enum (*11a*, *11b*, *11g*, *11n*, *11ab*, *11ac*)	Specifies the IEEE 802.11 standard used for the specified radio.
ht_mode	enum (*HT20*, *HT2040*, *HT40*, *HT40+*, *HT40-*, *HT80*, *HT160*, *HT80+80*)	Stands for "high throughput mode"; specifies the bandwidth of the selected channel, in MHz: ● **HT20**: single 20 MHz channel ● **HT2040**: *deprecated* ● **HT40:** dual 20 MHz channels, automatic selection of upper or lower secondary channel ● **HT40-:** dual 20 MHz channels, upper channel is primary or control ● **HT40+:** dual 20 MHz channels, lower channel is primary or control ● **HT80:** dual 40 MHz channels

		● **HT160:** dual 80 MHz channels ● **HT80+80:** *deprecated*
thermal_shutdown	integer	Temperature in degrees (℃) at which the unit shuts down
thermal_downgrade_temp	integer	Temperature at which the tx_chainmask is reduced
thermal_upgrade_temp	integer	Temperature at which the tx_chainmask is increased
thermal_integration	integer	Time required for the device to upgrade or downgrade chainmask after exceeding the temperature threshold
temperature_control	integer	*deprecated*
vif_configs	set if uuids	Array of VIF uuids linked to entries (columns) in the *Wifi_VIF_Config* table
tx_power	integer	Transmission power of the specified radio (in dBm)
bcn_int	integer	Beacon interval in TU (time units), which depend on the system, e.g., TU = 1.024 ms
tx_chainmask	integer	Transmission chainmask; bitmask that specify which radio chains are used for transmission
zero_wait_dfs (available in *OpenSync 2.0* and later)	enum	Adds zero-wait DFS options. ● "disable": default value. Executes the channel switch, switches channel and runs CAC. ● "enable": If the cloud enables this and chooses a DFS channel in Wifi_Radio_Config::channel the device runs background CAC. In case of success, the device switches to a new DFS channel. ● "precac": extends the “enable” mode. Runs background CAC on all DFS channels.

Example:

```
 # ovsh s Wifi_Radio_Config
----------------------------------------------------------------------------------------
_uuid                  | 5ee4~f412                       | a8e2~6fb9                   |
_version               | 121a~9f4d                       | bffa~b76c                   |
bcn_int                | ["set",[]]                      | ["set",[]]                  |
channel                | 1                               | 44                          |
channel_mode           | cloud                           | cloud                       |
channel_sync           | 0                               | 0                           |
country                | ["set",[]]                      | ["set",[]]                  |
enabled                | true                            | true                        |
freq_band              | 2.4G                            | 5G                          |
ht_mode                | HT40                            | HT80                        |
hw_config              | ["map",[["cwm_extbusythres",    | ["map",[["dfs_enable","1"], |
                       : "30"]]]                         : ["dfs_ignorecac","1"],      :
                       :                                 : ["dfs_usenol","2"]]]        :
hw_mode                | 11n                             | 11ac                        |
hw_type                | qca9563                         | qca9882                     |
if_name                | wifi0                           | wifi1                       |
temperature_control    | ["map",[]]                      | ["map",[]]                  |
thermal_downgrade_temp | ["set",[]]                      | ["set",[]]                  |
thermal_integration    | ["set",[]]                      | ["set",[]]                  |
thermal_shutdown       | ["set",[]]                      | ["set",[]]                  |
thermal_upgrade_temp   | ["set",[]]                      | ["set",[]]                  |
tx_chainmask           | ["set",[]]                      | ["set",[]]                  |
tx_power               | ["set",[]]                      | ["set",[]]                  |
vif_configs            | [0101~e570,abfd~860b,d22e~2caa] | [7b34~bec6,ed43~1967]       |
----------------------------------------------------------------------------------------
```

Wifi_Radio_State

Wifi_Radio_State is a table which represents the actual state of the device and is therefore read by the Cloud. The table is pushed by WM2 and reflects the current system-wide wireless radio (physical) configuration.

Entries marked in gray differ from the *Wifi_Radio_Config* table, while the other rows are equal.

Name	Type	Description
if_name	string	Interface name
radio_config	uuid	The uuid linked to the entity (column) in the *Wifi_Radio_config* table
freq_band	Enum (*2.4G*, *5G*, *5GL*, *5GU*)	The frequency band of the specified interface. If two separate 5GHz radios are available on the device, they should be specified as “lower”: **5GL** or “upper”: **5GU**.
enabled	boolean	The desired interface state. “True” if the interface should be in the UP state; otherwise the interface should be in the DOWN state (the equivalent of ifconfig up/down)
dsf_demo	boolean	Controls the demo mode for DFS
hw_type	string	Hardware type for a specific wifi radio. For example, use this value for differentiating the configurations between different hardware chips.
hw_params	string	Field for hardware-specific configuration
radar	string	Field for DFS-specific information
hw_config	string	Configuration type, complementing the “hw_type” parameter
country	string	Two-letter country code descriptor for setting wireless regulatory domain. Examples: “AT”, “AU”, “BE”, etc.
channel	integer	Channel number of the specified radio. The available channels list depends on the frequency band, regulatory domain, and channel width.
channel_sync	integer	TBD
channel_mode	enum (*auto*, *manual*, *cloud*)	Channel mode; specifies the method with which the channel is set: ● **auto:** device sets channel config automatically ● **manual:** manually override channel config ● **cloud:** the Cloud pushes channel configurations
mac	string	MAC address of the specified radio
hw_mode	enum	Specifies the IEEE 802.11 standard used for the specified radio

	(*11a*, *11b*, *11g*, *11n*, *11ab*, *11ac*)	
ht_mode	enum (*HT20*, *HT2040*, *HT40*, *HT40+*, *HT40-*, *HT80*, *HT160*, *HT80+80*)	Stands for “high throughput mode”; specifies the bandwidth of the selected channel, in MHz: ● **HT20**: single 20 MHz channel ● **HT2040**: *deprecated* ● **HT40:** dual 20 MHz channels, automatic selection of upper or lower secondary channel ● **HT40-:** dual 20 MHz channels, upper channel is primary or control ● **HT40+:** dual 20 MHz channels, lower channel is primary or control ● **HT80:** dual 40 MHz channels ● **HT160:** dual 80 MHz channels ● **HT80+80:** *deprecated*
thermal_shutdown	integer	Temperature in degrees (℃), at which the unit shuts down
thermal_downgrade_temp	integer	Temperature at which the tx_chainmask is reduced
thermal_upgrade_temp	integer	Temperature at which the tx_chainmask is increased
thermal_integration	integer	Time required for the device to upgrade or downgrade the chainmask after exceeding the temperature threshold
thermal_downgraded	boolean	Information if device chainmask is currently downgraded due to the exceeded thermal threshold
temperature_control	integer	*Deprecated*
vif_states	set of uuids	Array of VIF uuids linked to the entries (columns) in the Wifi_VIF_State table
tx_power	integer	Transmission power of the specified radio (in dBm).
bcn_int	integer	Beacon interval in TU (time units), which depend on the system, e.g. TU = 1.024 ms
tx_chainmask	integer	Transmission chainmask: bitmask to specify which radio chains are used for transmission
thermal_tx_chainmask	integer	Value used by thermal manager
allowed_channels	integer	The list of allowed channels for the specified interface. The list depends on several other parameters, like frequency and regulatory domain.
channels	map	The map is keyed with channel numbers to json state descriptions. The json is expected to be a dictionary and should include “state” keys with one of the 4 values: ● allowed: non-dfs channel ● nop_finished: dfs channel, requires cac before using ● cac_completed: dfs channel, cac completed, usable ● nop_started: dfs channel, radar was detected and it must not be used
zero_wait_dfs (available	enum	Adds zero-wait DFS options.

in *OpenSync 2.0* and later)		● "disable": default value. Executes the channel switch, switches channel and runs CAC. ● "enable": If the cloud enables this and chooses a DFS channel in Wifi_Radio_Config::channel the device runs background CAC. In case of success, the device switches to a new DFS channel. ● "precac": extends the “enable” mode. Runs background CAC on all DFS channels.

Wifi_Speedtest_Config

An update to this table configures and initiates a speedtest, either using Ookla or iPerf.

Name	Type	Description
select_server_id	integer	Ookla: Initiate Ookla speedtest with a specific server ID.
preferred_list	integer list	Ookla: a list of Ookla server IDs that will be preferred for the speedtest.
st_bw	integer	iPerf: target bandwidth (0 for unlimited) for UDP tests. [bits/s]
st_dir	enum (DL, UL, DL_UL)	iPerf: Specify direction (download, upload, both)
st_len	integer	iPerf: Time in seconds to transmit - test length (one way)
st_parallel	integer	iPerf: number of parallel client streams to run
st_pkt_len	integer	iPerf: MTU - 40 bytes
st_port	integer	iPerf: server port to listen on/connect to
st_server	string	iPerf: server host or IP to connect to or bind to
st_udp	boolean	iPerf: use UDP rather than TCP

Note: This table is available in *OpenSync 2.0* and later.

Wifi_Speedtest_Status

This table serves to report speedtest results.

Name	Type	Description
DL_duration	real	Duration of download part of speedtest (in s)
DL_jitter	real	iPerf: measured jitter in [ms] - download direction
DL_pkt_loss	real	iPerf: packet loss [percent] - download direction
UL_duration	real	Duration of upload part of speedtest (in s)
UL_jitter	real	iPerf: measured jitter (in ms) - upload direction
UL_pkt_loss	real	iPerf: packet loss (percentage) - upload direction
host_remote	string	iPerf: remote host IP
pref_selected	boolean	Ookla: preferred server was used
hranked_offered	boolean	Ookla: preferred server was chosen, even if non-preferred server was offered in a higher rank
jitter	real	Ookla: measured jitter (in ms)

Note: This table is available in *OpenSync 2.0* and later.

Wifi_VIF_Config

The *Wifi_VIF_Config* table is used for the system-wide wireless (logical) interface - AP and STA configuration.

Name	Type	Description
if_name	string	Interface name
enabled	boolean	Defines if the specified interface is enabled or not
mode	enum (*ap*, *monitor*, *sta*)	Interface mode: ● **ap:** access point ● **monitor:** monitor (promiscuous) mode - *not used* ● **sta:** station (client)
parent	string	Parent BSSID to which the STA needs to be connected (*Only for extenders*). If empty, any BSSID is allowed to match.
vif_radio_idx	integer	Consecutive radio index per VIF type, e.g. ● 1=bhaul-ap, ● 2=home-ap, etc.
vif_dbg_lvl	integer	TBD
wds	boolean	*Deprecated*
ssid	string	Interface SSID
ssid_broadcast	enum (*enabled*, *disabled*, *disabled_null*)	Determines if interface SSID is broadcast or not
security	key/value map	The map of the string specifying security parameters ● Encryption ○ OPEN ○ WEP ○ WPA-PSK ○ WPA-EAP ● Key (password) ● Mode (PSK 1, 2 or mixed) TBD: EAP modes and radius config
credential_configs	uuid	Wifi_Credential_Config. This is for extender only. This column is respected only if “ssid” is empty.
bridge	string	Specifies the bridge interface on which the authentication listens for EAPOL.
mac_list	string	ACL MAC list for layer 2 packet filtering
mac_list_type	enum (*whitelist*,	How to filter the MAC addresses:

	blacklist, *none*)	● **whitelist:** only forward traffic from addresses in *mac_list*, block all other traffic ● **blacklist:** only block traffic from addresses in *mac_list*, forward all other traffic ● **none:** no filtering
vlan_id	integer	Identification number for VLAN-tagged traffic
min_hw_mode	enum (*11b*, *11g*, *11a*, *11n*, *11ac*)	Minimal required IEEE 802.11 standard for the specified interface
uapsd_enable	boolean	Enable uAPSD power saving capability for the specified interface
group_rekey	integer	Time in seconds between the GTK rekeying. Valid for AP mode only. Empty or -1 means default, 0 means disabled, >0 is interval in seconds.
ap_bridge	boolean	AP isolation (intra-bss communication)
ft_psk	integer	"Fast transition" (802.11R) pre-shared key
ft_mobility_domain	integer	"Fast transition" (802.11R) mobility domain
btm	integer (0, 1)	Enable (1) or disable (0) WNM BSS Transition support.
rrm	integer (0, 1)	Enable (1) or disable (0) RRM support. Current minimum requirement is to enable RRM Neighbor Report support.
dynamic_beacon	boolean	Only applicable unless *ssid_broadcast* is enabled. When enabled, stops beaconing until a non-ap sta attempts to associate.
mcast2ucast (available in *OpenSync* 2.0 and later)	boolean	Enables multicast-to-unicast packet conversion: ● 0: disabled ● 1: enabled
multi_ap (available in *OpenSync* 2.0 and later)	string	Device type as defined by the Multi AP specification: ● Backhaul STA ● Backhaul BSS ● Fronthaul BSS
wps (available in *OpenSync* 2.0 and later)	boolean	Enable support for WPS on AP (i.e. broadcast support for WPS in beacons): ● "True": Enable WPS ● "False": Disable WPS
wps_pbc (available in *OpenSync* 2.0 and later)	boolean	Initiate WPS single session: ● "True": Start WPS session ● "False": Cancel ongoing WPS session
wps_pbc_key_id (available in *OpenSync* 2.0 and later)	string	The value must match one of the password key IDs stored in the security field of the AP's configuration.

Example:

```
root@caesar:~# ovsh s Wifi_VIF_Config -c -w if_name==bhaul-ap-24
---------------------------------------------------------------------------------------------------------------------------
---------------------
_uuid              : e960~1e9a
_version           : f01d~31e0
ap_bridge          : ["set",[]]
bridge             : ["set",[]]
btm                : ["set",[]]
Credential_configs : ["set",[]]
dynamic_beacon     : false
enabled            : true
ft_mobility_domain : ["set",[]]
ft_psk             : ["set",[]]
group_rekey        : ["set",[]]
if_name            : bhaul-ap-24
mac_list           : ["set",["60:b4:f7:02:31:81","60:b4:f7:f0:0f:46"]]
mac_list_type      : whitelist
min_hw_mode        : 11g
mode               : ap
parent             : ["set",[]]
rrm                : ["set",[]]
security           :
["map",[["encryption","WPA-PSK"],["key","E76090385FCE698EF16E83C0CC49B6A96468F7A952E575A56E44ABC4787F14E"],["mode","2"]]]
ssid               : we.backhaul
ssid_broadcast     : disabled
uapsd_enable       : ["set",[]]
vif_dbg_lvl        : ["set",[]]
vif_radio_idx      : 1
vlan_id            : ["set",[]]
wds                : ["set",[]]
---------------------------------------------------------------------------------------------------------------------------
---------------------
```

Wifi_VIF_State

Wifi_VIF_State is a table which represents the actual state on the device, and is therefore read by the Cloud. The table is pushed by WM2 and reflects the current system-wide wireless (logical) interface configuration.

Entries marked in gray are different to the *Wifi_VIF_Config* table, while the other rows are equal.

Name	Type	Description
vif_config	uuid	The associated Wifi_VIF_Config entry
if_name	string	Interface name
enabled	boolean	Defines if the specified interface is enabled or not
mode	enum (*ap*, *monitor*, *sta*)	Interface mode: ● **ap:** access point ● **monitor:** monitor (promiscuous) mode - *not used* ● **sta:** station (client)
state	string	TBD
channel	integer	Channel of the radio interface is attached (some platforms support multiple channel configurations).
mac	string	MAC address of the interface
vif_radio_idx	integer	VIF index of the radio interface
parent	string	Parent BSSID to which the STA needs to be connected (*Only for extenders*)
wds	boolean	Defines if the specified interface support supports WDS
ssid	string	Interface SSID
ssid_broadcast	enum (*enabled*, *disabled*, *disabled_null*)	Determines if the interface SSID is broadcast or not
security	key/value map	The map of strings specifying the security parameters ● Encryption ○ OPEN ○ WEP ○ WPA-PSK ○ WPA-EAP ● Key (password) ● Mode (PSK 1, 2 or mixed) TBD: EAP modes and radius config
bridge	string	Specifies the bridge interface to which the VIF is added

mac_list	string	ACL MAC list for layer 2 packet filtering
mac_list_type	enum (*whitelist*, *blacklist*, *none*)	How to filter MAC addresses: ● **whitelist:** only forward traffic from addresses in *mac_list*, block all other traffic ● **blacklist:** only block traffic from addresses in *mac_list*, forward all other traffic ● **none:** no filtering
associated_clients	uuid	MAC list of all associated clients
vlan_id	integer	Identification number for VLAN-tagged traffic
min_hw_mode	enum (*11a*, *11b*, *11g*, *11n*, *11ab*, *11ac*)	Minimal required IEEE 802.11 standard for the specified interface
uapsd_enable	boolean	Enable uAPSD power saving capability for the specified interface
group_rekey	integer	Time in seconds between the GTK rekeying. Valid for AP mode only. Empty or -1 means default, 0 means disabled, >0 is interval in seconds.
ap_bridge	boolean	AP isolation (infra-bss communication)
ap_vlan_sta_addr (available in *OpenSync* 2.0 and later)	string	MAC address of connected WDS station (Backhaul STA)
ft_psk	integer	“Fast transition” (802.11R) pre-shared key
ft_mobility_domain	integer	“Fast transition” (802.11R) mobility domain
btm	integer (0, 1)	● Indicates whether WNM BSS Transition support is enabled or not.
rrm	integer (0, 1)	Indicates if the RRM support is enabled. If RRM Neighbor Report isn’t available, 0 must be reported.
dynamic_beacon	boolean	Indicates if the dynamic_beacon configuration is enabled. It does not reflect if the feature is actively quiescing beacons at the moment or not.

Wifi_VIF_Neighbors

Wifi_VIF_Neighbors contains the neighboring AP information for steering and other features.

Name	Type	Description
bssid	string	BSSID of the neighbor
if_name	string	Associated interface name
channel	integer	Channel used by the neighbor
ht_mode	enum (*HT20*, *HT2040*, *HT40*, *HT40+*, *HT40-*, *HT80*, *HT160*, *HT80+80*)	Bandwidth used
priority	integer	*Not used (always set to 1)*

Wifi_Channels

The table enables configuration of maximum Tx power per radio from the Cloud using OVSDB.

Name	Type	Description
channel	integer	Channel number
radio_ifname	string	Radio interface name
tx_power	integer	Desired absolute max TX power in mdBm for the given channel
max_tx_power_allowed	integer	Absolute max TX power in mdBm that the device is capable of for the given channel
acs_enable	boolean	Enables or disables ACS
acs_is_enabled	boolean	Shows current ACS status

Wifi_Associated_Clients

Wifi_Associated_Clients lists all connected clients and their MAC addresses. *OpenSync* also updates the reference index in the proper *Wifi_VIF_State* row.

Name	Type	Description
mac	string	MAC address of the associated client
state	enum (*power save*, *idle*, *active*)	The state of client connection
capabilities	enum (*11b*, *11g*, *11a*, *11n*, *11ac*)	Client wireless capabilities, described by the conformity to one of the specified IEEE 802.11 standards
key_id	string	*(Not used)*
oftag	string	OpenFlow tag associated with the client
uapsd	integer	TBD
kick	string	TBD

The *Wifi_Associated_Clients* table needs to show all clients on any interface: backhaul, home, guest, etc.

Example:

```
 # ovsh s Wifi_Associated_Clients
-----------------------------------------------------------------------------------------------
_uuid        | 92bb~32be         | 1efb~ef4a         | 2057~2788         | b07e~9422         |
_version     | 3175~1516         | 44a0~556d         | 7024~ab29         | d802~5084         |
capabilities | ["set",[]]        | ["set",[]]        | ["set",[]]        | ["set",[]]        |
key_id       |                   |                   |                   |                   |
kick         | ["map",[]]        | ["map",[]]        | ["map",[]]        | ["map",[]]        |
mac          | 00:00:00:00:00:11 | 00:00:00:00:00:22 | 00:00:00:00:00:33 | 11:22:33:44:55:66 |
state        | active            | active            | active            | active            |
uapsd        | ["set",[]]        | ["set",[]]        | ["set",[]]        | ["set",[]]        |
-----------------------------------------------------------------------------------------------
```

Wifi_Credential_Config

Wifi_Credential_Config provides authentication details for the STA connection.

Name	Type	Description
ssid	string	SSID for which the credentials apply
security	key/value map	credentials encryption: encryption type (e.g., WPA-PSK) key: authentication key value
onboard_type	string	gre, no-gre

Example:

```
# ovsh s Wifi_Credential_Config
------------------------------------------------------
_uuid        | 03f2~d792                             |
_version     | f9b9~5a4a                             |
onboard_type | gre                                   |
security     | ["map",[["encryption","WPA-PSK"],     |
             : ["key","sample"]]]                    :
ssid         | sample                                |
------------------------------------------------------
```

Wifi_Inet_Config

The Wifi_Inet_Config is a configuration table and is therefore read by the NM. The table is pushed by the Cloud or CM. The table stores system-wide network configuration:

- Interface address assignment method (DHCP, static, PPPoE, etc.)
- Interface MTU
- Interface creation for certain interface types (most notably, for GRE tunnels)
- DNS services (dnsmasq)

Name	Type	Description
if_name	string	Interface name. Ethernet and VIF type interfaces must already exist. If they do not, an error should be reported. GRE, VLAN and possibly other types interfaces will be created with the name specified by this field.
if_type	enum (*bridge*, *eth*, *vif*, *gre*, *vlan*, *pppoe*, *softwds*, *tap*)	Interface type. Specifies the type of the selected interface. Some interface types are fixed (exist at boot), while the others must be created on the fly. Interface creation typically occurs when a row with the appropriate if_type is inserted into OVS. ● **bridge** - this row specifies the bridge interface. The interface is created on the fly. ● **eth** - this row contains the specific configuration for an Ethernet interface. The interface must already exist. ● **vif** - this row specifies a wifi interface. It is assumed that the interface is created by the WM. ● **gre** - GRE tunneling interface. The interface is created on-the-fly when this row is inserted.The gre_ifname, gre_remote_inet_addr, and gre_local_inet_addr fields must not be empty. ● **vlan** - this row specifies a VLAN interface. The interface is created on the fly when a row of this type is inserted. The parent_ifname field must not be empty. ● **pppoe** - PPP over Ethernet interface *(Not supported in OpenSync 2.0)* ● **softwds** - SoftWDS interface; deprecated
if_uuid	uuid	
enabled	boolean	The desired interface state. “True” if the interface should be in the UP state, otherwise the interface is in the DOWN state (the equivalent of ifconfig up/down).
network	boolean	In certain conditions, the interfaces must be in the UP state, but should not have any configuration applied. This field is “True” if the network configuration should be applied to the interface, and “False” if it should not.
NAT	boolean	“True” if NAT/Masquerading should be effective for the

		outgoing traffic on this interface.
ip_assign_scheme	enum (*none*, *dhcp*, *static*)	- **none** - the interface has no address configuration. This is the equivalent of setting the *network* field to false - **dhcp** - dynamic address configuration using the DHCP protocol. The dhcp or some other variant of the DHCP client should be started on this interface. - **static** - The interface has static IP configuration. The address present in the "inet_addr" and "netmask" fields should be used for the IP address configuration.
inet_addr	string	The IP address when "ip_assign_scheme" == "static"
netmask	string	Interface netmask when "ip_assign_scheme" == "static"
gateway	string	The default gateway when "ip_assign_scheme" == "static"
broadcast	string	The broadcast address when "ip_assign_scheme" == "static"
gre_remote_inet_addr	string	*Only applicable when "if_type" == "gre"*. Specifies the remote tunnel IP address.
gre_local_inet_addr	string	*Only applicable when "if_type" == "gre"*. Specifies the local tunnel IP address.
gre_remote_mac_addr	string	*Only applicable when "if_type" == "gre"*. Specifies the remote tunnel MAC address.
gre_ifname	string	*Only applicable when "if_type" == "gre"*. Specifies the parent interface for the GRE tunnel.
mtu	integer	Desired MTU
dns	key/value map	*Only applicable when "ip_assign_scheme" == "static".*DNS server list.
dhcpd	key/value map	If populated, the DHCP server should be enabled on this interface. This field specifies a list of DHCP options that should be used to configure the DHCP server. See DHCP Server Configuration.
upnp_mode	enum (*disabled*, *internal*, *external*)	If populatedand not set to "disabled", UPnP is enabled on this interface. For more information, see the UPnP Configuration - **disabled** - UPnP is disabled, this is the default. - **internal** - UPnP is enabled. Only one interface with this option value must exist. This specifies that a UPnP service must be started on this interface and that this is the interface facing the internal network. - **external** - UPnP is enabled. Only one interface with this option value must exist. This specifies that a UPnP service must be started on this interface and that this is the interface facing the "external" network (internet).

dhcp_sniff	boolean	TRUE if DHCP sniffing is enabled on this interface
vlan_id	integer	*Only applicable when "if_type" == "vlan"*. This is the VLAN ID.
parent_ifname	string	*Only applicable when "if_type" == "vlan"*. Parent interface name.
ppp_options	key/value map	*Only applicable when "if_type" == "pppoe"*. PPPoE options. Not supported in *OpenSync* 2.0.
softwds_mac_addr	string	*Only applicable when "if_type" == "softwds"*. Deprecated.
softwds_wrap	boolean	*Only applicable when "if_type" == "softwds"*. Deprecated.
igmp	boolean	*Enables IGMP multicast snooping on the interface. Applicable to OVS bridges only (mcast_snooping_enable).*
igmp_age	integer	*IGMP multicast snooping aging time in seconds (mcast-snooping-aging-time)*
igmp_tsize	integer	*IGMP multicast snooping table size (mcast-snooping-table-size)*
igmp_proxy	enum (*disabled,* IGMPv1, IGMPv2, IGMPv3)	Enables or disables ipv4 multicast proxy i.e IGMP on this interface. When a version is specified the proxy is enabled with that version.
mld_proxy	enum (*disabled,* MLDv1, MLDv2)	Enables or disables ipv6 multicast proxy i.e MLD on this interface. When a version is specified the proxy is enabled with that version.
role	string	

Note: IP forwarding must always be enabled on the device, regardless of the Wifi_Inet_Config table settings.

Example:

```
NAT                   : false
_uuid                 : 0ebc~239c
_version              : e6f5~ca31
broadcast             : 192.168.40.255
dhcp_sniff            : ["set",[]]
dhcpd                 :
["map",[["dhcp_option","3,192.168.40.1;6,192.168.40.1"],["force","false"],["lease_time","12h"],["start","1
92.168.40.50"],["stop","192.168.40.254"]]]
dns                   : ["map",[]]
enabled               : true
gateway               : ["set",[]]
gre_ifname            : ["set",[]]
gre_local_inet_addr   : ["set",[]]
gre_remote_inet_addr  : ["set",[]]
gre_remote_mac_addr   : ["set",[]]
if_name               : br-home
if_type               : bridge
if_uuid               :
igmp                  : true
igmp_age              : ["set",[]]
igmp_tsize            : ["set",[]]
igmp_proxy            : IGMPv2
inet_addr             : 192.168.40.1
mld_proxy             : MLDv2
ip_assign_scheme      : static
mtu                   : ["set",[]]
netmask               : 255.255.255.0
network               : true
parent_ifname         : ["set",[]]
ppp_options           : ["map",[]]
softwds_mac_addr      : ["set",[]]
softwds_wrap          : ["set",[]]
upnp_mode             : internal
vlan_id               : ["set",[]]
```

Wifi_Inet_State

This table reflects the current system status and is updated by the NM. The Cloud/CM reads this table to determine the current network status. If some data is not available on the platform, this data can be taken from the homonymous values that are available in the *Wifi_Inet_Config* table ***after*** they are applied to the system.

Name	Type	Description
if_name	string	Interface name
inet_config	Uuid reference	Reference to the corresponding row in Wifi_Inet_Config
if_type	enum (*bridge*, *eth*, *vif*, *gre*, *vlan*, *pppoe*, *softwds*, *tap*)	Interface type. Can be derived from the homonymous field in Wifi_Inet_Config.
if_uuid	uuid	
enabled	boolean	“True” if the interface is administratively enabled. Can be derived from the homonymous field in Wifi_Inet_config.
network	boolean	“True” if the network configuration is applied. Can be derived from the homonymous field in Wifi_Inet_Config.
NAT	boolean	“True” if the Network Address Translation is enabled for the outgoing traffic on this interface. Can be derived from the homonymous field in Wifi_Inet_Config.
ip_assign_scheme	string	The IP address assignment scheme. Can be derived from the homonymous field in Wifi_Inet_Config.
inet_addr	string	Currently configured IP address. If ip_assign_scheme == “dhcp”, this field must contain the assigned IP address; if ip_assign_scheme == “static”, the value can be derived from the homonymous field in Wifi_Inet_config.
netmask	string	Currently configured IP netmask. If ip_assign_scheme == “dhcp”, this field must contain the assigned IP netmask; if ip_assign_scheme == “static”, the value can be derived from the homonymous field in Wifi_Inet_config.
gateway	string	Currently configured default gateway associated with this interface. The value can be derived from the homonymous field in Wifi_Inet_config.

broadcast	string	Currently configured IP broadcast address. If ip_assign_scheme == “dhcp”, this field should contain the assigned IP broadcast address; if ip_assign_scheme == “static”, the value can be derived from the homonymous field in Wifi_Inet_config.
gre_remote_inet_addr	string	*Only applicable if if_type* == *“gre”.* Currently configured GRE remote tunnel address. Can be derived from the homonymous field in Wifi_Inet_config.
gre_local_inet_addr	string	*Only applicable if if_type* == *“gre”.* Currently configured GRE local tunnel address. Can be derived from the homonymous field in Wifi_Inet_config.
gre_ifname	string	*Only applicable if if_type* == *“gre”.* GRE parent interface. Can be derived from the homonymous field in Wifi_Inet_config.
mtu	integer	Currently configured MTU. Can be derived from the homonymous field in Wifi_Inet_config.
dns	key/value map	Currently applied DNS settings related to this interface; can be derived from the homonymous field in Wifi_Inet_config
dhcpd	key/value map	Currently applied DHCP server settings pertaining to this interface. Can be derived from the homonymous field in Wifi_Inet_config.
hwaddr	string	Interface hardware address
dhcpc	key/value map	*Only applicable if ip_assign_scheme* == *“dhcp”.* This field contains the list of received DHCP client options.
upnp_mode	string	Currently configured UPnP mode of operation. Can be derived from the homonymous field in Wifi_Inet_config.
vlan_id	integer	*Only applicable if if_type* == *“vlan”.* The VLAN tag associated with the current interface. can be derived from the homonymous field in Wifi_Inet_config.
parent_ifname	string	*Only applicable if if_type* == *“vlan”*, The parent interface. Can be derived from the homonymous field in Wifi_Inet_config.
softwds_mac_addr	string	*Only applicable if if_type* == *“softwds”*. Not supported in *OpenSync* 2.0.
softwds_wrap	boolean	*Only applicable if if_type* == *“softwds”*. Not supported in *OpenSync* 2.0.

Wifi_Master_State

The *Wifi_Master_State* table enables synchronization between WM, NM, and CM. CM uses this table for determining the preferred backhaul connection (read-only). WM and NM update the values in this table. WM updates interfaces of type vif, while NM updates the rest.

Name	Type	Description
dhcpc	map	DHCP client parameters
if_name	string	Interface name
if_type	enum (*bridge*, *eth*, *vif*, *gre*, *vlan*, *pppoe*, *softwds*)	Interface type
if_uuid	uuid	Pointer to Wifi_Inet_Config record
inet_addr	string	IP address
netmask	string	Netmask
network_state	enum (up, down)	Network state
onboard_type	string	Onboard type
port_state	enum (active, inactive)	Physical port state
uplink_priority	integer	Uplink priority *(deprecated)*

Example:

```
# ovsh s Wifi_Master_State
-----------------------------------------------------------------------------------
_uuid           | db99~6d3b     | 2222~fded    | 4057~936c  | fa3c~0c88           |
_version        | ccd5~6378     | 659d~b136    | 3334~d7d4  | f27c~cdcb           |
dhcpc           | ["map",[]]    | ["map",[]]   | ["map",[]] | ["map",[["dns",     |
                :               :              :            : "192.168.40.1"],    :
                :               :              :            : ["domain","lan"],   :
                :               :              :            : ["gateway",         :
                :               :              :            : "192.168.40.1"],    :
                :               :              :            : ["lease","43200"]]] :
if_name         | bhaul-ap-24   | bhaul-sta-24 | br-home    | br-wan              |
if_type         | vif           | vif          | bridge     | bridge              |
if_uuid         | 60f2~d995     | 6644~6bad    | aed4~fc05  | 2a77~0a35           |
inet_addr       | 169.254.6.1   | 0.0.0.0      | 0.0.0.0    | 192.168.40.88       |
netmask         | 255.255.255.0 | 0.0.0.0      | 0.0.0.0    | 255.255.255.0       |
network_state   | up            | down         | up         | up                  |
onboard_type    | ["set",[]]    | ["set",[]]   | ["set",[]] | ["set",[]]          |
port_state      | active        | ["set",[]]   | active     | active              |
uplink_priority | ["set",[]]    | ["set",[]]   | ["set",[]] | ["set",[]]          |
-----------------------------------------------------------------------------------
```

IP_Port_Forward

The *IP_Port_Forward* table contains the user-defined port forwarding rules.
This feature is only used when a device is configured to run in bridge mode.

Name	Type	Description
protocol	enum (*tcp*, *udp*)	Protocol type
src_ifname	string	Source interface for the rule
src_port	integer	Source port
dst_port	integer	Destination port
dst_ipaddr	string	Destination IP address

OVS_MAC_Learning

The *OVS_MAC_Learning* table reports the current bridge MAC learning table to the Cloud. In case of OVS bridges, the contents of this table are derived from various OVS tables.

Name	Type	Description
brname	string	Bridge name
ifname	string	Interface from which the learned MAC address originates
hwaddr	string	Learned MAC address
vlan	integer	VLAN ID

Example:

```
# ovsh s OVS_MAC_Learning
------------------------------
_uuid    | 631e~2674         |
_version | a321~1956         |
brname   | br-home           |
hwaddr   | 00:11:22:33:44:55 |
ifname   | eth0              |
vlan     | 0                 |
------------------------------
```

DHCP_leased_IP

The *DHCP_leased_IP* table presents all device-specific information retrieved through DHCP fingerprinting, such as IP (only IPv4 in *OpenSync* 2.0) or MAC address.

Name	Type	Description
hwaddr	string	MAC address of the associated client
inet_addr	string	IP address
hostname	string	Client hostname
fingerprint	string	DHCP options
lease_time	string	Lease time of DHCP

Example:

```
# ovsh s DHCP_leased_IP
------------------------------------------------------------------------------
_uuid       | 728f~a081         | 480c~b840         | 404e~ad5f              |
_version    | 3b7a~6e5f         | 94be~33ca         | 1f07~9a49              |
fingerprint |                   |                   | 1,121,33,3,6,12,15,28, |
            :                   :                   : 42,51,54,58,59,119     :
hostname    | P1K634058600_Pod  | P1K634073700_Pod  | rpi-4                  |
hwaddr      | 00:00:00:00:00:11 | 00:00:00:00:00:22 | 11:22:33:44:55:66      |
inet_addr   | 192.168.2.17      | 192.168.2.19      | 192.168.2.27           |
lease_time  | 259200            | 259200            | 259200                 |
------------------------------------------------------------------------------
```

DHCP_reserved_IP

The *DHCP_reserved_IP* table assigns a specific IP address to a known device MAC address.

Name	Type	Description
hostname	string	Client hostname
hw_addr	string	MAC address of associated client
ip_addr	string	IP address

Example:

```
# ovsh s DHCP_reserved_IP
------------------------------
_uuid    | ff76~1ece         |
_version | a18f~b288         |
hostname | ["set",[]]        |
hw_addr  | 11:22:33:44:55:66 |
ip_addr  | 192.168.40.77     |
------------------------------
```

DHCP Server Configuration

The DHCP server configuration is represented as a map of strings (array of keys, value pairs) in the *dhcpd* column of the *Wifi_Inet_Config* table. Since a map is a very flexible structure, the format used by the dhcpd column requires some explanation.

The format that is understood by the NM is as follows:

Key	Value / Description
dhcp_option	“option_id,value;option_id,value” Specifies a list of DHCP options to be provided by the DHCP server to the DHCP client. Example: “3,192.168.40.1;6,192.168.40.1” Note: DHCP option 3 = router; dhcp option 6 = DNS server
force	“true” or “false” *Not used in OpenSync 2.0*
lease_time	Lease time in the N[mhs] format.his format must be understood by the dnsmasq. Example: “12h”
start	IP pool start address
stop	IP pool end address

UPnP Configuration

The UPnP configuration requires exactly two interfaces to work properly: one WAN and one LAN interface. These interfaces are referred to as “external” and “internal” respectively.

These two interfaces create dynamic port forwards from the WAN interface to the IP address on the LAN interface.

DHCP Sniffing

The DHCP sniffing feature allows NM to capture DHCP packets on the configured interface, and extract the assigned hostname and DHCP fingerprint. This feature is mainly used when the device is configured to run in bridge mode.

Wifi_Stats_Config

The *Wifi_Stats_Config* table is used to configure the stats collecting on the device.

Name	Type	Description
stats_type	enum (*neighbor*, *survey*, *client*, *capacity*, *radio*, *essid*, *quality*, *device*, *rssi*, *steering*)	The Cloud selects the stats type when it wants to fetch certain statistics from the device. NOTE: Only *neighbor*, *survey*, *client*, *device*, *rssi*, and *steering* are used in *OpenSync* 2.0.
report_type	enum (*raw*, *average*, *histogram*, *percentile*, *diff*)	Cloud can specify different reporting formats: ● **raw:** bins of samples collected during the *sampling_interval* time ● **average:** value of samples or specific parameters collected during the *reporting_interval* time ● **histogram:** distributed sample values through the *reproting_interval* ● **percentile:** of sampled values inside the *reporting_interval* ● **diff:** special reporting when only diff values between samples are sent to the Cloud
radio_type	enum (*2.4G*, *5G*, *5GL*, *5GU*)	The selection of *radio_type* configuration depends on the device wireless capabilities listed inside the Wifi_VIF/Radio_State tables.
survey_type	enum (*on-chan*, *off-chan*, *full*)	For surveying (scan and utilization), the Cloud specifies the following types: ● **on-chan:** measurements on the home channel can be done frequently and without user interruption. They are done periodically. ● **off-chan:** measurements on the foreign channel must be done with care, since they involve switching to off-channel. The channel measurements listed in *channel_list* are periodically chosen using the round robin fashion (“1,6,11” -> t0=1, t1=6, t2=11, t3=1) to minimize user impact. ● **full:** the measurements are done on all channels at once and as specified in the *channel_list* (“1,6,11” -> t0=1,6,11 t1=1,6,11) which is very intrusive.
reporting_interval	integer	Interval specifying the time after which the report is sent

reporting_count	integer	Max number of consecutive reports. 0 means periodic.
sampling_interval	integer	Interval specifying the time between sample collections
survey_interval_ms	integer	Scan DWEL time used for surveying
channel_list	set of integers	Channel list that needs to be surveyed
threshold	key/value map	Threshold specifies the intrusiveness behavior of the scan, while it can also contain the diff thresholds: ● **max_delay:** max delay of measurement when the threshold is reached - each sampling interval the threshold delta is used ● **util:** utilization percentage that still allows measurements

Example:

```
# ovsh s Wifi_Stats_Config
-----------------------------------------------------------------------------------------------------...
_uuid              | b5db~37e6             | de8e~6de0  | 006d~c137         | 88e8~9d00  | 0d50~4d65  |
_version           | 0572~d01b             | 37ff~8d4e  | 4a66~909b         | 735f~01af  | 82a9~ea8f  |
channel_list       | ["set",[40,153]]      | ["set",[]] | ["set",[1,6,11]]  | ["set",[]] | ["set",[]] |
radio_type         | 5G                    | 2.4G       | 2.4G              | 5G         | 2.4G       |
reporting_count    | 0                     | 0          | 0                 | 0          | 0          |
reporting_interval | 0                     | 60         | 120               | 60         | 900        |
sampling_interval  | 0                     | 0          | 0                 | 10         | 0          |
stats_type         | survey                | neighbor   | neighbor          | client     | device     |
survey_interval_ms | 10                    | 0          | 0                 | ["set",[]] | ["set",[]] |
survey_type        | off-chan              | on-chan    | off-chan          | ["set",[]] | ["set",[]] |
threshold          | ["map",[["max_delay", | ["map",[]] | ["map",[]]        | ["map",[]] | ["map",[]] |
                   : 600],["util",10]]]    :            :                   :            :            :
-----------------------------------------------------------------------------------------------------...
```

```
  ...-----------------------------------------------------------------------------------------------
    | 27f5~40e1             | 3c28~cbb1  | 52e5~0480        | d943~6425  | ebe4~aa07  | 9205~441b  |
    | a271~1855             | a4aa~db93  | ad9a~918c        | cf24~f006  | da27~99f0  | ff08~5577  |
    | ["set",[1,6,11]]      | ["set",[]] | ["set",[40,153]] | ["set",[]] | ["set",[]] | ["set",[]] |
    | 2.4G                  | 5G         | 5G               | 2.4G       | 5G         | 2.4G       |
    | 0                     | 0          | 0                | 0          | 0          | 0          |
    | 120                   | 60         | 0                | 60         | 60         | 60         |
    | 10                    | 10         | 0                | 10         | 0          | 10         |
    | survey                | survey     | neighbor         | client     | neighbor   | survey     |
    | 50                    | 0          | 10               | ["set",[]] | 0          | 0          |
    | off-chan              | on-chan    | off-chan         | ["set",[]] | on-chan    | on-chan    |
    | ["map",[["max_delay", | ["map",[]] | ["map",[]]       | ["map",[]] | ["map",[]] | ["map",[]] |
    : 600],["util",10]]]    :            :                  :            :            :            :
  ...-----------------------------------------------------------------------------------------------
```

Band_Steering_Config

The table enables general configuration of band steering and client steering parameters.

Name	Type	Description
chan_util_avg_count	integer	Channel utilization average count
chan_util_check_sec	integer	Channel utilization sampling period for pre-association band steering (in seconds)
chan_util_hwm	integer	Channel utilization high water mark
chan_util_lwm	integer	Channel utilization low watermark
dbg_2g_raw_chan_util	boolean	Enables channel utilization logging on the 2.4G interface
dbg_2g_raw_rssi	boolean	Enables raw RSSI logging on the 2.4G interface
dbg_5g_raw_chan_util	boolean	Enables channel utilization logging on the 2.4G interface
dbg_5g_raw_rssi	boolean	Enables raw RSSI logging on the 5G interface
debug_level	integer	Sets overall logging severity level
def_rssi_inact_xing	integer	Inactive RSSI threshold
def_rssi_low_xing	integer	Inactive RSSI low threshold
def_rssi_xing	integer	Inactive RSSI high threshold
gw_only	boolean	Indicates if this is the only *OpenSync* device (gateway) in this location
if_name_2g	string	2.4G interface name
if_name_5g	string	5G interface name
inact_check_sec	integer	Client inactivity check period
inact_tmout_sec_normal	integer	Client inactivity timeout in normal mode
inact_tmout_sec_overload	integer	Client inactivity timeout in overload mode
kick_debounce_period	integer	Time for which client kick is blocked due to failed attempts
kick_debounce_thresh	integer	Number of failed attempts before client kicks are disabled for the time period defined by *kick_debounce_period*
stats_report_interval	integer	Time period in seconds which defines band steering related stats reporting to the Cloud

success_threshold_secs	integer	Time period in seconds which defines the time of the successful kick

Example:

```
# ovsh s Band_Steering_Config
----------------------------------------
_uuid                    | 48f4~5ae6   |
_version                 | 9153~edce   |
chan_util_avg_count      | 0           |
chan_util_check_sec      | 0           |
chan_util_hwm            | 80          |
chan_util_lwm            | 50          |
dbg_2g_raw_chan_util     | false       |
dbg_2g_raw_rssi          | false       |
dbg_5g_raw_chan_util     | false       |
dbg_5g_raw_rssi          | false       |
debug_level              | 0           |
def_rssi_inact_xing      | 0           |
def_rssi_low_xing        | 0           |
def_rssi_xing            | 0           |
gw_only                  | false       |
if_name_2g               | home-ap-24  |
if_name_5g               | home-ap-u50 |
inact_check_sec          | 10          |
inact_tmout_sec_normal   | 60          |
inact_tmout_sec_overload | 30          |
kick_debounce_period     | 0           |
kick_debounce_thresh     | 0           |
stats_report_interval    | 1           |
Success_threshold_secs   | 15          |
----------------------------------------
```

Band_Steering_Clients

This table contains per client band steering configuration and states.

Name	Type	Description
backoff_exp_base	integer	Exponential duration after pre-assoc steering failures
backoff_secs	integer	Backoff timer before another steering event is processed
cs_mode	enum (*off*, *home*, *away*)	Client steering mode: ● off ● home ● away
cs_params	string	Client steering parameters
cs_state	enum (*none*, *steering*, *expired*, *failed*, *xing_low*, *xing_high*, *xing_disabled*)	Client steering state: ● none ● steering ● expired ● failed ● xing_low ● xing_high ● xing_disabled
force_kick	enum (*none*, *speculative*, *directed*, *ghost_device*)	Trigger force client kick
hwm	integer	High watermark
kick_debounce_period	integer	When a kick has failed, another kick will not be attempted for this amount of seconds (and once client’s RSSI crosses the HWM/LWM threshold again).
kick_reason	integer	Client kick reason
kick_type	enum (*none*, *deauth*, *disassoc*, *bss_tm_req*, *rrm_br_req*, *btm_deauth*, *btm_disassoc*)	Different kick types: **deauth** - sending deauthentication frame to the client **disassoc** - sending disassociation frame to the client **bss_tm_req** - sending BSS Transition Management(802.11v) frame to the client **btm_deauth** - If client is 11v capable, send a

		802.11v frame, else send deauthentication frame. **btm_disassoc** - If client is 11v capable, send a 802.11v frame, else send disassociation frame.
kick_upon_idle	boolean	If kicking is enabled, attempt to “kick” the client only if it is not ‘busy’ (doing data transfer). If the client is busy, wait until it becomes idle.
lwm	integer	Low watermark - if kicking is enabled, when a client's active RSSI goes below this value, attempt to "kick" it for being a “sticky” client.
mac	string	Client MAC address
max_rejects	integer	If we get this number of rejects within the time period (*rejects_tmout_secs*), we consider steering as failed.
pre_assoc_auth_block	boolean	Block responses to authorization requests from a client during pre-association band steering.
pref_5g	enum (*hwm*, *never*, *always*)	Prefer connecting clients on 5 GHz radio based on these settings: ● **hwm** - depending on the signal strength on 5 GHz, the radio blocks 2.4 GHz ● **never** ● **always**
reject_detection	string	Why was the client rejected? Possible values: ● none ● probe_all ● probe_null ● probe_direcet ● auth_blocked
rejects_tmout_secs	integer	Number of rejects (*max_rejects*) within the time period (*rejects_tmout_secs*) to be considered a steering failure, and to not go into backoff for 300 seconds (*backoff_secs*)
rrm_bcm_rpt_params	map	802.11k beacon report request parameters
sc_kick_debounce_period	integer	Steering from Cloud(sc) client kick debounce period
sc_kick_reason	integer	Steering from Cloud(sc) client kick reason
sc_kick_type	enum (*none*, *deauth*, *disassoc*, *bss_tm_req*, *rrm_br_req*,	Possible kick types: ● none ● disassoc ● deauth ● 802.11v BSS Transition Request ● 80211k Beacon Report Request

	btm_deauth, *btm_disassoc*, *rrm_deauth*, *rrm_disassoc*)	● 802.11v request or deauth ● 80211v request or disassoc
sc_btm_params	map	802.11v BSS Transition request parameters
stats_2g	map	*Not used*
stats_5g	map	*Not used*
steer_during_backoff	bool	Allow/disallow steering during backoff
steering_btm_params	map	BSS Transition Management (802.11v) configuration
steering_fail_cnt	integer	Client steering failed attempt count
steering_kick_cnt	integer	Client steering kick count
steering_success_cnt	integer	Client steering successful attempt count
sticky_kick_cnt	integer	Sticky client kick count
sticky_kick_debounce_period	integer	Sticky client kick debounce period
sticky_kick_reason	integer	Sticky client kick reason
sticky_kick_type	enum (*none*, *deauth*, *disassoc*, *bss_tm_req*, *rrm_br_req*, *btm_deauth*, *btm_disassoc*)	Possible kick types: ● none ● disassoc ● deauth ● 802.11v BSS Transition Request ● 80211k Beacon Report Request ● 802.11v request or deauth ● 80211v request or disassoc

Example:

```
# ovsh s Band_Steering_Clients
---------------------------------------------------------------------------------------------------
_uuid                       | de8a~5d9e                                                           |
_version                    | b09a~1360                                                           |
backoff_exp_base            | 2                                                                   |
backoff_secs                | 120                                                                 |
cs_mode                     | ["set",[]]                                                          |
cs_params                   | ["map",[]]                                                          |
cs_state                    | ["set",[]]                                                          |
force_kick                  | ["set",[]]                                                          |
hwm                         | 35                                                                  |
kick_debounce_period        | 60                                                                  |
kick_reason                 | 1                                                                   |
kick_type                   | btm_deauth                                                          |
kick_upon_idle              | true                                                                |
lwm                         | 20                                                                  |
mac                         | 30:07:4d:33:60:c8                                                   |
max_rejects                 | 7                                                                   |
pre_assoc_auth_block        | true                                                                |
pref_5g                     | always                                                              |
reject_detection            | probe_all                                                           |
rejects_tmout_secs          | 120                                                                 |
rrm_bcn_rpt_params          | ["map",[]]                                                          |
sc_btm_params               | ["map",[]]                                                          |
Sc_kick_debounce_period     | 0                                                                   |
sc_kick_reason              | 0                                                                   |
sc_kick_type                | ["set",[]]                                                          |
stats_2g                    | ["map",[]]                                                          |
stats_5g                    | ["map",[]]                                                          |
steer_during_backoff        | false                                                               |
steering_btm_params         | ["map",[["abridged","1"],["bss_term","0"],["btm_max_retries","3"],  |
                            : ["btm_retry_interval","10"],["disassoc_imminent","1"],["pref","1"], :
                            : ["valid_interval","255"]]]                                          :
steering_fail_cnt           | 0                                                                   |
steering_kick_cnt           | 0                                                                   |
steering_success_cnt        | 0                                                                   |
sticky_btm_params           | ["map",[["abridged","1"],["bss_term","0"],["btm_max_retries","3"],  |
                            : ["btm_retry_interval","10"],["disassoc_imminent","1"],["inc_neigh", :
                            : "true"],["pref","0"],["valid_interval","255"]]]                     :
sticky_kick_cnt             | 0                                                                   |
sticky_kick_debounce_period | 60                                                                  |
sticky_kick_reason          | 1                                                                   |
sticky_kick_type            | btm_deauth                                                          |
---------------------------------------------------------------------------------------------------
```

AW_LM_Config

This table enables the logpull functionality.

Name	Type	Description
upload_location	string	URL for uploading the collected files
upload_token	string	Security token for uploading the collected files
name	string	*(Deprecated)*
periodicity	string	*(Deprecated)*

AW_LM_State

This table is DEPRECATED.

AW_Debug

The table is currently used for setting the log levels to registered modules. Note that this table is used only as a local interface.

Name	Type	Description
name	string	Module name used for registration to the *OpenSync* log library
log_severity	string	Dynamic log severity, which can be one of these values: ● **EMERG** - System is unusable ● **ALERT** - Action must be taken immediately ● **CRIT** - Critical conditions ● **ERR** - Error conditions ● **WARNING** - Warning conditions ● **NOTICE** - Normal but significant condition ● **INFO** - Informational message ● **DEBUG** - Debug message ● **TRACE** - Trace messages

Openflow_Config

This table contains all the flows that need to be applied to the system. Flows can be set constant or defined as a template by using variables described in Openflow_Tag or Openflow_Tag_Group table.

Name	Type	Description
action	string	Flow action (drop, normal, resubmit, etc.)
bridge	string	Bridge name
priority	integer	Flow priority
rule	string	Flow rule or flow rule template
table	integer	Flow table
token	string	Name of the flow

Openflow_State

This table reflects the status of all currently applied flows.

Name	Type	Description
bridge	string	Bridge name
openflow_config	string	NOT USED
success	boolean	Marks successfully applied flow
token	string	Name of the flow

Openflow_Tag

This table enables expansion of the packet flow rules. Defines variables used to construct a flow in combination with a template rule from Openflow_Config table

Name	Type	Description
cloud_value	string	Cloud-only tag value
device_value	string	Device-only tag value

name	string	Tag value name

Openflow_Local_Tag

If Captive Portal is used at a location, this table stores the white-listed domains for resolving the IP addresses. These are used subsequently to create openflow rules to allow traffic to the stored white-listed domains.

Name	Type	Description
values	string	White-listed domain address
name	string	Descriptive name of the white-listed domain

Openflow_Tag_Group

This table combines a list of tags defined in the Openflow_Tag table to create a new group tag. Group tag can be used to construct a flow in combination with a template rule from the Openflow_Config table.

Name	Type	Description
tags	string	List of tags from Openflow_Tag table
name	string	Tag group name

Client_Nickname_Config

Used for nickname synchronization between the device and the Cloud.

Name	Type	Description
mac	string	Client MAC address
nickname	string	Client nickname

Example:

```
# ovsh s Client_Nickname_Config
------------------------------
_uuid    | c431~a7a0         |
_version | 4913~5531         |
mac      | 00:11:22:33:44:55 |
nickname | device_test_name  |
------------------------------
```

Client_Freeze_Config

The device freeze feature restricts client access to the internet or local network. Users set the rules over mobile app/cloud or local GUI if present. The user can perform instant freeze or may arrange multiple scheduled freezes (Bedtime, School nights, custom).

Name	Type	Description
blocked	bool	Block or unblock the client
mac	string	Client MAC address
source	enum (*init*, *cloud*, *gw*)	Mark the source of truth: ● **init** - unknown owner after reboot ● **cloud** - cloud managed client ● **gw** - gateway or device is the owner
type	enum (*schedule*, *always*)	The cloud controller only sets "*always*" value here. However, some devices might set the values to "*schedule*" if the device supports a schedule mode that is configured outside the scope of the cloud controller.

Example:

```
# ovsh s Client_Freeze_Config
------------------------------
_uuid    | 5382~faee         |
_version | f039~bd24         |
blocked  | false             |
mac      | 69:78:65:66:76:69 |
source   | gw                |
type     | schedule          |
------------------------------
```

Flow_Service_Manager_Config

This table provisions the plugins to the FSM service. The table informs the FSM about the plugin name, and provides additional information: location of the shared library instantiating the plugin, and plugin type (parser dedicated to a specific traffic type, DPI plugin, or web categorization backend plugin).

Name	Type	Description
handler	string	Unique FSM plugin identifier
type	enum (*parser*, *web_cat_provider*, *dpi*)	Plugin type: - **parser**: dedicated to a specific traffic type - **dpi**: DPI plugin - **web_cat_provider**: dedicated to categorize the web access type
if_name	string	Name of the tap interface, from which the FSM gets the packets and forwards them to the plugin packet handler
pkt_capt_filter	string	An optional BPF filter applied to the incoming traffic. Originally used to filter out the multicast/broadcast traffic; now optional.
plugin	string	The DSO file instantiating the plugin
other_config	key/value map	A (key, value) map allowing the opaque parameters to pass to the plugin

Example:

```
# ovsh Flow_Service_Manager_Config -w handler==http
-------------------------------------------------------------------------------------------------
_uuid           | 63d6~3fc4                                                                     |
_version        | d3a1~0f61                                                                     |
handler         | http                                                                          |
if_name         | br-home.http                                                                  |
other_config    | ["map",[["mqtt_v","HTTP/Requests/dog1/4C7770142A/59efd33d2c93832025330a3e"]]] |
pkt_capt_filter | tcp dst port 80                                                               |
plugin          |                                                                               |
type            | ["set",[]]                                                                    |
-------------------------------------------------------------------------------------------------
```

FSM_Policy

The FSM_Policy table instructs the requesting FSM plugins which action to take in case of a specific event. Its main usage y targets advanced device typing, IP threat detection, and content filtering.

The organized policy names are contained in the rules columns. A rule is a set made of filters, an action, and a report policy. Should an event pass the filters, the plugin applies the rules's action, and reports the event according to the report policy.

Name	Type	Description
name	string	Policy rule name
idx	integer	Within a policy name space; the sorted index of the current rule
mac_op	enum (*in*, *out*)	{event mac} (mac_op) [set of macs] filter. IE is mac X in/out the given set of macs
macs	set of strings	The set of macs to compute an event against
fqdn_op	enum (*in*, *out*, *sfr_in*, *sfr_out*, *sfl_in*, *sfl_out*)	{event fqdn} (fqdn_op) [set of fqdns] filter ● sfr denotes "start from right" ● sfl denotes "start from left" These prefixes allow wildcard matching.
fqdns		The set of FQDNs to compute an event against
fqdncat_op	enum (*in*, *out*)	{event fqdn category} (fqdncat_op) [set of fqdn categories] filter. The FQDN category is determined by the web_cat_provider plugin.
fqdncats	set of integers	The set of FQDN categories to compute an event against
ipaddr_op	enum (*in*, *out*)	{ip addr} (ipaddr_op) [set of ip addr] filter
ipaddrs	set of strings	The set of ip addresses to compute an event against
risk_level	integer	The web remote resource risk level to compare against
risk_op	enum (*eq*, *neq*,	The risk operation to use in the risk assessment

	gt, *lt*, *gte*, *lte*)	
log	enum (*none*, *all*, *blocked*)	**none** - do not log the event **all** - log the event regardless of the action **blocked** - log only blocked events (web categorization)
action	enum (*allow*, *drop*)	Applies to web categorization events
next	key:integer	Enables jumping to the specific rule of the requested table, chaining up the event processing across multiple tables.
policy	string	Policy rule name
redirect[0..2]	string	Redirects a FQDN resolution to the provided Ipv4/IPv6/cname
other_config	key/value map	A (key, value) map allowing the opaque parameters to pass the rule.

Example:

```
# ovsh  s FSM_Policy -w idx==2
-------------------------------
_uuid        | 6333~af57      |
_version     | b5be~006c      |
action       | ["set",[]]     |
fqdn_op      | ["set",[]]     |
fqdncat_op   | ["set",[]]     |
fqdncats     | ["set",[]]     |
fqdns        | ["set",[]]     |
idx          | 2              |
ipaddr_op    | ["set",[]]     |
ipaddrs      | ["set",[]]     |
log          | all            |
mac_op       | out            |
macs         | $[dns-exclude] |
name         | dt_dns         |
next         | ["map",[]]     |
other_config | ["map",[]]     |
policy       | my_policy      |
redirect     | ["set",[]]     |
risk_level   | ["set",[]]     |
risk_op      | ["set",[]]     |
-------------------------------
```

FCM_Collector_Config

This table contains configurations for the FCM plugin. This plugin collects and reports the network flow statistics.

Name	Type	Description
name	string	Name of the plugin used for network flow collection. **Important!** Do not modify the name of the plugin.
interval	integer	Collection interval for network flow samples
filter_name	string	Name of FCM_Filter to be applied on network flows during the collection time
report_name	string	Name of FCM_Report_Config to be used for sending reports about the collected flows
other_config	key/value map	Contains plugin-specific configurations such as shared library path, entry function name of the plugin, and conntrack zone to be used for flow collection.

Example:

```
# ovsh s FCM_Collector_Config
-----------------------------------------------------------------------------------------
_uuid        | 7315~98ec                                                                |
_version     | 7fe0~97b1                                                                |
filter_name  | ip_filter                                                                |
interval     | 10                                                                       |
name         | ct_stats                                                                 |
other_config | ["map",[["ct_zone","1"],["dso_init","ct_stats_plugin_init"],["dso_path", |
             : "/usr/plume/lib/"]]]                                                     :
report_name  | ip_flow_report                                                           |
-----------------------------------------------------------------------------------------
```

FCM_Filter

This table contains various FCM filter attributes that can be applied to the network flows.

Name	Type	Description
name	string	Name of the filter
index	integer	Priority of the filter. Index 0 is applied first, and so on.
smac	set of strings	Array of source mac addresses. Accepts smac Tags.
dmac	set of strings	Array of destination mac addresses. Accepts dmac tags.
vlanid	set of integers	Array of VLAN ID values
src_ip	set of strings	Array of source IP addresses
dst_ip	set of strings	Array of destination IP addresses
src_port	set of strings	Array of source port values
dst_port	set of strings	Array of destination port values
proto	set of integers	Array of protocol number values
smac_op	enum (*in*, *out*, *none*)	**in** - Flows matching *‘smac’* values will be considered for final *‘action’* **out** - Flows not matching *‘smac’* values will be considered for final *‘action*’
dmac_op	enum (*in*, *out*, *none*)	**in** - Flows matching *‘dmac’* values will be considered for final *‘action’* **out** - Flows not matching *‘dmac’* values will be considered for final *‘action*’
vlanid_op	enum (*in*, *out*, *none*)	**in** - Flows matching ‘*vlanid’* values will be considered for final *‘action’* **out** - Flows not matching ’*vlanid’* values will be considered for final *‘action’*
src_ip_op	enum (*in*, *out*,	**in** - Flows matching ’*src_ip’* values will be considered for final *‘action’*

	none)	**out** - Flows not matching ’*src_ip’* values will be considered for final *‘action’*
dst_ip_op	enum (*in*, *out*, *none*)	**in** - Flows matching ’*dst_ip’* values will be considered for final *‘action’* **out** - Flows not matching ’*dst_ip’* values will be considered for final *‘action’*
src_port_op	enum (*in*, *out*, *none*)	**in** - Flows matching *’src_port’* values will be considered for final *‘action’* **out** - Flows not matching *‘src_port’* values will be considered for final *‘action’*
dst_port_op	enum (*in*, *out*, *none*)	**in** - Flows matching ‘*dst_port’* values will be considered for final *‘action’* **out** - Flows not matching *‘dst_port*’ values will be considered for final *‘action’*
proto_op	enum (*in*, *out*, *none*)	**in** - Flows matching ‘*proto’* values will be considered for final *‘action’* **out** - Flows not matching *‘proto’* values will be considered for final *‘action’*
pktcnt	integer	Number of packets in a flow
pktcnt_op	enum (*lt*, *leq*, *gt*, *geq*, *eq*, *neq*)	**le** - Flows having packet count less than *‘pktcnt’* will be considered for final *‘action’* **leq** - Flows having packet count less than or equal to ` *pktcnt’* will be considered for final *‘action’* **gt** - Flows having packet count greater than *‘pktcnt’* will be considered for final *‘action’* **geq** - Flows having packet count greater than or equal to *‘pktcnt’* will be considered for final *‘action’* **eq** - Flows having packet count equal to *‘pktcnt’* will be considered for final *‘action’* **neq** - Flows having packet count not equal to *‘pktcnt’* will be considered for final *‘action’*
action	enum (*include*, *exclude*)	Final action of the considered flows based on the above said options: **include** - Flow will be included for further processing in FCM **exclude** - Flow will be excluded from further processing in FCM
other_config	set of key:value	Not used

Example:

```
# ovsh s FCM_Filter
------------------------------------------------
_uuid        | ba66~04a1      | fe11~6498      |
_version     | 6227~3ad0      | 7b57~780b      |
action       | include        | include        |
dmac         | ${iot_devices} | ["set",[]]     |
dmac_op      | in             | ["set",[]]     |
dst_ip       | ["set",[]]     | ["set",[]]     |
dst_ip_op    | ["set",[]]     | ["set",[]]     |
dst_port     | ["set",[]]     | ["set",[]]     |
dst_port_op  | ["set",[]]     | ["set",[]]     |
index        | 1              | 2              |
name         | ip_filter      | ip_filter      |
other_config | ["map",[]]     | ["map",[]]     |
pktcnt       | ["set",[]]     | ["set",[]]     |
pktcnt_op    | ["set",[]]     | ["set",[]]     |
proto        | ["set",[]]     | ["set",[]]     |
proto_op     | ["set",[]]     | ["set",[]]     |
smac         | ["set",[]]     | ${iot_devices} |
smac_op      | ["set",[]]     | in             |
src_ip       | ["set",[]]     | ["set",[]]     |
src_ip_op    | ["set",[]]     | ["set",[]]     |
src_port     | ["set",[]]     | ["set",[]]     |
src_port_op  | ["set",[]]     | ["set",[]]     |
vlanid       | ["set",[]]     | ["set",[]]     |
vlanid_op    | ["set",[]]     | ["set",[]]     |
------------------------------------------------
```

FCM_Report_Config

This table is used for configuring statistics reports from FCM to the Cloud.

Name	Type	Description
name	string	Report config name
interval	integer	Reporting interval of collected flows to MQTT server
format	enum (*cumulative*, *delta*, *raw*)	Specifies format of reporting of flows. **cumulative** - Reports consolidated bytes/packets of flows from the start of flow at each report interval **delta** - Reports difference in bytes/packets of flows between each consecutive report intervals.
hist_interval	integer	Histogram interval of reports. Not used.
hist_filter	string	Name of FCM_Filter to be used for histogram reports. Not used.
report_filter	string	Name of FCM_Filter to be used when reporting
mqtt_topic	string	MQTT Topic to be used for reporting
other_config	set of key:value	Not used.

Example:

```
# ovsh s FCM_Report_Config
-----------------------------------------------------------------------
_uuid         | bfbe~c686                                             |
_version      | 1689~4b6d                                             |
format        | cumulative                                            |
hist_filter   | ["set",[]]                                            |
hist_interval | 0                                                     |
interval      | 60                                                    |
mqtt_topic    | IP/Flows/IoT/dog1/4C777013FE/5c58ae5050d44e0b8df6da11 |
name          | ip_flow_report                                        |
other_config  | ["map",[]]                                            |
report_filter | ["set",[]]                                            |
-----------------------------------------------------------------------
```

IP_Interface

This table includes information for L3 configuration and status reporting. The IPv6 tables refer to or are referenced by this table.

Name	Type	Description
name	string	Name of the IP interface
enable	boolean	Enable or disable the interface
status	enum	Current operational state of the interface. Enumeration of: ● up ● down ● unknown ● dormant ● notpresent ● lowerlayerdown ● error
interfaces	set of uuids	Reference to the table:Interface
if_name	string	Interface name as configured in the operating system
ipv4_addr	set of uuids	Reference to the table: IPv4_Address
ipv6_addr	set of uuids	Reference to the table: IPv6_Address
ipv6_prefix	set of uuids	Reference to the table: IPv6_Prefix

IPv4_Address (not used)

This table is used for IPv4 address configuration and reporting. Tables that directly require an IPv4 address (either for configuration or reporting) strongly references rows in this table.

Name	Type	Description
enable	boolean	Enables or disables this IPv4 address
status	enum (*disabled*, *enabled*, *error*)	The status of this IPv4 address entry. Enumeration of: ● disabled ● enabled ● error
address	string	IPv4 address
subnet_mask	string	Subnet mask (CIDR notation)
type	enum	Addressing method used to assign the IPv4 address. Enumeration of: ● dhcp ikev2 ● auto_ip ● ipcp ● static

Note: Not used in OpenSync 2.0.

IPv6_Address

This table enables IPv6 address configuration and reporting. Tables that directly require an IPv6 address (either for configuration or reporting) refers to the rows in this table.

Name	Type	Description
enable	bool	Enables or disables this IPv6 address
status	enum (*disabled*, *enabled*, *error*)	The status of this IPv6 address entry. Enumeration of: ● disabled ● enabled ● error
address_status	enum	The status of the IPv6 address, indicating whether it can be used for communication. Enumeration of: ● preferred ● deprecated ● invalid ● inaccessible ● unknown ● tentative ● duplicate ● optimistic
address	string	IPv6 address
origin	enum	Mechanism using which the IPv6 address is assigned. Enumeration of: ● auto_configured ● dhcp ● ikev2 ● map ● well_known ● static
prefix	string	Prefix of the IPv6 address (CIDR notation)
preferred_lifetime	string	The time at which this address will cease to be preferred (i.e. will become deprecated), or empty if not known. For an infinite lifetime, the parameter value MUST be infinite.
valid_lifetime	string	The time at which this address will cease to be valid (i.e. will become invalid), or empty if unknown. For an infinite lifetime, the parameter value MUST be infinite.

IPv6_Prefix

This table enables IPv6 prefix configuration and reporting. Tables that directly require an IPv6 prefix (either for configuration or reporting) refer to the rows in this table.

Name	Type	Description
enable	boolean	Enables or disables this IPv6 prefix
status	enum (*disabled*, *enabled*, *error*)	The status of this IPv6 prefix entry. Enumeration of: ● disabled ● enabled ● error
prefix_status	enum	The status of the IPv6 prefix, indicating whether it can be used for communication. Enumeration of: ● preferred ● deprecated ● invalid ● inaccessible ● unknown
address	string	IPv6 address prefix
origin	enum	Mechanism using which the IPv6 prefix was assigned or most recently updated. Enumeration of: ● auto_configured ● prefix_delegation ● ra ● well_known ● static ● child
static_type	enum	Static prefix sub-type. For a Static prefix, this can be set to PrefixDelegation or Child, thereby creating an unconfigured prefix of the specified type that will be populated with preference to creating a new instance. This allows the controller to pre-create the "prefix slots" with known path names that can be referred to from elsewhere in the data model before they are populated. Enumeration of: ● static ● inapplicable ● prefix_delegation ● child
parent_prefix	uuid	The value is a reference in the table:IPv6_Prefix. If the referenced object is deleted, the parameter value MUST be set to an empty string. Indicates the parent prefix from which this prefix was derived. The parent prefix is relevant only for Child prefixes and for Static Child prefixes (both of

		which will always be on downstream interfaces), i.e. for Origin=Child and for (Origin,StaticType) = (Static,Child) prefixes.
child_prefix_bits	string	A prefix that specifies the length of Static Child prefixes and how they are derived from their ParentPrefix. It will be used if, and only if, it is not an empty string and is longer than the parent prefix (if it is not used, derivation of such prefixes is implementation-specific). Any bits to the right of the parent prefix are set to the bits in this prefix. For example, for a parent prefix of fedc::/56, if this parameter had the value 123:4567:89ab:cdef::/64, the child /64 would be fedc:0:0:ef::/64. For a parent prefix of fedc::/60, the child /64 would be fedc:0:0:f::/64.
on_link	boolean	On-link flag [Section 4.6.2/RFC4861] as received (in the RA) for RouterAdvertisement. Indicates whether this prefix can be used for on-link determination.
autonomous	boolean	Autonomous address configuration flag [Section 4.6.2/RFC4861] as received (in the RA) for RouterAdvertisement. Indicates whether this prefix can be used for generating global addresses as specified by SLAAC [RFC4862].
preferred_lifetime	string	This parameter is based on ipAddressPrefixAdvPreferredLifetime from [RFC4293]. The time at which this prefix will cease to be preferred (i.e. will become deprecated), or empty if not known. For an infinite lifetime, the parameter value MUST be infinite.
valid_lifetime	string	This parameter is based on ipAddressPrefixAdvValidLifetime from [RFC4293]. The time at which this prefix will cease to be valid (i.e. will become invalid), or empty if not known. For an infinite lifetime, the parameter value MUST be infinite.

DHCPv4_Client (not used)

This table enables DHCPv4 client configuration and status reporting.

Name	Type	Description
enable	boolean	Enable or disable the DHCPv4 Client
ip_interface	uuid	Reference to the table:IP_interface
request_options[0..32]	set of integers	DHCPv4 options requested by the client
received_options[0..32]	set of uuids	Reference to table:DHCP_Option. DHCPv4 options are received from the server.
send_options[0..32]	set of uuids	Reference to table:DHCP_Option. DHCPv4 options sent to the server.

Note: Not used in OpenSync 2.0.

DHCPv6_Client

The table enables DHCPv6 client configuration and status reporting.

Name	Type	Description
enable	boolean	Enable or disable the DHCPv6 Client
ip_interface	uuid	Reference to the table:IP_interface
request_address	boolean	Enables or disables inclusion of the Identity Association for Non-Temporary Address.
request_prefixes	boolean	Enables or disables inclusion of the Identity Association for Prefix Delegation.
rapid_commit	boolean	Enables or disables inclusion of the Rapid Commit.
renew	boolean	When set to true, the Client renews its DHCPv6-supplied information.
request_options	set of integers	DHCPv6 options requested by the client
received_options	set of uuids	Reference to table:DHCP_Option. DHCPv4 options received from the server.
send_options	set of uuids	Reference to table:DHCP_Option. DHCPv4 options sent to the server.

DHCP_Option

This table stores various DHCP options related to DHCPv4 and DHCPv6 configuration, and status reporting. IPv6 and IPv4 DHCP configuration tables refer to the rows in this table.

Name	Type	Description
enable	bool	True if this option is active, or false if ignored
version	enum (*v4*, *v6*)	DHCP version for this option, used only to help read the table. It should be consistent with what is configured in the table:DHCPv4_Client or table:DHCPv6_Client. Enumeration of: ● v4 ● v6
type	enum (*rx*, *tx*)	Type of the DHCP option, used only to help read the table. It should be consistent with what is configured in the table:DHCPv4_Client or table:DHCPv6_Client. Enumeration of: ● rx ● tx
tag	integer	Option tag (code)
value	string	The DHCP Client option value. Base64-encoded. The maximum size of a 255-byte binary string encoded to base64 is 340.

DHCPv4_Server (not used)

Used for DHCPv4 server configuration and status reporting.

Name	Type	Description
interface	uuid	Reference to table:IP_Interface. Parent interface
status	enum (*disabled*, *enabled*, *error*)	Indicates the status of this entry. Enumeration of: - disabled (origin: cloud) - enabled (origin: cloud) - error (origin: device)
min_address	string	Specifies the first IPv4 address in the pool to be assigned by the DHCP server on the interface.
max_address	string	Specifies the last IPv4 address in the pool to be assigned by the DHCP server on the interface.
lease_time	integer	Lease time in seconds. -1 indicates infinite.
options[0..256]	set of uuids	Reference to table:DHCP_Option. DHCP options offered to clients.
static_address[0..64]	set of uuids	Reference to table:DHCPv4_Lease. List of statically assigned IP addresses.
leased_address[0..256]	set of uuids	Reference to table:DHCPv4_Lease. List of leased IP addresses.

Note: Not used in OpenSync 2.0.

DHCPv4_Lease (not used)

Used for configuration and reporting of DHCPv4 leases. The DHCPv4_Server table refers to this table.

Name	Type	Description
status	enum (*leased*, *static*, *error*)	Indicates the status of this entry. Enumeration of: **leased** - This is a lease, reported by manager (origin: device) **static** - Static lease entry (origin: cloud) **error** - Error applying static entry (origin: device)
address	string	IPv4 Address of lease/static entry
hwaddr	string	Hardware address (MAC) of lease/static entry
hostname	string	Hostname
leased_time	integer	Lease time. Only applicable if status==leased.
leased_fingerprint	string	DHCP fingerprint. Only applicable if status==leased.

Note: Not used in OpenSync 2.0.

DHCPv6_Server

This table enables DHCPv6 server configuration and status reporting.

Name	Type	Description
interface	uuid	Reference to IP_Interface
status	enum (*disabled*, *enabled*, *error*)	Indicate the status of this entry. Enumeration of: - disabled (origin: cloud) - enabled (origin: cloud) - error (origin: device)
prefixes	set of uuids	Reference to table:IPv6_Prefix. List of prefixes that the DHCPv6 server will be offering (stateful DHCP).
prefix_delegation	boolean	If “True”, this Server row is used for prefix delegation
options	set of uuids	Reference to table:DHCP_Option. DHCP options offered to clients.
lease_prefix	set of uuids	Reference to table:DHCPv6_Lease. List of leased IP prefixes.
static_prefix	set of uuids	Reference to table:DHCPv6_Lease. List of statically assigned IP prefixes/addresses.

DHCPv6_Lease

Used for configuration and reporting of DHCPv6 leases. The DHCPv6_Server table refers to this table.

Name	Type	Description
status	enum (*leased*, *static*, *error*)	Indicates the status of this entry. **leased** - This is a lease, reported by the manager (origin: device) **static** - Static lease entry (origin: cloud) **error** - Error applying static entry (origin: device)
prefix	string	IPv6 address of prefix (/XX notation)
duid	string	Client unique identifier (DUID) (2 bytes header + 128 bytes max data in hex notation without separators)
hwaddr	string	Hardware address (MAC) of lease/static entry in hex notation
hostname	string	Assigned/Requested hostname
leased_time	integer	Leased time

IPv6_RouteAdv

This table enables IPv6 Router Advertisement configuration and status reporting.

Name	Type	Description
interface	set of uuids	Reference to table:IP_Interface. Parent interface.
status	enum (*disabled*, *enabled*, *error*)	Router advertisement status: **disabled** - disable this entry (origin: cloud) **enabled** - enable Router Advertisement on interface (origin: cloud) **error** - error configuring Router Advertisement (origin: device)
prefixes	set of uuids	Reference to the table:IPv6_Prefix. Prefixes advertised through RA.
managed	boolean	Managed address configuration -- the (M) flag
other_config	boolean	Other configuration -- the (O) flag
home_agent	boolean	Home Agent flag -- the (H) flag
max_adv_interval	integer	Maximum time allowed between sending unsolicited multicast Router Advertisements from the interface; in seconds.
min_adv_interval	integer	Minimum time allowed between sending unsolicited multicast Router Advertisements from the interface; in seconds. The value must be greater than 3/4 * max_adv_interval.
default_lifetime	integer	Router lifetime. 0 indicates that the router should not be used as default router.
preferred_router	enum (*low*, *medium*, *high*)	reference associated with the default router, as either "low", "medium", or "high".
mtu	integer	The MTU option is used in router advertisement messages to ensure that all nodes on a link use the same MTU value in those cases where the link MTU is not a well- known value.
reachable_time	integer	The time, in milliseconds, that a node assumes a neighbor is reachable after having received a reachability confirmation. Used by the Neighbor Unreachability

		Detection algorithm (see Section 7.3 of RFC 4861). A value of zero means unspecified (by this router).
retrans_timer	integer	The time, in milliseconds, between retransmitted Neighbor Solicitation messages. Used by address resolution and the Neighbor Unreachability Detection algorithm (see Sections 7.2 and 7.3 of RFC 4861). A value of zero means unspecified (by this router).
current_hop_limit	integer	The default value that should be placed in the Hop Count field of the IP header for outgoing (unicast) IP packets. The value should be set to the current diameter of the Internet. The value zero means unspecified (by this router).
rdnss	set of uuids	Reference to the table:IPv6_Address. Recursive DNS servers advertised via RA.
dnssl	set of strings	List of DNS search domains that will be advertised via RA

IPv6_Neighbors

This table enables IPv6 neighbor reporting. This is a tentative list of neighbors that are participating in an IPv6-enabled network.

Name	Type	Description
address	string	Client device IPv6 address
hwaddr	string	Client device MAC address
if_name	string	Interface name

Example:

```
root@caesar:~# ovsh -T s IPv6_Neighbors -w if_name==br-home.tndp
+-----------+-----------+----------------------------------------+-------------------+--------------+
| _uuid     | _version  | address                                | hwaddr            | if_name      |
+-----------+-----------+----------------------------------------+-------------------+--------------+
| 3626~9727 | fd69~fc9b | fe80::cd1:6a3f:29a1:35f9               | 00:05:1B:D1:A5:7B | br-home.tndp |
| 9961~6ce5 | e5fa~7cd4 | fe80::225:90ff:fe87:175d               | 00:25:90:87:17:5D | br-home.tndp |
| 9895~3737 | 81dc~d033 | 2601:647:4900:1f63:115a:5cc2:f08f:7d68  | 26:16:29:D8:CF:99 | br-home.tndp |
| e754~0b49 | 5b84~decb | 2601:647:4900:1f63:cd32:d093:dd9e:dda1  | 26:16:29:D8:CF:99 | br-home.tndp |
+-----------+-----------+----------------------------------------+-------------------+--------------+
```

IPv4_Neighbors

ARP protocol provides IPv4-to-MAC mappings to the devices. This table stores the ARP mappings exchange. Collecting the ARP messages exchanged on the LAN enhances the *OpenSync* IPv4-to-MAC mapping performance.

Name	Type	Description
address	string	Client device IPv4 address
hwaddr	string	Client device MAC address
if_name	string	Incoming interface of the ARP packet which triggered the entry.
source	string	Source of ARP mapping - either FSM or the system ARP table.

IGMP_Config

This table sets various IGMP parameters in the system.

Name	Type	Description
fast_leave_enable	boolean	Enables leaving a group when a leave report is received without sending an IGMP query for the group.
last_member_query_interval	integer	The interval defines the maximum response time advertised in IGMP group-specific queries.
maximum_groups	integer	The maximum number of groups that can be subscribed to by the system.
maximum_members	integer	The maximum number of members in a group that can be supported.
maximum_sources	integer	Maximum number of addresses allowed in the source filter list for a multicast group.
query_interval	integer	Interval duration between queries.
query_response_interval	integer	The interval to wait for a response after sending out a query.
query_robustness_value	integer	Controls the IGMP query robustness variable. When the system receives an IGMP leave message on a shared network running IGMPv2, the query router must send an IGMP group query message for a specified number of times. The number of IGMP group query messages sent is determined by the robust count.

Note: This table is available in *OpenSync* 2.0 and later.

Example:

```
ovsh s IGMP_Config
----------------------------------------
_uuid                      | 3896~0ed6 |
_version                   | 1ef6~09ff |
fast_leave_enable          | false     |
last_member_query_interval | 150       |
maximum_groups             | 150       |
maximum_members            | 10        |
maximum_sources            | 30        |
query_interval             | 125       |
query_response_interval    | 20        |
query_robustness_value     | 10        |
---------------------------------------
```

MLD_Config

This table sets various MLD parameters in the system.

Name	Type	Description
fast_leave_enable	boolean	Enables leaving a group when a leave report is received without sending an MLD query for the group.
last_member_query_interval	integer	The interval defines the maximum response time advertised in MLD group-specific queries.
maximum_groups	integer	The maximum number of groups that can be subscribed to by the system.
maximum_members	integer	The maximum number of members in a group that can be supported.
maximum_sources	integer	Maximum number of addresses allowed in the source filter list for a multicast group.
query_interval	integer	The interval between different queries.
query_response_interval	integer	The interval to wait for a response after sending out a query.
query_robustness_value	integer	Controls the MLD query robustness variable.When the system receives an MLD leave message on a shared network running MLDv2, the query router must send an MLD group query message for a specified number of times. The number of MLD group query messages sent is determined by the robust count.

Note: This table is available in *OpenSync* 2.0 and later.

Example:

```
----------------------------------------
_uuid                       | fd46~7352 |
_version                    | 1d59~bee9 |
fast_leave_enable           | false     |
last_member_query_interval  | 160       |
maximum_groups              | 40        |
maximum_members             | 20        |
maximum_sources             | 30        |
query_interval              | 125       |
query_response_interval     | 20        |
query_robustness_value      | 4         |
----------------------------------------
```

Node_Services

This table lists all *OpenSync* services (managers) that are available on the device, their configuration and current status. Services can be dynamically started or stopped by modifying the **enable** column of this table.

Name	Type	Description
service	string	The name of the service name (executable name)
enable	bool	“True” if the service is enabled
status	enum	One of: ● enabled: If the service has been started ● disabled: If the service has been stopped ● error: If there was an error during starting/stopping of the service
other_config	map	Additional service configuration. This is the current list of values as interpreted by the DM: ● needs_plan_b (boolean): Must be stopped (“True”) whether a service crash requires restart of OpenSync ● restart_delay (integer): Restart delay in seconds ● always_restart (true): If true, services should always be restarted even if they are killed by the signals that usually do not trigger a restart.

Note: This table is available in *OpenSync* 2.0 and later.

Example:

```
+-----------+-----------+--------+------------------------------------------------------------+---------+---------+
| _uuid     | _version  | enable | other_config                                               | service | status  |
+-----------+-----------+--------+------------------------------------------------------------+---------+---------+
| 7b84~a134 | e9f0~b11a | true   | ["map",[["needs_plan_b","true"]]]                          | blem    | enabled |
| 5afe~e959 | 3952~a6b8 | true   | ["map",[]]                                                 | bm      | enabled |
| 80d5~cec1 | 204e~399e | true   | ["map",[["needs_plan_b","true"]]]                          | cm      | enabled |
| 24fb~7096 | 0db9~988e | true   | ["map",[]]                                                 | csc_man | enabled |
| 3582~7096 | e4a1~e3c3 | true   | ["map",[["always_restart","true"],["restart_delay","-1"]]] | fcm     | enabled |
| e35d~45ac | 8537~e9b2 | true   | ["map",[["needs_plan_b","true"]]]                          | fm      | enabled |
| 432c~9d74 | 4bc4~f874 | true   | ["map",[["always_restart","true"],["restart_delay","-1"]]] | fsm     | enabled |
| e178~f0fc | 7bd4~3fa5 | true   | ["map",[["needs_plan_b","true"]]]                          | nm      | enabled |
| d617~4bb9 | bf2b~2e44 | true   | ["map",[["needs_plan_b","true"]]]                          | om      | enabled |
| a4b5~5096 | be53~296c | true   | ["map",[]]                                                 | pm      | enabled |
| b391~47b0 | 88ad~88a8 | true   | ["map",[]]                                                 | ppm     | enabled |
| de54~48da | 9b1e~e683 | true   | ["map",[]]                                                 | qm      | enabled |
| 4fbe~ba7e | 8c23~f270 | true   | ["map",[]]                                                 | sm      | enabled |
| 4921~c04f | 834a~aab3 | true   | ["map",[["needs_plan_b","true"]]]                          | um      | enabled |
| 3f51~74e2 | d955~f38a | true   | ["map",[["needs_plan_b","true"]]]                          | wm      | enabled |
+-----------+-----------+--------+------------------------------------------------------------+---------+---------
```

OMS_Config

Normally, the device firmware upgrades require upgrading the entire firmware image. However, it is also possible to upgrade only specific firmware modules. Examples of such upgrades are signatures for third party components, or adding new modules or managers. OMS_Config is the table that the receiving modules are monitoring for the object updates. This table lists the available versions for a given object name. In case of multiple versions for the same object name, it is up to the receiving module to decide which one to use.

Name	Type	Description
object_name	string	Name of the object to be upgraded
version	string	Version of the object (set by the Cloud)
other_config	string	

Object_Store_Config

The Object_Store_Config table enables the Cloud-to-device communication when upgrading only specific firmware modules.

Name	Type	Description
dl_url	string	Download url: ● Set by the Cloud when a new object is available ● Cleared by device when download is done (even if download fails)
dl_timeout	integer	Download timeout: ● Set by the Cloud for download timeout ● Cleared by device when download is done (even if download fails)
name	string	Name of the object: ● Set by the Cloud when a new download is triggered ● Set by the device at boot (from the already installed objects)
version	string	Version of the object: ● Set by the Cloud when a new download is triggered ● Set by device at boot (from already installed objects)

Object_Store_State

This table enables the device-to-Cloud communication when upgrading only specific firmware modules.

Name	Type	Description
fw_integrated	boolean	Object pre integrated in the firmware image: ● True if the object is pre integrated in the image - it is installed by default. ● False if the object is installed from the Cloud (Object_Store_Config table). The pre- integrated objects can not be removed.
status	enum	Status of object: ● always set only by the device ● enum: ○ ACTIVE ○ ERROR ○ DOWNLOAD_STARTED ○ DOWNLOAD_COMPLETED ○ DOWNALOD_ERROR ○ INSTAL_FAILED ○ INSTAL_COMPLETED ○ VER_MISMATCH
name	string	Name of the object: ● Set by the device
version	string	Version of object: ● Set by the device at boot

Captive_Portal

This table configures Captive Portal authentication parameters on the devices.

Name	Type	Description
name	string	
uam_url	string	URL of the NAS server that will be accessed using the device and location id
additional_headers	string	Map of key-value pairs for any tinyproxy config
other_config	string	

Reboot_Status

This table enables reporting of device reboot/boot reason. Entries in this table are automatically populated by the device upon system startup. Table rows are persistent until deleted.

Name	Type	Description
count	int	This column indicates the reboot sequence number. Each time the system boots, this counter increases by 1.
reason	string	Optional: Information provided in addition to the type. This string may include, for example, a snippet from the kernel stack trace upon crash, reason why the health-check failed, etc.
type	enum	This field indicates the reboot type and is an enum of: ● UNKNOWN - Unable to determine reboot reason ● COLD_BOOT - First boot after device power on ● POWER_CYCLE - Device was power cycled ● WATCHDOG - Reboot due to watchdog timeout ● CRASH - Reboot due to system crash ● USER - Reboot issued by user (reboot command) ● DEVICE - Firmware initiated reboot ● HEALTH_CHECK - Reboot due to health check failure ● UPGRADE - Reboot due to system upgrade ● THERMAL - Reboot due to critical thermal event ● CLOUD - Cloud-initiated reboot (for example, reboot from NOC)

Note: This table is available in *OpenSync* 2.0 and later.

Example:

```
|+-----------+-----------+-------+--------------------------+-------------+
|  _uuid      | _version  | count | reason                   | type        |
|+-----------+-----------+-------+--------------------------+-------------+
|   a66d~d8f5 | 0277~bbc7 | 23    | delayed-reboot           | CLOUD       |
|   1455~5c54 | 1189~6df6 | 74    |                          | COLD_BOOT   |
|   8a66~c0aa | 118c~1c84 | 69    |                          | COLD_BOOT   |
|   9fe1~4367 | 11c3~ae43 | 61    | delayed-reboot           | CLOUD       |
|   dbdf~9a35 | 1a79~2b54 | 40    | delayed-reboot           | CLOUD       |
|   0676~345e | 21d5~831f | 43    | delayed-reboot           | CLOUD       |
|   3ac8~f9c2 | 26fe~d218 | 32    | delayed-reboot           | CLOUD       |
|   ef64~976b | 2e36~3c13 | 24    | delayed-reboot           | CLOUD       |
|   3ece~6f2a | 348d~6a46 | 50    | delayed-reboot           | CLOUD       |
|   37cd~5fba | 374f~3e3c | 84    | (unknown)                | POWER_CYCLE |
|   0569~53c5 | 3887~3617 | 67    | delayed-reboot           | CLOUD       |
|   dac1~7cfb | 3bf6~cd03 | 54    | delayed-reboot           | CLOUD       |
|+-----------+-----------+-------+--------------------------+-------------+
```

Service_Announcement

This table contains service name, metadata, and interval for mDNS service advertising.

Name	Type	Description
name	string	Name of the service being announced.
protocol	string	This is the type, i.e http or ftp. For example: _http._tcp or _smb._udp
port	integer	Port number of the service, for example: 80 or 443.
txt	string	Text records of the service. This could be any arbitrary text with the format “type=value” Example: "v1","https://bd.plume.com/api/v1/"

Note: This table is available in *OpenSync* 2.0 and later.

Example:

```
------------------------------------------------------------
_uuid    | aa8d~f70f                                       |
_version | 1d4c~787e                                       |
name     | bw.plume                                        |
port     | 3000                                            |
protocol | _smb._udp                                       |
txt      | ["map",[["v1","https://bd.plume.com/api/v1/"]]] |
------------------------------------------------------------
```