

---

# Tracemac Documentation

*Release 2.0.0*

**Mattias Måhl**

April 26, 2016



## CONTENTS

<b>1</b>	<b>Sources</b>	<b>3</b>
1.1	libs . . . . .	3
<b>2</b>	<b>Indices and tables</b>	<b>9</b>
	<b>Python Module Index</b>	<b>11</b>
	<b>Index</b>	<b>13</b>



Contents:

Specifications of modules in project Tracemac.

Modules:



## SOURCES

## 1.1 libs

### 1.1.1 Switch\_Object module

Created 2013-08-27

@author Mattias Måhl

Class Switch\_Object

This is an object to store Switchdata in.

**class** Switch\_Object.**Sw\_Object** (\*ipaddress)

Bases: object

Initilization of the Switch Object setting up the switch and getting the information from the switch through SNMP.

Defining the tuples and variables to store the switch data.

**append\_mac\_to\_interface** (interface, macaddress)

function append\_mac\_to\_interface

a function to add mac-address to a specific interface. if interface not found add new interface and add the mac\_address to it.

**append\_neighbor** (interface, \*args)

function append\_neighbor

function to append Neighbor to the switchobjects array och neighbors.

**check\_if\_alive** (ipaddress)

function check\_if\_alive

function to ping host to see if it's alive.

raises error if it's a fail.

**check\_ip\_address** (ipaddress)

function check\_ip\_address

function to check if the ip-address provided is acceptable.

i.e. number(dot)number(dot)number(dot)number

return False if not and returns the same string if True

```
class cl_switch_interface (interface, mac_address)
```

Bases: object

```
class cl_switch_interface
```

Switch\_objects interface object to store mac-addresses associated with the interface.

```
class Sw_Object.cl_switch_neighbors (interface)
```

Bases: object

```
class cl_switch_neighbores
```

Switch\_Objects neighbors object to store the switches neighbors and witch interface their on.

```
interface = 0
```

```
ip_address = “
```

```
name = “
```

```
remote_interface = 0
```

```
Sw_Object.find_my_mac_address (mac_address)
```

function to search the Mac-table of the switch to find a specific MAC-address.

```
Sw_Object.find_switch_mac_address ()
```

function find\_switch\_mac\_address()

Do snmp request för oid: 1.3.6.1.4.1.11.2.14.11.5.1.1.6.0 (BaseMacAddress)

```
Sw_Object.get_interface (interface)
```

function get\_interface

searches registered interfaces and returns the interface\_object

```
Sw_Object.get_mac_address_list ()
```

function get\_mac\_address\_list

Does a SNMP-request to gather the mac-address-table from this switch-object.

```
Sw_Object.get_neighbor_at_interface (interface)
```

function get\_neighbor\_at\_interface

function to get the neighbor at a specific interface.

searches registered neighbors and returns a hit.

```
Sw_Object.get_neighbors ()
```

function get\_neighbores

gets the list of neighbors and stores them in the list ‘switch\_neighbors’

```
Sw_Object.get_switch_data ()
```

Do snmp-recuest to get the system name of the target switch.

```
Sw_Object.is_interface_a_neighbor (interface)
```

function is\_interface\_a\_neighbor

Check to see if the interface has neighbors registered.

Returns True or False

```
exception Switch_Object.sw_error (error_msg)
```

Bases: Exception



## 1.1.2 Trace\_Functions module

Created 2013-08-27

@author Mattias Måhl

Class Tracefunctions

Functions to administrate search for MAC-address

**class** Trace\_Functions.**Tracefunctions**

Bases: object

**class** Trace\_arguments

Bases: object

Class Trace\_arguments Object to store the supplied arguments.

**dump\_file** = ""

**in\_file** = ""

**start\_ip\_address** = ""

**target\_ip\_address** = ""

**target\_mac\_address** = ""

**verbose** = False

**class** Tracefunctions.**Trace\_result**

Bases: object

Class Trace\_result Object to store a single result from the search. This stores the path the program took to find the port witch has the mac-address.

**SW\_O** = []

**failed** = False

**search\_ip** = ""

**search\_mac** = ""

**trace\_end** = ""

Tracefunctions.**chk\_system\_args** (argv)

function chk\_system\_args function to check if argumest supplied are correct and that mandatory argurments are supplied.

Tracefunctions.**fix\_macaddress** (MAC)

function fix\_macaddres this function fixes the mac-address to be exactly the same even if the user supplies it in different formats. i.e 121212-121212 will become 121212121212 and likewise 12:12:12:12:12:12 will become 121212121212.

Tracefunctions.**get\_mac\_address\_from\_ip** (ipaddress)

function get\_mac\_address\_from\_ip function to arping an ip address to get the Mac-address assosiated with it. It's important that the target ip address is on the same network as the machine running the program and NOT routed! If it's routed the routers mac address will be the one reported by the arping!

Tracefunctions.**get\_system\_args** (argv)

function get\_system\_args function to parse system arguments in cli-envioronment. Args:

-h, -help= Display helptext for cli-command.

**-i, --ipaddress=** Target ip-address to find in the network. *note* this implies access to both mgmnt- and target network (if separate)

**-s, --startingip=** IP-address of the first switch in the cascade.

**-o, --out=** Output logging top specified file.

**-m, --macaddress=** Target MAC-address to find in the network.

**-v, --verbose=** Enables verbose output to standard output and logging file if '-o/-out=' is used.

**-f, --in-file** Enables function to loop through a list of targets in a text file.  
*note* One target per line.

Tracefunctions.**ping\_my\_address** (*ipaddress, cnt*)  
function ping\_my\_address simple function to ping the target IP-address to keep the Mac-address table up-to-date.

Tracefunctions.**printhelp** ()  
function printhelp duh!

### 1.1.3 frm\_main module

Created 2016-04-12 @author: Mattias Måhl

```
class frm_main.frm_main
    Bases: tkinter.Tk

    Start object to render the application window.

    use:

    import libs.frm_main as mainwindow

    if __name__ == "__main__":
        app = mainwindow.App()
        app.mainloop()

    createWidgets (frame)
        Creates and lays out the widgets for the mainwindow.

    quit (*event)
```

### 1.1.4 tracemac module

**\*\* TraceMac \*\*** Traces a mac-address to a specific port in a switch. **\*\* Version: 1.7.2 \*\* License: GPLv2**

```
tracemac.pr (pr_str)
    function pr function to print out messages to stdout using a specified format. This is used for verbose output to
    stdout and logfile.

tracemac.printout (msg, *w)
    function printout stardart output to stdout and logfile.

tracemac.read_infile (filename)
    function read_infile this function serves to read the input file an store the targets for the search engine.

tracemac.split_string_into_chunks (text, length=94)
    function split_string_into_chunks this function serves to sprit output into chunks that's under specified length.
    Default value is 94 chars.
```

`tracemac.vreport` (*header*, *\*msg*)

function `vreport` creates and outputs verbose output of progress.

`tracemac.write_to_file` (*line*)

function `write_to_file` if there is a file specified in options this will write to it.



## INDICES AND TABLES

- `genindex`
- `modindex`
- `search`



**f**

frm\_main, [6](#)

**s**

Switch\_Object, [3](#)

**t**

Trace\_Functions, [5](#)

tracemac, [6](#)





append\_mac\_to\_interface() (Switch\_Object.Sw\_Object method), 3  
 append\_neighbor() (Switch\_Object.Sw\_Object method), 3  
 check\_if\_alive() (Switch\_Object.Sw\_Object method), 3  
 check\_ip\_address() (Switch\_Object.Sw\_Object method), 3  
 chk\_system\_args() (Trace\_Functions.Tracefunctions method), 5  
 createWidgets() (frm\_main.frm\_main method), 6  
 dump\_file (Trace\_Functions.Tracefunctions.Trace\_arguments attribute), 5  
 failed (Trace\_Functions.Tracefunctions.Trace\_result attribute), 5  
 find\_my\_mac\_address() (Switch\_Object.Sw\_Object method), 4  
 find\_switch\_mac\_address() (Switch\_Object.Sw\_Object method), 4  
 fix\_macaddress() (Trace\_Functions.Tracefunctions method), 5  
 frm\_main (class in frm\_main), 6  
 frm\_main (module), 6  
 get\_interface() (Switch\_Object.Sw\_Object method), 4  
 get\_mac\_address\_from\_ip() (Trace\_Functions.Tracefunctions method), 5  
 get\_mac\_address\_list() (Switch\_Object.Sw\_Object method), 4  
 get\_neighbor\_at\_interface() (Switch\_Object.Sw\_Object method), 4  
 get\_neighbors() (Switch\_Object.Sw\_Object method), 4  
 get\_switch\_data() (Switch\_Object.Sw\_Object method), 4  
 get\_system\_args() (Trace\_Functions.Tracefunctions method), 5  
 in\_file (Trace\_Functions.Tracefunctions.Trace\_arguments attribute), 5  
 interface (Switch\_Object.Sw\_Object.cl\_switch\_neighbors attribute), 4  
 ip\_address (Switch\_Object.Sw\_Object.cl\_switch\_neighbors attribute), 4  
 is\_interface\_a\_neighbor() (Switch\_Object.Sw\_Object method), 4  
 name (Switch\_Object.Sw\_Object.cl\_switch\_neighbors attribute), 4  
 ping\_my\_address() (Trace\_Functions.Tracefunctions method), 6  
 pr() (in module tracemac), 6  
 printhelp() (Trace\_Functions.Tracefunctions method), 6  
 printout() (in module tracemac), 6  
 quit() (frm\_main.frm\_main method), 6  
 read\_infile() (in module tracemac), 6  
 remote\_interface (Switch\_Object.Sw\_Object.cl\_switch\_neighbors attribute), 4  
 search\_ip (Trace\_Functions.Tracefunctions.Trace\_result attribute), 5  
 search\_mac (Trace\_Functions.Tracefunctions.Trace\_result attribute), 5  
 split\_string\_into\_chunks() (in module tracemac), 6  
 start\_ip\_address (Trace\_Functions.Tracefunctions.Trace\_arguments attribute), 5  
 sw\_error, 4  
 SW\_O (Trace\_Functions.Tracefunctions.Trace\_result attribute), 5  
 Sw\_Object (class in Switch\_Object), 3  
 Sw\_Object.cl\_switch\_interface (class in Switch\_Object), 3  
 Sw\_Object.cl\_switch\_neighbors (class in Switch\_Object), 4  
 Switch\_Object (module), 3  
 target\_ip\_address (Trace\_Functions.Tracefunctions.Trace\_arguments attribute), 5  
 target\_mac\_address (Trace\_Functions.Tracefunctions.Trace\_arguments attribute), 5  
 trace\_end (Trace\_Functions.Tracefunctions.Trace\_result attribute), 5  
 Trace\_Functions (module), 5

Tracefunctions (class in Trace\_Functions), [5](#)  
Tracefunctions.Trace\_arguments (class in  
Trace\_Functions), [5](#)  
Tracefunctions.Trace\_result (class in Trace\_Functions), [5](#)  
tracemac (module), [6](#)  
  
verbose (Trace\_Functions.Tracefunctions.Trace\_arguments  
attribute), [5](#)  
vreport() (in module tracemac), [6](#)  
  
write\_to\_file() (in module tracemac), [7](#)