



CHAPTER 1

Application Programming Interface (API) Functions

This chapter provides information about the following Python Application Programming Interface (API) functions. This chapter includes the following sections:

- [Routes\(\)](#), page 1-1
- [show_arp_table\(\)](#), page 1-2
- [show_vsh_routes\(\)](#), page 1-2
- [show_hw_routes\(\)](#), page 1-3
- [verify_routes\(\)](#), page 1-4
- [verify_arp_table\(\)](#), page 1-5
- [CheckPortDiscards\(\)](#), page 1-6
- [transfer\(\)](#), page 1-7
- [CLI\(\)](#), page 1-7
- [get_output\(\)](#), page 1-8
- [rerun\(\)](#), page 1-9
- [History\(\)](#), page 1-9
- [get_history\(\)](#), page 1-10
- [clear_history\(\)](#), page 1-10

Routes()

Synopsis

`Routes()` - Class Object

Syntax

`Routes()`

Description

Instantiates an object of the Routes class.

Send document comments to nexus5k-docfeedback@cisco.com.

Parameters

None.

Returns

An object of Routes class.

Example

```
rObj = Routes()
```

show_arp_table()

Synopsis

```
show_arp_table()
```

Syntax

```
Routes.show_arp_table()
```

Description

Executes the **show ip arp** command and returns the output.

Parameters

None.

Returns

Returns the ARP table entries on the switch.

Example

```
routeObj = Routes()
data = routeObj.show_arp_table().get_output()
```

Sample Output

Flags: D - Static Adjacencies attached to down interface

```
IP ARP Table for context default
Total number of entries: 4
Address          Age      MAC Address      Interface
50.1.201.2      00:02:10  547f.ee40.5a7c   Vlan201
50.1.1.10       00:07:53  547f.ee62.f801   Ethernet1/34
50.1.2.10       00:08:31  547f.ee62.f801   Ethernet1/35
50.1.3.10       00:08:31  547f.ee62.f801   Ethernet1/35.1
```

<cisco.CLI object at 0xb7c1462c>

show_vsh_routes()

Synopsis

```
show_vsh_routes()
```

Send document comments to nexus5k-docfeedback@cisco.com.

Syntax

```
Routes.show_vsh_routes()
```

Description

Executes the show ip fib route and returns the output.

Parameters

None.

Returns

Returns the software route entries.

Example

```
routeObj = Routes()
data = routeObj.show_vsh_routes().get_output()
```

Sample Output

IPv4 routes for table default/base

```
-----+-----+-----
Prefix      | Next-hop      | Interface
-----+-----+-----
0.0.0.0/32   | Drop          | Null0
50.1.1.0/24  | Attached      | Ethernet1/34
50.1.1.0/32  | Drop          | Null0
50.1.1.10/32 | 50.1.1.10     | Ethernet1/34
50.1.1.100/32 | Receive       | sup-eth1
50.1.1.255/32 | Attached      | Ethernet1/34
50.1.2.0/24  | Attached      | Ethernet1/35
50.1.2.0/32  | Drop          | Null0
50.1.2.10/32 | 50.1.2.10     | Ethernet1/35
50.1.2.100/32 | Receive       | sup-eth1
50.1.2.255/32 | Attached      | Ethernet1/35
50.1.3.0/24  | Attached      | Ethernet1/35.1
50.1.3.0/32  | Drop          | Null0
50.1.3.10/32 | 50.1.3.10     | Ethernet1/35.1
50.1.3.100/32 | Receive       | sup-eth1
50.1.3.255/32 | Attached      | Ethernet1/35.1
```

<cisco.CLI object at 0xb7b0a6ac>

show_hw_routes()

Synopsis

```
show_hw_routes()
```

Syntax

```
Routes.show_hw_routes()
```

Description

Computes the hardware routes and returns the output.

Send document comments to nexus5k-docfeedback@cisco.com.

Parameters

None.

Returns

Returns the hardware route entries.

Example

```
routeObj = Routes()
data = routeObj.show_hw_routes()
```

Sample Output

```
-----+-----+-----
Prefix      | Next-hop      | Interface
-----+-----+-----
50.1.1.100/32    Receive      sup-eth1
50.1.2.100/32    Receive      sup-eth1
50.1.201.1/32    Receive      sup-eth1
0.0.0.0/32      Drop         Null0
50.1.3.0/32     Drop         Null0
50.1.201.0/32   Drop         Null0
50.1.2.255/32   Attached     sup-hi
50.1.1.255/32   Attached     sup-hi
60.1.1.0/32     Drop         Null0
50.1.3.255/32   Attached     sup-hi
50.1.201.255/32 Attached     sup-hi
255.255.255.255/32 Receive     sup-eth1
```

verify_routes()

Synopsis

```
verify_routes()
```

Syntax

```
Routes.verify_routes()
```

Description

Verifies the software and hardware routes.

Parameters

None.

Returns

Returns the number of routes matched and unmatched between hardware and software.

Example

```
routeObj = Routes()
found,nfound = routeObj.verify_routes()
```

Sample Output

```
Routes verified and found: 26
```

```
Routes not found:
```

Send document comments to nexus5k-docfeedback@cisco.com.

```
50.1.205.0/24      3
51.1.1.0/24       3
51.1.2.0/24       4
51.1.3.0/24       6
100.1.1.0/24      7
100.1.2.0/24      7
100.1.3.0/24      7
101.1.1.0/24      7
101.1.2.0/24      7
101.1.3.0/24      7
120.1.1.0/24      7
```

verify_arp_table()

Synopsis

```
verify_arp_table()
```

Syntax

```
Routes.verify_arp_table()
```

Description

Verifies the software and hardware ARP table entries.

Parameters

None.

Returns

Returns the number of ARP table entries matched and unmatched between hardware and software.

Example

```
routeObj = Routes()
found,notfound = routeObj.verify_arp_table()
```

Sample Output

```
Flags: D - Static Adjacencies attached to down interface
```

```
IP ARP Table for context default
```

```
Total number of entries: 4
```

Address	Age	MAC Address	Interface
50.1.201.2	00:02:31	547f.ee40.5a7c	Vlan201
50.1.1.10	00:08:15	547f.ee62.f801	Ethernet1/34
50.1.2.10	00:08:53	547f.ee62.f801	Ethernet1/35
50.1.3.10	00:08:53	547f.ee62.f801	Ethernet1/35.1

```
mac address:54:7f:ee:40:5a:7c
```

```
Arp entry for 50.1.201.2 547f.ee40.5a7c Vlan201 found in HW
```

```
mac address:54:7f:ee:62:f8:01
```

```
Arp entry for 50.1.1.10 547f.ee62.f801 Ethernet1/34 found in HW
```

```
mac address:54:7f:ee:62:f8:01
```

```
Arp entry for 50.1.2.10 547f.ee62.f801 Ethernet1/35 found in HW
```

```
mac address:54:7f:ee:62:f8:01
```

```
Arp entry for 50.1.3.10 547f.ee62.f801 Ethernet1/35.1 found in HW
```

Send document comments to nexus5k-docfeedback@cisco.com.

CheckPortDiscards()

Synopsis

CheckPortDiscards(<port>)

Syntax

CheckPortDiscards('ethernet1/1')

Description

Check the input discards for given port. If discard is more than 0, query and print the discard reason from broadcom.

Parameters

port

Returns

None.

Example

```
c = CheckPortDiscards('eth1/1')
```

Sample Output

```
Ethernet1/1 is up
Hardware: 100/1000/10000 Ethernet, address: 547f.ee57.dd28 (bia 547f.ee57.dd28)
MTU 1500 bytes, BW 10000000 Kbit, DLY 10 usec,
    reliability 255/255, txload 1/255, rxload 1/255
Encapsulation ARPA
Port mode is trunk
full-duplex, 10 Gb/s, media type is 10G
Beacon is turned off
Input flow-control is off, output flow-control is off
Rate mode is dedicated
Switchport monitor is off
EtherType is 0x8100
Last link flapped 00:42:16
Last clearing of "show interface" counters never
30 seconds input rate 5016 bits/sec, 627 bytes/sec, 6 packets/sec
30 seconds output rate 3232 bits/sec, 404 bytes/sec, 5 packets/sec
Load-Interval #2: 5 minute (300 seconds)
  input rate 4.69 Kbps, 7 pps; output rate 2.82 Kbps, 4 pps
RX
  297 unicast packets  20588 multicast packets  5 broadcast packets
  20890 input packets  1848701 bytes
  0 jumbo packets  0 storm suppression packets
  0 giants  0 input error  0 short frame  0 overrun  0 underrun
  0 watchdog  0 if down drop
  0 input with dribble  0 input discard(includes ACL drops)
  0 Rx pause
TX
  262 unicast packets  16151 multicast packets  5 broadcast packets
  16418 output packets  1407200 bytes
  0 jumbo packets
  0 output errors  0 collision  0 deferred  0 late collision
  0 lost carrier  0 no carrier  0 babble
  0 Tx pause
2 interface resets
```

Send document comments to nexus5k-docfeedback@cisisco.com.

zero discards

transfer()

Synopsis

```
transfer()
```

Syntax

```
transfer (<protocol>, <host>, <source>, <dest>, <vrf>, <login_timeout>, <user>, <password>)
```

Description

API to transfer file specified in <source> from <host> to the path mentioned in <dest> using <protocol>. Protocol can be **scp**, **tftp**, **ftp** or **sftp**.

Parameters

protocol, host, source, dest, vrf, login_timeout, user, password.

Returns

Returns True if transfer was successful.

Example

Transfer using scp:

```
c = transfer("scp", "10.193.190.100", "/tftpboot/transfer_test_image", "transfer_test_image", user="scpUser", password="scpPasswd")
```

Transfer using sftp:

```
c = transfer("sftp", "10.193.190.100", "/tftpboot/transfer_test_image", "transfer_test_image", user="sftpUser", password="sftpPasswd")
```

Transfer using tftp:

```
c = transfer("tftp", "10.193.190.100", "/transfer_test_image", "transfer_test_image", user="", password="")
```

Transfer using ftp:

```
c = transfer("ftp", "10.193.190.51", "golden/home/su-ash/transfer_test_image", "transfer_test_image", user="ftpUser", password="ftpPasswd")
```

CLI()

Synopsis

CLI() - Class Object

Syntax

```
CLI (<command>, <do_print>)
```

Send document comments to nexus5k-docfeedback@cisco.com.

Description

Instantiates an object of the CLI class with the CLI command specified in <command>. <do_print> when set to False does not print the output of the command and prints the output when set to True, which is the default.



Note

The CLI command name is case sensitive (uppercase).

Parameters

command, do_print

Returns

An object of CLI class.

Example

```
c = CLI ('show runn inter eth1/1')
```

Sample Output

```
!Command: show running-config interface Ethernet1/1
!Time: Mon Feb 27 14:33:24 2012

version 5.0(3)U3(1)

interface Ethernet1/1
  switchport mode trunk
  udld enable
  channel-group 12

<cisco.CLI object at 0xb7ae948c>
```

get_output()

Synopsis

```
get_output()
```

Syntax

```
CLI.get_output()
```

Description

Returns the output of the CLI command.

Parameters

None.

Returns

Output of the CLI command.

Example

```
c = CLI ('show runn inter eth1/1')
c.get_output()
```


Send document comments to nexus5k-docfeedback@cisco.com.

Sample Output

```
['', '!Command: show running-config interface Ethernet1/1', '!Time: Mon Feb 27 14:36:10  
2012', '', 'version 5.0(3)U3(1)', '', 'interface Ethernet1/1', ' switchport mode trunk',  
' uddl enable', ' channel-group 12', '', '']
```

rerun()

Synopsis

```
rerun()
```

Syntax

```
CLI.rerun()
```

Description

Reruns the command.

Parameters

None.

Returns

None.

Example

```
c = CLI ('show runn inter eth1/1')  
c.rerun()
```

Sample Output

```
!Command: show running-config interface Ethernet1/1  
!Time: Mon Feb 27 14:37:05 2012  
  
version 5.0(3)U3(1)  
  
interface Ethernet1/1  
  switchport mode trunk  
  uddl enable  
  channel-group 12
```

History()

Synopsis

History() - Class Object

Syntax

```
History()
```

Description

Instantiates an object of the History class.

Send document comments to nexus5k-docfeedback@cisco.com.

Parameters

None.

Returns

An object of History class.

Example

```
a = History()
```

`get_history()`

Synopsis

```
get_history()
```

Syntax

```
History.get_history()
```

Description

Gets the history of CLI commands executed so far.

Parameters

None.

Returns

Returns the history of commands executed.

Example

```
a = History()
a.get_history()
```

`clear_history()`

Synopsis

```
clear_history()
```

Syntax

```
History.clear_history()
```

Description

Clears history.

Parameters

None.

Returns

None.

Example

Send document comments to nexus5k-docfeedback@cisco.com.

```
a = History()
a.clear_history()
```

■ `clear_history()`

Send document comments to nexus5k-docfeedback@cisco.com.