



CHAPTER 2

アプリケーション プログラミング インターフェイス (API) 関数

この章では、次の Python アプリケーション プログラミング インターフェイス (API) 関数について説明します。この章は、次の内容で構成されています。

- 「Routes()」 (P.2-1)
- 「show_arp_table()」 (P.2-2)
- 「show_vsh_routes()」 (P.2-3)
- 「show_hw_routes()」 (P.2-3)
- 「verify_routes()」 (P.2-4)
- 「verify_arp_table()」 (P.2-5)
- 「CheckPortDiscards()」 (P.2-6)
- 「クラス BufferDepthMonitor(CLI)」 (P.2-7)
- 「get_total_instant_usage()」 (P.2-7)
- 「get_remaining_instant_usage()」 (P.2-8)
- 「get_max_cell_usage()」 (P.2-8)
- 「get_switch_cell_count()」 (P.2-9)
- 「transfer()」 (P.2-10)
- 「CLI()」 (P.2-10)
- 「get_output()」 (P.2-11)
- 「rerun()」 (P.2-12)
- 「History()」 (P.2-12)
- 「get_history()」 (P.2-13)
- 「clear_history()」 (P.2-13)

Routes()

概要

Routes() - Class Object

■ show_arp_table()

構文

```
Routes ()
```

説明

Routes クラスのオブジェクトをインスタンス化します。

パラメータ

なし。

戻り値

Routes クラスのオブジェクト。

例

```
rObj = Routes ()
```

show_arp_table()

概要

```
show_arp_table ()
```

構文

```
Routes.show_arp_table ()
```

説明

show ip arp コマンドを実行し、出力を返します。

パラメータ

なし。

戻り値

スイッチの ARP テーブル エントリを返します。

例

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

出力例

```
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()

概要

```
show_vsh_routes()
```

構文

```
Routes.show_vsh_routes()
```

説明

show ip fib route を実行し、出力を返します。

パラメータ

なし。

戻り値

ソフトウェア ルート エントリを返します。

例

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

出力例

```
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()

概要

```
show_hw_routes()
```

構文

```
Routes.show_hw_routes()
```

説明

ハードウェア ルートを計算し、出力を返します。

パラメータ

なし。

戻り値

ハードウェア ルート エントリを返します。

例

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

出力例

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()

概要

```
verify_routes()
```

構文

```
Routes.verify_routes()
```

説明

ソフトウェアとハードウェアのルートを確認します。

パラメータ

なし。

戻り値

ハードウェアとソフトウェアの間で一致したルートおよび一致しないルートの数を返します。

例

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

出力例

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

```
Routes not found:
```

```
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()

概要

```
verify_arp_table()
```

構文

```
Routes.verify_arp_table()
```

説明

ソフトウェアとハードウェアの ARP テーブル エントリを確認します。

パラメータ

なし。

戻り値

ハードウェアとソフトウェアの間で一致した ARP テーブル エントリおよび一致しなかった ARP テーブル エントリの数を返します。

例

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

出力例

```
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
```

CheckPortDiscards()

概要

CheckPortDiscards (<port>)

構文

CheckPortDiscards ('ethernet1/1')

説明

特定のポートでの入力廃棄を確認します。廃棄が 0 を超える場合は、broadcom から廃棄の理由を照会して印刷します。

パラメータ

port

戻り値

なし。

例

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

出力例

```
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
```


■ `get_remaining_instant_usage()`**説明**

show hardware internal buffer info pkt-stats コマンドの出力からその瞬間の合計使用量を返すメソッド。

パラメータ

なし。

戻り値

その瞬間の合計使用量を返します。

例

```
b = BufferDepthMonitor()
b.get_total_instant_usage()
```

出力例

0

get_remaining_instant_usage()

概要

```
get_remaining_instant_usage()
```

構文

```
monitorObj = BufferDepthMonitor()
remUsage = monitorObj.get_remaining_instant_usage()
```

説明

show hardware internal buffer info pkt-stats コマンドの出力からその瞬間の残りの使用量を返すメソッド。

パラメータ

なし。

戻り値

その瞬間の合計使用量を返します。

例

```
b = BufferDepthMonitor()
b.get_remaining_instant_usage()
```

出力例

46080

get_max_cell_usage()

概要

```
get_max_cell_usage()
```


構文

```
monitorObj = BufferDepthMonitor()  
cellUsage = monitorObj.get_max_cell_usage()
```

説明

show hardware internal buffer info pkt-stats コマンドの出力からセル使用量を返すメソッド。

パラメータ

なし。

戻り値

その瞬間の合計使用量を返します。

例

```
b = BufferDepthMonitor()  
b.get_max_cell_usage()
```

出力例

19

get_switch_cell_count()

概要

```
get_switch_cell_count()
```

構文

```
monitorObj = BufferDepthMonitor()  
cellCount = monitorObj.get_switch_cell_count()
```

説明

show hardware internal buffer info pkt-stats コマンドの出力からセル カウントの使用量を返すメソッド。

パラメータ

なし。

戻り値

その瞬間の合計使用量を返します。

例

```
b = BufferDepthMonitor()  
b.get_switch_cell_count()
```

出力例

46080

transfer()

概要

transfer()

構文

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

説明

<source> で指定されたファイルを <host> から <protocol> を使用して <dest> に記載されているパスに転送する API。プロトコルには、**scp**、**tftp**、**ftp**、または **sftp** を使用できます。

パラメータ

protocol、host、source、dest、vrf、login_timeout、user、password。

戻り値

転送が成功した場合は **True** を返します。

例

scp を使用する転送:

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

sftp を使用する転送:

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

tftp を使用する転送:

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

ftp を使用する転送:

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

CLI()

概要

CLI() - Class Object

構文

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

説明

<command> で指定された CLI コマンドで CLI クラスのオブジェクトをインスタンス化します。

<do_print> を **False** に設定するとコマンドの出力が印刷されず、**True** (デフォルト) に設定すると出力が印刷されます。

パラメータ

command、do_print

戻り値

CLI クラスのオブジェクト。

例

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

出力例

```
!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
  uddl enable
  channel-group 12

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

get_output()

概要

get_output()

構文

```
CLI.get_output()
```

説明

CLI コマンドの出力を返します。

パラメータ

なし。

戻り値

CLI コマンドの出力。

例

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

出力例

```
['', '!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()

概要

rerun()

構文

```
CLI.rerun()
```

説明

コマンドを再実行します。

パラメータ

なし。

戻り値

なし。

例

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

出力例

```
!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
  udld enable
  channel-group 12
```

History()

概要

History() - クラス オブジェクト

構文

```
History()
```

説明

History クラスのオブジェクトをインスタンス化します。

パラメータ

なし。

戻り値

History クラスのオブジェクト。

例

```
a = History()
```

get_history()

概要

```
get_history()
```

構文

```
History.get_history()
```

説明

これまでに実行された CLI コマンドの履歴を取得します。

パラメータ

なし。

戻り値

実行されたコマンドの履歴を返します。

例

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

clear_history()

概要

```
clear_history()
```

構文

```
History.clear_history()
```

説明

履歴をクリアします。

パラメータ

なし。

戻り値

なし。

例

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

■ clear_history()